This page is part of a static HTML representation of the TiddlyWiki at https://tiddlywiki.com/

"A free, open source wiki revisited" by Mark Gibbs, NetworkWorld

4th February 2016 at 11:53pm

Articles

Interesting article giving the perspective of someone who has been away from TiddlyWiki for a few years:

http://www.networkworld.com/article/3028098/open-source-tools/tiddlywiki-a-free-open-source-wiki-revisited.html

> Way back in the mists of time (actually, January 2009) I wrote about a really cool tool called TiddlyWiki, a “non-linear personal web notebook”. Fast forward to today and I just had an out of body experience: Completely by accident I found a TiddlyWiki that I started when I wrote that piece and it still works!
>
> Finding code that works flawlessly after just two or three years is magical enough but after seven years?! And given that TiddlyWiki is written as a single page Web application and considering how different browsers are now than they were in 2009, the fact that the old version of TiddlyWiki still works is not short of miraculous.

"A Thesis Notebook" by Alberto Molina

2nd March 2013 at 9:45am

Examples

A thesis notebook based on TiddlyWiki.

http://tesis.tiddlyspot.com/

> This is an example of a thesis notebook powered by TiddlyWiki 5.0.8-beta.
>
> TiddlyWiki is a great piece of software created by Jeremy Ruston. It allows you, among other things, to take notes, organise ideas, store information, and display all your stuff the way you want. It is an incredibly flexible tool you can adapt to fit almost all your needs.
>
> This TiddlyWiki has been customized to serve as a philosophy notebook centered around authors, books and papers, concepts and theories, and personal notes. I use it along with Zotero, which is a dedicated bibliography software. Both are free, open source projects. TiddlyWiki can be downloaded at https://tiddlywiki.com.

"ATWiki" by Lamusia Project

6th January 2021 at 4:10pm

Other Resources

A desktop application for TiddlyWiki, running on Windows.

https://lamusia.github.io/#ATWiki

> TiddlyWiki for Windows. A unique non-linear notebook for capturing, organising and sharing complex information.

"BJTools" by buggyj

6th January 2021 at 4:10pm

Community Plugins

buggyj has created several useful plugins, including a WYSIWYG HTML editor, a configurable calendar and tag lists with draggable ordering.

http://bjtools.tiddlyspot.com

"BrainTest - tools for a digital brain" by Danielo Rodriguez

6th January 2021 at 4:10pm

Other Resources

A collection of customisations and macros from Danielo Rodriguez.

http://braintest.tiddlyspot.com/

> This is my personal tiddlywiki file for testing.
>
> Here I will post the tools and Macros that I develop for this awesome tool. That way, all the help the community gave me can came back to the community again.

"Cardo - Task and Project Management Wiki" by David Szego

6th January 2021 at 4:10pm

Community Editions

Task & Delegation Tracking, Meetings & Agenda Items, Project Roles, Progress & Status Updates, Reference Items, Tickler Calendar, Conversation Logging, Book Notes & Library, and More!

http://cardo.wiki

> Cardo is a standalone, browser-based tool that can be used as a simple task manager, or as a complex Project Management

> system (and indeed, I do use it this way in my daily work) as well as a fully Wiki-ized personal knowledge store. It runs completely independently in the browser, even without an Internet connection, making it possible to carry around on a USB stick, or to use on the morning commute.

"CouchDB Adaptor" by William Shallum

6th January 2021 at 4:10pm

Other Resources

An adaptor to enable TiddlyWiki to sync changes with a CouchDB database.

https://github.com/wshallum/couchadaptor

> *CouchDB* sync adaptor for TiddlyWiki 5. Requires TiddlyWiki >= 5.1.2.
>
> Works in limited testing. Not sure how well it handles conflicts.

"Creating a baby journal with TiddlyWiki 5" from A Penguin in Redmond

20th February 2014 at 9:45am

Tutorials

A guide to creating a baby journal with TiddlyWiki.

http://a-penguin-in-redmond.blogspot.co.uk/2014/02/creating-baby-journal-with-tiddlywiki-5.html

> Keeping a journal about your baby is fun. It allows you to keep track of those events in your baby's life and development that are most important to you, and to tell them in your own words. [...]
>
> There are plenty of online services that can help to keep such a diary, but you need to have an internet connection to access them, and you have to trust them to keep your data safe.

"Dracula - a dark palette for TiddlyWiki" by SlyBouhafs & zenorocha

1st January 2021 at 5:36pm

Community Palettes

A dark palette for TiddlyWiki.

https://draculatheme.com/tiddlywiki

> Install manually
>
> Open the Control Panel > Appereance > Palette, scroll down and clone a platte, rename it to Dracula and copy the code from the website.
>
> Activating theme
>
> - Open the Control Panel > Appereance > Palette
> - Scroll down and select Dracula

> - Boom! It's working

"Drift - Collect, Organise, Grow." by Tony K

10th January 2023 at 11:00pm

Community Editions

A adaptation of TiddlyWiki perfect for using as a Notebook sysetem.

https://github.com/bmann/drift-tiddlywiki-template/tree/master/drift

> Drift is an adaptation of TiddlyWiki with the goal of helping you Collect. Organise. and Grow. your ideas while keeping tab on how they interconnect together and fluidly Drift from one to another.
>
> The 2 main components of Drift are TWCrosslinks and *DailyNotes* so you never miss an idea
>
> Drift was inspired by *TiddlyBlink* / Stroll and adapted to fit my needs.

"Dropboard" by Reid Gould

6th January 2021 at 4:10pm

Other Resources

A TiddlyWiki plugin for kanban-like organisation in the style of Trello™ using boards, lists, and cards.

https://github.com/reidgould/tiddlywiki-dropboard

"Dynamic Tables" by Jed Carty

13th November 2021 at 2:06am

Other Resources

Dynamic Tables is a set of macros that allow you to create tables where the rows are created from tiddlers and the columns from the tiddler fields, or where the columns are created from tiddlers and the rows from their fields. Individual cells can be edited on the fly. Row tables can be sorted by columns and column tables can be sorted by rows.

http://ooktech.com/jed/ExampleWikis/DynamicTables/

"Encrypt single tiddler plugin" by Danielo Rodriguez

13th November 2021 at 2:06am

Community Plugins

A plugin that allows to encrypt your tiddlers individually Danielo Rodriguez.

http://danielorodriguez.com/TW5-EncryptTiddlerPlugin/

> Advantages:
>
> - You can specify a different password for each tiddler if you want.
> - You don't have to encrypt your whole wiky.
> - If you forget your password, you only lose a tiddler.
> - It's possible to edit the tiddler content , tags and fields except the encrypt field after encryption.

> - [...]

"file-backups" Extension for Firefox by pmario

6th January 2021 at 4:10pm

Firefox Linux Mac Other Resources plugins Saving Windows

An extension for Mozilla Firefox that smoothes out some of the friction from TiddlyWiki's built-in HTML5 saver, making it almost as easy to use as TiddlyFox. The workflow is intended to work out of the box, without configuration.

https://github.com/pmario/file-backups which contains links to the documentation and introduction video(s).

A chrome version is planned!

"Filter Examples" by Tobias Beer

6th January 2021 at 4:10pm

Community Resources Tutorials

This wiki gives examples for various filters and their use in the list widget

http://tobibeer.github.io/tw/filters/#Filter%20Examples

> This wiki gives examples for various filters and their use in the list widget. A good starting point to understand what filters yield which results also is test-filters.js.

"Font Awesome 5 Free SVGs for TiddlyWiki" by morosanuae

6th January 2021 at 4:10pm

Font Awesome 5 Images Other Resources SVG

The **Font Awesome 5 Free** SVG images collection converted to tiddlers that you can easily import (drag'n'drop) in your wiki.

http://fa5-free-svg.tiddlyspot.com

"Ghostwriter theme by Rory Gibson" adapted for TW5 by Riz

6th January 2021 at 4:10pm

Community Themes

An adaption of the "Ghostwriter" theme for TiddlyWiki.

https://ibnishak.github.io/ghostwriter/

> Ghostwriter is a popular, minimalist and mobile responsive theme for the blogging platform ghost. This is an adaptation for TiddlyWiki.

"Gospel Bubbles" by Rev. David Gifford

6th January 2021 at 4:10pm

Examples

Creates bubble maps of Biblical texts.

http://giffmex.org/gospels.bubbles.html#Gospels%20Bubbles

> This TiddlyWiki, which is still under construction, has several features that together make it lightning fast to find passages, types of passages, themes, people, places and images in the New Testament Gospels. The bubblemaps are color-coded overviews of each Gospel. Each type of passage (miracle story, parable, etc) was assigned a color. Bubblemaps allow you to do three things:
>
> 1. The **bubblemaps** are color-coded overviews of each Gospel. Each type of passage (miracle story, parable, etc) was assigned a color. Bubblemaps allow you to do three things:
>   1. See how the different types of passages are distributed throughout each Gospel.
>   2. Hover over passages to see the verses and titles.
>   3. Click on a passage to open it and see the themes and other data it mentions.
> 2. The **themes by passage** indexes are lists of the passages in each Gospel, followed by the themes and other data found in the passage. Clicking on a theme calls up a list of all the passages that mention that theme.
> 3. The **indexes of themes** contain lists of links to each theme, person, group, place and image. Clicking on a theme calls up a list of all the passages that mention that theme.
> 4. There is a **search window** hidden in the right hand menu. Use the search window to find passages quickly, by typing things like sower or paralytic. Click the arrows in the upper right of the screen to open the menu and search.

"Grok TiddlyWiki" by Soren Bjornstad

19th May 2021 at 1:02pm

Other Resources

This new textbook from Soren Bjornstad is highly recommended for learning TiddlyWiki. The presentation and design are also a first class example of using TiddlyWiki.

From the site:

> Grok TiddlyWiki is a textbook that helps you build a deep, lasting understanding of and proficiency with TiddlyWiki through a combination of detailed explanations, practical exercises, and spaced-repetition reviews of prompts called takeaways.

https://groktiddlywiki.com/read/

"GSD5" by Roma Hicks

26th February 2022 at 9:52pm

Community Editions Community Plugins

An adaptation of the TiddlyWiki powered GTD® system formerly known as MonkeyGTD for TiddlyWiki version 5.

http://gsd5.tiddlyhost.com/

> GSD5 is a Getting-Thing-Done tool for TiddlyWiki5 based off the mGSD classic TiddlyWiki.

"Hacks" by Thomas Elmiger

6th January 2021 at 4:10pm

Other Resources

A collection of helpers by Thomas Elmiger, among them a **tweet button** macro, a **text-stretch** solution, a **snippet extraction** macro and a tiddler that installs a **scroll-to-top button** via drag-and-drop.

http://tid.li/tw5/hacks.html

> Quick Extensions for TiddlyWiki 5 – […] As I take from the community, I am happy if I can give back. You can use my stuff to do whatever you like, but remember to save a back-up first.
>
> *Thomas*

"Heeg.ru" by sini-Kit

6th January 2021 at 4:10pm

Tutorials

A guide in Russian to creating an online shop with TiddlyWiki. No serverside code is needed, instead orders are handled by Disqus.

http://heeg.ru/

"In My Socks" by Jed Carty

6th January 2021 at 4:10pm

Other Resources

An extensive collection of hints and tips from Jed Carty.

http://inmysocks.tiddlyspot.com

> I made this so that some of my friends with absolutely no coding experience could use some of the better features of TW5 without having to learn much. I am hopefully going to continue updating it as I learn new things. There isn't really anything here that isn't available in plenty of other places, but I figured that since I spent a while putting it together and people keep saying that TW needs more documentation I would post it here.

"JD Mobile Layout plugin" by JD

6th January 2021 at 4:10pm

Community Plugins

A plugin that improves the usability of TiddlyWiki on smartphones by JD:

http://j.d.simplemobile.tiddlyspot.com

"JD's palettes" by JD

1st January 2021 at 5:45pm

Community Palettes

Some palettes by JD.

http://j.d.palette.tiddlyspot.com/

> You can install any or all of these by dragging any of the below links into your Tiddlywiki:
>
> - Brown One
> - C8 Dark
> - Cool Mint
> - Neonize
> - Spartan Day
> - Spartan Night

"Lucky Sushi" online shop by sini-Kit

6th January 2021 at 4:10pm

Examples

A complete online shop made in TiddlyWiki!

http://luckysushi.ru/habarovsk/heeg.html#index

"Mal's Sandbox" by Mal

6th January 2021 at 4:10pm

Community Plugins

A collection of extensions by @malgam

http://malsandbox.tiddlyspot.com

> The "loadnew" plugin provides the JavaScript module *$:/plugins/malgam/loadnew/loadnew.js* that implements an incremental load command for Tiddlywiki 5.
>
> The javascript code is based on the built-in module **$:/core/modules/commands/load.js** that loads tiddlers from a Tiddlywiki file into a node.js Tiddlywiki instance. This command loads all non-system tiddlers, irrespective of whether they have been modified since last loaded. As a result, all tiddlers in the node.js tiddlers folder will end up with a new modification date.
>
> In contrast, the loadnew command only loads tiddlers that either don't already exist in the tiddlers folder, or have a newer modified date than the existing tiddler.

> With a backup strategy that uses a file system snapshot process to preserve historical versions of tiddlers, the loadnew command minimises the sizes of the resulting snapshots.

"MathCell" by Stephen Kimmel

6th January 2021 at 4:10pm

Other Resources

Mathcell allows spreadsheet-like abilities in a completely contained TiddlyWiki.

http://mathcell.tiddlyspot.com/

> Mathcell allows spreadsheet-like abilities in a completely contained TiddlyWiki. Additional abilities allow it work as a general math engine.

"Microblogging con Tiddlywiki" by Juan

8th October 2014 at 3:43pm

Tutorials

A Spanish guide to using TiddlyWiki on Android.

http://3palmeras.wordpress.com/2014/11/05/microblogging-con-tiddlywiki/

> Tiddlywiki software is prodigious based on javascript which is capable of holding a notepad that can be edited with Internet Explorer or function as a full blog, contained in a single file.

Translation by Google Translate

"Moments: A little color won't hurt" by Riz

6th January 2021 at 4:10pm

Community Themes

An adaption of the "Moments" theme for TiddlyWiki.

http://tesseractmoments.surge.sh/

> Moments is a beautiful free and responsive theme available for wordpress from here: http://www.s5themes.com/theme/moments/. This is an adaptation of the same for TW5 platform. All the design credits goes to the site and the original designer. If TW5 is going to be here for 25 years, might as well suit up.

"Mono theme" by JD

1st January 2021 at 4:46pm

Community Themes

This theme by JD features a left side bar, advanced functionality from the search bar and plenty of customizability built in.

http://j.d.mono.tiddlyspot.com/

"muritest" by Simon Huber

14th November 2021 at 12:06am

Other Resources

A terrific demo of drag and drop, pan and zoom, and other sophisticated touch effects. Based on the *muuri plugin by Uwe Stuehler*.

http://muritest.tiddlyspot.com

"Notebook theme" by Nicolas Petton

31st January 2024 at 3:33pm

Community Themes

Notebook is a clean, uncluttered theme for TiddlyWiki.

https://saqimtiaz.github.io/sq-tw/notebook.html

> Notebook is a clean, uncluttered theme for TiddlyWiki.
>
> - The theme is responsive and comes with mobile support through the included notebook-mobile plugin.
> - Notebook comes with 3 default colour palettes, a grey, beige and dark one. To switch between colour palettes, click the color palette switch button in the top bar.
> - Notebook uses a custom left sidebar, with another sticky bar at the top of the page (or at the bottom on mobile).

"Noteself" by Danielo Rodríguez

13th November 2021 at 2:08am

Android Chrome Community Editions Edge Firefox Internet Explorer iOS Linux Mac Opera Other Resources PHP Safari Saving Windows

NoteSelf is your personal, private, customisable, Evernote-like experience. You want cloud? Fine! You don't? Fine too! It's all yours, It's your decision! Why not be a bit selfish?

https://noteself.github.io/

> NoteSelf is the perfect place to store your ideas, notes, thoughts, tips, tricks, recipes... whatever you want to put into it! NoteSelf will store, index, and make it easily searchable, so you will find it instantly whenever you need it!
>
> You already know and love Evernote, we know it. It is comfortable, it syncs, has search capabilities... Wow, it's almost perfect, but what about:
>
> - Privacy - Everything is stored on the Evernote's servers!
> - Customisation - If you don't like the interface of Evernote, there's nothing you can do
>
> NoteSelf is built on top of TiddlyWiki, a powerful, free, highly customisable and open-source personal wiki. We took the best of it, it's powerful customisation system, and mixed it with one of the best embedded databases available, PouchDb, for bringing the synchronisation capabilities you need.

"Notizen mit TiddlyWiki systemübergreifend nutzen" by Michael Sonntag

10th September 2014 at 12:28pm

Articles

A quick guide to using TiddlyWiki (written in German)

http://michaelsonntag.net/notizen-mit-tiddlywiki-systemuebergreifend-nutzen

> Those using multiple different computer platforms (if only PC and Android) knows for certain: You would like to sync notes across computers and thereby be independent of any apps or services. I am concerned at any rate so and so I am constantly looking for the perfect solution.
>
> With TiddlyWiki I found it mostly. As the name suggests, it TiddlyWiki is a Wikisystem. In contrast to all other wikis, TiddlyWiki is a single HTML file that runs in the browser and thus can be used on all modern operating systems. And because there is only one file, it can be very easily via the well-known cloud services (Dropbox, Google Drive, ownCloud) or synchronized with FTP or used on a USB stick.

(Translation from German by Google Translate)

"PETTIL - Forth for the Commodore PET" by Charlie Hitselberger

16th July 2014 at 10:45am

Examples

A fast Forth interpreter for the Commodore PET, written in 6502 assembly language. The TiddlyWiki containing program documentation is automatically generated from the source code: see https://github.com/chitselb/pettil.

http://pettilmobile.com/tw/

> Objectives of the project are, in no particular order:
>
> - make a Forth that runs on my Commodore PET 2001
> - have fun

> - improve my "6502 assembly golf" skills
> - find other people who are interested in this project

"Reveal.js" by Devin Weaver

14th November 2021 at 12:05am

Other Resources Resources

Embed Reveal.js presentations inside your TiddlyWiki.

https://sukima.github.io/tiddlywiki-reveal-js/

> reveal.js enables you to create beautiful interactive slide decks using HTML.
>
> With the power of TiddlyWiki and Reveal.js one can…
>
> - …make an interactive tiddler with fancy animations
> - …Make it go full screen for professional keynote
> - …Distribute the slideshow with examples and additional resources in one HTML file

"savetiddlers" Extension for Chrome and Firefox by buggyj

6th January 2021 at 4:10pm

Chrome Firefox Other Resources plugins Saving

An extension for Google Chrome and Mozilla Firefox that smoothes out some of the friction from TiddlyWiki's built-in HTML5 saver, making it almost as easy to use as TiddlyFox once it is set up correctly.

https://github.com/buggyj/savetiddlers

"Setting Up a Personal TiddlyWiki Server on OS X" by Kris Johnson

3rd April 2015 at 12:42pm

Articles

Detailed description of setting up TiddlyWiki on Node.js on OS X.

http://undefinedvalue.com/2015/04/02/setting-personal-tiddlywiki-server-os-x

> For a new job, I decided to set up a personal wiki to keep notes. I wanted to keep it simple, meeting these requirements:
>
> - All the data is in a Dropbox folder (so it can be automatically synced between machines)
> - It must support Markdown syntax
>
> After looking at the options, I settled on TiddlyWiki. I've used "classic TiddlyWiki" before, and liked its simplicity, but I was always a little annoyed with the weird steps you have to go through to save changes. The new version of TiddlyWiki includes support for running it as a real HTTP server, so you can use it just like an online wiki.
>
> But it took me a couple of hours to figure out how to set that up. The TiddlyWiki documentation is not clear ("not clear" is a euphemistic way of saying "terrible"). So, I've written up these instructions in the hope it will spare somebody else all the frustration I had.

"Shining Ark Using TiddlyWiki" by Helu

7th September 2013 at 10:45am

Tutorials

A gamers guide to using TiddlyWiki.

http://helugame.wordpress.com/2013/09/07/shining-ark-7-using-tiddlywiki/

> I've been using TiddlyWiki for this game mainly to record the monsters' locations and the items that they drop. This is so that when I need a certain item to upgrade/create weapons, I can quickly look it up; which monsters drop it and which quests have those monsters.
>
> As I play the game, as I encounter new information, I just input them into my Shining Ark TiddlyWiki. It's like I'm building a wiki from scratch. The reason why I don't just use online wikis is that 1) it's too easy to get spoiled and 2) sometimes online wikis provide too much information (who cares about the non-gameplay related descriptions that can be found in the game, etc).

"SK Plugins" by Stephen Kimmel

6th January 2021 at 4:10pm

Community Plugins

A collection of plugins by Stephen Kimmel, including a number of additional editor toolbar buttons.

http://skplugins.tiddlyspot.com/

> In keeping with the general spirit of the TiddlyWiki community, ie, you develop something useful and/or potentially interesting you

> share it with others, I'm sharing some of the plugins I've developed to go with the new Editor Toolbar. You'll find them and a more extensive discussion of what each does at http://skplugins.tiddlyspot.com/

"Slides and Stories" by Jan

6th January 2021 at 4:10pm

Community Plugins

Slides and Stories is a repository of the tools to optimise TW for scientific research and for presenting and publishing results.

http://slidesnstories.tiddlyspot.com

> ## SlidesnStories
>
> ...is a tool to save and reload sets of Tiddlers. This is already working, The aim is to develop a mechanism to transform these into instant slideshows.
>
> ## MetaTabs
>
> is a tool which gives you the possibility to attach and show specific information in tabs at the bottom of each Tiddler: ideas, notes, tasks, stories and presentations, comments, annotations and footnotes.
> It is inspired by MagicTabs and Tiddlyscholar by Alberto Molina Pérez, which are very elaborated tools that alas do not seem to be in active development any more. I wanted to change the behaviour in some ways. For example I wanted show tabs only if they have content.

> ...furthermore there is
>
> - an export-plugin,
> - a footnote-plugin,
> - and a plugin to rearrange Tiddlers in the StoryRiver

"Stroll - a Roam-like experience" by David Gifford

1st January 2021 at 9:14pm

Community Editions

A adaptation of TiddlyWiki perfect for using as a Notebook sysetem.

https://giffmex.org/stroll/stroll.html

> Stroll is a TiddlyWiki adaptation imitating certain features of Roam to help users easily see connections between their notes and navigate between them. Features include:
>
> - bi-directional links,
> - autocompletion when linking,
> - renaming of links upon changing tiddler titles,
> - side-by-side editing of multiple notes.

"TB5 - a pocket full of tips" by Tobias Beer

6th January 2021 at 4:10pm
Other Resources Tutorials

An enormous collection of tips beautifully curated by Tobias Beer. It is actually a collection of sites, with sub-sites dedicated to topics such as filtering (http://tobibeer.github.io/tw/filters/).

http://tobibeer.github.io/tb5/

> A collection of references and sandbox for testing concepts around TiddlyWiki 5...

"Tekan: Kanban that stays with you" by Riz

6th January 2021 at 4:10pm
Community Plugins

A Kanban implementation for TiddlyWiki with a familiar look and feel (see the demo).

https://ibnishak.github.io/Tesseract/projects/tekan/Tekan.html

> This is Tekan, a TW based kanban. The easiest way to know tekan is to use it. So let us begin, shall we?

"TiddlyServer" by Matt Lauber

14th November 2021 at 12:07am
Other Resources

Note: This is an older version of TiddlyServer. A modern version can be referenced here.

TiddlyServer is a special purpose Desktop app, designed to facilitate managing multiple instances of TiddlyWiki running as a server. It does not require internet acess to access the wikis.

https://github.com/mklauber/TiddlyServer/releases/

> *TiddlyServer* can import both TiddlyWiki files and *TiddlyFolder* wikis. For each wiki, you specify a prefix to serve it with and the source to import from. It will copy the wikis to its own internal store and begin serving them up at http://localhost:8080/{prefix}/. The export button for each wiki will convert it to a single file wiki.

"TiddlyStudy" by Kebi

10th January 2023 at 11:04pm

Community Editions

A adaptation of TiddlyWiki perfect for using as a Notebook sysetem.

https://postkevone.github.io/tiddlystudy/

> Tiddly Research is a free and local note-taking environment based on TiddlyWiki
>
> Core Features:
>
> - Bidirectional note linking on multiple levels
> - Tags
> - Linked references
> - Unlinked references
> - Transclusions

> - Bullet point based hierarchical note taking structure
> - Integration with the spaced repetition flashcards tool Anki
> - TODOs
> - and more

"TiddlyWiki 5 im Betatest" by besim

5th January 2014 at 9:45am

Articles

Reaction to TiddlyWiki5 from the translator of the German edition of TiddlyWikiClassic.

http://blog.netplanet.org/2014/01/03/tiddlywiki-5-im-betatest/

> There is little software that can still inspire me so even after years, as on the first day. This includes TiddlyWiki, the "pocket-wiki". The latter is because the Wiki completely fits into an HTML file, and this HTML file brings everything - JavaScript program logic, CSS appearance and the entire Wiki content as stored records. So a TiddlyWiki file is then sometimes happy times some megabytes in size, but just has the unbeatable advantage that it works just as fast file created locally in a variety of browsers.

(Translation from German by Google Translate)

"TiddlyWiki for Scholars" by Alberto Molina

6th January 2021 at 4:10pm

Community Editions

An early release of an edition of TiddlyWiki customised for rigorous, academic notetaking.

http://tw5.scholars.tiddlyspot.com/

> TiddlyWiki for Scholars is a personal customisation of TiddlyWiki 5.0.13-beta I made for note-taking from readings, but it can be useful for other purposes. The idea behind this adaptation is to show and to allow creating relevant data related to the current tiddler without leaving the tiddler.

"TiddlyWiki guide FR" by Sylvain Naudin

6th January 2021 at 4:10pm

Other Resources

A collection of tips and guides in French, introduced by this blog post:

http://sylvain.naud.in/post/2014/10/29/Ressources-pour-TiddlyWiki-5-%28FR%29

> Ce guide de prise en main de TiddlyWiki vous permettra de débuter sur ce logiciel : rédiger du texte avec la syntaxe wiki, insérer des images, créer des Tiddler, faire une recherche, ajouter des fonctionnalités, et bien d'autres choses.. Bonne lecture ! Sylvain

"TiddlyWiki Knowledge Network" by Dmitry Sokolov

6th January 2021 at 4:10pm

Other Resources

A Knowledge Network about TiddlyWiki.

http://confocal-manawatu.pbworks.com/w/page/113574373/TiddlyWiki

> Interconnected with the visual taxonomy at every node: http://debategraph.org/Stream.aspx?nid=457381&vt=bubble&dc=focus
>
> TWKN will be transferred into TWederation/TiddlyMap format as soon as I am convinced in reliable and secure versioning and multi-user operation in TW format. Your assistance and support would be highly appreciated.

"TiddlyWiki Notes" by James Anderson

6th January 2021 at 4:10pm

Other Resources

Notes and tips by a developer working on writing TiddlyWiki plugins – including TWExe, a widget for running Windows scripts and executables.

http://welford.github.io

"TiddlyWiki Posts" by Jeffrey Kishner

5th August 2023 at 4:07pm

Articles Other Resources

A collection of articles covering integration with Fargo, Font Awesome and Google Calendar, and tips for managing task lists. The original site is missing, but a link to an archive is provided.

https://web.archive.org/web/20221015011644/http://blog.jeffreykishner.com/tiddlywiki/

> I have become a regular user of TiddlyWiki and have become so accustomed to using Font Awesome icons in Fargo that I wanted

> to incorporate the icons into my wikis as well.

"TiddlyWiki Toolmap" by David Gifford

6th January 2021 at 4:10pm

Other Resources

A topical index of known plugins, tutorials, themes, and other helpful tools for TiddlyWiki.

https://dynalist.io/d/zUP-nIWu2FFoXH-oM7L7d9DM

Some of the topics indexed so far include:

- Background images and background colors
- Breadcrumbs
- Business - examples
- CSS / Styles, general
- Calendars, dates and timelines
- Checklists
- Collaboration
- Color palettes
- Colors
- Data visualizations
- Developer / advanced tools
- Diplay tiddlers horizontally

- Dropdowns
- Educational
- External files
- Fields
- Fonts
- Fun and games
- Home and lifestyle
- IFrames and embedding media
- Image galleries, modals and lightboxes
- Images, icons, emojis
- Importing and exporting between TiddlyWiki files
- Includes scaffolding within a tiddler, dragging to reorder tiddlers, choosing the number of columns
- Interfacing with, importing and exporting to and from other formats
- Knowledge bases, dictionaries and glossaries
- Layout/UI
- Lightboxes
- Links, linking
- Lists and filters
- Maps and geography
- Masonry, scaffolding
- Math
- MindMapping

- Mobile tools
- Music and sound
- Navigation
- node.js
- Note-taking
- Open the "filter" tab
- Partially hidden tiddlers
- Personal websites (examples)
- Presentations and slideshows
- Printing
- Project management
- Publishing and hosting *TiddlyWikis* on the web
- Searching tools
- Security and backup tools
- Social media, sharing, comments
- Spreadsheets and math
- Startup
- Sticky titles
- Story and screenplay creation
- Table of contents
- Tables, charts and graphs (cf data visualizations)
- Tabs

- Tags, tagging
- Task management and productivity - to do lists
- Themes
- Understanding TiddlyWiki
- Writing and editing

"TiddlyWiki 舞" by Bram Chen

6th January 2021 at 4:10pm

Other Resources

A translation of the tiddlywiki.com documentation from Bram Chen, TiddlyWiki's Chinese translator.

http://tw5-zh.tiddlyspot.com

"TiddlyWiki" by Sander de Boer

27th November 2014 at 5:32pm

Articles

Dutch blog post about TiddlyWiki

https://breinbout.wordpress.com/2014/11/26/tiddlywiki/

> I'm allergic to websites that are highly dependent on JavaScript. I think that JavaScript has many useful applications to add functionality to a site, but content needs to be visible if JavaScript is switched off. I use the *NoScript* extension for Firefox and surf

> with pleasure over the Internet without spontaneous things happen that I do not give permission. I love JavaScript on a short leash. More crazier maybe I'm so excited about TiddlyWiki.

(Translation from Dutch by Google Translate)

"TiddlyWiki5 Bourbon" by mkt_memory

6th January 2021 at 4:10pm

Other Resources

An extensive collection of tips and resources for users of TiddlyWiki in Japanese.

http://bacchus.ivory.ne.jp/bourbon/

> TiddlyWikiをHTML5にしたTiddlyWiki5が正式にリリースされました。このTiddlyWiki5の使い方について、自分で試した機能を中心に解説します。
>
> TiddlyWiki5は、非常に多機能なローカルWikiです。このブログに書いてあることは、TiddlyWiki5で出来ることのごく一部です。

"TiddlyWiki5 Forum on Reddit" by Riz

6th January 2021 at 4:10pm

Other Resources

A TiddlyWiki5 forum on Reddit.

https://www.reddit.com/r/TiddlyWiki5/

> A TiddlyWiki5 forum on Reddit. As of 2016, it claims to have the most up-to-date list of plugins available for TW5.

"TiddlyWiki5 Playground" by Ton Gerner

6th January 2021 at 4:10pm

Community Plugins Other Resources

Huge collection of customisations and tweaks from Ton Gerner.

http://tongerner.tiddlyspot.com/

> I am a longtime user of TiddlyWiki Classic and still use it daily since you can't do everything in TiddlyWiki 5 (yet) ;)
>
> I started experimenting with TW5 at the end of September 2013 (alpha10). Since I missed a few things in the layout I got used to in TiddlyWiki Classic 1, I started with modifying the layout of TW5. To share my knowledge, I made a few guides about these 'modifications'. The guides started as a non-linear personal web notebook (yeah, the subtitle of TiddlyWiki!). I started experimenting with layout things and used TW5 to document my experiments.

"TiddlyWiki5^2 documenting while learning TiddlyWiki5" by Iannis Zannos

6th January 2021 at 4:10pm

Other Resources

A wealth of hints, tips and notes about using TiddlyWiki on Node.js:

> TiddlyWiki is different from other wikis because of its principle of dynamically customizeable "storyline" based on tiddlers as basic units of information. That is, the user "composes" their own version of the webpage by clicking on tiddler links, which add tiddlers to the page in order to compose a storyline.
>
> The Node.js implementation in TiddlyWiki5 adds all the advantages of flat-file markup language based type of site. This makes TiddlyWiki an excellent alternative to flat-file based CMS/webpage/blog authoring systems for the web.
>
> Also very cool is the treatment of tags as menus everywhere.

http://larigot.avarts.ionio.gr/users/iani/wikis/tw5square.html

Also available on GitHub (download and save index.html and open it in your browser).

"TW Icons" by morosanuae

13th November 2021 at 11:59pm

Icons Images Other Resources SVG

> A large collection of icons for TiddlyWiki.

https://morosanuae.github.io/tw-icons

"TW5 Magick" by Stephan Hradek

14th November 2021 at 9:53pm

Community Plugins Other Resources Tutorials

Stephan Hradek's growing catalogue of tips, tutorials, and other resources. Stephan has also developed several plugins and macros that you can find at http://tiddlystuff.tiddlyspot.com/.

http://tw5magick.tiddlyspot.com/

> Some things in TiddlyWiki seem like Magic. As my preferred nick is Skeeve, I felt that TW5 Magick is a proper name for this collection of some of the "magic" tricks one can do with TiddlyWiki.

"TW5 Tribal Knowledge" from Scott Kingery

21st March 2014 at 9:45am

Tutorials

Tips and guides for using TiddlyWiki.

http://techlifeweb.com/tw5tribalknowledge.html

> I never used Node.js until a couple weeks ago when I wanted to try a TW5 installation on Node and expand my knowledge a bit. I figured someone else may be in the same boat so I wrote it all down

"TW5-TeXZilla" plugin by Joe Renes

6th January 2021 at 4:10pm

Community Plugins

An alternative *LaTeX* plugin for TiddlyWiki5. It supports a wider set of *LaTeX* commands than KaTeX, though not as extensive as *MathJax*

http://tw5-texzilla.tiddlyspot.com

> This plugin provides *LaTeX* support in TiddlyWiki5, using the TeXZilla latex parser to generate MathML.
>
> Only Firefox and Safari support MathML at the moment, so the plugin is only useful if you're using one of these browsers. It might be possible to combine this plugin with *MathJax* to convert MathML to something that other browsers can understand, e.g. HTML+CSS, but I haven't tried this.

"TWeb.at" by Mario Pietsch

6th January 2021 at 4:10pm

Other Resources

A collection of TiddlyWiki resources from Mario Pietsch, with a focus on cloud deployments. Mario also maintains the German translation of TiddlyWiki.

https://tweb.at

> My name is Mario Pietsch from Austria. I'm living near Salzburg.
>
> This page, will be the portal to my TiddlyWiki, TiddlyWeb related content.

"Un wiki sur votre BiblioBox avec TiddlyWiki" from A Penguin in Redmond

16th July 2014 at 10:45am

Tutorials

A guide (in French) to using TiddlyWiki on the BiblioBox, a variation of the PirateBox mobile communication and file sharing system.

http://bibliobox.net/blog/post/2014/07/16/un-wiki-sur-votre-bibliobox-avec-tiddlywiki

> I've known TiddlyWiki since its beginning in 2004. For personal or professional needs, I used it regularly for notepad staff in 2007 and 2012. But I admit I ignored the tool during the past two years. And then I said yesterday: but why not put TiddlyWiki on a *BiblioBox*? Because it is a single HTML file with javascript. In addition I knew the existence of a server version for saving online amendments.

Hand optimised from a translation by Google Translate

"Whitespace, a negative-spacious, webapp-nostalgic theme" by JD

1st January 2021 at 4:46pm

Community Themes

A beautiful theme that leverages the white spaces between elements.

http://j.d.whitespace.tiddlyspot.com/

"Wikilabs" by PMario

6th January 2021 at 4:10pm

Community Editions Community Plugins Other Resources

A collection of references for testing concepts around TiddlyWiki 5...

https://wikilabs.github.io

> This is PMario's playground with ramblings about TiddlyWiki and related stuff.
>
> It basically is a "jump off" page to various TiddlyWiki sites, that show editions, themes and plugins.

"Wills Q&D gTD" by Matabele

6th January 2021 at 4:10pm

Other Resources

Matabele's stylish and clean system for personal task management.

http://wills.tiddlyspot.com

> This is my work in progress version of TW5, incorporating various features and ideas as and when they intrigue me. To grab any feature you may wish to try, drag the listed tiddlers across to the dropzone of an empty TiddlyWiki 5.0.10-beta

"X3DOM for TiddlyWiki 5" by Jamal Wills

6th January 2021 at 4:10pm

Other Resources

An experiment to embed interactive 3D graphics within TiddlyWiki by Jamal Wills. It is based X3DOM, an open source JavaScript library.

http://x3dom.tiddlyspot.com/

"デザイン刷新。次の25年を目指すTiddlyWiki" by Moongift

11th November 2013 at 9:45am

Tutorials

An introduction to TiddlyWiki in Japanese.

http://www.moongift.jp/2013/11/tiddlywiki5-%E3%83%87%E3%82%B6%E3%82%A4%E3%83%B3%E5%88%B7%E6%96%B0%E3%80%82%E6%AC%A1%E3%81%AE25%E5%B9%B4%E3%82%92%E7%9B%AE%E6%8C%87%E3%81%99tiddlywiki-2/

A Gentle Guide to TiddlyWiki

19th May 2021 at 1:06pm

HelloThere

TiddlyWiki is an incredibly flexible and versatile tool that is conceived and constructed differently than most software. This can make it hard to understand until the moment when it clicks, and becomes a seamless extension of your brain.

Starting with the basics, TiddlyWiki is a note-taking web application you can download for free, store wherever you like and customise however you wish. Use it to capture, organise and share your notes in ways that word processors and other note-taking tools cannot.

TiddlyWiki is designed to be non-linear, structuring content with stories, tags, hyperlinks, and other features. You can organise and retrieve your notes in ways that conform to your personal thought patterns, rather than feel chained to one preset organisational structure.

You can use TiddlyWiki as a single file that you view and edit through any web browser, whether you are online or offline. Or you can use it as a powerful Node.js application that stores each of your notes as a separate file.

How can you make TiddlyWiki work for you? We recommend you start with our introductory documentation listed below, and then browse the TableOfContents, available in the Contents tab in the sidebar. Or just follow our simple instructions and try it out for yourself!

- A Gentle Guide to TiddlyWiki
- Discover TiddlyWiki
- Some of the things you can do with TiddlyWiki
- Ten reasons to switch to TiddlyWiki
- Examples
- What happened to the original TiddlyWiki?
- Funding TiddlyWiki
- Open Collective

Also see:

- "Grok TiddlyWiki" by Soren Bjornstad
- TiddlyWiki Video Tutorials by Francis Meetze

About

16th September 2014 at 3:20pm

TableOfContents

Information about the making of TiddlyWiki:

- Acknowledgements
- Contributors
- Funding TiddlyWiki
- History of TiddlyWiki
- License
- Open Collective
- Releases
- RoadMap
- TiddlyWiki Archive

abs Operator

11th June 2019 at 7:42pm

Filter Operators Mathematics Operators Unary Mathematics Operators

purpose	calculate the absolute value of a list of numbers
input	a selection of titles
output	the absolute value of the input numbers

Learn more about how to use Filters

ⓘ **Introduced in v5.1.20** See Mathematics Operators for an overview.

Examples

abs Operator (Examples)

11th June 2019 at 7:42pm

abs Operator Operator Examples

```
[[-2000]abs[]]
```

Try it

```
=-1 =-2 =3 =4 =5 +[abs[]]
```

Try it

Ace Editor Plugin by Joerg Plewe

6th January 2021 at 4:10pm

Community Plugins

A plugin to integrate the Ace editor into TiddlyWiki.

http://innoq.tiddlyspot.com

> Being quite new to TW5 development I tried my best to draft a *SyncAdaptor* to store tiddlers in *IndexedDb*. This is quite useful if an application / plugin installation is not possible (we have quite some restrictions at work...).

Acknowledgements

12th September 2014 at 4:51pm

About

TiddlyWiki wouldn't be possible without the ongoing support of the TiddlyWiki Community. Their attention and feedback has made it possible to gain an understanding of what is wanted from the product, and their passion for it has taught me that it is worth the investment.

TiddlyWiki incorporates code from these fine OpenSource projects:

- The Stanford Javascript Crypto Library
- The Jasmine JavaScript Test Framework
- Normalize.css by Nicolas Gallagher

And media from these projects:

- World flag icons from Wikipedia

acos Operator

22nd October 2021 at 12:28am

Filter Operators Mathematics Operators Trigonometric Operators Unary Mathematics Operators

purpose	calculate the arccosine value (in radians) of a list of numbers
input	a selection of titles
output	the arccosine (in radians) of the input numbers

Learn more about how to use Filters

Introduced in v5.1.21 See Mathematics Operators for an overview.

This is the inverse operation of cos: cos calculate the cosine of an angle (in radian), but acos calculate the angle (in radian) of a cosine.

Examples

acos Operator (Examples)

20th October 2021 at 4:27pm

acos Operator Operator Examples

```
[[2]acos[]]
```

Try it

```
=1 =2 =3 =4 +[acos[]]
```

Try it

ActionConfirmWidget

15th November 2020 at 5:03pm

ActionWidgets Widgets

Introduction

ⓘ **Introduced in v5.1.23** The **action-confirm** widget is an action widget that prompts the user for confirmation and invokes other action widgets contained within it only if the user confirms. ActionWidgets are used within triggering widgets such as the ButtonWidget.

Content and Attributes

The **action-confirm** widget is invisible. Any content within it is only processed if the user confirms the action, or the confirmation has been disabled by the `$prompt` attribute.

Attribute	Description
$message	Optional message displayed to the user when asking for confirmation.
$prompt	Optional flag, set to "no" to disable the prompt for confirmation. Defaults to "yes"

Examples

Here is an example of a button that asks the user for confirmation, before deleting the caption and tags fields of the current tiddler:

copy to clipboard

```
<$button>
<$action-confirm $message="Do you wish to delete the caption and tags?">
<$action-deletefield caption tags/>
Delete "caption" and "tags"
</$action-confirm>
</$button>
```

That renders as:

Delete "caption" and "tags"

Here is an example of a button that uses the optional `$prompt` attribute to control whether to prompt the user before deleting the text field of the tiddler HelloThere:

copy to clipboard

```
<$button>
<$action-confirm $message="Do you wish to delete the text field?" $prompt={{$:/state/promptUser}}>
<$action-deletefield $tiddler="HelloThere" $field="text"/>
</$action-confirm>
Delete text from ~HelloThere
</$button>
```

That renders as:

Delete text from HelloThere

ActionCreateTiddlerWidget

15th January 2023 at 9:47am

ActionWidgets Widgets

Introduction

The **`$action-createtiddler`** widget is an action widget that creates new tiddlers. Action widgets are used within triggering widgets such as the ButtonWidget.

There are several *differences* from the tm-new-tiddler message:

- The new tiddler is not automatically displayed in the story river
- The title of the new tiddler is made available for subsequent operations

Content and Attributes

The **action-createtiddler** widget is invisible.

Attribute	Description

$basetitle	The initial title that will be attempted. If a tiddler with that title already exists, then a numerical counter is added to the title and incremented until it is unique
$savetitle	**DEPRECATED SINCE: 5.1.20** (see ActionCreateTiddlerWidget Example 5) A text reference identifying a field or index into which the title of the newly created tiddler will be stored after it is created
$savedrafttitle	**DEPRECATED SINCE: 5.1.20** (see ActionCreateTiddlerWidget Example 5) A text reference identifying a field or index into which the draft title associated with the newly created tiddler will be stored after it is created. This is useful when using a sequence of action widgets to create a new tiddler, put it into edit mode, and position it within the list of its parent tag
$timestamp	Specifies whether the timestamp(s) of the target tiddler will be updated (**modified** and **modifier**, plus **created** and **creator** for newly created tiddlers). Can be "yes" (the default) or "no"
$template	ⓘ **Introduced in v5.1.22** The title of a template tiddler, that will be used to create a new tiddler
$overwrite	ⓘ **Introduced in v5.1.22** If set to "yes", it will overwrite existing tiddlers. **Be careful!**
{any attributes not starting with $}	Each attribute name specifies a field to be created in the new tiddler

Variables

ⓘ **Introduced in v5.2.0** The content of the **`$action-createtiddler`** widget is executed after the new tiddler has been created. The title of the newly created tiddler is stored in the variable **`createTiddler-title`**.

Variables	Description
`createTiddler-title`	The tittle of the tiddler that has been created. See ActionCreateTiddlerWidget Example 5
`createTiddler-`	This variable only exists to have feature parity with the deprecated parameters. It contains the title of a

draftTitle	"draft tiddler"

Examples

copy to clipboard

```
Caption for new button: <$edit-text tiddler="$:/state/new-button-caption" tag="input"
default=""/>

<$button>
<$action-createtiddler $basetitle="Homemade Button" tags="$:/tags/PageControls" text={{$:/state/
new-button-caption}}>
<$action-navigate $to=<<createTiddler-title>>/>
</$action-createtiddler>
Create a new non-functional page control button and open the tiddler
</$button>
```

That renders as:

Caption for new button:

Create a new non-functional page control button and open the tiddler

This example will create "New Tiddler", "New Tiddler 1", " New Tiddler 2" and so on

```
\define testCreate()
<$action-createtiddler />
\end

<$button actions=<<testCreate>> >
Create Tiddler
</$button>
```

Create Tiddler - Clicking this button, will also open the Right sidebar: Recent tab

---

This example will use the base title defined in **$:/language/DefaultNewTiddlerTitle**.

It will overwrite the tiddler, if the button is clicked several times.

```
\define testCreate()
<$action-createtiddler $basetitle={{$:/language/DefaultNewTiddlerTitle}} $overwrite="yes"/>
\end

<$button actions=<<testCreate>> >
Create Tiddler
</$button>
```

Create Tiddler - Clicking this button, will also open the Right sidebar: Recent tab

---

This example will uses a base-title and a template: ActionCreateTiddlerWidget Template

It will create: "base", "base 1", "base 2" and so on

```
\define testCreate()
<$action-createtiddler $basetitle="base" $template="ActionCreateTiddlerWidget Template"/>
\end

<$button actions=<<testCreate>> >
Create Tiddler
</$button>
```

Create Tiddler - Clicking this button will also open the Right sidebar: Recent tab

---

This example uses a base-title "base" and a template: ActionCreateTiddlerWidget Template.

There will be new fields "aa" and "bb" which are added to the new tiddlers.

```
\define testCreate()
<$action-createtiddler $basetitle="base" $template="ActionCreateTiddlerWidget Template" aa="new
field aa" bb="new field bb" />
\end

<$button actions=<<testCreate>> >
Create Tiddler
</$button>
```

Create Tiddler - Clicking this button will also open the Right sidebar: Recent tab

---

This example will use the base title defined in ActionCreateTiddlerWidget Template.

It will create: "base", "base 1", "base 2" and so on, and navigate to this tiddler in draft mode.

```
\define testCreate()
<$action-createtiddler $basetitle="base" $template="ActionCreateTiddlerWidget Template">
    <$action-sendmessage $message="tm-edit-tiddler" $param=<<createTiddler-title>>/>
</$action-createtiddler>
\end

<$button actions=<<testCreate>> >
Create Tiddler
</$button>
```

> Create Tiddler - Clicking this button, will also open the Right sidebar: Recent tab

ActionCreateTiddlerWidget Example

13th November 2021 at 2:10am

ActionCreateTiddlerWidget

Caption for new button:

Create a new non-functional page control button and open the tiddler

ActionCreateTiddlerWidget Example 1

14th November 2021 at 12:04am

ActionCreateTiddlerWidget

This example will create "New Tiddler", "New Tiddler 1", " New Tiddler 2" and so on

```
\define testCreate()
<$action-createtiddler />
\end

<$button actions=<<testCreate>> >
Create Tiddler
</$button>
```

Create Tiddler - Clicking this button, will also open the Right sidebar: Recent tab

ActionCreateTiddlerWidget Example 2

13th November 2021 at 2:10am

ActionCreateTiddlerWidget

This example will use the base title defined in **$:/language/DefaultNewTiddlerTitle**.

It will overwrite the tiddler, if the button is clicked several times.

```
\define testCreate()
<$action-createtiddler $basetitle={{$:/language/DefaultNewTiddlerTitle}} $overwrite="yes"/>
\end

<$button actions=<<testCreate>> >
Create Tiddler
</$button>
```

Create Tiddler - Clicking this button, will also open the Right sidebar: Recent tab

ActionCreateTiddlerWidget Example 3

13th November 2021 at 2:11am

ActionCreateTiddlerWidget

This example will uses a base-title and a template: ActionCreateTiddlerWidget Template

It will create: "base", "base 1", "base 2" and so on

```
\define testCreate()
<$action-createtiddler $basetitle="base" $template="ActionCreateTiddlerWidget Template"/>
\end

<$button actions=<<testCreate>> >
Create Tiddler
</$button>
```

Create Tiddler - Clicking this button will also open the Right sidebar: Recent tab

ActionCreateTiddlerWidget Example 4

13th November 2021 at 2:11am

ActionCreateTiddlerWidget

This example uses a base-title "base" and a template: ActionCreateTiddlerWidget Template.

There will be new fields "aa" and "bb" which are added to the new tiddlers.

```
\define testCreate()
<$action-createtiddler $basetitle="base" $template="ActionCreateTiddlerWidget Template" aa="new field aa"
bb="new field bb" />
\end
```

```
<$button actions=<<testCreate>> >
Create Tiddler
</$button>
```

Create Tiddler - Clicking this button will also open the Right sidebar: Recent tab

ActionCreateTiddlerWidget Example 5

11th January 2023 at 11:09pm

ActionCreateTiddlerWidget

This example will use the base title defined in ActionCreateTiddlerWidget Template.

It will create: "base", "base 1", "base 2" and so on, and navigate to this tiddler in draft mode.

```
\define testCreate()
<$action-createtiddler $basetitle="base" $template="ActionCreateTiddlerWidget Template">
    <$action-sendmessage $message="tm-edit-tiddler" $param=<<createTiddler-title>>/>
</$action-createtiddler>
\end

<$button actions=<<testCreate>> >
Create Tiddler
</$button>
```

Create Tiddler - Clicking this button, will also open the Right sidebar: Recent tab

ActionCreateTiddlerWidget Template

16th November 2021 at 8:25pm

test Widget Examples

This is a tiddler used as a template in: ActionCreateTiddlerWidget Example 3

ActionDeleteFieldWidget

20th February 2015 at 5:20pm

ActionWidgets Widgets

Introduction

The **action-deletefield** widget is an action widget that deletes specified fields of a tiddler. ActionWidgets are used within triggering widgets such as the ButtonWidget.

Content and Attributes

The **action-deletefield** widget is invisible. Any content within it is ignored.

Attribute	Description

$tiddler	The title of the tiddler whose fields are to be modified (if not provided defaults to the current tiddler)
$field	Optional name of a field to delete
{any attributes not starting with $}	Each attribute name specifies a field to be deleted. The attribute value is ignored and need not be specified

Examples

Here is an example of a button that deletes the caption and tags fields of the current tiddler:

copy to clipboard

```
<$button>
<$action-deletefield caption tags/>
Delete "caption" and "tags"
</$button>
```

That renders as:

Delete "caption" and "tags"

Here is an example of a button that deletes the modified date and tags fields of the tiddler HelloThere:

copy to clipboard

```
<$button>
```

```
<$action-deletefield $tiddler="HelloThere" modified tags/>
Delete "modified" and "tags" from ~HelloThere
</$button>
```

That renders as:

Delete "modified" and "tags" from HelloThere

Here is an example of a button that uses the optional $field attribute to delete the text field of the tiddler HelloThere:

copy to clipboard

```
<$button>
<$action-deletefield $tiddler="HelloThere" $field="text"/>
Delete text from ~HelloThere
</$button>
```

That renders as:

Delete text from HelloThere

ActionDeleteTiddlerWidget

6th November 2014 at 6:34pm

ActionWidgets Widgets

Introduction

The **action-deletetiddler** widget is an action widget that deletes tiddlers. ActionWidgets are used within triggering widgets such as the ButtonWidget.

There are several differences compared to the WidgetMessage: tm-delete-tiddler:

- The user is not prompted to confirm the deletion
- No automatic updating of the story list
- No special handling of draft tiddlers

Content and Attributes

The **action-deletetiddler** widget is invisible. Any content within it is ignored.

Attribute	Description
$tiddler	Optional title of the tiddler to be deleted
$filter	Optional filter identifying tiddlers to be deleted

Examples

Here is an example of a button that deletes the tiddler HelloThere:

copy to clipboard

```
<$button>
<$action-deletetiddler $tiddler="HelloThere"/>
Delete "~HelloThere"
</$button>
```

That renders as:

Delete "HelloThere"

Here is an example of a button that deletes all tiddlers tagged TableOfContents:

copy to clipboard

```
<$button>
<$action-deletetiddler $filter="[tag[TableOfContents]]"/>
Delete tiddlers tagged "~TableOfContents"
</$button>
```

That renders as:

Delete tiddlers tagged "TableOfContents"

ActionListopsWidget

5th August 2023 at 12:35pm

ActionWidgets Widgets

Introduction

The **action-listops** widget is an action widget that manipulates user lists in any field or data index. ActionWidgets are used within triggering widgets such as the ButtonWidget.

Content and Attributes

The **action-listops** widget is invisible. Any content within it is ignored.

Attribute	Description
$tiddler	The title of the tiddler whose lists are to be modified (if not provided defaults to the current tiddler)
$field	The name of a field to be manipulated as a list (defaults to 'list')
$index	Optional index of a property in a data tiddler index to be manipulated as a list
$filter	An optional filter expression, the output of which will be saved to the field/index being manipulated
$subfilter	An optional subfilter expression, which takes the list being manipulated as input, and saves the modified list back to the field/index being manipulated
$tags	An optional subfilter expression, which takes the **`tags`** field of the target tiddler as input, and saves the modified list of tags back to the **`tags`** field

Note on subfilter expressions

If the manipulation depends on the current contents of the list, e.g. when using the **`toggle`** operator to toggle the presence of an element, the Filter Run would be prefixed with the `+` / `:and` filter run prefix so that it properly receives the list as input.

```
<$action-listops $subfilter="+[toggle[List Item]]"/>
```

The above widget will toggle the presence of the element **`List Item`** in the field **`list`** of the current tiddler, removing or adding the element as necessary.

Similarly, if an element is to always be removed when it is present, the `-` / `:except` filter run prefix can be used. Both of the following yield the same result:

```
<$action-listops $subfilter="-[[List Item]]"/>
<$action-listops $subfilter="+[remove[List Item]]"/>
```

Without any prefixes, the filter run output is simply dominantly appended to the list.

See also the Examples.

Using $filter or $subfilter

Standalone use of the `$subfilter` attribute can be replaced by using a (more complicated) `$filter` attribute value.

For example, the items "abc" and "123" can be appended to the field `myfield` using the `$subfilter` attribute:

```
<$action-listops $field="myfield" $subfilter="abc 123"/>
```

The same can be achieved using the `$filter` attribute and prepending the Filter Run `[enlist{!!myfield}]` to the Filter Expression:

```
<$action-listops $field="myfield" $filter="[enlist{!!myfield}] abc 123"/>
```

The short form is more convenient, but the long form is useful for live-debugging complicated `$subfilter` values using the filter tab of **$:/AdvancedSearch**. By using **$:/AdvancedSearch**, the Filter Expression can be tested before using **action-listops** to modify actual tiddler fields. For this use case, the `all[current]` portion of the expression needs to be changed to select the proper test tiddler.

Using $tags or $subfilter

Tagging is implemented using a tiddler's 'tags' field, so appending the tags "abc" and "123" using the `$tags` attribute like this:

```
<$action-listops $tags="abc 123"/>
```

is mostly equivalent to using `$subfilter` along with "tags" for the value of `$field`:

```
<$action-listops $field="tags" $subfilter="abc 123"/>
```

Comparison to ActionSetFieldWidget

In general, ActionSetFieldWidget is better for setting multiple fields at once and for replacing the value of a field, which can also be a list. The ActionListopsWidget is better for modifying a list field based on the existing list and for using a Filter Expression to derive the value of the field.

The ActionSetFieldWidget sets the value of a field using `$field` and `$value` attribute pairs or attributes that do not start with a `$`. A single ActionSetFieldWidget can be used to set any number of fields of a single tiddler.

The ActionListopsWidget replaces or modifies a single field's value using filter expressions.

The following widgets are functionally equivalent:

```
<$action-setfield $field="myfield" $value="abc 123"/>
<$action-setfield myfield="abc 123"/>
<$action-listops $field="myfield" $filter="abc 123"/>
```

Note that **`abc 123`** in the first two cases is a literal string that is assigned to the field **`myfield`**, but in the third case a filter expression which evaluates to the same string.

Extended Filter Operators

A number of Extended Listops Filters are necessary for the manipulation of lists. These operators have been designed primarily for use in subfilter expressions whereby the modified current list is returned in place of the current list.

Notes on de-duplication

In some cases, there may occur unexpected de-duplication of lists.

Assignments to the **`list`** field

When assigning filter results to the **`list`** field (default), the generated list is automatically de-duplicated, so

```
<$action-listops $filter="[[1]] :and[[1]]"/>
```

will result in the **list** field of the current tiddler containing the string **1**, but not **1 1**.

Input to the subfilter expression

The input to the subfilter expression in the `$subfilter` attribute is also de-duplicated. If you rely on lists containing duplicates, consider using this alternative using the `$filter` attribute:

copy to clipboard

```
<$button>
<$action-listops $field="myfield" $filter="[enlist:raw{!!myfield}] :all[[abc]]" />
Add 'abc' to 'myfield'
</$button>

<$list filter="[enlist:raw{!!myfield}]" template="$:/core/ui/ListItemTemplate" />
```

That renders as:

Add 'abc' to 'myfield'

The enlist Operator with `raw` suffix will enlist the list saved in **myfield** of the current tiddler without de-duplication, while e.g. the list Operator will always de-duplicate. The widget then adds the item **abc** – whether or not it is already included in the list – and replaces the original list in **myfield**.

Examples

ActionListopsWidget (Examples)

5th August 2023 at 12:36pm

ActionListopsWidget

In this example we shall populate and then clear a list in an ordinary field (myfield) of this tiddler (the default).

copy to clipboard

```
<$button>
<$action-listops $field="myfield" $filter="efg hlm pqr"/>
Populate 'myfield'
</$button>
<$button>
<$action-listops $field="myfield" $subfilter="abc xyz"/>
Append More Items
</$button>
<$button>
<$action-listops $field="myfield" $subfilter="-abc -hlm"/>
Remove Items
</$button>
<$button>
<$action-listops $field="myfield" $filter="[[]]"/>
Clear 'myfield'
</$button>
```

```
<$list filter="[list[!!myfield]]">

</$list>
```

That renders as:

Populate 'myfield' | Append More Items | Remove Items | Clear 'myfield'

In this example we shall append and remove items from a list in an ordinary field (myfield) of this tiddler (the default) and sort the resultant list. We shall then remove some of the appended items and sort the resulting list in reverse order.

copy to clipboard

```
<$button>
<$action-listops $field="myfield" $subfilter="-efg ijk xyz [[this is a title]] +[sort[]]"/>
Mangle List
</$button>
<$button>
<$action-listops $field="myfield" $subfilter="-xyz -[[this is a title]] +[!sort[]]"/>
Unmangle List
</$button>

<$list filter="[list[!!myfield]]">
```

```
</$list>
```

That renders as:

Mangle List Unmangle List

The following example toggles the tag **Examples** for the current tiddler.

copy to clipboard

```
<$button>
<$action-listops $tags="+[toggle[Examples]]"/>
Toggle 'Examples' tag
</$button>
```

That renders as:

Toggle 'Examples' tag

In this example we shall append a few tags to the 'tags' field of this tiddler (the default). We shall then remove some of the appended tags.

copy to clipboard

```
<$button>
```

```
<$action-listops $tags="+[append{Days of the Week!!short}] $:/tag1 $:/tag2 $:/tag3"/>
Populate 'tags'
</$button>
<$button>
<$action-listops $tags="+[!remove:2{!!tags}]"/>
Remove Last Two Tags
</$button>
<$button>
<$action-listops $tags="+[!prefix[$:/]]"/>
Remove System Tags
</$button>
<$button>
<$action-listops $tags="-Mon -Tue"/>
Remove Mon and Tue
</$button>
<$button>
<$action-listops $tags="+[prefix[$:/]] ActionWidgets Widgets"/>
Remove User Tags
</$button>
<$button>
<$action-listops $tags="+[[]] ActionWidgets Widgets"/>
Clear Tags
</$button>

<$list filter="[list[!!tags]]">
```

```
</$list>
```

That renders as:

Populate 'tags' | Remove Last Two Tags | Remove System Tags | Remove Mon and Tue | Remove User Tags | Clear Tags

ActionListopsWidget

ActionLogWidget

20th November 2020 at 4:52pm

ActionWidgets Debugging Widgets Widgets

Introduction

Introduced in v5.1.23 The **action-log** widget is an action widget that can be used to output debugging information to the JavaScript console supported by most browsers. This can be useful because otherwise it is difficult to observe what is going on within a sequence of actions.

ActionWidgets are used within triggering widgets such as the ButtonWidget.

For debugging outside of actions see LogWidget

Content and Attributes

The **action-log** widget is invisible. Any content within it is ignored.

When the actions are invoked, the names and values of all attributes are logged to the JavaScript console. In addition there are optional attributes that can be used:

Optional Attribute	Description
$$filter	All variables matching this filter will also be logged.
$$message	A message to display as the title of the information logged. Useful when several `action-log` widgets are used in sequence.
$$all	Set to "yes" to log all variables in a collapsed table. Note that if there is nothing specified to log, all variables are always logged instead.



A handy tip if an action widget is not behaving as expected is to temporarily change it to an `<$action-log>` widget so that the attributes can be observed.

Example

Here is an example of logging two variables:

```
<$action-log myVar=<<myVar>> otherVar=<<otherVar>>/>
```

To log all variables:

```
<$action-log />
```

To log two variables as well as all core variables (which start with `tv-`):

```
<$action-log myVar=<<myVar>> other={{!!status}} $$filter="[prefix[tv-]]"/>
```

ActionNavigateWidget

20th February 2015 at 5:20pm

ActionWidgets Widgets

Introduction

The **action-navigate** widget is an action widget that sends a tm-navigate message back up the widget tree. ActionWidgets are used within triggering widgets such as the ButtonWidget.

Content and Attributes

The **action-navigate** widget is invisible. Any content within it is ignored.

Attribute	Description
$to	The title of the target tiddler for the navigation (if not provided defaults to the current tiddler
$scroll	Optional parameter determining whether the navigation will also cause a scroll to the target tiddler (see below)

Scroll handling

The optional `$scroll` attribute can be set to "yes" to force scrolling to occur to bring the target tiddler into view. If set to "no" then scrolling does not occur. If the `$scroll` attribute is omitted then scrolling occurs unless either:

- the control key is pressed
- the action was initiated with the middle mouse button (if available)

Note that if navigating to multiple tiddlers at once you should use the same `$scroll` setting for all of them.

Examples

Here is an example of button that navigates to two different tiddlers at once:

copy to clipboard

```
<$button>
<$action-navigate $to="ButtonWidget"/>
<$action-navigate $to="ActionWidgets"/>
Click me!
</$button>
```

That renders as:

Click me!

ActionPopupWidget

31st July 2023 at 9:30pm

ActionWidgets Widgets

Introduction

The **action-popup** widget is an action widget that triggers the display of a popup defined via a state tiddler, or clears all displayed popups. ActionWidgets are used within triggering widgets such as the ButtonWidget.

Content and Attributes

The **action-popup** widget is invisible. Any content within it is ignored.

Attribute	Description
$state	The title of the state tiddler for the popup
$coords	Optional coordinates for the handle to which popup is positioned (see Coordinate Systems for the supported formats)
$floating	ⓘ **Introduced in v5.2.0** Optional. Defaults to `no`. Set to `yes` to create a popup that must be closed explicitly.

ⓘ **Introduced in v5.1.23** If the **$coords** attribute is missing or empty then all popups are cancelled.
ⓘ **Introduced in v5.2.4** The **$coords** attribute supports absolute and relative coordinates. See Coordinate Systems for more information.



Delete the state tiddler for a floating popup to close it.

Examples

Here is an example of button that triggers the "more" button in the sidebar "Tools" tab. You may need to scroll to see the popup

copy to clipboard

```
<$button>
<$action-setfield $tiddler="$:/state/tab/sidebar--595412856" $value="$:/core/ui/SideBar/Tools"/>
<$action-popup $state="$:/state/popup/more--810643385" $coords="(0,20,0,0)"/>
Click me!
</$button>
```

That renders as:

Click me!

ActionSendMessageWidget

23rd October 2021 at 6:01pm

ActionWidgets Widgets

Introduction

The **action-sendmessage** widget is an action widget that sends a message back up the widget tree. ActionWidgets are used within triggering widgets such as the ButtonWidget.

Content and Attributes

The **action-sendmessage** widget is invisible. Any content within it is ignored.

Attribute	Description
$message	The message to send (eg, WidgetMessage: tm-new-tiddler)
$param	Optional parameter string whose meaning is dependent on the message being sent
$name	Optional name of additional parameter
$value	Value for optional parameter whose name is specified in `$name`
$names	ⓘ **Introduced in v5.2.1** Optional filter evaluating to a list of additional parameter names
$values	ⓘ **Introduced in v5.2.1** Optional filter evaluating to a list of parameter values corresponding to the parameters names specified in `$names`
{any attributes not starting with $}	Multiple additional, optional named parameters that are attached to the message

Examples

Here is an example of button that displays both a notification and a wizard, and creates a new tiddler with tags and text:

copy to clipboard

```
<$button>
<$action-sendmessage $message="tm-modal" $param="SampleWizard"/>
<$action-sendmessage $message="tm-notify" $param="SampleNotification"/>
<$action-sendmessage $message="tm-new-tiddler" title="This is newly created tiddler" tags="OneTag
[[Another Tag]]" text=<<now "Today is DDth, MMM YYYY">>/>
Click me!
</$button>
```

That renders as:

Click me!

ActionSetFieldWidget

6th August 2015 at 7:14pm

ActionWidgets Widgets

Introduction

The **action-setfield** widget is an action widget that assigns values to the fields of a tiddler. ActionWidgets are used within triggering widgets such as the ButtonWidget.

Content and Attributes

The **action-setfield** widget is invisible. Any content within it is ignored.

Attribute	Description
$tiddler	The title of the tiddler whose fields are to be modified (if not provided defaults to the current tiddler)
$field	Optional name of a field to be assigned the $value attribute
$index	Optional index of a property in a data tiddler to be assigned the $value attribute
$value	The value to be assigned to the field or index identified by the $field or $index attribute. If neither is specified then the value is assigned to the text field. If no value is specified, $field or $index will be deleted.
$timestamp	Specifies whether the timestamp(s) of the target tiddler will be updated (**modified** and **modifier**, plus **created** and **creator** for newly created tiddlers). Can be "yes" (the default) or "no"
{any attributes not starting with $}	Each attribute name specifies a field to be modified with the attribute value providing the value to assign to the field

Examples

Here is an example of a pair of buttons that open the control panel directly to specified tabs. They work by using **action-setfield** to set the state tiddler for the control panel tabs.

copy to clipboard

```
<$button>
<$action-setfield $tiddler="$:/state/tab-1749438307" text="$:/core/ui/ControlPanel/Appearance"/>
```

```
<$action-navigate $to="$:/ControlPanel"/>
Go to Control Panel "Appearance" tab
</$button>

<$button>
<$action-setfield $tiddler="$:/state/tab-1749438307" text="$:/core/ui/ControlPanel/Settings"/>
<$action-navigate $to="$:/ControlPanel"/>
Go to Control Panel "Settings" tab
</$button>
```

That renders as:

Go to Control Panel "Appearance" tab

Go to Control Panel "Settings" tab

Here is an example of a button that assigns tags and fields to the tiddler HelloThere, and then navigates to it and opens the tiddler info panel on the "Fields" tab:

copy to clipboard

```
<$button>
<$action-setfield $tiddler="HelloThere" tags="NewTag [[Another New Tag]]" color="red"/>
<$action-setfield $tiddler="$:/state/popup/tiddler-info--1779055697" text="(568,1443,33,39)"/>
<$action-setfield $tiddler="$:/state/tab--1890574033" text="$:/core/ui/TiddlerInfo/Fields"/>
```

```
<$action-navigate $to="HelloThere"/>
Modify ~HelloThere
</$button>
```

That renders as:

Modify HelloThere

Here is an example of a button that assigns tags and fields to the tiddler HelloThere, and then initiates editing it:

copy to clipboard

```
<$button>
<$action-setfield $tiddler="HelloThere" tags="MoreTag [[Further More Tags]]" color="green"/>
<$action-sendmessage $message="tm-edit-tiddler" $param="HelloThere"/>
Edit ~HelloThere
</$button>
```

That renders as:

Edit HelloThere

Here is an example of a button that opens the control panel directly to the "Appearance" tabs:

copy to clipboard

```
<$button>
<$action-setfield $tiddler="$:/state/tab-1749438307" $field="text" $value="$:/core/ui/ControlPanel/
Appearance"/>
<$action-navigate $to="$:/ControlPanel"/>
Go to Control Panel "Appearance" tab
</$button>
```

That renders as:

Go to Control Panel "Appearance" tab

ActionSetMultipleFieldsWidget

23rd October 2021 at 6:01pm

ActionWidgets Widgets

Introduction

The **action-setmultiplefields** widget is an action widget that assigns multiple fields to a tiddler where the names and values of the fields are specified as separate filters. ActionWidgets are used within triggering widgets such as the ButtonWidget.

Content and Attributes

The **action-setmultiplefields** widget is invisible. Any content within it is ignored.

Attribute	Description
$tiddler	Optional title of the tiddler to modify (defaults to the current tiddler)
$fields	Optional filter evaluating to the names of a list of fields to assign. Either `$fields` or `$indexes` must be specified
$indexes	Optional filter evaluating to the names of a list of indexes to assign. Either `$fields` or `$indexes` must be specified
$values	Filter evaluating to the values to be assigned to the fields or indexes
$timestamp	Specifies whether the timestamp(s) of the target tiddler will be updated (**modified** and **modifier**, plus **created** and **creator** for newly created tiddlers). Can be "yes" (the default) or "no"

Examples

Here is an example of button that uses the data in Days of the Week to assign fields `day-Monday`, `day-Tuesday` etc. to the tiddler "HelloThere":

copy to clipboard

```
<$button>
<$action-setmultiplefields $tiddler="HelloThere" $fields="[list[Days of the Week]addprefix[day-]]"
$values="[list[Days of the Week]]"/>
Click me!
</$button>
```

That renders as:

Click me!

actionTiddler Variable

18th January 2019 at 9:46am

Core Variables Variables

The variable `actionTiddler` is used in subtly different ways by different widgets:

- Within the **actions** string of the DroppableWidget, the **actionTiddler** variable contains the title of the tiddler being dropped.
- Within the **startactions** and **endactions** string of the DroppableWidget, the **actionTiddler** variable contains a quoted Title List of all of the titles being dragged.

ActionWidget Execution Modes

16th November 2021 at 8:32pm

Widgets

Introduced in v5.2.0 The default behaviour of action widgets has some peculiarities that often cause confusion. There is now an improved mode that simplifies how things work, but due to *BackwardsCompatibility* constraints, it must be explicitly engaged in order to take advantage of it.

The peculiarities relate to the way that the results of previous action widgets are available to subsequent action widgets. By default, action

widgets are refreshed before each execution which ensure that they reflect the results of previous actions. However, ordinary widgets are not updated in the same way.

In the following contrived example, a button triggers a series of actions that should result in the string `foo` being assigned to the **text** field of the tiddler *ActionTestTiddler*. However, it fails to produce the expected result because the `<$set>` widget is not refreshed with the new value of *ActionTestTiddler* after the execution of the first `<$action-setfield>` widget.

copy to clipboard

```
\define actions()
<$action-setfield $tiddler="ActionTestTiddler" $field="text" $value="FOO"/>
<$set name="newvalue" value={{{ [{ActionTestTiddler}lowercase[]] }}}>
<$action-setfield $tiddler="ActionTestTiddler" $field="text" $value=<<newvalue>>/>
</$set>
\end

Current value of ActionTestTiddler: {{ActionTestTiddler}}

<$button actions=<<actions>>>
Click me
</$button>
```

That renders as:

Current value of *ActionTestTiddler*:

Click me

The new behaviour avoids these problems by refreshing all widgets before execution, not just action widgets. It is engaged by running the actions in a scope that includes the variable `tv-action-refresh-policy` set to the value `always`. (The default value for `tv-action-refresh-policy` is `once`).

The assignment can be done within an action string, or via a local variable declaration containing the widget triggering the action.



Do not attempt to set `tv-action-refresh-policy` globally; the core will only work correctly with the default setting

The example above works as expected with the addition of `tv-action-refresh-policy`:

copy to clipboard

```
\define tv-action-refresh-policy() always

\define actions()
<$action-setfield $tiddler="ActionTestTiddler" $field="text" $value="FOO"/>
<$set name="newvalue" value={{{ [{ActionTestTiddler}lowercase[]] }}}>
<$action-setfield $tiddler="ActionTestTiddler" $field="text" $value=<<newvalue>>/>
</$set>
\end

Current value of ActionTestTiddler: {{ActionTestTiddler}}

<$button actions=<<actions>>>
Click me
</$button>
```

That renders as:

Current value of *ActionTestTiddler*:

Click me

ActionWidgets

31st October 2021 at 6:29pm

Widgets

Action widgets are a special type of widget that have no visual appearance but perform an action when triggered (such as sending a message, navigating to a tiddler, or changing the value of a tiddler). Action widgets are used in association with other widgets that trigger those actions (for example, the ButtonWidget).

The following action widgets are provided:

- action-confirm
- action-createtiddler
- action-deletefield
- action-deletetiddler
- action-listops
- action-log
- action-navigate
- action-popup
- action-sendmessage
- action-setfield

- action-setmultiplefields

There are two ways to use action widgets:

- Placing them within an action string attribute (typically called `actions`) on the triggering widget (this is the preferred way)
- Embedding the actions within the triggering widget (an older technique that is now deprecated)

Using action string attributes

The action widgets are passed as a string to the `actions` attribute of the triggering widget. Usually, it is more convenient to use a macro to assign the action widgets to a variable. For example, here is a button that triggers two actions of sending different messages:

```
\define my-actions()
<$action-sendmessage $message="tm-home"/>
<$action-sendmessage $message="tm-full-screen"/>
\end

<$button actions=<<my-actions>>>
Click me!
</$button>
```

Directly embedding action widgets

The action widgets need not be immediate children of their triggering widget, but they must be descendents of it. The actions are performed in sequence. Here is the above example rewritten to use embedding:

```
<$button>
<$action-sendmessage $message="tm-home"/>
<$action-sendmessage $message="tm-full-screen"/>
Click me!
</$button>
```

Action Execution Modes

Introduced in v5.2.0 The default behaviour of action widgets has some peculiarities that often cause confusion. There is now an improved mode that simplifies how things work, but due to *BackwardsCompatibility* constraints, it must be explicitly engaged in order to take advantage of it.

The peculiarities relate to the way that the results of previous action widgets are available to subsequent action widgets. By default, action widgets are refreshed before each execution which ensure that they reflect the results of previous actions. However, ordinary widgets are not updated in the same way.

In the following contrived example, a button triggers a series of actions that should result in the string `foo` being assigned to the **text** field of the tiddler *ActionTestTiddler*. However, it fails to produce the expected result because the `<$set>` widget is not refreshed with the new value of *ActionTestTiddler* after the execution of the first `<$action-setfield>` widget.

copy to clipboard

```
\define actions()
<$action-setfield $tiddler="ActionTestTiddler" $field="text" $value="FOO"/>
<$set name="newvalue" value={{{ [{ActionTestTiddler}lowercase[]] }}}>
```

```
<$action-setfield $tiddler="ActionTestTiddler" $field="text" $value=<<newvalue>>/>
</$set>
\end

Current value of ActionTestTiddler: {{ActionTestTiddler}}

<$button actions=<<actions>>>
Click me
</$button>
```

That renders as:

Current value of *ActionTestTiddler*:

Click me

The new behaviour avoids these problems by refreshing all widgets before execution, not just action widgets. It is engaged by running the actions in a scope that includes the variable `tv-action-refresh-policy` set to the value `always`. (The default value for `tv-action-refresh-policy` is `once`).

The assignment can be done within an action string, or via a local variable declaration containing the widget triggering the action.



Do not attempt to set `tv-action-refresh-policy` globally; the core will only work correctly with the default setting

The example above works as expected with the addition of `tv-action-refresh-policy`:

copy to clipboard

```
\define tv-action-refresh-policy() always

\define actions()
<$action-setfield $tiddler="ActionTestTiddler" $field="text" $value="FOO"/>
<$set name="newvalue" value={{{ [{ActionTestTiddler}lowercase[]] }}}>
<$action-setfield $tiddler="ActionTestTiddler" $field="text" $value=<<newvalue>>/>
</$set>
\end

Current value of ActionTestTiddler: {{ActionTestTiddler}}

<$button actions=<<actions>>>
Click me
</$button>
```

That renders as:

Current value of *ActionTestTiddler*:

Click me

add Operator

21st March 2023 at 2:06pm

Binary Mathematics Operators Filter Operators Mathematics Operators

purpose	treating each input title as a number, add to each the numeric value of the parameter
input	a selection of titles
parameter	`N` = a number
output	the input as numbers, but with `N` added to each one

Learn more about how to use Filters

Introduced in v5.1.20 See Mathematics Operators for an overview.

Examples

add Operator (Examples)

11th June 2019 at 2:55pm

add Operator Operator Examples

```
[[23]add[19]]
```

Try it

```
=1 =2 =3 =4 +[add[4]]
```

Try it

Adding a table of contents to the sidebar

6th November 2018 at 11:41pm

Customise TiddlyWiki

The standard mechanisms in TiddlyWiki for creating Table-of-Contents are macros known collectively as "toc" macros (click on link to learn more about macros). They use `Tagging` as their means of creating relationships, so be sure to review the Tagging topic tiddler if tagging is a new concept for you.

A customisable table of contents can be added to the sidebar with the following steps:

1. Create a tiddler called TableOfContents
2. Give it the tag **$:/tags/SideBar**
3. Set the text to

```
<div class="tc-table-of-contents">

<<toc-selective-expandable 'TableOfContents'>>

</div>
```

4. Add a **caption** field with the text **Contents**

5. Add a **list-after** field with the text **$:/core/ui/SideBar/Open**

Add entries to the table of contents by creating tiddlers tagged **TableOfContents**. An easy way is to choose **new here** from the tiddler toolbar of the **TableOfContents** tiddler. (if you don't see the "new here" button, click on the down arrow to see more menu options.)

To create child tiddlers (tiddlers that come below other tiddlers), tag them with the name of the parent tiddler.

Adding a Twitter Follow button

19th September 2014 at 6:15pm

Learning

Here's a macro that provides a Twitter Follow button for a particular username:

```
\define twitterFollowButton(username)
<iframe allowtransparency="true" frameborder="0" scrolling="no" src="//platform.twitter.com/widgets/
follow_button.html?screen_name=$username$" style="width:300px; height:20px;"></iframe>
\end
```

The only change from the version published at https://dev.twitter.com/docs/follow-button is that new lines have been removed.

Note that the src URL is given without a protocol (ie "http" or "https"). It's done this way so that it works on sites whether they are hosted on a HTTP or HTTPS domain. If you want the Twitter button to work while using a TiddlyWiki offline on a "file://" URL, then you'll need to manually add the protocol. For example:

```
<iframe allowtransparency="true" frameborder="0" scrolling="no" src="https://platform.twitter.com/
widgets/follow_button.html?screen_name=jermolene" style="width:300px; height:20px;"></iframe>
```

addprefix Operator

3rd February 2015 at 7:18pm

Filter Operators String Operators

purpose	extend each input title with a prefix
input	a selection of titles
parameter	**S** = a string of characters
output	the input, but with **S** added to the start of each title

Learn more about how to use Filters

Examples

addprefix Operator (Examples)

18th January 2015 at 7:31pm

addprefix Operator Operator Examples

```
Cat Garden [[Favourite Armchair]] +[addprefix[My ]]
```

Try it

addsuffix Operator

3rd February 2015 at 7:32pm

Filter Operators String Operators

purpose	extend each input title with a suffix
input	a selection of titles
parameter	**S** = a string of characters
output	the input, but with **S** added to the end of each title

Learn more about how to use Filters

Examples

addsuffix Operator (Examples)

18th January 2015 at 7:31pm

addsuffix Operator Operator Examples

```
[[London]addsuffix[ Underground]]
```

Try it

Adopt a Titles Policy

12th September 2014 at 5:01pm

Learning

As the structures within your TiddlyWiki documents get more complex it can be hard to keep the titles of tiddlers consistent. For example, should terms be defined in the plural or the singular? Camel case or separate words?

Recording a formal titling policy can help to reduce confusion. For example, the titling policies for this wiki are recorded in the Documentation Style Guide.

SystemTiddler Titles

A useful convention is to use the prefix `$:/_` for any system tiddlers that you create to ensure that they are near the top of the system tiddler listing in the sidebar

after Operator

3rd February 2015 at 7:32pm

Filter Operators Order Operators

purpose	find which input title follows a specified one
input	a selection of titles
parameter	**T** = one of those titles
output	the title that immediately follows **T** in the input

Learn more about how to use Filters

If **T** is not present in the input, or is the last title there, then the output is empty.

Examples

after Operator (Examples)

18th January 2015 at 7:31pm

after Operator Operator Examples

These examples make use of the Days of the Week tiddler. The Thursday tiddler shows a further example.

```
[list[Days of the Week]after[Monday]]
```

Try it

```
[list[Days of the Week]after[Sunday]]
```

Try it

AlertMechanism

6th June 2016 at 2:59pm

Features Mechanisms

Alerts are displayed as yellow boxes overlaying the main TiddlyWiki window. Each one corresponds to a tiddler with the tag *$:/tags/Alert*.

Clicking the 🗑 delete icon on an alert deletes the corresponding tiddler.

Here's a demo alert.

Alert tiddlers should have the following fields:

Field	Description
title	By default, alert titles have the prefix `$:/temp/alerts/`
text	The text of the alert message
modified	Date of the alert (used for ordering the alerts on screen)
component	Component name associated with the alert
tags	Must include *$:/tags/Alert*

Alice in Wonderland

Demonstrations external-text

Trying to load external content from **./text/Alice%2520in%2520Wonderland.tid**

If this message doesn't disappear, either the tiddler content type doesn't match the type of the external content, or you may be using a browser that doesn't support external content for wikis loaded as standalone files. See https://tiddlywiki.com/#ExternalText

All Filter Run Prefix

11th July 2023 at 10:46am

Named Filter Run Prefix

purpose	union of sets without de-duplication
input	all titles from previous filter runs
output	output titles are appended to the output of previous filter runs without de-duplication.

Introduced in v5.1.23
TypeError: setting getter-only property "parentNode"

This prefix has an optional shortcut syntax symbol `=run`

all Operator

21st March 2023 at 2:14pm

Common Operators Filter Operators Selection Constructors Selection Constructors: Conditional

purpose	find all titles of a fundamental category
input	ignored, unless the parameter is empty
parameter	zero or more categories
output	the titles that belong to all the specified categories

Learn more about how to use Filters

The parameter specifies zero or more fundamental categories using the following filter step syntax:

TypeError: setting getter-only property "parentNode"

Category	Members	Sorted

`current`	just the current tiddler	–
`missing`	all non-existent tiddlers to which there is at least one hard link	no
`orphans`	all tiddlers to which there are *NO* hard links	by title
`shadows`	all the shadow tiddlers that exist, including any that have been overridden with non-shadow tiddlers	no
`tags`	all the tags in use on non-shadow tiddlers	no
`tiddlers`	all the non-shadow tiddlers that exist	no

If the parameter specifies more than one category, they are processed from left to right. The overall output is initially empty, and each category's output is dominantly appended to it in turn. Unrecognised categories contribute nothing to the output.

As a special case, if the parameter is empty, the output is simply a copy of the input. This can be useful when the parameter is a soft parameter.

The **`is`** operator is similar, but its scope is restricted to its input.

Examples

all Operator (Examples)

18th January 2015 at 7:31pm

all Operator Operator Examples

```
[all[shadows]]
```

Try it

```
[all[shadows+tiddlers]]
```

Try it

```
[all[tiddlers+shadows]]
```

Try it

```
[all[orphans+missing+current]]
```

Try it

```
[all[current]]
```

Try it

```
[all[current]tag[Operator Examples]]
```

Try it

```
[all[current]tag[Recipes]]
```

Try it

```
Monday Thursday +[all[]]
```

Try it

allafter Operator

8th November 2015 at 6:15am

Filter Operators Listops Operators Order Operators

purpose	discard all items except those after the marker
input	a list of items
suffix	specifying a suffix ('include') will include the marker in the output
parameter	**`marker`** = the list item to be used as a marker

output	all items after the marker

Learn more about how to use Filters

Examples

allafter Operator (Examples)

8th November 2015 at 9:05am

allafter Operator Operator Examples

These examples make use of the Days of the Week tiddler.

```
[list[Days of the Week]] +[allafter[Wednesday]]
```

Try it

```
[list[Days of the Week]] +[allafter:include[Wednesday]]
```

Try it

allbefore Operator

8th November 2015 at 6:15am

Filter Operators Listops Operators Order Operators

purpose	discard all items except those before the marker
input	a list of items
suffix	specifying a suffix ('include') will include the marker in the output
parameter	**`marker`** = the list item to be used as a marker
output	all items before the marker

Learn more about how to use Filters

Examples

allbefore Operator (Examples)

8th November 2015 at 6:14am

allbefore Operator Operator Examples

These examples make use of the Days of the Week tiddler.

```
[list[Days of the Week]allbefore[Wednesday]]
```

Try it

```
[list[Days of the Week]allbefore:include[Wednesday]]
```

Try it

```
A B C D E +[allbefore:include[C]count[]]
```

Try it

AllTiddlers

25th February 2014 at 10:19pm

Current tiddlers:

"A free, open source wiki revisited" by Mark Gibbs, NetworkWorld
"A Thesis Notebook" by Alberto Molina
"ATWiki" by Lamusia Project
"BJTools" by buggyj
"BrainTest - tools for a digital brain" by Danielo Rodriguez
"Cardo - Task and Project Management Wiki" by David Szego
"CouchDB Adaptor" by William Shallum
"Creating a baby journal with TiddlyWiki 5" from A Penguin in Redmond
"Dracula - a dark palette for TiddlyWiki" by SlyBouhafs & zenorocha
"Drift - Collect, Organise, Grow." by Tony K
"Dropboard" by Reid Gould

"Dynamic Tables" by Jed Carty
"Encrypt single tiddler plugin" by Danielo Rodriguez
"file-backups" Extension for Firefox by pmario
"Filter Examples" by Tobias Beer
"Font Awesome 5 Free SVGs for TiddlyWiki" by morosanuae
"Ghostwriter theme by Rory Gibson" adapted for TW5 by Riz
"Gospel Bubbles" by Rev. David Gifford
"Grok TiddlyWiki" by Soren Bjornstad
"GSD5" by Roma Hicks
"Hacks" by Thomas Elmiger
"Heeg.ru" by sini-Kit
"In My Socks" by Jed Carty
"JD Mobile Layout plugin" by JD
"JD's palettes" by JD
"Lucky Sushi" online shop by sini-Kit
"Mal's Sandbox" by Mal
"MathCell" by Stephen Kimmel
"Microblogging con Tiddlywiki" by Juan
"Moments: A little color won't hurt" by Riz
"Mono theme" by JD
"muritest" by Simon Huber
"Notebook theme" by Nicolas Petton
"Noteself" by Danielo Rodríguez
"Notizen mit TiddlyWiki systemübergreifend nutzen" by Michael Sonntag
"PETTIL - Forth for the Commodore PET" by Charlie Hitselberger

"Reveal.js" by Devin Weaver
"savetiddlers" Extension for Chrome and Firefox by buggyj
"Setting Up a Personal TiddlyWiki Server on OS X" by Kris Johnson
"Shining Ark Using TiddlyWiki" by Helu
"SK Plugins" by Stephen Kimmel
"Slides and Stories" by Jan
"Stroll - a Roam-like experience" by David Gifford
"TB5 - a pocket full of tips" by Tobias Beer
"Tekan: Kanban that stays with you" by Riz
"TiddlyServer" by Matt Lauber
"TiddlyStudy" by Kebi
"TiddlyWiki 5 im Betatest" by besim
"TiddlyWiki for Scholars" by Alberto Molina
"TiddlyWiki guide FR" by Sylvain Naudin
"TiddlyWiki Knowledge Network" by Dmitry Sokolov
"TiddlyWiki Notes" by James Anderson
"TiddlyWiki Posts" by Jeffrey Kishner
"TiddlyWiki Toolmap" by David Gifford
"TiddlyWiki 舞" by Bram Chen
"TiddlyWiki" by Sander de Boer
"TiddlyWiki5 Bourbon" by mkt_memory
"TiddlyWiki5 Forum on Reddit" by Riz
"TiddlyWiki5 Playground" by Ton Gerner
"TiddlyWiki5^2 documenting while learning TiddlyWiki5" by Iannis Zannos
"TW Icons" by morosanuae

"TW5 Magick" by Stephan Hradek
"TW5 Tribal Knowledge" from Scott Kingery
"TW5-TeXZilla" plugin by Joe Renes
"TWeb.at" by Mario Pietsch
"Un wiki sur votre BiblioBox avec TiddlyWiki" from A Penguin in Redmond
"Whitespace, a negative-spacious, webapp-nostalgic theme" by JD
"Wikilabs" by PMario
"Wills Q&D gTD" by Matabele
"X3DOM for TiddlyWiki 5" by Jamal Wills
"デザイン刷新。次の25年を目指すTiddlyWiki" by Moongift
A Gentle Guide to TiddlyWiki
About
abs Operator
abs Operator (Examples)
Ace Editor Plugin by Joerg Plewe
Acknowledgements
acos Operator
acos Operator (Examples)
ActionConfirmWidget
ActionCreateTiddlerWidget
ActionCreateTiddlerWidget Example
ActionCreateTiddlerWidget Example 1
ActionCreateTiddlerWidget Example 2
ActionCreateTiddlerWidget Example 3
ActionCreateTiddlerWidget Example 4

ActionCreateTiddlerWidget Example 5
ActionCreateTiddlerWidget Template
ActionDeleteFieldWidget
ActionDeleteTiddlerWidget
ActionListopsWidget
ActionListopsWidget (Examples)
ActionLogWidget
ActionNavigateWidget
ActionPopupWidget
ActionSendMessageWidget
ActionSetFieldWidget
ActionSetMultipleFieldsWidget
actionTiddler Variable
ActionWidget Execution Modes
ActionWidgets
add Operator
add Operator (Examples)
Adding a table of contents to the sidebar
Adding a Twitter Follow button
addprefix Operator
addprefix Operator (Examples)
addsuffix Operator
addsuffix Operator (Examples)
Adopt a Titles Policy
after Operator

after Operator (Examples)
AlertMechanism
Alice in Wonderland
All Filter Run Prefix
all Operator
all Operator (Examples)
allafter Operator
allafter Operator (Examples)
allbefore Operator
allbefore Operator (Examples)
AllTiddlers
AlphaReleases
Alternative page layouts
Amazon Web Services Plugin
Anchor Links using HTML
And Filter Run Prefix
append Operator
append Operator (Examples)
Apple
applypatches Operator
Articles
asin Operator
asin Operator (Examples)
atan Operator
atan Operator (Examples)

atan2 Operator
atan2 Operator (Examples)
Attribute Selectors
Audio
AutoSave
average Operator
average Operator (Examples)
backlinks Operator
backlinks Operator (Examples)
Base64
before Operator
before Operator (Examples)
Behaviour of invoked variables depends on how the variable was declared
Behaviour of variables invoked via filter expression function call
Behaviour of variables invoked via filter operator parameter
Behaviour of variables invoked via normal wikitext
Behaviour of variables invoked via widget attributes
BetaReleases
bf Operator
Block Mode WikiText
Block Mode WikiText (Examples)
Block Quotes in WikiText
Blog Edition
Blurry Lawn.jpg
Brackets

Brownies
BrowserCompatibility
BrowserStorage Plugin
BrowseWidget
BT
BuildCommand
Building TiddlyWikiClassic
butfirst Operator
butlast Operator
butlast Operator (Examples)
ButtonWidget
CamelCase
Caruso - Ave Maria
Cascade Filter Run Prefix
Cascade Filter Run Prefix (Examples)
Cascades
Cascading Style Sheets
ceil Operator
ceil Operator (Examples)
changecount Macro
changecount Macro (Examples)
Changes to filters in 5.0.9-beta
charcode Operator
charcode Operator (Examples)
CheckboxWidget

CheckboxWidget (field Mode)
CheckboxWidget (filter Mode)
CheckboxWidget (indeterminate)
CheckboxWidget (index Mode)
CheckboxWidget (listField Mode)
CheckboxWidget (listIndex Mode)
CheckboxWidget (tag Mode)
Chick Peas
Chinese (Simplified) Edition
Chinese (Traditional) Edition
ClearPasswordCommand
Code Blocks in WikiText
Code styles and auto format settings for IDEs
CodeBlockWidget
CodeMirror Plugin
colour Macro
colour Macro (Examples)
colour-picker Macro
colour-picker Macro (Example 1)
colour-picker Macro (Examples)
ColourPalettes
Comma-Separated Values
Commands
commands Operator
commands Operator (Examples)

CommandsCommand
Community
Community Editions
Community Links Aggregator
Community Palettes
Community Plugins
Community Themes
compare Operator
compare Operator (Examples)
Compose ballad
Concatenating text and variables using macro substitution
Concepts
Conditional Operators
Conditional Shortcut Syntax
Configuring startup tiddlers
Configuring the default TiddlerInfo tab
Consent Banner Plugin
Constructing JSON tiddlers
contains Operator
contains Operator (Examples)
Contents
ContentType
contrastcolour Macro
Contributing
ContributingTemplate

Contributor License Agreement
Contributors
Coordinate Systems
copy-to-clipboard Macro
copy-to-clipboard Macro (Examples)
Copying tiddlers between TiddlyWiki files
Core Icons
Core Macros
Core Variables
cos Operator
cos Operator (Examples)
count Operator
count Operator (Examples)
CountWidget
Creating a custom export format
Creating a splash screen
Creating and editing tiddlers
Creating journal tiddlers
Creating new toolbar buttons
Creating SubStories
csvtiddlers Macro
Current Tiddler
currentTiddler Variable
currentTiddler Variable (Examples)
Custom data-styles

Custom styles by data-tags
Custom styles by data-tiddler-title
Custom styles by user-class
Custom Styles FAQ
Custom tag pill styles
Custom Widgets
Customise TiddlyWiki
Customising search results
Customising Tiddler File Naming
Customising TiddlyWiki's user interface
Customizing EditTemplate field rendering
cycle Operator
cycle Operator (Examples)
D3 Plugin
Dashes in WikiText
Data URI
DataTiddlers
datauri Macro
datauri Macro (Examples)
Date Fields
DateFormat
datepicker plugin based on Pikaday, by kixam
Days of the Week
days Operator
days Operator (Examples)

Debugging Stylesheets
decodebase64 Operator
decodehtml Operator
decodehtml Operator (Examples)
decodeuri Operator
decodeuri Operator (Examples)
decodeuricomponent Operator
decodeuricomponent Operator (Examples)
DefaultTiddlers
Definitions
Definitions in WikiText
DeleteTiddlersCommand
Demo Tiddler List with Custom Story Tiddler Template
Demonstration: keyboard-driven-input Macro
Deprecated - What does it mean
deserialize Operator
deserialize Operator (Examples)
Deserializers
deserializers Operator
deserializers Operator (Examples)
Deutsch (Deutschland) Edition
Deutsch (Österreich) Edition
Developers
DictionaryTiddlers
DiffTextWidget

Disabling Plugins
Discover TiddlyWiki
Disqus comments plugin by bimlas
divide Operator
divide Operator (Examples)
Document Object Model
Documentation Macros
Documentation Style Guide
Dominant Append
done
DraftMechanism
Drag and Drop
DragAndDropMechanism
DraggableWidget
DroppableWidget
DropzoneWidget
dumpvariables Macro
dumpvariables Macro (Examples)
duplicateslugs Operator
Dynaview Plugin
each Operator
each Operator (Examples)
eachday Operator
eachday Operator (Examples)
Edición en Castellano

Edit Template Body Cascade
EditBitmapWidget
Editing Tiddlers with Emacs
Editing Tiddlers with Vim
Édition en Français (France)
editiondescription Operator
Editions
editions Operator
EditionsCommand
Editor toolbar
EditTextWidget
EditWidget
Else Filter Run Prefix
else Operator
else Operator (Examples)
Emergency Tiddler Export
Empty Edition
encodebase64 Operator
encodehtml Operator
encodehtml Operator (Examples)
encodeuri Operator
encodeuri Operator (Examples)
encodeuricomponent Operator
encodeuricomponent Operator (Examples)
Encryption

EncryptWidget
enlist Operator
enlist Operator (Examples)
enlist-input Operator
enlist-input Operator (Examples)
EntityWidget
Environment Variables on Node.js
ErrorWidget
escapecss Operator
escapecss Operator (Examples)
escaperegexp Operator
escaperegexp Operator (Examples)
Eucaly's Tiddly World
EventCatcherWidget
Example config-tiddlyweb-host for IIS
Example for tag Macro
Example package.json for IIS
Example Table of Contents: Expandable
Example Table of Contents: Selectively Expandable
Example Table of Contents: Selectively Expandable (custom icons)
Example Table of Contents: Simple
Example Table of Contents: Sorted Expandable
Example Table of Contents: Tabbed External
Example Table of Contents: Tabbed Internal
Example tiddlywiki.info for IIS

Example web.config for IIS
Examples
Except Filter Run Prefix
exponential Operator
exponential Operator (Examples)
Extended Listops Filters
External Attachments Plugin
ExternalImages
Favicon template.svg
Favorites by Mohammad
Features
Federatial
Federatial.png
FetchCommand
Field Editor Cascade
field Operator
field Operator (Examples)
FieldMangler Widget (Examples)
FieldManglerWidget
fields Operator
fields Operator (Examples)
FieldsWidget
FillWidget
Filter Expression
Filter Filter Run Prefix

Filter Filter Run Prefix (Examples)
filter Operator
filter Operator (Examples)
Filter Operators
Filter Parameter
Filter Run
Filter Run Prefix
Filter Run Prefix (Examples)
Filter Step
Filter Syntax
Filter Whitespace
Filtered Attribute Values
Filters
First
first Operator
first Operator (Examples)
FirstOne
FirstThree
FirstTwo
fixed Operator
fixed Operator (Examples)
floor Operator
floor Operator (Examples)
format Operator
format Operator (Examples)

Formatting in WikiText
Formatting List Results as Tables (no CSS)
Formatting List Results as Tables with CSS - Specified Columns Methods
Formatting List Results as Tables with CSS - Variable Column Method
Formatting text in TiddlyWiki
Forums
Fourth
Free Spaced Repetition Scheduler for TiddlyWiki by oflg
Friday
Full Edition
Full Text Search Plugin by Rob Hoelz
function Operator
Functions
Funding TiddlyWiki
Funding.png
Future Proof
Generating Static Sites with TiddlyWiki
GenesisWidget
get Operator
get Operator (Examples)
Get the Ring
getindex Operator
getindex Operator (Examples)
Getting Started Video
GettingStarted

GettingStarted - Android
GettingStarted - Chrome
GettingStarted - Firefox
GettingStarted - Internet Explorer
GettingStarted - iOS
GettingStarted - Node.js
GettingStarted - Online
GettingStarted - Safari
getvariable Operator
getvariable Operator (Examples)
Git
GitHub
GitHub Saver Tutorial by Mohammad
GitLab
Go to Mordor
Grok TiddlyWiki Banner
GroupedLists
GuerillaWiki
Hamlet
Hard and Soft Links
Hard Linebreaks in WikiText
Hard Linebreaks with CSS
Hard Linebreaks with CSS - Example
has Operator
has Operator (Examples)

haschanged Operator
Headings in WikiText
Height of text editor
HelloThere
HelloThumbnail
HelloThumbnail - Federatial
HelloThumbnail - Funding
HelloThumbnail - Grok TiddlyWiki
HelloThumbnail - Introduction Video
HelloThumbnail - Latest Version
HelloThumbnail - Marketplace
HelloThumbnail - Newsletter
HelloThumbnail - TiddlyWikiLinks
HelpCommand
HelpingTiddlyWiki
Hidden Setting: Default Tiddler Icon
Hidden Setting: Disable Drag and Drop
Hidden Setting: Disable Lazy Loading
Hidden Setting: Enable File Import in Editor
Hidden Setting: Filename for Save Wiki Button
Hidden Setting: HTML Parser Sandbox
Hidden Setting: More Tabs Horizontal
Hidden Setting: Navigate on Enter
Hidden Setting: New-Image Type
Hidden Setting: Retain Story ordering

Hidden Setting: Scroll Top Adjustment
Hidden Setting: Search AutoFocus
Hidden Setting: Search Minimum Length
Hidden Setting: Show Edit Preview per Tiddler
Hidden Setting: Sync Logging
Hidden Setting: Sync Polling Interval
Hidden Setting: Sync System Tiddlers From Server
Hidden Setting: Tab Index for Edit-Inputs
Hidden Setting: Tags Minimum Length
Hidden Setting: Template for Save Wiki Button
Hidden Setting: Typing Refresh Delay
Hidden Setting: ViewTemplate and EditTemplate
Hidden Settings
Hidden Settings: Import Content Types for Editor
Highlight Plugin
History of TiddlyWiki
HistoryMechanism
Horizontal Rules in WikiText
How to add a banner for GitHub contributions
How to add a new tab to the sidebar
How to apply custom styles
How to apply custom styles by tag
How to build a TiddlyWiki5 from individual tiddlers
How to change the sort order of sub-branches in a TOC macro
How to create a custom story tiddler template

How to create a custom tiddler colour rule
How to create a custom tiddler icon rule
How to create dynamic editor toolbar buttons
How to create keyboard shortcuts
How to Customize TiddlyDesktop
How to embed PDF and other documents
How to export tiddlers
How to hide the author's and other fields with CSS
How to put the last modification date in a banner
How to remove stop words
How to turn off camel case linking
How to update TiddlyWiki to the latest version
How to widen tiddlers (aka storyriver)
HTML Block Elements
HTML Entities
HTML in WikiText
HTML Links in WikiText
HyperText Markup Language
Icon Gallery
image-picker Macro
image-picker Macro (Example 1)
image-picker Macro (Example 2)
image-picker Macro (Examples)
ImageGallery Example
Images in WikiText

ImageWidget
ImportCommand
Importing Tiddlers
ImportVariablesWidget
Improving TiddlyWiki Documentation
IndexedDB Plugin by Andreas Abeck
indexes Operator
indexes Operator (Examples)
InfoMechanism
InfoPanel
InitCommand
Inline Mode WikiText
Innerwiki Plugin
Insert link
Insert picture
insertafter Operator
insertafter Operator (Examples)
insertbefore Operator
insertbefore Operator (Examples)
Installing a plugin from the plugin library
Installing custom plugins on Node.js
Installing official plugins on Node.js
Installing TiddlyWiki on Microsoft Internet Information Server
Installing TiddlyWiki on Node.js
Installing TiddlyWiki Prerelease on Node.js

Instruction Tiddlers
Interactive Git Documentation by Devin Weaver
Interchangeable Filter Run Prefixes
Intersection Filter Run Prefix
Intersection Filter Run Prefix (Examples)
Introducing TiddlyDesktop Video
Introduction to filter notation
Introduction to Lists
Introduction Video
Introduction Video Thumbnail.jpg
is Operator
is Operator (Examples)
Japanese (Japan) Edition
JavaScript
JavaScript Object Notation
JeremyRuston
Jermolene
join Operator
join Operator (Examples)
JSON in TiddlyWiki
Json Mangler plugin by Joshua Fontany
jsonextract Operator
jsonget Operator
jsonindexes Operator
jsonset Operator

jsonset Operator (Examples)
jsonstringify Operator
jsonstringify Operator (Examples)
jsontiddler Macro
JSONTiddlers
jsontiddlers Macro
JSONTiddlerWidget
jsontype Operator
JSZip Plugin
KaTeX Plugin
kdi-demo-configtid
Keyboard Codes
Keyboard Shortcut Descriptor
Keyboard Shortcut Tiddler
keyboard-driven-input Macro
KeyboardShortcuts
KeyboardWidget
Kill the Dragon
Kin filter operator by bimlas
Kookma Plugin Library by Mohammad
Korean (Korea Republic) Edition
Language Icon: ca-ES
Language Icon: cs-CZ
Language Icon: da-DK
Language Icon: de-AT

Language Icon: de-DE
Language Icon: el-GR
Language Icon: en-US
Language Icon: es-ES
Language Icon: fr-FR
Language Icon: hi-IN
Language Icon: ia-IA
Language Icon: it-IT
Language Icon: ja-JP
Language Icon: ko-KR
Language Icon: nl-NL
Language Icon: pa-IN
Language Icon: pt-PT
Language Icon: ru-RU
Language Icon: sk-SK
Language Icon: sv-SE
Language Icon: zh-Hans
Language Icon: zh-Hant
LanguageGallery Example
Languages
last Operator
last Operator (Examples)
Latest
LazyLoading
Leaflet maps plugin by Sylvain Comte

Learning
Legacy
length Operator
length Operator (Examples)
LetWidget
levenshtein Operator
levenshtein Operator (Examples)
License
limit Operator
limit Operator (Examples)
lingo Macro
lingo Macro (Examples)
LinkCatcherWidget
Linking in WikiText
links Operator
links Operator (Examples)
LinkWidget
list Operator
list Operator (Examples)
list-links Macro
list-links Macro (Examples)
list-links-draggable Macro
list-links-draggable Macro (Examples)
list-tagged-draggable Macro
list-tagged-draggable Macro (Examples)

list-thumbnails Macro
list-thumbnails Macro (Examples)
listed Operator
listed Operator (Examples)
ListenCommand
ListField
ListopsData
Lists in WikiText
ListWidget
Literal Attribute Values
LoadCommand
Locator plugin by bimlas
log Operator
log Operator (Examples)
LogWidget
lookup Operator
lookup Operator (Examples)
lowercase Operator
lowercase Operator (Examples)
Macro Call Syntax
Macro Calls
Macro Calls in WikiText
Macro Calls in WikiText (Examples)
Macro Definition Syntax
Macro Definitions

Macro Definitions in WikiText
Macro Parameter Handling
Macro Pitfalls
Macro Syntax
MacroCallWidget
Macros
Macros in WikiText
Make the beds
makedatauri Macro
makedatauri Macro (Examples)
MakeLibraryCommand
makepatches and applypatches Operator (Examples)
makepatches Operator
Making a custom journal button
Making curved text with SVG
Manually installing a plugin
Map Filter Run Prefix
Map Filter Run Prefix (Examples)
Markdown
Markdown Plugin
match Operator
match Operator (Examples)
Mathematics Operators
MathJax Plugin by Martin Kantor
MathML

max Operator
max Operator (Examples)
maxall Operator
maxall Operator (Examples)
Mechanisms
median Operator
median Operator (Examples)
Meetups
MessageCatcherWidget
MessageHandlerWidgets
Messages
Milk
min Operator
min Operator (Examples)
minall Operator
minall Operator (Examples)
minlength Operator
minlength Operator (Examples)
Mobile Drag And Drop Shim Plugin
Modals
modifier Variable
Modifying JSON tiddlers
moduleproperty Operator
moduleproperty Operator (Examples)
Modules

modules Operator
modules Operator (Examples)
ModuleType
moduletypes Operator
moduletypes Operator (Examples)
Monday
More actions
Motovun Jack.ascii
Motovun Jack.jpg
Motovun Jack.pdf
Motovun Jack.svg
move Operator
move Operator (Examples)
multiply Operator
multiply Operator (Examples)
MultiTiddlerFiles
MultiTiddlerFileSyntax
Named Filter Run Prefix
NamedCommandParameters
namespace Variable
Naming of System Tiddlers
Navigating between open tiddlers
NavigatorWidget
negate Operator
negate Operator (Examples)

New Release Banner
Newnham Horizon.jpg
next Operator
next Operator (Examples)
node-webkit
Node.js
Notes for upgrading to 5.0.11-beta
Notes for upgrading to 5.0.8-beta
Notifications
now Macro
now Macro (Examples)
nsort Operator
nsort Operator (Examples)
nsortcs Operator
nsortcs Operator (Examples)
nth Operator
nth Operator (Examples)
NW.js
OfficialPlugins
Open Collective
Open Collective Logo
OpenSource
Or Filter Run Prefix
Order of Tagged Tiddlers
order Operator

Osmosoft
Other Resources
OutputCommand
OXTWIG
pad Operator
pad Operator (Examples)
Paragraphs in WikiText
ParametersWidget
ParametersWidget (Examples)
PasswordCommand
PasswordWidget
Percent Encoding
Performance
Performance Instrumentation
PerlinLight.jpg
PermaLinks
PESpot Lesson Planner by Patrick Detzner
Philosophy of Tiddlers
Pinstripe.gif
Places where the parser ignores WikiText
Platforms
Plugin Dependencies
Plugin Editions
Plugin Fields
Plugin Information Tiddlers

Plugin Ordering
Plugin Types
PluginFolders
PluginMechanism
Plugins
Plugins by TheDiveO
PluginsCS
plugintiddlers Operator
plugintiddlers Operator (Examples)
PopupMechanism
power Operator
power Operator (Examples)
Pragma
Pragma: \define
Pragma: \function
Pragma: \import
Pragma: \parameters
Pragma: \parsermode
Pragma: \procedure
Pragma: \rules
Pragma: \whitespace
Pragma: \widget
Pragmas
precision Operator
precision Operator (Examples)

prefix Operator
prefix Operator (Examples)
prepend Operator
prepend Operator (Examples)
Preserving open tiddlers at startup
previous Operator
previous Operator (Examples)
Procedure Calls
Procedure Definitions
Procedure Parameter Handling
Procedures
Product Hunt Link
product Operator
product Operator (Examples)
ProductHunt-Badge.svg
Projectify by Nicolas Petton
putafter Operator
putafter Operator (Examples)
putbefore Operator
putbefore Operator (Examples)
putfirst Operator
putfirst Operator (Examples)
putlast Operator
putlast Operator (Examples)
Q: How can I style a tiddler if it has "this" AND "that" tag?

Q: How can I style a tiddler if it has "this" OR "that" tag?
Q: How can I use a custom field to style a tiddler?
Q: Is there a way to create dynamic stylesheets?
Q: What if a tiddler has no tags?
qualify Macro
qualify Macro (Examples)
QualifyWidget
Quick Start
Quick Start: Desktop
Quick Start: DIY
Quick Start: Tiddlyhost
Quick Start: Xememex
Quine
RadioWidget
RadioWidget Example
RadioWidget ExampleTemplate
Railroad Diagrams
Railroad Plugin
range Operator
RangeWidget
RangeWidget Example
rboue's plugins for TiddlyWiki
Reading data from JSON tiddlers
ReadMe
ReadMeBinFolder

Reddit Logo
Reduce Filter Run Prefix
Reduce Filter Run Prefix (Examples)
reduce Operator
reduce Operator (Examples)
Reference
Reference Tiddlers
Refnotes by Mohammad
RefreshMechanism
RefreshThrottling
RegExp in Tiddlywiki by Mohammad
regexp Operator
regexp Operator (Examples)
Release 5.0.0-alpha.11
Release 5.0.0-alpha.12
Release 5.0.0-alpha.13
Release 5.0.0-alpha.14
Release 5.0.0-alpha.15
Release 5.0.0-alpha.16
Release 5.0.0-alpha.17
Release 5.0.1-alpha
Release 5.0.10-beta
Release 5.0.11-beta
Release 5.0.12-beta
Release 5.0.13-beta

Release 5.0.14-beta
Release 5.0.15-beta
Release 5.0.16-beta
Release 5.0.17-beta
Release 5.0.18-beta
Release 5.0.2-beta
Release 5.0.3-beta
Release 5.0.4-beta
Release 5.0.5-beta
Release 5.0.6-beta
Release 5.0.7-beta
Release 5.0.8-beta
Release 5.0.9-beta
Release 5.1.0
Release 5.1.1
Release 5.1.10
Release 5.1.11
Release 5.1.12
Release 5.1.13
Release 5.1.14
Release 5.1.15
Release 5.1.16
Release 5.1.17
Release 5.1.18
Release 5.1.19

Release 5.1.2
Release 5.1.20
Release 5.1.21
Release 5.1.22
Release 5.1.23
Release 5.1.3
Release 5.1.4
Release 5.1.5
Release 5.1.6
Release 5.1.7
Release 5.1.8
Release 5.1.9
Release 5.2.0
Release 5.2.1
Release 5.2.2
Release 5.2.3
Release 5.2.4
Release 5.2.5
Release 5.2.6
Release 5.2.7
Release 5.3.0
Release 5.3.1
Release 5.3.2
Release 5.3.3
Releases

ReleaseTemplate
remainder Operator
remainder Operator (Examples)
remove Operator
remove Operator (Examples)
removeprefix Operator
removeprefix Operator (Examples)
removesuffix Operator
removesuffix Operator (Examples)
RenderCommand
RenderTiddlerCommand
RenderTiddlersCommand
replace Operator
replace Operator (Examples)
ReportingBugs
resolvepath Macro
resolvepath Macro (Examples)
rest Operator
rest Operator (Examples)
Résumé Builder Edition
RevealWidget
reverse Operator
reverse Operator (Examples)
Rice Pudding
RoadMap

round Operator
round Operator (Examples)
SafeMode
sameday Operator
sameday Operator (Examples)
Sample Headings 1,2,3
Sample Headings 3,4,5
Sample Headings 4,5,6
Sample Tiddler Template
SampleModal
SampleNotification
SampleTabFour
SampleTabOne
SampleTabThree
SampleTabTwo
sampletag1
sampletag2
SampleTemplate
SampleTiddlerFirst
SampleTiddlerSecond
SampleTiddlerThird
SampleWindowTemplate
SampleWizard
SampleWizard2
Saturday

SaveCommand
SaveTiddlerCommand
SaveTiddlersCommand
SaveTrail Plugin
SaveWikiFolderCommand
Saving
Saving on a PHP Server
Saving on Android
Saving on Browser with File System Access API
Saving on Browser with TiddlyStow
Saving on iPad/iPhone
Saving on TiddlyDesktop
Saving on TiddlyHost
Saving on TiddlySpot
Saving on TidGi Desktop
Saving to a Git service
Saving via a Minimal Ruby Server
Saving via a Minimal Web Server
Saving via WebDAV
Saving with FireFox
Saving with Polly
Saving with the HTML5 saver
Saving with TiddlyFox
Saving with TiddlyFox on Android
Saving with TiddlyIE

Saving with TW Receiver
SavingMechanism
Scalability
Scripts for TiddlyWiki on Node.js
ScrollableWidget
search Operator
search Operator (Examples)
search-replace Operator
search-replace Operator (Examples)
Searching in TiddlyWiki
Searchwikis by Mohammad
Second
SecondOne
SecondThree
SecondThreeOne
SecondThreeThree
SecondThreeTwo
SecondTwo
SeeAlso by Matias Goldman
Selection Constructors
SelectWidget
Semantic Colors by Mohammad
sentencecase Operator
sentencecase Operator (Examples)
ServerCommand

Serving TW5 from Android
SetFieldCommand
SetMultipleVariablesWidget
Setting a favicon
Setting a page background image
SetVariableWidget
SetWidget
sha256 Operator
sha256 Operator (Examples)
shadowsource Operator
shadowsource Operator (Examples)
ShadowTiddlers
Share Plugin
Sharing a TiddlyWiki on Dropbox
Sharing your tiddlers with others
Shiraz by Mohammad
Shortcut Filter Run Prefix
sign Operator
sign Operator (Examples)
Signing the Contributor License Agreement
Simple ways to write protect tiddlers
Simple Zork-like Game by Jed Carty
sin Operator
sin Operator (Examples)
SingleFileApplication

SinglePageApplication
Slider by Mohammad
SlotWidget
slugify Operator
Some of the things you can do with TiddlyWiki
Sort Filter Run Prefix
Sort Filter Run Prefix (Examples)
sort Operator
sort Operator (Examples)
sortan Operator
sortan Operator (Examples)
sortby Operator
sortby Operator (Examples)
sortcs Operator
sortcs Operator (Examples)
sortsub Operator
sortsub Operator (Examples)
Spelling
split Operator
split Operator (Examples)
splitbefore Operator
splitbefore Operator (Examples)
splitregexp Operator
splitregexp Operator (Examples)
standard-deviation Operator

standard-deviation Operator (Examples)
Stanford JavaScript Crypto Library
StartupActions
StateMechanism
Story River
Story Tiddler Template
Story Tiddler Template Cascade
storyTiddler Variable
storyTiddler Variable (Examples)
storyviews Operator
storyviews Operator (Examples)
stringify Operator
stringify Operator (Examples)
Structuring TiddlyWiki
Styles and Classes in WikiText
Stylesheet Macros
subfilter Operator
subfilter Operator (Examples)
substitute Operator
substitute Operator (Examples)
Substituted Attribute Values
subtiddlerfields Operator
subtiddlerfields Operator (Examples)
subtract Operator
subtract Operator (Examples)

suffix Operator
suffix Operator (Examples)
sum Operator
sum Operator (Examples)
Sunday
SystemTag: $:/tags/AboveStory
SystemTag: $:/tags/AdvancedSearch
SystemTag: $:/tags/AdvancedSearch/FilterButton
SystemTag: $:/tags/Alert
SystemTag: $:/tags/BelowStory
SystemTag: $:/tags/ClassFilters/PageTemplate
SystemTag: $:/tags/ClassFilters/TiddlerTemplate
SystemTag: $:/tags/ControlPanel
SystemTag: $:/tags/ControlPanel/Advanced
SystemTag: $:/tags/ControlPanel/Appearance
SystemTag: $:/tags/ControlPanel/Info
SystemTag: $:/tags/ControlPanel/Plugins
SystemTag: $:/tags/ControlPanel/Saving
SystemTag: $:/tags/ControlPanel/Settings
SystemTag: $:/tags/ControlPanel/Toolbars
SystemTag: $:/tags/EditorToolbar
SystemTag: $:/tags/EditorTools
SystemTag: $:/tags/EditPreview
SystemTag: $:/tags/EditTemplate
SystemTag: $:/tags/EditToolbar

SystemTag: $:/tags/Exporter
SystemTag: $:/tags/Filter
SystemTag: $:/tags/Global
SystemTag: $:/tags/Global/View
SystemTag: $:/tags/Global/View/Body
SystemTag: $:/tags/Image
SystemTag: $:/tags/ImportPreview
SystemTag: $:/tags/KeyboardShortcut
SystemTag: $:/tags/Layout
SystemTag: $:/tags/Macro
SystemTag: $:/tags/Macro/View
SystemTag: $:/tags/Macro/View/Body
SystemTag: $:/tags/Manager/ItemMain
SystemTag: $:/tags/Manager/ItemSidebar
SystemTag: $:/tags/MoreSideBar
SystemTag: $:/tags/MoreSideBar/Plugins
SystemTag: $:/tags/PageControls
SystemTag: $:/tags/PageTemplate
SystemTag: $:/tags/Palette
SystemTag: $:/tags/PluginLibrary
SystemTag: $:/tags/RawMarkup
SystemTag: $:/tags/RawMarkupWikified
SystemTag: $:/tags/RawMarkupWikified/BottomBody
SystemTag: $:/tags/RawMarkupWikified/TopBody
SystemTag: $:/tags/RawMarkupWikified/TopHead

SystemTag: $:/tags/RawStaticContent
SystemTag: $:/tags/RemoteAssetInfo
SystemTag: $:/tags/SearchResults
SystemTag: $:/tags/ServerConnection
SystemTag: $:/tags/SideBar
SystemTag: $:/tags/SideBarSegment
SystemTag: $:/tags/StartupAction
SystemTag: $:/tags/StartupAction/Browser
SystemTag: $:/tags/StartupAction/Node
SystemTag: $:/tags/Stylesheet
SystemTag: $:/tags/TagDropdown
SystemTag: $:/tags/TextEditor/Snippet
SystemTag: $:/tags/TiddlerInfo
SystemTag: $:/tags/TiddlerInfo/Advanced
SystemTag: $:/tags/TiddlerInfoSegment
SystemTag: $:/tags/ToolbarButtonStyle
SystemTag: $:/tags/TopLeftBar
SystemTag: $:/tags/TopRightBar
SystemTag: $:/tags/ViewTemplate
SystemTag: $:/tags/ViewTemplate/Subtitle
SystemTag: $:/tags/ViewToolbar
SystemTags
SystemTiddlers
TabbedExampleType
Table Classes, Captions, Headers and Footers

table-example
Table-of-Contents Macros
Table-of-Contents Macros (Examples)
TableOfContents
Tables in WikiText
Tables in WikiText CSS Utility Classes
tabs Macro
tabs Macro (Examples)
tag Macro
tag Macro (Examples)
tag Operator
tag Operator (Examples)
tag-picker Macro
tag-pill Macro
Tagging
tagging Operator
tagging Operator (Examples)
tags Operator
tags Operator (Examples)
TagTiddlers
tan Operator
tan Operator (Examples)
task
TaskManagementExample
TaskManagementExample (Draggable)

TaskManagementExampleDraggableTemplate
Technical Prose Style
TemplateTiddlers
Temporary Tiddlers
Ten reasons to switch to TiddlyWiki
Testimonials - Joe Armstrong
Testimonials - Network World
Testimonials - Product Hunt
Text preview
Text-Slicer Edition
TextReference
TextWidget
The Extended Listops Filters
The First Rule of Using TiddlyWiki
Then Filter Run Prefix
Then Filter Run Prefix (Examples)
then Operator
then Operator (Examples)
Third
ThirdOne
ThirdThree
ThirdTwo
thisTiddler Variable
thisTiddler Variable (Examples)
thumbnail Macro

thumbnail Macro (Examples)
Thursday
Tiddler Colour Cascade
Tiddler Commander by Mohammad
Tiddler Fishes.svg
Tiddler Icon Cascade
Tiddler Poster.png
Tiddler Structure
Tiddler Title Policy
TiddlerFields
TiddlerFiles
TiddlerLinks
Tiddlers
TiddlerWidget
TiddlyBucket - Save to AWS or Google Storage
TiddlyChrome by Arlen Beiler
TiddlyClip by buggyjay
TiddlyDesktop
TiddlyDesktop Releases
TiddlyDrive Add-on for Google Drive by Joshua Stubbs
TiddlyFox
TiddlyFox Apocalypse
TiddlyFox Apocalypse.png
Tiddlyhost
Tiddlyhost Logo

TiddlyIE
TiddlyMap Plugin by Felix Küppers
TiddlyServer by Arlen Beiler
Tiddlyshow by Mohammad
TiddlySpace
TiddlyWeb
TiddlyWeb JSON tiddler format
TiddlyWiki
TiddlyWiki Archive
TiddlyWiki Camp Paris
TiddlyWiki Classic.png
TiddlyWiki Cloud
TiddlyWiki European Meetup 2016
TiddlyWiki European Meetup 2017
TiddlyWiki extensions for Sublime Text 3 by roma0104
TiddlyWiki extensions for ViM
TiddlyWiki Hangouts
TiddlyWiki in the Sky for TiddlyWeb
TiddlyWiki Marketplace
TiddlyWiki Marketplace Banner
TiddlyWiki Newsletter
TiddlyWiki Newsletter Badge.png
TiddlyWiki on Firefox for Android Video
TiddlyWiki on Node.js
TiddlyWiki Releases

TiddlyWiki Video Tutorials by Francis Meetze
tiddlywiki.files Files
tiddlywiki.info Files
TiddlyWiki.mp3
TiddlyWiki2ReadMe
TiddlyWiki5
TiddlyWiki5 Versioning
TiddlyWikiClassic
TiddlyWikiFolders
TiddlyWikiLinks
Tidme by oflg
timeline Macro
timeline Macro (Examples)
Timelines by Mohammad
Timimi: WebExtension and Native Host by Riz
Tinka by Andreas Hahn
Title List
title Operator
title Operator (Examples)
Title Selection
titlecase Operator
titlecase Operator (Examples)
Todolist by Mohammad
toggle Operator
toggle Operator (Examples)

Transcluded Attribute Values
TranscludeWidget
Transclusion
Transclusion and Substitution
Transclusion Basic Usage
Transclusion in WikiText
transclusion Variable
transclusion Variable (Examples)
Transclusion with Templates
Translate TiddlyWiki into your language
TranslationMechanism
translink Macro
translink Macro (Examples)
Trashbin by Mohammad
tree Macro
tree Macro (Examples)
tree-macro-example-car
tree-macro-example-car-boot
tree-macro-example-car-boot-handle
tree-macro-example-car-boot-lock
tree-macro-example-car-roof
tree-macro-example-car-roof-aerial
tree-macro-example-car-roof-rails
tree-macro-example-house
tree-macro-example-house-attic

tree-macro-example-house-attic-roof
tree-macro-example-house-attic-window
tree-macro-example-house-garden
tree-macro-example-house-garden-lawn
tree-macro-example-house-garden-shed
tree-macro-example-house-kitchen
tree-macro-example-house-kitchen-sink
tree-macro-example-house-kitchen-table
tree-macro-example-house-kitchen-window
TriggeringWidgets
trim Operator
trim Operator (Examples)
TriTarget.org by Devin Weaver
trunc Operator
trunc Operator (Examples)
Tuesday
Tutorials
tv-adjust-heading-level Variable
tv-adjust-heading-level Variable (Examples)
tv-auto-open-on-import Variable
tv-config-static Variable
tv-config-toolbar-class Variable
tv-config-toolbar-class Variable (Examples)
tv-config-toolbar-icons Variable
tv-config-toolbar-icons Variable (Examples)

tv-config-toolbar-text Variable
tv-config-toolbar-text Variable (Examples)
tv-filter-export-link Variable
tv-get-export-image-link Variable
tv-get-export-image-link Variable (Examples)
tv-get-export-link Variable
tv-get-export-path Variable
tv-history-list Variable
tv-show-missing-links Variable
tv-story-list Variable
tv-tiddler-preview Variable
tv-wikilink-template Variable
tv-wikilink-tooltip Variable
tv-wikilink-tooltip Variable (Examples)
tv-wikilinks Variable
tv-wikilinks Variable (Examples)
TW-Scripts by Mohammad
TW2Parser Plugin
TW5-firebase: TiddlyWiki5 for Google Firebase by Peter Neumark
TW5-SingleExecutable by Jed Carty
TWaddle by Matias Goldman
TWCommunitySearch
TWEUM Thumbnail.jpg
Twexe: Single File Tiddlywiki5 executable
Twitter Plugin

twproxy by Steve Gattuso
Typed Blocks in WikiText
Typography
Uninstalling a plugin
Uninstalling a plugin with Node.js
unique Operator
unique Operator (Examples)
UnpackPluginCommand
untagged Operator
untagged Operator (Examples)
untrunc Operator
untrunc Operator (Examples)
unusedtitle Macro
unusedtitle Macro (Examples 1)
unusedtitle Macro (Examples)
UpgradeMechanism
Upgrading
Upgrading TiddlyWiki on Node.js
uppercase Operator
uppercase Operator (Examples)
URI
Using a custom path prefix with the client-server edition
Using Excise
Using HTTPS
Using links to navigate between tiddlers

Using Stamp
Using Stylesheets
Using SVG
Using the external JavaScript template
Using the integrated static file server
Using the read-only single tiddler view
Using TiddlyWiki for GitHub project documentation
Using TiddlyWiki on Node.js
Utility by Mohammad
Utility Classes
Variable Attribute Values
Variable Usage
Variables
Variables in WikiText
variables Operator
variables Operator (Examples)
variance Operator
variance Operator (Examples)
VarsWidget
VerboseCommand
version Macro
version Macro (Examples)
VersionCommand
Videos
View Template Body Cascade

View Template Title Cascade
ViewWidget
vis.js Timeline by emkay, revived by kixam
Visible Transclusions
Web Developer Tools
WebServer
WebServer Anonymous Access
WebServer API
WebServer API: Delete Tiddler
WebServer API: Force Basic Authentication Login
WebServer API: Get All Tiddlers
WebServer API: Get Favicon
WebServer API: Get File
WebServer API: Get Rendered Tiddler
WebServer API: Get Server Status
WebServer API: Get Tiddler
WebServer API: Get Wiki
WebServer API: Put Tiddler
WebServer Authentication
WebServer Authorization
WebServer Basic Authentication
WebServer Guides
WebServer Header Authentication
WebServer Parameter: admin
WebServer Parameter: anon-username

WebServer Parameter: authenticated-user-header
WebServer Parameter: credentials
WebServer Parameter: csrf-disable
WebServer Parameter: debug-level
WebServer Parameter: host
WebServer Parameter: password
WebServer Parameter: path-prefix
WebServer Parameter: port
WebServer Parameter: readers
WebServer Parameter: required-plugins
WebServer Parameter: root-render-type
WebServer Parameter: root-serve-type
WebServer Parameter: root-tiddler
WebServer Parameter: system-tiddler-render-template
WebServer Parameter: system-tiddler-render-type
WebServer Parameter: tiddler-render-template
WebServer Parameter: tiddler-render-type
WebServer Parameter: tls-cert
WebServer Parameter: tls-key
WebServer Parameter: tls-passphrase
WebServer Parameter: username
WebServer Parameter: writers
WebServer Parameters
Wednesday
What happened to the original TiddlyWiki?

Widdly by Opennota
Widget Attributes
Widget Documentation Style Guide
WidgetMessage: tm-add-field
WidgetMessage: tm-add-tag
WidgetMessage: tm-auto-save-wiki
WidgetMessage: tm-browser-refresh
WidgetMessage: tm-cancel-tiddler
WidgetMessage: tm-clear-password
WidgetMessage: tm-close-all-tiddlers
WidgetMessage: tm-close-all-windows
WidgetMessage: tm-close-other-tiddlers
WidgetMessage: tm-close-tiddler
WidgetMessage: tm-close-window
WidgetMessage: tm-copy-to-clipboard
WidgetMessage: tm-delete-tiddler
WidgetMessage: tm-download-file
WidgetMessage: tm-edit-bitmap-operation
WidgetMessage: tm-edit-text-operation
WidgetMessage: tm-edit-tiddler
WidgetMessage: tm-focus-selector
WidgetMessage: tm-fold-all-tiddlers
WidgetMessage: tm-fold-other-tiddlers
WidgetMessage: tm-fold-tiddler
WidgetMessage: tm-full-screen

WidgetMessage: tm-home
WidgetMessage: tm-http-cancel-all-requests
WidgetMessage: tm-http-request
WidgetMessage: tm-http-request Example - Random Dog
WidgetMessage: tm-http-request Example - Zotero
WidgetMessage: tm-import-tiddlers
WidgetMessage: tm-load-plugin-from-library
WidgetMessage: tm-load-plugin-library
WidgetMessage: tm-login
WidgetMessage: tm-logout
WidgetMessage: tm-modal
WidgetMessage: tm-navigate
WidgetMessage: tm-new-tiddler
WidgetMessage: tm-notify
WidgetMessage: tm-open-external-window
WidgetMessage: tm-open-window
WidgetMessage: tm-perform-import
WidgetMessage: tm-permalink
WidgetMessage: tm-permaview
WidgetMessage: tm-print
WidgetMessage: tm-relink-tiddler
WidgetMessage: tm-remove-field
WidgetMessage: tm-remove-tag
WidgetMessage: tm-rename-tiddler
WidgetMessage: tm-save-tiddler

WidgetMessage: tm-save-wiki
WidgetMessage: tm-scroll
WidgetMessage: tm-server-refresh
WidgetMessage: tm-set-password
WidgetMessage: tm-unfold-all-tiddlers
WidgetMessage: tm-unload-plugin-library
Widgets
Widgets in WikiText
Wiki
WikificationMechanism
WikifyWidget
WikiText
WikiText parser mode transitions
WikiText parser mode: HTML examples
WikiText parser mode: macro examples
WikiText parser mode: transclusion examples
WikiText Parser Modes
Windows HTA Hack
Working with the TiddlyWiki5 repository
Working with the TiddlyWiki5 repository video
Working with TiddlyWiki
Xememex
Xememex Logo
XLSX Utilities Edition
zth Operator

zth Operator (Examples)
中文社区 - Chinese Community

AlphaReleases

20th December 2023 at 12:30pm
AlphaReleaseNotes Releases

Here are the details of the alpha releases of TiddlyWiki5. See TiddlyWiki5 Versioning for details of how releases are named.

5.0.1-alpha
5.0.0-alpha.17
5.0.0-alpha.16
5.0.0-alpha.15
5.0.0-alpha.14
5.0.0-alpha.13
5.0.0-alpha.12
5.0.0-alpha.11

Release 5.0.1-alpha

Released 6th December 2013 at 18:53

See GitHub for detailed change history of this release

Improvements

- Changes to the importing process to enable a smoother Upgrading process
  - Ignores attempts to import plugins that are older than currently installed plugins
  - System tiddlers are now imported as usual
- If `$:/theme` isn't defined or refers to a missing tiddler, then fallback through **Snow White** to **Vanilla**. This means that `empty.html` now defaults to **Snow White**
- Added support for Block Quotes in WikiText

    - Contributed by *StephanHradek* (@Skeeve on GitHub)

Bug fixes

- Fixed bug that was preventing `$:/tags/PageControls` tiddlers from being reordered

Internal changes

- Changed the TiddlyWiki5 Versioning policy
    - Beta releases will be `5.0.x-beta` and the final release will be `5.1.x`

Alternative page layouts

26th November 2021 at 1:03pm

Customise TiddlyWiki

Introduced in v5.1.23 You can have multiple alternative page layouts and switch between them. To see a list of available layouts and switch between them, use the keyboard shortcut `ctrl-shift-L`.

Creating an alternative page layout

Creating an alternative layout goes beyond *adding or removing features* from the default interface and allows you to create an entirely new layout from scratch.

To create an alternative page layout and have the ability to switch to it, you need to create an alternative page template tiddler with the SystemTag: $:/tags/Layout.

This alternative page template can either be a tweaked and modified version of the **default page template**, or something entirely different. The layout switching mechanism requires that your page template tiddler has the fields `name` and `description`, which are used in the listing in the switching user interface.

Amazon Web Services Plugin

3rd July 2017 at 9:33pm

OfficialPlugins

The Amazon Web Services Plugin provides several tools for working with Amazon Web Services:

- Templates for saving a TiddlyWiki as a single JavaScript file in a ZIP file that can be executed as an AWS Lambda function. In this form, TiddlyWiki is a self contained single file containing both code and data, just like the standalone HTML file configuration
- Commands that can be used to interact with AWS services, under both the Node.js and Lambda configurations of TiddlyWiki

Anchor Links using HTML

30th November 2020 at 7:43am

HTML in WikiText

HTML Anchor Links in Tiddlers

- What do they do?

- How do I make them?
- Limitations and things to look out for
- More examples
- Suggestions for HTML elements to use
- Suggestions for target names to use
- Avoiding duplicate IDs when transcluding
- Style Hints

What do they do?

In TiddlyWiki anchor links can help us link to target points and distinct sections within rendered tiddlers. They can help the reader navigate longer tiddler content.

For example this link, Bottom, should take you to the bottom of this tiddler. The "Back to the top" link below should return you back to the top of this tiddler.

Back to the top

How do I make them?

There are only 2 steps to making anchor links.

- First, create the target point (called an 'anchor') where you want it.
- Second, add a link to it where the reader might need it.

That's it.

Step 1: Create an 'anchor' (link target)

```
<a id="#Bottom_of_tiddler"></a>
```

- Add an **id** attribute to any HTML5 tag and place a single `#` before the 'address' value. Here are some suggestions of tags you can use.
- Place this code at the target location. E.g. at the bottom of the tiddler.

Step 2: Create a link to the target anchor

```
<a href="##Bottom_of_tiddler">Bottom</a>
```

- Create an `<a href>` link using exactly the same 'address' value as the target but with double `##` characters preceding it.
- Place this code where it will most help the reader.

The above example can be adapted to work for many situations in TiddlyWiki, such as the table of contents at the top of this tiddler.

Is it really that simple?

Yes, it's that simple. BUT... in TiddlyWiki there are some small differences from standard HTML5 anchor links and some specific TiddlyWiki limitations.

Back to the top

Limitations and things to watch out for

Step 1: Creating the target anchor

- Unlike standard HTML5, in TiddlyWiki you must put a single `#` character in front of the target address value that follows the **`id`** attribute. (Notice the single `#` character used in the #Bottom_of_tiddler in Example 1.)
- Unlike standard HTML5, TiddlyWiki anchor links are case sensitive. In the example above #Bottom_of_tiddler works, but #bottom_of_Tiddler wouldn't.
- Avoid using the % character in your link, as some browsers have a bug where navigation to anchors containing a % character can sometimes fail (see warning below).
- Avoid using any whitespace in your link (see note 1 below).
- Each link address should be unique in the TiddlyWiki. If transclusions create multiple instances of the same **`id`** attribute, navigation will take place to the first instance found that satisfies the above criteria. To avoid this, you can use the qualify macro in both the **`id`** and the **`href`** attributes in a similar style to the example given below: Avoiding duplicate IDs when transcluding.

Step 2: Creating the link to the target

- Remember that the **`href`** value should start with **two** `#` characters, one more than the **`id`** value of the target anchor.
- Remember that TiddlyWiki anchor links are case sensitive

Step 3: Navigating to the target

Normally you can click on the link and TiddlyWiki will scroll the window so the target anchor is at the top of the window, but there are several limitations:

- Anchor links in TiddlyWiki do not work if the tiddler is not in the Story River, or is not displayed in the sidebar.
- Anchor links in TiddlyWiki do not work if the tidlder is in the Story River or sidebar but anchor marked target point is not visible on the

page (e.g., the tiddler is folded or the anchor is within a non-displaying $reveal widget). Ctrl+Click on such an anchor link will open a new browser window/tab loading the base web page and will not navigate to the expected tiddler.

- Anchor links in Tiddlywiki do not work if Control Panel > Appearance > Story View is set to Zoomin (the single page view) and the tiddler is not at the top of the story river (i.e. not displayed).
- When an anchor link is clicked the browser's navigation history will be updated with the link's href value appended to the web page's URL. This will happen even if the Control Panel > Settings > Navigation History value is set to "Do not update history".

Note

1. The HTML5 specification for the **id** attribute says that its value should not contain any whitespace. Currently (Dec 2020) an **id** value with whitespace can sometimes work. However, it may not work in all browsers and is not guaranteed to work in the future.



Some browsers may fail to navigate to an anchor target whose **id** value includes a `%` character followed by 2 digits. This is a bug in the browser and nothing that TiddlyWiki can fix. To be safe, avoid using the `%` character.

Back to the top

More examples

In each case the target code is placed at the destination the reader navigates to, the link code is placed where it will help the reader navigate from.

```
<h2 id="#heading-01"> My Target Heading</h2>

<a href="##heading-01">Link to My Target Heading</a>
```

This looks like:

My Target Heading

Link to My Target Heading

```
<section id="#unique001">

!! My Target section

A `<section>` HTML tag is generally used to define sections in an HTML document, such as chapters,
headers, footers, or any other sections of the document.
</section>

Clicking this anchor link goes to: <a href="##unique001">Link to My Target Section</a>
```

This looks like:

My Target section

A `<section>` HTML tag is generally used to define sections in an HTML document, such as chapters, headers, footers, or any other sections of the document.

Clicking this anchor link goes to: Link to My Target Section

Back to the top

Suggestions for HTML elements to use

Some suggestions for the HTML element you choose for marking a position using the **id** attribute:

Element	Sample code	Observations
anchor		Should work perfectly everywhere
heading	<h2 id="#a-new-heading">A New Heading</h2>	Headings can have **id** attributes as well
section	<section id="#sect001-of-TiddlerName">This is a new section</section>	If you use the `section` element it should probably have an **id** of its own
span	Marker 007 is here!	`span` elements can be used to insert an **id** mid-paragraph

Back to the top

Suggestions for target names to use

	Sample code	Observations
Recommended		Include the tiddler name for easy unique **id**s
Avoid: whitespace	<h2 id="#0 0 1">A New Heading</h2>	Whitespace is not guaranteed to work in all browsers
Avoid: %	<section id="#sect001%20of%20TiddlerName">This is a new section</section>	URI encoding (turning space into %20 and so on) may not work in some browsers
Avoid: duplicate IDs	Marker 007 is here!	**id** values should be unique within entire wiki

Avoid: duplicate IDs	A different marker 007	If **id** values are duplicated, one will be ignored

Back to the top

Avoiding duplicate IDs when transcluding

The **id** and anchor link to this header has been made as follows:

```
<a href=<<qualify "##qualify-example">>>Avoiding duplicate IDs when transcluding</a>

<h2 id=<<qualify "#qualify-example">>>Avoiding duplicate IDs when transcluding</h2>
```

For this to work:

- The **href** and **id** values use the macro invocation syntax for attributes, using the qualify Macro to create a unique **id** value.
- Both the **id** and the anchor link must be in the same tiddler.
- There must not be any quote characters between the `=` and the `<<`.
- The double quotes around the target name are optional, but recommended.
- There should be a total of three `>` characters after the target name, two `>>` to close the `<<qualify` macro, and one `>` to close the `<a` tag.

Back to the top

Style Hints

If you don't want the link to look like an external link, give the `<a>` element's **`class`** attribute the value **tc-tiddlylink** and if you want the link to also be italic & bold, give the additional classes **tc-tiddlylink-missing tc-tiddlylink-shadow** like below:

```
<a class="tc-tiddlylink" href="##Introduction:Anchor-Links-using-HTML">Back to the top</a> or just back to <a class="tc-tiddlylink tc-tiddlylink-missing tc-tiddlylink-shadow" href=<<qualify ##Hints>>>Hints</a>.
```

Back to the top or just back to ***Style Hints***.

If you want to include an **`id`** attribute to a link in your sidebar Table of Contents, you can include it in your listed tiddler's **`caption`** field similar to the way shown below:

```
<span id=#some-value>Your caption</span>
```

Back to the top

And Filter Run Prefix

11th July 2023 at 10:47am

Named Filter Run Prefix

input	the filter output of all previous runs so far
output	output titles replace the output of previous filter runs

Introduced in v5.1.23

TypeError: setting getter-only property "parentNode"

This prefix has an optional shortcut syntax symbol `+run`

append Operator

21st March 2023 at 2:16pm

Filter Operators Listops Operators Order Operators

purpose	append a range of items from an array to the list
input	a list of items
suffix	an integer N, defaulting to all
parameter	**`list`** = the array of items to be appended to the tail of the list
output	a list with items appended from the head of the parameter array
`!` **output**	a list with items appended from the tail of the parameter array

Learn more about how to use Filters

Examples

append Operator (Examples)

8th November 2015 at 6:15am

append Operator Operator Examples

These examples make use of the Days of the Week tiddler.

```
[list[Days of the Week]append[Tomorrow]]
```

Try it

```
[list[Days of the Week]append[Yesterday Today Tomorrow]]
```

Try it

Append the first 4 short days of the week to our list

```
[list[Days of the Week]append:4{Days of the Week!!short}]
```

Try it

Apple

27th February 2022 at 10:08pm

Demonstrations

This example tiddler is used to illustrate some of the Filter Operators.

It has a **`list`** field containing the following items:

- *Apple 6*

- *Apple 7*
- *Apple 8*
- *Apple 9*
- *Apple 10*
- *Apple 20*
- *Apple 30*
- *Apple 100*

applypatches Operator

4th March 2023 at 4:48pm

Filter Operators String Operators

purpose	applies a set of patches to transform the input
input	a selection of titles
parameter	**P** = a string containing patches from the makepatches Operator
output	the transformed input to which the patches **P** have been applied

Learn more about how to use Filters

ⓘ Introduced in v5.2.6

Examples

Articles

2nd June 2016 at 7:27pm

Community

Here are some recent articles written about TiddlyWiki. Submit new articles via GitHub, Twitter or by posting in the TiddlyWiki Groups.

"TiddlyWiki Posts" by Jeffrey Kishner

Posted 7 months ago

A collection of articles covering integration with Fargo, Font Awesome and Google Calendar, and tips for managing task lists. The original site is missing, but a link to an archive is provided.

https://web.archive.org/web/20221015011644/http://blog.jeffreykishner.com/tiddlywiki/

> I have become a regular user of TiddlyWiki and have become so accustomed to using Font Awesome icons in Fargo that I wanted to incorporate the icons into my wikis as well.

"A free, open source wiki revisited" by Mark Gibbs, NetworkWorld

Posted 8 years ago

Interesting article giving the perspective of someone who has been away from TiddlyWiki for a few years:

http://www.networkworld.com/article/3028098/open-source-tools/tiddlywiki-a-free-open-source-wiki-revisited.html

> Way back in the mists of time (actually, January 2009) I wrote about a really cool tool called TiddlyWiki, a “non-linear personal web notebook”. Fast forward to today and I just had an out of body experience: Completely by accident I found a TiddlyWiki that I started when I wrote that piece and it still works!

> Finding code that works flawlessly after just two or three years is magical enough but after seven years?! And given that TiddlyWiki is written as a single page Web application and considering how different browsers are now than they were in 2009, the fact that the old version of TiddlyWiki still works is not short of miraculous.

"Setting Up a Personal TiddlyWiki Server on OS X" by Kris Johnson

Posted 8 years ago

Detailed description of setting up TiddlyWiki on Node.js on OS X.

http://undefinedvalue.com/2015/04/02/setting-personal-tiddlywiki-server-os-x

> For a new job, I decided to set up a personal wiki to keep notes. I wanted to keep it simple, meeting these requirements:
>
> - All the data is in a Dropbox folder (so it can be automatically synced between machines)
> - It must support Markdown syntax
>
> After looking at the options, I settled on TiddlyWiki. I've used "classic TiddlyWiki" before, and liked its simplicity, but I was always a little annoyed with the weird steps you have to go through to save changes. The new version of TiddlyWiki includes support for running it as a real HTTP server, so you can use it just like an online wiki.
>
> But it took me a couple of hours to figure out how to set that up. The TiddlyWiki documentation is not clear ("not clear" is a euphemistic way of saying "terrible"). So, I've written up these instructions in the hope it will spare somebody else all the frustration I had.

"TiddlyWiki" by Sander de Boer

Posted 9 years ago

Dutch blog post about TiddlyWiki

https://breinbout.wordpress.com/2014/11/26/tiddlywiki/

> I'm allergic to websites that are highly dependent on JavaScript. I think that JavaScript has many useful applications to add functionality to a site, but content needs to be visible if JavaScript is switched off. I use the *NoScript* extension for Firefox and surf with pleasure over the Internet without spontaneous things happen that I do not give permission. I love JavaScript on a short leash. More crazier maybe I'm so excited about TiddlyWiki.

(Translation from Dutch by Google Translate)

"Notizen mit TiddlyWiki systemübergreifend nutzen" by Michael Sonntag

Posted 9 years ago

A quick guide to using TiddlyWiki (written in German)

http://michaelsonntag.net/notizen-mit-tiddlywiki-systemuebergreifend-nutzen

> Those using multiple different computer platforms (if only PC and Android) knows for certain: You would like to sync notes across computers and thereby be independent of any apps or services. I am concerned at any rate so and so I am constantly looking for the perfect solution.
>
> With TiddlyWiki I found it mostly. As the name suggests, it TiddlyWiki is a Wikisystem. In contrast to all other wikis, TiddlyWiki is a single HTML file that runs in the browser and thus can be used on all modern operating systems. And because there is only one file, it can be very easily via the well-known cloud services (Dropbox, Google Drive, ownCloud) or synchronized with FTP or used on a USB stick.

(Translation from German by Google Translate)

"TiddlyWiki 5 im Betatest" by besim

Posted 10 years ago

Reaction to TiddlyWiki5 from the translator of the German edition of TiddlyWikiClassic.

http://blog.netplanet.org/2014/01/03/tiddlywiki-5-im-betatest/

> There is little software that can still inspire me so even after years, as on the first day. This includes TiddlyWiki, the "pocket-wiki". The latter is because the Wiki completely fits into an HTML file, and this HTML file brings everything - JavaScript program logic, CSS appearance and the entire Wiki content as stored records. So a TiddlyWiki file is then sometimes happy times some megabytes in size, but just has the unbeatable advantage that it works just as fast file created locally in a variety of browsers.

(Translation from German by Google Translate)

asin Operator

22nd October 2021 at 12:28am

Filter Operators Mathematics Operators Trigonometric Operators Unary Mathematics Operators

purpose	calculate the arcsine value (in radians) of a list of numbers
input	a selection of titles
output	the arcsine (in radians) of the input numbers

Learn more about how to use Filters

Introduced in v5.1.21 See Mathematics Operators for an overview.



This is the inverse operation of sin: sin calculate the sine of an angle (in radian), but asin calculate the angle (in radian) of a sine.

Examples

asin Operator (Examples)

20th October 2021 at 4:27pm

asin Operator Operator Examples

```
[[2]asin[]]
```

Try it

```
=1 =2 =3 =4 +[asin[]]
```

Try it

atan Operator

22nd October 2021 at 12:28am

Filter Operators Mathematics Operators Trigonometric Operators Unary Mathematics Operators

purpose	calculate the arctangent value (in radians) of a list of numbers
input	a selection of titles
output	the arctangent (in radians) of the input numbers

Learn more about how to use Filters

ⓘ **Introduced in v5.1.21** See Mathematics Operators for an overview.

Examples

atan Operator (Examples)

20th October 2021 at 4:13pm

atan2 Operator Operator Examples

```
[[2]atan[]]
```

Try it

```
=1 =2 =3 =4 +[atan[]]
```

Try it

atan2 Operator

22nd October 2021 at 12:28am

Binary Mathematics Operators Filter Operators Mathematics Operators Trigonometric Operators

purpose	returns the angle in the plane (in radians) between the positive x-axis and the ray from (0,0) to the point (x,y), for [**Y**]atan2[**X**]
input	a selection of titles = coordinate Y (**Y**)
parameter	**X** = coordinate X

output	the angle in radians (in [- π , π]) between the positive x-axis and the ray from (0,0) to the point (x,y)

Learn more about how to use Filters

Introduced in v5.1.21 See Mathematics Operators for an overview.

Examples

atan2 Operator (Examples)

20th October 2021 at 4:08pm

atan2 Operator Operator Examples

```
[[2]atan2[5]]
```

Try it

```
=1 =2 =3 =4 +[atan2[4]]
```

Try it

Attribute Selectors

11th April 2018 at 7:39pm

Custom data-styles

[attr]
Represents an element with an attribute name of attr.

[attr="value"]
Represents an element with an attribute name of attr and whose value is exactly "value".

[attr~="value"]
Represents an element with an attribute name of attr whose value is a whitespace-separated list of words, one of which is exactly "value".

[attr|="value"]
Represents an element with an attribute name of attr. Its value can be exactly “value” or can begin with “value” immediately followed by “-” (U+002D). It can be used for language subcode matches.

[attr^="value"]
Represents an element with an attribute name of attr and whose first value is prefixed by "value".

[attr$="value"]
Represents an element with an attribute name of attr and whose last value is suffixed by "value".

[attr*="value"]
Represents an element with an attribute name of attr and whose value contains at least one occurrence of string "value" as substring.

[attr "operator value" i]
Adding an i (or I) before the closing bracket causes the value to be compared case-insensitively (for characters within the ASCII range).

Learn more at: Attribute selectors - CSS or CSS-Specification

Audio

19th October 2014 at 10:06pm

Features Working with TiddlyWiki

Audio files can be incorporated into TiddlyWiki in a very similar way to images.

Embedded Audio

Small audio files can be embedded directly within TiddlyWiki. Embedding isn't suitable for large files (over a few hundred kilobytes) because it increases the size of the TiddlyWiki file.

For example, the tiddler TiddlyWiki.mp3 contains an MP3 recording of the word "TiddlyWiki". If you visit that tiddler, you should see an audio player that will play back the recording.

You can also transclude audio files. For example:

copy to clipboard

```
{{TiddlyWiki.mp3}}
```

That renders as:

External Audio

External audio tiddlers use the **_canonical_uri** field to point to an external audio file/stream, and have their **text** field blocked. This reduces their size considerably, but still allows for playback.

For example, the tiddler Caruso - Ave Maria points to an online audio recording hosted on http://archive.org:

copy to clipboard

```
{{Caruso - Ave Maria}}
```

That renders as:

AutoSave

10th June 2016 at 10:15am

Features

If there is a *SaverModule* available that supports it, TiddlyWiki will automatically trigger a save of the current document on clicking ✓ **ok** or 🗑 **delete** when editing a tiddler.

You should see a yellow notification at the top right of the window to confirm that an automatic save has taken place.

Automatic saving can be enabled or disabled through the **Settings** tab of the **control panel** ⚙. Behind the scenes, it is controlled through the configuration tiddler **$:/config/AutoSave**, which must have the value **yes** to enable automatic saving.

average Operator

26th April 2021 at 3:15pm

Filter Operators Mathematics Operators Reducing Mathematics Operators

purpose	treating each input title as a number, compute their arithmetic mean
input	a selection of titles
output	the arithmetic mean of the input as numbers

Learn more about how to use Filters

Introduced in v5.2.0 See Mathematics Operators for an overview.

Examples

average Operator (Examples)

26th April 2021 at 3:15pm

average Operator Operator Examples

```
=1 =3 =4 =5 +[average[]]
```

Try it

Note that if there is no input the operator returns `NaN`

```
[tag[NotATiddler]get[price]] +[average[]]
```

Try it

backlinks Operator

3rd February 2015 at 7:33pm

Filter Operators

purpose	find the titles that link to each input title
input	a selection of titles
parameter	none
output	any non-system titles that contain hard links to the input titles

Learn more about how to use Filters

Each input title is processed in turn. The corresponding tiddler's list of backlinks is generated, sorted alphabetically by title, and then dominantly appended to the operator's overall output.

Examples

backlinks Operator (Examples)

18th January 2015 at 7:31pm

backlinks Operator Operator Examples

```
[[HelloThere]backlinks[]]
```

Try it

```
[tag[TableOfContents]backlinks[]]
```

→ tiddlers that link to tiddlers tagged TableOfContents

Try it

```
[all[current]backlinks[]]
```

→ tiddlers that link to this one

Try it

Base64

21st February 2015 at 7:17pm

Definitions

Base64 is a way of representing binary data, such an image, as a string of text.

before Operator

3rd February 2015 at 8:19pm

Filter Operators Order Operators

purpose	find which input title precedes a specified one
input	a selection of titles
parameter	**T** = one of those titles
output	the title that immediately precedes **T** in the input

Learn more about how to use Filters

If **T** is not present in the input, or is the first title there, then the output is empty.

Examples

before Operator (Examples)

18th January 2015 at 7:31pm

before Operator Operator Examples

These examples make use of the Days of the Week tiddler. The Thursday tiddler shows a further example.

```
[list[Days of the Week]before[Saturday]]
```

Try it

```
[list[Days of the Week]before[Monday]]
```

Try it

Behaviour of invoked variables depends on how the variable was declared

26th July 2023 at 4:57pm

Variable Usage

Invoked in normal wikitext context: `<$transclude $variable=macro/>` or `<<macro>>`

how declared	behaviour
\define	All wikitext and variable substitution and textual substitution takes place
$set, **$let**,	All wikitext and variable substitution takes place

$vars, \procedure, \widget	
\function	Invoking a function in this way (`<<macro>>`) is a synonym for `<$text text={{{[function[macro]]}}}/>`. As with any filtered transclusion (i.e. triple curly braces), all results except the first are discarded

Invoked via widget attribute: `<div class=<<macro>>/>`

how declared	behaviour
\define	Textual substitution of parameters is performed on the body text. No further processing takes place. The result after textual substitution is used as the attribute's value
$set, **$let**, **$vars**, \procedure, \widget	Body text is retrieved as-is and used as the attribute's value.
\function	When a function is invoked as `<div class=<<macro>>/>`, it is a synonym for `<div class={{{[function[macro]]}}}/>`. As with any filtered transclusion (i.e. triple curly braces), all results except the first are discarded. That first result is used as the attribute's value. Note that functions are recursively processed even when invoked in this form. In other words a filter expression in a function can invoke another function and the processing will continue

Invoked via filter operator parameter: `[<macro>]`

how	behaviour

declared	
\define	Textual substitution of parameters is performed on the body text. No further processing takes place. The result after textual substitution is used as the filter operator's parameter.
$set, **$let**, **$vars**, \procedure, \widget	Body text is retrieved as-is and used as the filter operator's parameter.
\function	The body text of the function is treated as a filter expression and evaluated. The first result is passed to the operator as a parameter. The remaining results are discarded.

Invoked via function call in a filter expression: `[function[macro]]`

how declared	behaviour
\define, **$set**, **$let**, **$vars**, \procedure, \widget	Every function is a variable, but only variables defined using \function are invokable using the **function** filter operator. Attempts to use a non-function variable is the same as if the function doesn't exist. The behavior in this case is like the identity function. All filter input is passed unchanged to the output.
\function	The body text of the function is treated as a filter expression and evaluated. This filter expression can itself contain a function call. Filter expressions can be factored out into functions arbitrarily deep.

Examples

Below is an example macro, procedure and function definition. All three forms of parameter substitution `$a1$`, `<<__a1__>>`, and `<<a1>>` are included in each definition. The output helps illustrate when each form of substitution will or will not have affect.

```
\define m1(a1) $a1$ - <<__a1__>> - <<a1>>
\procedure p1(a1) $a1$ - <<__a1__>> - <<a1>>
\function f1(a1) $a1$ "-" [<__a1__>] "-" [<a1>] :and[join[ ]]
```

Variable transclusion	output
`<<m1 foo>>`	foo - foo -
`<<p1 foo>>`	$a1$ - - foo
`<<f1 foo>>`	$a1$ - - foo
Widget attribute	**output**
`<$text text=<<m1 foo>>/>`	foo - <<__a1__>> - <<a1>>
`<$text text=<<p1 foo>>/>`	$a1$ - <<__a1__>> - <<a1>>
`<$text text=<<f1 foo>>/>`	$a1$ - - foo
Filter operator parameter	**output**
`[<m1 foo>]`	foo - <<__a1__>> - <<a1>>
`[<p1 foo>]`	$a1$ - <<__a1__>> - <<a1>>
`[<f1 foo>]`	$a1$ - - foo
Function call in filter expression	**output**
`[function[m1],[foo]]`	"A free, open source wiki revisited" by Mark Gibbs, NetworkWorld
`[function[p1],[foo]]`	"A free, open source wiki revisited" by Mark Gibbs, NetworkWorld
`[function[f1],[foo]]`	$a1$ - - foo

Behaviour of variables invoked via filter expression function call

26th July 2023 at 5:06pm

Variable Usage

how declared	behaviour
\define, `$set`, `$let`, `$vars`, \procedure, \widget	Every function is a variable, but only variables defined using \function are invokable using the `function` filter operator. Attempts to use a non-function variable is the same as if the function doesn't exist. The behavior in this case is like the identity function. All filter input is passed unchanged to the output.
\function	The body text of the function is treated as a filter expression and evaluated. This filter expression can itself contain a function call. Filter expressions can be factored out into functions arbitrarily deep.

Behaviour of variables invoked via filter operator parameter

26th July 2023 at 5:06pm

Variable Usage

how declared	behaviour
\define	Textual substitution of parameters is performed on the body text. No further processing takes place. The result after textual substitution is used as the filter operator's parameter.

$set, **$let**, **$vars**, \procedure, \widget	Body text is retrieved as-is and used as the filter operator's parameter.
\function	The body text of the function is treated as a filter expression and evaluated. The first result is passed to the operator as a parameter. The remaining results are discarded.

Behaviour of variables invoked via normal wikitext

26th July 2023 at 5:06pm

Variable Usage

how declared	behaviour
\define	All wikitext and variable substitution and textual substitution takes place
$set, **$let**, **$vars**, \procedure, \widget	All wikitext and variable substitution takes place
\function	Invoking a function in this way (`<<macro>>`) is a synonym for `<$text text={{{[function[macro]]}}}/>`. As with any filtered transclusion (i.e. triple curly braces), all results except the first are discarded

Behaviour of variables invoked via widget attributes

26th July 2023 at 5:06pm

Variable Usage

how declared	behaviour
\define	Textual substitution of parameters is performed on the body text. No further processing takes place. The result after textual substitution is used as the attribute's value
$set, **$let**, **$vars**, \procedure, \widget	Body text is retrieved as-is and used as the attribute's value.
\function	When a function is invoked as `<div class=<<macro>>/>`, it is a synonym for `<div class={{{[function[macro]]}}}/>`. As with any filtered transclusion (i.e. triple curly braces), all results except the first are discarded. That first result is used as the attribute's value. Note that functions are recursively processed even when invoked in this form. In other words a filter expression in a function can invoke another function and the processing will continue

BetaReleases

20th December 2023 at 12:30pm

BetaReleaseNotes Releases

Here are the details of the beta releases of TiddlyWiki5. See TiddlyWiki5 Versioning for details of how releases are named.

5.0.18-beta
5.0.17-beta
5.0.16-beta
5.0.15-beta
5.0.14-beta
5.0.13-beta
5.0.12-beta
5.0.11-beta
5.0.10-beta
5.0.9-beta
5.0.8-beta
5.0.7-beta
5.0.6-beta
5.0.5-beta
5.0.4-beta
5.0.3-beta
5.0.2-beta

Release 5.0.18-beta

Released 17th September 2014 at 23:10

See GitHub for detailed change history of this release

This is a minor release prior to the full release of TiddlyWiki on September 20th. The documentation has been cleaned up and improved (with more improvements to come).

File Layout of tiddlywiki.com

The layout of files on tiddlywiki.com has been adjusted to make it more logical. See the ticket for a discussion. You can see the source files that make up tiddlywiki.com at https://github.com/Jermolene/jermolene.github.com

Hackability Improvements

- Added first iteration of a **`list-links`** macro (further improvements are planned)
- Added support for importing `.markdown` and `.md` files
- Extended **`timeline`** macro to support a subfilter

Bug Fixes

- Fixed problem with digits being classified as lower case letters for wiki link matching

- Fixed crash when sorting missing tiddlers by fields other than title
- Fixed problem with handling `.jpeg` file extensions
- Fixed problem with RadioWidget and missing tiddlers
- Fixed problem with dragging a partially selected link

Contributors

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:

- @BramChen
- @buggyj
- @Eucaly
- @pmario
- @xcazin

bf Operator

3rd February 2015 at 7:37pm

Filter Operators Order Operators

purpose	same as **`rest`**

Learn more about how to use Filters

Block Mode WikiText

22nd January 2022 at 7:28pm

WikiText Parser Modes

Some WikiText is recognised only while the parser is in block mode.

Common characteristics of such block mode WikiText:

- At least one entire line is required to delimit the WikiText.
- The closing punctuation must come at the end of the line (in some cases the end of the line *is* the closing punctuation.

WikiText	Punctuation
Block Quotes in WikiText	Multi-line block quotes are enclosed by lines containing only the text `<<<`; single line block quotes are also possible.
Code Blocks in WikiText	Enclosed by lines containing only the text ```` ``` ````
Definitions in WikiText	Each term is on its own line and each definition is on its own line.
Hard Linebreaks in WikiText	Enclosed by lines containing only the text `"""`.
Headings in WikiText	Entire line starting with `!`.
Horizontal Rules in WikiText	A line containing only the text `---`.

Lists in WikiText	Each list item is on its own line.
Paragraphs in WikiText	Any text other than the start punctuation of one of the other block mode WikiText will start a paragraph. Even the start punctuation of inline mode WikiText will start a paragraph. The parser includes all following lines into the paragraph until it encounters a blank line.
Styles and Classes in WikiText	Enclosed by lines starting with `@@`.
Tables in WikiText	Each table row is a line starting and ending with `|`.
Typed Blocks in WikiText	Enclosed by lines starting with `$$`.

The above WikiText types are only recognised in **block mode**. However, the text *ENCLOSED* by most of them will be parsed in **inline mode** (Block Quotes in WikiText and Styles and Classes in WikiText are the two exceptions in which the parser will continue in **block mode**). While in **inline mode** the parser may encounter something which moves it to **block mode** (see WikiText parser mode transitions).

At the end of the terminating line, the parser will return to **block mode**.



Note: Hard Linebreaks in WikiText require an extra blank line after the trailing `"""` before the parser will return to **block mode**.

If the punctuation for the above types of WikiText is encountered while the parser is in **inline mode**, it will be *ignored* and output as-is.

Examples

Block Mode WikiText (Examples)

15th January 2022 at 5:10pm

Block Mode WikiText

Paragraphs are the most common WikiText. It is important to know they do not end until a blank line is encountered. Once a paragraph starts the parser will be in **inline mode**. Until that blank line is encountered other **block mode** syntax will be ignored:

copy to clipboard

```
This is a paragraph.
Only __//inline mode//__ punctuation is recognised here.
Block mode punctuation will be <b>ignored</b> until
the paragraph ends (i.e. a blank line is encountered).
For example:
* List item punctuation is ignored
* So is this horizontal rule:
---
|tables|are|ignored|
! headings are ignored
<<<
multi-line block quotes are ignored
<<<
```

That renders as:

This is a paragraph. Only *inline mode* punctuation is recognised here. Block mode punctuation will be **ignored** until the paragraph ends (i.e. a blank line is encountered). For example: * List item punctuation is ignored * So is this horizontal rule: — |tables|are| ignored| ! headings are ignored <<< multi-line block quotes are ignored <<<

To get the parser to recognise the above **block mode** WikiText, a blank line needs to be inserted after the "For example:" line. The blank line will be treated as the end of the paragraph and the parser will start recognising **block mode** punctuation again. Like this:

```
This is a paragraph.
Only __//inline mode//__ punctuation is recognised here.
Block mode punctuation will be <b>ignored</b> until
the paragraph ends (i.e. a blank line is encountered).
For example, here is a blank line:

* due to above blank line,
* list items work
* the following horizontal rule also works:
---
|!tables|!work|
|cell 1| cell 2|
! Headings work
<<<
multi-line block quotes work
<<<
; Term
: Definition of that term
Another paragraph can start here, but it will not end until blank line
```

That renders as:

This is a paragraph. Only *inline mode* punctuation is recognised here. Block mode punctuation will be **ignored** until the paragraph ends (i.e. a blank line is encountered). For example, here is a blank line:

> - due to above blank line,
> - list items work
> - the following horizontal rule also works:
>
> ---
>
> | tables | work |
> |---|---|
> | cell 1 | cell 2 |
>
> # Headings work
>
> > multi-line block quotes work
>
> **Term**
> : Definition of that term
>
> Another paragraph can start here, but it will not end until blank line

All the block mode examples above except for the paragraph are written one line after the other without blanks between. Those types of WikiText don't require the extra blank line to terminate.

Block Quotes in WikiText

17th April 2017 at 6:51pm

WikiText

There are two ways to produce HTML block quotes in TiddlyWiki5, one for content spread across multiple lines, and one for single line content.

Multi-line Block Quotes

The syntax for multi-line block quotes (❝❞) is:

copy to clipboard

```
<<<
This is a block quoted paragraph
written in English
<<<
```

That renders as:

> This is a block quoted paragraph written in English

... and the underlying HTML is:

```
<blockquote class="tc-quote"><p>This is a block quoted paragraph
written in English
</p></blockquote>
```

Citation

A citation can be added to the quote like this:

copy to clipboard

```
<<<
Computers are like a bicycle for our minds
<<< Steve Jobs
```

That renders as:

> Computers are like a bicycle for our minds
>
> *Steve Jobs*

... and the underlying HTML is:

```
<blockquote class="tc-quote"><p>Computers are like a bicycle for our minds
</p><cite>Steve Jobs</cite></blockquote>
```

CSS Classes

CSS classes can be added to a block quote:

copy to clipboard

```
<<<.myClass.another-class
Operating systems are like a brick wall for our minds
```

```
<<< Nobody
```

That renders as:

> Operating systems are like a brick wall for our minds
>
> *Nobody*

... and the underlying HTML is:

```
<blockquote class="tc-quote myClass another-class"><p>Operating systems are like a brick wall for our
minds
</p><cite>Nobody</cite></blockquote>
```

The core includes the class `tc-big-quote` that renders block quotes with outsize double quotes:

copy to clipboard

```
<<<.tc-big-quote
A dramatic quote
<<< Somebody Important
```

That renders as:

> A dramatic quote

— Somebody Important

... and the underlying HTML is:

```
<blockquote class="tc-quote tc-big-quote"><p>A dramatic quote
</p><cite>Somebody Important</cite></blockquote>
```

Single-line Block Quotes

The single-line syntax for block quotes is actually an extension of the syntax for Lists in WikiText. For example:

copy to clipboard

```
> Quoted text
> Another line of quoted text
```

That renders as:

> Quoted text
>
> Another line of quoted text

... and the underlying HTML is:

```
<blockquote><div>Quoted text</div><div>Another line of quoted text</div></blockquote>
```

You can also nest quotes like this:

```
> A top quote
>> A subquote
> Another top quote
```

Which renders as:

> A top quote
>
> > A subquote
>
> Another top quote

You can also mix block quotes with other list items. For example:

copy to clipboard

```
* List One
** List Two
**> A quote
**> Another quote
* List Three
```

That renders as:

- List One
  - List Two

    > A quote
    >
    > Another quote

- List Three

... and the underlying HTML is:

```
<ul><li>List One<ul><li>List Two<blockquote><div>A quote</div><div>Another quote</div></blockquote></
li></ul></li><li>List Three</li></ul>
```

Blog Edition

2nd September 2015 at 2:32pm

Editions

The **Blog** edition of TiddlyWiki contains tools to help publish static HTML blogs written in TiddlyWiki. It is currently designed for use under Node.js.

While documentation is being prepared, see https://github.com/Jermolene-blog/blog for an example of use.

Blurry Lawn.jpg

external-image picture



Brackets

28th July 2022 at 5:04pm

Definitions

WikiText syntax uses a number of different types of brackets. Their names are shown in the table below:

Appearance	Name	Short name	Usage
( )	Round brackets	Parenthesis	Not used in WikiText
[ ]	Square brackets	Brackets	Links, Filters
{}	Curly brackets	Braces	Text references, Filtered attributes
<>	Angle brackets	Chevrons	HTML elements and widgets, Macros

Brownies

6th October 2020 at 8:12pm

shopping

This is a sample shopping list item for the Shopping List Example

BrowserCompatibility

Working with TiddlyWiki

TiddlyWiki is designed to work with HTML5-compatible browsers.

The following table summarises the browser versions that are known to work with TiddlyWiki

Browser	Status
Internet Explorer	Version 10 and above
Chrome	All recent versions
Firefox	All recent versions
Firefox for Android	All recent versions
Safari	Version 6 and above

BrowserStorage Plugin

6th February 2019 at 7:12pm

OfficialPlugins

The BrowserStorage Plugin enables TiddlyWiki to save tiddlers in browser local storage. This means that changes are stored within the browser, and automatically re-applied any time the base wiki is reloaded.

Browser local storage is not a panacea for TiddlyWiki:

- Browsers limit the amount of local storage available to a page, typically to 5 or 10MB
- Keeping personal data in browser local storage can lead to unexpected privacy violations

- Browsers reserve the right to without warning delete data stored in local storage at any time
- Browsers tie local storage to a URL which can lead to problems if you move a wiki to a URL previously occupied by a different wiki

Please use this plugin with caution. There are a number of unresolved issues and open questions.

The BrowserStorage Plugin can be installed from the plugin library.

BrowseWidget

13th November 2023 at 10:33am

Widgets

Introduction

The browse widget displays an HTML file browser button that allows the user to choose one or more files to import. It sends a WidgetMessage: tm-import-tiddlers carrying a JSON representation of the tiddlers imported from the files up through its parents. This message is usually handled by the NavigatorWidget which adds the tiddlers to the store and updates the story to display them.

Content and Attributes

The content of the **$**`BrowseWidget` widget is ignored.

Attribute	Description
multiple	Set to "multiple" to select multiple file upload

deserializer	ⓘ **Introduced in v5.1.15** Optional name of deserializer to be used (by default the deserializer is derived from the file extension or type)
tooltip	Optional tooltip text
accept	ⓘ **Introduced in v5.1.23** Optional comma delimited list of file accepted extensions and/or MIME types
message	Optional override of widget message to be generated. The files will be passed in the JavaScript object `event.target.files`
param	Optional parameter to be passed with the custom message
data-*	ⓘ **New in v5.3.2** Optional data attributes to be assigned to the HTML element
style.*	ⓘ **New in v5.3.2** Optional CSS properties to be assigned to the HTML element

On iPhone/iPad choosing the multiple option will remove the ability to take photographs/videos directly into TiddlyWiki.

e.g.

```
<$browse/>
```

renders as:

Parcourir… Aucun fichier sélectionné.

BT

17th November 2021 at 8:55pm

Definitions

BT (née British Telecom) is the UK's largest telecommunications company. In 2007, Osmosoft was acquired by BT. JeremyRuston subsequently left BT in 2011.

BuildCommand

27th April 2014 at 11:04pm

Commands

Build the specified build targets for the current wiki. If no build targets are specified then all available targets will be built.

```
--build <target> [<target> ...]
```

Build targets are defined in the `tiddlywiki.info` file of a wiki folder.

See tiddlywiki.info Files for details of defining build targets.

Building TiddlyWikiClassic

12th September 2014 at 4:16pm

TiddlyWiki on Node.js

TiddlyWiki5 can be used to build older 2.x.x versions of TiddlyWikiClassic from their constituent components. Doing so involves these features:

- The `tiddlywiki/classictools` plugin, containing a deserializer module which allows tiddlers to be loaded from TiddlyWiki 2.x.x `.recipe` files
- The `stripcomments` format for the ViewWidget, which strips single line JavaScript comments starting `//#`
- The `stripTitlePrefix='yes'` attribute of the FieldsWidget, which removes prefixes wrapped in curly braces from the `title` attribute
    - For example, `{tiddler}HelloThere` would be transformed to `HelloThere`

Usage

TiddlyWikiClassic is built from the command line by running TiddlyWiki on Node.js. A typical usage would be:

```
node ../../tiddlywiki.js \
    --verbose \
    --load <path_to_recipe_file> \
    --rendertiddler $:/core/templates/tiddlywiki2.template.html <path_to_write_index_file> text/plain \
    || exit 1
```

butfirst Operator

3rd February 2015 at 7:37pm

Filter Operators Order Operators

purpose	same as **`rest`**

Learn more about how to use Filters

butlast Operator

26th February 2022 at 5:33am

Filter Operators Order Operators

purpose	discard the last **`N`** input titles

input	a selection of titles
parameter	`N` = an integer, defaulting to 1
output	all but the last `N` input titles

Learn more about how to use Filters

Introduced in v5.2.2 The `butlast` operator returns the input list unchanged if `N` is 0. This is consistent with the behaviour of the rest Operator.

Examples

butlast Operator (Examples)

26th February 2022 at 5:33am

butlast Operator Operator Examples

These examples make use of the Days of the Week tiddler.

```
[list[Days of the Week]butlast[]]
```

Try it

```
[list[Days of the Week]butlast[0]]
```

Try it

```
[list[Days of the Week]butlast[2]]
```

Try it

```
A B C D E F G H I J K L M +[butlast[7]]
```

Try it

ButtonWidget

13th November 2023 at 10:33am

TriggeringWidgets Widgets

Introduction

The button widget displays an HTML `<button>` element that can perform a combination of optional actions when clicked:

- Executing any ActionWidgets passed in the `actions` attribute
- Executing any ActionWidgets that are immediate children of the button widget
- Execute any integrated actions:
  - Navigate to a specified tiddler
  - Dispatch a user defined widget message
  - Trigger a user defined popup
  - Assign new text to a specified tiddler

The integrated actions are provided as a shortcut for invoking common actions. The same functionality is available via ActionWidgets, with the exception of the support for highlighting selected popups.

Content and Attributes

The content of the `<$button>` widget is displayed within the button.

Attribute	Description
actions	A string containing ActionWidgets to be triggered when the key combination is detected. ⓘ **Introduced in v5.1.23** the **modifier** variable lists the modifier keys that are pressed when the action is invoked. The possible modifiers are **ctrl**, **ctrl-alt**, **ctrl-shift**, **alt**, **alt-shift**, **shift** and **ctrl-alt-shift**
to	The title of the tiddler to navigate to
message	The name of the widget message to send when the button is clicked
param	The optional parameter to the message
set	A TextReference to which a new value will be assigned

setTitle	A title to which a new value will be assigned, **without** TextReference. Gets preferred over **`set`**
setField	A **field name** to which the new value will be assigned, if the attribute **`setTitle`** is present. Defaults to the **text** field
setIndex	An **index** to which the new value will be assigned, if the attribute **`setTitle`** is present
setTo	The new value to assign to the TextReference identified in the `set` attribute or the text field / the field specified through **`setField`** / the index specified through **`setIndex`** of the title given through **`setTitle`**
selectedClass	An optional additional CSS class to be assigned if the popup is triggered or the tiddler specified in **`set`** already has the value specified in **`setTo`**
default	Default value if **`set`** tiddler is missing for testing against **`setTo`** to determine **`selectedClass`**
popup	Title of a state tiddler for a popup that is toggled when the button is clicked. See PopupMechanism for details
popupTitle	Title of a state tiddler for a popup that is toggled when the button is clicked. In difference to the **popup** attribute, **no** TextReference is used. See PopupMechanism for details
popupAbsCoords	ⓘ **Introduced in v5.2.4** If set to **yes** writes absolute coordinates to the tiddler referenced by the **popup**. If set to **no** (the default) uses relative coordinates. See Coordinate Systems for details
aria-label	Optional *Accessibility* label
tooltip	Optional tooltip
class	An optional CSS class name to be assigned to the HTML element
data-*	ⓘ **New in v5.3.2** Optional data attributes to be assigned to the HTML element
style.*	ⓘ **New in v5.3.2** Optional CSS properties to be assigned to the HTML element
style	An optional CSS style attribute to be assigned to the HTML element
tag	An optional html tag to use instead of the default "button"
dragTiddler	An optional tiddler title making the button draggable and identifying the payload tiddler. See DraggableWidget for details
dragFilter	An optional filter making the button draggable and identifying the list of payload tiddlers. See DraggableWidget for details
tabindex	Sets the `tabindex` attribute of the button to the given value
disabled	ⓘ **Introduced in v5.1.23** Optional, disables the button if set to "yes". Defaults to "no"

Note: In almost all other cases where a TextReference is used as a widget attribute, it will be placed between curly brackets, to transclude the value currently stored there. However, when we use a TextReference as the value of a button widget's **`set`** attribute, we are referencing *the storage location itself*, rather than the value stored there, so we do **not** use curly brackets there. *Example:* we could code a button widget that sets the `caption` field of *TiddlerA* to be the same as that of *TiddlerB* as:



ⓘ **Introduced in v5.1.18** **`setTitle`**, **`setField`** and **`setIndex`** attributes allow specifying Tiddler states **directly**, without interpreting them as TextReferences. This is useful for edge-cases where titles may contain characters that are used to denote Tiddler fields or indices (`!!` , `##`)

```
<$button set="TiddlerA!!caption" setTo={{TiddlerB!!caption}} >

Press me!

</$button>
```

Tip: Set **class** to `tc-btn-invisible tc-tiddlylink` to have a button look like an internal link.

CamelCase

28th August 2013 at 8:53pm

Definitions

CamelCase is formed by taking a phrase, capitalising the initial letter of each word, and smashing the words together to form a portmanteau word. Most wikis use CamelCase to signal phrases that should automatically become links.

Caruso - Ave Maria

Demonstrations



Cascade Filter Run Prefix

10th July 2023 at 9:33am

Named Filter Run Prefix

purpose	modify input titles by successively evaluating a list of filters and applying the first result
input	the filter output of all previous runs so far
output	the input titles as modified by the filters returned by this filter run

Introduced in v5.2.1

TypeError: setting getter-only property "parentNode"

The filter expression for this filter run is evaluated to return a list of filters. Each input title is then evaluated against each of the filters in turn, and the input title is replaced with the first result of the first filter that returns a non-empty list. If none of the filters return a result for an input title, it is replaced with an empty string.

The following variables are available within the filter run:

- **`currentTiddler`** - the input title

- **`..currentTiddler`** - the value of the variable **`currentTiddler`** outside the filter run.

Cascade Filter Run Prefix (Examples)

Cascade Filter Run Prefix (Examples)

5th March 2023 at 1:52pm

Cascade Filter Run Prefix

The **`:cascade`** filter run prefix is used by the TiddlyWiki core to choose the colour for each tiddler as used in the tag pill generated by the tag macro.

```
:cascade[all[shadows+tiddlers]tag[$:/tags/TiddlerColourFilter]!is[draft]get[text]]
```

The filter expression inside the above **`:cascade`** filter run returns a list of filters from the text fields of tiddlers tagged $:/tags/TiddlerColourFilter as shown below:

- `[has[color]then{!!color}]`
- ```
  [tag[TableOfContents]then[#1e90ff]]
  [tag[Working with TiddlyWiki]then[darkorchid]]
  ```
- `[[$:/config/DefaultTiddlerColour]has[text]get[text]trim[]]`

For each input title, each of these filters is now applied to in turn until the first filter that returns a non-empty list. The first result from this non-empty list replaces that input title and is used as the colour for that tiddler.
  ```

```
[all[shadows+tiddlers]] :cascade[all[shadows+tiddlers]tag[$:/tags/TiddlerColourFilter]!is[draft]get[text]] +[!is[blank]limit[10]]
```

Try it

Extended example

This example illustrates step-by-step how the **:cascade** filter run works.

A filter expression whose output returns a list of filters is required. Typically such a filter expression will construct the list of filters by querying some field in a set of tiddlers (likely via tag). To keep this example self-contained, 3 filters are stored in the *filter1*, *filter2* and *filter3* fields of this tiddler. Text references and instances of the append operator are used to combine the fields into a list of filters.

The filter `[{!!filter1}append{!!filter2}append{!!filter3}]`, generates this list of filters:

- `[prefix[ca]then[ca]]`
- `[suffix[at]then[at]]`
- `other`

For this example, `cat can bat bug` will be used as the input titles. Each input title will be evaluated by **:cascade** against each of the above filters in turn. To help understand what **:cascade** sees internally, the following table uses the Map Filter Run Prefix to show the result of each individual filter.

```
cat can bat bug
:map[prefix[ca]then[ca]]
```

Try it

```
cat can bat bug
:map[suffix[at]then[at]]
```

Try it

```
cat can bat bug
:map[[other]]
```

Try it

Click "Try it" on the above 3 examples. The first non-blank value from each row as read from left to right will be returned when **:cascade** is used. Click "Try it" below to see the final result:

```
cat can bat bug :cascade[{!!filter1}append{!!filter2}append{!!filter3}]
```

Try it

Cascades

6th December 2021 at 5:47pm

Concepts

Introduction

Cascades are a key mechanism used to construct and customise TiddlyWiki's user interface.



See Customising TiddlyWiki's user interface for an overview of all the customisation mechanisms

How cascades work

Cascades provide a means to select one of multiple values based on flexible, extensible criteria. They can be thought of as a list of conditions that are evaluated in turn until one of them matches.

Each cascade is defined by a special tag which identifies the tiddlers containing the conditions to be matched. The Order of Tagged Tiddlers determines the order in which the conditions are processed, and provides the means for arranging new rules at specific points in the cascade.

The conditions are defined by a Filter Expression in the text field. The filters are evaluated with the list of input titles set to the title of the tiddler to be considered. The `currentTiddler` variable is also set to the title of the tiddler to be considered.

If the filter returns no result then that result will be ignored and the cascade proceeds to test the next condition. If the filter expression does return at least one result then it serves as the result of the entire cascade.

The Cascade Filter Run Prefix provides the implementation of the underlying logic.

Example

The Story Tiddler Template Cascade contains the logic for determining a tiddler should be displayed as an editable draft or in view mode. By default, it consists of two tiddlers containing the following list of rules:

1. `[is[draft]then{$:/config/ui/EditTemplate}]` – if it is a draft tiddler, use the template title given in the tiddler **$:/config/ui/EditTemplate**
2. `[{$:/config/ui/ViewTemplate}]` – otherwise, use the template title given in the tiddler **$:/config/ui/ViewTemplate**

Usage in the core

The TiddlyWiki core uses cascades to choose the following elements:

Story Tiddler Template Cascade	The template used to display a particular tiddler in the story river. By default, the edit template is chosen for draft tiddlers, and the view template for others
Tiddler Icon Cascade	The optional icon associated with a particular tiddler (displayed alongside the title)
Tiddler Colour Cascade	The optional colour associated with a particular tiddler (used to colour the tiddler icon and if the tiddler is used as a tag also provides the colour for the tag pill)
View Template Title Cascade	The template used to display the title of a particular tiddler (used by the default view template to display the tiddler title)
View Template Body Cascade	The template used to display the view mode body of a particular tiddler (used by the default view template to display the tiddler body)
Edit Template Body Cascade	The template used to display the edit mode body of a particular tiddler (used by the default edit template to display the tiddler body editor)
Field Editor Cascade	The template used to display the edit mode of a tiddler field (used by the default edit template to display the field editor)

You can see the current settings for each cascade in **$:/ControlPanel** under the **Info** -> **Advanced** -> **Cascades** tab.

See Also

- Edit Template Body Cascade
- Field Editor Cascade
- How to create a custom story tiddler template

- How to create a custom tiddler colour rule
- How to create a custom tiddler icon rule
- Story Tiddler Template Cascade
- Tiddler Colour Cascade
- Tiddler Icon Cascade
- View Template Body Cascade
- View Template Title Cascade

Cascading Style Sheets

21st February 2015 at 7:17pm

Definitions

CSS is a standard plain-text format used for defining the presentational style of the various elements on a web page.

ceil Operator

11th June 2019 at 7:44pm

Filter Operators Mathematics Operators Unary Mathematics Operators

purpose	rounds a list of numbers up to the next largest integer
input	a selection of titles
output	rounds each of the input numbers up to the next largest integer

Learn more about how to use Filters

Introduced in v5.1.20 See Mathematics Operators for an overview.

Examples

ceil Operator (Examples)

11th June 2019 at 7:45pm

ceil Operator Operator Examples

```
[[1.6]ceil[]]
```

Try it

```
[[-1.6]ceil[]]
```

Try it

```
=-1.2 =-2.4 =3.6 =4.8 =5.1 +[ceil[]]
```

Try it

changecount Macro

21st December 2022 at 6:45pm

Core Macros Macros

The **changecount** macro returns the number of times the current tiddler has been created, stored or deleted during the current TiddlyWiki session.

If a tiddler is deleted and subsequently recreated, its **changecount** will be increased by two.

Parameters

(none)

Examples

changecount Macro (Examples)

21st February 2015 at 4:12pm

changecount Macro Macro Examples

copy to clipboard

```
<<changecount>>
```

Try it

The value will increase if you edit this tiddler and store it again, even without making any changes to its content.

To access the **changecount** of a different tiddler, use a **$tiddler** widget:

copy to clipboard

```
<$tiddler tiddler="Draft of 'New Tiddler'">
<<changecount>>
</$tiddler>
```

Try it

The value shown will increase whenever you create, store or delete *New Tiddler*.

Changes to filters in 5.0.9-beta

4th April 2014 at 12:35am

Release 5.0.9-beta

Introduction

This release resolves a number of inconsistencies with the way that filters are handled. The changes mean that existing filters may need to be updated - particularly those that must deal with missing or shadow tiddlers.

Changes to **is** and addition of **all**

Most filter operators act by choosing some or all of their source titles to pass through or re-order. Those that add new entries that are not drawn from the source list are referred to as selectors. Prior to 5.0.9-beta, a few filter operators were inconsistent in whether they filtered from the source list or selected new entries into it.

The specific changes are:

- The **is** operator now always strictly filters from the currently selected list of tiddlers
- The new **all** operator acts as a selector by replacing the current list with a combination of tiddlers from specific sources:
  - **current** for the current tiddler
  - **missing** for all missing tiddlers
  - **orphans** for all orphan tiddlers
  - **shadows** for all shadow tiddlers
  - **tiddlers** for all non-shadow tiddlers (including both system and non-system tiddlers)

The sources for the **all** operator can be combined with the `+` character. For example, `[all[shadows+tiddlers]]` returns all shadow tiddlers and all ordinary tiddlers.

Previously, it was common to have `[is[shadow]]` at the start of a filter string to select all the shadow tiddlers. In 5.0.9 and above, this will not return all the shadow tiddlers, but instead just those ordinary tiddlers that are also shadow tiddlers (by virtue of having overridden one). The resolution is to use the new **all** operator. For example, consider this filter from 5.0.8:

```
[is[shadow]!has[draft.of]tag[$:/tags/AdvancedSearch]] [!is[shadow]!has[draft.of]tag[$:/tags/AdvancedSearch]] +[tag[$:/tags/AdvancedSearch]]
```

In 5.0.9, that filter has been changed to:

```
[all[shadows+tiddlers]tag[$:/tags/AdvancedSearch]!has[draft.of]]
```

Note how the **all** operator allows operations to be performed on tiddlers from combinations of sources.

Changes to `[is[current]]`

One result of the changes is that `[is[current]]` now strictly filters from the source tiddlers; so, if the current tiddler is a missing tiddler not in the source list, then `[is[current]]` will return an empty list.

The solution is generally to use `[all[current]]` instead. It doesn't read as well, but has the required behaviour of returning just the current tiddler, regardless of whether it is in the source tiddlers.

Changes to **title** and **field**

There are minor changes to the way that the **title** and **field** operators work.

The **title** operator is a selector: it returns the specified title regardless of whether it is in the current source. **title** is used as the default operator if none is specified

The **field** operator is a filter: it only returns a subset of the source tiddlers. **field** is used as the default operator if the supplied operator is not

defined (the supplied operator is passed as the suffix to the field operator, so `[description[Missing]]` is equivalent to `[field:description[Missing]]`).

charcode Operator

22nd June 2021 at 11:44pm

Filter Operators

purpose	generates string characters from their numeric character codes
input	ignored
parameter	**c** = numeric character code
output	a string formed from concatenating the characters specified by the numeric codes given in the parameter(s)

Learn more about how to use Filters

ⓘ Introduced in v5.2.0

This operator returns a string formed from concatenating the characters specified by the numeric codes given in one or more parameters. It is useful for generating special characters such as tab (`charcode[9]`) or new line (`charcode[13],[10]`).

Examples

charcode Operator (Examples)

22nd June 2021 at 11:48pm

charcode Operator Operator Examples

```
[charcode[65]match[A]]
```

Try it

CheckboxWidget

13th November 2023 at 10:33am

TriggeringWidgets Widgets

Introduction

The checkbox widget displays an HTML `<input type="checkbox">` element that is dynamically bound to either:

- the presence or absence of a specified tag on a specified tiddler
- the value of a specified field of a specified tiddler

Content and Attributes

The content of the `<$checkbox>` widget is displayed within an HTML `<label>` element immediately after the checkbox itself. This means that clicking on the content will toggle the checkbox.

Attribute	Description

tiddler	Title of the tiddler to manipulate (defaults to the Current Tiddler)
tag	The name of the tag to which the checkbox is bound
invertTag	When set to **yes**, flips the tag binding logic so that the absence of the tag causes the checkbox to be checked
field	The name of the field to which the checkbox is bound
listField	ⓘ **Introduced in v5.2.3** The name of the field that contains the list to which the checkbox is bound
index	ⓘ **Introduced in v5.1.14** The property of the DataTiddler to which the checkbox is bound
listIndex	ⓘ **Introduced in v5.2.3** Like **index**, but treats the value as a list the same way that **listField** does
filter	ⓘ **Introduced in v5.2.3** A filter whose output determines the checked state of the checkbox
checked	The value of the field corresponding to the checkbox being checked
unchecked	The value of the field corresponding to the checkbox being unchecked
default	The default value to use if the field is not defined
indeterminate	Whether ambiguous values can produce indeterminate checkboxes (see below)
class	The class that will be assigned to the `<label>` element ⓘ **Introduced in v5.2.3** `tc-checkbox` is always applied by default, as well as `tc-checkbox-checked` when checked
actions	ⓘ **Introduced in v5.1.14** A string containing ActionWidgets to be triggered when the status of the checkbox changes (whether it is checked or unchecked)
uncheckactions	ⓘ **Introduced in v5.1.16** A string containing ActionWidgets to be triggered when the checkbox is unchecked
checkactions	ⓘ **Introduced in v5.1.20** A string containing ActionWidgets to be triggered when the checkbox is checked
disabled	ⓘ **Introduced in v5.1.23** Optionally disables the checkbox if set to **yes** (defaults to **no**)
data-*	ⓘ **New in v5.3.2** Optional data attributes to be assigned to the HTML `<input>` element
style.*	ⓘ **New in v5.3.2** Optional CSS properties to be assigned to the HTML `<input>` element

tag | **field** | **listField** | **index** | **listIndex** | **filter** | **indeterminate**

tag Mode

Using the checkbox widget in tag mode requires the **tag** attribute to specify the name of the tag. The checkbox will be checked if the tiddler specified in the **tiddler** attribute has the specified tag and unchecked if it does not.

This example creates a checkbox that flips the **done** tag on the current tiddler:

copy to clipboard

```
<$checkbox tag="done"> Is it done?</$checkbox>
```

That renders as:

☐ Is it done?

When the attribute **invertTag** is set to **yes**, the checkbox will be checked if the tiddler does *NOT* have the specified tag and unchecked if it does.

copy to clipboard

```
<$checkbox tag="done" invertTag="yes"> Is it not done?</$checkbox>
```

That renders as:

☑ Is it not done?

CheckboxWidget (field Mode)

17th March 2023 at 5:11pm

CheckboxWidget

`field` Mode

Using the checkbox widget in field mode requires the **`field`** attribute to specify the name of the field. The **`checked`** and **`unchecked`** attributes specify the values to be assigned to the field to correspond to its checked and unchecked states respectively. The **`default`** attribute is used as a fallback value if the field is missing or contains a value that does not correspond to the value of the **`checked`** or **`unchecked`** attributes.

This example creates a checkbox that is checked if the field **`status`** is equal to **`open`** and unchecked if the field is equal to **`closed`**. If the field value is undefined then it defaults to **`closed`**.

copy to clipboard

```
<$checkbox field="status" checked="open" unchecked="closed" default="closed"> Is it open?</$checkbox>

''status:'' {{!!status}}
```

That renders as:

☐ Is it open?

status:

CheckboxWidget (filter Mode)

25th March 2023 at 11:14am

CheckboxWidget

`filter` Mode

Using the checkbox widget in filter mode requires the **`filter`** attribute to contain a filter whose output will determine the checked state of the checkbox. In filter mode, checking the checkbox will not automatically make changes to any field of any tiddler. Instead, you can use the **`actions`** attribute (or the **`checkactions`** and **`uncheckactions`** attributes) to specify what should happen when the checkbox is toggled. It is your responsibility to make sure the actions cause changes to the tiddlers or fields that the filter results depend on, so that the checkbox becomes properly checked or unchecked after the actions have triggered and the filter has updated.

If the filter returns an empty result, the checkbox will be unchecked. Otherwise, if the filter result is non-empty, the checkbox will be checked.

However, if either the **`checked`** or **`unchecked`** attributes (or both) are specified, then their values will be looked for in the filter result, instead of considering any non-empty value to mean **`checked`**.

This example creates the same checkbox as in the listField mode example , selecting between **`red`** and **`green`** in the **`colors`** list field, but using filters and actions to make the change.

copy to clipboard

```
\define checkActions() <$action-listops $field="colors" $subfilter="-red green"/>
```

```
\define uncheckActions() <$action-listops $field="colors" $subfilter="red -green"/>
<$checkbox filter="[list[!!colors]]" checked="green" unchecked="red" default="red"
checkactions=<<checkActions>> uncheckactions=<<uncheckActions>> > Is "green" in colors?</$checkbox>

''colors:'' {{!!colors}}
```

That renders as:

☐ Is "green" in colors?

colors:

CheckboxWidget (indeterminate)

17th March 2023 at 5:09pm

CheckboxWidget

`indeterminate` Checkboxes

If both the **`checked`** and **`unchecked`** attributes are specified, but neither one is found in the specified field (or index), the result can be ambiguous. Should the checkbox be checked or unchecked? Normally in such cases the checkbox will be unchecked, but if the **`indeterminate`** attribute is set to **`yes`** (default is **`no`**), the checkbox will instead be in an "indeterminate" state. An indeterminate checkbox counts as false for most purposes — if you click it, the checkbox will become checked and the **`checkactions`**, if any, will be triggered — but indeterminate checkboxes are displayed differently in the browser.

This example shows indeterminate checkboxes being used for categories in a shopping list (which could also be sub-tasks in a todo list, or many other things). If only some items in a category are selected, the category checkbox is indeterminate. You can click on the category checkboxes to see how indeterminate states are treated the same as the unchecked state, and clicking the box checks it and applies its check actions (in this case, checking all the boxes in that category). Try editing the **`fruits`** and **`vegetables`** fields on this tiddler and see what happens to the example when you do.

copy to clipboard

```
\define check-all(field-name:"items") <$action-listops $field="selected-$field-name$" $filter="[list[!!
$field-name$]]" />
\define uncheck-all(field-name:"items") <$action-listops $field="selected-$field-name$" $filter="[[]]" />

<$checkbox filter="[list[!!selected-fruits]count[]]" checked={{{ [list[!!fruits]count[]] }}}
unchecked="0" checkactions=<<check-all fruits>> uncheckactions=<<uncheck-all fruits>>
indeterminate="yes"> fruits</$checkbox>
<ul style="list-style: none">
<$list variable="fruit" filter="[list[!!fruits]]">
<li><$checkbox listField="selected-fruits" checked=<<fruit>>> <<fruit>></$checkbox></li>
</$list>
</ul>
<$checkbox filter="[list[!!selected-vegetables]count[]]" checked={{{ [list[!!vegetables]count[]] }}}
unchecked="0" checkactions=<<check-all vegetables>> uncheckactions=<<uncheck-all vegetables>>
indeterminate="yes"> veggies</$checkbox>
<ul style="list-style: none">
<$list variable="veggie" filter="[list[!!vegetables]]">
<li><$checkbox listField="selected-vegetables" checked=<<veggie>>> <<veggie>></$checkbox></li>
```

```
</$list>
</ul>

<p>Selected veggies: {{!!selected-vegetables}}<br/>
Selected fruits: {{!!selected-fruits}}</p>
```

That renders as:

- [x] fruits
- [x] veggies

Selected veggies:
Selected fruits:

CheckboxWidget (index Mode)

17th March 2023 at 5:09pm

CheckboxWidget

index Mode

To use the checkbox widget in index mode set the **index** attribute to a property of a DataTiddler. The **checked** and **unchecked** attributes specify the values to be assigned to the property and correspond to its checked and unchecked states respectively. The **default** attribute is used as a fallback value if the property is undefined.



Make sure to set **`tiddler`** correctly, because non-DataTiddlers will be overwritten without warning

The example below creates a checkbox that is checked if the property in the tiddler *ExampleData* by the name of the current tiddler is equal to **`selected`** and unchecked if the property is an empty string. If the property is undefined then it defaults to an empty string, meaning the checkbox will be unchecked if the property is missing.

copy to clipboard

```
<$checkbox tiddler="ExampleData" index=<<currentTiddler>> checked="selected" unchecked="" default="">
Selected?</$checkbox>
```

That renders as:

☐ Selected?

CheckboxWidget (listField Mode)

17th March 2023 at 5:09pm

CheckboxWidget

`listField` Mode

Using the checkbox widget in list mode requires the **`listField`** attribute to specify the name of a field containing a list. The **checked** attribute specifies the value that should be present or absent in the list when the checkbox is checked or unchecked respectively. If **checked** is absent (or empty) but **unchecked** is present, then the logic will be inverted: the checkbox will be checked when the **unchecked** value is missing from

the list, and unchecked when the **unchecked** value is found in the list. If both **checked** and **unchecked** are present, the checkbox will work like a toggle, replacing the **checked** value with the **unchecked** value and vice-versa. Finally, if neither **checked** nor **unchecked** is specified, the checkbox will be checked if the field has anything in it, but unchecked if the field is missing or empty. (This is rarely useful. Most of the time you want to specify **checked** or **unchecked** or both.)

The **default** attribute is used as a fallback for the checkbox state if the field is not defined.

The following table summarizes the possible combinations:

defined attributes	☐	☑	☐
neither	field missing or list empty no **default** defined	field has any value	–
checked=**item1**	**item1** removed from list	**item1** added to list	–
unchecked=**item2**	**item2** added to list	**item2** removed from list	–
both	**item1** removed from list **item2** added to list	**item1** added to list **item2** removed from list	**item1** not in list **item2** not in list no **default** defined

This example creates a checkbox that is checked if the list field named **colors** contains **green** and unchecked if the field contains **red**. If the field is undefined, or if neither **green** nor **red** appears in the field, then it defaults to **green**, meaning that the checkbox will be checked.

copy to clipboard

```
<$checkbox listField="colors" checked="green" unchecked="red" default="green"> Is "green" in colors?</
$checkbox><br />''colors:'' {{!!colors}}
```

That renders as:

☑ Is "green" in colors?
colors:

Try editing the **colors** field of this tiddler to see how the example changes.

CheckboxWidget (listIndex Mode)

17th March 2023 at 5:09pm
CheckboxWidget

listIndex Mode

Using the checkbox widget in index list mode requires the **listIndex** attribute to specify the the property of a DataTiddler. This property contains a list. The **checked** attribute specifies the value that should be present or absent in the list when the checkbox is checked or unchecked respectively. If **checked** is absent (or empty) but **unchecked** is present, then the logic will be inverted: the checkbox will be checked when the **unchecked** value is missing from the list, and unchecked when the **unchecked** value is found in the list. If both **checked** and **unchecked** are present, the checkbox will work like a toggle, replacing the **checked** value with the **unchecked** value and vice-versa. Finally, if neither **checked** nor **unchecked** is specified, the checkbox will be checked if the field has anything in it, but unchecked if the field is missing or empty. (This is rarely useful. Most of the time you want to specify **checked** or **unchecked** or both.)

The **default** attribute is used as a fallback for the checkbox state if the property is undefined.

The following table summarizes the possible combinations:

defined attributes	☐	☑	☐

neither	property missing or list empty no **default** defined	property has any value	–
checked=item1	**item1** removed from list	**item1** added to list	–
unchecked=item2	**item2** added to list	**item2** removed from list	–
both	**item1** removed from list **item2** added to list	**item1** added to list **item2** removed from list	**item1** not in list **item2** not in list no **default** defined



Make sure to set **tiddler** correctly, because non-DataTiddlers will be overwritten without warning

The example below creates three checkboxes that each control a different value in a property of the *ExampleData* tiddler.

copy to clipboard

```
<$set name=indexName filter="[<currentTiddler>addsuffix[ Colors]]" >
<$checkbox tiddler="ExampleData" listIndex=<<indexName>> checked="green" unchecked="red" default="red">
Green or red?</$checkbox><br/>
<$checkbox tiddler="ExampleData" listIndex=<<indexName>> checked="yellow" unchecked="blue"
default="blue"> Yellow or blue?</$checkbox><br/>
<$checkbox tiddler="ExampleData" listIndex=<<indexName>> checked="orange" unchecked="purple"
default="purple"> Orange or purple?</$checkbox><br/>
Colors list: <$text text={{{ [[ExampleData]getindex<indexName>] }}} />
</$set>
```

That renders as:

☐ Green or red?
☐ Yellow or blue?
☐ Orange or purple?
Colors list:

CheckboxWidget (tag Mode)

17th March 2023 at 5:11pm
CheckboxWidget

`tag` Mode

Using the checkbox widget in tag mode requires the **`tag`** attribute to specify the name of the tag. The checkbox will be checked if the tiddler specified in the **`tiddler`** attribute has the specified tag and unchecked if it does not.

This example creates a checkbox that flips the **done** tag on the current tiddler:

copy to clipboard

```
<$checkbox tag="done"> Is it done?</$checkbox>
```

That renders as:

☐ Is it done?

When the attribute **`invertTag`** is set to **`yes`**, the checkbox will be checked if the tiddler does *NOT* have the specified tag and unchecked if it does.

copy to clipboard

```
<$checkbox tag="done" invertTag="yes"> Is it not done?</$checkbox>
```

That renders as:

☑ Is it not done?

Chick Peas

6th October 2020 at 8:12pm

shopping

This is a sample shopping list item for the Shopping List Example

Chinese (Simplified) Edition

20th September 2014 at 5:02am

Languages

TiddlyWiki 的简体中文翻译版本：

- **空白版本**: https://tiddlywiki.com/languages/zh-Hans/empty.html

另请参阅 中文 (正體) 版。

Chinese (Traditional) Edition

20th September 2014 at 5:02am

Languages

TiddlyWiki 的正體中文翻譯版本：

- **空白版本**: https://tiddlywiki.com/languages/zh-Hant/empty.html

另請參閱 中文 (简体) 版。

ClearPasswordCommand

Commands

Clear the password for subsequent crypto operations

```
--clearpassword
```

Code Blocks in WikiText

13th May 2022 at 2:03pm

WikiText

You can use triple backticks ``` to mark code blocks (abc):

```
```
This will be monospaced
```
```

Renders as:

```
This will be monospaced
```

To be interpreted correctly, the three backticks need to be at the start of the line and immediately followed by a line-break.

Be aware that any preceding paragraph content should be properly terminated with a double line break, too. So, this example is wrong:

```
This is an ordinary paragraph
```
This will be monospaced
```
```

The correct version is:

```
This is an ordinary paragraph

```
This will be monospaced
```
```

Note that some keyboard layouts treat the backtick as a dead key, making it hard to type. The trick is to type three backticks followed by a space. Alternatively, type all six backticks in one go, then a space, and then move the cursor back three characters to type or paste the content.

Code styles and auto format settings for IDEs

6th January 2021 at 4:10pm

Other Resources

Implementation of TiddlyWiki Coding Style Guidelines for different IDEs.

https://gitlab.com/bimlas/tw5-code-style-auto-format

CodeBlockWidget

17th August 2016 at 7:53pm

Widgets

Introduction

The **codeblock** widget renders text in `<pre>` and `<code>` blocks, causing it to be displayed monospace. A language may optionally be specified using the **language** attribute, however syntax highlighting will only be used if the Highlight Plugin is installed.

Content and Attributes

The content of the `<$codeblock>` widget is ignored.

Attribute	Description
code	Contents of the block to render as code
language	Programming language for syntax highlighting

The `language` attribute accepts either:

- a Highlight.js language code (see https://highlightjs.org/static/demo/ for a list)
- a MIME type (eg, `text/html` or `image/svg+xml`)

Examples

Here is an example embedding the contents of a tiddler as a code block.

copy to clipboard

```
<$codeblock code={{$:/editions/tw5.com/macro-examples/say-hi}} />
```

That renders as:

```
\define sayhi(name:"Bugs Bunny",address:"Rabbit Hole Hill")
Hi, I'm $name$ and I live in $address$.
\end
```

A codeblock may also specify a language.

copy to clipboard

```
<$codeblock code="SELECT * FROM users WHERE deleted = false" language="sql" />
```

That renders as:

```
SELECT * FROM users WHERE deleted = false
```

CodeMirror Plugin

28th February 2017 at 11:25am

OfficialPlugins Plugin Editions

The *CodeMirror* plugin adds a sophisticated web-based editor to TiddlyWiki.

See https://tiddlywiki.com/plugins/tiddlywiki/codemirror for a demo.

colour Macro

28th February 2020 at 3:28pm

Core Macros Macros

The **colour** (or **color**) macro returns the CSS value of one the colours in the current palette.

If no such entry exists in the current palette, the **vanilla palette** is used instead. If no such entry exists in the vanilla palette, the system looks for

a configuration tiddler with the title `$:/config/DefaultColourMappings/<colour-name>` and transcludes the colour from the text field. This enables to plugins to ship defaults for colours that are not present in the core palettes.

Parameters

name

The name of the palette entry, e.g. `page-background`

Examples

colour Macro (Examples)

21st February 2015 at 4:48pm

colour Macro Macro Examples

copy to clipboard

```
<<colour code-border>>
```

Try it

copy to clipboard

```
<<colour foreground>>
```

Try it

copy to clipboard

```
<<colour page-background>>
```

Try it

copy to clipboard

```
<<colour sidebar-tab-background>>
```

Try it

copy to clipboard

```
<<colour tag-foreground>>
```

Try it

colour-picker Macro

18th April 2016 at 5:43pm

Core Macros Macros

The **colour-picker** macro displays an interactive colour picker, as can be seen in the core bitmap editor.

Parameters

actions

Wikitext for the action widgets that should be executed when the user selects a colour. Within the text, the variable `colour-picker-value` contains the selected colour.

Examples

colour-picker Macro (Example 1)

16th November 2021 at 4:46am

Macro Examples

Colour: (unset)

Recent:

---

colour-picker Macro (Examples)

18th April 2016 at 5:53pm

colour-picker Macro Macro Examples

copy to clipboard

```
Colour: <$edit-text tiddler='$:/_MyColour' tag='input' placeholder='(unset)' default=''/>

---

<$macrocall $name='colour-picker' actions="

<$action-setfield $tiddler='$:/_MyColour' $value=<<colour-picker-value>>/>

"/>
```

Try it

ColourPalettes

17th June 2016 at 12:12pm

Concepts

A colour palette is a data tiddler that supplies a *CSS* colour value, such as **yellow** or **#fe0**, for each of several colour names, like this:

```
page-background: #fe0
table-border: #ccc
...
```

Several palettes form part of the core. The system tiddler **$:/palette** always contains the title of the currently selected palette tiddler. You can change a palette using the **palette** button, found on the "Tools" tab in the sidebar.

To retrieve the value of a named colour from the current palette, e.g. for use in a stylesheet tiddler, use the **colour** macro:

```
<<colour page-background>>
```

Palette tiddlers should have the following fields:

Name	Value
title	Typically starting with `$:/`
type	`application/x-tiddler-dictionary`

tags	`$:/tags/Palette`
name	Displayed in the palette browser
description	Displayed in the palette browser
text	`name: value` colour definitions

Comma-Separated Values

21st February 2015 at 7:13pm

Definitions

CSV is a standard plain-text format for storing a table of data.

Each row of the table is called a **record** and occupies one line.

The columns are called **fields**. Each field of a row is separated from the next by a comma, and is often delimited by quotation marks.

The `csvtiddlers` macro returns tiddler content in this format.

Commands

26th June 2018 at 2:23pm

Concepts Reference

A **command** is one of the following words, written with a `--` prefix and used as a command-line option under Node.js, indicating which action is desired. See Using TiddlyWiki on Node.js for details of how to use them.

- build

- clearpassword
- commands
- deletetiddlers
- editions
- fetch
- help
- import
- init
- listen
- load
- makelibrary
- output
- password
- render
- rendertiddler
- rendertiddlers
- save
- savetiddler
- savetiddlers
- savewikifolder
- server
- setfield

- unpackplugin
- verbose
- version

commands Operator

3rd February 2015 at 7:41pm

Filter Operators Selection Constructors Special Operators

purpose	select the titles of all the Node.js commands
input	ignored
parameter	none
output	the command words that can be given to TiddlyWiki on Node.js

Learn more about how to use Filters

Examples

commands Operator (Examples)

18th January 2015 at 7:31pm

commands Operator Operator Examples

`[commands[]]`

Try it

CommandsCommand

4th December 2022 at 9:25pm

Commands

Sequentially run the command tokens returned from a filter

```
--commands <filter>
```

Examples

```
--commands "[enlist:raw{$:/build-commands-as-text}]"
```

```
--commands "[{$:/build-commands-as-json}jsonindexes[]] :map[{$:/build-commands-as-json}
jsonget<currentTiddler>]"
```

Community

22nd March 2021 at 4:22pm

TableOfContents



The latest and most useful links are now being gathered in the Community Links Aggregator.

Once all the relevant links have been transferred over these entries will be removed from the main tiddlywiki.com site.

Forums | Latest | Tutorials | Community Editions | Community Plugins | Community Themes | Community Palettes | Other Resources | Examples | Articles | Meetups

The latest news, articles, resources and examples.

"TiddlyWiki Posts" by Jeffrey Kishner

Posted 7 months ago

A collection of articles covering integration with Fargo, Font Awesome and Google Calendar, and tips for managing task lists. The original site is missing, but a link to an archive is provided.

https://web.archive.org/web/20221015011644/http://blog.jeffreykishner.com/tiddlywiki/

> I have become a regular user of TiddlyWiki and have become so accustomed to using Font Awesome icons in Fargo that I wanted to incorporate the icons into my wikis as well.

GitHub Saver Tutorial by Mohammad

Posted 7 months ago

GitHub Saver is a step by step tutorial that shows how to integrate Tiddlywiki 5 and GitHub Pages to create websites hosted on https://github.com/.

https://kookma.github.io/TW5-GitHub-Saver/

This instruction is based on Tiddlywiki single html file model, while it can use subfolder for extra materials like images, audios, videos, pdfs,... in separate folders.

Other tutorials

Tiddlywiki, Travis-CI and GitHub Pages
https://kookma.github.io/Tiddlywiki-Travis-CI/
This wiki shows how to set up websites hosted on GitHub Pages using Travis-CI and Tiddlywiki 5 on Node.js.

Tiddlywiki and GitHub Pages
https://kookma.github.io/Tiddlywiki-and-GitHub-Pages/
This instruction is based on local edit, save and push to GitHub. It does NOT use the new GitHub Saver mechanism (requires TW 5.1.20+) which lets edit and save directly from Tiddlywiki!

"TW5 Magick" by Stephan Hradek

Posted 2 years ago

Stephan Hradek's growing catalogue of tips, tutorials, and other resources. Stephan has also developed several plugins and macros that you can find at http://tiddlystuff.tiddlyspot.com/.

http://tw5magick.tiddlyspot.com/

> Some things in TiddlyWiki seem like Magic. As my preferred nick is Skeeve, I felt that TW5 Magick is a proper name for this collection of some of the "magic" tricks one can do with TiddlyWiki.

"Reveal.js" by Devin Weaver

Posted 2 years ago

Embed Reveal.js presentations inside your TiddlyWiki.

https://sukima.github.io/tiddlywiki-reveal-js/

> reveal.js enables you to create beautiful interactive slide decks using HTML.
>
> With the power of TiddlyWiki and Reveal.js one can…
>
> - …make an interactive tiddler with fancy animations
> - …Make it go full screen for professional keynote
> - …Distribute the slideshow with examples and additional resources in one HTML file

TW-Scripts by Mohammad

Posted 3 years ago

TW-Scripts is one of the most comprehensive collections of solutions for Tiddlywiki 5.

https://kookma.github.io/TW-Scripts/

TW-Scripts includes:

- Collected solutions to questions in Tiddlywiki Google groups
- Learn through examples
- Templates, stylesheets
- Wikitext, macros, and snippets
- Tips and tricks
- Search tools

TiddlyWiki Video Tutorials by Francis Meetze

Posted 3 years ago

An extensive tutorial for getting started with TiddlyWiki. Recommended

https://www.youtube.com/playlist?list=PLzZCajspPU_UjFn0uy-J9URz0LP4zhxRK

> This tutorial is aimed at giving you a basic foundation so that you can start using TiddlyWiki right away. Here we take a look at the technology behind Tiddlywiki, how to install it and create your first Tiddler.

RegExp in Tiddlywiki by Mohammad

Posted 3 years ago

RegExp in Tiddlywiki contains practical use cases of regular expression in Tiddlywiki.

http://tw-regexp.tiddlyspot.com/

The regular expression is a way to describe complex search patterns using sequences of characters .

RegExp in Tiddlywiki has four main parts

- Using regular expression to match a pattern in tiddler title
- Using regular expression to match a pattern in tiddler fields excluding tiddler text (body)
- Using regular expression to match a pattern in tiddler body (text field)
- Using regular expression for validation like username, password, etc.

"TB5 - a pocket full of tips" by Tobias Beer

Posted 3 years ago

An enormous collection of tips beautifully curated by Tobias Beer. It is actually a collection of sites, with sub-sites dedicated to topics such as filtering (http://tobibeer.github.io/tw/filters/).

http://tobibeer.github.io/tb5/

> A collection of references and sandbox for testing concepts around TiddlyWiki 5...

Community Editions

1st January 2021 at 4:18pm

Community

These are prepackaged editions created by the TiddlyWiki Community. These are *TiddlyWikis* with added plugins and configurations to facilitate a certain use-case. These are great starting points if you want to quickly jump into TiddlyWiki and start using it without spending too much time configuring yourself.

Tidme by oflg

Posted 5 months ago

Lifelong knowledge, deep in Mind.

https://github.com/oflg/Tidme

Tidme uses advanced Incremental Learning concepts to make it your powerful second brain for acquiring lifelong knowledge.

- Read Articles like *SuperMemo*
- Learn languages like *LingQ*
- Memory Notes like Anki

"TiddlyStudy" by Kebi

Posted 13 months ago

A adaptation of TiddlyWiki perfect for using as a Notebook sysetem.

https://postkevone.github.io/tiddlystudy/

> Tiddly Research is a free and local note-taking environment based on TiddlyWiki
>
> Core Features:
>
> - Bidirectional note linking on multiple levels
> - Tags
> - Linked references
> - Unlinked references
> - Transclusions
> - Bullet point based hierarchical note taking structure
> - Integration with the spaced repetition flashcards tool Anki
> - TODOs
> - and more

"Drift - Collect, Organise, Grow." by Tony K

Posted 13 months ago

A adaptation of TiddlyWiki perfect for using as a Notebook sysetem.

https://github.com/bmann/drift-tiddlywiki-template/tree/master/drift

> Drift is an adaptation of TiddlyWiki with the goal of helping you Collect. Organise. and Grow. your ideas while keeping tab on how they interconnect together and fluidly Drift from one to another.
>
> The 2 main components of Drift are TWCrosslinks and *DailyNotes* so you never miss an idea

> Drift was inspired by *TiddlyBlink* / Stroll and adapted to fit my needs.

Projectify by Nicolas Petton

Posted 23 months ago

Project & todo management for TiddlyWiki.

https://github.com/ThaddeusJiang/Projectify

Projectify is a TiddlyWiki plugin for managing projects & todo lists, inspired by products like Todoist and Basecamp.

Projectify features:

- An inbox for collecting tasks and thoughts
- Projects to structure tasks together into separate todo-lists
- Support for scheduling tasks
- A dashboard to quickly view all projects, the inbox, or scheduled tasks

`Note:` The original site, https://projectify.wiki, no longer exists. The provided site is a maintained fork, though not by the original creator.

"GSD5" by Roma Hicks

Posted 2 years ago

An adaptation of the TiddlyWiki powered GTD® system formerly known as MonkeyGTD for TiddlyWiki version 5.

http://gsd5.tiddlyhost.com/

> GSD5 is a Getting-Thing-Done tool for TiddlyWiki5 based off the mGSD classic TiddlyWiki.

"Noteself" by Danielo Rodríguez

Posted 2 years ago

NoteSelf is your personal, private, customisable, Evernote-like experience. You want cloud? Fine! You don't? Fine too! It's all yours, It's your decision! Why not be a bit selfish?

https://noteself.github.io/

> NoteSelf is the perfect place to store your ideas, notes, thoughts, tips, tricks, recipes... whatever you want to put into it! NoteSelf will store, index, and make it easily searchable, so you will find it instantly whenever you need it!
>
> You already know and love Evernote, we know it. It is comfortable, it syncs, has search capabilities... Wow, it's almost perfect, but what about:
>
> - Privacy - Everything is stored on the Evernote's servers!
> - Customisation - If you don't like the interface of Evernote, there's nothing you can do
>
> NoteSelf is built on top of TiddlyWiki, a powerful, free, highly customisable and open-source personal wiki. We took the best of it, it's powerful customisation system, and mixed it with one of the best embedded databases available, PouchDb, for bringing the synchronisation capabilities you need.

"Wikilabs" by PMario

Posted 3 years ago

A collection of references for testing concepts around TiddlyWiki 5...

https://wikilabs.github.io

> This is PMario's playground with ramblings about TiddlyWiki and related stuff.

> It basically is a "jump off" page to various TiddlyWiki sites, that show editions, themes and plugins.

"TiddlyWiki for Scholars" by Alberto Molina

Posted 3 years ago

An early release of an edition of TiddlyWiki customised for rigorous, academic notetaking.

http://tw5.scholars.tiddlyspot.com/

> TiddlyWiki for Scholars is a personal customisation of TiddlyWiki 5.0.13-beta I made for note-taking from readings, but it can be useful for other purposes. The idea behind this adaptation is to show and to allow creating relevant data related to the current tiddler without leaving the tiddler.

"Cardo - Task and Project Management Wiki" by David Szego

Posted 3 years ago

Task & Delegation Tracking, Meetings & Agenda Items, Project Roles, Progress & Status Updates, Reference Items, Tickler Calendar, Conversation Logging, Book Notes & Library, and More!

http://cardo.wiki

> Cardo is a standalone, browser-based tool that can be used as a simple task manager, or as a complex Project Management system (and indeed, I do use it this way in my daily work) as well as a fully Wiki-ized personal knowledge store. It runs completely independently in the browser, even without an Internet connection, making it possible to carry around on a USB stick, or to use on the morning commute.

"Stroll - a Roam-like experience" by David Gifford

Posted 3 years ago

A adaptation of TiddlyWiki perfect for using as a Notebook sysetem.

https://giffmex.org/stroll/stroll.html

> Stroll is a TiddlyWiki adaptation imitating certain features of Roam to help users easily see connections between their notes and navigate between them. Features include:
>
> - bi-directional links,
> - autocompletion when linking,
> - renaming of links upon changing tiddler titles,
> - side-by-side editing of multiple notes.

Community Links Aggregator

22nd March 2021 at 4:18pm

Community

The TiddlyWiki Community Links Aggregator is a collection of regularly updated links to useful and interesting TiddlyWiki material curated by our team of community editors. The site aggregates links curated by individual members of the TiddlyWiki community. It lets you see the latest links, and explore them through categories and timelines.

https://links.tiddlywiki.org/

This site works best with a crowd of people posting links. The pressure on individuals is reduced because not everybody needs to catch every interesting link that flies past. The aggregation effects reduce the impact of mistakes. For example, if one person mis-tags a link under an inappropriate topic, the site will show that only one person added that tag, versus the majority using more appropriate tags. In that way, we hope that a sort of wisdom of the crowds will emerge, with consensus emerging as to the most useful ways to describe and categorise links.

Community Palettes

1st January 2021 at 4:25pm

Community

This is a list of palettes made by members of the TiddlyWiki Community. Palettes change the colourscheme of TiddlyWiki and can be used in combination with themes.

"JD's palettes" by JD

Posted 3 years ago

Some palettes by JD.

http://j.d.palette.tiddlyspot.com/

> You can install any or all of these by dragging any of the below links into your Tiddlywiki:
>
> - Brown One
> - C8 Dark
> - Cool Mint
> - Neonize
> - Spartan Day
> - Spartan Night

"Dracula - a dark palette for TiddlyWiki" by SlyBouhafs & zenorocha

Posted 3 years ago

A dark palette for TiddlyWiki.

https://draculatheme.com/tiddlywiki

> Install manually
>
> Open the Control Panel > Appereance > Palette, scroll down and clone a platte, rename it to Dracula and copy the code from the website.
>
> Activating theme
>
> - Open the Control Panel > Appereance > Palette
> - Scroll down and select Dracula
> - Boom! It's working

Community Plugins

1st January 2021 at 4:14pm

Community

These are plugins created by the TiddlyWiki Community. Visit their website to try out the plugin. Carefully read the installing instructions and back up before installing! These plugins may not always be up to date and can contain bugs or unwanted behaviour. For even more plugins, visit the TiddlyWiki Groups. New plugins get posted and announced there first.

Free Spaced Repetition Scheduler for TiddlyWiki by oflg

Posted 5 months ago

TiddlyWiki-based memory programme using advanced FSRS algorithm

https://github.com/open-spaced-repetition/fsrs4tw

Projectify by Nicolas Petton

Posted 23 months ago

Project & todo management for TiddlyWiki.

https://github.com/ThaddeusJiang/Projectify

Projectify is a TiddlyWiki plugin for managing projects & todo lists, inspired by products like Todoist and Basecamp.

Projectify features:

- An inbox for collecting tasks and thoughts
- Projects to structure tasks together into separate todo-lists
- Support for scheduling tasks
- A dashboard to quickly view all projects, the inbox, or scheduled tasks

`Note:` The original site, https://projectify.wiki, no longer exists. The provided site is a maintained fork, though not by the original creator.

TiddlyClip by buggyjay

Posted 23 months ago

BuggyJay's TiddlyClip browser extension for Firefox permits clipping of text and graphics from web pages:

http://tiddlyclip.tiddlyspot.com

> TiddlyClip allows parts of webpages to be clipped into a TiddlyWiki, and consists of two parts, the browser addon and the TiddlyWiki plugin. The addon is completely memoryless, any configuration is determined by the current TiddlyWiki that the user has select to work with (we say that the addon is docked to the TiddlyWiki). TiddlyClip is designed to work (in a basic mode) without configuration. Once the addon and plugin are installed, all the user has to do is select which TW to dock to.

"GSD5" by Roma Hicks

Posted 2 years ago

An adaptation of the TiddlyWiki powered GTD® system formerly known as MonkeyGTD for TiddlyWiki version 5.

http://gsd5.tiddlyhost.com/

> GSD5 is a Getting-Thing-Done tool for TiddlyWiki5 based off the mGSD classic TiddlyWiki.

"TW5 Magick" by Stephan Hradek

Posted 2 years ago

Stephan Hradek's growing catalogue of tips, tutorials, and other resources. Stephan has also developed several plugins and macros that you can find at http://tiddlystuff.tiddlyspot.com/.

http://tw5magick.tiddlyspot.com/

> Some things in TiddlyWiki seem like Magic. As my preferred nick is Skeeve, I felt that TW5 Magick is a proper name for this collection of some of the "magic" tricks one can do with TiddlyWiki.

TiddlyMap Plugin by Felix Küppers

Posted 2 years ago

An interactive network visualisation plugin based on Vis.js. A demo that also contains installation instructions can be found here: http://tiddlymap.org. The plugin's GitHub repository can be found here.

> TiddlyMap is a TiddlyWiki plugin that allows you to link your wiki-topics (tiddlers) in order to create clickable graphs. By creating relations between your topics you can easily do the following:
>
> - Create concept maps and quickly manifest your ideas in tiddlers.
> - Create task-dependency graphs to organise and describe your tasks.
> - Visualise your topic structures to get an immediate grasp of topics and relations.
>
> In general you may create, visualise and describe any network-structure you have in mind.

"Encrypt single tiddler plugin" by Danielo Rodriguez

Posted 2 years ago

A plugin that allows to encrypt your tiddlers individually Danielo Rodriguez.

http://danielorodriguez.com/TW5-EncryptTiddlerPlugin/

> Advantages:
>
> - You can specify a different password for each tiddler if you want.
> - You don't have to encrypt your whole wiky.
> - If you forget your password, you only lose a tiddler.
> - It's possible to edit the tiddler content , tags and fields except the encrypt field after encryption.
> - [...]

vis.js Timeline by emkay, revived by kixam

Posted 3 years ago

An interactive timeline visualisation plugin based on Vis.js. A demo can be found here: http://kixam.github.io/TW5-visjsTimeline/.

> vis.js Timeline is a TiddlyWiki plugin that allows you to link your wiki-topics (tiddlers) in order to create clickable timelines. This project is a direct continuation of emkay's plugin.

Tinka by Andreas Hahn

Posted 3 years ago

The Tinka plugin provides the ability to create and modify plugins in the browser, without requiring the use of Node.js.

See https://github.com/TinkaPlugin/Tinka for the GitHub repo.

https://tinkaplugin.github.io/

> This is a Control Panel extension that aims to simplify the plugin creation and editing process. After installing, you will find a new tab in your control panel that makes creating and modifying plugins a little bit easier.

rboue's plugins for TiddlyWiki

Posted 3 years ago

Featuring plugins that integrate two interesting new capabilities:

- SocialCalc providing a powerful spreadsheet
- Three.js providing interactive 3D graphics

http://rboue.tiddlyspot.com

> I developed some plugins related to engineering tasks.
>
> After using them for personal projects I'm glad to share with TiddlyWiki community.

Plugins by TheDiveO

Posted 3 years ago

A collection of plugins from *TheDiveO*.

http://thediveo.github.io

TheDiveO's Third Flow plugin construction system:

> The ThirdFlow plugin brings to you another way to develop customisation plugins for TiddlyWiki 5. It is not enforcing a specific development flow, it simply tries to help you. Otherwise, it tries to stay out of your way.

TheDiveO's FontAwesome plugin:

> The *FontAwesome* plugin supports embedding Font Awesome in TiddlyWiki 5. There is no need to install this font into your operating system in order to use it with TiddlyWiki 5. The font is already embedded in this TiddlyWiki 5 customisation plugin instead, so nothing else to install.

MathJax Plugin by Martin Kantor

Posted 3 years ago

An experimental *MathJax* plugin for TiddlyWiki version 5. As Martin says, the implementation is a bit of a hack but may be useful until we have a better alternative.

http://mathjax-tw5.kantorsite.net

> Welcome. I have created plugin for TiddlyWiki 5 which allows you to use *MathJax* (math in *TeX* and MathML) inside TiddlyWiki 5. It's unofficial plugin and it doesn't follow general policy of TiddlyWiki as stand-alone solution but it works. So you can use it if you want.

Locator plugin by bimlas

Posted 3 years ago

For those who use many tags or store many different topics in a common wiki the Locator plugin is a table of contents widget and an enhanced search engine that gives you the opportunity to filter results by related tags. Unlike table of contents, standard search and list of tags, this plugin offers these features in an organic, collaborative way.

https://bimlas.github.io/tw5-locator/

Leaflet maps plugin by Sylvain Comte

Posted 3 years ago

A mapping plugin based on Leaflet library. Allows you to create maps based upon geographical data contained in tiddlers.

http://sycom.github.io/TiddlyWiki-Plugins/#Leaflet%20plugin

> The leaflet plugin is a (working) attempt to integrate the leaflet library in TiddlyWiki in order to display geographical purpose tiddlers.
>
> For now `<$leafmap>` widget displays an interactive map. Select size, location and zoom, clustering distance, and background. You can display data : geojson, point(s), polygon(s) and/or polyline(s) directly or calling *GeoTiddler*(s). Tiddler can be called individually, by list or with a filter.
>
> Have a look at demo page on the web.

Kin filter operator by bimlas

Posted 3 years ago

The purpose of the kin operator with examples:

- Finds related tags, related tiddlers in any depth
- Finds out where base tiddler originates and what other elements originate from it
- Finds the ancestors and successors of a family member
- Finds the "leaves" of the branch of the base tiddler in a tree-like structure (where the base tiddler is a leaf)
- Finds the super- and subsets / groups of a mathematical set (where the base tiddler is a set)

https://bimlas.github.io/tw5-kin-filter/

Json Mangler plugin by Joshua Fontany

Posted 3 years ago

Extend tiddlywiki to parse complex ("nested") json data tiddlers.

Json Mangler introduces a new path syntax for indexes of json data tiddlers , and includes many supporting tools, filters, widgets, etc.

Example Wiki: https://joshuafontany.github.io/TW5-JsonMangler/

Source: https://github.com/joshuafontany/TW5-JsonMangler

IndexedDB Plugin by Andreas Abeck

Posted 3 years ago

A plugin that allows changes to be synchronised with the *IndexedDB* database that is built-in to most browsers.

http://tw5-dev.cibm.de

> Being quite new to TW5 development I tried my best to draft a *SyncAdaptor* to store tiddlers in *IndexedDb*. This is quite useful if an application / plugin installation is not possible (we have quite some restrictions at work...).

Full Text Search Plugin by Rob Hoelz

Posted 3 years ago

A plugin to integrate the popular lunr.js search engine, giving TiddlyWiki much more sophisticated search capabilities:

https://hoelz.ro/files/fts.html

> Provides an alternative search result list that orders results by search relevance and ignores differences in word forms (ex. *tag* vs *tags*).
>
> On my personal wiki, I have the problem that there are terms I use across a lot of tiddlers, and sometimes I'll use different forms (such as the aforementioned *tag* vs *tags*). I wanted a plugin to allow me to find the tiddler I'm looking for quickly and didn't require me to worry about how I declined a noun or inflected a verb - so I wrote this plugin, which provides an alternative search list powered by lunr.js.

Eucaly's Tiddly World

Posted 3 years ago

I collect my tw5 creations on-line as http://eucaly-tw5.tiddlyspot.com/

Available Plugins :

- *TitleMe* - Return in-place tiddler title, to link / show / edit the source tiddler
- *QuickJump* - Quick jump to tiddlers in story list

- *FlexWidth* - Flexibly tweak sidebar width for TiddlyWiki 5 (Inspired by TW5 tristate Sidebar)
- *MatchFilter* - returns matching text instead of a list of the tiddlers from *FieldFilter*
- *PopupTagger* - *TagBar* / *TagTable* with popup list of tags (Inspired by *MonkeyTaggerMacro* for TiddlyWiki classic)

Disqus comments plugin by bimlas

Posted 3 years ago

Use this plugin to give your visitors the opportunity to comment on your tiddlers without changing the wiki itself.

Disqus is a networked community platform used by hundreds of thousands of sites all over the web. With Disqus, your website gains a feature-rich comment system complete with social network integration, advanced administration and moderation options, and other extensive community functions.

https://bimlas.github.io/tw5-disqus/

datepicker plugin based on Pikaday, by kixam

Posted 3 years ago

An interactive date picker plugin based on Pikaday. A demo can be found here: http://kixam.github.io/TW5-datePicker/.

> datepicker is a TiddlyWiki plugin that allows you to use a widget much like EditTextWidget to pick a date, and optionaly a time. It shows a nice interactive mini-calendar when clicked, allows you to set a date format for display, and is compatible with core date fields such as `modified` and `created`.

Ace Editor Plugin by Joerg Plewe

Posted 3 years ago

A plugin to integrate the Ace editor into TiddlyWiki.

http://innoq.tiddlyspot.com

> Being quite new to TW5 development I tried my best to draft a *SyncAdaptor* to store tiddlers in *IndexedDb*. This is quite useful if an application / plugin installation is not possible (we have quite some restrictions at work...).

"Wikilabs" by PMario

Posted 3 years ago

A collection of references for testing concepts around TiddlyWiki 5...

https://wikilabs.github.io

> This is PMario's playground with ramblings about TiddlyWiki and related stuff.
>
> It basically is a "jump off" page to various TiddlyWiki sites, that show editions, themes and plugins.

"TW5-TeXZilla" plugin by Joe Renes

Posted 3 years ago

An alternative *LaTeX* plugin for TiddlyWiki5. It supports a wider set of *LaTeX* commands than KaTeX, though not as extensive as *MathJax*

http://tw5-texzilla.tiddlyspot.com

> This plugin provides *LaTeX* support in TiddlyWiki5, using the TeXZilla latex parser to generate MathML.
>
> Only Firefox and Safari support MathML at the moment, so the plugin is only useful if you're using one of these browsers. It might be possible to combine this plugin with *MathJax* to convert MathML to something that other browsers can understand, e.g. HTML+CSS, but I haven't tried this.

"TiddlyWiki5 Playground" by Ton Gerner

Posted 3 years ago

Huge collection of customisations and tweaks from Ton Gerner.

http://tongerner.tiddlyspot.com/

> I am a longtime user of TiddlyWiki Classic and still use it daily since you can't do everything in TiddlyWiki 5 (yet) ;)
>
> I started experimenting with TW5 at the end of September 2013 (alpha10). Since I missed a few things in the layout I got used to in TiddlyWiki Classic 1, I started with modifying the layout of TW5. To share my knowledge, I made a few guides about these 'modifications'. The guides started as a non-linear personal web notebook (yeah, the subtitle of TiddlyWiki!). I started experimenting with layout things and used TW5 to document my experiments.

"Tekan: Kanban that stays with you" by Riz

Posted 3 years ago

A Kanban implementation for TiddlyWiki with a familiar look and feel (see the demo).

https://ibnishak.github.io/Tesseract/projects/tekan/Tekan.html

> This is Tekan, a TW based kanban. The easiest way to know tekan is to use it. So let us begin, shall we?

"Slides and Stories" by Jan

Posted 3 years ago

Slides and Stories is a repository of the tools to optimise TW for scientific research and for presenting and publishing results.

http://slidesnstories.tiddlyspot.com

> SlidesnStories
>
> ...is a tool to save and reload sets of Tiddlers. This is already working, The aim is to develop a mechanism to transform these into instant slideshows.
>
> MetaTabs
>
> is a tool which gives you the possibility to attach and show specific information in tabs at the bottom of each Tiddler: ideas, notes, tasks, stories and presentations, comments, annotations and footnotes.
> It is inspired by MagicTabs and Tiddlyscholar by Alberto Molina Pérez, which are very elaborated tools that alas do not seem to be in active development any more. I wanted to change the behaviour in some ways. For example I wanted show tabs only if they have content.
>
> ...furthermore there is
>
> - an export-plugin,
> - a footnote-plugin,
> - and a plugin to rearrange Tiddlers in the StoryRiver

"SK Plugins" by Stephen Kimmel

Posted 3 years ago

A collection of plugins by Stephen Kimmel, including a number of additional editor toolbar buttons.

http://skplugins.tiddlyspot.com/

> In keeping with the general spirit of the TiddlyWiki community, ie, you develop something useful and/or potentially interesting you share it with others, I'm sharing some of the plugins I've developed to go with the new Editor Toolbar. You'll find them and a more extensive discussion of what each does at http://skplugins.tiddlyspot.com/

"Mal's Sandbox" by Mal

Posted 3 years ago

A collection of extensions by @malgam

http://malsandbox.tiddlyspot.com

> The "loadnew" plugin provides the JavaScript module *$:/plugins/malgam/loadnew/loadnew.js* that implements an incremental load command for Tiddlywiki 5.
>
> The javascript code is based on the built-in module **$:/core/modules/commands/load.js** that loads tiddlers from a Tiddlywiki file into a node.js Tiddlywiki instance. This command loads all non-system tiddlers, irrespective of whether they have been modified since last loaded. As a result, all tiddlers in the node.js tiddlers folder will end up with a new modification date.
>
> In contrast, the loadnew command only loads tiddlers that either don't already exist in the tiddlers folder, or have a newer modified date than the existing tiddler.
>
> With a backup strategy that uses a file system snapshot process to preserve historical versions of tiddlers, the loadnew command minimises the sizes of the resulting snapshots.

"JD Mobile Layout plugin" by JD

Posted 3 years ago

A plugin that improves the usability of TiddlyWiki on smartphones by JD:

http://j.d.simplemobile.tiddlyspot.com

"BJTools" by buggyj

Posted 3 years ago

buggyj has created several useful plugins, including a WYSIWYG HTML editor, a configurable calendar and tag lists with draggable ordering.

http://bjtools.tiddlyspot.com

Slider by Mohammad

Posted 3 years ago

Slider is a plugin to create an ordered set of tiddlers also called Trail.

https://kookma.github.io/slider/

A trail can be used to create a sequence of selected contents like step by step tutorial, guided help, lessons and similar.

The slider plugin user interface contains three sections

- A sidebar tab called Trails to manage trails
- A dashboard for each trail (where trail structure is managed)
- Tiddlers in a trail, also called slides

Tiddlyshow by Mohammad

Posted 3 years ago

Tiddlyshow is a small application of Tiddlywiki for presentation and slideshow. It can also be used as a plugin.

https://kookma.github.io/Tiddlyshow/

Tiddlyshow contains the following features

- Tools for preparing slides
- Shortcut keys for navigation (forward and backward)
- Themes to colorify and customise the slideshow

Refnotes by Mohammad

Posted 3 years ago

Refnotes plugin is a set of macros and stylesheets for creating abbreviations, footnotes and citations. It also makes tables of footnotes, abbreviations (glossary) and references (bibliography table).

https://kookma.github.io/TW-Refnotes/

Refnotes contains codes and elements to

Create and insert
Abbreviation
Footnote
Reference

Create tables of
Abbreviations or glossary
Footnotes and endnotes
References (bibliography) using different output style

Timelines by Mohammad

Posted 3 years ago

Timelines plugin is for creating vertical and horizontal timeline and sequences.

https://kookma.github.io/TW-Timelines/

The content or description and data of events are stored in individual tiddlers.

Timelines contain two timeline macros and two sequence macros. A sequence macro shows event contents in sequence not necessarily in chronological order. The order can be determined by tag or list field or other methods.

Kookma Plugin Library by Mohammad

Posted 3 years ago

This library contains most useful plugins developed under the name Kookma for Tiddlywiki 5.

https://kookma.github.io/TW-PluginLibrary/

It has two parts

The plugin library introduces a very simple mechanism like Tiddlywiki Official Plugin Library to let select among the published plugins and install any number of them you like.

Searchwikis by Mohammad

Posted 3 years ago

Searchwikis plugin uses index (dataTiddler) to search multiple external Tiddlywikis.

https://kookma.github.io/TW-Searchwikis/

It has two parts

1. An indexer, to build an index of all tiddlers in an external wiki
2. A search tool to search indexes and display a link to a tiddler found in an external wiki

Then one master wiki can host many index tiddlers and lets to search several external wikis through index tiddlers.

Searchwikis enable to have a central wiki and search all other wikis from one place.

Utility by Mohammad

Posted 3 years ago

The utility plugin objective is to provide set of tools for developers and authors.

https://kookma.github.io/TW-Utility/

These tools include simple transclusions, show raw contents of tiddlers, author tools, wikitext macros, show fields in view mode, etc.

Favorites by Mohammad

Posted 3 years ago

Favorites plugin is a set of tool for creating favorites (bookmarks) in Tiddlywiki. Each favorite item is a shortcut to a tiddler.

https://kookma.github.io/TW-Favorites/

A favorite item also called a bookmark is a shortcut, you are creating for quick access to it. You can access that bookmark at any time and view that as

many times as you want without having to search and find it again.

The Favorites plugin contains three modes

- flat mode: one favorite list, no folder
- structured mode: use folder, search tool, export and delete tool
- frozen list: read only, simple table of content style

Trashbin by Mohammad

Posted 3 years ago

The concept behind Trashbin plugin is to have a simple mechanism to move deleted tiddlers to Trashbin and be able to restore them later if required.

https://kookma.github.io/TW-Trashbin/

The trash (also known as the Recycle Bin in Microsoft Windows) is a temporary storage for tiddlers that have been deleted in a Tiddlywiki by the user, but not yet permanently erased.

Typically, a trash bin is presented as a special storage, allowing the user to browse deleted (removed) tiddlers, undelete those that were deleted by mistake, or delete them permanently (either one by one, or by the "Empty Trash" function).

Todolist by Mohammad

Posted 3 years ago

Todolist is a small pure wikitext plugin, contain all tools to work with todo list.

https://kookma.github.io/TW-Todolist/

With Todolist, easily organise and prioritise your tasks and projects so you'll always know exactly what to work on next.

Todolist creates custom UI and lets you to add new items, set priority, done/undone items, archive, delete. Using Todolist plugin it is possible to create several todo lists in one Tiddlywiki.

Shiraz by Mohammad

Posted 3 years ago

Shiraz plugin is a very small in size and framework to create stylish contents in Tiddlywiki.

https://kookma.github.io/TW-Shiraz/

Shiraz plugin contains extended markups, macros, styles, and many customisation to empty Tiddlywiki and can be used as a **starter kit**.

Some of Shiraz features are:

- Customized elements like, alerts, cards, panels, badges
- Dynamic tables
- Sortable tables
- Display on demand, slider, and details
- Images, basic image macros, slidein, and overlay images, polaroid and pretty images
- Table customisation tools
- Sticky footer
- Multi columns tiddler
- Multi column story river
- List search
- Badge status
- Notebook style
- Test utilities
- Stylish buttons

Adding Shiraz plugin to any Tiddlywiki converts it to a full production tool. Shiraz uses modified CSS classes from Bootstrap.

Tiddler Commander by Mohammad

Posted 3 years ago

Tiddler Commander plugin, in short **Commander** is a unique tool for batch operations on tiddlers.

https://kookma.github.io/TW-Commander/

Commander has many features including:

- Bulk tiddler creation/deletion
- Selective operation
- Combo search to filter and select tiddlers
- Title operation: add, remove prefixes and suffixes, also remove cahras form begining and end of title (on renaming tiddlers, relink can be used to update title in other tiddlers)
- Tag operation: add, remove, replace
- Field operation: add, remove, rename, and set field value
- SnR, search and replace in all fields including text, tags, *title*, and common fields
- Inspect, to review and inspect tiddlers in one place, scroll among them and edit all fields (including common fields), tags, text (title is an exception!)
- Log, create logs of all operations
- Search, *save and load* any combination of filter search

Community Themes

1st January 2021 at 4:25pm

Community

This is a list of themes made by members of the TiddlyWiki Community. Themes change the look and feel of TiddlyWiki, allowing for another

degree of customisation. Installing themes works the same as plugins.

"Notebook theme" by Nicolas Petton

Posted 37 days ago

Notebook is a clean, uncluttered theme for TiddlyWiki.

https://saqimtiaz.github.io/sq-tw/notebook.html

> Notebook is a clean, uncluttered theme for TiddlyWiki.
>
> - The theme is responsive and comes with mobile support through the included notebook-mobile plugin.
> - Notebook comes with 3 default colour palettes, a grey, beige and dark one. To switch between colour palettes, click the color palette switch button in the top bar.
> - Notebook uses a custom left sidebar, with another sticky bar at the top of the page (or at the bottom on mobile).

"Moments: A little color won't hurt" by Riz

Posted 3 years ago

An adaption of the "Moments" theme for TiddlyWiki.

http://tesseractmoments.surge.sh/

> Moments is a beautiful free and responsive theme available for wordpress from here: http://www.s5themes.com/theme/moments/. This is an adaptation of the same for TW5 platform. All the design credits goes to the site and the original designer. If TW5 is going to be here for 25 years, might as well suit up.

"Ghostwriter theme by Rory Gibson" adapted for TW5 by Riz

Posted 3 years ago

An adaption of the "Ghostwriter" theme for TiddlyWiki.

https://ibnishak.github.io/ghostwriter/

> Ghostwriter is a popular, minimalist and mobile responsive theme for the blogging platform ghost. This is an adaptation for TiddlyWiki.

"Mono theme" by JD

Posted 3 years ago

This theme by JD features a left side bar, advanced functionality from the search bar and plenty of customizability built in.

http://j.d.mono.tiddlyspot.com/

"Whitespace, a negative-spacious, webapp-nostalgic theme" by JD

Posted 3 years ago

A beautiful theme that leverages the white spaces between elements.

http://j.d.whitespace.tiddlyspot.com/

compare Operator

12th April 2020 at 8:15pm

Filter Operators Mathematics Operators Negatable Operators String Operators

purpose	filter the input by comparing each item against the parameter
input	a selection of titles
suffix	the **compare** operator uses a rich suffix, see below for details
parameter	the value to compare
output	those input titles matching the specified comparison
! **output**	those input titles *NOT* matching the specified comparison

Learn more about how to use Filters

Introduced in v5.1.22 The **compare** filter allows numerical, string and date comparisons to be performed.

The **compare** operator uses an extended syntax to specify all the options:

```
[compare:<type>:<mode>[<parameter>]]
```

The **type** can be:

- "number" - invalid numbers are interpreted as zero
- "integer" - invalid integers are interpreted as zero
- "string"
- "date" - invalid dates are interpreted as 1st January 1970
- "version" - invalid versions are interpreted as "v0.0.0"

The **mode** can be:

- "eq" - equal to
- "ne" - not equal to
- "gteq" - greater than or equal to
- "gt" - greater than
- "lteq" - less than or equal to
- "lt" - less than

The operator compares each item in the selection against the value of the parameter, retaining only those items that pass the specified condition.

For example:

```
[[2]compare:number:eq[3]] returns nothing
[[2]compare:number:lt[3]] returns "2"
[[2]compare:number:eq[2]] returns "2"
```

Note that several of the variants of the **compare** operator are synonyms for existing operators, and are provided in the interests of consistency. For example, `compare:string:eq[x]` is a synonym for `match[x]`.

Examples

compare Operator (Examples)

12th April 2020 at 11:29pm

compare Operator Operator Examples

```
[[20200101]compare:date:gt[201912311852]]
```

→ compares two partial dates

Try it

```
[[202001011852]compare:integer:gt[20191231]]
```

→ compares the same two strings as integers

Try it

```
[list[Days of the Week]compare:string:gt[M]compare:string:lt[W]]
```

Try it

```
[[v5.1.23-prerelease]compare:version:gt[v5.1.22]]
```

Try it

```
[[1]compare:number:gt[2]then[yes]else[no]]
```

Try it

Compose ballad

15th November 2021 at 2:31am

task TaskManagementExample

This is a sample task for the TaskManagementExample

Concatenating text and variables using macro substitution

15th June 2023 at 1:45pm

Learning

New in v5.3.0 It is recommended to use substituted attributes or the substitute filter operator to concatenate text and variables.

It's a frequent use case in TiddlyWiki that you will want to put the results of variables together with various bits of strings of text. This process in some programming languages is often referred to as "concatenating" text.

You might, for instance want to set up a template for your customer database, where links will automatically refer to additional contact information about your customer. Inside your tiddler, you might try something like this:

⚠ Warning: Don't do it this way!	`[[Additional Info|<<currentTiddler>>-Contact]]`

But that won't work. If you try this, the link will be interpreted very literally, and will attempt to take you to:

```
<<currentTiddler>>-Contact
```

The solution is to use a macro to put the rendered value of `<<currentTiddler>>` together with the bit of additional text, `-Contact`.

Create a macro at the top of the tiddler like this:

```
\define linkup(link) [[Additional Info|$link$-Contact]]
```

You might be tempted to invoke the new macro like this:

⚠ Warning: Don't do it this way!	`<<linkup <<currentTiddler>> >>`

But if you do, you will find that `<<currentTiddler>>` doesn't get rendered, but instead gets passed literally.

Instead, you could use the MacroCallWidget widget, like this:

```
<$macrocall $name="linkup" link=<<currentTiddler>> />
```

In this case, we passed the value of a variable directly to our macro. This is often a general way to go about this task. If you wanted to create more links based on other variables you could re-use the macro for each situation.

If, as in this case, the only variable you are using is `currentTiddler` then you could write a simple macro, like this:

```
\define linkup() [[Additional Info|$(currentTiddler)$-Contact]]
```

Notice that in this case we don't pass an argument. Instead, we reference the variable using the special syntax `$(variable)$`. Since we don't pass an argument, we can invoke it without the `<$macrocall>` widget more simply, like this:

```
<<linkup>>
```

Concepts

1st December 2021 at 11:06am

Reference

These are the concepts underlying TiddlyWiki. Understanding how these ideas fit together is the key to getting the most from TiddlyWiki.

- Cascades
- ColourPalettes
- Commands
- Current Tiddler
- DataTiddlers
- Date Fields
- DefaultTiddlers
- DictionaryTiddlers
- ExternalImages

- Filters
- Hard and Soft Links
- JSONTiddlers
- Keyboard Shortcut Tiddler
- Macros
- Messages
- Modules
- PermaLinks
- Plugins
- Railroad Diagrams
- ShadowTiddlers
- Story River
- SystemTags
- SystemTiddlers
- Tagging
- TemplateTiddlers
- TextReference
- TiddlerFields
- TiddlerLinks
- Tiddlers
- TiddlyWiki
- TiddlyWiki5

- Title List
- Title Selection
- Transclusion
- Variables
- Widgets
- Wiki
- WikiText
- Coordinate Systems
- Custom Widgets
- Customizing EditTemplate field rendering
- Functions
- Pragmas
- Procedures
- Story Tiddler Template
- Temporary Tiddlers

Conditional Operators

11th July 2023 at 10:23am

Filters

Introduced in v5.1.20 The conditional filter operators allow **if-then-else** logic to be expressed within filters.

The foundation is the convention that an empty list can be used to represent the Boolean value **`false`** and a list with at one (or more) entries to

represent **`true`**.

The conditional operators are:

- then Operator replaces any input values with a constant string. For example:
  - `[[HelloThere]is[missing]then[FOO]]` evaluates to (empty)
  - `[[Missing Tiddler]is[missing]then[FOO]]` evaluates to `FOO`
- else Operator if the title list is empty then returns a constant string, otherwise returns the original title list
  - `[[HelloThere]is[tiddler]else[BAR]]` evaluates to `HelloThere`
  - `[[Missing Tiddler]is[tiddler]else[BAR]]` evaluates to `BAR`

These operators can be combined. For example:

```
[[New Tiddler]is[missing]then[I am missing]else[No I am not missing]]
```

Try it

The **`else`** operator can be used to apply a defaults for missing values. In this example, we take advantage of the fact that the **`get`** operator returns an empty list if the field or tiddler does not exist:

```
[[HelloThere]get[custom-field]else[default-value]]
```

Try it

Filter Run Prefixes

The :then and :else named filter run prefixes serve a similar purpose as the conditional operators.

Also see: Named Filter Run Prefix

Conditional Shortcut Syntax

1st September 2023 at 2:31pm

WikiText

New in v5.3.2 The conditional shortcut syntax provides a convenient way to express if-then-else logic within WikiText. It evaluates a filter expression and considers the condition to be true if there is at least one result (regardless of the value of that result).

Within an "if" or "elseif" clause, the variable `condition` contains the value of the first result of evaluating the filter condition.

A simple example:

copy to clipboard

```
<% if [{$:/info/url/protocol}match[file:]] %>
  Loaded from a file URI
<% else %>
  Not loaded from a file URI
<% endif %>
```

That renders as:

Not loaded from a file URI

One or more `<% elseif %>` clauses may be included before the `<% else %>` clause:

copy to clipboard

```
<% if [{$:/info/url/protocol}match[file:]] %>
  Loaded from a file URI
<% elseif [{$:/info/url/protocol}match[https:]] %>
  Loaded from an HTTPS URI
<% elseif [{$:/info/url/protocol}match[http:]] %>
  Loaded from an HTTP URI
<% else %>
  Loaded from an unknown protocol
<% endif %>
```

That renders as:

Loaded from an HTTPS URI

The conditional shortcut syntax can be nested:

copy to clipboard

```
\procedure test(animal)
<% if [<animal>match[Elephant]] %>
  It is an elephant
```

```
<% else %>
  <% if [<animal>match[Giraffe]] %>
    It is a giraffe
  <% else %>
    It is completely unknown
  <% endif %>
<% endif %>
\end

<<test "Giraffe">>

<<test "Elephant">>

<<test "Antelope">>
```

That renders as:

It is a giraffe It is an elephant It is completely unknown

Notes:

- Clauses are parsed in inline mode by default. Force block mode parsing by following the opening `<% if %>`, `<% elseif %>` or `<% else %>` with two line breaks
- Widgets and HTML elements must be within a single conditional clause; it is not possible to start an element in one conditional clause and end it in another

- The conditional shortcut syntax cannot contain pragmas such as procedure definitions

Configuring startup tiddlers

6th March 2018 at 5:14pm

Customise TiddlyWiki

You can configure TiddlyWiki to start up using whatever individual or group of tiddlers you want using the default tiddlers mechanism.

In the control panel under the info tab is an input field labeled "Choose which tiddlers are displayed at startup". You can list whatever tiddlers you want to open at startup. Use double square brackets for titles containing spaces. For example:

```
FirstTiddler
SecondTiddler
[[Third Tiddler]]
```

You can also use filter expressions to open more than one tiddler. For example:

```
[tag[HelloThere]]
```

will open all tiddlers tagged as HelloThere .

You can also use this technique to preserve open tiddlers at startup

See also StartupActions for controlling more advanced startup behaviours.

Configuring the default TiddlerInfo tab

12th September 2014 at 4:59pm

Customise TiddlyWiki

The configuration tiddler **$:/config/TiddlerInfo/Default** contains the title of the tiddler containing the default tiddler info tab.

The default value is `$:/core/ui/TiddlerInfo/Tools` corresponding to the **Tools** tab. Other possible values are:

- $:/core/ui/TiddlerInfo/Tools
- $:/core/ui/TiddlerInfo/References
- $:/core/ui/TiddlerInfo/Tagging
- $:/core/ui/TiddlerInfo/List
- $:/core/ui/TiddlerInfo/Listed
- $:/core/ui/TiddlerInfo/Fields
- $:/core/ui/TiddlerInfo/Advanced
- $:/editions/tw5.com/TiddlerInfo/Sources

Consent Banner Plugin

19th October 2020 at 10:46am

OfficialPlugins

The Consent Banner Plugin helps make websites that are compliant with "cookie legislation" such as the EU General Data Protection Regulation. It presents a banner inviting the user to accept or reject cookies, keeping track of their consent in local storage so that the banner

can be hidden on subsequent visits.

By default, content embedded with <iframe>, <embed> and <object> is blocked unless the user consents to accept cookies.

Consent status is available via a configuration tiddler so that it is possible to construct content that behaves differently depending upon whether consent has been granted. As an example, a macro is provided for embedding YouTube videos that automatically uses the youtube-nocookie.com variant of video URLs unless the user has accepted cookies.

Please note that using this plugin does not guarantee compliance with any particular legislation. You will need to understand the technical issues specific to your situation, and if necessary seek legal advice.

Constructing JSON tiddlers

22nd September 2023 at 2:25pm

JSON in TiddlyWiki Learning

See JSON in TiddlyWiki for an overview of using JSON in TiddlyWiki.

JSON data is just plain text, and so there are an wide variety of techniques to generate it in wikitext.

At a high level, we have several ways to generate JSON data in TiddlyWiki's own tiddler format:

- JSONTiddlerWidget
- jsontiddler Macro
- jsontiddlers Macro

When constructing JSON data manually, the jsonstringify Operator is needed to ensure that any special characters are properly escaped.

contains Operator

15th August 2018 at 2:46pm

Field Operators Filter Operators Negatable Operators

purpose	filter the input by searching list fields for a value
input	a selection of titles
suffix	**F** = the name of a field (defaults to `list`)
parameter	**S** = a possible value to be found in list field **F**
output	those input tiddlers in which the list field **F** contains the value **S**
! **output**	those input tiddlers in which the list field **F** does *NOT* contain the value **S**

Learn more about how to use Filters

Examples

contains Operator (Examples)

15th August 2018 at 2:49pm

contains Operator Operator Examples

```
[contains:tags[HelloThere]]
```

→ equivalent to [tag[HelloThere]]

Try it

```
[[$:/StoryList]contains[HelloThere]]
```

→ checks whether the tiddler 'HelloThere' is currently displayed in the story river

Try it

Contents

14th November 2021 at 2:36am

Table-of-Contents Demonstrations

This is an example tiddler. See Table-of-Contents Macros (Examples).

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

ContentType

21st February 2015 at 1:08pm

Definitions

Used in Internet protocols to indicate the type that should be used to interpret the content of a web resource.

In TiddlyWiki, the `type` field gives the content type to apply to the main `text` field.

List of Common Content Types

Group	Type	Content of `type` field
Developer	Data dictionary	application/x-tiddler-dictionary
	JavaScript code	application/javascript
	JSON data	application/json
	Static stylesheet	text/css
Image	GIF image	image/gif
	ICO format icon file	image/x-icon
	JPEG image	image/jpeg
	PDF image	application/pdf
	PNG image	image/png
	Structured Vector Graphics image	image/svg+xml
Text	HTML markup	text/html
	CSS stylesheet	text/css
	Comma-separated values	text/csv

Plain text	text/plain
TiddlyWiki 5	text/vnd.tiddlywiki
TiddlyWiki Classic	text/x-tiddlywiki

contrastcolour Macro

21st February 2015 at 4:13pm

Core Macros Macros

The **contrastcolour** macro returns whichever of two given CSS colours is deemed to contrast best with another.

An example can be seen in the **template tiddler for tag pills**.

Parameters

target

The colour to contrast against (typically a background colour)

fallbackTarget

An alternative colour to contrast against. This is used when `target` is undefined or not a valid colour

colourA

The first of the two colours to choose between

colourB

The second of the two colours to choose between

Contributing

28th March 2022 at 12:54pm

Community

Here we focus on contributions via GitHub Pull Requests but there are many other ways that anyone can help the TiddlyWiki project, such as reporting bugs or helping to improve our documentation.

Rules for Pull Requests

PRs must meet these minimum requirements before they can be considered for merging:

- The material in the PR must be free of licensing restrictions. Which means that either:
  - The author must hold the copyright in all of the material themselves
  - The material must be licensed under a license compatible with TiddlyWiki's BSD license
- The author must sign the Contributors License Agreement (see below)
- Each PR should only make a single feature change
- The title of the PR should be 50 characters or less
- The title of the PR should be capitalised, and should not end with a period
- The title of the PR should be written in the imperative mood. See below
- Adequate explanation in the body of the PR for the motivation and implementation of the change. Focus on the *why* and *what*, rather than the *how*
- PRs must be self-contained. Although they can link to material elsewhere, everything needed to understand the intention of the PR should

be included
- Any visual changes introduced by the PR should be noted and illustrated with before/after screenshots
- Documentation as appropriate for end-users or developers
- Observe the coding style
- Read the developers documentation
- Please open a consultation issue prior to investing time in making a large PR

Imperative Mood for PR Titles

The "imperative mood" means written as if giving a command or instruction. See this post for more details, but the gist is that the title of the PR should make sense when used to complete the sentence "If applied, this commit will...". So for example, these are good PR titles:

- If applied, this commit will *update the contributing guidelines*
- If applied, this commit will *change css-escape-polyfill to a $tw.utils method*
- If applied, this commit will *make it easier to subclass the wikitext parser with a custom rule set*

These a poorly worded PR titles:

- ~~If applied, this commit will *edit text widgets should use default text for missing fields*~~
- ~~If applied, this commit will *signing the CLA*~~
- ~~If applied, this commit will *don't crash if options.event is missing*~~

PR titles may also include a short prefix to indicate the subsystem to which they apply. For example:

- *Menu plugin: Include menu text in aerial rotator*

Commenting on Pull Requests

One of the principles of open source is that many pairs of eyes on the code can improve quality. So, we welcome comments and critiques of pending PRs. Conventional Comments has some techniques to help make comments as constructive and actionable as possible. Notably, they recommend prefixing a comment with a label to clarify the intention:

praise	Praises highlight something positive. Try to leave at least one of these comments per review. Do not leave false praise (which can actually be damaging). Do look for something to sincerely praise
nitpick	Nitpicks are small, trivial, but necessary changes. Distinguishing nitpick comments significantly helps direct the reader's attention to comments requiring more involvement
suggestion	Suggestions are specific requests to improve the subject under review. It is assumed that we all want to do what's best, so these comments are never dismissed as “mere suggestions”, but are taken seriously
issue	Issues represent user-facing problems. If possible, it's great to follow this kind of comment with a suggestion
question	Questions are appropriate if you have a potential concern but are not quite sure if it's relevant or not. Asking the author for clarification or investigation can lead to a quick resolution
thought	Thoughts represent an idea that popped up from reviewing. These comments are non-blocking by nature, but they are extremely valuable and can lead to more focused initiatives and mentoring opportunities
chore	Chores are simple tasks that must be done before the subject can be “officially” accepted. Usually, these comments reference some common process. Try to leave a link to the process description so that the reader knows how to resolve the chore

Contributor License Agreement

Like other OpenSource projects, TiddlyWiki5 needs a signed contributor license agreement from individual contributors. This is a legal

agreement that allows contributors to assert that they own the copyright of their contribution, and that they agree to license it to the *UnaMesa* Association (the legal entity that owns TiddlyWiki on behalf of the community).

- For individuals use: licenses/CLA-individual
- For entities use: licenses/CLA-entity

How to sign the CLA

Create a GitHub pull request to add your name to `cla-individual.md` or `cla-entity.md`, with the date in the format (YYYY/MM/DD).

step by step

1. Navigate to licenses/CLA-individual or licenses/CLA-entity according to whether you are signing as an individual or representative of an organisation
2. Ensure that the "branch" dropdown at the top left is set to `tiddlywiki-com`
3. Click the "edit" button at the top-right corner (clicking this button will fork the project so you can edit the file)
4. Add your name at the bottom
    - eg: `Jeremy Ruston, @Jermolene, 2011/11/22`
5. Below the edit box for the CLA text you should see a box labelled **Propose file change**
6. Enter a brief title to explain the change (eg, "Signing the CLA")
7. Click the green button labelled **Propose file change**
8. On the following screen, click the green button labelled **Create pull request**

The CLA documents used for this project were created using Harmony Project Templates. "HA-CLA-I-LIST Version 1.0" for "CLA-individual" and

"HA-CLA-E-LIST Version 1.0" for "CLA-entity".

ContributingTemplate

25th August 2013 at 4:49pm

Contributing to TiddlyWiki5

Here we focus on contributions via GitHub Pull Requests but there are many other ways that anyone can help the TiddlyWiki project, such as reporting bugs or helping to improve our documentation.

Rules for Pull Requests

PRs must meet these minimum requirements before they can be considered for merging:

- The material in the PR must be free of licensing restrictions. Which means that either:
  - The author must hold the copyright in all of the material themselves
  - The material must be licensed under a license compatible with TiddlyWiki's BSD license
- The author must sign the Contributors License Agreement (see below)
- Each PR should only make a single feature change
- The title of the PR should be 50 characters or less
- The title of the PR should be capitalised, and should not end with a period
- The title of the PR should be written in the imperative mood. See below

- Adequate explanation in the body of the PR for the motivation and implementation of the change. Focus on the *why* and *what*, rather than the *how*
- PRs must be self-contained. Although they can link to material elsewhere, everything needed to understand the intention of the PR should be included
- Any visual changes introduced by the PR should be noted and illustrated with before/after screenshots
- Documentation as appropriate for end-users or developers
- Observe the coding style
- Read the developers documentation
- Please open a consultation issue prior to investing time in making a large PR

Imperative Mood for PR Titles

The "imperative mood" means written as if giving a command or instruction. See this post for more details, but the gist is that the title of the PR should make sense when used to complete the sentence "If applied, this commit will...". So for example, these are good PR titles:

- If applied, this commit will *update the contributing guidelines*
- If applied, this commit will *change css-escape-polyfill to a $tw.utils method*
- If applied, this commit will *make it easier to subclass the wikitext parser with a custom rule set*

These a poorly worded PR titles:

- ~~If applied, this commit will *edit text widgets should use default text for missing fields*~~
- ~~If applied, this commit will *signing the CLA*~~
- ~~If applied, this commit will *don't crash if options.event is missing*~~

PR titles may also include a short prefix to indicate the subsystem to which they apply. For example:

- *Menu plugin: Include menu text in aerial rotator*

Commenting on Pull Requests

One of the principles of open source is that many pairs of eyes on the code can improve quality. So, we welcome comments and critiques of pending PRs. Conventional Comments has some techniques to help make comments as constructive and actionable as possible. Notably, they recommend prefixing a comment with a label to clarify the intention:

praise	Praises highlight something positive. Try to leave at least one of these comments per review. Do not leave false praise (which can actually be damaging). Do look for something to sincerely praise
nitpick	Nitpicks are small, trivial, but necessary changes. Distinguishing nitpick comments significantly helps direct the reader's attention to comments requiring more involvement
suggestion	Suggestions are specific requests to improve the subject under review. It is assumed that we all want to do what's best, so these comments are never dismissed as “mere suggestions”, but are taken seriously
issue	Issues represent user-facing problems. If possible, it's great to follow this kind of comment with a suggestion
question	Questions are appropriate if you have a potential concern but are not quite sure if it's relevant or not. Asking the author for clarification or investigation can lead to a quick resolution
thought	Thoughts represent an idea that popped up from reviewing. These comments are non-blocking by nature, but they are extremely valuable and can lead to more focused initiatives and mentoring opportunities
chore	Chores are simple tasks that must be done before the subject can be “officially” accepted. Usually, these comments reference some common process. Try to leave a link to the process description so that the reader knows how to resolve the chore

Contributor License Agreement

Like other OpenSource projects, TiddlyWiki5 needs a signed contributor license agreement from individual contributors. This is a legal agreement that allows contributors to assert that they own the copyright of their contribution, and that they agree to license it to the *UnaMesa* Association (the legal entity that owns TiddlyWiki on behalf of the community).

- For individuals use: licenses/CLA-individual
- For entities use: licenses/CLA-entity

How to sign the CLA

Create a GitHub pull request to add your name to `cla-individual.md` or `cla-entity.md`, with the date in the format (YYYY/MM/DD).

step by step

1. Navigate to licenses/CLA-individual or licenses/CLA-entity according to whether you are signing as an individual or representative of an organisation
2. Ensure that the "branch" dropdown at the top left is set to `tiddlywiki-com`
3. Click the "edit" button at the top-right corner (clicking this button will fork the project so you can edit the file)
4. Add your name at the bottom
    - eg: `Jeremy Ruston, @Jermolene, 2011/11/22`
5. Below the edit box for the CLA text you should see a box labelled **Propose file change**
6. Enter a brief title to explain the change (eg, "Signing the CLA")
7. Click the green button labelled **Propose file change**

8. On the following screen, click the green button labelled **Create pull request**

The CLA documents used for this project were created using Harmony Project Templates. "HA-CLA-I-LIST Version 1.0" for "CLA-individual" and "HA-CLA-E-LIST Version 1.0" for "CLA-entity".

This file was automatically generated by TiddlyWiki5

Contributor License Agreement

26th February 2022 at 6:55pm

Like other OpenSource projects, TiddlyWiki5 needs a signed contributor license agreement from individual contributors. This is a legal agreement that allows contributors to assert that they own the copyright of their contribution, and that they agree to license it to the *UnaMesa* Association (the legal entity that owns TiddlyWiki on behalf of the community).

- For individuals use: licenses/CLA-individual
- For entities use: licenses/CLA-entity

Contributors

16th September 2014 at 3:31pm

About

The following individuals have generously given their time to contribute to the development of TiddlyWiki:

- Jeremy Ruston (@Jermolene)

- Dave Gifford (@giffmex)
- Bram Chen (@BramChen)
- Mario Pietsch (@pmario)
- Xavier Cazin (@xcazin)
- Eric Shulman (@ericshulman)
- Stephan Hradek (@Skeeve)
- BuggyJef (@buggyj)
- Nathan Cain (@natecain)
- Ton Gerner (@gernert)
- Devin Weaver (@sukima)
- João Bolila (@jbolila)
- David Johnston (@Brennall)
- David Jade (@davidjade)
- Bob Robison (@grayeul)
- @nameanyone
- Maurizio Pollio (@mpollio)
- Ogoshi Masayuki (@ogoshima)
- Chris Sugden (@csugden)
- @asampal
- Bow Yi Jang (@Eucaly)
- Mal Gamble (@malgam)
- Michael Fogleman (@mwfogleman)

- Daniel Barrett (@shendaras)
- Simon Baird (@simonbaird)
- James Anderson (@welford)
- Jon Lister (@jayfresh)
- @TheDiveO
- Danielo Rodríguez (@danielo515)
- @willover
- @slotik
- Irene Knapp (@IreneKnapp)
- Brian Shaver (@shakerlxxv)
- Tobi Beer (@tobibeer)
- Craig Cook (@BoyCook)

Coordinate Systems

10th August 2022 at 10:16pm

Concepts

TiddlyWiki (primarily the RevealWidget) supports two coordinate systems for positioning popups (see PopupMechanism to learn more about popups).

ⓘ **Introduced in v5.2.4** We introduced absolute coordinates that may not work with all extensions and plugins. For maximum backwards compatibility, use absolute coordinates only where necessary.

Relative coordinate system

The default coordinate system is relative to the nearest positioned ancestor element. This is either:

- an element with a non-static position, or
- a **td**, **th**, **table** in case the element itself is static positioned.

For tiddlers the nearest positioned ancestor element mostly is the body of the tiddler. Read the next chapter to learn about the exceptions.

Relative coordinates are expressed in the form **(x,y,w,h)**. Where **x** and **y** represent the position and **w** and **h** the width and height of the element.

Absolute coordinate system

The relative coordinate system works flawless most of the time. Problems occure if the target element (for example, a popup) and the source element (the triggering button) do not share the same positioned ancestor element. This is often the case if the popup is declared outside a table and the triggering button is declared within a table cell. In this case the coordinate systems have different origins and the popup will be displayed in the wrong location.

Absolute coordinates can fix this problem by using the root element of the page (the upper-left corner of the page) as the origin of the coordinate system. Absolute coordinates are expressed in the form **@(x,y,w,h)**. Where **x** and **y** represent the position and **w** and **h** the width and height of the element. The leading @-symbol marks these coordinates as absolute.

The ButtonWidget has an option (**popupAbsCoords**) to put absolute coordinates into the state tiddler. The DraggableWidget and the EventCatcherWidget provide the absolute coordinate of an event within the attribute `tv-popup-abs-coords`.

Example

The following example shows a popup that is triggered from within a table cell. The table cell is the nearest positioned ancestor element. The popup was defined outside the table cell. The button using relative coordinates will open the popup in the wrong location because the button and the popup do not agree on the same coordinate system. Using absolute coordinates fixes this problem.

copy to clipboard

```
<$reveal type="popup" state="$:/state/CoordinateSampleReveal">
<div class="tc-drop-down">
Popup
</div>
</$reveal>

| Table Row 1 |<$button popup="$:/state/CoordinateSampleReveal">Relative coordinates</$button>|
| Table Row 2 |<$button popup="$:/state/CoordinateSampleReveal" popupAbsCoords="yes">Absolute
coordinates</$button>|
```

That renders as:

Table Row 1	Relative coordinates
Table Row 2	Absolute coordinates

copy-to-clipboard Macro

16th December 2017 at 11:49am

Core Macros Macros

The **copy-to-clipboard** macro displays a button that copies specified text to the clipboard. A notification is displayed if the operation is successful; some browsers do not permit the operation.

Parameters

src
: The text to be copied to the clipboard

class
: Optional CSS classes to be assigned to the button (defaults to `tc-btn-invisible`)

style
: Optional CSS styles to be assigned to the button

Examples

copy-to-clipboard Macro (Examples)

16th December 2017 at 11:51am

copy-to-clipboard Macro Macro Examples

copy to clipboard

```
<<copy-to-clipboard "Mary had a little lamb">>
```

Try it

copy to clipboard

```
<$macrocall $name="copy-to-clipboard" src={{$:/SiteTitle}}/>
```

Try it

Copying tiddlers between TiddlyWiki files

19th September 2014 at 6:15pm

Learning

You can copy an individual tiddler from one TiddlyWiki file to another by dragging a link to the tiddler from one browser window to another.

If you want to drag a link, first move it vertically, because horizontal movement is recognized by the browser as text selection.

Core Icons

23rd April 2023 at 12:31pm

Features

Introduction

TiddlyWiki includes over 100 custom vector icons. They feature in the user interface and are also available for authors to use in their own applications. See Icon Gallery for a complete listing.

Usage

The core icons are used by transcluding them. For example:

copy to clipboard

```
{{$:/core/images/new-image-button}}
```

That renders as:



The core icons are parameterised. The first parameter `size` specified the size at which the icon should be rendered:

copy to clipboard

```
{{$:/core/images/picture|64px}}
<$transclude $tiddler="$:/core/images/picture" size="32px"/>
```

That renders as:





Here is an example of dynamically resizing icons:

copy to clipboard

```
<$list filter="[range[24,56,8]]" variable=iconSize>
<$text text={{{ [<iconSize>addsuffix[px]] }}} />
<$transclude $tiddler="$:/core/icon" size=<<iconSize>>/>
</$list>
```

That renders as:

24px  32px  40px  48px  56px 

Some icons take further parameters to customise how they are rendered. For example, the **$:/core/images/new-journal-button** icon takes an additional parameter `day` that specifies the day of the month that should be shown on the calendar. It defaults to the current date if not specified

copy to clipboard

```
{{$:/core/images/new-journal-button|48px|17}}
<$transclude $tiddler="$:/core/images/new-journal-button" day="17"/>
```

That renders as:





The core icons are implemented as embedded SVG elements, and not as full-blown SVG images. This means that they can be styled using CSS. For example, the CSS property `fill` can be used to change the colour of the icons. For example:

copy to clipboard

```
<span style="fill: red;">{{$:/core/images/opacity}}</span>
```

That renders as:



Core Macros

21st February 2015 at 11:20pm

The following macros are built into TiddlyWiki's core:

- changecount
- colour
- colour-picker
- contrastcolour
- copy-to-clipboard
- csvtiddlers
- datauri
- dumpvariables
- image-picker
- jsontiddler
- jsontiddlers
- keyboard-driven-input Macro
- lingo
- list-links
- list-links-draggable
- list-tagged-draggable
- list-thumbnails
- makedatauri
- now
- qualify
- resolvepath
- Stylesheet Macros
- Table-of-Contents Macros
- tabs
- tag
- tag-picker
- tag-pill
- thumbnail
- timeline
- translink
- tree
- unusedtitle
- version

Core Variables

28th February 2015 at 2:41pm

The following variables are built into TiddlyWiki's core:

- actionTiddler Variable
- currentTiddler
- modifier Variable
- namespace
- storyTiddler
- thisTiddler
- transclusion
- tv-history-list Variable
- tv-show-missing-links Variable
- tv-story-list Variable
- tv-tiddler-preview

The core will also use various configuration variables and macros if you define them:

- tv-adjust-heading-level
- tv-auto-open-on-import
- tv-config-static
- tv-config-toolbar-class
- tv-config-toolbar-icons
- tv-config-toolbar-text
- tv-filter-export-link
- tv-get-export-image-link
- tv-get-export-link

- tv-get-export-path
- tv-wikilink-template
- tv-wikilink-tooltip
- tv-wikilinks

cos Operator

22nd October 2021 at 12:29am

Filter Operators Mathematics Operators Trigonometric Operators Unary Mathematics Operators

purpose	calculate the cosine value of a list of angles (given in radians)
input	a selection of titles
output	the cosine of the input angles (numeric value between -1 and 1)

Learn more about how to use Filters

ⓘ **Introduced in v5.1.21** See Mathematics Operators for an overview.

Examples

cos Operator (Examples)

20th October 2021 at 4:21pm

cos Operator Operator Examples

```
[[2]cos[]]
```

Try it

```
=1 =2 =3 =4 +[cos[]]
```

Try it

count Operator

8th February 2017 at 9:09pm

Filter Operators Group Operators

purpose	count the number of entries in a list
input	a selection of titles
output	a new list containing the number of items in the input list as a decimal string

Learn more about how to use Filters

The number of the entries in the input list is counted and a decimal representation returned as a single string.

Examples

count Operator (Examples)

8th February 2017 at 9:10pm

each Operator Operator Examples

```
[tag[HelloThere]count[]]
```

Try it

CountWidget

31st December 2018 at 2:01pm

Widgets

Introduction

The count widget displays the number of unique items in the output of specified filter expressions.

Content and Attributes

The content of the `<$count>` widget is ignored.

Attribute	Description
filter	The filter expression to count

Example

copy to clipboard

```
There are <$count filter="[tag[Examples]]"/> tiddlers tagged with Examples
```

That renders as:

There are 8 tiddlers tagged with Examples

Creating a custom export format

24th November 2014 at 6:35pm

Customise TiddlyWiki

To create a custom export format that exports tiddlers as their raw body text:

1. Create a tiddler with the following fields:
    - **tags**: *$:/tags/Exporter*
    - **description**: Description of this exporter
    - **extension**: Default file extension for this export format (including the dot; for example `.tid`)

2. Set the following content:

```
\define renderContent()
{{{ $(exportFilter)$ ||$:/core/templates/plain-text-tiddler}}}
\end
<<renderContent>>
```

The variable `exportFilter` contains a filter defining which tiddlers should be exported.

Creating a splash screen

27th September 2018 at 10:44am

Customise TiddlyWiki

By default, TiddlyWiki displays a blank screen while it is loading. You can add a special "splash screen" that is displayed while the wiki loads.

Larger TiddlyWiki files and those loaded over a slow network connection may take a little time to load. Once fully loaded, performance improves, because everything is now running entirely within the browser. Using a splash screen ensures people know the loading process is taking place, reducing the chance they will leave the page.

In order for the splash screen to be displayed before TiddlyWiki is initialised or decrypted, it is embedded as static HTML/CSS within the TiddlyWiki HTML file. This is done with the SystemTag: $:/tags/RawMarkupWikified/TopBody, or any of the other system tags beginning with *$:/tags/RawMarkup.*

In order to remove the splash screen when the wiki has finished loading, the HTML should be wrapped in a container with the special class `tc-remove-when-wiki-loaded`. Any DOM elements with this class are automatically deleted by the core once the wiki has loaded.

Here's an example of a simple splash screen that just displays the text *Loading*. To use it, copy the text into a new tiddler, give it the type "text/plain" and the tag "*$:/tags/RawMarkupWikified/TopBody*":

```
<div class="tc-remove-when-wiki-loaded">
Loading...
</div>
```

If the tiddler is given the WikiText type `text/vnd.tiddlywiki` then the contents are wikified as the file is saved, allowing transclusion etc. For example, here's an example that quotes the site title in the loading message. To use it, copy the text into a new tiddler, give it the type "text/vnd.tiddlywiki" and the tag "*$:/tags/RawMarkupWikified/TopBody*":

```
\rules only filteredtranscludeinline transcludeinline
<div class="tc-remove-when-wiki-loaded">
Please wait while {{$:/SiteTitle}} is loading
</div>
```

The `\rules` directive is used to limit the wikitext syntax that is recognised to the two forms of inline transclusion. This avoids accidental wikification of tiddler content.

There is a more complex example splash screen using CSS animations and images configured in this wiki: see $:/SplashScreen

Creating and editing tiddlers

7th June 2016 at 5:37pm

Working with TiddlyWiki

Creating tiddlers

You create a tiddler either by clicking the + button in the sidebar, or by clicking on a link to a missing tiddler. Links to missing tiddlers are shown in *blue italics*.

See also:

- Creating journal tiddlers

Editing tiddlers

To edit an existing tiddler, click the button at the top right of the tiddler.

Draft mode

When you create a new tiddler or edit an existing one, the tiddler will go into draft mode. This presents a control panel for modifying the tiddler in various ways. It has several parts, from top to bottom:

- **The title field** - Use this to change the title of the tiddler
- **The tag selector** - Use this to add or remove tags. As you type a tag name in the box, a dropdown list will show you any existing tags that match. You can pick from this list or create a completely new tag. Then click the **add** button or hit the `Enter` key to add the tag to the tiddler. Each tag is shown as a coloured pill. Click the "×" on a pill to remove that tag
- **The text area** - Use this to edit the main content of the tiddler. Click the **preview** button (/) to see what your changes will look like

- **The type selector** - Use this when a tiddler needs to be displayed in a special way, such as an image. See ContentType for a list of the options. The default is `text/vnd.tiddlywiki`, which means the tiddler contains WikiText
- **The field selector** - Use this to add or remove fields on the tiddler. For example, if you are editing a tiddler that's being used to tag other tiddlers, you can add a "list" field to change the order in which those tiddlers will be listed

Save, cancel or delete

When you have finished editing, click a button at the top right of the tiddler:

- The **ok** button (✓) stores your changes to this one tiddler and leaves draft mode. If your wiki is configured to AutoSave, your changes will be permanently saved. Otherwise they will only be stored temporarily in your web browser, and you will lose them if you close your TiddlyWiki page without first clicking the master **save changes** button (⊙) in the sidebar.
- The **cancel** button (⊗) discards your changes (after asking you to confirm) and leaves draft mode.
- The **delete** button (🗑) deletes the entire tiddler (after asking you to confirm).

Creating journal tiddlers

18th December 2018 at 7:00pm

Working with TiddlyWiki

Introduction

Journal tiddlers are tiddlers that use a date and/or time as their title. They are typically used as a quick way to record time-stamped information.

You can use additional tags on a journal tiddler to link it to other tiddlers, helping to establish the relationships between items of information.

For example, you might use a journal tiddler called **10th October 2014** to record thoughts and information captured on that particular day. The tags **Shopping** and **London** might be used to indicate that the entry concerns shopping in London.

Creating a journal tiddler

The easiest way to create a journal tiddler is to use the **new journal** button in the **Tools** tab of the sidebar. If you find yourself often using the button, click the checkbox next to it to make the button available just above the search box.

The **new journal** button creates a journal entry as a blank tiddler with the tag **Journal** and a title derived from today's date. If a journal tiddler with that title already exists, then this is opened for editing.

Creating a tagged journal tiddler

A common sequence of actions is to create (or reopen) today's journal entry and tag it with the title of another tiddler. This can be done with the **new journal here** button in the other tiddler's toolbar. You can find this button in the **Tools** tab of the tiddler's InfoPanel.

For example, you might be reviewing a tiddler called **Oxford Street** and realise that it's relevant for planning your shopping trip. Click the **new journal here** button on the **Oxford Street** tiddler to bring up a journal entry tagged with **Oxford Street**.

Customising journal tiddlers

To configure how new journal entries are created, visit the **Basics** tab under **Info** in the **control panel** ⚙:

- "Title of new journal tiddlers" specifies how these tiddlers should be named, as a date format string. The default setting of `DDth MMM YYYY` causes new entries to have titles of the form "10th October 2014"
- "Tags for new journal tiddlers" [specifies|Title List] tags that will automatically appear on new journal entries. For example: `Journal [[Summer vacation]]`

Hint: if you want to create a separate journal tiddler whenever you click **new journal** (even if you do this several times in the same day), you can include the clock time in the title format. Specify something like `YYYY-0MM-0DD at 0hhh0mm'0ss''` as the date format.

Another useful trick is to include `<<currentTiddler>>` somewhere in the title format. This means that if you click **new journal here** on several different tiddlers, the title of each of those tiddlers will form part of the name of the resulting journal entries.

Making a custom new journal button

Visit the Making a custom journal button tiddler for instructions on how to make your own custom New Journal button

Creating new toolbar buttons

26th November 2021 at 5:29pm

Customise TiddlyWiki

Let's say you have a skeleton tiddler called 'Recipe template', and you want to have a button available in the tiddler *ViewToolbar* to create new recipe tiddlers on demand. This will require the following steps:

1. You will want an image for your button. If none of the core images (shadow tiddlers with the prefix *$:/core/images/*) work for you, then you will need to create or acquire an SVG image (for example, one of the images at http://flaticon.com), drag it into your file so that it becomes a tiddler, edit the tiddler and adjust the height and width to 22px
2. You will want to create the tiddler that contains your tiddler. Create it, title it, and add the button code (see the code at the bottom of this tiddler for an example, with hints where you will need to adapt it). Tag it **$:/tags/ViewToolbar**
3. You will need to create a tiddler that tells TiddlyWiki whether your button should be visible in the toolbar or hidden. Let's title it *$:/config/ViewToolbarButtons/Visibility/Recipe*. Type `show` into the text area, and save. If you want to hide it, type `hide` into the text area and save. The button will also be accessable from the *ControlPanel* **: Appearance : Toolbars :** *ViewToolbar* tab
4. You will want to position the button properly. Open the tiddler **$:/tags/ViewToolbar** and insert your button tiddler's title in the appropriate place in the list field.

```
\define newHereButtonTags()
[[$(currentTiddler)$]]
\end
\define newHereButton()
<$button class=<<tv-config-toolbar-class>>>
<$action-sendmessage
  $message="tm-new-tiddler"
$param="TITLE OF YOUR SKELETON BUTTON"
title="New tiddler"
  tags=<<newHereButtonTags>> />
<$list filter="[<tv-config-toolbar-icons>match[yes]]">
{{TITLE OF YOUR SVG IMAGE TIDDLER}}
</$list>
<$list filter="[<tv-config-toolbar-text>match[yes]]">
```

```
<span class="tc-btn-text"><$text text="CAPTION FOR YOUR BUTTON"/></span>
</$list>
</$button>
\end

<<newHereButton>>
```

Creating SubStories

27th November 2021 at 4:22am

Learning

This example shows how to create a sub-story within a tiddler that is independent of the main story. The NavigatorWidget and ListWidget work together to:

1. Add tiddlers to the story when links are clicked. Navigate to the tiddler corresponding to the link click. (**story** attribute of the NavigatorWidget)
2. Remove tiddlers from the story when tiddler close buttons are clicked. (**story** attribute of NavigatorWidget)
3. Animate the opening and closing of tiddlers. (**storyview** attribute of the ListWidget)
4. Display the list of open tiddlers. (**filter** and **template** attributes of the ListWidget)
5. Display tiddlers in draft mode. (**editTemplate** attribute of the ListWidget)
6. Track navigation history. (**history** attribute of both NavigatorWidget and ListWidget)
7. And more!

Here is the example code (see also StateMechanism for related discussion):

```
<$navigator story="$:/temp/DemoStoryList" history="$:/temp/DemoHistoryList">

{{$:/core/ui/SideBarSegments/search}}

<$list
  filter="[list[$:/temp/DemoStoryList]]"
  history="$:/temp/DemoHistoryList"
  template={{$:/config/ui/ViewTemplate}}
  editTemplate={{$:/config/ui/EditTemplate}}
  storyview="classic"
  emptyMessage="Use search box above. Clicked search links will open here instead of in the main
story.<p>No search ideas? Click this link to get started: HelloThere</p>"/>

</$navigator>
```

That renders as:

Use search box above. Clicked search links will open here instead of in the main story.

No search ideas? Click this link to get started: HelloThere

csvtiddlers Macro

21st February 2015 at 4:13pm

Core Macros Macros

The **csvtiddlers** macro returns the fields of a selection of tiddlers in CSV form, with one record (row) per tiddler.

An example can be seen in the **template tiddler for CSV exports**.

Parameters

filter

A filter selecting which tiddlers to include

format

Reserved for future extension. Should be set to `quoted-comma-sep`

Current Tiddler

21st February 2015 at 10:50pm

Concepts

The **current tiddler** provides the context in which several aspects of WikiText are interpreted.

For example, `{{!!title}}` denotes the value of the **`title`** field of whatever the current tiddler happens to be. This technique can be used to create general-purpose template tiddlers.

The title of the current tiddler can always be found in the **`currentTiddler`** variable.

The two most common ways in which the current tiddler is changed are:

- the **$tiddler** widget
- the **$list** widget (when its **variable** attribute is not overridden)

The current tiddler is therefore often *NOT* the same as the tiddler that is being viewed or edited.

currentTiddler Variable

21st February 2015 at 4:20pm

Core Variables Variables

The **currentTiddler** variable contains the title of the current tiddler.

Several aspects of WikiText use this tiddler as their context. As a result, within a **$list** widget or a template tiddler, there is often no need to explicitly specify a tiddler title.

Compare **storyTiddler**.

Examples

currentTiddler Variable (Examples)

21st February 2015 at 3:13pm

currentTiddler Variable Variable Examples

The following two examples have the same meaning:

copy to clipboard

```
<$view field=title/>
```

Try it

copy to clipboard

```
<$view tiddler=<<currentTiddler>> field=title/>
```

Try it

This next example shows how the **$list** widget changes the current tiddler:

copy to clipboard

```
<ol>
<$list filter="[prefix[J]]">
<li><<currentTiddler>></li>
</$list>
</ol>
```

Try it

Custom data-styles

11th April 2018 at 7:39pm

$:/tags/Stylesheet data-tags-styles How to apply custom styles

```
[data-tiddler-title="Custom styles by data-tiddler-title"] {
  border: 1px solid blue;
}[data-tags*="example-test"] {
  border: 2px solid pink;
}[data-tags*="example-hardlinebreaks"] .tc-tiddler-body {
  word-break: normal;
  word-wrap: break-word;
  white-space: pre-wrap;
}[data-tags*="data-tags-styles"] .tc-tiddler-body {
  display: block;
  padding: 14px;
  margin-top: 1em;
  margin-bottom: 1em;
  word-break: normal;
  word-wrap: break-word;
  white-space: pre-wrap;
  background-color: #f5f5f5;
  border: 1px solid #cccccc;
  padding: 0 3px 2px;
```

```
  border-radius: 3px;
  font-family: Monaco, Consolas, "Lucida Console", "DejaVu Sans Mono", monospace;
}
```

Custom styles by data-tags

11th April 2018 at 7:39pm

example-test How to apply custom styles

Attribute: data-tags

Introduced in v5.1.16

The TiddlyWiki core adds several `attributes` to the rendered content. These make it possible to apply custom styles to tiddlers.

For example this tiddler is tagged: How to apply custom styles and example-test so the attribute looks like this:

```
data-tags="[[How to apply custom styles]] example-test"
```

Important: Tiddler tags are **not** sorted so the order in the rendered output may be different!

Examples

The following CSS is defined in Custom data-styles and creates a pink border for all tiddlers (including this one) tagged with `example-test`.

```
[data-tags*="example-test"] {
  border: 2px solid pink;
}
```

Styling Stylesheets

So to display tiddlers tagged: `data-tags-styles` in a decent way we can use the following code. (I could have used: `$:/tags/Stylesheet`, but that would affect all stylesheets in this wiki, which is not intended. amt ;)

Important: Don't forget to also specify `.tc-tiddler-body` or the whole tiddler, including the title, will be changed! see: Custom data-styles

```
[data-tags*="data-tags-styles"] .tc-tiddler-body {
  display: block;
  padding: 14px;
  margin-top: 1em;
  margin-bottom: 1em;
  word-break: normal;
  word-wrap: break-word;
  white-space: pre-wrap;
  background-color: #f5f5f5;
  border: 1px solid #cccccc;
  padding: 0 3px 2px;
  border-radius: 3px;
  font-family: Monaco, Consolas, "Lucida Console", "DejaVu Sans Mono", monospace;
}
```

Hard Linebreaks

This mechanism may be handy for users who want to write prose text! See: Hard Linebreaks with CSS

More Possibilities

[attr]
Represents an element with an attribute name of attr.

[attr="value"]
Represents an element with an attribute name of attr and whose value is exactly "value".

[attr~="value"]
Represents an element with an attribute name of attr whose value is a whitespace-separated list of words, one of which is exactly "value".

[attr|="value"]
Represents an element with an attribute name of attr. Its value can be exactly “value” or can begin with “value” immediately followed by “-” (U+002D). It can be used for language subcode matches.

[attr^="value"]
Represents an element with an attribute name of attr and whose first value is prefixed by "value".

[attr$="value"]
Represents an element with an attribute name of attr and whose last value is suffixed by "value".

[attr*="value"]
Represents an element with an attribute name of attr and whose value contains at least one occurrence of string "value" as substring.

[attr "operator value" i]
Adding an i (or I) before the closing bracket causes the value to be compared case-insensitively (for characters within the ASCII range).

Learn more at: Attribute selectors - CSS or CSS-Specification

Custom styles by data-tiddler-title

11th April 2018 at 7:39pm

How to apply custom styles

Attribute: data-tiddler-title

Introduced in v5.1.16

The TiddlyWiki core adds several `attributes` to the rendered content. With those attributes it's possible to apply custom styles to the tiddler content.

For example this tiddler is named: "Custom styles by data-tiddler-title" so the attribute looks like this:

```
data-tiddler-title="Custom styles by data-tiddler-title"
```

Examples

The following CSS is defined in Custom data-styles and creates a blue border for exactly this tiddler.

```
[data-tiddler-title="Custom styles by data-tiddler-title"] {
  border: 1px solid blue;
}
```

To create a green border for every tiddler that starts with `$:/` aka system tiddlers, you'd need to use CSS like so: (not applied here but you can experiment with it!)

```
[data-tiddler-title^="$:/"] {
  border: 1px solid green;
}
```

More Possibilities

[attr]
: Represents an element with an attribute name of attr.

[attr="value"]
: Represents an element with an attribute name of attr and whose value is exactly "value".

[attr~="value"]
: Represents an element with an attribute name of attr whose value is a whitespace-separated list of words, one of which is exactly "value".

[attr|="value"]
: Represents an element with an attribute name of attr. Its value can be exactly “value” or can begin with “value” immediately followed by “-” (U+002D). It can be used for language subcode matches.

[attr^="value"]
: Represents an element with an attribute name of attr and whose first value is prefixed by "value".

[attr$="value"]
: Represents an element with an attribute name of attr and whose last value is suffixed by "value".

[attr*="value"]

Represents an element with an attribute name of attr and whose value contains at least one occurrence of string "value" as substring.

[attr "operator value" i]

Adding an i (or I) before the closing bracket causes the value to be compared case-insensitively (for characters within the ASCII range).

Learn more at: Attribute selectors - CSS or CSS-Specification

Custom styles by user-class

11th April 2018 at 7:39pm

How to apply custom styles

Tiddler Field: `class` **Introduced in v5.1.16**

The **tag manager** allows us to set a tiddler `color` field, that is used to define the "tag-pill" colour. Since: **Introduced in v5.1.16** we can define a `class` field, that is directly inserted into the **ViewTemplate** and it can be used for styling:

title: `anyName`
tags: `$:/tags/Stylesheet`
class: `myClass`

Every tiddler, that has a `class` field can be styled that way!

```
.myClass {
  border: 2px solid blue;
}
```

Learn more at: How to apply custom styles

Custom Styles FAQ

11th April 2018 at 7:39pm

FAQ-group How to apply custom styles

ⓘ **Introduced in v5.1.16**

- Q: How can I style a tiddler if it has "this" AND "that" tag?
- Q: How can I style a tiddler if it has "this" OR "that" tag?
- Q: How can I use a custom field to style a tiddler?
- Q: Is there a way to create dynamic stylesheets?
- Q: What if a tiddler has no tags?

Custom tag pill styles

8th June 2023 at 2:34pm

How to apply custom styles

Attribute: data-tag-title

ⓘ **Introduced in v5.2.0** The attribute **`data-tag-title`** was added to tag pills visible in the tiddler view template.

ⓘ **New in v5.3.0** The attribute was added to every tag pill visible in the standard TiddlyWiki UI. Especially the edit template tag list, the tag-picker

dropdown, the Right sidebar -> More -> Tags tab and the **$:/TagManager**

The **`data-tag-title`** HTML attribute only contains the tag-title visible in the tag pill. It can be used to style the tag-pill.

If you want to style the whole tiddler have a look at: Custom styles by data-tiddler-title

Examples

If you use the following CSS in a new tiddler tagged: `$:/tags/Stylesheet` every tag that starts with a `#` will have a new border radius. So those tags stand out in contrast to the default tags.

You have to define both CSS rules, due to the existing UI structure to catch all tag-pills in the existing TW UI.

```
[data-tag-title^="#"] .tc-tag-label,
[data-tag-title^="#"].tc-tag-label {
   border-radius: 3px;
}
```

More Possibilities

[attr]
: Represents an element with an attribute name of attr.

[attr="value"]
: Represents an element with an attribute name of attr and whose value is exactly "value".

[attr~="value"]

Represents an element with an attribute name of attr whose value is a whitespace-separated list of words, one of which is exactly "value".

[attr|="value"]

Represents an element with an attribute name of attr. Its value can be exactly “value” or can begin with “value” immediately followed by “-” (U+002D). It can be used for language subcode matches.

[attr^="value"]

Represents an element with an attribute name of attr and whose first value is prefixed by "value".

[attr$="value"]

Represents an element with an attribute name of attr and whose last value is suffixed by "value".

[attr*="value"]

Represents an element with an attribute name of attr and whose value contains at least one occurrence of string "value" as substring.

[attr "operator value" i]

Adding an i (or I) before the closing bracket causes the value to be compared case-insensitively (for characters within the ASCII range).

Learn more at: Attribute selectors - CSS or CSS-Specification

Custom Widgets

21st September 2023 at 8:03pm

Concepts Reference

Introduction

ⓘ **New in v5.3.0** A **custom widget** is a special kind of procedure that can be called using the same syntax as widgets.

Custom widgets can also be used to override built-in JavaScript widgets to customise their behaviour.

Defining Custom Widgets

Custom widgets are usually defined with the Pragma: \widget:

```
\widget $my.widget(attribute:"Default value")
This is the widget, and the attribute is <<attribute>>.
\end
```

The name of the widget must start with a dollar sign. If it is a user defined widget that does not override an existing widget then it must include at least one period (dot) within the name (for example `$my.widget` or `$acme.logger`).

Note that the Pragma: \whitespace setting is inherited from the parsing context in which the procedure definition occurs. That means that a tiddler containing multiple procedure definitions only needs a single whitespace pragma at the top of the tiddler, and the setting will be automatically inherited by the procedure definitions without needing the pragma to be repeated.

Using Custom Widgets

Custom widgets are called in the same way as ordinary built-in widgets:

```
<$my.widget/>

<$my.widget attribute="The parameter"/>
```

The attributes that are specified in the widget call are made available as parameter variables.

Accessing Content of Custom Widgets

Within the definition of a custom widget the content of the calling widget is available via the `<$slot $name="ts-raw"/>` widget. The contents of the **`$slot`** widget is used as the default content if the widget was called without any content.

For example:

copy to clipboard

```
\widget $my.widget(one:'Jaguar')
<$text text=<<one>>/>
<$slot $name="ts-raw">
    Whale
</$slot>
\end

<$my.widget one="Dingo">
    Crocodile
</$my.widget>

<$my.widget/>
```

That renders as:

Dingo Crocodile

Jaguar Whale

How Custom Widgets Work

Custom widgets are implemented as a special kind of variable. The only thing that distinguishes them from ordinary variables is the way that they can be called as a custom widget with attributes mapped to parameters.

Overriding Core JavaScript Widgets

Custom widgets can use the **$genesis** widget to invoke the original widget, bypassing the override. For example, here we override the **$codeblock** widget to add ≤≥ symbols around each string of text.

copy to clipboard

```
\widget $codeblock(code)
<$genesis $type="$codeblock" $remappable="no" code={{{ [<code>addprefix[≤]addsuffix[≥]] }}}/>
\end

<$codeblock code="Kangaroo"/>

<$codeblock code={{$:/SiteTitle}}/>

```
Python
```

<$let test="Tiger">
<$codeblock code=<<test>>/>
</$let>
```

That renders as:

```
≤Kangaroo≥
```

```
≤TiddlyWiki≥
```

```
≤Python≥
```

```
≤Tiger≥
```

Customise TiddlyWiki

6th December 2021 at 5:22pm

TableOfContents

Information about customising TiddlyWiki. See Customising TiddlyWiki's user interface for an overview of the mechanisms employed.

- Adding a table of contents to the sidebar
- Alternative page layouts
- Configuring startup tiddlers
- Configuring the default TiddlerInfo tab
- Creating a custom export format
- Creating a splash screen
- Creating new toolbar buttons
- Customising search results
- Customising TiddlyWiki's user interface
- Hidden Settings

- How to add a new tab to the sidebar
- How to apply custom styles
- How to create a custom story tiddler template
- How to create a custom tiddler icon rule
- How to create a custom tiddler colour rule
- How to create keyboard shortcuts
- How to turn off camel case linking
- How to widen tiddlers (aka storyriver)
- Making a custom journal button
- Preserving open tiddlers at startup
- Setting a favicon
- Setting a page background image
- Using Stylesheets

Customising search results

7th December 2020 at 7:11pm

Customise TiddlyWiki

By default, the results for the sidebar search box are displayed as a simple list of tiddler titles. The search results can be customised by adding plugin visualisations that show the search results in different ways. (Tabs are shown automatically if an additional search result visualisation is detected).

Search result visualisations are stored in tiddlers tagged *$:/tags/SearchResults*. The default search result listing is implemented in the system

tiddler **$:/core/ui/DefaultSearchResultList**.

To create a new search result visualisation:

1. Create a new tiddler tagged *$:/tags/SearchResults*
2. Use the widget variable **searchTiddler** to access the title of the tiddler containing the current search term

If you'd like the new visualisation to be the default, create a tiddler called **$:/config/SearchResults/Default** containing the title of the tiddler containing the search visualisation that you want to display by default.

Here is an example of an alternative visualisation that displays results in reverse chronological order:

```
\define searchResults()
<$set name="resultCount" value="""<$count filter="[!is[system]search{$(searchTiddler)$}]"/>""">

{{$:/language/Search/Matches}}

</$set>
<<timeline subfilter:"!is[system]search{$(searchTiddler)$}">>
\end
<<searchResults>>
```

Introduced in v5.1.23 The sidebar search introduces a more sophisticated search-mechanism which makes it possible to navigate the search results using the keyboard shortcuts `Down` and `Up`. To add that mechanism to your own custom search results follow these simple steps:

1. Your tiddler tagged $:/tags/SearchResults is accessible through the **`configTiddler`** variable
2. The user-input in the search field is accessible through the **`userInput`** variable

3. Use the fields **`first-search-filter`** and **`second-search-filter`** to store the filters used for your search results. See the tiddler **$:/core/ui/DefaultSearchResultList** for details
4. Use the following form so that navigated search-results are highlighted, modify it to match your needs:

```
<$list filter="[<userInput>minlength[1]]" variable="ignore">
<$list filter={{{ [<configTiddler>get[first-search-filter]] }}}>
<span class={{{[<currentTiddler>addsuffix[-primaryList]] -[<searchListState>get[text]] +[then[]else[tc-
list-item-selected]] }}}>
<$transclude tiddler="$:/core/ui/ListItemTemplate"/>
</span>
</$list>
</$list>
```



Note that the **`searchTiddler`** variable still contains the name of the tiddler that is used for sorting the search results. The **`editTiddler`** variable contains the name of the tiddler that is being edited.

Customising Tiddler File Naming

3rd August 2021 at 10:46pm

TiddlyWiki on Node.js

By default, a TiddlyWiki on Node.js instance using a wiki folder will create new tiddler files by using the sanitised and disambiguated title as filename. All filepath operations are relative to a `default-tiddler-location` which defaults to the wiki folder's `tiddlers/` directory. This can be overridden by mapping a path in the wiki's tiddlywiki.info file, by using a `default-tiddler-location` property in the `config` object.

The default file extension of `.tid` is used for tiddlers that are missing the `type` field, or for tiddlers of type "text/vnd.tiddlywiki". Tidders of other types are saved according to their MIME types (defined at boot startup).

Both the logical path (directory and file name) and the file extension can be customised independently by creating optional tiddlers *$:/config/FileSystemPaths* and *$:/config/FileSystemExtensions*.

File System Paths

The logical path can be customised by creating a *$:/config/FileSystemPaths* tiddler containing one or more filter expressions, each on a line of its own. Every time a tiddler is saved to disk it is tested against each filter in turn, and the first output of the first filter to produce any output is taken as a logical path to be used for the tiddler file. If the logical path has changed a new file is created and the old file is deleted.

Tiddlers are limited to being written to the wiki folder, the path defined in the `default-tiddler-location` setting, or the specific path saved in the $:/config/OriginalTiddlerPaths tiddler (see tiddlywiki.files Files). Any error saving a tiddler to disk, with a logical path that does not start with the wiki folder's path the most common error, causes the filepath to be encoded via Javascript's `encodeURIComponent()` method and the tiddler is saved as this file in the wiki folder's `default-tiddler-location`.

Logical paths do not include the file-on-disk's extension (see below), and they can use `/` or `\` as directory separator (when generating the physical path, this is replaced by the correct separator for the platform TiddlyWiki is running on). If none of the filters match, the logical path is simply the title with all occurrences of the characters `/\<>~:"|?*^` replaced by `_` in order to guarantee that the resulting path is legal on all supported platforms. Logical paths are also limited to 200 characters. If a file with this name already exists, a space and a number will be appended to the final filepath, and with the number incremented until an unused path is found.

Example

```
[is[system]!has[draft.of]removeprefix[$:/]addprefix[_system/]]
[is[draft]search-replace:g:regexp[/|\\],[_]addprefix[drafts/]]
[tag[task]addprefix[mytasks/]]
[!tag[externalnote]addprefix[wiki/]]
```

Note

> All paths are relative to the wiki's `default-tiddler-location`.

This will store newly created system tiddlers that are not drafts of other tiddlers in `./_system/` (after stripping the `$:/` prefix). Next, all drafts have the path separator characters in their titles replaced by "_" and are stored in `./drafts/`. Then tiddlers tagged task are stored in a subdirectory `./mytasks/`. Finally, all tidders not tagged with "externalnote" will match the final `[!tag[externalnote]addprefix[wiki/]]` storing these in `./wiki/`. In this example, tiddlers tagged with "externalnote" have been imported using tiddlywiki.files Files with an "isEditableFile" flag set to true, causing the server to remember their original file path in the $:/config/OriginalTiddlerPaths tiddler.

Whenever a tiddler generates a *$:/config/FileSystemPaths* filter match, any `/` or `\` in the tiddler title is mapped to a path separator. With the above filters, the non-system, non-draft tiddler titled `some/thing/entirely/new` (with no tags) will be saved to `./wiki/some/thing/entirely/new.tid` (ie, the file `new.tid` in a directory called `entirely/`). Thus, *$:/config/FileSystemPaths* itself will end up in `./_system/config/FileSystemPaths.tid` or `.\_system\config\FileSystemPaths.tid`, depending on the platform.

File System Extensions

Normally, the file system extension of a tiddler on disk is determined by the presence of field values containing newlines or field values that start or end with whitespace (other than the text field), in which case the single file ".json" tiddler file format is used.

If the tiddler does not have such field values, then the `type` field is referenced to find a matching file-type, with `.tid` used for tiddlers without a type value. The boot engine defines a set of these tiddler-type to file-type relationships in the $:/boot/boot.js tiddler. Search for `// Add file extension information` to find the section of code that defines these relationships.

The file extension of individual tidders can be customised by creating a tiddler *$:/config/FileSystemExtensions* containing one or more filter expressions, each on a line of its own. Every time a tiddler is saved to disk it is tested against these filters, and the first output of the first filter to produce any output is taken as the file extension to be used for the tiddler file. Extensions should always start with a leading dot (see example). If no filter matches, the default extension is used. If the extension has changed a new file is created and the old file is deleted.

Note

> A result of ".tid" will force the tiddler to be written to disk as a single-file text tiddler. A result of ".json" will force the tiddler to be written to disk as a single file tiddler in json-format (a single tiddler fields object in an array), NOT as a tiddler of type "application/json". All other recognised file-types will be saved using their defined extension along with an accompanying *.meta file of the same name which describes all fields but the "text" field.

Example

```
[tag[.txt]then[.txt]]
[tag[.json]then[.json]]
[tag[.tid]then[.tid]]
```

This will cause all tidders that have the tag ".txt" to be saved at the filepath determined by the File System Paths filters, but with their text field saved as a *.txt file, and all other fields saved as a *.txt.meta file.

Next, all tiddlers that have the ".json" tag are saved as *.json files. Finally, all tiddlers that have tag ".tid" are saved as single files. If a tiddler matches none of the filters, the default extension determined by the tiddlers `type` field would be used.

Customising TiddlyWiki's user interface

4th December 2021 at 1:00pm

Customise TiddlyWiki

TiddlyWiki's user interface is designed to be highly extensible. Every element can be augmented, removed, or rearranged.

Several different mechanisms are used to achieve this:

- special fields
- special titles
- special tags
- cascades

Here we provide an overview of those mechanisms and how they relate together.

Special Fields

Special fields are used to assign an appearance or behaviour to individual tiddlers. They can be thought of as flags or values that directly control the tiddler to which they are applied.

For example:

- Set the `icon` field to the title of an image tiddler to be used as the icon for this tiddler
- Set the `color` field to a CSS colour that is then used for icons and tag pills associated with this tiddler
- Set the `hide-body` field to `yes` to hide the view template body for this tiddler

See TiddlerFields for details of all the special fields.

Special Titles

Certain special titles identify configuration tiddlers that customise TiddlyWiki's appearance or behaviour. They can be thought of as global settings that affect an entire wiki.

For example:

- $:/DefaultTiddlers is a filter yielding the titles of the tiddlers that should be displayed at startup
- $:/SiteTitle specifies the text of the wiki title
- **$:/config/Tags/MinLength** controls the minimum number of characters to trigger tag autocompletion

Many of TiddlyWiki's configuration tiddlers are presented as options in **$:/ControlPanel**. Less commonly used configuration tiddlers do not have a user interface, but are documented in Hidden Settings.

Special Tags

Special tags assign special behaviour or appearance to all of the tiddlers to which they are applied. They can be thought of as establishing ordered lists of tiddlers that are processed or displayed in a particular way.

For example:

- *$:/tags/Global* causes the definitions in a tiddler to be available globally
- *$:/tags/Stylesheet* causes the tiddler to be interpreted as a CSS stylesheet
- **$:/tags/SideBar** causes the tiddler to be displayed as a sidebar tab

See SystemTags for details of all the special tags.

The entire TiddlyWiki user interface is constructed from lists formed from special system tags.

The ordering of these lists is determined by the order of tagged tiddlers rules. Users can re-order tags using drag and drop within a tag dropdown.

Cascades

Cascades provide a means to select one of multiple values based on flexible, extensible criteria. They can be thought of as a list of conditions that are evaluated in turn until one of them matches.

For example, the core uses the template **$:/core/ui/ViewTemplate** to display tiddlers in view mode, and **$:/core/ui/EditTemplate** to display tiddlers in edit mode. A cascade is used to choose which template to use for a particular tiddler:

1. If the tiddler is a draft, then use **$:/core/ui/EditTemplate**
2. Otherwise, use **$:/core/ui/ViewTemplate**

The list of conditions is defined via a special tag, making it possible to insert additional conditions anywhere in the list.

For example, a plugin might add a special template *$:/plugins/example/map-template* for tiddlers that have the tag *$:/tags/Map* by inserting an additional rule:

1. If the tiddler is a draft, then use **$:/core/ui/EditTemplate**
2. If the tiddler is tagged *$:/tags/Map*, then use *$:/plugins/example/map-template*
3. Otherwise, use **$:/core/ui/ViewTemplate**

See Cascades for more details.

Customizing EditTemplate field rendering

13th April 2022 at 6:55pm

Concepts

When editing a tiddler the **EditTemplate** normally renders fields as simple input boxes. To modify this behaviour, the cascade mechanism can be used. Via the Field Editor Cascade the name of the tiddler used for rendering the field editor can be specified. The content of this tiddler is transcluded to represent the content of the field.

To modify the appearance of all fields whose name ends with `-date` create a new tiddler and add the `$:/tags/FieldEditorFilter` tag to it. Add a `list-before` field and assign the value `$:/config/FieldEditorFilters/default`. Now you have to put the filter for the cascade into the tiddler's text field: `[suffix[-date]then[$:/config/EditTemplateFields/Templates/dates]]`. This will transclude the tiddler named `$:/config/EditTemplateFields/Templates/dates` to render the input elements for all fields with names matching the regular expression.

The variables `currentTiddler`, `currentField` and `currentFieldName` are set to pass information about the tiddler and field that are edited to the transcluded tiddler.

`currentTiddler`	The tiddler that must be used to store the field value.
`currentField`	The field within the `currentTiddler` that must be used to store the field name. This is an opaque value hat may contain any field name (even `text`), use `currentFieldName` to make decisions based on the actual name of the currently edited field.
`currentFieldName`	The name of the currently edited field.

For example, a tiddler containing the following WikiText would render the field as an HTML input element of the type `date`. This will show a date picker for the fields on all modern browsers:

```
<$edit-text tiddler=<<currentTiddler>> field=<<currentField>> tag="input" type="date" class="tc-edit-texteditor tc-edit-fieldeditor" placeholder="Set your date" tabindex={{$:/config/EditTabIndex}} cancelPopups="yes"/>
```



The `currentField` variable will be set to `text` for new fields. Make sure that your editor will handle this correctly. For example, by setting the `tag` attribute on the EditTextWidget. If you want to know the name of the currently edited/added field, use the `currentFieldName` variable.

The `tabindex` and `cancelPopups` attributes make sure the HTML input element behaves exactly the default elements provided by TiddlyWiki.

Not only the EditTextWidget can be used. A tiddler containing the following WikiText will render the field as a drop-down-list that allows the user to select the name of a tiddler. The name of the selected tiddler will be stored in the field.

```
<$select tiddler=<<currentTiddler>> field=<<currentField>> class="tc-edit-texteditor tc-edit-fieldeditor" cancelPopups="yes">
    <$list filter='[all[tiddlers]sort[title]]'>
        <option value=<<currentTiddler>>><$view field='title'/></option>
    </$list>
</$select>
```

The classes `tc-edit-texteditor` and `tc-edit-fieldeditor` should be used to style the `input` and `select` elements to match the theme of the TiddlyWiki installation.

Persistence of values when creating fields

When using multiple field editors for creating fields within the **EditTemplate**, every field editor tiddler returned by the Field Editor Cascade gets its own storage tiddler. This is done to prevent problems with incompatible values when the user is switching between fields governed by different field editors.

Example

There is a cascade that returns a special field editor for all fields starting with the string "my-". All other fields use the default field editor. If you type a new value into the "field value" input box and select any field not starting with "my-", the value will be kept. If you switch to a field, that starts with "my-", the "field value" input field will be empty again because a new type of field editor is used. If you now type a value and switch to another field starting with "my-" the value will be kept. If you switch to a field that does not start with "my-" the previously typed value (that was stored for the default editor) will reappear.

cycle Operator

18th November 2020 at 8:21pm

Filter Operators Listops Operators Order Operators

purpose	toggle the titles specified in the first parameter in a cyclical manner
input	a list of items
parameter	the **`cycle`** operator accepts 1 or 2 parameters, see below for details
output	the input list with the titles specified in the parameter toggled in a cyclical manner

Learn more about how to use Filters

Introduced in v5.1.23

The **cycle** operator requires at least one parameter.

```
[cycle[<titles>],[step-size]]
```

- **titles** : a title list to toggle in the input list cyclically. If no title from the parameter is present in the input, the first title is added. If a title from the parameter is present in the input, it is replaced with the next title from the parameter. Note that all titles specified in this parameter should be unique.
- **step-size**: (optional). Defaults to 1. Specifies the number of steps in the parameter list to move each time. Can be a negative number.



While the **cycle** operator interprets its first parameter as a list of titles to cycle through, the toggle Operator accepts an unlimited number of distinct parameters and offers similar functionality.

Examples

cycle Operator (Examples)

18th November 2020 at 6:54pm

cycle Operator Operator Examples

Cycle through a list of values to add as a tag:

```
<$action-listops $tiddler="target" $tags="+[cycle[todo soon now maybe done ]]" />
```

Cycle through a list of values to add as a tag, in reverse order:

```
<$action-listops $tiddler="target" $tags="+[cycle[todo soon now maybe done ],[-1]]" />
```

These examples make use of the Days of the Week tiddler.

```
[list[Days of the Week]first[]] +[cycle{Days of the Week!!list}]
```

Try it

```
[list[Days of the Week]first[]] +[cycle{Days of the Week!!list},[2]]
```

Try it

D3 Plugin

28th February 2017 at 11:25am

OfficialPlugins Plugin Editions

The D3 plugin integrates the D3 visualisation library with TiddlyWiki.

See https://tiddlywiki.com/plugins/tiddlywiki/d3/

See the demo at https://tiddlywiki.com/plugins/tiddlywiki/d3

Dashes in WikiText

13th May 2022 at 1:19pm

WikiText

You can create an n-dash with a double hyphen – and an m-dash with a triple hyphen `---`.

copy to clipboard

```
* -- n-dash example
* --- m-dash example --- have fun!
```

That renders as:

- – n-dash example
- — m-dash example — have fun!

... and the underlying HTML is:

```
<ul><li>– n-dash example </li><li>— m-dash example — have fun!</li></ul>
```

Data URI

21st February 2015 at 7:17pm

Definitions

A **data URI** is a way of storing data (such as an image) in a way that is compatible with the addresses used by web pages and stylesheets to access external resources.

The `datauri` macro can be used to generate data URIs within TiddlyWiki.

DataTiddlers

27th April 2022 at 7:13pm

Concepts

A data tiddler is a miniature database contained within a tiddler.

There are two standard formats:

- DictionaryTiddlers
- JSONTiddlers

Other formats of tiddler can also be parsed to yield blocks of data that behave like data tiddlers.

See JSON in TiddlyWiki for an overview.

datauri Macro

21st February 2015 at 11:28pm

Core Macros Macros

The **datauri** macro returns a data URI for the content of a tiddler.

It is often used in stylesheet tiddlers to reference things like inline images and fonts:

```
background: url(<<datauri "Motovun Jack.jpg">>);
```

The data URI is automatically base64-encoded in the case of a non-text tiddler.

Parameters

title

The title of a tiddler, such as an image

Examples

datauri Macro (Examples)

21st February 2015 at 11:35pm

datauri Macro Macro Examples

copy to clipboard

```
<style>
.jack {
    background: url(<$macrocall $name="datauri" title="Motovun Jack.jpg" $output="text/plain"/>);
    height: 300px;
}
</style>
```

```
<div class="jack"/>
```

Try it

The example has to invoke **datauri** through the **$macrocall** widget. It needs to prevent the macro's output from being parsed as WikiText, as that would transform the data URI into a **$link** widget and break the example. If the example was in a CSS tiddler, you could simply write:

```
background: url(<<datauri "Motovun Jack.jpg">>);
```

Date Fields

26th February 2023 at 3:46pm

Concepts

Certain fields of a tiddler are used to store dates and times. TiddlyWiki supports dates from the year -9999 to the year 9999.

The two standard date fields are **created** and **modified**.

Values of date fields are 17 or 18-character strings:

- ⓘ **Introduced in v5.1.23** an optional minus sign `-` to indicate a negative year
- 4 digits for the year
- 2 digits for the month
- 2 digits for the day
- 2 digits for the hour

- 2 digits for the minute
- 2 digits for the second
- 3 digits for the millisecond

To avoid problems arising from differences of time zone, TiddlyWiki always uses UTC.

The DateFormat template for storage of dates in these fields is `[UTC]YYYY0MM0DD0hh0mm0ss0XXX`.

As an example, the **created** field of this tiddler has the value **20150117190213631**.

Dates can be converted to other formats for display:

copy to clipboard

```
<$view field="created" format="date" template="DDD DDth MMM YYYY"/>
```

That renders as:

Saturday 17th January 2015

DateFormat

26th February 2023 at 3:43pm

Features

The default representation of dates is a compact string such as **20211002153802059**. The associated template is

`[UTC]YYYY0MM0DD0hh0mm0ss0XXX`. For example, the **`created`** and **`modified`** fields are stored like this.

The display format for this string can be controlled with a template. For example, transcluding the **`modified`** field automatically applies a template to display the date as **`Sat Oct 02 2021 17:40:50 GMT+0200 (Central European Summer Time)`**. A few widgets and filter operators allow you to manually specify a template, for example the ViewWidget:

```
<$view field=modified format=date template="DDth mmm YYYY 0hh:0mm:0ss" />
```

The date string is processed with the following substitutions:

Token	Substituted Value
`ddddd`	ⓘ **Introduced in v5.2.0** Day of year (1 to 365, or 366 for leap years)
`0ddddd`	ⓘ **Introduced in v5.2.0** Zero padded day of year (001 to 365, or 366 for leap years)
`DDD`	Day of week in full (eg, "Monday")
`ddd`	Short day of week (eg, "Mon")
`dddd`	ⓘ **Introduced in v5.2.0** Weekday number from 1 through 7, beginning with Monday and ending with Sunday
`DD`	Day of month
`0DD`	Adds a leading zero
`DDth`	Adds a suffix
`WW`	ISO-8601 week number of year
`0WW`	Adds a leading zero
`MMM`	Month in full (eg, "July")
`mmm`	Short month (eg, "Jul")
`MM`	Month number
`0MM`	Adds leading zero

`YYYY`	Full year
`YY`	Two digit year
`wYYYY`	Full year with respect to week number
`aYYYY`	ⓘ **Introduced in v5.1.23** Full year but negative dates are displayed as positive
`wYY`	Two digit year with respect to week number
`{era:BCE||CE}`	ⓘ **Introduced in v5.1.23** Displays a different string for years that are negative, zero or positive (see below)
`hh`	Hours
`0hh`	Adds a leading zero
`hh12`	Hours in 12 hour clock
`0hh12`	Hours in 12 hour clock with leading zero
`mm`	Minutes
`0mm`	Minutes with leading zero
`ss`	Seconds
`0ss`	Seconds with leading zero
`XXX`	Milliseconds
`0XXX`	Milliseconds with leading zero
`am` or `pm`	Lower case AM/PM indicator
`AM` or `PM`	Upper case AM/PM indicator
`TZD`	Timezone offset
`TIMESTAMP`	ⓘ **Introduced in v5.2.4** Number of milliseconds since the ECMAScript epoch, 1 January 1970.
`\x`	Used to escape a character that would otherwise have special meaning
`[UTC]`	Time-shift the represented date to UTC. Must be at very start of format string

Note that other text is passed through unchanged, allowing commas, colons or other separators to be used.

The `{era:BCE||CE}` notation can specify different strings for years that are negative, zero or positive. For example `{era:BC|Z|AD}` would display **BC** for negative years, **AD** for positive years, and **Z** for year zero.

Examples

Template	Output
`DDth MMM YYYY`	16th February 2011
`DDth MMM \M\M\M YYYY`	16th February MMM 2011
`DDth mmm YYYY 0hh:0mm:0ss`	16th Feb 2011 11:38:42

Using **TIMESTAMP** to calculate time difference

You can calculate the difference between two dates by doing the following:

1. Convert both dates to timestamps
2. Subtract the later date from the earlier one – if you don't know which one is earlier use the **abs** operator to get an absolute value after subtraction
3. Divide the resulting number by the number of milliseconds in your chosen interval

Here is an example of calculating the number of days that passed between creation and last modification of current tiddler:

- Convert the **created** and **modified** fields to timestamps
- Divide their difference by **86400000** which is the number of milliseconds in a day
    - 1000 milliseconds per second × 60 seconds per minute × 60 minutes per hour × 24 hours per day = 86,400,000 milliseconds per

day

copy to clipboard

```
<$let
    timestamp-modified={{{ [{!!modified}format:date[TIMESTAMP]] }}}
    timestamp-created={{{ [{!!created}format:date[TIMESTAMP]] }}}
    difference-days={{{ [<timestamp-modified>subtract<timestamp-created>divide[86400000]floor[]] }}}>

* ''Modified date:'' <$text text={{{ [{!!modified}format:date[YYYY-0MM-0DD]] }}}/>
* ''Created date:'' <$text text={{{ [{!!created}format:date[YYYY-0MM-0DD]] }}}/>
* ''Difference in days:'' <<difference-days>> days
</$let>
```

Try it

datepicker plugin based on Pikaday, by kixam

6th January 2021 at 4:10pm

Community Plugins

An interactive date picker plugin based on Pikaday. A demo can be found here: http://kixam.github.io/TW5-datePicker/.

> datepicker is a TiddlyWiki plugin that allows you to use a widget much like EditTextWidget to pick a date, and optionaly a time. It shows a nice interactive mini-calendar when clicked, allows you to set a date format for display, and is compatible with core date

> fields such as `modified` and `created`.

Days of the Week

16th November 2021 at 11:12pm

Operator Examples

This example tiddler is used to illustrate some of the Filter Operators.

It has a **`list`** field containing the following items:

- Monday
- Tuesday
- Wednesday
- Thursday
- Friday
- Saturday
- Sunday

days Operator

Date Operators Filter Operators Negatable Operators

purpose	filter the input by date

input	a selection of titles
suffix	**F** = the name of a date field, defaulting to **modified**
parameter	**D** = a { positive | negative } number of days, defaulting to 0
output	those input tiddlers in which field **F** is **D** days in the { future | past } or any time { before | after } that, including { past | future }
! output	those input tiddlers in which field **F** is at least **D** days in the { future | past }

Learn more about how to use Filters

Select tiddlers where a specified date field (default "modified") is within a specified date range. Time portion is ignored.

Examples

days Operator (Examples)

days Operator Operator Examples

`[days[-14]]`
→ tiddlers modified within the last 14 days

Try it

`[!days:created[-800]]`
→ tiddlers created more than 800 days ago

Try it

The filter can be used to highlight new items in a list. For example:

copy to clipboard

```
<ul>
  <$list filter="[tag[ReleaseNotes]!<currentTiddler>!sort[modified]]">
    <li>
      <$link><$view field="title"/></$link>
         <$list filter="[<currentTiddler>days[-180]]"> @@color:red;^^new^^@@</$list>
         <$list filter="[<currentTiddler>days[-500]!days[-180]]"> @@color:black;^^recent^^@@</$list>
    </li>
  </$list>
</ul>
```

That renders as:

- Release 5.3.3 new
- Release 5.3.2 new
- Release 5.3.1 recent
- Release 5.3.0 recent
- Release 5.2.7 recent
- Release 5.2.6 recent

- Release 5.2.5 recent
- Release 5.2.4 recent
- Release 5.2.3
- Release 5.2.2
- Release 5.2.1
- Release 5.2.0
- Release 5.1.23
- Release 5.1.22
- Release 5.1.21
- Release 5.1.20
- Release 5.1.19
- Release 5.1.18
- Release 5.1.17
- Release 5.1.16
- Release 5.1.15
- Release 5.1.14
- Release 5.1.13
- Release 5.1.12
- Release 5.1.11
- Release 5.1.10
- Release 5.1.9
- Release 5.1.8

- Release 5.1.3
- Release 5.1.5
- Release 5.1.6
- Release 5.1.7
- Release 5.1.4
- Release 5.1.2
- Release 5.1.1
- Release 5.1.0

Debugging Stylesheets

25th November 2021 at 6:59pm

Using Stylesheets

The core shadow tiddler **$:/snippets/DebugStylesheets** contains a simple tool for tracking down certain problems with mismatched brackets and other nested structures.

In the list below, any stylesheets containing an error will be marked with a red cross:

- $:/_tw5.com-styles
- $:/editions/tw5.com/doc-styles
- Custom data-styles

decodebase64 Operator

Filter Operators String Operators

purpose	apply base 64 decoding to a string
input	a selection of titles
suffix	optional: `binary` to produce binary output, `urlsafe` for URL-safe input
output	the input with base 64 decoding applied

Learn more about how to use Filters

Introduced in v5.2.6

See Mozilla Developer Network for details of base 64 encoding. TiddlyWiki uses library code from @nijikokun to handle the conversion.

The input strings must be base64 encoded. The output strings are the text (or binary data) decoded from base64 format.

The optional `binary` suffix, if present, changes how the input is processed. The input is normally assumed to be UTF-8 text encoded in base64 form (such as what the **encodebase64** operator produces), so only certain byte sequences in the input are valid. If the input is binary data encoded in base64 format (such as an image, audio file, video file, etc.), then use the optional `binary` suffix, which will allow all byte sequences. Note that the output will then be binary, **not** text, and should probably not be passed into further filter operators.

The optional `urlsafe` suffix, if present, causes the decoder to assume that the base64 input uses `-` and `_` instead of `+` and `/` for the 62nd and 63rd characters of the base64 "alphabet", which is usually referred to as "URL-safe base64" or "bae64url".

Examples

decodehtml Operator

17th October 2016 at 5:29pm

Filter Operators String Operators

purpose	apply HTML decoding to a string
input	a selection of titles
output	the input with HTML decoding applied

Learn more about how to use Filters

Introduced in v5.1.14

"HTML decoding" means replacing HTML entities that represent special characters with that character:

- `&` replaced with `&`
- ` ` replaced with ` ` (non breaking space)
- `<` replaced with `<`
- `>` replaced with `>`
- `"` replaced with `"`

Examples

decodehtml Operator (Examples)

17th October 2016 at 5:53pm

decodehtml Operator Operator Examples

```
[[Title with &lt; angle brackets &gt;]] +[decodehtml[]]
```

Try it

```
[[Title with an &amp; ampersand]] +[decodehtml[]]
```

Try it

decodeuri Operator

17th October 2016 at 5:29pm

Filter Operators String Operators

purpose	apply URI decoding to a string
input	a selection of titles
output	the input with URI decoding applied

Learn more about how to use Filters

ⓘ Introduced in v5.1.14

See Mozilla Developer Network for details of the decodeURI operation.

Examples

decodeuri Operator (Examples)

17th October 2016 at 5:53pm

decodeuri Operator Operator Examples

```
[[Title%20with%20Space]] +[decodeuri[]]
```

Try it

```
[[Title%20with%20Space]] [[Another%20title%20with%20Space]] +[decodeuri[]]
```

Try it

decodeuricomponent Operator

17th October 2016 at 5:26pm

Filter Operators String Operators

purpose	apply URI component decoding to a string
input	a selection of titles
output	the input with URI component decoding applied

Learn more about how to use Filters

Introduced in v5.1.14

See Mozilla Developer Network for details of the decodeURIComponent operation.

Examples

decodeuricomponent Operator (Examples)

17th October 2016 at 5:53pm

decodeuricomponent Operator Operator Examples

```
[[Title%20with%20Space]] +[decodeuricomponent[]]
```

Try it

```
[[Title%20with%20Space]] [[Another%20title%20with%20Space]] +[decodeuricomponent[]]
```

Try it

DefaultTiddlers

6th March 2018 at 5:12pm

Concepts

"Default tiddlers" refers to the list of tiddlers that is opened at startup in the browser if the address bar does not include a permalink).

At startup, the text of the system tiddler $:/DefaultTiddlers is retrieved and evaluated as a filter expression. The resulting list of titles is then inserted into the story river.

The **control panel** **info**/**basics** tab includes a text box for direct access to $:/DefaultTiddlers.

Definitions

25th August 2013 at 4:51pm

Reference

These are definitions of technical words and phrases used in this documentation. (As distinct from the Concepts that make up TiddlyWiki itself).

- Base64
- Brackets
- BT
- CamelCase
- Cascading Style Sheets
- Comma-Separated Values
- ContentType
- Data URI
- Document Object Model
- Federatial

- Git
- GitHub
- GitLab
- GuerillaWiki
- HTML Block Elements
- HyperText Markup Language
- JavaScript
- JavaScript Object Notation
- JeremyRuston
- Jermolene
- Keyboard Shortcut Descriptor
- Legacy
- Markdown
- MathML
- ModuleType
- node-webkit
- Node.js
- NW.js
- OpenSource
- Osmosoft
- Percent Encoding
- Quine

- SingleFileApplication
- SinglePageApplication
- Stanford JavaScript Crypto Library
- TiddlyDesktop
- TiddlyFox
- Tiddlyhost
- TiddlyIE
- TiddlySpace
- TiddlyWeb
- TiddlyWikiClassic
- URI
- Web Developer Tools
- Xememex

Definitions in WikiText

5th December 2013 at 5:04pm

WikiText

HTML definition lists are created with this syntax:

copy to clipboard

```
; Term being defined
: Definition of that term
```

```
; Another term
: Another definition
```

That renders as:

Term being defined
: Definition of that term

Another term
: Another definition

... and the underlying HTML is:

```
<dl><dt>Term being defined</dt><dd>Definition of that term</dd><dt>Another term</dt><dd>Another definition</dd></dl>
```

DeleteTiddlersCommand

30th May 2019 at 5:54pm

Commands

Introduced in v5.1.20 Deletes a group of tiddlers identified by a filter.

```
--deletetiddlers <filter>
```

Demo Tiddler List with Custom Story Tiddler Template

4th December 2021 at 2:16pm

$:/tags/TiddlerList

This is a demo tiddler with a custom story tiddler template that displays the tiddlers named in the list field as a fan.

See How to create a custom story tiddler template for details.

Demonstration: keyboard-driven-input Macro

23rd November 2021 at 4:45am

KeyboardDrivenInput Learning

The keyboard-driven-input Macro is used to create filtered lists that update as the user types. It also allows navigating the lists, and invoking macros using list items, with the keyboard.

It consists of an edit-text widget wrapped in keyboard widgets. There are **keyboard** widgets to listen for `((input-accept))`, `((input-accept-variant))`, and `((input-cancel))` keyboard shortcuts, but there is no default behaviour in response to these events; macros must be written to suit the use-case. There is also no default visualisation of the filtered options list.

Types of keyboard input handled by the **`keyboard-driven-input`** macro:

Non-configurable:

The following keyboard events invoke macros defined alongside **`keyboard-driven-input`** in **$:/core/macros/keyboard-driven-input**.

Input	Purpose	Macro
typing input	composing a string to be used within list filters	**keyboard-input-actions**
`((input-up))` (`Up`)	temporarily selecting the previous item in the filtered list	**input-next-actions** with parameters **afterOrBefore**= `"before"` , **reverse**= `"reverse[]"`
`((input-down))` (`Down`)	temporarily selecting the next item in the filtered list	**input-next-actions** with **afterOrBefore**= `"after"` , **reverse**= `""`

All of the above actions generate or modify data which **keyboard-driven-input** keeps in tiddlers specified using the macro's parameters. The data can then be accessed not only by the macros invoked by keyboard shortcuts, but also outside of these, e.g. a macro to display the filtered list(s).

Configurable through parameters:

The following keyboard events invoke macros whose names are specified in parameters to **keyboard-driven-input**. The intended purpose is suggested by the parameter name, but there are no default macros defined within **$:/core/macros/keyboard-driven-input**.

Keyboard shortcut descriptor	Key combination	Parameter
`((input-accept))`	`Enter`	**inputAcceptActions**
`((input-accept-variant))`	`ctrl-Enter`	**inputAcceptVariantActions**
`((input-cancel))`	`Escape`	**inputCancelActions**

keyboard-driven-input can be seen in action as part of various core features in TiddlyWiki, e.g.: the search feature, the tag-picker Macro, and dropdown interfaces in the Editor toolbar such as Insert link.

Minimal ingredients for a demonstration

Keeping in mind that the **`keyboard-driven-input`** macro does not, by itself, display list results, or do anything with a selected option, a minimal demonstration of the **`keyboard-driven-input`** macro requires:

1. **a tiddler, containing a filter** whose results depend on the user's text input, to generate the options from which the user can select
2. **several parameters:**
    - **a filter** to return the title of the tiddler where the filter described in step 1 can be found, and
    - **tiddler titles** to use for storing state information in response to input events. These tiddlers do not need to exist already. If they do exist, the **`keyboard-driven-input`** macro will change their contents.

The search filter(s)

By default, **`keyboard-driven-input`** will look for filters in the **`first-search-filter`** and **`second-search-filter`** fields of a tiddler (whose title is specified by a parameter discussed below).

This filter can refer to a variable called **`userInput`**, which shows the contents of the **edit-text** widget, as stored in a state tiddler (discussed below), at the time of the latest `Up` or `Down` event.

Note: If this filter is to be referred to in a context outside the **`keyboard-driven-input`** macro (such as in a popup showing the filtered options), the variable **`userInput`** has to be defined in those contexts as well (by reading it from a state tiddler).

Minimal parameters

Parameter name	Notes

Parameter name	Notes
configTiddlerFilter	This is a filter, rather than a tiddler title, allowing conditional behaviour (e.g. checking for an active tab, or preferring a filter that may not be present, with a fallback). The title returned must belong to an existing tiddler, containing at least one filter with which **keyboard-driven-input** can generate its results list.
tiddler	This tiddler contains either the typed input, or the instantaneous result selection, depending upon the most recent event. It is updated with each keystroke in the **edit-text** widget, *and* when the user uses the `Up` or `Down` key to cycle through filtered results.
storeTitle	This tiddler always reflects the user input (transcluded from the tiddler **tiddler** after each keystroke into the **edit-text** widget).
selectionStateTitle	This tiddler is updated on `Up` or `Down` events and contains either the user input with the suffix `-userInput`, or the instantaneous selection with the suffix `-primaryList` or `-secondaryList`, depending on which of up to two filters generated the list it came from.

Demonstration setup

I have created a tiddler called kdi-demo-configtid and put the following filter into its **first-search-filter** field:

`[!is[system]search:title<userInput>sort[]]`.

This filters for non-system tiddlers whose titles contain the text the user has typed.

To use the above filter with **keyboard-driven-input**, the value of the parameter **configTiddlerFilter** should be a filter that returns kdi-demo-configtid.

I can select **`tiddler`**, **`storeTitle`**, and **`selectionStateTitle`** fairly arbitrarily (just making sure not to use titles of tiddlers that I do not want changed).

This demonstration can now be invoked with the following macro call:

```
<$macrocall $name=keyboard-driven-input
        tiddler="kdi-demo-tiddler"
        storeTitle="kdi-demo-storeTitle"
        selectionStateTitle="kdi-demo-selectionStateTitle"
        configTiddlerFilter="[[kdi-demo-configtid]]"
 />
```

Demo

Try typing in here:

Observe the changes in the various state tiddlers in the below table. Use `Up` and `Down` keys to navigate among filter results. If nothing changes, try a shorter input to widen the filter. If the input has zero length, the list will contain all non-system tiddlers.

Parameter name	Tiddler title	Contents of `text` field of the tiddler
`tiddler`	*kdi-demo-tiddler*	
`storeTitle`	*kdi-demo-storeTitle*	
`selectionStateTitle`	*kdi-demo-selectionStateTitle*	

The **keyboard-driven-input** macro has many parameters available, including all the attributes of the enclosed **edit-text** widget, which make it very flexible in how it is used and how results can be displayed and interacted with.

See also: Customising search results and **$:/core/ui/DefaultSearchResultList**

Deprecated - What does it mean

4th July 2022 at 7:42pm

Deprecated features of TiddlyWiki are those that have been superseded by newer, improved ways of doing the same thing.

Deprecated features will still work, but are not recommended for new content.

Deprecated features are marked with a special warning button. See: How to apply custom styles by tag for an example.

Tiddlers tagged `$:/deprecated`

> - How to apply custom styles by tag - deprecated
> - macrocall
> - TiddlyWiki in the Sky for TiddlyWeb - deprecated

deserialize Operator

27th June 2023 at 11:41am

Filter Operators Special Operators

purpose	extract JSON representation of tiddlers from the input strings

input	a selection of strings
parameter	the deserializer module to be used to extract tiddlers from the input
output	JSON representations of tiddlers extracted from input titles.

Learn more about how to use Filters

New in v5.3.0

Deserializer modules parse text in various formats into their JSON representation as tiddlers. The deserializer modules available in a wiki can be seen using the deserializers operator and can be used with the deserialize Operator.

The TiddlyWiki core provides the following deserializers:

Deserializer	Description
(DOM)	Extracts tiddlers from a DOM node, should not be used with the **`deserialize[]`** operator
application/javascript	Parses a JavaScript module as a tiddler extracting fields from the header comment
application/json	Parses JSON into tiddlers
application/x-tiddler	Parses the .tid file format as a tiddler
application/x-tiddler-html-div	Parses the <DIV>.tiddler file format as a tiddler
application/x-tiddlers	Parses the MultiTiddlerFile format as tiddlers
text/css	Parses CSS as a tiddler extracting fields from the header comment
text/html	Parses an HTML file into tiddlers. Supports TiddlyWiki Classic HTML files, TiddlyWiki5 HTML files and ordinary HTML files
text/plain	Parses plain text as a tiddler

Examples

deserialize Operator (Examples)

2nd June 2023 at 12:50pm

deserialize Operator Operator Examples

This example uses the predefined variable `html-data`:

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Test Data</title>
</head>
<body>

<!--~~ Ordinary tiddlers ~~-->
<div id="storeArea" style="display:none;"><div title="Hello &quot;There&quot;" type="text/
vnd.tiddlywiki">
<pre>Abacus</pre>
</div>
</div>
<script class="tiddlywiki-tiddler-store" type="application/json">[{"title":"Hello
\"There\"","text":"Calculator"},{"title":"Hello \"There\"","text":"Protractor"}]</script>
</body>
</html>
```

```
[<html-data>deserialize[text/html]]
```

Try it

Deserializers

27th June 2023 at 11:43am

Features

Deserializer modules parse text in various formats into their JSON representation as tiddlers. The deserializer modules available in a wiki can be seen using the deserializers operator and can be used with the deserialize Operator.

The TiddlyWiki core provides the following deserializers:

Deserializer	Description
(DOM)	Extracts tiddlers from a DOM node, should not be used with the **`deserialize[]`** operator
application/javascript	Parses a JavaScript module as a tiddler extracting fields from the header comment
application/json	Parses JSON into tiddlers
application/x-tiddler	Parses the .tid file format as a tiddler
application/x-tiddler-html-div	Parses the <DIV>.tiddler file format as a tiddler
application/x-tiddlers	Parses the MultiTiddlerFile format as tiddlers
text/css	Parses CSS as a tiddler extracting fields from the header comment
text/html	Parses an HTML file into tiddlers. Supports TiddlyWiki Classic HTML files, TiddlyWiki5 HTML files and ordinary

	HTML files
text/plain	Parses plain text as a tiddler

deserializers Operator

27th June 2023 at 11:42am

Filter Operators Selection Constructors Special Operators

input	ignored
parameter	none
output	the title of each available deserializer

Learn more about how to use Filters

Introduced in v5.2.0



You can specify a specific deserializer for a DropzoneWidget to use

Examples

deserializers Operator (Examples)

6th May 2021 at 1:54pm

deserializer Operator Operator Examples

```
[deserializers[]]
```

Try it

Deutsch (Deutschland) Edition

19th September 2014 at 11:58pm

Languages

Die deutsche Übersetzung von TiddlyWiki ist verfügbar unter:

- **Dokumentation**: https://tiddlywiki.com/languages/de-DE/index.html
- **Leer**: https://tiddlywiki.com/languages/de-DE/empty.html

Siehe auch: Deutsch (Österreich) Edition.

Deutsch (Österreich) Edition

19th September 2014 at 11:59pm

Languages

Die österreichische Übersetzung von TiddlyWiki ist verfügbar unter:

- **Dokumentation**: https://tiddlywiki.com/languages/de-AT/index.html
- **Leer**: https://tiddlywiki.com/languages/de-AT/empty.html

Siehe auch: Deutsch (Deutschland) Edition.

Developers

22nd December 2020 at 12:47pm

Community Reference

There are several resources for developers to learn more about TiddlyWiki and to discuss and contribute to its development.

- tiddlywiki.com/dev is the official developer documentation
- Get involved in the development on GitHub
  - Discussions are for Q&A and open-ended discussion
  - Issues are for raising bug reports and proposing specific, actionable new ideas
- The older TiddlyWikiDev Google Group is now closed in favour of GitHub Discussions but remains a useful archive: https://groups.google.com/group/TiddlyWikiDev
  - An enhanced group search facility is available on mail-archive.com
- Follow @TiddlyWiki on Twitter for the latest news
- Chat at https://gitter.im/TiddlyWiki/public (development room coming soon)

DictionaryTiddlers

27th April 2022 at 7:10pm

Concepts

A dictionary tiddler is a kind of data tiddler that contains a simple list of `name: value` pairs.

Its ContentType is `application/x-tiddler-dictionary`.

The `text` field consists of one or more lines of the form *`name: value`*.

ColourPalettes, such as the **default Vanilla palette**, are dictionary tiddlers.

DiffTextWidget

16th March 2018 at 5:27pm

Widgets

Introduction

Introduced in v5.1.16 The diff text widget analyses the differences between a pair of source and destination text strings and displays the results as highlighted insertions and deletions (similar to the "track changes" function of a word processor). For example:

8 differences

```
Hey Jude, don't ~~make it bad. Take a sad song and make it better. Remember to~~be afraid. You were made to go out and get her. The minute you let her ~~into~~under your ~~heart↵
skin.~~ Then you ~~can start~~begin to make it better.
```

Content and Attributes

Content

The content of the `<$diff-text>` widget is rendered immediately before the diffs. Within it, the variable `diff-count` is available, containing the number of differences found. If the widget has no content then it automatically transcludes the tiddler **$:/language/Diffs/CountMessage**.

> In other words, these three invocations are all equivalent:
>
> ```
> <$diff-text source={{FirstTiddler}} dest={{SecondTiddler}}>
> {{$:/language/Diffs/CountMessage}}
> </$diff-text>
>
> <$diff-text source={{FirstTiddler}} dest={{SecondTiddler}}>
> </$diff-text>
>
> <$diff-text source={{FirstTiddler}} dest={{SecondTiddler}}/>
> ```

Attributes

Attribute	Description
source	The source text
dest	The destination text
cleanup	See below

The **cleanup** attribute determines which optional post-processing should be applied to the diffs:

- **none**: no cleanup is performed

- **semantic** (default): rewrites the diffs for human readability
- **efficient**: rewrites the diffs to minimise the number of operations for subsequent processing

(Note that in many cases the results will be the same regardless of the cleanup option. See the docs of the underlying library for more details).

Examples

In this example we compare two texts:

copy to clipboard

```
<$edit-text tiddler="SampleTiddlerFirst"/>

<$edit-text tiddler="SampleTiddlerSecond"/>

<$diff-text source={{SampleTiddlerFirst}} dest={{SampleTiddlerSecond}}/>
```

That renders as:

This is a test tiddler called SampleTiddlerFirst.	This test tiddler is called SampleTiddlerSecond.

4 differences

```
This ~~is a~~ test tiddler is called SampleTiddler~~First~~Second.
```

Disabling Plugins

15th October 2016 at 2:17pm

PluginMechanism

Disabling Plugins

Plugins can be disabled by creating a tiddler titled `$:/config/Plugins/Disabled/` concatenated with the plugin title, and setting its text to `yes`.

For example, to disable the plugin `$:/plugins/tiddlywiki/highlight`, the title would be:

```
$:/config/Plugins/Disabled/$:/plugins/tiddlywiki/highlight
```

Discover TiddlyWiki

23rd October 2015 at 6:53pm

HelloThere

You've never seen *anything* like TiddlyWiki. TiddlyWiki is:

a tool,
a toolbox,
a community,
and a philosophy.

TiddlyWiki is a Tool

TiddlyWiki is first and foremost a tool: it is a free downloadable tool for capturing and organising content from the web, from your documents or from your brain. It is a tool for note-taking, bookmarking, pinning, writing, managing to-do lists and projects, collaborating, blogging, and publishing.

In TiddlyWiki you create or paste content into notes called tiddlers, then connect your tiddlers with hyperlinks and tags. You can then quickly retrieve your notes through features such as tag pills, sidebar tabs, and TiddlyWiki's lightning fast search window. You can even dynamically include one tiddler's content inside another - similar to using building blocks - to create articles, lists, presentations and more.

TiddlyWiki is a Toolbox

In addition to being a versatile tool, **TiddlyWiki is also a toolbox.**

Where other note-taking products hook you with the basic program then charge you for the really helpful features, TiddlyWiki has an ever-expanding collection of completely free visual themes, colour palettes, plugins, widgets and macros, which you can then mix and match so that you can tweak and tailor your TiddlyWiki to get it just the way you want it.

TiddlyWiki is also a Community

We are a community of users and developers who help each other imagine new ways of thinking and organising and create new solutions, so that TiddlyWiki is continually adapting to better serve your needs. TiddlyWiki users and developers share questions and advice at a TiddlyWiki Google group. They also create tutorials, adaptations, and plugins to enhance your TiddlyWiki experience. See the section **Community** of the

TableOfContents for more details.

Finally, TiddlyWiki is a philosophy

The purpose of taking and organising a note (or recording any kind of content) is to be able to retrieve it later. If you can't find your notes in your note-taking system, your note-taking becomes a colossal waste of time.

The TiddlyWiki philosophy is that the best way to take notes is to separate them into tiddlers, the smallest semantic units possible. A tiddler might be an image, a webpage link, a concept, the definition of a term, or a specific customisation such as a macro.

These tiddlers can then be woven together to create longer units: stories, articles, lists, image galleries, and much more. TiddlyWiki's features such as tagging, hyperlinking, and filters are specially designed to help you relate and connect tiddlers together in multiple ways, facilitating your future retrieval of your notes and even helping you see unexpected relationships among your tiddlers and the information they contain.

Disqus comments plugin by bimlas

6th January 2021 at 4:10pm

Community Plugins

Use this plugin to give your visitors the opportunity to comment on your tiddlers without changing the wiki itself.

Disqus is a networked community platform used by hundreds of thousands of sites all over the web. With Disqus, your website gains a feature-rich comment system complete with social network integration, advanced administration and moderation options, and other extensive community functions.

https://bimlas.github.io/tw5-disqus/

divide Operator

11th June 2019 at 2:59pm

Binary Mathematics Operators Filter Operators Mathematics Operators

purpose	treating each input title as a number, divide it by the numeric value of the parameter
input	a selection of titles
parameter	`N` = a number
output	the input as numbers, but with each one divided by `N`

Learn more about how to use Filters

Introduced in v5.1.20 See Mathematics Operators for an overview.

Examples

divide Operator (Examples)

11th June 2019 at 3:01pm

divide Operator Operator Examples

```
[[355]divide[113]]
```

Try it

```
=1 =2 =3 =4 +[divide[2]]
```

Try it

Document Object Model

28th February 2015 at 2:40pm

Definitions

The **DOM** of a web page is a tree-shaped model of its content, maintained internally by the web browser as the user interacts with that content. Each point in the tree is called a **node**.

When TiddlyWiki is running in a web browser, its widgets are rendered into DOM nodes for display.

When TiddlyWiki is running on Node.js, it uses a fake DOM to simulate the behaviour of a browser.

Documentation Macros

25th March 2023 at 5:14pm

Improving TiddlyWiki Documentation

The following macros are used throughout TiddlyWiki's documentation. Their names start with a dot to keep them out of the way of names that a user might try experimenting with.

General

Macro	Used for	Example
.def	the defining instance of a term	**widget**
.em	minor emphasis within a sentence	*NOT*
.place	a placeholder for the user to fill in	`tagname`
.strong	major emphasis within a tiddler	**Important!**
.word	a mention of an ordinary word or phrase	"hello world"
.icon	an icon, sized to match the surrounding text	

Advice

Macro	Used for	Example	
.tip	hints and tips		Turn your screen on, otherwise you won't be able to see much.
.warning	warning advice		Make a backup of your file before you upgrade.

Blocks

Macro	Used for
.preamble	an introductory sentence that stands apart from the rest of the tiddler

Tiddlers and fields

Macro	Used for	Example
.tid	a tiddler title	**Example**
.tag	a tag	**Example**
.field	a field name	**example**
.value	a field value	**example value**
.op	a filter operator	**backlinks**
.var	a variable or macro name	**currentTiddler**
.wid	a widget name	**$list**
.attr	an attribute name	**filter**
.param	a macro parameter name	**text**
.tiddler-fields	a list of tiddler fields	Monday *created* 20150117192113000 *modified* 20211116221111294 *tags* [[Operator Examples]] *text* <<.this-is-operator-example>> *type* text/vnd.tiddlywiki

Links

Macro	Used for	Example
.link	a link containing WikiText	*an ~~example~~*
.clink	a code link	`<$list>`
.dlink	a link on a defining instance of a term	**widget**
.dlink-ex	an external link on a defining instance of a term	**Example**
.flink	a link to a field	**list**
.mlink	a link to a macro	**qualify**
.mlink2	a link to a macro, with specified target	***foo***
.olink	a link to an operator	**prefix**
.olink2	a link to an operator, with specified target	**foo**
.vlink	a link to a variable	**currentTiddler**
.vlink2	a link to a variable, with specified target	***foo***
.wlink	a link to a widget	**$button**
.wlink2	a link to a widget, with specified text	foo

Tabs

Macro	Used for	Example
.doc-tabs	showing a tab set in a documentation tiddler	–
.doc-tab-link	button to activate a tab	–
.widget-attr-link	button with a widget attribute name to activate a tab	–

User interface

Macro	Used for	Example
.key	a key on the keyboard	**Escape**
.keycombo	a key combination	**Ctrl**+**Enter**

Tabs

Macro	Used for	Example
.sidebar-tab	the name of a sidebar tab	**More**
.more-tab	the name of a subtab of the More tab	**Shadows**
.info-tab	the name of a tiddler info tab	**Fields**
.controlpanel-tab	the name of a Control Panel tab	**Settings**
.advancedsearch-tab	the name of an Advanced Search tab	**Filter**
.toc-tab	name of the tw5.com TOC tab	**Contents**
.example-tab	an example tab name	**Notes**

Parameters for .sidebar-tab

Open	**Open**
Recent	**Recent**
Tools	**Tools**

More	**More**

Parameters for .more-tab

All	**All**
Recent	**Recent**
Tags	**Tags**
Missing	**Missing**
Drafts	**Drafts**
Orphans	**Orphans**
Types	**Types**
System	**System**
Shadows	**Shadows**

Parameters for .info-tab

Tools	**Tools**
References	**Backlinks**
Tagging	**Tagging**
List	**List**
Listed	**Listed**
Fields	**Fields**
Advanced	**Advanced**

Parameters for .controlpanel-tab

Info	**Info**
Appearance	**Appearance**
Settings	**Settings**
Saving	**Saving**
Plugins	**Plugins**

Parameters for .advancedsearch-tab

Standard	**Standard**
System	**System**
Shadows	**Shadows**
Filter	**Filter**

Buttons

Macro	Used for	Example
.button	a standard button name and icon	+ **new tiddler**

Parameters for .button

Tiddler toolbar

clone	**clone**
close	**close**
close-others	**close others**
edit	**edit**
export-tiddler	**export tiddler**
info	**info**
more-tiddler-actions	**more**
new-here	**new here**
new-journal-here	**new journal here**
permalink	**permalink**

Edit-mode toolbar

cancel	**cancel**
delete	**delete**
save	**ok**

Page toolbar

advanced-search	**advanced search**
close-all	**close all**
control-panel	**control panel**
encryption	**encryption**
export-page	**export all**

full-screen	**full-screen**
home	**home**
import	**import**
language	**language**
more-page-actions	**more**
new-journal	**new journal**
new-tiddler	**new tiddler**
permaview	**permaview**
refresh	**refresh**
save-wiki	**save changes**
storyview	**storyview**
tag-manager	**tag manager**
theme	**theme**

Documentation Style Guide

17th January 2015 at 4:25pm

Improving TiddlyWiki Documentation

The documentation for TiddlyWiki tries to follow a consistent editorial style. It has two main areas, each with its own tone and audience:

- Instruction Tiddlers
- Reference Tiddlers

We keep the two areas distinct. This avoids overwhelming relative newcomers, while still providing quick access to the information that expert

users need.

Additional topics:

- Tiddler Title Policy
- Tiddler Structure
- Spelling
- Typography
- Documentation Macros
- Technical Prose Style

Dominant Append

10th June 2019 at 6:52pm

Filters

Filters manipulate sets of titles in which no title may appear more than once. Furthermore, they often need to append one such set to another.

This is done in such a way that, if a title would be duplicated, the earlier copy of that title is discarded. The titles being appended are dominant.

For example, if a selection contains `Andrew Becky Clara Daniel` and `Andrew Barney Clara Daisy` is then appended to it, the result is `Becky Daniel Andrew Barney Clara Daisy`.

This behaviour can cause unexpected results when working with Mathematics Operators. For example, `1 2 3 +[sum[]]` evaluates to `6`, as expected. But `1 1 1 +[sum[]]` evaluates to `1`. Removing the `+[sum[]]` from each filter reveals the problem: `1 2 3` evaluates to the list `1`, `2`, `3`, while `1 1 1` evaluates to the single item `1` due to de-duplication.

In such situations, the `=` prefix can be used to disable the de-duplication. For example, `=1 =1 =1 +[sum[]]` evaluates to `3` as expected. Alternatively, the split Operator can be used: `[[1,1,1]split[,]sum[]]`.

✓done

15th November 2021 at 2:33am

TaskManagementExample

DraftMechanism

20th August 2013 at 3:24pm

Mechanisms

Tiddlers that have a `draft.of` field are treated as pending drafts of the tiddler specified in the field. Draft tiddlers should also have a `draft.title` field that specifies the title that will be given to the tiddler when it is saved.

Several features work in concert to give the desired behaviour for draft tiddlers:

- The ListWidget can optionally render draft tiddlers through a different template
- The NavigatorWidget incorporates handlers for the following events:
  - `tm-new-tiddler` for creating a new tiddler in draft mode
  - `tm-edit-tiddler` for moving a tiddler into edit mode
  - `tm-cancel-tiddler` for cancelling a tiddler out of edit mode
  - `tm-save-tiddler` for saving a draft tiddler
- Draft tiddlers are automatically excluded from search operations

Drag and Drop

28th March 2017 at 7:38pm

Features

TiddlyWiki uses drag and drop to power two separate features:

- Importing Tiddlers into TiddlyWiki
- Manipulating tiddlers within a TiddlyWiki

Tiddler manipulation via drag and drop is supported by the core user interface in the following contexts:

- Entries in the "Open" tab of the sidebar can be reordered by drag and drop; new tiddlers can be opened by dragging their titles into the list
- Entries within a tag pill dropdown can be reordered by drag and drop; new tiddlers can be assigned the tag by dragging their titles into the list
- Entries in the **control panel** "Appearance"/"Toolbars" tab can be reordered by drag and drop. (Less usefully, new entries can be added to the toolbars by dragging their titles into the list)

All tiddler links are draggable by default. They can be dragged within a browser window for manipulating tiddlers, or dragged to a different browser window to initiate an import operation

If you want to drag a link, first move it vertically, because horizontal movement is recognized by the browser as text selection.

Tag pills are also draggable, and are equivalent to simultaneously dragging all of the individual tiddlers carrying the tag.

Some common scenarios for drag and drop tiddler manipulation are available as reusable macros:

- list-links-draggable Macro for reordering the entries in a tiddler ListField
- list-tagged-draggable Macro for reordering the tiddlers that carry a specified tag

See DragAndDropMechanism for details of how to use the low level drag and drop primitives to build more complex interactions.

The standard HTML 5 drag and drop APIs used by TiddlyWiki are not generally available on mobile browsers on smartphones or tablets. The Mobile Drag And Drop Shim Plugin adds an open source library that implements partial support on many mobile browsers, including iOS and Android.

DragAndDropMechanism

6th April 2017 at 10:58am

Drag and Drop Importing Tiddlers Mechanisms

This tiddler discusses the internal mechanisms that are used to implement drag and drop features in TiddlyWiki. See Drag and Drop for a general description of the features.



TiddlyWiki uses the standard HTML 5 drag and drop APIs. However, this is an area that is rightly notorious for cross-browser compatibility problems. Therefore, some features that you might expect to work won't necessarily work in all browsers. In particular,

The following widgets are concerned with drag and drop features:

- The DraggableWidget creates a draggable element that represents one or more tiddlers for dragging
  - The ButtonWidget and LinkWidget incorporate the functionality of the DraggableWidget
- The DroppableWidget creates an area into which tiddlers can be dragged to trigger customisable actions
- The DropzoneWidget handles importing external tiddlers from files or by drag and drop from another browser window

The general sequence of a drag and drop operation is as follows:

1. The user clicks down and drags the pointer on a draggable element such as the DraggableWidget, ButtonWidget or LinkWidget
2. The draggable element moves with the mouse pointer until the click is released
3. Moving the pointer over droppable elements such as the DroppableWidget displays a highlight indicating that the item can be dropped
4. The configured actions are performed if the drag ends on a droppable element

DraggableWidget

21st November 2023 at 11:14am

TriggeringWidgets Widgets

The DraggableWidget creates a DOM element that can be dragged by the user. It only works on browsers that support drag and drop, which typically means desktop browsers, but there are workarounds.

The draggable element can be assigned a list of tiddlers that are used as the payload. If desired it can invoke actions when dragging starts and when it ends. See DragAndDropMechanism for an overview.

Content and Attributes

Attribute	Description
tiddler	Optional title of the payload tiddler for the drag
filter	Optional filter defining the payload tiddlers for the drag
tag	Optional tag to override the default "div" element created by the widget
selector	ⓘ **Introduced in v5.2.2** Optional CSS Selector to identify a DOM element within the widget that will be used as the drag

	handle
class	Optional CSS classes to assign to the DOM element created by the widget. The class `tc-draggable` is added to the the DOM element created by the widget unless the **`selector`** attribute is used. The class `tc-dragging` is applied to the DOM element created by the widget while the element is being dragged
enable	ⓘ **Introduced in v5.2.3** Optional value "no" to disable the draggable functionality (defaults to "yes")
startactions	Optional action string that gets invoked when dragging **starts**
endactions	Optional action string that gets invoked when dragging **ends**
dragimagetype	ⓘ **Introduced in v5.2.0** Optional type of drag image: `dom` (the default) or `blank` to disable the drag image
data-*	ⓘ **New in v5.3.2** Optional data attributes to be assigned to the HTML element
style.*	ⓘ **New in v5.3.2** Optional CSS properties to be assigned to the HTML element

Either or both of the **tiddler** and **filter** attributes must be specified in order for there to be a payload to drag.

The actionTiddler Variable is accessible in both *startactions* and *endactions*. It holds the payload tiddler(s) specified through the *tiddler* and *filter* attributes as a Title List using double square brackets to quote titles that include whitespace.



Note that the actionTiddler Variable holds a Title List quoted with double square brackets. This is unlike the DroppableWidget which uses the same variable to pass a single unquoted title.



When specifying a DOM node to use as the drag handle with the **`selector`** attribute, give it the class `tc-draggable` in order for it to have the appropriate cursor and the attribute `draggable` with the value `true` to make it draggable.

The LinkWidget incorporates the functionality of the DraggableWidget via the **draggable** attribute.

ⓘ **Introduced in v5.2.3** The following variables are accessible in the *startactions* and the *endactions*:

Variables	Description
`modifier`	The modifier Variable contains the Modifier Key held while dragging
`dom-*`	All DOM attributes of the node being dragged are made available as variables, with the prefix `dom-`
`tv-popup-coords`	A relative co-ordinate string that can be used with the ActionPopupWidget to trigger a popup at the DOM node matching the selector where the event originated (see Coordinate Systems for more information)
`tv-popup-abs-coords`	ⓘ **Introduced in v5.2.4** An absolute co-ordinate string that can be used with the ActionPopupWidget to trigger a popup at the DOM node matching the selector where the event originated (see Coordinate Systems for more information)
`tv-selectednode-posx`	`x` offset position of the dragged DOM node
`tv-selectednode-posy`	`y` offset position of the dragged DOM node
`tv-selectednode-width`	`offsetWidth` of the dragged DOM node
`tv-selectednode-height`	`offsetHeight` of the dragged DOM node
`event-fromselected-posx`	`x` position of the event relative to the dragged DOM node
`event-fromselected-posy`	`y` position of the event relative to the dragged DOM node
`event-fromviewport-posx`	`x` position of the event relative to the viewport
`event-fromviewport-posy`	`y` position of the event relative to the viewport

DroppableWidget

21st November 2023 at 11:14am

TriggeringWidgets Widgets

The `droppable` widget creates a DOM element onto which dragged items can be dropped by the user, triggering the specified actions. It only works on browsers that support drag and drop, which typically means desktop browsers, but there are workarounds.

See DragAndDropMechanism for an overview.

Content and Attributes

Attribute	Description
actions	Actions to be performed when items are dropped
class	Optional CSS classes to assign to the draggable element. The class `tc-droppable` is added automatically, and the class `tc-dragover` is applied while an item is being dragged over the droppable element
tag	Optional tag to override the default of a "div" element when the widget is rendered in block mode, or a "span" element when it is rendered in inline mode
enable	ⓘ **Introduced in v5.1.22** Optional value "no" to disable the droppable functionality (defaults to "yes")
data-*	ⓘ **New in v5.3.2** Optional data attributes to be assigned to the HTML element
style.*	ⓘ **New in v5.3.2** Optional CSS properties to be assigned to the HTML element

Within the action string, there are two Variables generated by the DroppableWidget:

- The actionTiddler Variable contains the title of the item being dropped

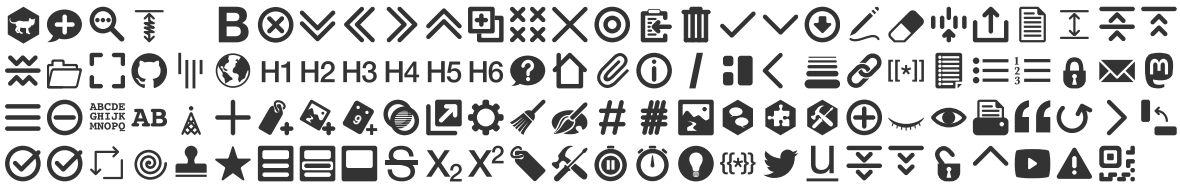
- The modifier Variable contains the Modifier Key held while dragging (can be normal, ctrl, shift or ctrl-shift)

If multiple items are dropped then the actions are performed repeatedly, once for each dropped item.

Note that the actionTiddler Variable holds a single, unquoted title. This is unlike the DraggableWidget which uses the same variable to pass a quoted Title List.

Examples

This example displays a palette of icons. Dragging a tiddler onto one of the icons assigns that icon to the tiddler.



Similarly, this example shows a palette of colours. Dragging a tiddler onto one of the colours assigns that colour to be used for rendering the icon of the tiddler.

DropzoneWidget

9th October 2021 at 2:22pm

TriggeringWidgets Widgets

Introduction

The dropzone widget creates an area into which the user can drag files and other objects. It also supports pasting via the clipboard, although browser support is currently limited.

It sends a WidgetMessage: tm-import-tiddlers carrying a JSON representation of the tiddlers to be imported up through its parents. This message usually trapped by the NavigatorWidget which adds the tiddlers to the store and updates the story to display them.

Content and Attributes

Attribute	Description
deserializer	ⓘ **Introduced in v5.1.15** Optional name of deserializer to be used (by default the deserializer is derived from the file extension)
enable	ⓘ **Introduced in v5.1.22** Optional value "no" to disable the dropzone functionality (defaults to "yes")
class	ⓘ **Introduced in v5.1.22** Optional CSS class to be assigned to the DOM node created by the dropzone (defaults to "tc-dropzone")
autoOpenOnImport	ⓘ **Introduced in v5.1.23** Optional value "no" or "yes" that can override tv-auto-open-on-import
importTitle	ⓘ **Introduced in v5.1.23** Optional tiddler title to use for import process instead of $:/Import

actions	ⓘ **Introduced in v5.2.0** Optional actions string to be invoked after the `tm-import-tiddlers` message has been sent. The variable `importTitle` provides the title of the tiddler used for the import process.
contentTypesFilter	ⓘ **Introduced in v5.2.0** Optional filter that specifies the content types accepted by the dropzone.
filesOnly	ⓘ **Introduced in v5.2.0** Optional. When set to "yes", the dropzone only accepts files and not string data. Defaults to "no"



Use the `prefix` filter operator to easily accept multiple related content types. For example this filter will accept all image content types: `[prefix[image/]]`



The list of available deserializers can be seen by using the filter `[deserializers[]]` in the Filter tab in **Advanced Search**.

The dropzone widget displays any contained content within the dropzone.

Display

The dropzone widget creates an HTML `<div class="tc-dropzone">` to contain its content. During a drag operation the class `tc-dragover` is added. CSS is used to provide user feedback. For custom styling of this state where a drag is in progress, specify a custom class name with the `class` attribute and use CSS with the selector `.myclass.tc-dragover`.

Data types supported

The following data transfer types are supported:

- **text/vnd.tiddler** - a list of tiddlers in JSON format (this format is generated by the LinkWidget)
- **text/plain** - plain text

- **text/uri-list** - a list of URIs as a single tiddler

dumpvariables Macro

21st February 2015 at 4:14pm

Core Macros Macros

The **dumpvariables** macro returns a table showing the values of all variables and macros that exist at that position in the widget tree.

It is useful for debugging and exploring TiddlyWiki's internals.

Placeholders are replaced with values in the normal way, but using the default values for all macro parameters.

Parameters

(none)

Examples

dumpvariables Macro (Examples)

21st February 2015 at 4:14pm

dumpvariables Macro Macro Examples

copy to clipboard

```
<$set name="EXAMPLE" value="123..789">
<$set name="EXAMPLE2" value="456">
```

```
<<dumpvariables>>

</$set>
</$set>
```

Try it

duplicateslugs Operator

9th May 2020 at 4:17pm

Filter Operators

purpose	find titles that yield duplicate slugs
input	a selection of titles
output	input titles that yield duplicate slugs

Learn more about how to use Filters

Introduced in v5.1.23 The `slugify` operator can be used to transform arbitrary tiddler titles into human readable strings suitable for use in URLs or filenames. However, it is possible for multiple different titles to slugify to the same string. The `duplicateslugs` operator can be used to display a warning. For example:

copy to clipboard

```
<$list filter="[!is[system]duplicateslugs[]limit[1]]" emptyMessage="There are no duplicate slugs">
The following tiddlers have duplicate slugs:

<ul>
<$list filter="[!is[system]duplicateslugs[]]">
<li><$link><$text text=<<currentTiddler>>/></$link></li>
</$list>
</ul>
</$list>
```

That renders as:

There are no duplicate slugs

Dynaview Plugin

13th November 2018 at 9:41am

OfficialPlugins

The Dynaview Plugin makes it possible to build user interfaces that dynamically respond to changes in the browser viewport via scrolling or zooming:

- CSS classes that allow rendering to be deferred until the output is scrolled into view

- CSS classes that allow the opacity of DOM elements to vary according to the current zoom level

See the demo at https://tiddlywiki.com/plugins/tiddlywiki/dynaview

each Operator

9th December 2019 at 9:59am

Filter Operators Group Operators

purpose	select one of each group of input titles by field
input	a selection of titles
suffix	optionally, `list-item` or `value`
parameter	**F** = the name of a field, defaulting to **title**
output	a selection containing the first input title encountered for each distinct value of field **F**

Learn more about how to use Filters

Each input title is processed in turn. The value of field **F** in the corresponding tiddler is examined.

each

As long as the value of the field is unique (i.e. has not been encountered before), the title is appended to the output.

each:list-item

The value is treated as a title list. Each title in the list considered in turn. If it has not been encountered before, it is appended to the output.

each:value

As long as the title is unique it is appended to the output whether or not the corresponding tiddler exists.

Note that if a tiddler does not contain field **F** , it is treated as if the value of the field were empty. Thus, a filter expression such as `[each[motovun]]` will return one tiddler that doesn't have a `motovun` field, as well as one tiddler with each distinct value of that field, if any. To obtain just the tiddlers that have a non-blank value for the `motovun` field one can use `[each[motovun]has[motovun]]`.

Examples

each Operator (Examples)

18th January 2015 at 7:31pm

each Operator Operator Examples

`[each[color]]`

Try it

`[sort[title]each[type]]`

→ the alphabetically first tiddler of each type

Try it

`[each:list-item[list]]`

→ all tiddlers listed anywhere in the core list field

Try it

```
[[Non existing]] [[GettingStarted]] +[each:value[]]
```

→ Compare this to `+[each[]]` below

Try it

```
[[Non existing]] [[GettingStarted]] +[each[]]
```

→ Compare this to `+[each:value[]]` above

Try it

For an example of using the **each** operator to generate a two-tier list of groups and members, see GroupedLists.

eachday Operator

10th February 2015 at 5:16pm

Date Operators Filter Operators Group Operators

purpose	select one of each group of input titles by date
input	a selection of titles

parameter	**F** = the name of a date field, defaulting to **modified**
output	a selection containing the first input title encountered for each distinct value (ignoring times of day) of field **F**

Learn more about how to use Filters

Each input title is processed in turn. The value of field **F** in the corresponding tiddler is examined, and as long as this indicates a date that has not been encountered before, the title is appended to the output.

If a tiddler doesn't contain field **F**, it contributes nothing to the output.

Examples

eachday Operator (Examples)

18th January 2015 at 7:31pm

eachday Operator Operator Examples

```
[eachday[created]]
```

→ the first tiddler created on each day

Try it

For an example of using the **eachday** operator to generate a two-tier list of groups and members, see GroupedLists.

Edición en Castellano

11th May 2016 at 6:02pm

Languages

La edición española de TiddlyWiki se encuentra aquí:

- **Documentación** : https://tiddlywiki.com/languages/es-ES/index.html
- **TiddlyWiki en blanco** : https://tiddlywiki.com/languages/es-ES/empty.html

Edit Template Body Cascade

6th December 2021 at 5:19pm

Cascades

The edit template body cascade is a cascade used by the default edit template to choose the template for displaying the tiddler body.

The core edit template body cascade can be found in **$:/core/ui/EditTemplate/body**

The default edit template body cascade consists of:

1. If the tiddler has the field **_canonical_uri** then use the template **$:/core/ui/EditTemplate/body/canonical-uri** to display the remote URL
2. Otherwise, use the template **$:/core/ui/EditTemplate/body/default** which provides the default editing interface

You can see the current settings for the view template body cascade in **$:/ControlPanel** under the **Info** -> **Advanced** -> **Cascades** -> **Edit Template Body** tab.

EditBitmapWidget

20th February 2015 at 5:21pm

Widgets

Introduction

The edit bitmap widget provides a user interface in the browser for editing bitmap tiddlers.

Content and Attributes

The content of the `<$edit-bitmap>` widget is ignored.

Attribute	Description
tiddler	The tiddler to edit (defaults to the current tiddler)

Configuration

The edit bitmap widget can be configured with these system tiddlers:

- **$:/config/BitmapEditor/LineWidth** determines the line width: 3px
- **$:/config/BitmapEditor/Colour** determines the line color:

Editing Tiddlers with Emacs

19th September 2014 at 6:15pm

Learning

Michael Fogleman has written an Emacs major mode called tid-mode, which is for editing TiddlyWiki .tid files. It is derived from text-mode, uses the useful minor modes org-struct and subword, and updates the modified times when you save a .tid file.

He also wrote two helper functions for using TiddlyWiki in Emacs. The first opens a tiddlers directory in Dired; the second opens TiddlyWiki in the browser.

```
(defun open-wiki ()
  "Opens a TiddlyWiki directory in Dired."
  (interactive)
  (dired "~/Dropbox/wiki/tiddlers/"))
```

```
(defun browse-wiki ()
  "Opens TiddlyWiki in the browser."
  (interactive)
  (browse-url "127.0.0.1:8080/"))
```

This latter function may require specifying a browser:

```
(setq browse-url-browser-function 'browse-url-generic
      browse-url-generic-program "chromium")
```

You can bind either of these functions with the global-set-key function:

```
(global-set-key (kbd "C-c w") 'open-wiki)
```

At the moment, these are not integrated into tid-mode.

Editing Tiddlers with Vim

11th January 2016 at 8:00pm

Learning

Devin Weaver has written a Vim plugin at https://github.com/sukima/vim-tiddlywiki

It supports most of the TiddlyWiki syntax and includes a filetype plugin which will auto update the `modified` field.

Édition en Français (France)

23rd November 2014 at 5:29pm

Languages

La traduction en Français (France) de TiddlyWiki démarre ici :

- **documentation** : https://tiddlywiki.com/languages/fr-FR/index.html
- **empty** : https://tiddlywiki.com/languages/fr-FR/empty.html

editiondescription Operator

3rd February 2015 at 7:45pm

Filter Operators Special Operators

purpose	select the descriptions of the input editions
input	a selection of edition names
parameter	none
output	the description string of each edition in the input

Learn more about how to use Filters

Each input title is processed in turn, ignoring any that is not the name of a TiddlyWiki edition.

Note

> This operator is *NOT* available when TiddlyWiki is running in a web browser.

Editions

7th January 2016 at 11:21pm

TableOfContents

TiddlyWiki is distributed in several distinct editions that are tuned for specific purposes.

An edition consists of the TiddlyWiki core components along with plugins, documentation and sample content to get you up and running as quickly as possible.

You can mix and match the components of these editions, to make a unique solution for your particular application.

- Empty Edition
- Blog Edition
- Full Edition
- Plugin Editions
- Résumé Builder Edition
- Text-Slicer Edition
- XLSX Utilities Edition



Note that editions should not be confused with the platforms on which TiddlyWiki runs (eg as a single HTML file vs. under Node.js)

editions Operator

3rd February 2015 at 7:46pm

Filter Operators Selection Constructors Special Operators

purpose	select the names of all the TiddlyWiki editions
input	ignored
parameter	none
output	the name of each TiddlyWiki edition, in alphabetical order

Learn more about how to use Filters

Note

This operator is *NOT* available when TiddlyWiki is running in a web browser.

EditionsCommand

29th October 2014 at 12:43pm

Commands

Lists the names and descriptions of the available editions. You can create a new wiki of a specified edition with the `--init` command.

```
--editions
```

Editor toolbar

18th January 2019 at 5:13pm

Features Formatting text in TiddlyWiki

How it Works

The editor toolbar is a toolbar that helps you format text easily. It appears above the text input field in a tiddler when in edit mode. It has a similar appearance to desktop text editors like Microsoft Word or Libre Office Write.

What the Editor toolbar does

When you press the button for a function, it will insert the WikiText in the text field. As an example, if you press the bold button it will insert `''''`.

If you **highlight** a piece of text the markup code will automatically go around the highlighted text.

If you are in another mode, like MarkDown, the toolbar will change to that syntax and insert appropriate code. However It will not change previously typed text or markup.

Besides the standard WikiText formatting, the Editor toolbar has the following buttons:

- Height of text editor
- Insert link
- Insert picture
- More actions
- Text preview
- Using Excise
- Using Stamp

EditTextWidget

22nd January 2023 at 10:00pm

Widgets

Introduction

The edit text widget provides a user interface in the browser for editing text tiddler fields. The editing element is dynamically bound to the underlying tiddler value: changes to the tiddler are instantly reflected, and any edits are instantly propagated.

By default, applying the EditTextWidget to the `text` field of a tiddler will generates an HTML `<textarea>` element, i.e. a multi-line editor.

Applying the EditTextWidget to any other field generates an HTML `<input type="text">` element, a single-line editor. This behaviour can be overridden with the `tag` and `type` attributes.

Content and Attributes

The content of the `<$edit-text>` widget is ignored.

Attribute	Description
tiddler	The tiddler to edit (defaults to the current tiddler)
field	The field to edit (defaults to `text`). Takes precedence over the `index` attribute
index	The index to edit
default	The default text to be provided when the target tiddler doesn't exist
class	A CSS class to be assigned to the generated HTML editing element
placeholder	Placeholder text to be displayed when the edit field is empty
focusPopup	Title of a state tiddler for a popup that is displayed when the editing element has focus
focus	Set to "yes" or "true" to automatically focus the editor after creation
focusSelectFromStart	ⓘ **Introduced in v5.2.6** If the `focus` attribute is enabled, determines the position of the start of the selection: `0` (default) places the start of the selection at the beginning of the text, `1` places the start of the selection after the first character, etc.
focusSelectFromEnd	ⓘ **Introduced in v5.2.6** If the `focus` attribute is enabled, determines the position of the end of the selection: `0` (default) places the end of the selection at the end of the text, `1` places the start of the selection before the final character, etc.
tabindex	Sets the `tabindex` attribute of the input or textarea to the given value

autocomplete	ⓘ **Introduced in v5.1.23** An optional string to provide a hint to the browser how to handle autocomplete for this input
tag	Overrides the generated HTML editing element tag. For a multi-line editor use `tag=textarea`. For a single-line editor use `tag=input`
type	Overrides the generated HTML editing element `type` attribute
size	The size of the input field (in characters). This exact result depends on browser and font. Use the `class` attribute to style width for precise control
autoHeight	Either "yes" or "no" to specify whether to automatically resize `textarea` editors to fit their content (defaults to "yes")
minHeight	Minimum height for automatically resized `textarea` editors, specified in CSS length units such as "px", "em" or "%"
rows	Sets the rows attribute of a generated textarea
cancelPopups	ⓘ **Introduced in v5.1.23** if set to "yes", cancels all popups when the input gets focus
inputActions	ⓘ **Introduced in v5.1.23** Optional actions that are triggered every time an input event occurs within the input field or textarea. ⓘ **Introduced in v5.2.1** The variable `actionValue` is available to the `inputActions` and contains the value of the input field.
refreshTitle	ⓘ **Introduced in v5.1.23** An optional tiddler title that makes the input field update whenever the specified tiddler changes
disabled	ⓘ **Introduced in v5.1.23** Optional, disables the text input if set to "yes". Defaults to "no"
fileDrop	ⓘ **Introduced in v5.2.0** Optional. When set to "yes" allows dropping or pasting images into the editor to import them. Defaults to "no"

Examples

If you wanted to change the field *myconfig* of the tiddler *AppSettings*, you could use an EditTextWidget to edit the field, and then show the result

anywhere else by using `{{AppSettings!!myconfig}}`. Note that this will create tiddler *AppSettings* if it doesn't already exist.

copy to clipboard

```
<$edit-text tiddler="AppSettings" field="myconfig"/><p/>
Value of ''myconfig'' : {{AppSettings!!myconfig}}
```

Try it

Text Selection

If the edit field already contains text or a default value is provided, you can use the `focusSelectFromStart` and `focusSelectFromEnd` attributes to only select part of the text when using `focus="yes"`.

Partial selection when editing this tiddler's *caption* field:

copy to clipboard

```
<$edit-text tiddler=<<currentTiddler>> field="caption" focus="yes" focusSelectFromStart="5" />
```

Try it

Provide a dated heading for this example where only the placeholder (but not the date) is selected for easier text input:

copy to clipboard

```
<$edit-text tiddler=<<currentTiddler>> field="heading" size="25" focus="yes" focusSelectFromEnd="13"
default={{{ [[Heading Text ()] [<now YYYY-0MM-0DD>] [()]] +[join[]] }}} />
```

Try it

EditWidget

9th October 2021 at 2:16pm

TriggeringWidgets Widgets

Introduction

The edit widget provides a general purpose interface for editing a tiddler. It dynamically chooses the appropriate widget depending on the type of the tiddler (currently either the EditTextWidget or the EditBitmapWidget).

Content and Attributes

The content of the `<$edit>` widget is ignored.

Attribute	Description
tiddler	The tiddler to edit (defaults to the current tiddler)

field	The field to edit (defaults to `text`). Takes precedence over the `index` attribute
index	The index to edit
class	A CSS class to be added the generated editing widget
tabindex	Sets the `tabindex` attribute of the input or textarea to the given value
autocomplete	ⓘ **Introduced in v5.1.23** An optional string to provide a hint to the browser how to handle autocomplete for this input
cancelPopups	ⓘ **Introduced in v5.1.23** if set to "yes", cancels all popups when the input gets focus
inputActions	ⓘ **Introduced in v5.1.23** Optional actions that are triggered every time an input event occurs within the input field or textarea
refreshTitle	ⓘ **Introduced in v5.1.23** An optional tiddler title that makes the input field update whenever the specified tiddler changes

Else Filter Run Prefix

22nd March 2023 at 3:07pm

Named Filter Run Prefix

purpose	the filter run is only evaluated if the filter output of all previous runs so far is an empty list
input	all titles from previous filter runs
output	if the filter output so far is an empty list then the output titles of the run are dominantly appended to the filter's output. if the filter output so far is not an empty list then the run is ignored.

ⓘ **Introduced in v5.1.23**

TypeError: setting getter-only property "parentNode"

This prefix has a Shortcut Filter Run Prefix symbol `~run`

else Operator

2nd August 2019 at 1:39pm

Conditional Operators Filter Operators

purpose	if the list of input titles is empty then return a list consisting of a single constant string, otherwise return the original titles
input	a selection of titles
parameter	**E** = a string
output	the original input titles unless empty, in which case return a list with the single entry **E**

Learn more about how to use Filters

Introduced in v5.1.20 See Conditional Operators for an overview.

Examples

else Operator (Examples)

2nd August 2019 at 1:35pm

else Operator Operator Examples

```
[[HelloThereMissing]is[missing]else[yes]]
```

Try it

```
[[HelloThere]is[missing]else[yes]]
```

Try it

Emergency Tiddler Export

7th May 2020 at 10:28pm

Android Chrome Edge Firefox Internet Explorer Linux Mac Opera Safari Saving Windows

This method is useful if, for any reason, you should find your current TiddlyWiki instance is not saving (e.g. a plugin or a server has stopped working). It should work on just about any platform.

- Go to advanced search
  - Goto the filter tab
  - Enter the following filter text:

```
[!is[system]!sort[modified]limit[25]]
```

- Check the list of tiddlers.
- Adjust the number "25" in the filter to make sure you found all your recently modified tiddlers
- Press the bucket with the up arrow [⇧] which appears on the right
- A dialogue window will ask for a location to download a file called tiddler.json on your local drive, or depending on browser configuration, just alert you that such a file will be downloaded. Press save.
- The `tiddlers.json` file can be imported (tools in sidebar) or drag and drop the file on the top line of the story river of another TW .
  - You can (de)select specific tiddlers.
  - Finally, press `import`.

Empty Edition

2nd July 2016 at 10:42am

Editions

The "empty" edition of TiddlyWiki is a vanilla distribution, with no additional plugins or configuration beyond the core code.

The empty edition can be downloaded from:

- Link: empty.html
- Built-in: Download

Your browser may ask you to accept the download before it begins

encodebase64 Operator

Filter Operators String Operators

purpose	apply base 64 encoding to a string
input	a selection of titles
suffix	optional: `binary` to treat input as binary data, `urlsafe` for URL-safe output
output	the input with base 64 encoding applied

Learn more about how to use Filters

Introduced in v5.2.6

See Mozilla Developer Network for details of base 64 encoding. TiddlyWiki uses library code from @nijikokun to handle the conversion.

The input strings are interpreted as UTF-8 encoded text (or binary data instead if the `binary` suffix is present). The output strings are base64 encoded.

The optional `binary` suffix, if present, causes the input string to be interpreted as binary data instead of text. Normally, an extra UTF-8 encoding step will be added before the base64 output is produced, so that emojis and other Unicode characters will be encoded correctly. If the input is binary data, such as an image, audio file, video, etc., then the UTF-8 encoding step would produce incorrect results, so using the `binary` suffix causes the UTF-8 encoding step to be skipped.

The optional `urlsafe` suffix, if present, will use the alternate "URL-safe" base64 encoding, where `-` and `_` are used instead of `+` and `/` respectively, allowing the result to be used in URL query parameters or filenames.

Examples

encodehtml Operator

17th October 2016 at 5:30pm

Filter Operators String Operators

purpose	apply HTML encoding to a string
input	a selection of titles
output	the input with HTML encoding applied

Learn more about how to use Filters

Introduced in v5.1.14

"HTML encoding" means replacing special HTML characters with the corresponding HTML entity:

- `&` replaced with `&`
- `<` replaced with `<`
- `>` replaced with `>`
- `"` replaced with `"`

Examples

encodehtml Operator (Examples)

17th October 2016 at 5:53pm

encodehtml Operator Operator Examples

```
[[Title with <angle brackets>]] +[encodehtml[]]
```

Try it

```
[[Title with an & ampersand]] +[encodehtml[]]
```

Try it

encodeuri Operator

17th October 2016 at 5:28pm

Filter Operators String Operators

purpose	apply URI encoding to a string
input	a selection of titles
output	the input with URI encoding applied

Learn more about how to use Filters

Introduced in v5.1.14

See Mozilla Developer Network for details of the encodeURI operation.

Examples

encodeuri Operator (Examples)

17th October 2016 at 5:53pm

encodeuri Operator Operator Examples

```
[[Title with Space]] +[encodeuri[]]
```

Try it

```
[[Title with Space]] [[Another title with Space]] +[encodeuri[]]
```

Try it

encodeuricomponent Operator

19th January 2023 at 6:43pm

Filter Operators String Operators

purpose	apply URI component encoding to a string
input	a selection of titles
output	the input with URI component encoding applied

Learn more about how to use Filters

Introduced in v5.1.14

See Mozilla Developer Network for details of the encodeURIComponent operation.

Introduced in v5.2.6 In addition to the characters mentioned in the article above, the following additional characters are also percent encoded: `!'()*`

Examples

encodeuricomponent Operator (Examples)

17th October 2016 at 5:53pm

encodeuricomponent Operator Operator Examples

```
[[Title with Space]] +[encodeuricomponent[]]
```

Try it

```
[[Title with Space]] [[Another title with Space]] +[encodeuricomponent[]]
```

Try it

Encryption

10th June 2016 at 10:33am

Features Working with TiddlyWiki

When used as a single HTML file, TiddlyWiki5 allows content to be encrypted using the Stanford JavaScript Crypto Library.

1. Switch to the **Tools** tab in the sidebar and look for the button with a padlock icon
2. If the button is labelled **set password** then the current wiki is not encrypted. Clicking the button will prompt for a password that will be used to encrypt subsequent saves
3. If the button is labelled **clear password** then the current wiki is already encrypted. Clicking the button will remove the password so that subsequent saves will be unencrypted
4. Optionally, open the saved file in a text editor and verify that your data is encrypted
5. Open the file in your browser. You will be prompted for a password before the content is displayed

Note that TiddlyWiki has two other unrelated features concerned with passwords/encryption:

- The ability to set a password when saving to *TiddlySpot*. This is done in the "Saving" tab of **control panel** .
- The ability to use standard HTTP basic authentication with the Node.js server configuration. This is done on the command line with the ServerCommand. Combined with SSL, this gives the same level of transit encryption as you'd get with online services like Google or Dropbox, but there is no encryption of data on disk

EncryptWidget

30th October 2013 at 9:37am

Widgets

Introduction

The encrypt widget renders a filtered list of tiddlers to an encrypted block with the password currently held in the *PasswordVault*. The encrypted block can subsequently be decrypted by the TiddlyWiki5 *BootMechanism*. See the *EncryptionMechanism* for more details.

Content and Attributes

The content of the `<$encrypt>` widget is ignored.

Attribute	Description
filter	Filter defining the tiddlers to be included in the encrypted block. If not specified then all non-system tiddlers are used

enlist Operator

5th May 2021 at 12:02pm

Field Operators Filter Operators Negatable Operators Selection Constructors

purpose	select titles from the parameter interpreted as a title list
input	ignored
`!` **input**	a selection of titles
suffix	`D` = ⓘ **Introduced in v5.1.20** `dedupe` (the default) to remove duplicates, `raw` to leave duplicates untouched
parameter	`L` = a title list
output	the titles stored as a title list at `L`

! **output** those input titles that are *NOT* listed in **L**

Learn more about how to use Filters



Literal filter parameters cannot contain square brackets but you can work around the issue by using a variable. **Learn more at:** SetWidget documentation under the heading "Filtered List Variable Assignment"

```
<$set name="myList" value="first [[second with a space]] third">
<$list filter="[enlist<myList>]">
...
```

Examples

enlist Operator (Examples)

10th June 2019 at 6:53pm

enlist Operator Operator Examples

```
[enlist[one two three]addsuffix[!]]
```

Try it

```
[enlist[one one one]]
```

Try it

```
[enlist:raw[one one one]]
```

Try it

```
[enlist{$:/StoryList!!list}]
```

Try it

enlist-input Operator

2nd November 2020 at 11:18pm

Filter Operators String Operators

purpose	select titles by interpreting each input title as a title list
input	a selection of titles
suffix	**D** = `dedupe` (the default) to remove duplicates, `raw` to leave duplicates untouched
output	the titles stored as a title list in each input title

Learn more about how to use Filters

Introduced in v5.1.23

Compare with enlist operator that interprets its parameter as a title list

```
<$vars days={{{ [[Days of the Week]get[list]] }}}>

{{{ [enlist<days>] }}}

</$vars>
```

is equivalent to:

```
{{{ [[Days of the Week]get[list]enlist-input[]] }}}
```

Examples

enlist-input Operator (Examples)

3rd November 2020 at 10:06am

enlist-input Operator Operator Examples

```
[[Days of the Week]get[list]enlist-input[]]
```

Try it

Contrast with:

```
[[Days of the Week]get[list]]
```

Try it

```
'HelloThere GettingStarted [[Customise TiddlyWiki]]' +[enlist-input[]]
```

Try it

Contrast with:

```
'HelloThere GettingStarted [[Customise TiddlyWiki]]'
```

Try it

```
'HelloThere HelloThere GettingStarted [[Customise TiddlyWiki]]' +[enlist-input:raw[]]
```

Try it

Contrast with:

```
'HelloThere HelloThere GettingStarted [[Customise TiddlyWiki]]' +[enlist-input[]]
```

Try it

EntityWidget

14th March 2016 at 5:56pm

Widgets

Introduction

The entity widget displays a specified HTML character entity.

Content and Attributes

The content of the `<$entity>` widget is not used.

Attribute	Description
entity	The entity to display (eg, `×`)

Environment Variables on Node.js

13th June 2022 at 1:41pm

TiddlyWiki on Node.js

TiddlyWiki on Node.js supports the following OS environment variables for specifying a delimited list of paths to search for plugins and editions:

- `TIDDLYWIKI_PLUGIN_PATH` - Search path for ordinary plugins
- `TIDDLYWIKI_THEME_PATH` - Search path for themes
- `TIDDLYWIKI_LANGUAGE_PATH` - Search path for languages
- `TIDDLYWIKI_EDITION_PATH` - Search path for editions (used by the InitCommand)

Note

> **1.** The delimiter may vary between operating systems. While on Windows a semicolon `;` is used, Linux implements a colon `:`.
>
> **2.** On Linux systems, it may be necessary to ***export*** the variable as well as to define it.

The additional paths should each point to folders structured like the equivalent directories in the TiddlyWiki5 GitHub repository: the plugin, theme and language directories contain `publisher/pluginname/<files>` while the edition directories contain `editionname/<files>`

For example:

```
export TIDDLYWIKI_PLUGIN_PATH=~/MyPluginStore
tiddlywiki mywiki --build index
```

ErrorWidget

19th April 2023 at 12:31pm

Widgets

Introduced in v5.2.4 The `$error` widget is used by the core to display error messages such as the recursion errors reported by the `$transclude` widget.

The `$error` does not provide any useful functionality to end users. It is only required by the core for technical reasons.

Content and Attributes

The content of the `$error` widget is ignored.

Attribute	Description
$message	The error message

escapecss Operator

29th June 2019 at 9:28am

Filter Operators String Operators

purpose	apply CSS escaping to a selection of titles through the CSS.escape() method / operation
input	a selection of titles
output	the input with CSS escaping applied through the CSS.escape() method

Learn more about how to use Filters

Introduced in v5.1.20

See Mozilla Developer Network for details of the CSS.escape() operation.

Examples

escapecss Operator (Examples)

29th June 2019 at 9:28am

escapecss Operator Operator Examples

```
[all[tiddlers]tag[Resources]escapecss[]]
```

Try it

escaperegexp Operator

17th October 2016 at 5:32pm

Filter Operators String Operators

purpose	escape special characters used in regular expressions
input	a selection of titles
output	the input with escaping applied to special regular expression characters

Learn more about how to use Filters

Introduced in v5.1.14

See Mozilla Developer Network for details of the regular expression syntax.

Examples

escaperegexp Operator (Examples)

17th October 2016 at 5:55pm

escaperegexp Operator Operator Examples

```
[[Title with dots . and dollars $]] +[escaperegexp[]]
```

Try it

Eucaly's Tiddly World

6th January 2021 at 4:10pm

Community Plugins

I collect my tw5 creations on-line as http://eucaly-tw5.tiddlyspot.com/

Available Plugins :

- *TitleMe* - Return in-place tiddler title, to link / show / edit the source tiddler
- *QuickJump* - Quick jump to tiddlers in story list
- *FlexWidth* - Flexibly tweak sidebar width for TiddlyWiki 5 (Inspired by TW5 tristate Sidebar)
- *MatchFilter* - returns matching text instead of a list of the tiddlers from *FieldFilter*
- *PopupTagger* - *TagBar* / *TagTable* with popup list of tags (Inspired by *MonkeyTaggerMacro* for TiddlyWiki classic)

EventCatcherWidget

12th October 2022 at 9:42pm

TriggeringWidgets Widgets

Introduction

ⓘ Introduced in v5.1.23

This is an advanced widget intended for use by those familiar with HTML, CSS and JavaScript handling of DOM events.

The event catcher widget traps DOM-initiated Javascript events dispatched within its child content, and allows invoking a series of ActionWidgets in response to those events.

In order for the events to be trapped:

- The event must be of one of the events specified in the event catcher widget's `events` attribute
- The event must target a DOM node with an ancestor that matches the widget's `selector` attribute
- ⓘ **Introduced in v5.2.2** Optionally, the DOM node targeted by the event must also match the widgets `matchSelector` attribute
- The event must support event bubbling

Use of the event catcher widget is beneficial when using large numbers of other trigger widgets such as the ButtonWidget is causing performance problems. The workflow it enables is akin to what is referred to as "event delegation" in JavaScript parlance.

Content and Attributes

The content of the `<$eventcatcher>` widget is displayed normally.

Attribute	Description
selector	A CSS selector. Only events originating inside a DOM node with this selector will be trapped
matchSelector	ⓘ **Introduced in v5.2.2** An optional CSS selector. Only events targeting DOM nodes matching this selector will be trapped
{any attributes starting with $}	ⓘ **Introduced in v5.2.0** Each attribute name (excluding the $) specifies the name of an event, and the value specifies the action string to be invoked. For example: `$click=<<clickActions>>`
tag	Optional. The HTML element the widget creates to capture the events, defaults to: » `span` when parsed in inline-mode » `div` when parsed in block-mode
class	Optional. A CSS class name (or names) to be assigned to the widget HTML element
stopPropagation	ⓘ **Introduced in v5.2.0** Optional. Set to "always" to always stop event propagation even if there are no corresponding actions to invoke, "never" to never stop event propagation, or the default value "onaction" with which event propagation

	is only stopped if there are corresponding actions that are invoked.
events	*(deprecated – see below)* Space separated list of JavaScript events to be trapped, for example "click" or "click dblclick"
actions-*	*(deprecated – see below)* Action strings to be invoked when a matching event is trapped. Each event is mapped to an action attribute name of the form `actions-event` where *`event`* represents the type of the event. For example: `actions-click` or `actions-dblclick`

Introduced in v5.2.0 Note that the attributes `events` and `actions-*` are no longer needed. Instead you can use attributes starting with $ where the attribute name (excluding the $) specifies the name of the event and the value specifies the action string to be invoked. If any attributes with the prefix $ are present then the `types` attribute is ignored.

Variables

The following variables are made available to the actions:

Variables	Description
`dom-*`	All DOM attributes of the node matching the given selector are made available as variables, with the prefix `dom-`
`modifier`	The modifier Variable contains the Modifier Key held during the event (can be "normal", "ctrl", "shift", "alt" or combinations thereof)
`event-mousebutton`	The mouse button (if any) used to trigger the event (can be "left", "right" or "middle"). Note that not all event types support the mousebutton property
`event-type`	The type property of the JavaScript event
`event-detail-*`	Any properties in the detail attribute of the event are made available with the prefix `event-detail-`
`tv-popup-coords`	A relative co-ordinate string that can be used with the ActionPopupWidget to trigger a popup at the DOM node matching the selector where the event originated (see Coordinate Systems for more information)

`tv-popup-abs-coords`	ⓘ **Introduced in v5.2.4** An absolute co-ordinate string that can be used with the ActionPopupWidget to trigger a popup at the DOM node matching the selector where the event originated (see Coordinate Systems for more information)
`tv-widgetnode-width`	ⓘ **Introduced in v5.2.3** `offsetWidth` of the DOM node created by the eventcatcher widget
`tv-widgetnode-height`	ⓘ **Introduced in v5.2.3** `offsetHeight` of the DOM node created by the eventcatcher widget
`tv-selectednode-posx`	`x` offset position of the selected DOM node
`tv-selectednode-posy`	`y` offset position of the selected DOM node
`tv-selectednode-width`	`offsetWidth` of the selected DOM node
`tv-selectednode-height`	`offsetHeight` of the selected DOM node
`event-fromselected-posx`	`x` position of the event relative to the selected DOM node
`event-fromselected-posy`	`y` position of the event relative to the selected DOM node
`event-fromcatcher-posx`	`x` position of the event relative to the event catcher DOM node
`event-fromcatcher-posy`	`y` position of the event relative to the event catcher DOM node
`event-`	ⓘ **Introduced in v5.2.0** `x` position of the event relative to the viewport

`fromviewport-posx`	
`event-fromviewport-posy`	ⓘ **Introduced in v5.2.0** `y` position of the event relative to the viewport

Example

This example uses the ActionLogWidget and will log the `data-item-id` attribute of the clicked DOM node to the browser's JavaScript console

```
\define clickactions()
<$action-log item=<<dom-data-item-id>> event=<<event-type>>/>
\end

\define contextmenu-actions()
<$action-log item=<<dom-data-item-id>> event=<<event-type>>/>
\end

<$eventcatcher selector=".item" $click=<<clickactions>> $contextmenu=<<contextmenu-actions>> tag="div">

<div class="item" data-item-id="item1">
Click events here will be trapped
</div>

<div class="item" data-item-id="item2">
And here too
```

```
</div>

<div data-item-id="item3">
Not here
</div>

<div class="item" data-item-id="item4">
And here
</div>

</$eventcatcher>
```

Example config-tiddlyweb-host for IIS

28th March 2018 at 4:52pm

Saving

```
title: $:/config/tiddlyweb/host
text: $protocol$//$host$/MyApp/
```

Example for tag Macro

16th November 2021 at 4:49am

Macro Examples tag Macro (Examples)

This tiddler exists to demonstrate the **tag** macro.

Example package.json for IIS

28th March 2018 at 4:52pm

Saving

```
{
    "name": "MyStuff",
    "description": "A description of this wiki",
    "dependencies": {
        "sax": "1.2.4",
        "tiddlywiki": "*"
    }
}
```

Example Table of Contents: Expandable

21st February 2015 at 9:37pm

table-of-contents-example

copy to clipboard

```
<div class="tc-table-of-contents">
<<toc-expandable "Contents">>
</div>
```

Try it

Example Table of Contents: Selectively Expandable

21st February 2015 at 9:39pm

table-of-contents-example

copy to clipboard

```
<div class="tc-table-of-contents">
<<toc-selective-expandable "Contents">>
</div>
```

Try it

Example Table of Contents: Selectively Expandable (custom icons)

19th August 2022 at 10:10am

table-of-contents-example

Introduced in v5.2.4 It is possible to define the `toc-open-icon` and `toc-closed-icon` variable, to change the icons used by the toc macros. This setting works for all toc-macro variants

copy to clipboard

```
<$let toc-open-icon="$:/core/images/fold-button" toc-closed-icon="$:/core/images/folder">
<div class="tc-table-of-contents">
<<toc-selective-expandable "Contents">>
</div>
</$let>
```

Try it

Example Table of Contents: Simple

21st February 2015 at 9:37pm

table-of-contents-example

copy to clipboard

```
<div class="tc-table-of-contents">
<<toc "Contents">>
</div>
```

Try it

Example Table of Contents: Sorted Expandable

21st February 2015 at 9:39pm

table-of-contents-example

copy to clipboard

```
<div class="tc-table-of-contents">
<<toc-expandable "Contents" "sort[title]">>
</div>
```

Try it

Example Table of Contents: Tabbed External

21st February 2015 at 10:00pm

table-of-contents-example

HelloThere
Learning
Working with TiddlyWiki

Select a topic in the table of contents. Click the arrow to expand a topic.

- Customise TiddlyWiki
- Features
- Filters
- Languages
- Editions
- Plugins
- Platforms
- Reference
- Community
- About

```
<$macrocall
    $name="toc-tabbed-external-nav"
    tag="TableOfContents"
    selectedTiddler="$:/temp/toc/selectedTiddler"
    unselectedText="<p>Select a topic in the table of contents. Click the arrow to expand a topic.</p>"
    missingText="<p>Missing tiddler.</p>"
/>
```

Example Table of Contents: Tabbed Internal

21st February 2015 at 10:00pm

table-of-contents-example

- HelloThere
- Learning
- Working with TiddlyWiki
- Customise TiddlyWiki
- Features
- Filters
- Languages
- Editions
- Plugins
- Platforms
- Reference
- Community
- About

Select a topic in the table of contents. Click the arrow to expand a topic.

```
<$macrocall
    $name="toc-tabbed-internal-nav"
    tag="TableOfContents"
    selectedTiddler="$:/temp/toc/selectedTiddler"
    unselectedText="<p>Select a topic in the table of contents. Click the arrow to expand a topic.</p>"
```

```
    missingText="<p>Missing tiddler.</p>"
/>
```

Example tiddlywiki.info for IIS

28th March 2018 at 5:12pm

Saving

```
{
    "description": "My wiki",
    "plugins": [
        "tiddlywiki/tiddlyweb",
        "tiddlywiki/filesystem"
    ],
    "themes": [
        "tiddlywiki/vanilla",
        "tiddlywiki/snowwhite"
    ]
}
```

Example web.config for IIS

1st July 2018 at 8:52pm

Saving

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<configuration>
    <system.webServer>
        <handlers>
            <add
                name="httpplatformhandler"
                path="*"
                verb="*"
                modules="httpPlatformHandler"
                resourceType="Unspecified"
                requireAccess="Script" />
        </handlers>
        <httpPlatform
           stdoutLogEnabled="true"
           stdoutLogFile=".\node.log"
           startupTimeLimit="20"
           processPath="C:\Program Files\nodejs\node.exe"
           arguments=".\node_modules\tiddlywiki\tiddlywiki.js ./wiki --listen port=PORT path-prefix=/
MyApp">
            <environmentVariables>
                <environmentVariable name="PORT" value="%HTTP_PLATFORM_PORT%" />
                <environmentVariable name="NODE_ENV" value="Production" />
            </environmentVariables>
        </httpPlatform>
    </system.webServer>
</configuration>
```

Examples

2nd June 2016 at 7:27pm

Community HelloThere

This collection showcases inspiring and interesting examples of TiddlyWiki being used in the wild. Submit new entries to this collection via GitHub, Twitter or by posting in the TiddlyWiki Groups.

"Lucky Sushi" online shop by sini-Kit

Posted 3 years ago

A complete online shop made in TiddlyWiki!

http://luckysushi.ru/habarovsk/heeg.html#index

"Gospel Bubbles" by Rev. David Gifford

Posted 3 years ago

Creates bubble maps of Biblical texts.

http://giffmex.org/gospels.bubbles.html#Gospels%20Bubbles

> This TiddlyWiki, which is still under construction, has several features that together make it lightning fast to find passages, types of passages, themes, people, places and images in the New Testament Gospels. The bubblemaps are color-coded overviews of each Gospel. Each type of passage (miracle story, parable, etc) was assigned a color. Bubblemaps allow you to do three things:
>
> 1. The **bubblemaps** are color-coded overviews of each Gospel. Each type of passage (miracle story, parable, etc) was assigned a

> color. Bubblemaps allow you to do three things:
>
> 1. See how the different types of passages are distributed throughout each Gospel.
> 2. Hover over passages to see the verses and titles.
> 3. Click on a passage to open it and see the themes and other data it mentions.
>
> 2. The **themes by passage** indexes are lists of the passages in each Gospel, followed by the themes and other data found in the passage. Clicking on a theme calls up a list of all the passages that mention that theme.
> 3. The **indexes of themes** contain lists of links to each theme, person, group, place and image. Clicking on a theme calls up a list of all the passages that mention that theme.
> 4. There is a **search window** hidden in the right hand menu. Use the search window to find passages quickly, by typing things like sower or paralytic. Click the arrows in the upper right of the screen to open the menu and search.

TriTarget.org by Devin Weaver

Posted 7 years ago

A personal website built with TiddlyWiki.

https://tritarget.org

> *TriTarget*.org is a name I came up with in high school. I was big into computer programming and wanted a company name that was kinda catchy. When I made a few programs in BASIC I would brand them with *TriTarget* as a way to show off my company (Even though I didn't have one).

Interactive Git Documentation by Devin Weaver

Posted 8 years ago

A git choose-your-own-adventure!ⓡ for walking yourself though fixing a *broken* Git repository.

http://sukima.github.io/GitFixUm/

> This document is an attempt to be a fairly comprehensive guide to recovering from what you did not mean to do when using git. It isn't that git is so complicated that you need a large document to take care or your particular problem, it is more that the set of things that you might have done is so large that different techniques are needed depending on exactly what you have done and what you want to have happen.

PESpot Lesson Planner by Patrick Detzner

Posted 8 years ago

A system for lesson planning built with TiddlyWiki.

http://pespot.tiddlyspot.com

> I built a program to help enter and store lesson plan information and then facilitate/automate some of the tedious work involved in creating the actual lesson plan document.

Simple Zork-like Game by Jed Carty

Posted 9 years ago

A simple game built with TiddlyWiki, introduced in this post.

http://zorklike.tiddlyspot.com

> I decided to see if it was possible to make some sort of game using only the core tiddlywiki with no plugins or javascript. I made a very bare bones zork/interactive fiction type game. It is currently the simplest thing I could make and claim it was a game, but I may add on to it in the future. It uses the 5.1.5 prerelease because the action-setfield widget saved lots of work making macros.
>
> One goal of this is to use nothing besides what is contained in the core tiddlywiki, so there is no javascript and no plugins.

"PETTIL - Forth for the Commodore PET" by Charlie Hitselberger

Posted 9 years ago

A fast Forth interpreter for the Commodore PET, written in 6502 assembly language. The TiddlyWiki containing program documentation is automatically generated from the source code: see https://github.com/chitselb/pettil.

http://pettilmobile.com/tw/

> Objectives of the project are, in no particular order:
>
> - make a Forth that runs on my Commodore PET 2001
> - have fun
> - improve my "6502 assembly golf" skills
> - find other people who are interested in this project

"A Thesis Notebook" by Alberto Molina

Posted 11 years ago

A thesis notebook based on TiddlyWiki.

http://tesis.tiddlyspot.com/

> This is an example of a thesis notebook powered by TiddlyWiki 5.0.8-beta.
>
> TiddlyWiki is a great piece of software created by Jeremy Ruston. It allows you, among other things, to take notes, organise ideas, store information, and display all your stuff the way you want. It is an incredibly flexible tool you can adapt to fit almost all your needs.
>
> This TiddlyWiki has been customized to serve as a philosophy notebook centered around authors, books and papers, concepts and theories, and personal notes. I use it along with Zotero, which is a dedicated bibliography software. Both are free, open source projects. TiddlyWiki can be downloaded at https://tiddlywiki.com.

Except Filter Run Prefix

22nd March 2023 at 3:06pm

Named Filter Run Prefix

purpose	if output titles of this filter run are contained in the output of previous filter runs, they are removed, and otherwise ignored
input	all titles from previous filter runs
output	output titles are removed from the filter's output (if such tiddlers exist)

Introduced in v5.1.23

TypeError: setting getter-only property "parentNode"

This prefix has a Shortcut Filter Run Prefix symbol `-run`

exponential Operator

11th June 2019 at 5:45pm

Binary Mathematics Operators Filter Operators Mathematics Operators

purpose	convert each number to exponential notation with **N** digits
input	a selection of titles
parameter	**N** = a number
output	the input as numbers converted to exponential notation with **N** digits

Learn more about how to use Filters

Introduced in v5.1.20 See Mathematics Operators for an overview.

The **exponential** operator returns a string representation of the input number in exponential notation with the specified number of digits. If a number has more digits than requested, the number is rounded to the nearest number represented by the specified number of digits

Examples

exponential Operator (Examples)

11th June 2019 at 5:45pm

exponential Operator Operator Examples

```
123456789 +[exponential[4]]
```

Try it

```
123456789 +[exponential[13]]
```

Try it

```
1.23E23 +[exponential[90]]
```

Try it

Extended Listops Filters

27th February 2023 at 6:55pm

Filters

Introduction

A number of filters are used to manipulate lists.

Some filters are designed to move items from the tail of the list and insert them at specified locations in the list, e.g. putafter or putbefore. Items

are often appended to the list before using these filters. In general, these filters accept a suffix specifying the number of items to move (default to 1.)

Some filters are designed to either add or remove from the list, a selected range of items from an array, e.g. append or remove. These filters are best used with a reference to an array, stored in a field or data index elsewhere in the wiki (they may be used with a simple list of items, provided the items do not include white space.) In general, these filters accept a suffix specifying the number of items to move (default to All.)

Listops Operators			
Operator	**Purpose**	✓	!
allafter	discard all items except those after the marker		
allbefore	discard all items except those before the marker		
append	append a range of items from an array to the list		
cycle	toggle the titles specified in the first parameter in a cyclical manner		
insertafter	insert an item **T** into a list immediately after an item **A**		
insertbefore	insert an item **T** into a list immediately before an item **B**		
move	move marker N places in the list		
prepend	prepend a range of items from an array to the list		
putafter	move N trailing items after the marker		
putbefore	move N trailing items before the marker		
putfirst	move N trailing items to the head of the list		
putlast	move N leading items to the tail of the list		
remove	remove a list of titles specified in the parameter from the input		
replace	replace marker with N trailing items		
sortby	sort the current list in the order of the list referenced in the parameter		

toggle	toggle the title specified in the parameter in the input		
unique	remove all duplicate items from the current list		

Examples

In this example we shall populate the 'DataIndex' index of the tiddler 'MyData' with the names of the days of the week, then clear this list.

copy to clipboard

```
<$button>
<$action-listops $tiddler="ListopsData" $index="DataIndex" $filter="[list[Days of the Week]]"/>
Get days-of-the-week
</$button>
<$button>
<$action-listops $tiddler="ListopsData" $index="DataIndex" $filter="[[]]"/>
Clear
</$button>

{{ListopsData}}
```

That renders as:

Get days-of-the-week Clear

```
DataIndex:
```

In this example we shall slice the populated list from the '*DaysOfTheWeek*' index of the tiddler 'MyData' in order to insert items before and after a marker item (Wednesday) that are first appended to the list.

copy to clipboard

```
<$button>
```

```
<$action-listops $tiddler="ListopsData" $index="DataIndex" $subfilter="one two +[putbefore:
2[Wednesday]]"/>
Put 2 Items Before Wednesday
</$button>
<$button>
<$action-listops $tiddler="ListopsData" $index="DataIndex" $subfilter="four five +[putafter:2[Wednesday]]
three +[putbefore[Wednesday]]"/>
Put One Item Before & Two Items After Wednesday
</$button>

{{ListopsData}}
```

That renders as:

Put 2 Items Before Wednesday | Put One Item Before & Two Items After Wednesday

```
DataIndex:
```

In this example we shall slice the populated list from the '*DaysOfTheWeek*' index of the tiddler 'MyData' in order to replace the marker item (Wednesday) with items which are first appended to the list. We shall then move 3 items to the head of the list which have first been appended to the list from referenced fields.

copy to clipboard

```
<$button>
```

```
<$action-listops $tiddler="ListopsData" $index="DataIndex" $subfilter="[[---o]] [[o---]] +[replace:2{!!
marker}]"/>
Replace '!!marker' with 2 Items
</$button>
<$button>
<$action-listops $tiddler="ListopsData" $index="DataIndex" $subfilter="[{!!item1}] [{!!item2}] [{!!item3}
] +[putfirst:3[]]"/>
Put 3 Items First
</$button>

{{ListopsData}}
```

That renders as:

Replace '!!marker' with 2 Items | Put 3 Items First

```
DataIndex:
```

In this example we shall slice the populated list from the '*DaysOfTheWeek*' index of the tiddler 'MyData' in order to append to the truncated list, items from a referenced field. We shall then remove the first two of the items added.

copy to clipboard

```
|list: |<$view field="list"/> |
```

```
<$button>
<$action-listops $tiddler="ListopsData" $index="DataIndex" $subfilter="+[allbefore:include[Wednesday]] +
[prepend{!!list}]"/>
Prepend '!!list' to items before 'Wednesday'
</$button>
<$button>
<$action-listops $tiddler="ListopsData" $index="DataIndex" $subfilter="+[remove:2{!!list}]"/>
Remove first two items in '!!list' from current list
</$button>
<$button>
<$action-listops $tiddler="ListopsData" $index="DataIndex" $subfilter="+[!remove:1{!!list}]"/>
Remove last item in '!!list' from current list
</$button>

{{ListopsData}}
```

That renders as:

list: Yesterday Today Tomorrow

Prepend '!!list' to items before 'Wednesday' Remove first two items in '!!list' from current list Remove last item in '!!list' from current list

```
DataIndex:
```

In this example we shall populate the list with numbers, then move items one by one from the head to the tail and from the tail to the head (best seen by clicking the lower buttons several times.)

This example illustrates that the append[] and prepend[] operators do not enforce unique instances of an item and that, with the next run, any duplicates are removed.

copy to clipboard

```
<$button>
<$action-listops $tiddler="ListopsData" $index="DataIndex" $filter="[[]]" $subfilter="+[append:3{!!
numbers}]"/>
Setup some numbers
</$button>
<$button>
<$action-listops $tiddler="ListopsData" $index="DataIndex" $subfilter="+[!append:6{!!numbers}]"/>
Append more numbers
</$button>

<$button>
<$action-listops $tiddler="ListopsData" $index="DataIndex" $subfilter="+[putfirst:2[]]"/>
Move last 2 items to the head
</$button>
<$button>
<$action-listops $tiddler="ListopsData" $index="DataIndex" $subfilter="+[putlast[]]"/>
Move the head to the last item
</$button>
```

```
{{ListopsData}}
```

That renders as:

Setup some numbers | Append more numbers

Move last 2 items to the head | Move the head to the last item

```
DataIndex:
```

External Attachments Plugin

31st October 2017 at 6:24pm

OfficialPlugins Plugin Editions

The External Attachments Plugin provides support for importing tiddlers as external attachments. That means that instead of importing binary files as self-contained tiddlers, they are imported as "skinny" tiddlers that reference the original file via the **_canonical_uri** field. This reduces the size of the wiki and thus improves performance. However, it does mean that the wiki is no longer fully self-contained.

This plugin only works when using TiddlyWiki with platforms such as TiddlyDesktop that support the **path** attribute for imported/dragged files.

ExternalImages

12th September 2014 at 4:13pm

Concepts Features

External images in TiddlyWiki are tiddlers that point to the URI of an image, rather than embedding the full image data. They can perform better than embedded images, particularly with large numbers or sizes of images. However, using them breaks the single file pattern of TiddlyWiki.

External images are used in the browser. They can be created by the Node.js configuration when it builds a TiddlyWiki, or they can be created manually within the browser.

What is an External Image

An external image is an ordinary image tiddler that has a **_canonical_uri** field containing the URI of the image. The URI can be absolute or relative to the HTML document. If the canonical URI is provided then the **text** field of the tiddler is ignored and so should be omitted.

Manually Creating External Images

To manually create an external image just create the tiddler with the appropriate image content type, and add a **_canonical_uri** field with a URI pointing to the actual image location.

IMPORTANT: Double-check your spelling. `_canonical_uri` is spelled URI, not URL.

Creating external images under Node.js

The following steps are used to create a static HTML file version of a wiki accompanied by an **images** folder containing the referenced external

images:

1. Create image tiddlers in your TiddlyWikiFolders in the usual way
2. Save the images as separate files (by convention, in a subfolder named **images**)
3. Externalise the image tiddlers by giving them a **_canonical_uri** field
4. Save the main HTML file

Note the image files must be saved before they are externalised. Externalising them destroys the **text** field within the in-memory copy of the wiki store, meaning that attempts to save them will fail.

For an example see the **externalimages** build target of the demo **tw5.com** wiki:

```
--savetiddlers [is[image]] images
--setfield [is[image]] _canonical_uri $:/core/templates/canonical-uri-external-image text/plain
--setfield [is[image]] text "" text/plain
--rendertiddler $:/core/save/all externalimages.html text/plain
```

Saving Separate Image Files

The following `--savetiddlers` command can be used to save the images of a wiki into an **images** subfolder:

```
--savetiddlers [is[image]] images
```

Externalising Image Tiddlers

Two `--setfield` commands are used: the first sets the **_canonical_uri** field to a URI derived from the title of the tiddler, and the second

clears the text field.

```
--setfield [is[image]] _canonical_uri $:/core/templates/canonical-uri-external-image text/plain
--setfield [is[image]] text "" text/plain
```

The template tiddler **$:/core/templates/canonical-uri-external-image** contains:

```
<!--

This template is used to assign the ''_canonical_uri'' field to external images.

Change the `./images/` part to a different base URI. The URI can be relative or absolute.

-->
./images/<$view field="title" format="doubleurlencoded"/>
```

Note that these operations modify the tiddlers in the wiki store and so may affect the operation of subsequent commands.

Using External Images

You can't edit an external image directly in the browser except by changing the URI field to point to a different image.

Favicon template.svg

picture



Favorites by Mohammad

17th November 2020 at 4:59pm

Community Plugins

Favorites plugin is a set of tool for creating favorites (bookmarks) in Tiddlywiki. Each favorite item is a shortcut to a tiddler.

https://kookma.github.io/TW-Favorites/

A favorite item also called a bookmark is a shortcut, you are creating for quick access to it. You can access that bookmark at any time and view that as many times as you want without having to search and find it again.

The Favorites plugin contains three modes

- flat mode: one favorite list, no folder
- structured mode: use folder, search tool, export and delete tool
- frozen list: read only, simple table of content style

Features

7th January 2016 at 11:58pm

TableOfContents

Further information about specific TiddlyWiki features:

- AlertMechanism
- Audio
- AutoSave
- Core Icons
- DateFormat
- Deserializers
- Drag and Drop
- Editor toolbar
- Encryption
- ExternalImages
- Future Proof
- Importing Tiddlers
- InfoPanel
- JSON in TiddlyWiki
- KeyboardShortcuts
- LazyLoading

- Modals
- Notifications
- Performance Instrumentation
- PermaLinks
- SafeMode
- Scalability
- Searching in TiddlyWiki
- StartupActions
- Upgrading
- Using SVG
- WebServer

Last but not least, TiddlyWiki is a rare example of a practical quine

Federatial

4th December 2022 at 5:56pm

Definitions

Federatial Limited helps organisations explore new user interaction concepts through rapid prototyping of sophisticated web-based tools. Services include:

- Consultancy advice on TiddlyWiki adoption
- Development of custom solutions based on TiddlyWiki

- Multi-user TiddlyWiki hosting – no outages since 2016
- Sponsorship of TiddlyWiki core development

See https://federatial.com/ and https://twitter.com/federatial for more information.

JeremyRuston founded Federatial Limited in 2011 to support his work on TiddlyWiki for the advertised 25 years. Since then, Federatial has been privileged to have worked with a wide range of fantastic clients in very different sectors, on some very diverse projects:

- For a law firm in the Washington DC, developing a custom multi-user TiddlyWiki application. It replaced a Microsoft Word document hosted on SharePoint that had grown to 18,000 pages and 10 million words, and was increasingly unwieldy to manage. We converted the document to a non-linear TiddlyWiki structure. Startup, searching and browsing are an order of magnitude faster than using Word and SharePoint
- For a publishing firm in Paris, devising a new TiddlyWiki-based format for interactive electronic publications (including conversion from the established EPUB format). See https://twpub-tools.org/
- For a charity in London, maintaining a multiuser instance of TiddlyWiki running on Amazon's serverless infrastructure. It has more than 1,000 users, and over 15,000 tiddlers in 500 interlinked wikis. See https://manuals.annafreud.org/ambit

Federatial.png

picture



FetchCommand

18th February 2017 at 2:15pm

Commands

Fetch one or more files over HTTP/HTTPS, and import the tiddlers matching a filter, optionally transforming the incoming titles.

```
--fetch file <url> <import-filter> <transform-filter>
--fetch files <url-filter> <import-filter> <transform-filter>
--fetch raw-file <url> <transform-filter>
--fetch raw-files <url-filter> <transform-filter>
```

The "file" and "files" variants fetch the specified files and attempt to import the tiddlers within them (the same processing as if the files were dragged into the browser window). The "raw-file" and "raw-files" variants fetch the specified files and then store the raw file data in tiddlers, without applying the import logic.

With the "file" and "raw-file" variants only a single file is fetched and the first parameter is the URL of the file to read.

With the "files" and "raw-files" variants, multiple files are fetched and the first parameter is a filter yielding a list of URLs of the files to read. For example, given a set of tiddlers tagged "remote-server" that have a field "url" the filter `[tag[remote-server]get[url]]` will retrieve all the available URLs.

For the "file" and "files" variants, the `<import-filter>` parameter specifies a filter determining which tiddlers are imported. It defaults to `[all[tiddlers]]` if not provided.

For all variants, the `<transform-filter>` parameter specifies an optional filter that transforms the titles of the imported tiddlers. For example, `[addprefix[$:/myimports/]]` would add the prefix `$:/myimports/` to each title.

Preceding the `--fetch` command with `--verbose` will output progress information during the import.

Note that TiddlyWiki will not fetch an older version of an already loaded plugin.

The following example retrieves all the non-system tiddlers from https://tiddlywiki.com and saves them to a JSON file:

```
tiddlywiki --verbose --fetch file "https://tiddlywiki.com/" "[!is[system]]" "" --rendertiddler "$:/core/templates/exporters/JsonFile" output.json text/plain "" exportFilter "[!is[system]]"
```

The following example retrieves the "favicon" file from tiddlywiki.com and saves it in a file called "output.ico". Note that the intermediate tiddler "Icon Tiddler" is quoted in the "–fetch" command because it is being used as a transformation filter to replace the default title, while there are no quotes for the "–savetiddler" command because it is being used directly as a title.

```
tiddlywiki --verbose --fetch raw-file "https://tiddlywiki.com/favicon.ico" "[[Icon Tiddler]]" --savetiddler "Icon Tiddler" output.ico
```

Field Editor Cascade

5th March 2022 at 6:39pm

Cascades

The field editor cascade is a cascade used to choose a template for rendering the field editor within the **EditTemplate**.

The core field editor cascade can be found in **$:/core/ui/EditTemplate/fields**

The default field editor cascade only contains one element:

1. Use the tiddler **$:/core/ui/EditTemplate/fieldEditor/default** to render the field

See Customizing EditTemplate field rendering for more details.

You can see the current settings for the field editor cascade in **$:/ControlPanel** under the **Info** -> **Advanced** -> **Cascades** -> **Field Editor** tab.

field Operator

16th March 2023 at 12:19pm

Common Operators Field Operators Filter Operators Negatable Operators

purpose	filter the input by field
input	a selection of titles
suffix	**F** = the name of a field
parameter	**S** = a possible value of field **F**
output	those input tiddlers in which field **F** has the value **S**
! **output**	those input tiddlers in which field **F** does *NOT* have the value **S**

Learn more about how to use Filters

If **S** is empty, **`field`** will match both of the following:

- tiddlers that don't contain field **F**
- tiddlers in which field **F** exists but has an empty value

The syntax of a filter step treats any unrecognised filter operator as if it was the suffix to the **`field`** operator.

Examples

field Operator (Examples)

18th January 2015 at 7:32pm

field Operator Operator Examples

`[field:author[JeremyRuston]]`

→ plugins authored by JeremyRuston

Try it

`[plugin-type[theme]author[JeremyRuston]]`

→ themes authored by JeremyRuston

Try it

FieldMangler Widget (Examples)

6th July 2015 at 7:29pm

FieldManglerWidget Widget Examples

copy to clipboard

```
<$fieldmangler>
Add tag ''example'' to this tiddler (current)<$button message="tm-add-tag" param="example">{{$:/core/
images/new-button}}</$button>
<br>
```

```
Remove tag ''example'' to this tiddler (current)<$button message="tm-remove-tag" param="example">{{$:/
core/images/delete-button}}</$button>
</$fieldmangler>
```

Try it

copy to clipboard

```
<$fieldmangler tiddler="Hello World">
Add tag ''example'' to the Hello World tiddler<$button message="tm-add-tag" param="example">{{$:/core/
images/new-button}}</$button>
<br>
Remove tag ''example'' to the Hello World tiddler<$button message="tm-remove-tag" param="example">{{$:/
core/images/delete-button}}</$button>
</$fieldmangler>
```

Try it

FieldManglerWidget

9th October 2021 at 2:28pm

MessageHandlerWidgets Widgets

Introduction

The field mangler widget manipulates the fields and tags of a tiddler. It does so in response to the following Messages:

Message	Description
tm-remove-field	Remove the field specified in `event.param`
tm-add-field	Add the field specified in `event.param`
tm-remove-tag	Remove the tag specified in `event.param`
tm-add-tag	Add the tag specified in `event.param`

Content and Attributes

The field mangler widget displays any contained content, and responds to Messages dispatched within it.

Attribute	Description
tiddler	Title of the tiddler to manipulate (defaults to the current tiddler)

Examples

fields Operator

29th January 2020 at 5:50pm

Field Operators Filter Operators

purpose	select all field names of the input titles
input	a selection of titles
suffix	ⓘ **Introduced in v5.1.22** optional: `include`, `exclude` parameter list
parameter	ⓘ **Introduced in v5.1.22** optional: a list of field names
output	all the field names contained in the input tiddlers

Learn more about how to use Filters

Each input title is processed in turn. Its list of field names is retrieved (in no particular order) and then dominantly appended to the operator's output.

ⓘ **Introduced in v5.1.22** If the `include` suffix is used, fields are only included, if they exist. It doesn't matter, if fields have a value. The `exclude` suffix is there for convenience, since it would be possible to use a second filter run. For more info see the examples.

Examples

fields Operator (Examples)

29th January 2020 at 5:56pm

fields Operator Operator Examples

```
[[HelloThere]fields[]]
```

→ fields of HelloThere

Try it

```
[[HelloThere]fields:include[list title text non-existing]]
```

→ fields of HelloThere using a "short list" of fields. Fields are only shown, if they exist

Try it

```
[[HelloThere]fields:include[list title text]sortby[title list text]]
```

→ fields of HelloThere special sorting

Try it

```
[[HelloThere]fields:exclude[list title text]]
```

→ fields of HelloThere using the exclude suffix

Try it

```
[[HelloThere]fields[]] -list -title -text
```

→ fields of HelloThere, using several filter runs instead of exclude suffix

Try it

```
[tag[Common Operators]fields[]]
```

→ fields of all tiddlers tagged as *Common Operators*

Try it

FieldsWidget

20th February 2015 at 5:12pm

Widgets

Introduction

The fields widget renders each field of a specified tiddler through a simple text template. A list of fields to exclude can be provided. It is used internally by TiddlyWiki5, notably by the *FileSavingMechanism*.

Template Handling

The provided template is rendered with the following special substitutions:

Symbol	Substitution
$name$	Field name
$value$	Field value
$encoded_value$	HTML encoded form of field value

Content and Attributes

The content of the `<$fields>` widget is ignored.

Attribute	Description
tiddler	Title of the tiddler from which the fields are to be displayed (defaults to the current tiddler)
template	Text of the template (see above)
exclude	Lists of fields to be excluded (defaults to "text")
include	Lists of fields to be included, if the field exists. This parameter takes precedence over "exclude"
sort	Sorts the fields by name (defaults to "yes"). Set to "no", if "include" order should be retained!
sortReverse	Reverses the sort order
stripTitlePrefix	If set to "yes" then curly bracketed prefixes are removed from titles (for example `{prefix}HelloThere` converts to `HelloThere`)

The `stripTitlePrefix` attribute is used when building TiddlyWiki Classic; see `editions/tw2` in the TiddlyWiki5 repo.

FillWidget

11th May 2023 at 2:34pm

Widgets

Introduction

New in v5.3.0 The `$fill` widget is used within a `$transclude` widget to specify the content that should be copied to the named "slot". Slots are defined by the `$slot` widget within the transcluded content.

See the `$transclude` widget for details and an example. More examples can be found in the `$slot` widget documentation.

Attributes

The content of the `$fill` widget is used as the content to be passed to the transclusion.

Attribute	Description
$name	The name of the slot to be filled

> The $name attribute must be specified as a literal string

Filter Expression

10th July 2023 at 9:45am

Filter Syntax

A **filter expression** is the outermost level of the filter syntax. It consists of filter runs with optional filter run prefixes. Multiple filter runs are separated by whitespace.

TypeError: setting getter-only property "parentNode"

> If the diagram has a single start and end line, as shown above, it means there is more info in the linked level above. The breadcrumbs can be used to navigate

If the diagram has no start and no end, as used in lower levels, it means that higher level syntax elements have been removed, to increase readability and simplicity. The breadcrumbs can be used to navigate

Filter Filter Run Prefix

10th July 2023 at 9:33am

Named Filter Run Prefix

purpose	remove every input title for which the filter run output is an empty list
input	the filter output of all previous runs so far
output	the input titles for which the filter run is not empty

Introduced in v5.1.23
TypeError: setting getter-only property "parentNode"

It receives the filter output so far as its input. The next run is evaluated for each title of the input, removing every input title for which the output is an empty list.

Note that within the filter run, the **`currentTiddler`** variable is set to the title of the tiddler being processed. This permits filter runs like `:filter[{!!price}multiply{!!cost}compare:integer:gteq[5]]` to be used for computation. The value of currentTiddler outside the run is available in the variable **`..currentTiddler`**.

The following variables are available within the filter run:

- **`currentTiddler`** - the input title
- **`..currentTiddler`** - the value of the variable `currentTiddler` outside the filter run.
- **`index`** - **Introduced in v5.2.1** the numeric index of the current list item (with zero being the first item in the list).

- **`revIndex`** - ⓘ **Introduced in v5.2.1** the reverse numeric index of the current list item (with zero being the last item in the list).
- **`length`** - ⓘ **Introduced in v5.2.1** the total length of the input list.



Compare named filter run prefix `:filter` with filter Operator which applies a subfilter to every input title, removing the titles that return an empty result from the subfilter

Filter Filter Run Prefix (Examples)

Filter Filter Run Prefix (Examples)

5th March 2023 at 1:53pm

Filter Filter Run Prefix

```
[tag[HelloThere]] :filter[get[text]length[]compare:integer:gteq[1000]]
```

Try it

```
[tag[HelloThere]] :filter[get[text]length[]compare:integer:lteq[2000]]
```

Try it

```
[tag[HelloThere]] :filter[get[text]length[]compare:integer:gteq[1000]]
```

```
:filter[get[text]length[]compare:integer:lteq[2000]]
```

Try it

```
[tag[Features]] :filter[links[]!is[shadow]is[missing]]
```

Try it

```
A B C D E F :filter[<index>remainder[2]compare:integer:eq[0]]
```

→ Return every other element of the input list

Try it

```
[tag[shopping]] :filter[{!!quantity}compare:integer:gt[4]] :map[addprefix[ ]addprefix{!!quantity}]
```

Try it

```
[tag[shopping]] :filter[{!!quantity}compare:integer:lteq[4]] :map[addprefix[ ]addprefix{!!quantity}]
```

Try it

Comparison between `:filter` and `:and`/`+` filter run prefixes

The functionality of the **`:filter`** filter run prefix has some overlap with the **`:and`** prefix (alias **`+`**). Notice the filter expressions above all have the following two properties:

1. The start of the filter expression transforms the input titles (i.e. `get[text]length[]`, `links[]`, `<index>remainder[2]`, `{!!quantity}`.
2. The end of the filter expression only sends input items to the output if they meet some condition (i.e. `compare`, `is`).

The purpose of the **`:filter`** prefix is to return the original input titles despite property #1 causing those input titles to be transformed. That way, the "meet some condition" check can be performed on something other than the original input. If some filter expression does not require property #1, then the **`:and`** prefix can be used instead.

`:filter`	**`:and`**

`[tag[HelloThere]] :filter[get[text]length[]compare:integer:gteq[1000]]` Try it	`[tag[HelloThere]] :and[get[text]length[]compare:integer:gteq[1000]]` Try it
`[tag[Features]] :filter[links[]! is[shadow]is[missing]]` Try it	`[tag[Features]] :and[links[]! is[shadow]is[missing]]` Try it
results are the same	
`cat can bat bug :filter[suffix[at]minlength[3]]` Try it	`cat can bat bug :and[suffix[at]minlength[3]]` Try it

filter Operator

22nd May 2021 at 6:25pm

Filter Operators Negatable Operators

purpose	apply a subfilter to each input title and return the titles that return a non-empty result from the subfilter
input	a selection of titles passed as input to the filter
! **input**	a selection of titles passed as input to the filter
parameter	**S** = a filter expression
output	the selection of titles that pass the filter **S**
! **output**	those input titles that *DO NOT* pass the filter **S**

Learn more about how to use Filters

ⓘ **Introduced in v5.1.23** The **`filter`** operator runs a subfilter for each input title, and returns those input titles for which the subfilter returns a non-empty result (in other words the result is not an empty list). The results of the subfilter are thrown away.

Simple filter operations can be concatenated together directly (eg `[tag[HelloThere]search[po]]`) but this doesn't work when the filtering operations require intermediate results to be computed. The **`filter`** operator can be used to filter on an intermediate result which is discarded. To take the same example but to also filter by those tiddlers whose text field is longer than 1000 characters:

```
<$vars myfilter="[get[text]length[]compare:integer:gteq[1000]]">
<$list filter="[tag[HelloThere]search[po]filter<myfilter>]">
<div>
<$link>
<$text text=<<currentTiddler>>/>
</$link>
</div>
</$list>
```

```
</$vars>
```

Note that within the subfilter, the "currentTiddler" variable is set to the title of the tiddler being processed. The value of currentTiddler outside the subfilter is available in the variable "..currentTiddler". **Introduced in v5.2.0**

Compare with the similar subfilter operator which runs a subfilter and directly returns the results

Compare with the analogous named filter run prefix `:filter`

```
\define larger-than-1k() [get[text]length[]compare:integer:gteq[1000]]

{{{ [tag[HelloThere]filter<larger-than-1k>] }}}
```

is equivalent to:

```
{{{ [tag[HelloThere]] :filter[get[text]length[]compare:integer:gteq[1000]] }}}
```

Examples

filter Operator (Examples)

4th October 2020 at 4:33pm

filter Operator Operator Examples

These examples use the following predefined variables:

- **larger-than-1k**: `[get[text]length[]compare:integer:gteq[1000]]`

- **smaller-than-2k**: `[get[text]length[]compare:integer:lteq[2000]]`
- **contains-missing-links**: `[links[]!is[shadow]is[missing]]`

`[tag[HelloThere]filter<larger-than-1k>]`

Try it

`[tag[HelloThere]filter<smaller-than-2k>]`

Try it

`[tag[HelloThere]filter<larger-than-1k>filter<smaller-than-2k>]`

Try it

`[tag[Features]filter<contains-missing-links>]`

Try it

Filter Operators

10th April 2023 at 1:41pm

Filters

A **filter operator** is a predefined keyword attached to an individual step of a filter. It defines the particular action of that step.

Important: In general, each first filter step of a filter run not given any input titles receives the output of [all[tiddlers]] as its input.

Table legend:

- ✓ ... Used to mark the most common ones
- ! ... The column indicates whether an operator allows **negation** using the ! prefix.
  For specifics as to each operator's negated output please refer to its documentation
- C ... Most steps process the selection of titles that are supplied as their input, but a few construct an entirely new selection instead, as indicated by the last column.
  A C? indicates it might construct a new selection, depending on usage. For specifics as to each operator's selection creation please refer to its documentation

The following table lists all core operators:

Operator	Purpose	✓	!	C
all	find all titles of a fundamental category	✓		C?
backlinks	find the titles that link to each input title			
charcode	generates string characters from their numeric character codes			
contains	filter the input by searching list fields for a value		!	

count	count the number of entries in a list			
days	filter the input by date		!	
duplicateslugs	find titles that yield duplicate slugs			
each	select one of each group of input titles by field			
eachday	select one of each group of input titles by date			
else	if the list of input titles is empty then return a list consisting of a single constant string, otherwise return the original titles			
enlist	select titles from the parameter interpreted as a title list		!	C
field	filter the input by field	✓	!	
fields	select all field names of the input titles			
filter	apply a subfilter to each input title and return the titles that return a non-empty result from the subfilter		!	
function	apply a function to the input list, and return the result			
get	select all values of a field in the input titles			
getindex	select all values of a data property in the input titles			
has	filter the input by field existence	✓	!	
haschanged	filter the input by tiddler modification status		!	
indexes	select all data properties of the input titles			
is	filter the input by fundamental category	✓	!	
join	join a list of strings together with the separator **S**			
jsonextract	retrieve the JSON string of a property from JSON strings			
jsonget	retrieve the value of a property from JSON strings			
jsonindexes	retrieve the value of a property from JSON strings			
jsonset	set the value of a property in JSON objects			
jsontype	retrieve the type of a property from JSON strings			

length	returns the number of characters of each item in the list			
links	find the titles linked to by each input title	✓		
list	select titles via a list field		!	C
listed	find the titles that list the input titles			
lookup	applies a prefix to each input title to yield the title of a tiddler from which the final value is retrieved. With a single parameter, the default field is "text" and the default index is "0". If a second parameter is provided, that becomes the target field or index.			
lowercase	returns each item in the list as lowercase			
match	returns each item in the list that matches the parameter string			
minlength	filter items shorter than the specified minimum length			
range	generate a range of numbers		!	C
reduce	apply a subfilter to each input title in turn, accumulating a single value			
regexp	filter the input by pattern-matched field		!	
sameday	filter the input by date			
search	filter the input by searching tiddler content	✓	!	
sentencecase	returns each item in the list with the first letter capitalised			
slugify	returns each item in the list in a human-readable form for use in URLs or filenames			
split	returns each item in the list split into separate strings according to the specified separator **S** ; duplicates are *NOT* removed			
splitregexp	returns each item in the list split into separate strings according to the specified regular expression **R**			
subfilter	select titles from the parameter interpreted as a filter expression		!	C?
then	replace input titles by a constant string			
title	select a single title	✓	!	C
titlecase	returns each item in the list with each word capitalised			

uppercase	returns each item in the list as uppercase			
Order Operators				
after	find which input title follows a specified one			
before	find which input title precedes a specified one			
bf	same as **rest**			
butfirst	same as **rest**			
butlast	discard the last **N** input titles			
first	select the first **N** input titles			
last	select the last **N** input titles			
limit	select the first or last **N** input titles	✓	!	
next	find which titles in a **list** field follow the input ones			
nsort	sort the input by number field		!	
nsortcs	sort the input titles by number field, treating upper and lower case as different		!	
nth	select the **N** th input title			
order	selectively reverse the input list	✓		
previous	find which titles in a **list** field precede the input ones			
rest	discard the first **N** input titles			
reverse	reverse the order of the input titles			
sort	sort the input by text field	✓	!	
sortan	sort the input by text field considering them as alphanumerics	✓	!	
sortcs	sort the input by text field, treating upper and lower case as different		!	
sortsub	sort the input by the result of evaluating a subfilter for each item		!	
zth	select the **Z** th input title			

Listops Operators				
allafter	discard all items except those after the marker			
allbefore	discard all items except those before the marker			
append	append a range of items from an array to the list			
cycle	toggle the titles specified in the first parameter in a cyclical manner			
insertafter	insert an item **T** into a list immediately after an item **A**			
insertbefore	insert an item **T** into a list immediately before an item **B**			
move	move marker N places in the list			
prepend	prepend a range of items from an array to the list			
putafter	move N trailing items after the marker			
putbefore	move N trailing items before the marker			
putfirst	move N trailing items to the head of the list			
putlast	move N leading items to the tail of the list			
remove	remove a list of titles specified in the parameter from the input			
replace	replace marker with N trailing items			
sortby	sort the current list in the order of the list referenced in the parameter			
toggle	toggle the title specified in the parameter in the input			
unique	remove all duplicate items from the current list			
String Operators				
addprefix	extend each input title with a prefix			
addsuffix	extend each input title with a suffix			
applypatches	applies a set of patches to transform the input			
compare	filter the input by comparing each item against the parameter		!	

decodebase64	apply base 64 decoding to a string			
decodehtml	apply HTML decoding to a string			
decodeuri	apply URI decoding to a string			
decodeuricomponent	apply URI component decoding to a string			
encodebase64	apply base 64 encoding to a string			
encodehtml	apply HTML encoding to a string			
encodeuri	apply URI encoding to a string			
encodeuricomponent	apply URI component encoding to a string			
enlist-input	select titles by interpreting each input title as a title list			
escapecss	apply CSS escaping to a selection of titles through the CSS.escape() method / operation			
escaperegexp	escape special characters used in regular expressions			
format	format the input string according to one of supported formats			
jsonstringify	apply JSON string encoding to a string, see also the similar **stringify**			
levenshtein	determine the Levenshtein distance of the input title(s) and a given string			
makepatches	returns a set of patches that transform the input to a given string			
pad	returns each item in the list padded to the specified length			
prefix	filter the input titles by how they start		!	
removeprefix	filter the input titles by how they start, deleting that prefix			
removesuffix	filter the input titles by how they end, deleting that suffix			
search-replace	returns each item in the list, replacing within each title the string specified by the first parameter with the second parameter			
sha256	apply sha256 hash to a string			
splitbefore	select a delimited prefix from each input title			
stringify	apply JavaScript string encoding to a string, see also the similar **jsonstringify**			

substitute	returns each item in the list, replacing within each title placeholders for filters, parameters and variables with their corresponding values			
suffix	filter the input titles by how they end		!	
trim	returns each item in the list with whitespace, or a given character string, trimmed from the start and/or end			
Mathematics Operators				
abs	calculate the absolute value of a list of numbers			
acos	calculate the arccosine value (in radians) of a list of numbers			
add	treating each input title as a number, add to each the numeric value of the parameter			
asin	calculate the arcsine value (in radians) of a list of numbers			
atan	calculate the arctangent value (in radians) of a list of numbers			
atan2	returns the angle in the plane (in radians) between the positive x-axis and the ray from (0,0) to the point (x,y), for [**Y**]atan2[**X**]			
average	treating each input title as a number, compute their arithmetic mean			
ceil	rounds a list of numbers up to the next largest integer			
compare	filter the input by comparing each item against the parameter		!	
cos	calculate the cosine value of a list of angles (given in radians)			
divide	treating each input title as a number, divide it by the numeric value of the parameter			
exponential	convert each number to exponential notation with **N** digits			
fixed	convert each number to fixed point notation with **N** digits after the decimal point			
floor	rounds a list of numbers to the largest integer less than or equal to each number			
log	treating each input title as a number, return its logarithm with base equal to the numeric value of the parameter if specified, otherwise base e			

max	treating each input title as a number, take the maximum of its value and the numeric value of the parameter
maxall	find the largest of a list of numbers
median	treating each input title as a number, compute their median value
min	treating each input title as a number, take the minimum of its value and the numeric value of the parameter
minall	find the smallest of a list of numbers
multiply	treating each input title as a number, multiply it by the numeric value of the parameter
negate	calculate the negation of a list of numbers
power	treating each input title as a number, raise it to the power of the numeric value of the parameter
precision	convert each number to a string with **N** significant digits
product	produce the product of the input numbers
remainder	treating each input title as a number, return the remainder when divided by the numeric value of the parameter
round	rounds a list of numbers to the nearest integer
sign	return -1, 0 or 1 for a list of numbers according to whether each number is negative, zero, or positive
sin	calculate the sine value of a list of angles (given in radians)
standard-deviation	treating each input title as a number, compute their standard-deviation
subtract	treating each input title as a number, subtract from each the numeric value of the parameter
sum	produce the sum of the input numbers
tan	calculate the tangent value of a list of angles (given in radians)
trunc	truncates a list of numbers to their integer part, removing any fractional part
untrunc	rounds a list of numbers to the next integer with largest absolute value, that is, away from zero
variance	treating each input title as a number, compute their variance

Tag Operators				
tag	filter the input by tag	✓	!	
tagging	find the tiddlers that have the input tags			
tags	select all tags of the input tiddlers			
untagged	discard any input titles that have tags		!	
Special Operators				
commands	select the titles of all the Node.js commands			C
deserialize	extract JSON representation of tiddlers from the input strings			
deserializers				C
editiondescription	select the descriptions of the input editions			
editions	select the names of all the TiddlyWiki editions			C
getvariable	select all values of variables named in the input titles			
modulesproperty	retrieve a module property			
modules	select the names of all modules of the input module types			
moduletypes	select the names of all TiddlyWiki module types			C
plugintiddlers	select all shadow titles in the input plugins			
shadowsource	select the plugin titles that contain the input shadows			
storyviews	select the names of all the story views			C
subtiddlerfields	select all fields present in the selected tiddlers within a plugin			
variables	select the names of all the actively defined variables			C

A typical step is written as `[operator[parameter]]`, although not all of the operators need a parameter.

For the exact rules, see Filter Syntax.

Filter Parameter

10th July 2023 at 9:44am

Filter Step

TypeError: setting getter-only property "parentNode"

The parameter to a filter operator can be:

hard

`[like this]`

The parameter is the exact text that appears between the square brackets.

soft

indirect

`{like this}`

The parameter is the text indicated by the text reference whose name appears between the curly brackets, i.e. a field of a specified tiddler, or the value of a property of a specified data tiddler.

variable

`<like this>`

The parameter is the current value of the variable whose name appears between the angle brackets. Macro parameters are *NOT* supported up to v5.2.0

ⓘ **Introduced in v5.2.0** Literal macro parameters are supported. For example: `[<now [UTC]YYYY0MM0DD0hh0mm0ssXXX>]`.

ⓘ **Introduced in v5.1.23** Filter Steps support multiple parameters which are separated by a `,` character.

For example: `[param1],[param2]` or `<param1>,{param2}`

Filter Run

10th July 2023 at 9:43am

Filter Expression

TypeError: setting getter-only property "parentNode"

A **run** consists of steps, and it outputs a selection that contributes to a larger filter expression.

The steps are processed from left to right. The input to the first step is same as the input to the run. For each subsequent step, the input is the output of the previous step.

The lower three options in the diagram match syntax like `HelloThere`, `"HelloThere"`, `'HelloThere'` and `"Filter Operators"`. They are short for `[title[...]]`.

The quoted options exist to support titles that contain square brackets, as in `"An [[[[Unusual]]]] Tiddler"`.

Filter Run Prefix

11th July 2023 at 11:09am

Filter Expression

There are 2 types of filter run prefixes that are interchangeable. Named prefixes and shortcut prefixes.

TypeError: setting getter-only property "parentNode"

Filter Run Prefix (Examples)

15th March 2023 at 4:28pm

`:filter` examples

See Filter Filter Run Prefix (Examples)

`:intersection` examples

See Intersection Filter Run Prefix (Examples)

`:reduce` examples

See Reduce Filter Run Prefix (Examples)

`:sort` examples

See Sort Filter Run Prefix (Examples)

Filter Step

10th July 2023 at 9:44am

Filter Run

A **filter step** represents a single operation within a **filter run**.

In programming terms, it is akin to a function call to which the step's input is passed as a parameter. A step's output is a title selection that contributes to a filter run and hence to the entire filter expression that contains it.

TypeError: setting getter-only property "parentNode"

The step's **operator** is drawn from a list of predefined keywoards which are known as filter operators.

Many steps require an explicit **parameter**, that further defines what the step is to do.

The **suffix** is additional text, often the name of a field, that extends the meaning of certain operators.

If a step's **operator** and **suffix** are *omitted* altogether, it defaults to the title operator.

ⓘ **Introduced in v5.1.23** Some steps accept multiple **parameter**s which are separated by a `,` character.

Any unrecognised operator is treated as if it was the suffix to the **`field`** operator.

Filter operators can be extended by plugins.

The output of a Filter Step depends on its operator:

- Most operators derive their output from their input. For example, many of them output a subset of their input, and thus truly live up to the name of "filters", narrowing down the overall output of the containing run. These operators are called **selection modifiers**.
- A few operators ignore their input and generate an independent output instead. These are called **selection constructors**: they construct an entirely new selection.

A good example of a constructor is `title`. The output of `[title[A]title[B]]` is just **B**. But the `field` operator is a modifier, so `[title[A]field:title[B]` outputs nothing at all.

Filter Syntax

10th July 2023 at 9:43am

Filters

> Filters follow a grammar that is presented here, using railroad diagrams, for those who find formal syntax descriptions helpful. However, you can learn to write filters without needing to understand this group of tiddlers.

A **filter** is a pipeline for transforming an **input** into an **output**. Both the input and the output are ordered sets of titles of tiddlers and fields.

Filters are **expressions** constructed from smaller building blocks, called **runs**, which are built using **steps**. Eeach of which also transforms an input to an output.

A filter starts with an empty output. Its runs are processed from left to right, progressively modifying the output.

The "Filter Syntax" description starts with:

TypeError: setting getter-only property "parentNode"



> The railroad boxes as the one above can be used to navigate

Filter Whitespace

10th July 2023 at 9:44am

Filter Expression

TypeError: setting getter-only property "parentNode"

Whitespace characters can appear between each run of a filter expression.

Filtered Attribute Values

15th June 2023 at 6:52am

Widget Attributes WikiText

Filtered attribute values are indicated with triple curly braces around a Filter Expression. The value will be the first item in the resulting list, or the empty string if the list is empty.

Introduced in v5.2.2 To improve readability, newlines can be included anywhere that whitespace is allowed within filtered attributes.

This example shows how to add a prefix to a value:

```
<$text text={{{ [<currentTiddler>addprefix[$:/myprefix/]] }}} />
```

The value of the attribute will be the exact text from the first item in the resulting list. Any wiki syntax in that text will be left as-is.

Filters

10th July 2023 at 9:45am

Concepts Reference TableOfContents

You can think of TiddlyWiki as a database in which the records are tiddlers. A database typically provides a way of discovering which records

match a given pattern, and in TiddlyWiki this is done with filters.

A **filter** is a concise notation for selecting a particular set of tiddlers, known as its **output**. Whenever TiddlyWiki encounters a filter, it calculates the output. Further work can then be done with just those tiddlers, such as counting or listing them.

The following example passes a filter to the **`list-links`** macro. It displays a list of all tiddlers using the **`tag`** "Filters":

copy to clipboard

```
<<list-links "[tag[Filters]]">>
```

That renders as:

- Introduction to filter notation
- Filter Syntax
- Conditional Operators
- Dominant Append
- Extended Listops Filters
- Filter Operators
- Mathematics Operators
- Selection Constructors

A filter output can change as tiddlers are added and deleted in the wiki. TiddlyWiki recalculates on the fly, automatically updating any filter-based counts or lists as well.

Find out more:

- **Advanced Search** – has a **Filter** tab that makes it easy to experiment with filters.
- Filtered Transclusions – If you want to use filter results in your text

First

14th November 2021 at 2:36am

Contents Table-of-Contents Demonstrations

This is an example tiddler. See Table-of-Contents Macros (Examples).

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

first Operator

3rd February 2015 at 8:19pm

Filter Operators Order Operators

purpose	select the first `N` input titles
input	a selection of titles
parameter	`N` = an integer, defaulting to 1
output	the first `N` input titles

Learn more about how to use Filters

Examples

first Operator (Examples)

18th January 2015 at 7:32pm

first Operator Operator Examples

These examples make use of the Days of the Week tiddler.

```
[list[Days of the Week]first[]]
```

Try it

```
[list[Days of the Week]first[5]]
```

Try it

```
[tag[Filter Operators]!sort[title]first[]]
```

Try it

FirstOne

14th November 2021 at 2:36am

First Table-of-Contents Demonstrations

This is an example tiddler. See Table-of-Contents Macros (Examples).

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

FirstThree

14th November 2021 at 2:36am

First Table-of-Contents Demonstrations

This is an example tiddler. See Table-of-Contents Macros (Examples).

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

FirstTwo

14th November 2021 at 2:36am

First Table-of-Contents Demonstrations

This is an example tiddler. See Table-of-Contents Macros (Examples).

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

fixed Operator

11th June 2019 at 5:09pm

Binary Mathematics Operators Filter Operators Mathematics Operators

purpose	convert each number to fixed point notation with `N` digits after the decimal point
input	a selection of titles
parameter	`N` = a number
output	the input as numbers converted to fixed point notation with `N` digits after the decimal point

Learn more about how to use Filters

ⓘ **Introduced in v5.1.20** See Mathematics Operators for an overview.

The **fixed** operator returns a string representation of the input number that does not use exponential notation and has exactly the specified number of digits after the decimal place. The number is rounded if necessary, and the fractional part is padded with zeros if necessary so that it has the specified length.

Examples

fixed Operator (Examples)

11th June 2019 at 5:07pm

fixed Operator Operator Examples

```
10.123456789 +[fixed[5]]
```

Try it

```
[[355]divide[113]fixed[7]]
```

Try it

```
[[1]divide[2]fixed[3]]
```

Try it

floor Operator

13th June 2019 at 10:44am

Filter Operators Mathematics Operators Unary Mathematics Operators

purpose	rounds a list of numbers to the largest integer less than or equal to each number
input	a selection of titles
output	rounds each of the input numbers to the largest integer less than or equal to the given number

Learn more about how to use Filters

Introduced in v5.1.20 See Mathematics Operators for an overview.

Examples

floor Operator (Examples)

13th June 2019 at 10:45am

floor Operator Operator Examples

```
[[1.6]floor[]]
```

Try it

```
[[-1.6]floor[]]
```

Try it

```
=-1.2 =-2.4 =3.6 =4.8 =5.1 +[floor[]]
```

Try it

format Operator

21st March 2023 at 2:22pm

Filter Operators String Operators

purpose	format the input string according to one of supported formats
input	a selection of titles
suffix	**B** = one of supported formats
parameter	**C** = optional format string for the formats
output	input strings formatted according to the specified suffix **B**

Learn more about how to use Filters

Introduced in v5.1.23

The suffix **B** is one of the following supported string formats:

Format	Description
`date`	The input string is interpreted as a UTC date and displayed according to the DateFormat specified in the optional parameter **C**. (Defaults to "YYYY MM DD 0hh:0mm")
`json`	ⓘ **Introduced in v5.2.4** The input string is interpreted as JSON and displayed with standard formatting. The optional parameter **C** specifies the number of spaces to use for indenting, or a string to use for indenting. Nothing is returned if the input string is not valid JSON
`relativedate`	The input string is interpreted as a UTC date and displayed as the interval from the present instant. Any operator parameters **C** are ignored
`timestamp`	ⓘ **New in v5.3.0** The input string is interpreted as number of milliseconds since the ECMAScript epoch, 1 January 1970, and displayed according to the DateFormat specified in the optional operator parameter. (Defaults to "[UTC]YYYY0MM0DD0hh0mm0ss0XXX")
`titlelist`	ⓘ **Introduced in v5.2.0** The input string wrapped in double square brackets if it contains a space. Appropriate for use in a title list.

Invalid input strings are dropped by the **format** operator.



The Title List format cannot reliably represent items that contain certain specific character sequences such as `]]`. Thus it should not be used where there is a possibility of such sequences occurring.

Examples

format Operator (Examples)

26th February 2023 at 2:56pm

format Operator Operator Examples

These examples use the tiddler HelloThere.

Created date with no format string specified:

```
[[HelloThere]get[created]format:date[]]
```

Try it

Created date with a format string supplied as operator parameter:

```
[[HelloThere]get[created]format:date[DDth MMM YYYY]]
```

Try it

Modified date shown as a relative date:

```
[[HelloThere]get[modified]format:relativedate[]]
```

Try it

Get the date and time exactly 24 hours (86,400,000 milliseconds) from now:

```
[<now [UTC]YYYY0MM0DD0hh0mm0ss0XXX>format:date[TIMESTAMP]add[86400000]format:timestamp[DDth mmm YYYY 0hh:0mm:0ss]]
```

Try it

A tiddler title with spaces formatted as a title list:

```
[[Hello There]format:titlelist[]]
```

Try it

All tiddler titles tagged with TableOfContents formatted as a title list:

```
[tag[TableOfContents]format:titlelist[]]
```

Try it

A JSON string formatted as JSON – note how the JSON string is normalised to remove the duplicated properties:

```
[[{"one":"first","one":"another","two":"second"}]format:json[]]
```

Try it



To create a string to save a title list into a list field, use `format:titlelist[]` with the join operator

```
[tag[TableOfContents]format:titlelist[]join[ ]]
```

Try it

For example, to save titles tagged `TableOfContents` to the titles field of the tiddler *format titlelist test*:

copy to clipboard

```
<$button>Try it
<$action-setfield $tiddler="format titlelist test" titles={{{
[tag[TableOfContents]format:titlelist[]join[ ]] }}}/>
</$button>
```

That renders as:

Try it

Formatting in WikiText

3rd March 2023 at 10:54pm

WikiText

Overview

Available character formatting in WikiText includes:

Wikitext	Button	Shortcut	Rendered Output
Double single quotes are used for `''bold text''`	B	ctrl-B	Double single quotes are used for **bold text**
Double slashes are used for `//italic text//`	I	ctrl-I	Double slashes are used for *italic text*
Double underscores are used for `__underlined text__`	U	ctrl-U	Double underscores are used for underlined text
Double circumflex accents are used for `^^superscripted^^` text	X²	ctrl-shift-P	Double circumflex accents are used for superscripted text
Double commas are used for `,,subscripted,,` text	X₂	ctrl-shift-B	Double commas are used for subscripted text
Double tilde signs are used for `~~strikethrough~~` text	S	ctrl-T	Double tilde signs are used for ~~strikethrough~~ text
Single backticks are used for `` `code` ``	AB	ctrl-M	Single backticks are used for `code`
Double @ characters are used to create a `@@highlight@@`	-	-	Double @ characters are used to create a highlight

See Styles and Classes in WikiText for more formatting options

Embedded Backticks

Wikitext	Rendered Output

Wikitext	Rendered Output
``double backticks allows `embedded` backticks``	`double backticks allows `embedded` backticks`

Keyboard Shortcuts

The full list of KeyboardShortcuts can be found in the **$:/ControlPanel** -> **Keyboard Shortcuts** tab.

Semantic Information

Wikitext	Resulting HTML Code
Double single quotes are used for `''bold text''`	Double single quotes are used for `<strong>bold text</strong>`
Double slashes are used for `//italic text//`	Double slashes are used for `<em>italic text</em>`
Double underscores are used for `__underlined text__`	Double underscores are used for `<u>underlined text<u>`
Double circumflex accents are used for `^^superscripted^^` text	Double circumflex accents are used for `<sup>superscripted</sup>` text
Double commas are used for `,,subscripted,,` text	Double commas are used for `<sub>subscripted</sub>` text
Double tilde signs are used for `~~strikethrough~~` text	Double tilde signs are used for `<strike>strikethrough</strike>` text
Single backticks are used for `` `code` ``	Single backticks are used for `<code>code</code>`
Double @ characters are used to create a `@@highlight@@`	Double @ characters are used to create a `<span class="tc-inline-style">highlighted</span>`

Bold vs Strong

In TW we use the term "bold" instead of "strong" because most users are used to it. ... The rendering process converts our "bold text" into the STRONG HTML element.

> The `<strong>` element is for content that is of greater importance, while the `<b>` element is used to draw attention to text without indicating that it's more important.
>
> https://developer.mozilla.org/en-US/docs/Web/HTML/Element/strong#b_vs._strong

Underline vs. Underscore

> Use underline to describe text formatting that puts a line under the characters. Use underscore to refer to the underscore character (_).
>
> https://learn.microsoft.com/en-us/style-guide/a-z-word-list-term-collections/u/underline-vs-underscore

Emphasis vs Italics

> The `<em>` element represents stress emphasis of its contents, while the `<i>` element represents text that is set off from the normal prose, such as a foreign word, fictional character thoughts, or when the text refers to the definition of a word instead of representing its semantic meaning. (The title of a work, such as the name of a book or movie, should use `<cite>`.)
>
> https://developer.mozilla.org/en-US/docs/Web/HTML/Element/em#i_vs._em

Formatting List Results as Tables (no CSS)

20th December 2016 at 5:17pm

Learning

Sometimes you want the results of a `<$list>` widget to be formatted in the form of multiple columns instead of just one straight listing. The following method creates an actual table structure and uses the nth operator to provide break points for the rows. It is not responsive, that is, it doesn't re-position to display fewer columns if the window is too small.

In the first, outer list structure you must provide a count to indicate at item number rows should occur. So, in the following example, each row breaks after 4 items, so the sequence is 1,5,9, etc. Note that this requires you to know in advance the maximum number of items there will be. There is also an internal limit that is set to n-1 items, where n is the number of columns you want.

Note also that you need to repeat the driving filter operator inside of the internal `<$list>` widget. Obviously this technique lends itself to a macro implementation.

For other table-making techniques see also:

- Formatting List Results as Tables with CSS - Variable Column Method
- Formatting List Results as Tables with CSS - Specified Columns Methods

Example code for a four-column table with fewer than 70 items

```
<table>
<$list filter="1 5 9 13 17 21 25 29 33 37 41 45 49 53 57 61 65" variable ="rows">
    <$list filter="[tag[Filter Operators]limit[50]] +[nth<rows>]" variable="cell">
```

```
        <tr>
        <td> <<cell>> </td>
        <$list filter="[tag[Filter Operators]limit[50]] +[allafter<cell>limit[3]]" variable="this">
            <td> <<this>> </td>
        </$list>
        </tr>
    </$list>
</$list>
</table>
```

Result

abs Operator	acos Operator	add Operator	addprefix Operator
addsuffix Operator	after Operator	all Operator	allafter Operator
allbefore Operator	append Operator	applypatches Operator	asin Operator
atan Operator	atan2 Operator	average Operator	backlinks Operator
before Operator	bf Operator	butfirst Operator	butlast Operator
ceil Operator	charcode Operator	commands Operator	compare Operator
contains Operator	cos Operator	count Operator	cycle Operator
days Operator	decodebase64 Operator	decodehtml Operator	decodeuri Operator
decodeuricomponent Operator	deserialize Operator	deserializers Operator	divide Operator
duplicateslugs Operator	each Operator	eachday Operator	editiondescription Operator
editions Operator	else Operator	encodebase64 Operator	encodehtml Operator
encodeuri Operator	encodeuricomponent Operator	enlist Operator	enlist-input Operator

escapecss Operator | escaperegexp Operator

Formatting List Results as Tables with CSS - Specified Columns Methods

20th December 2016 at 5:16pm

Learning

Sometimes you want the results of a `<$list>` widget to be formatted in the form of multiple columns, instead of just one straight listing. This method uses ~CSS to set up listing as columns and assumes that you know how many columns you want. The method here is to create a style that reflects the number of columns you want your table to be in, and then apply that style to the resulting list output.

For other table-making techniques see also:

- Formatting List Results as Tables with CSS - Variable Column Method
- Formatting List Results as Tables (no CSS)

1) Create a tiddler for the columns tagged with `$:/tags/Stylesheet`, containing:

```
/* FOUR COLUMN MODE */
.fourcolumns {
   display:block;
   column-count:4;
   column-gap:1em;
   -moz-column-count:4;
```

```
   -moz-column-gap:1em;
   -webkit-column-count: 4;
   -webkit-column-gap:1em;
}
```

Note the various places you need to indicate the number of columns

2) Then format your output like this:

```
@@.fourcolumns
<$list filter="[tag[Filter Operators]]" variable="foo">
<<foo>><br>
</$list>
@@
```

Example showing partial list of filter operators

abs Operator
acos Operator
add Operator
addprefix Operator
addsuffix Operator
after Operator
all Operator
allafter Operator
allbefore Operator
append Operator
applypatches Operator
asin Operator
atan Operator
atan2 Operator
average Operator
backlinks Operator
before Operator
bf Operator
butfirst Operator
butlast Operator
ceil Operator
charcode Operator
commands Operator
compare Operator

Formatting List Results as Tables with CSS - Variable Column Method

20th December 2016 at 5:28pm

Learning

Sometimes you want the results of a `<$list>` widget to be formatted in the form of multiple columns, instead of just one straight listing. This method uses CSS to set up listing as columns. It is responsive, that is, re-positioning to display fewer columns if the window is too small.

You don't directly specify a fixed number of columns but instead specify the max-width for the list (which could be a transclusion of the tiddler width) and the width for each item. It lists from left to right, then wraps to a new row.

For other table-making techniques see also:

- Formatting List Results as Tables with CSS - Specified Columns Methods
- Formatting List Results as Tables (no CSS)

Example listing using 50 existing TiddlyWiki tags

```
<div class="dynamic-table">
  <$list filter="[has[tags]tags[]sort[title]first[50]]">
    <span class="item">
      <$transclude tiddler="$:/core/ui/TagTemplate"/>
    </span>
  </$list>
```

```
</div>
```

Example stylesheet to use with listing

```
<style>
.dynamic-table {
  max-width:700px; /* could transclude tiddler width instead */
  -ms-box-orient: vertical; /* might be unnecessary */
  display: -webkit-box;
  display: -moz-box;
  display: -ms-flexbox;
  display: -moz-flex;
  display: -webkit-flex;
  display: inline-flex;
  -webkit-flex-wrap: wrap;
  flex-wrap: wrap;
  flex-direction: row;
}

.item {
  max-width:160px; min-width:160px;
  flex: 0 0 2em; /* -grow, -shrink, -basis */
}
</style>
```

Results

$:/deprecated	$:/tags/EditTemplate	$:/tags/Global	$:/tags/Macro
$:/tags/MenuBar	$:/tags/RawMarkupWikified/TopBody	$:/tags/SideBar	$:/tags/StoryTiddlerTemplateFilter
$:/tags/Stylesheet	$:/tags/TiddlerColourFilter	$:/tags/TiddlerIconFilter	$:/tags/TiddlerInfo
$:/tags/TiddlerList	$:/tags/TiddlyWikiSites	$:/tags/ViewTemplate	About
abs Operator	acos Operator	ActionCreateTiddlerWidg	ActionListopsWidget
ActionWidgets	add Operator	addprefix Operator	addsuffix Operator
after Operator	all Operator	allafter Operator	allbefore Operator
AlphaReleaseNotes	Android	append Operator	applypatches Operator
Articles	asin Operator	atan2 Operator	average Operator
backlinks Operator	before Operator	BetaReleaseNotes	Binary Mathematics Operators
Block Mode WikiText	butlast Operator	Cascade Filter Run Prefix	Cascades
ceil Operator	changecount Macro	charcode Operator	CheckboxWidget
Chrome	colour Macro		

Formatting text in TiddlyWiki

19th September 2014 at 11:48pm

Working with TiddlyWiki

Within the text of a tiddler you can use special formatting called WikiText to control how the text is displayed.

WikiText can be typed by using the Editor toolbar or by typing by hand. While the first is convenient the later can be faster when you know the WikiText markup code.

Simple Formatting

At its simplest, WikiText lets you use familiar word-processing features like bold, italic, lists and tables. For example:

```
The ''quick'' brown ~~flea~~ fox //jumps// over the `lazy` dog
```

… displays as:

The **quick** brown ~~flea~~ fox *jumps* over the `lazy` dog

Working with Tiddlers

In WikiText, you can link to tiddlers using double square brackets, or by taking advantage of the automatic linking of CamelCase words:

```
This is a link to HelloThere, and one to [[History of TiddlyWiki]]
```

… displays as:

This is a link to HelloThere, and one to History of TiddlyWiki

Macros

Macros let you package repetitive fragments of WikiText so that you can easily reuse them.

For example, here is the definition of a macro that generates a YouTube video URL from its unique identifier:

```
\define youtube(video)
https://www.youtube.com/watch?v=$video$
\end
```

With that definition in place, `<<youtube 1g66s7UbyuU>>` generates the URL https://www.youtube.com/watch?v=1g66s7UbyuU

Advanced WikiText

Advanced WikiText features allow you to produce automated lists and interactive features like dropdown menus. In fact, the entire user interface of TiddlyWiki itself is written in WikiText, so any feature that you see in TiddlyWiki can be adapted for use in your own wikis.

Some of the advanced features require complex coding. TiddlyWiki includes several built-in macros that simplify common user interface tasks, like tabs, tables of content, and lists of tiddlers.

Find out more

See WikiText for a detailed introduction to writing WikiText.

Forums

31st January 2022 at 5:51pm

Community

Official Forums

The new official forum for talking about TiddlyWiki: requests for help, announcements of new releases and plugins, debating new features, or just sharing experiences. You can participate via the associated website, or subscribe via email.

https://talk.tiddlywiki.org/

Note that talk.tiddlywiki.org is a community run service that we host and maintain ourselves. The modest running costs are covered by community contributions.

For the convenience of existing users, we also continue to operate the original TiddlyWiki group (hosted on Google Groups since 2005):

https://groups.google.com/group/TiddlyWiki

Developer Forums

There are several resources for developers to learn more about TiddlyWiki and to discuss and contribute to its development.

- tiddlywiki.com/dev is the official developer documentation
- Get involved in the development on GitHub
    - Discussions are for Q&A and open-ended discussion

    - Issues are for raising bug reports and proposing specific, actionable new ideas
- The older TiddlyWikiDev Google Group is now closed in favour of GitHub Discussions but remains a useful archive: https://groups.google.com/group/TiddlyWikiDev
    - An enhanced group search facility is available on mail-archive.com
- Follow @TiddlyWiki on Twitter for the latest news
- Chat at https://gitter.im/TiddlyWiki/public (development room coming soon)

Other Forums

- TiddlyWiki Subreddit
- Chat with Gitter at https://gitter.im/TiddlyWiki/public !
- Chat on Discord at https://discord.gg/HFFZVQ8

Documentation

There is also a discussion group specifically for discussing TiddlyWiki documentation improvement initiatives: https://groups.google.com/group/tiddlywikidocs

Fourth

14th November 2021 at 2:36am

Contents Table-of-Contents Demonstrations

This is an example tiddler. See Table-of-Contents Macros (Examples).

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Free Spaced Repetition Scheduler for TiddlyWiki by oflg

5th October 2023 at 8:02am

Community Plugins

TiddlyWiki-based memory programme using advanced FSRS algorithm

https://github.com/open-spaced-repetition/fsrs4tw

Friday

16th November 2021 at 11:11pm

Operator Examples

This example tiddler is used to illustrate some of the Filter Operators.

Full Edition

2nd June 2016 at 7:15am

Editions

The "full" edition of TiddlyWiki consists of all the available languages, themes and plugins for TiddlyWiki. It is intended for use in testing

TiddlyWiki - in particular, for switching between all the available language plugins.

The full edition can be downloaded from:

https://tiddlywiki.com/editions/full/

Full Text Search Plugin by Rob Hoelz

6th January 2021 at 4:10pm

Community Plugins plugins

A plugin to integrate the popular lunr.js search engine, giving TiddlyWiki much more sophisticated search capabilities:

https://hoelz.ro/files/fts.html

> Provides an alternative search result list that orders results by search relevance and ignores differences in word forms (ex. *tag* vs *tags*).
>
> On my personal wiki, I have the problem that there are terms I use across a lot of tiddlers, and sometimes I'll use different forms (such as the aforementioned *tag* vs *tags*). I wanted a plugin to allow me to find the tiddler I'm looking for quickly and didn't require me to worry about how I declined a noun or inflected a verb - so I wrote this plugin, which provides an alternative search list powered by lunr.js.

function Operator

19th April 2023 at 12:31pm

Filter Operators

purpose	apply a function to the input list, and return the result
input	a selection of titles passed as input to the function **F**
parameter	**F** = first parameter is the function name, subsequent parameters are passed to the function by position
output	the selection of titles returned from the function **F**

Learn more about how to use Filters

New in v5.3.0 The **`function`** operator applies a named function to the input titles, and returns the results from the function. The function is invoked once with all of the input titles (in contrast, the filter Operator invokes its function separately for each input title).

The first parameter of the **`function`** operator specifies the name of the function to be called. Subsequent parameters are passed to the function.

The mapping between the parameters is *positional*, with each consecutive parameter specified in the function call mapped to the corresponding parameter in the function definition. Any parameters that are not provided are given their default values.

Compare with the similar filter and subfilter operators which take a filter strings as their parameter instead of a named function, and does not permit parameters to be passed

Examples

Functions

19th April 2023 at 12:31pm

Concepts Reference

Introduction

New in v5.3.0 A **function** is a named snippet of text containing a Filter Expression. Functions can have named parameters which are available within the function as variables.

Functions are usually defined with the Pragma: \function:

```
\function my-function(parameter:"2")
[<parameter>multiply[1.5]]
\end
```

Functions can be invoked in several ways:

- Directly transclude functions with the syntax `<<myfn param:"value">>`
- Assign functions to widget attributes with the syntax `<div class=<<myfn param:"value">>>`
- Invoke functions via the function Operator with the syntax `[function[myfn],[value],...]`
- Directly invoke functions whose names contain a period as custom filter operators with the syntax `[my.fn[value]]` or `[.myfn[value]]`

How Functions Work

Functions are implemented as a special kind of variable. The only thing that distinguishes them from ordinary variables is the way that the parameters are handled.

Funding TiddlyWiki

4th December 2022 at 5:56pm

About HelloThere

TiddlyWiki is more useful to everybody if it is free to use, with no financial barriers to long term adoption. It is not altruism; we believe that removing or reducing barriers to adoption will help to ensure TiddlyWiki's future by making the community larger and stronger.

Nonetheless, TiddlyWiki is a relatively big, complex machine that requires a significant amount of ongoing work to maintain and improve. Some community infrastructure also requires monthly fees to operate (notably the TiddlyWiki forum).

The people in the community that do the work have widely varying needs:

- At one end, a good proportion of the work on TiddlyWiki is performed by community members on a purely voluntary basis. For those people, the satisfaction of helping others is sufficient reward. Indeed, for many people, unpaid voluntary activities are a satisfying antidote to everyday paid work
- At the other extreme, JeremyRuston and some other contributors are trying to make a full-time living working on TiddlyWiki by offering commercial products and services around it
- In between, there are other people who would appreciate an ocassional token to reward them for their work

To support these needs in the community, we have two initiatives:

- We use Open Collective to collect donations for the infrastructure costs of the Community and to crowdfund specific developments by individuals or organisations
- The TiddlyWiki Marketplace provides a shop window for individuals and organisations offering commercial products and services

Funding.png

picture



Future Proof

29th March 2017 at 12:44pm

Features

From Network World magazine in January 2016:

> Way back in the mists of time (actually, January 2009) I wrote about a really cool tool called TiddlyWiki, a “non-linear personal web notebook”. Fast forward to today and I just had an out of body experience: Completely by accident I found a TiddlyWiki that I started when I wrote that piece and it still works!

> Finding code that works flawlessly after just two or three years is magical enough but after seven years?! And given that TiddlyWiki is written as a single page Web application and considering how different browsers are now than they were in 2009, the fact that the old version of TiddlyWiki still works is not short of miraculous.

TiddlyWiki is designed with the long term needs of its users in mind. Because it is OpenSource and needs no infrastructure, we can be confident that all we'll need to access a TiddlyWiki file even in the far future is an ordinary HTML browser. If you're starting to use TiddlyWiki at the beginning of your career you can be confident that it will carry you through to retirement.

Generating Static Sites with TiddlyWiki

21st April 2020 at 2:34am

TiddlyWiki on Node.js

TiddlyWiki5 can be used to generate static HTML representations of a TiddlyWiki that doesn't need JavaScript. This process requires that TiddlyWiki be installed on Node.js on your local system. See Installing TiddlyWiki on Node.js for details.

There is much flexibility in how the static HTML is generated. The following scenarios are all illustrated on https://tiddlywiki.com.

Wiki Snapshots and Tiddler Snapshots

You can explore a static representation of the main TiddlyWiki site at https://tiddlywiki.com/static.html. That file is a static snapshot of the current DefaultTiddlers. Any tiddlers that it links to are referred to via URLs of the form `/static/HelloThere.html` that point to static snapshots of individual tiddlers. The tiddler HTML files reference a `static.css` stylesheet file.

The following commands are used to generate the sample static version of the TiddlyWiki5 site:

```
tiddlywiki wikipath --rendertiddlers '[!is[system]]' $:/core/templates/static.tiddler.html static text/
plain
tiddlywiki wikipath --rendertiddler $:/core/templates/static.template.html static.html text/plain
tiddlywiki wikipath --rendertiddler $:/core/templates/static.template.css static/static.css text/plain
```

The first RenderTiddlersCommand generates the HTML representations of individual tiddlers, the second RenderTiddlerCommand saves the static version of the DefaultTiddlers, and the final RenderTiddlerCommand saves the stylesheet. (All the files are placed in the `output` folder of the wiki folder).

Wiki Snapshot with Internal Links

It is also possible to produce a single HTML file that contains static representations of tiddlers, and uses standard HTML anchor links to jump between them.

For example: https://tiddlywiki.com/alltiddlers.html

The example is built by the following commands:

```
--rendertiddler $:/core/templates/alltiddlers.template.html alltiddlers.html text/plain
```

GenesisWidget

14th December 2023 at 10:37am

Widgets

Introduction

Introduced in v5.2.4 The **$genesis** widget allows the dynamic construction of another widget, where the name and attributes of the new widget can be dynamically determined, without needing to be known in advance.

Content and Attributes

The content of the **$genesis** widget is used as the content of the dynamically created widget.

Attribute	Description
$type	The type of widget or element to create (an initial `$` indicates a widget, otherwise an HTML element will be created)
$remappable	Set to "no" to prevent the generated widget from being affected by any custom widget overrides. Needed when invoking the original widget within a custom widget definition
$names	An optional filter evaluating to the names of a list of attributes to be applied to the widget
$values	An optional filter evaluating to the values corresponding to the list of names specified in **$names**
$mode	An optional override of the parsing mode. May be "inline" or "block"
{other attributes starting with $}	Other attributes starting with a single dollar sign are reserved for future use
{attributes starting with $ $}	Attributes starting with two dollar signs are applied as attributes to the output widget, but with the attribute name changed to use a single dollar sign
{attributes not starting with $}	Any other attributes that do not start with a dollar are applied as attributes to the output widget or HTML Element

Introduced in v5.2.6 If the `$type` attribute is missing or blank, the **`$genesis`** widget does not render an intrinsic element, instead just rendering its children.

Note that attributes explicitly specified take precedence over attributes with the same name specified in the `$names` filter.

Examples

copy to clipboard

```
<$genesis $type="div" class="tc-thing" label="Squeak">Mouse</$genesis>
```

That renders as:

Mouse

... and the underlying HTML is:

```
<p><div class="tc-thing" label="Squeak">Mouse</div></p>
```

copy to clipboard

```
\define my-banner(mode:"inline",caption)
<$genesis $type={{{ [<__mode__>match[inline]then[span]else[div]] }}} class="tc-mybanner">
<<__caption__>>
</$genesis>
\end
```

```
<<my-banner caption:"I'm in a SPAN">>

<<my-banner mode:"block" caption:"I'm in a DIV">>
```

That renders as:

I'm in a SPAN

I'm in a DIV

... and the underlying HTML is:

```
<p><span class="tc-mybanner">
I'm in a SPAN
</span></p><p><div class="tc-mybanner">
I'm in a DIV
</div></p>
```

Important



In the following example the widget attribute named `one` is not present in the HTML output. This is because HTML attributes starting with the prefix `on` are removed for security reasons. See HTML in WikiText "Security" for more details. This restriction only affects generated HTML elements, and does not prevent the use of attributes prefixed `on` with other widgets

copy to clipboard

```
<$genesis $type="my-element" $names="one two" $values="1 2">Test Genesis Widget</$genesis>
```

That renders as:

Test Genesis Widget

... and the underlying HTML is:

```
<p><my-element two="2">Test Genesis Widget</my-element></p>
```

get Operator

9th March 2023 at 5:38pm

Field Operators Filter Operators

purpose	select all values of a field in the input titles
input	a selection of titles
parameter	**F** = the name of a field
output	the values of field **F** in each of the input titles

Learn more about how to use Filters

Each input title is processed in turn. If the corresponding tiddler contains field **F**, and the value of this field is not empty, then its value is appended to the output.



Unlike most other Filter Operators, the selection output by **`get`** can contain duplicates. To avoid duplicates, use `get[F]unique[]`.

Examples

get Operator (Examples)

25th March 2023 at 5:34pm

get Operator Operator Examples

`[all[current]get[draft.of]]`

→ the title of the tiddler of which the current tiddler is a draft

Try it

`[get[tags]]`

→ returns the tags of all tiddlers without de-duplication

Try it

`[get[tags]unique[]]`

→ returns the tags of all tiddlers with de-duplication

Try it



If a data tiddler contains a field with an empty value, the empty string is not appended to the results.

`[all[current]get[myfield]]`

→ the empty value of field **myfield** is not returned by the **get** operator

Try it

`[all[current]has:field[myfield]] :map[get[myfield]]`

→ also returns the empty string

Try it

The above example works by first checking if the input title has the field **myfield** and then using the Map Filter Run Prefix to replace the title with their value of that field. If the input tiddler does not have the field, an empty *SELECTION* is returned. The subsequent Map Filter Run Prefix outputs an empty *STRING* when its run returns an empty selection (because the field is empty).

```
[all[tiddlers]] :filter[get[created]compare:date:lt{HelloThere!!created}]
```

→ return all tiddlers that are older than HelloThere

Try it

The above example demonstrates two different ways of accessing field values in filters: Use **get** when the title is not known in advance as with the Filter Filter Run Prefix where **currentTiddler** is set to the current input title. Use a TextReference as an indirect Filter Parameter when the title is known.

Get the Ring

15th November 2021 at 2:32am

task TaskManagementExample

This is a sample task for the TaskManagementExample

getindex Operator

9th March 2023 at 5:38pm

Field Operators Filter Operators

purpose	select all values of a data property in the input titles
input	a selection of titles
parameter	**P** = the name of a property

output	the values of property **P** in each of the input titles

Learn more about how to use Filters

Each input title is processed in turn, and is ignored if it does not denote a data tiddler. If the corresponding tiddler contains property **P**, and the value of this property is not empty, then its value is appended to the output.



Unlike most other Filter Operators, the selection output by **getindex** can contain duplicates. To avoid duplicates, use `getindex[P]unique[]`.

Examples

getindex Operator (Examples)

25th March 2023 at 5:35pm

getindex Operator Operator Examples

`[[$:/palettes/Vanilla]getindex[background]]`

→ returns the value of property **background** of the DataTiddler **$:/palettes/Vanilla**

Try it

`[all[shadows+tiddlers]tag[$:/tags/Palette]getindex[background]]`

→ returns all background colors defined in any of the ColourPalettes (notice the duplicates in the resulting list)

Try it



If a data tiddler contains a property with an empty value, the empty string is not appended to the results.

```
[[ListopsData]getindex[DataIndex]]
```

→ the empty value of the property **DataIndex** in ListopsData is not returned by the **getindex** operator

Try it

```
[[ListopsData]has:index[DataIndex]] :map[getindex[DataIndex]]
```

→ also returns the empty string

Try it

The above example works by first checking if the input title has the property **DataIndex** and then using the Map Filter Run Prefix to replace the title with their value of that property. If the input tiddler does not have the property, an empty *SELECTION* is returned. The subsequent Map Filter Run Prefix outputs an empty *STRING* when its run returns an empty selection (because the property is empty).

```
[[$:/palettes/Vanilla]indexes[]] :filter[[$:/palettes/
Vanilla]getindex<currentTiddler>count[]compare:number:eq[0]]
```

→ returns those colors in **$:/palettes/Vanilla** which are defined, but have no value assigned

Try it

In the above example, **count** is used to check if **getindex** returns a result (i.e. the corresponding property has a value) or not.

Getting Started Video

19th September 2014 at 6:10pm

Videos Working with TiddlyWiki

This brief tutorial takes you through the basics of saving changes with a standalone TiddlyWiki file.

Note that the video is a bit out of date, and will be updated soon!

GettingStarted

19th August 2022 at 6:10am

Working with TiddlyWiki

The easiest way to use TiddlyWiki is to sign up for a free account with Tiddlyhost, an independently run community service. If you find Tiddlyhost useful, please consider donation or sponsorship.

Click here to download an empty copy of TiddlyWiki: Download

The next step is to choose a method for saving changes. There's a wide variety of methods available, with different features and limitations. Click on the badge for a method to see more information about it. You can also click on one of the platform filters to restrict the listing to methods that work with that platform.



Don't attempt to use the browser **File**/**Save** menu option to save changes (it doesn't work)

OS

- ☐ Windows
- ☐ Mac
- ☐ Linux
- ☐ Android
- ☐ iOS

Browser

- ☐ Firefox
- ☐ Chrome
- ☐ Edge
- ☐ Internet Explorer
- ☐ Safari
- ☐ Opera
- ☐ Standalone App

Use the checkboxes to explore the methods of saving that work with your platform(s)

GettingStarted - Android

14th November 2021 at 4:16am

GettingStarted

There are three options for using TiddlyWiki on Android:

Using Firefox and TiddlyFox

(Alternatively, see the video tutorial)

1. Ensure you have the latest version of *Firefox for Android*
   - http://getfirefox.com
2. Install the latest release of the TiddlyFox extension from:
   - https://addons.mozilla.org/en-GB/firefox/addon/tiddlyfox/
3. Install this extension to be able to save the TiddlyWiki file locally:

    - https://addons.mozilla.org/en-GB/android/addon/save-link-menus/
4. *Download* an empty TiddlyWiki by saving this link:
    - https://tiddlywiki.com/empty.html

    > (to save the link, ensure you've installed the "save-link-menus" extension and then long-press on the link and choose "Save link")

5. When the file has downloaded, click on it within the notification tray or the download manager application
6. Choose to open the file in Firefox (rather than the default Android viewer)
7. Click **OK** in response to the prompt from TiddlyFox that asks whether to enable saving for this file
8. Try creating a new tiddler using the **new tiddler** + button in the sidebar. Type some content for the tiddler, and click the ✓ **ok** button
9. Save your changes by clicking the ⊘ **save changes** button in the sidebar
    - Look for the yellow notification **Saved wiki** at the top right of the window
10. Refresh the browser window to verify that your changes have been saved correctly

Using the *AndTidWiki* App

The Tiddloid and Tiddloid Lite app are Android apps that makes it possible to edit and save changes to TiddlyWiki HTML files.

- For more information: Tiddloid Tiddloid Lite.
- Download apks: Tiddloid, TiddloidLite

Features

- Create new TiddlyWiki importing latest edition from internet
- Import existing TiddlyWikis stored on device/internal storage. (TiddloidLite supports external SD card too)
- Fork interesting ~Tiddlywikis from internet (Supports TW5 only)
- TiddlyWiki detection
- Locally stored ~Tiddlywikis are updated simultaneously on saving changes to TiddlyWikis imported to the app
- Backup system that is compatible with TiddlyDesktop, the desktop TiddlyWiki saver
- Creating shortcuts to existing TiddlyWiki on Android Homepage
- TiddloidLite supports cloud storages like GDrive and OneDrive

Please note

- Tiddloid Lite feature better support for devices running Android Q or later. It also supports cloud storages like GDrive and OneDrive, while Tiddloid keeps the compatibility to TiddlyWikiClassic. To know more about differences between Tiddloid and Tiddloid Lite, please visit Tiddloid's homepage.
- You should keep the `.html` or `.htm` extension of the files to be imported.

Using Node.js in Termux

Termux is and open source android application providing limited Unix environment enabling users to install Node.js and npm modules in android. Users can install and run TiddlyWiki on Node.js using Termux.

Instructions

- Download and install Termux from Google Play Store or Fdroid
- Open termux and run the following commands one by one

```
apt update
apt upgrade
apt install nodejs
npm install -g tiddlywiki
```

- If you need to create/serve **TiddlyWiki on Node.js** from the internal storage, you need to give termux storage permission by running the following command in termux

```
termux-setup-storage
```

- Now you can create and serve **TiddlyWiki on Node.js** from internal storage. In the example given below, user is creating a new wiki called "mynewwiki" in his internal folder.

```
cd storage/shared
tiddlywiki mynewwiki --init server
tiddlywiki mynewwiki --listen
```

- Visit http://127.0.0.1:8080/ in your browser
- From then on, as long as *Termux* is not closed, you may access your wiki anytime from your favourite Web browser pointing on the expected address and port.

- For more information regarding TiddlyWiki on Node.js, please see Installing TiddlyWiki on Node.js

- For more information and tips regarding termux, please refer to Termux wiki



In *Termux*, you may as well install *git*, *emacs* or *vi*, in order to edit and maintain individual tiddler files. This would probably require that you also attach a more powerful keyboard to your Android, like the Hacker's Keyboard application or a Bluetooth external device.

GettingStarted - Chrome

14th November 2021 at 4:16am

GettingStarted

TiddlyWiki on Google Chrome can only save changes using the HTML5-compatible saver module.

This is the default method of saving if no other method is installed. It uses your browser's built-in "download a file" handler, and has the advantage of working on almost all desktop browsers, and many mobile browsers.

1. *Download* an empty TiddlyWiki by clicking this button:

   > Download
   >
   > If the button doesn't work save this link: https://tiddlywiki.com/empty.html
   >
   > Your browser may ask you to accept the download before it begins

2. Locate the file you just downloaded
   - You may rename it, but be sure to keep the `.html` or `.htm` extension
3. Open the file in your browser

4. Try creating a new tiddler using the **new tiddler** + button in the sidebar. Type some content for the tiddler, and click the ✓ **ok** button
5. Save your changes by clicking the **save changes** button in the sidebar
6. Your browser will download a new copy of the wiki incorporating your changes
7. Locate the newly downloaded file and open it in your browser
8. Verify that your changes have been saved correctly

Tip: most browsers have an option to prompt each time for the download location. This allows you to select the existing version of the file and replace it.

GettingStarted - Firefox

3rd August 2023 at 11:30pm

GettingStarted

Firefox provides the best user experience for using TiddlyWiki with the following browser extensions:

- file-backups
- savetiddlers

1. Restart *Firefox*
2. *Download* an empty TiddlyWiki by clicking this button:

   Download

3. Locate the file you just downloaded

    - You may rename it, but be sure to keep the `.html` or `.htm` extension
4. Open the file in *Firefox*
5. Try creating a new tiddler using the **new tiddler** + button in the sidebar. Type some content for the tiddler, and click the ✓ **ok** button
6. Save your changes by clicking the ⊘ **save changes** button in the sidebar
    - Look for the yellow notification **Saved wiki** at the top right of the window
7. Refresh the browser window to verify that your changes have been saved correctly

GettingStarted - Internet Explorer

14th November 2021 at 4:16am

GettingStarted

1. Install the TiddlyIE add-on from:
    - https://github.com/davidjade/TiddlyIE/releases
2. Restart Internet Explorer. IE will prompt you to enable the TiddlyIE add-on.

    > You may also see a prompt to enable the *Microsoft Script Runtime*

3. *Download* an empty TiddlyWiki by saving this link:

    > https://tiddlywiki.com/empty.html

4. Locate the file you just downloaded
    - You may rename it, but be sure to keep the `.html` or `.htm` extension
5. Open the file in Internet Explorer
6. Try creating a new tiddler using the **new tiddler** + button in the sidebar. Type some content for the tiddler, and click the ✓ **ok** button

7. Save your changes by clicking the ⊘ **save changes** button in the sidebar. Internet Explorer will ask for your consent to save the file locally by presenting a file **Save As** dialog.
8. Refresh the browser window to verify that your changes have been saved correctly

The Windows HTA Hack describes an alternative method of using TiddlyWiki with Internet Explorer.

GettingStarted - iOS

14th November 2021 at 4:16am

GettingStarted

The iPad/iPhone app **Quine 2** makes it possible to view, edit and then save changes to TiddlyWiki5 on iOS. Download it here.

Instructions for use:

1. Open Quine 2
2. Tap the + toolbar button to create and open a new TiddlyWiki
3. From the file list tap an existing TiddlyWiki file to open it
4. Edit the TiddlyWiki as normal, and save as normal using either Autosave or the TiddlyWiki save button ⊘
5. Tap the left hand "Documents" toolbar button to close an open TiddlyWiki

- Quine 2 works natively in iOS with the local file system and the iCloud file system
- Quine 2 also allows you to open, edit and save TiddlyWiki files stored with cloud file providers
- Quine 2 allows you to follow embedded WikiText links and canonical links to external files for cloud-like file providers which support "folder level sharing".
    - This includes the apps "Secure Shellfish" and "Working Copy". Most providers, though, do not allow apps like Quine 2 to access

linked files this way.

- If you wish to enable such links for "well behaved" file providers, toggle "on" the "Enable folder selection for out-of-sandbox links" setting in iOS Settings for Quine 2

Note that Quine is published independently of TiddlyWiki

GettingStarted - Node.js

14th November 2021 at 4:16am

GettingStarted

1. Install Node.js
    - Linux:

        > *Debian/Ubuntu*:
        > `apt install nodejs`
        > May need to be followed up by:
        > `apt install npm`
        >
        > *Arch Linux*
        > `yay -S tiddlywiki`
        > (installs node and tiddlywiki)

    - Mac

        > `brew install node`

- Android

  > Termux for Android

- Other

  > See http://nodejs.org

2. Open a command line terminal and type:

   > `npm install -g tiddlywiki`
   >
   > If it fails with an error you may need to re-run the command as an administrator:
   >
   > `sudo npm install -g tiddlywiki` (Mac/Linux)

3. Ensure TiddlyWiki is installed by typing:

   > `tiddlywiki --version`

   - In response, you should see TiddlyWiki report its current version (eg "5.3.3". You may also see other debugging information reported.)

4. Try it out:
   1. `tiddlywiki mynewwiki --init server` to create a folder for a new wiki that includes server-related components
   2. `tiddlywiki mynewwiki --listen` to start TiddlyWiki
   3. Visit http://127.0.0.1:8080/ in your browser
   4. Try editing and creating tiddlers

5. Optionally, make an offline copy:
    - click the **save changes** button in the sidebar, **OR**
    - `tiddlywiki mynewwiki --build index`

The `-g` flag causes TiddlyWiki to be installed globally. Without it, TiddlyWiki will only be available in the directory where you installed it.



If you are using Debian or Debian-based Linux and you are receiving a `node: command not found` error though node.js package is installed, you may need to create a symbolic link between `nodejs` and `node`. Consult your distro's manual and `whereis` to correctly create a link. See github issue 1434.

Example Debian v8.0: `sudo ln -s /usr/bin/nodejs /usr/bin/node`



You can also install prior versions like this:
`npm install -g tiddlywiki@5.1.13`

GettingStarted - Online

14th November 2021 at 4:16am

GettingStarted

TiddlyWiki can also be hosted on online services such as Dropbox and Cloudant.

Dropbox (syncs the entire HTML file)

To edit files stored in your Dropbox, go to http://tiny.cc/tw5inthesky/. This will allow you to open any HTML files found.

Cloudant (or another *CouchDB*-type server) (syncs individual tiddlers)

http://noteself.github.io/

GettingStarted - Safari

14th November 2021 at 4:16am

GettingStarted

TiddlyWiki on Safari can only save changes using the manual HTML5-compatible fallback saver module.

getvariable Operator

30th March 2019 at 11:01am

Filter Operators Special Operators

purpose	select all values of variables named in the input titles
input	a selection of variable names
parameter	ignored
output	the values of each of the variables named in the input titles (or blank if the variable is not defined)

Learn more about how to use Filters

ⓘ **Introduced in v5.1.20** The usual way to retrieve a variable value within a filter is with the angle brackets notation. For example, `[<currentTiddler>]` will retrieve the value of the variable called `currentTiddler`.

The `getvariable` filter operator provides an alternative way to retrieve a variable. For example, `[[currentTiddler]getvariable[]]` is

another way to retrieve the value of the variable `currentTiddler`.

The advantage of `getvariable` is that it makes it possible to work with variables whose name is computed, and not known in advance. For example, `[<myvariable>getvariable[]]` gets the value of the variable whose name is given in the variable `myvariable`.

Examples

getvariable Operator (Examples)

30th March 2019 at 11:01am

getvariable Operator Operator Examples

`[[currentTiddler]getvariable[]]`

→ returns the value of the variable **currentTiddler**

Try it

Git

23rd July 2023 at 9:30am

Definitions

> Git (/ɡɪt/) is a distributed version control system that tracks changes in any set of computer files, usually used for coordinating work among programmers collaboratively developing source code during software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows (thousands of parallel branches running on different computers).

> https://en.wikipedia.org/wiki/Git

GitHub

23rd July 2023 at 9:43am

Definitions

GitHub is a hosting service for distributed projects that use git as their version-control system. It allows free hosting and management of open-source projects and facilitates collaborative development on the source code. Using GitHub for non-open-source endeavours requires additional fees.

The code and documentation of TiddlyWiki is hosted on GitHub at:

https://github.com/Jermolene/TiddlyWiki5

GitHub also offer a free web hosting service called GitHub Pages that can be used directly from the single file configuration. See Saving to a Git service.

An alternative to GitHub is GitLab

GitHub Saver Tutorial by Mohammad

23rd July 2023 at 9:45am

Tutorials

GitHub Saver is a step by step tutorial that shows how to integrate Tiddlywiki 5 and GitHub Pages to create websites hosted on https://github.com/.

https://kookma.github.io/TW5-GitHub-Saver/

This instruction is based on Tiddlywiki single html file model, while it can use subfolder for extra materials like images, audios, videos, pdfs,... in separate folders.

Other tutorials

Tiddlywiki, Travis-CI and GitHub Pages

https://kookma.github.io/Tiddlywiki-Travis-CI/

This wiki shows how to set up websites hosted on GitHub Pages using Travis-CI and Tiddlywiki 5 on Node.js.

Tiddlywiki and GitHub Pages

https://kookma.github.io/Tiddlywiki-and-GitHub-Pages/

This instruction is based on local edit, save and push to GitHub. It does NOT use the new GitHub Saver mechanism (requires TW 5.1.20+) which lets edit and save directly from Tiddlywiki!

GitLab

23rd July 2023 at 9:39am

Definitions

> GitLab Inc. is an open-core company that operates GitLab, a DevOps software package which can develop, secure, and operate software. The open source software project was created by Ukrainian developer Dmytro Zaporozhets and Dutch developer Sytse Sijbrandij
>
> https://en.wikipedia.org/wiki/GitLab

Both GitLab and GitHub use Git a distributed version control system, that can be used to store, view and edit TiddlyWiki wikis using GitLab Pages

Learn more at: Saving to a Git service

Go to Mordor

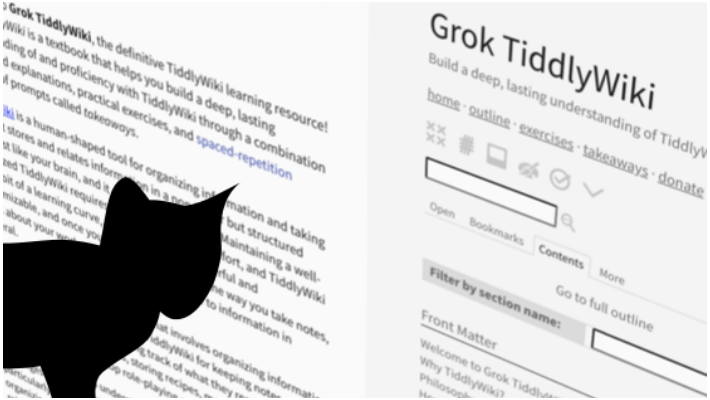
15th November 2021 at 7:10pm

done task TaskManagementExample

This is a sample task for the TaskManagementExample

Grok TiddlyWiki Banner

picture



GroupedLists

22nd February 2015 at 12:01am

Lists ListWidget

The following sidebar tabs give examples of grouped lists created by nesting.

Types Tab

For the **Types** tab, the outer list filter as shown below selects each discrete value found in the `type` field. The inner list filter selects all the

(non-system) tiddlers with that type.

```
\whitespace trim
<$list filter={{$:/core/Filters/TypedTiddlers!!filter}}>
<div class="tc-menu-list-item">
<$view field="type"/>
<$list filter="[type{!!type}!is[system]sort[title]]">
<div class="tc-menu-list-subitem">
<$link to={{!!title}}><$view field="title"/></$link>
</div>
</$list>
</div>
</$list>
```

Recent Tab

The list in the **Recent** tab is generated using the **timeline** macro. Here, the outer list filter selects each discrete day found in the `modified` field, while the inner list filter selects all the tiddlers dated the same day in the `modified` field.

$:/core/macros/timeline:

```
<!-- Override one or both of the following two macros with a global or local macro of the same name
if you need to change how titles are displayed on a timeline -->

\define timeline-title() <$view field="title"/>
```

```
\define timeline-link() <$link to={{!!title}}><<timeline-title>></$link>
\define timeline(limit:"100",format:"DDth MMM YYYY",subfilter:"",dateField:"modified")
\whitespace trim
<div class="tc-timeline">
<$list filter="[!is[system]$subfilter$has[$dateField$]!
sort[$dateField$]limit[$limit$]eachday[$dateField$]]">
<div class="tc-menu-list-item">
<$view field="$dateField$" format="date" template="$format$"/>
<$list filter="[sameday:$dateField${!!$dateField$}!is[system]$subfilter$!sort[$dateField$]]">
<div class="tc-menu-list-subitem">
<<timeline-link>>
</div>
</$list>
</div>
</$list>
</div>
\end
```

GuerillaWiki

17th August 2014 at 4:10pm

Definitions

TiddlyWiki makes a great GuerillaWiki in situations where it is not practical to use a traditional wiki.

For instance, in a corporate setting, persuading an over-worked IT department to install a Wiki server for you is seldom going to be possible overnight. And if your PC is locked down you can't install a conventional Wiki yourself. Equally, you can't go and use one of the public hosted Wiki services because your Information Security department would not allow all that corporate data to flow into an outside server.

TiddlyWiki slices through those barriers by being usable on virtually all PCs.

Hamlet

4th March 2023 at 5:21pm

Operator Examples


```
{
    "Shakespeare-old": "Hamlet: Do you see yonder cloud that's almost in shape of a camel?\nPolonius: By the mass, and 'tis like a camel, indeed.\nHamlet: Methinks it is like a weasel.\nPolonius: It is backed like a weasel.\nHamlet: Or like a whale?\nPolonius: Very like a whale.\n-- Shakespeare",
    "Shakespeare-new": "Hamlet: Do you see the cloud over there that's almost the shape of a camel?\nPolonius: By golly, it is like a camel, indeed.\nHamlet: I think it looks like a weasel.\nPolonius: It is shaped like a weasel.\nHamlet: Or like a whale?\nPolonius: It's totally like a whale.\n-- Shakespeare",
    "Trekkie-old": "Kirk: Do you see yonder cloud that's almost in shape of a Klingon?\nSpock: By the mass, and 'tis like a Klingon, indeed.\nKirk: Methinks it is like a Vulcan.\nSpock: It is backed like a Vulcan.\nKirk: Or like a Romulan?\nSpock: Very like a Romulan.\n-- Trekkie"
}
```


Hard and Soft Links

26th February 2015 at 5:31pm

Concepts

A **hard link** is one that can be detected by a superficial examination of WikiText.

A link is **soft** if it is:

- contained in text trancluded from elsewhere
- supplied via a macro or variable
- generated by a link widget whose **`to`** attribute is a transclusion, macro or variable



Soft links are not detected by link-related filter operators such as **`backlinks`**, **`links`**, **`all`** and **`is`**.

Hard Linebreaks in WikiText

14th December 2013 at 6:01pm

WikiText

The usual handling of paragraphs in wikitext causes single line breaks to be ignored, and double linebreaks to be interpreted as the end of a paragraph.

This behaviour isn't convenient when dealing with material that incorporates hard linebreaks - for instance, poetry. You can mark a block of content as containing hard line breaks like this:

copy to clipboard

```
"""
This is a line
```

```
and this is a new line
while this is yet another line
and this is the final one
apart from this one
"""
```

That renders as:

This is a line
and this is a new line
while this is yet another line
and this is the final one
apart from this one

... and the underlying HTML is:

```
<p>This is a line<br>and this is a new line<br>while this is yet another line<br>and this is the final
one<br>apart from this one<br></p>
```

Hard Linebreaks with CSS

26th January 2017 at 10:10pm

Hard Linebreaks in WikiText

Since TiddlyWiki Version 5.1.16, it's possible to apply customs styles with `data-` attributes. see: How to apply custom styles

The Custom data-styles stylesheet tiddler contains the following definition:

```
[data-tags*="example-hardlinebreaks"] .tc-tiddler-body {
  word-break: normal;
  word-wrap: break-word;
  white-space: pre-wrap;
}
```

If rendered without the tag, the content of "Hard Linebreaks with CSS - Example" will be shown like this:

> The following text is rendered without any special WikiText syntax. So most of the WikiText functionality will be accessible. This is a line and this is a new line while this is yet another line and this is the final one apart from this one

---

Open the tiddler to see the custom CSS rules applied: Hard Linebreaks with CSS - Example

Hard Linebreaks with CSS - Example

26th January 2017 at 10:10pm

example-hardlinebreaks Hard Linebreaks in WikiText

The following text is rendered without any special WikiText syntax. So most of the WikiText functionality will be accessible.

This is a line
and this is a new line
while this is yet another line
and this is the final one

apart from this one

has Operator

6th March 2023 at 3:32pm

Common Operators Field Operators Filter Operators Negatable Operators

purpose	filter the input by field existence
input	a selection of titles
suffix	**S** = ⓘ **Introduced in v5.1.14** optionally, the keyword `field` or ⓘ **Introduced in v5.1.22** optionally, the keyword `index`
parameter	**F** = the name of a field or, optionally an index
output	**without suffix** » those input tiddlers in which field **F** has a non-empty value **suffix** `field` » those input tiddlers in which field **F** exists **suffix** `index` » those input data tiddlers in which index **F** exists
`!` **output**	**without suffix** » those input tiddlers in which field **F** does *NOT* exist or has an empty value **suffix** `field` » those input tiddlers in which field **F** does *NOT* exist **suffix** `index` » those input data tiddlers in which index **F** does *NOT* exist

Learn more about how to use Filters

Examples

has Operator (Examples)

11th November 2015 at 4:02pm

has Operator Operator Examples

```
[has[color]]
```

Try it

```
[tag[Concepts]!has[modified]]
```

Try it

```
[has:field[emptyfield]]
```

Try it

```
[all[current]!has:field[doesntexist]]
```

Try it

```
[all[tiddlers+shadows]has:index[foreground]]
```

Try it

```
[all[current]!has:index[doesntexist]]
```

Try it

haschanged Operator

8th February 2015 at 8:18pm

Filter Operators Negatable Operators

purpose	filter the input by tiddler modification status
input	a selection of titles
parameter	none
output	those input tiddlers that have been modified during this session
! **output**	those input tiddlers that have *NOT* been modified during this session

Learn more about how to use Filters

A tiddler is deemed to have been modified if it has been written back to the wiki since the start of the current TiddlyWiki session. If you edit a tiddler and immediately store it again without making any changes, that is enough to mark it as modified.

Examples

Headings in WikiText

13th May 2022 at 1:08pm

WikiText

Headings are specified with one up to six leading `!` characters:

```
! This is a level 1 heading

!! This is a level 2 heading

!!! This is a level 3 heading
```

CSS classes can be assigned to individual headings like this:

copy to clipboard

```
!!.myStyle This heading has the class `myStyle`
```

That renders as:

This heading has the class `myStyle`

... and the underlying HTML is:

```
<h2 class="myStyle">This heading has the class <code>myStyle</code></h2>
```

Height of text editor

17th August 2016 at 12:54pm

Editor toolbar

The button  let you adjust the height of the text input field.

HelloThere

23rd December 2023 at 11:22am

TableOfContents

Welcome to TiddlyWiki, a unique non-linear notebook for capturing, organising and sharing complex information

Use it to keep your to-do list, to plan an essay or novel, or to organise your wedding. Record every thought that crosses your brain, or build a flexible and responsive website.

TiddlyWiki lets you choose where to keep your data, guaranteeing that in the decades to come you will still be able to use the notes you take today.

Quick Start

Tiddlyhost

The easiest way to get started with an online TiddlyWiki

Create Account

Desktop

Download the official desktop application for macOS, Windows and Linux

Download

DIY

Find the configuration that is right for you to get the full benefits of TiddlyWiki

Explore

Xememex

For companies and teams, a multiuser TiddlyWiki from Federatial

Find out more

中文社区
Chinese Community

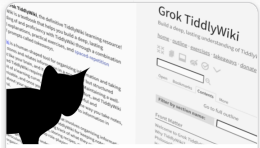
Find Out More



TiddlyWiki



Introduction to



Grok TiddlyWiki



What's New in v5.3.3

Newsletter

Subscribe to the TiddlyWiki Newsletter, a fortnightly summary of the most interesting and relevant news from the TiddlyWiki community

TiddlyWiki

Short video introducing basic TiddlyWiki concepts

Everything you need to know to get the best out of TiddlyWiki

The latest version v5.3.3 of TiddlyWiki (released on 23rd December 2023)



links.tiddlywiki.org

Links to TiddlyWiki-related content collected by the community



How is TiddlyWiki Funded?

Find out how you can help support TiddlyWiki financially



TiddlyWiki Marketplace

Explore commercial products and services for TiddlyWiki



Federatial

Support the development of TiddlyWiki by hiring Jeremy Ruston through Federatial Limited

Community

TalkTW YouTube Twitter Mastodon GitHub Gitter Reddit

Open Collective

Testimonials & Reviews

Joe Armstrong, Co-inventor of Erlang

"The TiddlyWiki is the best software I've ever found for organising my ideas.

It's well worth spending an hour or so playing with it to see how it can help you. This will be time well-spent and will change how you think and how you organise your ideas."

Mark Gibbs, Network World

"TiddlyWiki gets a Gearhead rating of 6 out of 5 (it's that good).

Finding code that works flawlessly after just two or three years is magical enough but after seven years?!"

Product Hunt



HelloThumbnail

14th April 2015 at 9:09am

HelloThumbnail - Federatial

HelloThumbnail

Support the development of TiddlyWiki by hiring Jeremy Ruston through Federatial Limited

HelloThumbnail - Funding

HelloThumbnail

Find out how you can help support TiddlyWiki financially

HelloThumbnail - Grok TiddlyWiki

HelloThumbnail

Everything you need to know to get the best out of TiddlyWiki

HelloThumbnail - Introduction Video

HelloThumbnail

Short video introducing basic TiddlyWiki concepts

HelloThumbnail - Latest Version

HelloThumbnail

The latest version v5.3.3 of TiddlyWiki (released on 23rd December 2023)

HelloThumbnail - Marketplace

HelloThumbnail

Explore commercial products and services for TiddlyWiki

HelloThumbnail - Newsletter

HelloThumbnail

Subscribe to the TiddlyWiki Newsletter, a fortnightly summary of the most interesting and relevant news from the TiddlyWiki community

HelloThumbnail - TiddlyWikiLinks

HelloThumbnail

Links to TiddlyWiki-related content collected by the community

HelpCommand

25th February 2014 at 8:57pm

Commands

Displays help text for a command:

```
--help [<command>]
```

If the command name is omitted then a list of available commands is displayed.

HelpingTiddlyWiki

10th October 2015 at 3:12pm

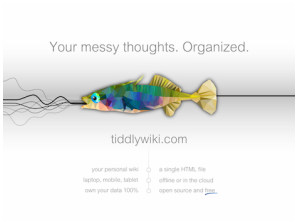
Community

If you find TiddlyWiki useful, there are lots of ways you can help assure its future and make it better.

Teach and Tell

OpenSource projects like TiddlyWiki thrive on the feedback and engagement of users. TiddlyWiki becomes more useful to everyone as more and more people use it. So, if you find TiddlyWiki useful, spread the word. The best possible way to assure its future is for it to become a hundred times more popular than before.

- Tweet
- Tweet about TiddlyWiki: I love TiddlyWiki because...
- Star 7.7k
- Star the TiddlyWiki5 GitHub Repository
- Display the TiddlyWiki Poster



Help improve the documentation and code

There are many ways you can contribute to TiddlyWiki:

- Writing tutorials
- Contributing to the documentation on tiddlywiki.com
- Making video screencasts
- Curating relevant links, hints and tips on a wiki

The main TiddlyWiki documentation and code lives on GitHub, and welcomes contributions:

- https://github.com/Jermolene/TiddlyWiki5

Hidden Setting: Default Tiddler Icon

6th March 2020 at 3:50pm

Hidden Settings

A default tiddler icon can be specified by creating a tiddler called *$:/config/DefaultTiddlerIcon* containing the title of the tiddler containing the icon.

Hidden Setting: Disable Drag and Drop

19th May 2021 at 5:54pm

Hidden Settings

Introduced in v5.1.22 To disable all the drag and drop operations that are built into the core, set the following tiddler to "no":

$:/config/DragAndDrop/Enable

To selectively re-enable drag and drop for an instance of the list-tagged-draggable Macro or list-links-draggable Macro you must ensure that the

variable `tv-enable-drag-and-drop` is set to `yes` for the scope of the macro invocation. For example, note how it is still possible to use drag and drop within this list even if *$:/config/DragAndDrop/Enable* is set to "no":

copy to clipboard

```
<$set name="tv-enable-drag-and-drop" value="yes">

<<list-tagged-draggable tag:"HelloThere">>

</$set>
```

That renders as:

A Gentle Guide to TiddlyWiki
Discover TiddlyWiki
Some of the things you can do with TiddlyWiki
Ten reasons to switch to TiddlyWiki
Examples
What happened to the original TiddlyWiki?
Funding TiddlyWiki
Open Collective

Note that when using the DropzoneWidget and the DroppableWidget directly the **enable** attribute works independently of the global setting.

Hidden Setting: Disable Lazy Loading

25th October 2019 at 12:01pm

Hidden Settings

LazyLoading can be disabled by setting this value to `yes`

$:/config/SyncDisableLazyLoading

Hidden Setting: Enable File Import in Editor

19th May 2021 at 6:00pm

Hidden Settings

Introduced in v5.2.0

$:/config/Editor/EnableImportFilter

This filter determines whether dragging and dropping files in the editor works for a given tiddler or not. A non-empty result enables drag and drop in the editor for that tiddler. This filter is used in such a manner that it respects the global drag and drop setting.

Hidden Setting: Filename for Save Wiki Button

22nd January 2019 at 1:00am

Hidden Settings

Introduced in v5.1.19 When saving a TiddlyWiki using the **save changes** button, the default file name used for saving is constructed

using the Title of the TiddlyWiki as entered in the **Control Panel**, which is stored in the tiddler $:/SiteTitle.

The value used is in the form `{{$:/SiteTitle}}.html`. This causes the filename to be constructed from the site Title with a `.html` extension.

If a tiddler named *$:/config/SaveWikiButton/Filename* is created, the text in that tiddler will be used as the default file name used for saving the TiddlyWiki.

Hidden Setting: HTML Parser Sandbox

11th April 2021 at 12:01pm

Hidden Settings

Introduced in v5.2.0 By default, tiddlers with the type `text/html` are displayed in an iframe with the sandbox attribute set to the empty string. This causes all security restrictions to be applied, disabling many features such as JavaScript, downloads and external file references. This is the safest setting.

To globally disable the sandbox, set the tiddler *$:/config/HtmlParser/DisableSandbox* to `yes`. This will mean that the code in the iframe has full access to TiddlyWiki's internals, which means that a malicious HTML page could exfiltrate data from a private wiki.

To keep the sandbox but control which restrictions are applied, ensure that *$:/config/HtmlParser/DisableSandbox* is not set to `yes`, and then set *$:/config/HtmlParser/SandboxTokens* to the desired list of tokens from the MDN documentation.

Note that these are global settings. To control the sandboxing on an individual tiddler basis will require a custom `<iframe>` to be used.

Hidden Setting: More Tabs Horizontal

28th November 2022 at 10:29am

Hidden Settings

Set the *$:/config/ui/SideBar/More/horizontal* tiddler to **yes**, to align the **More -> Tabs** in horizontal orientation.

Hidden Setting: Navigate on Enter

8th November 2020 at 11:49am

Hidden Settings

The configuration Tiddler *$:/config/Search/NavigateOnEnter/enable* can be used to enable (if set to **yes**) Navigation to and Creation of missing Tiddlers in the various search input fields when hitting `Enter` respectively `ctrl-Enter`

Hidden Setting: New-Image Type

16th November 2021 at 11:25pm

Hidden Settings

By default new images are created with the image-type `jpeg`

The hidden setting in **$:/config/NewImageType** can be set to another Image-Type which is used for new Image Tiddlers, like `png`

Hidden Setting: Retain Story ordering

Hidden Settings

Introduced in v5.2.4 The tiddler **$:/config/ControlPanel/Basics/DefaultTiddlers/RetainStory** contains the value that is assigned to $:/DefaultTiddlers when clicking the "retain story ordering" button in **$:/ControlPanel** under the **Info** -> **Basics** tab.

Hidden Setting: Scroll Top Adjustment

16th August 2018 at 1:06pm

Hidden Settings

As part of navigating to a tiddler, TiddlyWiki scrolls the page so that the top of the tiddler lines up with the top of the browser window. This means that the upper part of tiddlers can be obscured when using custom `position:fixed` toolbars at the top of the window.

To adjust the scroll position to allow for the toolbar, add the CSS class `tc-adjust-top-of-scroll` to the toolbar. TiddlyWiki's scrolling mechanism will then dynamically adjust the scroll position by the height of that element.

Hidden Setting: Search AutoFocus

19th June 2015 at 6:26pm

Hidden Settings

When TiddlyWiki first opens in the browser the search box is automatically given the focus by default. If this causes problems, you can change the default by changing this value from **true** to **false**:

$:/config/Search/AutoFocus

Hidden Setting: Search Minimum Length

11th October 2016 at 9:42am

Hidden Settings

Introduced in v5.1.14 Controls the minimum length of a search string before results are displayed.

Defaults to "3".

$:/config/Search/MinLength

Hidden Setting: Show Edit Preview per Tiddler

29th October 2021 at 11:03am

Hidden Settings

Controls whether the tiddler editing preview is controlled globally (the default) or on a per-tiddler basis.

Set it to **yes** to enable per-tiddler mode.

$:/config/ShowEditPreview/PerTiddler

Hidden Setting: Sync Logging

3rd September 2019 at 9:23pm

Hidden Settings

Specifies whether Syncadaptor should log information to the browser's developer console or not.

Defaults to `yes`. Set to `no` to disable logging.

Changing needs restart to take effect.

$:/config/SyncLogging

Hidden Setting: Sync Polling Interval

3rd September 2019 at 9:23pm

Hidden Settings

Specifies the interval at which Syncadaptor synchronizes tiddlers between the server and the browser.

Defaults to "60000" (60 * 1000 ms = 1 min).

Changing needs restart to take effect.

$:/config/SyncPollingInterval

Hidden Setting: Sync System Tiddlers From Server

9th September 2022 at 11:43am

Hidden Settings

ⓘ **Introduced in v5.1.23** Determines whether system tiddlers are synced from the server under Node.js. (Note that this is a one-way setting; system tiddlers are always synced *to* the server).

- `no` – system tiddlers are not synced from the server (default)
- `yes` – system tiddlers are synced from the server

Engaging sync of system tiddlers means that tiddlers such as $:/StoryList and $:/HistoryList get synced, which can lead to unexpected outcomes when multiple users are connected to the same server at the same time (it means that the story sequence is synced between all the users).

$:/config/SyncSystemTiddlersFromServer

Hidden Setting: Tab Index for Edit-Inputs

2nd July 2019 at 9:48am

Hidden Settings

The input fields of Tiddlers that are being edited can be assigned a **tabindex** value (preferably 1) so that the `Tab` key can be used to directly move the focus from one input to the next or backwards using `shift-Tab`

$:/config/EditTabIndex

Hidden Setting: Tags Minimum Length

4th December 2017 at 12:31am

Hidden Settings

ⓘ **Introduced in v5.1.16** Tag Dropdown: Controls the minimum length of an input string before results are displayed.

Defaults to "0".

$:/config/Tags/MinLength

Hidden Setting: Template for Save Wiki Button

15th December 2018 at 4:48pm

Hidden Settings

Determines the template used when saving via the "save wiki" page control button.

Defaults to `$:/core/save/all`.

You can change this setting to choose what is saved when the "save wiki" button is clicked.

$:/config/SaveWikiButton/Template

Hidden Setting: Typing Refresh Delay

14th October 2019 at 11:18am

Hidden Settings

TiddlyWiki defers processing changes to draft tiddlers and tiddlers with the prefix `$:/temp/volatile/` until a timeout has elapsed (this is called throttling). The mechanism can be extended to other tiddlers by adding a `throttle.refresh` field. See RefreshThrottling for details.

The default value of 400ms gives a good balance of responsiveness in most cases but isn't always optimal on lower powered mobile devices.

The timeout can now be changed by changing this value (in milliseconds):

$:/config/Drafts/TypingTimeout

Hidden Setting: ViewTemplate and EditTemplate

4th July 2019 at 7:35am

Hidden Settings

The configuration Tiddlers **$:/config/ui/ViewTemplate** and **$:/config/ui/EditTemplate** can be used to change the **ViewTemplate** respectively the **EditTemplate** used in the **Story PageTemplate**

Hidden Settings

19th June 2015 at 6:24pm

Customise TiddlyWiki

The most useful configuration settings for TiddlyWiki are exposed in the **control panel** but some more advanced settings do not have a full user interface. Instead, the underlying configuration tiddlers are edited directly.

To change these settings on an empty TiddlyWiki, you can drag the links to the individual configuration tiddlers directly from this wiki into yours.

- Hidden Setting: Default Tiddler Icon
- Hidden Setting: Disable Drag and Drop
- Hidden Setting: Disable Lazy Loading
- Hidden Setting: Enable File Import in Editor
- Hidden Setting: Filename for Save Wiki Button
- Hidden Setting: HTML Parser Sandbox
- Hidden Setting: More Tabs Horizontal

- Hidden Setting: Navigate on Enter
- Hidden Setting: New-Image Type
- Hidden Setting: Retain Story ordering
- Hidden Setting: Scroll Top Adjustment
- Hidden Setting: Search AutoFocus
- Hidden Setting: Search Minimum Length
- Hidden Setting: Show Edit Preview per Tiddler
- Hidden Setting: Sync Logging
- Hidden Setting: Sync Polling Interval
- Hidden Setting: Sync System Tiddlers From Server
- Hidden Setting: Tab Index for Edit-Inputs
- Hidden Setting: Tags Minimum Length
- Hidden Setting: Template for Save Wiki Button
- Hidden Setting: Typing Refresh Delay
- Hidden Setting: ViewTemplate and EditTemplate
- Hidden Settings: Import Content Types for Editor

Hidden Settings: Import Content Types for Editor

19th May 2021 at 6:02pm

Hidden Settings

ⓘ Introduced in v5.2.0

$:/config/Editor/ImportContentTypesFilter

This filter determines which `contentTypes` can be imported by dragging and dropping into the editor. It used by a DropzoneWidget wrapped around the editor, for the `contentTypesFilter` attribute.

Highlight Plugin

22nd February 2022 at 10:43am

OfficialPlugins Plugin Editions

The Highlight plugin provides the ability to apply syntax colouring to text.

See https://tiddlywiki.com/plugins/tiddlywiki/highlight



The latest version of the Highlight Plugin requires a modern browser that fully supports JavaScript ES6 (released in 2015). The older version is still available as the **highlight-legacy** plugin for users who need to use an older browser.

History of TiddlyWiki

23rd September 2014 at 4:19pm

About

These are personal reflections on the history and development of TiddlyWiki from JeremyRuston.

There is also a podcast from 2016 discussing TiddlyWiki's backstory as well as a recording from 2021.

Origins of TiddlyWiki

Back in 1997 a colleague introduced me to Ward Cunningham's original wiki. I was impressed that something so powerful could fit into just 700 lines of Perl, and fascinated by the radical reimagining of security and permissions. Like many other developers, I took every opportunity I could to try out various wikis, and to explore their use at work.

The allure of the wiki for me was the feeling that it could eventually disrupt the prevailing paradigm of print-oriented documents and emails.

After watching people use wikis for a few years, I noticed that power users made extensive use of the ability to open multiple wiki pages at once in several browser tabs, making it easier for them to compare and review pages, to copy text between them and to act as a sort of queue of pages yet to be read.

I felt that this ability to manipulate multiple pages at once was central to the ability to refactor a wiki, and it is generally accepted that a wiki that is lovingly refactored tends to be more useful. And yet, standard wiki user interfaces have always been designed exclusively for the presentation and manipulation of single pages at once.

All of these thoughts came together when I saw GMail in April 2004, which used Ajax cleverly to blend individual emails into threaded conversations.

I started experimenting with HTML and JavaScript to explore the idea further. I'd had virtually no experience of either, just having put together some static pages and simple ASP sites in previous lives. Getting my head around these client-side technologies was painful; like everyone else, I was horrified to discover how appalling were the incompatibilities and inconsistencies of web programming.

Launch of TiddlyWiki

So, in September 2004 I released a primitive first version of TiddlyWiki. It was the smallest possible thing that demonstrated the idea: it was a simple, self-contained static 48KB HTML file.

The downside of writing the first version of TiddlyWiki in this way was that it made it completely impractical to use for editing - when you click 'save changes' it just pops up a window showing the data that would be saved if it were possible for an HTML page to write to the file system.

Much of the early feedback was that TiddlyWiki was neat, but that it would be more useful when it was possible to properly save changes. I was a little frustrated, as I thought I knew that it was impossible for an HTML file running in the browser to save changes to the local file system.

Within a few months I saw an experimental Firefox extension that enabled TiddlyWiki to save changes in the browser. Examining the code, I realised that the APIs that it used to write to the file system were actually available in ordinary HTML files - as long as they were loaded via a `file://` URI.

I adapted the Firefox code into the core of TiddlyWiki, and soon added a similar ability for Internet Explorer (making use of an old *ActiveX* control that Microsoft distributed with Internet Explorer).

Growth of TiddlyWiki

A major milestone in the growth of TiddlyWiki was the creation of "*GTDTiddlyWiki*" by Nathan Bowers. He took the vanilla TiddlyWiki product and adapted it for the specific application of keeping track of tasks using the popular Getting Things Done methodology. *GTDTiddlyWiki* was an immediate hit, being enthusiastically greeted on websites like *LifeHacker*.

Over the next couple of years TiddlyWiki continued to grow in popularity, and gained new features and capabilities. Within a year I was able to support myself by performing bespoke development work on TiddlyWiki, notably working with wiki pioneer *SocialText* on the ability to synchronise changes with an online server

BT Acquisition

In May 2007, BT acquired Osmosoft, my consultancy company. It was an unusual decision to acquire a company with a single employee and a tiny trickle of revenue - Osmosoft didn't even own the intellectual property in TiddlyWiki since I had handed it over to *UnaMesa* to assure its future for the community.

BT's motivation was to help them understand community-based ecosystems. I joined the organisation as "Head of Open Source Innovation", taking responsibility for open source governance, and providing advice and expertise on how to participate in open soure communities.

Osmosoft and TiddlySpace

I built a team in BT under the name Osmosoft. Our purpose was to evangelise the benefits of open source, and to help other teams realise those benefits in practice. We also found that it was necessary to evangelise the use of the web in general, and web standards in particular.

Our approach was to focus on showing rather than telling. We worked with the TiddlyWiki community to extend the ecosystem and we built numerous internal systems for BT (some based on TiddlyWiki and some not).

Osmosoft's chief contribution to the TiddlyWiki community was the creation of TiddlyWeb and TiddlySpace. TiddlyWeb was a robust, internet scale server for tiddlers that could also compose TiddlyWiki views of those tiddlers. TiddlySpace was an attempt to package TiddlyWeb into a more directly usable form.

Leaving BT

By the end of 2011 I was increasingly feeling that I would be better placed to realise the potential of TiddlyWiki outside of the corporate confines BT. Accordingly, I left and started work as an independent developer, primarily working on a brand new reboot of TiddlyWiki in the shape of TiddlyWiki5.

Development of TiddlyWiki5

I worked on new release of TiddlyWiki from November 2011. As a programmer, working on "version 2.0" of something that I had already written is a very attractive proposition. It means that the requirements were fully understood, allowing me to focus on evolving the architecture needed to support the desired functionality.

The Future

Now that TiddlyWiki5 has finally left "beta" status behind, my hope is that it will have a long life. Because it only uses standard features of HTML5 and Node.js, there is no reason why it cannot be fully operational for many years to come. My goal is for it to last for at least 25 years.

Jeremy Ruston, 20th September 2014

HistoryMechanism

7th February 2015 at 2:10pm

Mechanisms

The system tiddler $:/HistoryList keeps track of a list of tiddlers comprising the navigation history. Each time you click on a link to a tiddler, the title of the target tiddler is added to the top of the stack.

The history list is stored in JSON to allow additional details about the coordinates of the DOM node that initiated the navigation.

The history list also maintains the field **current-tiddler** that contains the name of the tiddler at the top of the stack. This field can be used like so:

```
<$list filter="[list[$:/StoryList]]" history="$:/HistoryList" storyview="pop">

<$button message="tm-close-tiddler" class="tc-btn-invisible tc-btn-mini">&times;</$button> <$link><$view
field="title"/> <$reveal type="match" state="$:/HistoryList!!current-tiddler"
text=<<currentTiddler>>>&#x2713;</$reveal></$link>

</$list>
```

Which renders the same as the "Open" sidebar tab, with the addition of a tick against the tiddler that was last navigated to.

× Learning ✓

× HelloThere

× GettingStarted

× Community

Empty Story

To display content when the story is empty, create *$:/config/EmptyStoryMessage* and enter the desired contents. The following would show the GettingStarted tiddler when all others are closed.

```
{{GettingStarted||$:/core/ui/ViewTemplate}}
```

Horizontal Rules in WikiText

13th May 2022 at 1:12pm

WikiText

You can include a horizontal rule with three or more dashes on their own on a line:

copy to clipboard

```
---
```

That renders as:

... and the underlying HTML is:

```
<hr>
```

How to add a banner for GitHub contributions

23rd November 2014 at 10:44am

Learning

When you edit a tiddler on https://tiddlywiki.com you will see a small ribbon inviting you to edit the source of the tiddler on GitHub.

If you are using Node.js, you can replicate this feature for your own TiddlyWiki-based site as follows:

1. Make sure the following setting is included in the `tiddlywiki.info` file in your wiki folder:

```
"config": {
    "retain-original-tiddler-path": true
}
```

2. Copy the tiddler $:/ContributionBanner to your wiki
3. Make the following changes:
    1. Adjust the GitHub link URL from https://github.com/Jermolene/TiddlyWiki5/edit/master/editions/tw5.com/tiddlers/ to point to your own GitHub repo
    2. Make sure the wording of the text starting "Can you help us improve this documentation?" is appropriate for your visitors
    3. Adjust the link to Improving TiddlyWiki Documentation to point to your own tiddler with instructions for the contribution procedure

How to add a new tab to the sidebar

4th February 2017 at 8:12pm

Customise TiddlyWiki Learning

To create a new tab in the sidebar menu:

1. Create a tiddler and tag it with the SystemTag tag **$:/tags/SideBar**
    - To create a table of contents you can populate the new tab tiddler using the table-of-contents macros. For example, see the TableOfContents used here
2. By default, the tiddler title is used as the tab title but you can override it using the `caption` field

3. To define the tab ordering, use the `list-after` or `list-before` fields as discussed in Tagging
    - For example: set `list-after` to **$:/core/ui/SideBar/Open** to place a sidebar tab immediately after the "Open" tab

Note that you can create new tabs under the "More" tab in the same way by using the tag `$:/tags/MoreSideBar`.

How to apply custom styles

11th April 2018 at 7:40pm

Customise TiddlyWiki

There are several ways to apply custom styles.

- Custom styles by data-tags
- Custom styles by data-tiddler-title
- Custom data-styles
- Custom styles by user-class
- Custom Styles FAQ
- Custom tag pill styles
- How to apply custom styles by tag - deprecated
- Simple ways to write protect tiddlers
- Styles and Classes

How to apply custom styles by tag

4th July 2022 at 7:42pm

$:/deprecated How to apply custom styles

DEPRECATED SINCE: 5.1.16 (see Custom styles by data-tags). Also see: How to apply custom styles

---

You can apply custom styles to tiddlers that have a particular tag by defining a CSS class with the name `tc-tagged-<Tag Name>`.

For example, to make tiddlers tagged "*NightReader*" appear in a special colour scheme suitable for night-time reading, create a stylesheet defining the class `tc-tagged-NightReader` like this:

```
.tc-tagged-NightReader {
    background-color:black;
    color: orange;
    padding: 35px 35px;
}

.tc-tagged-NightReader .tc-tiddler-body {
    font-size: 1.5em;
}
```

The `tc-tagged-NightReader` class is applied to the entire tiddler and not just the tiddler text. If you want to target a smaller portion of the tiddler you can qualify the CSS selector, as is done here with `.tc-tagged-NightReader .tc-tiddler-body`.

Note that tags containing spaces or non-alphanumeric characters will be converted using URI encoding, making the generated CSS classname hard to predict. For example:

Tag	Generated Class Name

`$:/mytag`	`tc-tagged-%24%3A%2Fmytag`
`Doctor Who`	`tc-tagged-Doctor%20Who`
`£35.23`	`tc-tagged-%C2%A335.23`

Although TiddlyWiki will generate these tags, to actually use them in your css, you will need to escape the percent character in your stylesheet, like:

```
.tc-tagged-Doctor\%20Who {
    background-image: url(./tardis_back.svg);
    background-repeat: no-repeat;
    background-position: right;
    color:#FBFBFB;
}
```

A utility function is available in JavaScript to perform the conversion:

```
$tw.utils.tagToCssSelector("$:/tags/Stylesheet")
```

Generates the following output:

```
tc-tagged-\%24\%3A\%2Ftags\%2FStylesheet
```

How to build a TiddlyWiki5 from individual tiddlers

12th September 2014 at 4:16pm

TiddlyWiki on Node.js

First install TiddlyWiki as described in Installing TiddlyWiki on Node.js.

1. Create an empty TiddlyWikiFolder
    1. Create a new folder in a convenient place (for example `~/MyWiki`)
    2. Create a file called `tiddlywiki.info` containing the following text:
        - `{"themes": ["tiddlywiki/vanilla","tiddlywiki/snowwhite"]}`
    3. Create a subfolder called `tiddlers`
        - Alternatively, just copy the `editions/empty` folder from the TiddlyWiki5 repo
2. Create individual TiddlerFiles in the `~/MyWiki/tiddlers` directory
3. Execute the following command from the TiddlyWiki5 root directory to build a TiddlyWiki5 file from the tiddlers:
    1. `tiddlywiki ~/MyWiki --rendertiddler $:/core/save/all index.html text/plain`

How to change the sort order of sub-branches in a TOC macro

20th February 2019 at 1:00am

Learning

Imagine that you are using a Table of Contents macro similar to this:

```
<<toc-selective-expandable  "TableOfContents" "sort[title]">>
```

The sorting is fine for most cases, but you would like all your items tagged `Journal` to be sorted by the `created` field. How can you apply a separate sort order to just those sub-items tagged `Journal`?

The trick is to add a field to the parent tagging tiddler (i.e. `Journal`) that points to a different sort criteria. Let's call the field `fuzzy`, and populate it with the value `created` (`created` is the name of the field that contains a tiddler's creation date).

Now change your Table of Contents to look like this:

```
<<toc-selective-expandable  "TableOfContents" "sort{!!fuzzy}">>
```

Now your Table of Contents will sort by title everywhere, except for the children of the `Journal` tiddler, which will sort by the `created` date.

The downside to this trick is that you can't apply a reverse sort based on just one special field. If you reverse the sort order, it will be reversed everywhere.

How to create a custom story tiddler template

6th December 2021 at 5:51pm

Cascades Customise TiddlyWiki Learning Story Tiddler Template Story Tiddler Template Cascade

Here we show how to add a rule to the Story Tiddler Template Cascade that causes tiddlers tagged $:/tags/TiddlerList to be displayed with a custom appearance incorporating thumbnails of the tiddlers named in the list field.

Open Demo Tiddler List with Custom Story Tiddler Template to see the demo.

First, we add a new step to the story tiddler template cascade. The new filter step is:

```
[tag[$:/tags/TiddlerList]then[$:/_tw5.com/CustomStoryTiddlerTemplateDemo/Template]]
```

It can be read as:

```
If the tiddler has the tag $:/tags/TiddlerList then return $:/_tw5.com/CustomStoryTiddlerTemplateDemo/
Template
```

> The template $:/_tw5.com/CustomStoryTiddlerTemplateDemo/Template also depends on styles defined in $:/_tw5.com/CustomStoryTiddlerTemplateDemo/Styles.

The story tiddler template cascade is defined by the tag $:/tags/StoryTiddlerTemplateFilter , so we need to create a configuration tiddler with that tag.

We also need to make sure that it is inserted at the correct point in the cascade:

- The new rule must be after the "draft" rule to ensure that draft tiddlers with the tag *$:/tags/TiddlerList* can be edited
- The new rule must be before the "default" rule otherwise it will never be executed

The control panel "Cascades" tab shows that this can be achieved by inserting the new step immediately before the default step using the `list-before` field.

The full list of fields in the configuration tiddler is:

$:/_tw5.com/CustomStoryTiddlerTemplateDemo/Filter

code-body	yes
list-before	$:/config/StoryTiddlerTemplateFilters/default
tags	$:/tags/StoryTiddlerTemplateFilter
text	[tag[$:/tags/TiddlerList]then[$:/_tw5.com/CustomStoryTiddlerTemplateDemo/Template]]

Finally, we need to create the demo tiddler itself:

Demo Tiddler List with Custom Story Tiddler Template

created	20211204122107920
filter	HelloThere Community GettingStarted Features Reference Plugins Learning
modified	20211204131610322
tags	$:/tags/TiddlerList
text	This is a demo tiddler with a custom story tiddler template that displays the tiddlers named in the list field as a fan. See [[How to create a custom story tiddler template]] for details.
type	text/vnd.tiddlywiki

How to create a custom tiddler colour rule

6th December 2021 at 5:51pm

Cascades Customise TiddlyWiki Learning

Here we show how to add a rule to the Tiddler Colour Cascade that causes tiddlers with the tag TableOfContents to be given a bright blue colour and tiddlers with the tag Working with TiddlyWiki to be given a dark purple colour.



How to create a custom tiddler icon rule describes how the same tiddlers have been given a custom icon

First, we add a new step to the tiddler colour cascade. The new filter step is:

```
[tag[TableOfContents]then[#1e90ff]]
[tag[Working with TiddlyWiki]then[darkorchid]]
```

It can be read as:

```
If the tiddler has the tag "TableOfContents" then return the colour #1e90ff
Else if the tiddler has the tag "Working with TiddlyWiki" then return the colour darkorchid
```

The tiddler colour cascade is defined by the tag `$:/tags/TiddlerColourFilter`, so we need to create a configuration tiddler with that tag.

We also need to make sure that it is inserted at the correct point in the cascade:

- The new rule must be after the "colour-field" rule to ensure that tiddlers with an explicit **color** field will respect the value of that field
- The new rule must be before the "default" icon otherwise it will never be seen if the default colour is specified

The control panel "Cascades" tab shows that this can be achieved by inserting the new step immediately before the default step using the `list-before` field.

The full list of fields in the configuration tiddler is:

$:/_tw5.com/CustomTiddlerColourCascadeDemo

created	20211206114433294
list-before	$:/config/TiddlerColourFilters/default
modified	20211206114433294
tags	$:/tags/TiddlerColourFilter
text	[tag[TableOfContents]then[#1e90ff]] [tag[Working with TiddlyWiki]then[darkorchid]]
type	text/vnd.tiddlywiki

How to create a custom tiddler icon rule

6th December 2021 at 5:51pm

Cascades Customise TiddlyWiki Learning Tiddler Icon Cascade

Here we show how to add a rule to the Tiddler Icon Cascade that causes tiddlers with the tag TableOfContents to be given the "globe" icon and tiddlers with the tag Working with TiddlyWiki to be given the "help" icon .



How to create a custom tiddler colour rule describes how the same tiddlers have been given a custom colour

First, we add a new step to the tiddler icon cascade. The new filter step is:

```
[tag[TableOfContents]then[$:/core/images/globe]]
[tag[Working with TiddlyWiki]then[$:/core/images/help]]
```

It can be read as:

```
If the tiddler has the tag "TableOfContents" then return $:/core/images/globe
Else if the tiddler has the tag "Working with TiddlyWiki" then return $:/core/images/help
```

The tiddler icon cascade is defined by the tag $:/tags/TiddlerIconFilter , so we need to create a configuration tiddler with that tag.

We also need to make sure that it is inserted at the correct point in the cascade:

- The new rule must be after the "icon-field" rule to ensure that tiddlers with an explicit **icon** field will respect the value of that field
- The new rule must be before the "default" icon otherwise it will never be seen if the default icon is specified

The control panel "Cascades" tab shows that this can be achieved by inserting the new step immediately before the default step using the `list-before` field.

The full list of fields in the configuration tiddler is:

$:/_tw5.com/CustomTiddlerIconCascadeDemo

created	20211205195110592
list-before	$:/config/TiddlerIconFilters/default
modified	20211205195217941
tags	$:/tags/TiddlerIconFilter
text	[tag[TableOfContents]then[$:/core/images/globe]] [tag[Working with TiddlyWiki]then[$:/core/images/help]]
type	text/vnd.tiddlywiki

How to create dynamic editor toolbar buttons

18th October 2021 at 12:23pm

Create a New Toolbar Button

The easiest way to create new editor toolbar button is to clone and open one.

> Create a Temporary Bold Button

This tiddler contains all the necessary elements that are important for toolbar buttons.

text
: We don't discuss the text field details in this howto

caption
: The caption field is used to display the shortcut name in the **$:/ControlPanel** : Keyboard Shortcuts tab

condition
: A filter, that defines the button visibility state

condition-disabled ⓘ **Introduced in v5.1.23**
: A **filter**, that allows us to define the "disabled" attribute for buttons. eg: `[[$:/temp/bold/disabled]get[state-disabled]else[no]]`
This condition **must** return "no", if the "state tiddler" or "state field" doesn't exist! So there has to be a `else[no]` element in the filter.

description
: Is used as the button tooltip

icon
: Assigns the button icon. We use `$:/core/images/spiral`  here. More button icons can be found in the Icon Gallery

shortcuts
: This is the Keyboard Shortcut Descriptor eg: `((temp-bold))`

Disabled State

You can use any "state tiddler" to define the button disabled state. It's important, that the **condition-disabled** field is a **filter**. For our example we use the `state-disabled` field from tiddler: *$:/temp/bold/disabled*

The easiest way to test filters is with the **$:/AdvancedSearch** : Filter tab

> The temporary "bold button" is disabled: "*no*" ... You can see the button in the editor toolbar if you edit any tiddler!
>
> Toggle Temporary Bold Button Visisbility

Create Keyboard Shortcuts for New Button

For our HowTo we use the **Keyboard Shortcut Descriptor**: `((temp-bold))`

To create a valid shortcut configuration we need 2 tiddlers:

> *$:/config/ShortcutInfo/***temp-bold** .. and
>
> *$:/config/shortcuts/***temp-bold**
>
> Create Shortcut Tiddlers

Clean up the Configuration Tiddlers

If you don't need the config tiddlers anymore you can

> Remove the Shortcut Configuration

How to create keyboard shortcuts

Customise TiddlyWiki Learning



TiddlyWiki distinguishes two types of keyboard shortcuts. Those that are limited to work within **`input`** fields or **`textareas`** which are handled by the **`$keyboard`** widget and keyboard shortcuts that work globally - which means that only the page needs to have focus in order for them to be accessible. The latter are handled by a mechanism that starts to work when a wiki is loaded

The configuration mechanism is the same for both shortcut-types

A tiddler with the **prefix** `$:/config/ShortcutInfo/` and a **unique suffix** appended, like `my-shortcut`, makes the new shortcut appear in the **$:/ControlPanel** within the **Keyboard Shortcuts Tab**. An optional description can be added within its text field

In the **Keyboard Shortcuts Tab** the **key combination** that should trigger the shortcut can be configured:

> Look for the **unique suffix** defined for the new shortcut and click the ✎ button to open a popup that detects **key combinations** and shows the detected combination in its input field
>
> The **add shortcut** assigns the key-combination to the shortcut

Depending on the chosen platform the configuration mechanism creates a tiddler with a specific nomenclature

> $:/config/shortcuts/ + the **shortcut suffix** for shortcuts that will work on all platforms (operating systems)
>
> $:/config/shortcuts-(mac/not-mac/windows/not-windows/linux/not-linux)/ + the **shortcut suffix** for shortcuts that will work on the chosen platform only or that will not work on the excluded platform

Examples

- $:/config/shortcuts/`my-shortcut`
- $:/config/shortcuts-mac/`my-shortcut`
- $:/config/shortcuts-windows/`my-shortcut`
- $:/config/shortcuts-linux/`my-shortcut`
- $:/config/shortcuts-not-mac/`my-shortcut`
- $:/config/shortcuts-not-windows/`my-shortcut`
- $:/config/shortcuts-not-linux/`my-shortcut`



Note that those platform-specific tiddlers don't have to be created manually if the **ShortcutInfo** tiddler as mentioned above is created first and the shortcut is configured in the ControlPanel

At this point the shortcut is defined and its **actions** can be configured

Using the Keyboard Widget

> A **`$keyboard`** widget detects the key combinations pressed within an **`input`** or **`textarea`** field within its scope. The **`$keyboard`** widget needs to "embrace" the **`input`** or **`textarea`** field and its **`key`** attribute either points to a configuration tiddler using the Key Descriptor Syntax or explicitely contains the key

```
<$keyboard key="((my-shortcut))" actions='<$action-setfield $tiddler="my-tiddler" $field="my-field"
$value="my-value"/>'>
```

```
<$edit-text tag="input" tiddler="my-tiddler"/>

</$keyboard>
```

```
<$keyboard key="alt-shift-M" actions='<$action-setfield $tiddler="my-tiddler" $field="my-field"
$value="my-value"/>'>

<$edit-text tag="input" tiddler="my-tiddler"/>

</$keyboard>
```

> If actions are defined , the **`$keyboard`** widget will invoke the actions defined in its **`actions`** attribute when it detects the key-combination defined in its **`key`** attribute being pressed in any **`input`** or **`textarea`** field within its scope

Using global Keyboard Shortcuts

> See Keyboard Shortcut Tiddler for detailed information about creating new global keyboard shortcuts.
>
> The actions for **global** keyboard shortcuts are stored in the **text** field of tiddlers tagged with $:/tags/KeyboardShortcut
>
> The **key field** connects an action-tiddler with the corresponding shortcut through the `((my-shortcut))` syntax, called Keyboard Shortcut Descriptor

Syntax: `((` + `a chosen suffix` + `))`

> If the tiddler has the tag $:/tags/KeyboardShortcut , the field **key** with the Keyboard Shortcut Descriptor as its value and some actions in its text field, the actions will be triggered when the mechanism detects the configured key-combination

> **Macros** defined **outside** a global keyboard-shortcut (through a tiddler tagged `$:/tags/Global`) need to be **imported** in order to be accessible. The import pragma can be used for that

```
\import [subfilter{$:/core/config/GlobalImportFilter}]
```

If the tiddler that contains the macro definition is known and - for example - titled `my-macro-tiddler`

```
\import my-macro-tiddler
```

> Some actions require to be wrapped within a **`$navigator`** widget. That is necessary for the following list of widgets and messages in order to work if used within global keyboard shortcuts

Widgets

`$action-navigate` (ActionNavigateWidget)

Messages

WidgetMessage: tm-cancel-tiddler

WidgetMessage: tm-close-all-tiddlers
WidgetMessage: tm-close-other-tiddlers
WidgetMessage: tm-close-tiddler
WidgetMessage: tm-delete-tiddler
WidgetMessage: tm-edit-tiddler
WidgetMessage: tm-import-tiddlers
WidgetMessage: tm-new-tiddler
WidgetMessage: tm-perform-import
WidgetMessage: tm-relink-tiddler
WidgetMessage: tm-rename-tiddler
WidgetMessage: tm-save-tiddler

How to Customize TiddlyDesktop

18th November 2017 at 8:45pm

Learning

Prelude/caution:

Before experimenting with TiddlyDesktop internals you should find the location of the “user configuration folder” - click the “Settings” button on the main TiddlyDesktop window, and then click the button “Open user config folder”. Then move up to locate the parent folder, which should be called “TiddlyDesktop”. If things go wrong, you can reset TiddlyDesktop to its factory state by deleting this entire folder.

Click the `Backstage` button on the main TiddlyDesktop window to open the backstage wiki. This is a Node.js-like instance of TiddlyWiki that runs the UI and logic of TiddlyDesktop.

How do I get rid of the "Add a TiddlyWiki Folder" button.

Within the backstage wiki, locate the tiddler `WikiListToolbar` (there's a link in HelloThere). Click it into edit mode and you'll see the wikitext for all the toolbar buttons. If you make modifications and click "done" then the changes will immediately be reflected in the main TiddlyDesktop window.

How would I change the "advanced" button to the "reveal backups" button.

You can copy the code for the "reveal backups" button from the tiddler `$:/TiddlyDesktop/Settings` and paste it into `WikiListToolbar`.

How do I change the colors?

Just as usual in TiddlyWiki: you can use the palette, or define your own custom CSS.

How do I change the order in which the tiddlers are listed (which appears to be alphabetical)?

Open the `WikiList` tiddler, edit it, and scroll to the end. Then change the `<$list>` widget inside the **wikilist** macro:

```
<$list filter="[tag[wikilist]sort[title]!has[draft.of]]" emptyMessage="Add a ~TiddlyWiki file or folder to get started. Click the buttons above to browse, or drag and drop from your file Explorer/Finder" storyview="pop">
```

How to embed PDF and other documents

29th December 2016 at 6:57pm

Learning

Note about embedding versus external linking

Note: Before we dive into how you can embed files of a certain type, you may want to consider not embedding and instead simply linking to external resources like this:

```
[ext[title|path/to/that.file]]
```

Then let your browser decide how to handle the resource. The advantage to this approach is that it is simple and quick. The disadvantage is that the file doesn't actually travel with the TW5 file and that you can't use `node.js` with method #2 below to quickly file paths.

Two ways to embed a PDF in your TW5 file.

1. Embedding with Drag and Drop

One is to drag and drop the file onto your TW5 file. Wait for the green bar to appear at the top, drop, and then import from the import tiddler. This actually embeds the contents of your imported PDF (or other file) into your TW5 file.

This method be OK as long as your PDF is not too big. There can be concerns if your TW5 file becomes too big.

2. Embedding with '_canonical_uri'

The other way is to create a tiddler link to the external file. In this method the file is not actually incorporated into your TW5 file, but can be accessed with the `{{My Image File.jpg}}` transclusion syntax just like an embedded file. The location address of the file can also be changed under *node.js*. See ExternalImages for details of using external images with node.js.

Create a tiddler with a field `_canonical_uri`. Put in the local address to the external file. Set the `type` field to `application/pdf`.

With either way, with an appropriately configured browser, you should see your embedded PDF file.

Other File Types

There's only so many file formats that TW5 knows about. There is a list at ContentType. If your browser and/or TW5 doesn't know what to do with a particular file format, then you can only the use the second method to embed and then download a file. When you edit the tiddler a link is presented. You can right click on the link in order to download the file.

How to export tiddlers

10th June 2016 at 10:40am

Learning

Selecting All tiddlers for export

To export **all** tiddlers click on the "Tools" tab of the Sidebar. Find and click the **export all** button. A pop-up menu will offer you the chance to export tiddlers in multiple formats.

Exporting a tiddler

If you want to export a particular tiddler, first navigate to that tiddler. Then click on the **more** button that appears above the tiddler. From the list that appears select **export tiddler**. A pop-up menu will offer you the chance to export the tiddler in multiple formats.

Exporting tiddlers matching a criteria (filter)

To export a selection of tiddlers, click the little magnifying glass next to the search area on the sidebar. This will open the **Advanced Search** tiddler. Click on the "Filter" tab of the Advanced Search tiddler. Only the Filter tab will allow you to export a selection of tiddlers. Filters follow a particular syntax. Click on this Filters link to learn about how to make filters.

Once you have written a filter, a list of tiddlers matching the filter will appear. Now you can click on the **export tiddler** button to the right of the filter input field. A pop-up menu will offer you the chance to export tiddlers in multiple formats.

Export formats

Currently, the export formats available are:

- CSV file format
- JSON file format
- Static HTML file
- ".tid" file format

The `".tid"` format is unique from the other formats in that only one tiddler will be exported, even if you are attempting to export all tiddlers or a selection of tiddlers.

How to hide the author's and other fields with CSS

10th March 2018 at 9:12pm

Learning

Sometimes you might want to save some screen space by hiding away the author's name in all tiddlers. Here's a quick way to do it:

- Create a new tiddler with any title you want.
- Give the tiddler the tag `$:/tags/Stylesheet`
- In the text field of the tiddler put:

```
.tc-subtitle .tc-tiddlylink {display:none;}
```

- Save the tiddler. The author's name field should no longer appear.

Similarly, the entire subtitle field including author and date can be removed with:

```
.tc-subtitle {display:none;}
```

Unfortunately, you can't hide just the date without also hiding the author using CSS.

And finally, for a truly minimalist look, you can remove the title with:

```
h2.tc-title {display:none;)
```

How to put the last modification date in a banner

17th June 2016 at 2:59pm

Learning

Here's how to display the last modification date of a wiki in a banner in the corner of the window:

1. Install the plugin **Corner ribbon** in your TiddlyWiki
2. Save (⊘) and reload (↺) your wiki
3. Create a new tiddler called *$:/_MyRibbon* tagged **$:/tags/PageControls** and containing:

```
<div class="github-fork-ribbon-wrapper right">
<div class="github-fork-ribbon" style="background-color:#DF4848;">
<$list filter="[!is[system]!has[draft.of]!sort[modified]limit[1]]">
<$link>
<$view field="modified" format="date" template="DD mmm YYYY at 0hh:0mm"/>
</$link>
</$list>
</div>
</div>
```

4. If required, change the background-color value to your preference
5. You can also try changing the positioning class from `right` to `right-bottom`
    1. To make the banner appear at the top left you'll need to tag the tiddler **$:/tags/PageTemplate** instead of **$:/tags/PageControls** and then change the position class to `left`

How to remove stop words

18th November 2020 at 11:34am

Operator Examples search-replace Operator sortsub Operator

You can use the search-replace Operator in combination with the sortsub Operator to ignore stop words like "A" and "The" at the beginning of titles when sorting. (Note for example that "A Tale of Two Cities" sorts under "T" rather than "A")

```
[enlist<book-list>sortsub<compare-without-stopwords>]
```

Try it

You can also customise the listing so that "A Tale of Two Cities" is listed as "Tale of Two Cities, A" but still links to the correct tiddler.

copy to clipboard

```
<ul>
<$list filter="[enlist<book-list>sortsub<compare-without-stopwords>]">
<li><$link><$text text={{{[<currentTiddler>search-replace:i:regexp[(^The |A )(.*)],[$2, $1]]}}}/></
$link></li>
</$list>
</ul>
```

That renders as:

- *Animal Farm*
- *Dice Man, The*
- *Great Expectations*
- *New Kind of Science, A*
- *Recursive Universe, The*

- *Tale of Two Cities, A*

How to turn off camel case linking

4th February 2017 at 8:13pm

Customise TiddlyWiki Learning

CamelCase is used in most Wikis, but it can interfere with text presentation in some situations.

Turning off Camel Case locally

To turn off a particular instance of a CamelCase text, simply put a tilde (~) in front of the word, like this:

```
~CamelCase
```

Turning off Camel Case globally

To turn off CamelCase in all tiddlers, navigate to the **Control Panel**. Select the `Settings` tab and then scroll or read down to the `Camel Case Wiki Links` section and unselect `Enable automatic CamelCase linking`.

How to update TiddlyWiki to the latest version

27th April 2022 at 12:12am

Working with TiddlyWiki

There are regular releases of TiddlyWiki with bug fixes and improvements. It's a good idea to keep up to date by regularly upgrading to the latest version.

Introduction

The process described here is for upgrading standalone TiddlyWiki files. There is a different procedure for upgrading TiddlyWiki on Node.js.

> When upgrading, please remember the The First Rule of Using TiddlyWiki:
>
> *You are responsible for looking after your own data; take care to make backups, especially when upgrading the TiddlyWiki core*

Online upgrading

This process will work on most desktop browsers. Note that none of your personal data leaves your browser with this process.

1. Locate your TiddlyWiki file in the file system (i.e. using Windows Explorer, the Finder on Mac OS X, or your file manager on Linux)
2. Visit https://tiddlywiki.com/upgrade.html in your browser
3. Drag your old TiddlyWiki HTML file into the browser window
    - If the file is encrypted you will be prompted for the password
4. Review the list of tiddlers that will be upgraded
5. Click **Upgrade**
6. Save changes to save the new version (⊘)

This will download a file called **upgrade.html** to your computer. This file is the upgrade of your old file. You may need to open the location where

upgrade.html was downloaded, rename **upgrade.html** with the name of the old file you are upgrading, and replace the old file by moving the new file in its place.

Offline upgrading

You can also download https://tiddlywiki.com/upgrade.html locally and perform the same drag-and-drop procedure to upgrade your files.

Problems with Upgrades

Firefox Security Restrictions

The following error occurs when trying to perform the online upgrade procedure using Firefox:

> Error while saving:
>
> Error:NS_ERROR_DOM_BAD_URI: Access to restricted URI denied

The upgrade operation falls foul of a security restriction in Firefox. Until this can be resolved, we suggest either using the offline upgrader, or using Chrome to perform the upgrade:

1. Use Chrome to open https://tiddlywiki.com/upgrade.html, then drag the TiddlyWiki HTML file to be upgraded into the upgrader window, as described above in **Online upgrading**
2. After you've saved your upgraded file, you should be able to open it in Firefox and save using TiddlyFox again

Incompatible Customisations

It is possible for a customisation applied in a previous version to break when upgraded to the latest version. There are two techniques you can use to help track down issues:

- Try repeating the upgrade having selectively unchecked any tiddlers that may be applying customisations to TiddlyWiki
- Use SafeMode to disable all customisations of shadow tiddlers

You can see which shadow tiddlers have been overridden in the **Filter** tab of **advanced search**. Select "Overridden shadow tiddlers" from the dropdown.

How to widen tiddlers (aka storyriver)

8th December 2016 at 1:00am

Customise TiddlyWiki Learning

You can change the width of the main column of tiddlers (also known as the story river) by:

- Clicking on the settings cog to open the control panel
- In the control panel select "Appearance" and then "Theme Tweaks".
- Under Theme tweaks, scroll down.

There you will see settings for `story right`, `story width`, and `tiddler width`. By setting these numbers to something larger, you can widen the tiddlers. In general, `story right` and `story width` should be the same size, and `tiddler width` a little bit smaller. You could even use percentages for tiddler width. You will also probably want to change the sidebar breakpoint to something larger than the story river, unless you don't mind the sidebar floating to the top of the page.

HTML Block Elements

18th August 2022 at 11:21am

Definitions

> HTML (Hypertext Markup Language) elements historically were categorized as either "block-level" elements or "inline-level" elements. Since this is a presentational characteristic it is nowadays specified by CSS in the Flow Layout. A Block-level element occupies the entire horizontal space of its parent element (container), and vertical space equal to the height of its contents, thereby creating a "block".
>
> *https://developer.mozilla.org/en-US/docs/Web/HTML/Block-level_elements*

HTML Entities

25th October 2022 at 2:46pm

WikiText

Summary

Use HTML entities to enter characters that cannot easily be typed on an ordinary keyboard. They take the form of an ampersand (`&`), an identifying string, and a terminating semi-colon (`;`), e.g. `&` for the **&** character.

Markup

```
The value of Tiddlers&trade; cannot even be expressed in &pound;, &euro; or &dollar;.
```

Displays as:

The value of Tiddlers™ cannot even be expressed in £, € or $.

Entity References

Comprehensive lists of html entities can be found at...

- Mozilla Developer Network -- Entities
- HTML Spec official list -- Entities

Examples Of Common And Useful Entities

HTML Entities					
		no-break space	'	'	single quote, apostrophe
–	–	en dash	"	"	quotation mark
—	—	em dash	′	′	prime; minutes; feet
…	…	horizontal ellipsis	″	″	double prime; seconds; inches
©	©	Copyright symbol	‘	‘	left single quote
®	®	Registered symbol	’	’	right single quote
™	™	Trademark symbol	“	“	left double quote
†	†	dagger	”	”	right double quote
‡	‡	double dagger	«	«	left angle quote
¶	¶	paragraph sign	»	»	right angle quote
§	§	section sign	×	×	multiplication symbol

↑	↑	up arrow	↓	↓	down arrow
←	←	left arrow	→	→	right arrow
⇐	⇐	double left arrow	⇒	⇒	double right arrow
↔	↔	left right arrow	⇔	⇔	double left right arrow

Accented Characters

The table below shows how accented characters can be built up by *substituting* the underscore (_) in the second table into the corresponding character. eg:

Code	Character	Example	Result
`Ä`	Ä	`Äpfel`	Äpfel

Accented Characters																
grave accent	&_grave;	À	à	È	è	Ì	ì	Ò	ò	Ù	ù					
acute accent	&_acute;	Á	á	É	é	Í	í	Ó	ó	Ú	ú			Ý	ý	
circumflex accent	&_circ;	Â	â	Ê	ê	Î	î	Ô	ô	Û	û					
umlaut mark	&_uml;	Ä	ä	Ë	ë	Ï	ï	Ö	ö	Ü	ü			Ÿ	ÿ	
tilde	&_tilde;	Ã	ã					Õ	õ			Ñ	ñ			
ring	&_ring;	Å	å													
slash	&_slash;							Ø	ø							
cedilla	&_cedil;														Ç	ç

HTML in WikiText

15th June 2023 at 8:01am

WikiText

HTML tags and comments

HTML tags and comments can be used directly in WikiText. For example:

```
<article class="hello">
This is my nice and simple block of text. HelloThere
<!-- This comment will not appear in the wikified output -->
</article>
```

Pragma Comments

Introduced in v5.2.0 Comments can now be freely intermixed with pragmas as well as within the main body of a block of wikitext.

```
<!-- NEW: Comment that describes the macro -->
\define test()
some text <!-- inline comment -->
\end

<<test>>
```

Important

Security



Note that any HTML attributes prefixed with `on` are removed from the rendered HTML content. This is done to prevent event handlers (such as "onclick") being used as a clandestine way to execute untrusted JavaScript. A design goal of TiddlyWiki is to ensure that executable JavaScript can only enter the system through explicit JavaScript module tiddlers or raw markup tiddlers. This makes it possible to filter unsafe content in multiuser environments, and also makes it safer to copy untrusted wikitext examples.

Widgets



Widgets share the same syntax as HTML tags
and so the following information applies to them, too.

Block mode versus Inline mode

To get the content of an HTML element to be parsed in block mode, the opening tag must be followed by two linebreaks.

Without the two linebreaks, the tag content will be parsed in inline mode which means that block mode formatting such as wikitext tables, lists and headings is not recognised.

See also WikiText parser mode: HTML examples and WikiText parser mode transitions.

Self closing elements

The following tags are treated as 'void'. This means that `<tag>` is treated as if it were `<tag/>`, and that no terminating `</tag>` is needed (if one is provided it will be ignored and treated as plain text).

- `<area>`, `<base>`, `<br>`, `<col>`, `<command>`, `<embed>`, `<hr>`, `<img>`, `<input>`, `<keygen>`, `<link>`, `<meta>`, `<param>`, `<source>`, `<track>`, `<wbr>`

If you do not close any other tag then it will behave as if the missing closing tag were at the end of the tiddler.

Attributes

In an extension of conventional HTML syntax, attributes of elements and widgets can be specified in several different ways:

- a literal string
- a transclusion of a textReference
- a transclusion of a macro/variable
- as the result of a filter expression
- ⓘ **New in v5.3.0** as the result of performing filter and variable substitutions on the given string

Style Attributes

ⓘ **Introduced in v5.2.2** TiddlyWiki supports the usual HTML **style** attribute for assigning CSS styles to elements:

```
<div style="color:red;">Hello</div>
```

In an extension to HTML, TiddlyWiki also supports accessing individual CSS styles as independent attributes. For example:

```
<div style.color="red">Hello</div>
```

The advantage of this syntax is that it simplifies assigning computed values to CSS styles. For example:

```
<div style.color={{!!color}}>Hello</div>
```

HTML Links in WikiText

21st October 2016 at 12:20pm

HTML in WikiText

It is often useful to be able to create HTML links to external resources. For example, here the value of the `href` attribute will be set to the value of the tiddler *MyLinkDestination*:

```
<a href={{MyLinkDestination}}>link</a>
```

However, there is an unexpected security issue that means that most of the time the link should have the `rel` attribute set to `noopener noreferrer` to maintain privacy of the URLs of private TiddlyWiki's (eg on Dropbox). See https://mathiasbynens.github.io/rel-noopener/ for more information.

```
<a href={{MyLinkDestination}} rel="noopener noreferrer">link</a>
```

HyperText Markup Language

28th February 2015 at 2:39pm

Definitions

HTML is a standard plain-text format used for defining the content of a web page.

It consists of a tree of elements expressed using a system of special **tags** enclosed in angle brackets.

Almost the whole of HTML can be used unchanged in the WikiText of a tiddler.

Icon Gallery

18th October 2021 at 12:23pm

Core Icons Learning

Click an icon to copy the title to the clipboard





image-picker Macro

18th April 2016 at 5:58pm

Core Macros Macros

The **image-picker** macro displays an interactive image picker, as can be seen in the core text editor.

Parameters

actions

Wikitext for the action widgets that should be executed when the user selects an image. Within the text, the variable `imageTitle` contains the title of the tiddler containing the selected image.

subfilter

An optional extra filter step, e.g. `tag[MyTag]`

The images are selected by means of a filter expression, into which the **`subfilter`** parameter is spliced as follows:

```
[all[shadows+tiddlers]is[image]$subfilter$!has[draft.of]] -[type[application/pdf]] +[sort[title]]
```

Examples

image-picker Macro (Example 1)

16th November 2021 at 4:50am

Macro Examples

Image: (unset)

☐ Include system tiddlers



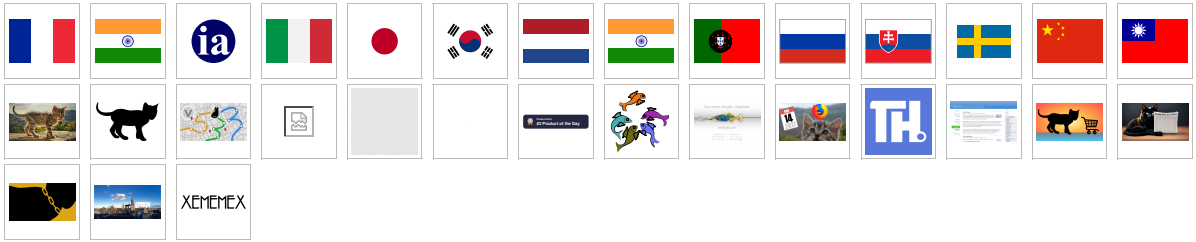


image-picker Macro (Example 2)

16th November 2021 at 4:51am

Macro Examples

Image: (unset)

Include system tiddlers



image-picker Macro (Examples)

18th April 2016 at 5:56pm

image-picker Macro Macro Examples

copy to clipboard

```
Image: <$edit-text tiddler='$:/_MyImage' tag='input' placeholder='(unset)' default=''/>

<$transclude tiddler={{$:/_MyImage}}/>

---

<$macrocall $name='image-picker' actions="

<$action-setfield $tiddler='$:/_MyImage' $value=<<imageTitle>>/>

"/>
```

Try it

This next example shows how the **`subfilter`** parameter limits the list of images (here to those with the "Language" prefix):

copy to clipboard

```
Image: <$edit-text tiddler='$:/_MyImage' tag='input' placeholder='(unset)' default=''/>
```

```
<$transclude tiddler={{$:/_MyImage}}/>

---

<$macrocall $name='image-picker' actions="

<$action-setfield $tiddler='$:/_MyImage' $value=<<imageTitle>>/>

" subfilter="prefix[Language]"/>
```

Try it

ImageGallery Example

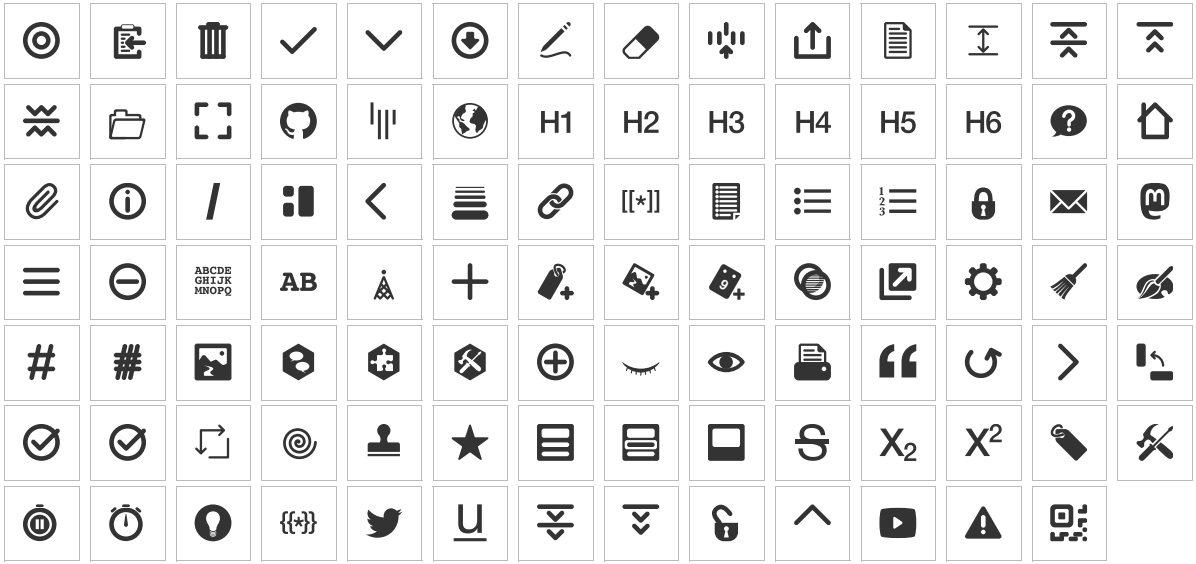
22nd October 2021 at 9:52pm

The ImageGallery Example has been replaced with the new Icon Gallery which is transcluded below

---

Click an icon to copy the title to the clipboard





Images in WikiText

29th January 2022 at 4:26pm

WikiText Working with TiddlyWiki

Image Formatting

Images can be included in WikiText with the following syntax:

```
[img[Motovun Jack.jpg]]
[img[https://tiddlywiki.com/favicon.ico]]
```

You can also insert image tiddlers from the editor toolbar. Click **picture** (🖼) and select a picture file.

If the image source is the title of an image tiddler then that tiddler is directly displayed. Otherwise it is interpreted as a URL and an HTML `<img>` tag is generated with the `src` attribute containing the URL.

A tooltip can also be specified:

```
[img[An explanatory tooltip|Motovun Jack.jpg]]
```

Attributes can be provided to specify CSS classes and the image width and height:

```
[img width=32 [Motovun Jack.jpg]]
[img width=32 class="tc-image" [Motovun Jack.jpg]]
```

Note that attributes can be specified as transclusions or variable references:

```
[img width={{!!mywidth}} class=<<image-classes>> [Motovun Jack.jpg]]
```

The image syntax is a shorthand for invoking the ImageWidget.

Displaying Images via Transclusion

You can also display an image stored in a tiddler by transcluding that tiddler. The disadvantage of this approach is that there is no direct way to control the size of the image.

```
{{Motovun Jack.jpg}}
```

Renders as:



Images as Links

```
<$link to="HelloThere" tooltip="Custom tooltip">{{$:/core/icon}}</$link>
```

Renders as:

Importing Images

Use the **import** button (under the **Tools** tab in the sidebar), or drag and drop. See Importing Tiddlers for details.

Introduced in v5.2.0 You can also import images by dropping or pasting images into the tiddler editor.

ImageWidget

21st November 2023 at 12:43pm

Widgets

Introduction

The image widget displays images that can be specified as a remote URL or the title of a local tiddler containing the image.

Content and Attributes

Any content of the `<$image>` widget is ignored.

Attribute	Description

source	The URL of the image, or the title of an image tiddler
width	The width of the image
height	The height of the image
tooltip	The tooltip to be displayed over the image
alt	The alternative text to be associated with the image
class	CSS classes to be assigned to the `<img>` element
loading	ⓘ **Introduced in v5.2.3** Optional. Set to `lazy` to enable lazy loading of images loaded from an external URI
usemap	ⓘ **New in v5.3.2** Optional usemap attribute to be assigned to the `<img>` element for use with HTML image maps

The width and the height can be specified as pixel values (eg "23" or "23px") or percentages (eg "23%"). They are both optional; if not provided the browser will use CSS rules to size the image.

Image Status Classes

The following CSS classes are automatically added to the `<img>` element to indicate the status of the image. Note that only one of these classes will be added at the same time.

tc-image-loading	When the image is being loaded
tc-image-loaded	The image has been loaded successfully
tc-image-error	The image could not be loaded

External Images and the **_canonical_uri** field

When used to display tiddler-based images, the image widget operates in two distinct modes:

- If the **_canonical_uri** field is present then it is used as the **src** attribute of the generated `<img>` element and the **text** field is ignored
- Without the **_canonical_uri** field, the image widget generates an `<img>` element that embeds the image data directly using a `data:` URI.

See ExternalImages for more details.

ImportCommand

12th July 2017 at 5:38pm

Commands

Import tiddlers from TiddlyWiki (`.html`), `.tiddler` , `.tid` , `.json` or other local files. The deserializer must be explicitly specified, unlike the `load` command which infers the deserializer from the file extension.

```
--import <filepath> <deserializer> [<title>] [<encoding>]
```

The deserializers in the core include:

- application/javascript
- application/json
- application/x-tiddler
- application/x-tiddler-html-div
- application/x-tiddlers
- text/html

- text/plain

The title of the imported tiddler defaults to the filename.

The encoding defaults to "utf8", but can be "base64" for importing binary files.

Note that TiddlyWiki will not import an older version of an already loaded plugin.

Importing Tiddlers

28th March 2017 at 6:02pm

Features

You can import tiddlers into a TiddlyWiki from external files or directly from another TiddlyWiki.

Importing content from external files

There are several ways to import content (including text, images, pdf documents, etc.) from external files:

- Use the 📎 **import** button (under the **Tools** tab in the sidebar) to select a local file
- Drag and drop files from Windows Explorer or OS X Finder etc. into the TiddlyWiki browser window
- Paste content directly from the clipboard using the menu or keyboard shortcut (`ctrl-V` or `cmd-V`)
    - Currently supported in Chrome, Firefox and Edge (but not Internet Explorer)

Most files are imported as individual tiddlers. The exceptions are:

- **.html** files that are recognised as a TiddlyWiki file are parsed to extract the tiddlers within them

- **.json** files are parsed to extract the tiddlers within them. They can be created by exporting them from another TiddlyWiki. TiddlyWiki accepts JSON files that contain either a single tiddler fields object, or an array of tiddler fields objects

Importing content from other TiddlyWiki browser windows

Tiddlers can be imported from other TiddlyWiki browser windows via Drag and Drop. Drag a TiddlyWiki internal link or a tag from one TiddlyWiki browser window to another. Dragging an internal link will import a single tiddler while dragging a tag pill will import all of the tiddlers that carry that tag.

ImportVariablesWidget

28th September 2018 at 5:00pm

Widgets

Introduction

The ImportVariablesWidget imports macro and variable definitions from a list of other tiddlers and makes them available to its children. For example:

```
<$importvariables filter="[tag[mySpecialMacros]]">
All the macros defined in tiddlers with the tag "mySpecialMacros" are available here
</$importvariables>
```

Attributes and Content

The content of the importvariables widget is the scope within which the imported variable definitions are available.

Attribute	Description
filter	Tiddler filter defining the tiddlers from which macro definitions will be imported

Global Macros

So-called global macros are implemented within the main page template (**$:/core/ui/PageTemplate**) by wrapping the page content in the following importvariables widget:

```
<$importvariables filter="[[$:/core/ui/PageMacros]] [all[shadows+tiddlers]tag[$:/tags/Macro]!
has[draft.of]]">
...
</$importvariables>
```

`\import` Pragma

Introduced in v5.1.18 The Pragma: \import is an alternative syntax for using the ImportVariablesWidget. For example, the previous example could be expressed as:

```
\import [[$:/core/ui/PageMacros]] [all[shadows+tiddlers]tag[$:/tags/Macro]!has[draft.of]]
```

Improving TiddlyWiki Documentation

15th January 2019 at 5:56pm

Community

Anyone can submit improvements to the TiddlyWiki documentation that appears on https://tiddlywiki.com. (For improvements to the core code a more complicated process must be used because of the need to test the changes both in the browser and under Node.js before submission).



If you already know GitHub, note that documentation updates must be directed to the `tiddlywiki-com` branch

1. Read and observe the Documentation Style Guide
2. Create an account on https://github.com if you don't already have one
3. If you haven't done so already, sign the Contributor License Agreement as described in Signing the Contributor License Agreement
4. On https://tiddlywiki.com, click "edit" on the tiddler you want to improve
5. You should see a pink banner with the text: *Can you help us improve this documentation? Find out how to edit this tiddler on GitHub*
6. Click on the external link ...**this tiddler on GitHub**
    1. You will be prompted that "you need to fork this repository to propose changes". A "fork" is your own copy of the repository that incorporates the changes you are proposing
7. A new browser tab should open ready to edit the tiddler on github.com
8. Below the edit box for the tiddler text you should see a box labelled **Propose file change**
9. Enter a brief title to explain the change (eg, "Clarify attribute syntax instability")
10. If necessary, enter a longer description too
11. Click the green button labelled **Propose file change**
12. On the following screen, click the green button labelled **Create pull request**

Jermolene or one of the other core developers will then have the opportunity to merge your pull request so that it is incorporated into the next

build of https://tiddlywiki.com.

Mario Pietsch has created these short video tutorials:

IndexedDB Plugin by Andreas Abeck

6th January 2021 at 4:10pm

Community Plugins

A plugin that allows changes to be synchronised with the *IndexedDB* database that is built-in to most browsers.

http://tw5-dev.cibm.de

> Being quite new to TW5 development I tried my best to draft a *SyncAdaptor* to store tiddlers in *IndexedDb*. This is quite useful if an

> application / plugin installation is not possible (we have quite some restrictions at work...).

indexes Operator

29th December 2022 at 10:42am

Filter Operators

purpose	select all data properties of the input titles
input	a selection of titles
parameter	none
output	all the property names or indices contained in the input data tiddlers

Learn more about how to use Filters

Each input title is processed in turn, and is ignored if it does not denote a data tiddler. The sorted list of property names is retrieved from the data tiddler and then dominantly appended to the operator's output.

Note that behind the scenes, the name/value pairs in a dictionary tiddler are represented as a JSON object. JSON objects do not maintain a guaranteed ordering and so we cannot reliably retain the ordering of the data in the file.

Where the content of a tiddler is in JSON format with an array as its root, the **`indexes`** operator retrieves a selection of integer indices instead.

Examples

indexes Operator (Examples)

19th January 2015 at 8:10pm

indexes Operator Operator Examples

```
[{$:/palette}indexes[]sort[title]]
```

→ all the colours defined in the current colour palette

Try it

```
[[$:/HistoryList]indexes[]]
```

→ integer output because the history list is an array

Try it

InfoMechanism

24th November 2020 at 7:58pm

Mechanisms

System tiddlers in the namespace `$:/info/` are used to expose information about the system (including the current browser) so that WikiText applications can adapt themselves to available features.

Information Tiddlers

Title	Description
$:/info/startup-timestamp	ⓘ **Introduced in v5.1.23** Startup timestamp in TiddlyWiki date format
$:/info/browser	Running in the browser? ("yes" or "no")
$:/info/mobile	ⓘ **Introduced in v5.2.3** Is running on a mobile device? ("yes" or "no" - eg, **https://tiddlywiki.com/**)
$:/info/browser/language	ⓘ **Introduced in v5.1.20** Language as reported by browser (note that some browsers report two character codes such as `en` while others report full codes such as `en-GB`)
$:/info/browser/screen/width	Screen width in pixels
$:/info/browser/screen/height	Screen height in pixels
$:/info/node	Running under Node.js? ("yes" or "no")
$:/info/url/full	ⓘ **Introduced in v5.1.14** Full URL of wiki (eg, **https://tiddlywiki.com/**)
$:/info/url/host	ⓘ **Introduced in v5.1.14** Host portion of URL of wiki (eg, **tiddlywiki.com**)
$:/info/url/hostname	ⓘ **Introduced in v5.1.14** Hostname portion of URL of wiki (eg, **tiddlywiki.com**)
$:/info/url/origin	ⓘ **Introduced in v5.1.14** Origin portion of URL of wiki (eg, **https://tiddlywiki.com**)
$:/info/url/pathname	ⓘ **Introduced in v5.1.14** Pathname portion of URL of wiki (eg, /)
$:/info/url/port	ⓘ **Introduced in v5.1.14** Port portion of URL of wiki (eg,)
$:/info/url/protocol	ⓘ **Introduced in v5.1.14** Protocol portion of URL of wiki (eg, **https:**)
$:/info/url/search	ⓘ **Introduced in v5.1.14** Search portion of URL of wiki (eg,)
$:/info/darkmode	ⓘ **Introduced in v5.1.23** Is dark mode enabled? ("yes" or "no")

InfoPanel

29th November 2020 at 7:30pm

Features

Each tiddler has a panel of additional information. To reveal it, click the ⌄ button in the tiddler's toolbar and then choose ⓘ **info** from the dropdown list.

The info panel has the following tabs:

- **Tools** - This offers buttons for various actions you can perform on the tiddler. The checkbox next to each button lets you promote an action to the tiddler's toolbar - this will affect all of the tiddlers in your wiki
- **Backlinks**, **Tagging**, **List** and **Listed** - These list various kinds of related tiddlers. See Using links to navigate between tiddlers
- **Fields** - This summarises all of the tiddler's fields, except for **text**
- **Advanced** - This indicates whether the tiddler is a shadow. If it is, this also reveals which plugin it comes from and whether it has been overridden by an ordinary tiddler

To close the info panel, click anywhere outside it.

InitCommand

17th June 2014 at 11:25pm

Commands

Initialise an empty *WikiFolder* with a copy of the specified edition.

```
--init <edition> [<edition> ...]
```

For example:

```
tiddlywiki ./MyWikiFolder --init empty
```

Note:

- The wiki folder directory will be created if necessary
- The "edition" defaults to **empty**
- The init command will fail if the wiki folder is not empty
- The init command removes any `includeWikis` definitions in the edition's `tiddlywiki.info` file
- When multiple editions are specified, editions initialised later will overwrite any files shared with earlier editions (so, the final `tiddlywiki.info` file will be copied from the last edition)
- `--editions` returns a list of available editions

See also:

- Environment Variables on Node.js

Inline Mode WikiText

17th September 2022 at 9:49am

WikiText Parser Modes

Some WikiText is recognised only while the parser is in inline mode.

These WikiText types can be expressed without an entire line of text. They aren't required to be all on one line, just that they *CAN* be expressed within a single line. And as such, more than one can appear within a single line. In other words, line endings are not involved while the parser tries to find where the particular WikiText begins and ends While the parser is in inline mode, it will recognise the punctuation for these WikiText

types:

- Dashes in WikiText
- Formatting in WikiText
- HTML in WikiText
- Images in WikiText
- Linking in WikiText
- Macro Calls
- Styles and Classes in WikiText (single line version only)
- Transclusion in WikiText
- Variables in WikiText
- Widgets in WikiText



Macro Calls and Transclusion in WikiText will be recognised in block mode if the macro call or transclusion spans an entire line.



The other **inline mode** WikiText types are technically *ONLY* detected while the parser is in **inline mode**. However, the opening punctuation will also trigger the start of Paragraphs in WikiText which will automatically cause the parser to go into **inline mode**. Therefore, practically speaking, it is just as useful to consider these WikiText types as recognised while the parser is in either **inline mode** or **block mode**

While processing the *enclosed* text of some of these WikiText types, the parser will not look for new WikiText. But for rest of these WikiText types, the parser will continue in **inline mode** for the *enclosed* text. While parsing that text, it might encounter something which moves it to block mode.

Innerwiki Plugin

27th January 2019 at 11:41am

OfficialPlugins

The Innerwiki Plugin enables TiddlyWiki to embed a modified copy of itself (an "innerwiki"). The primary motivation is to be able to produce screenshot illustrations that are automatically up-to-date with the appearance of TiddlyWiki as it changes over time, or to produce the same screenshot in different languages.

In the browser, innerwikis are displayed as an embedded iframe. Under Node.js Google's Puppeteer is used to load the innerwikis as off-screen web pages and then snapshot them as a PNG image.

See the demo at https://tiddlywiki.com/plugins/tiddlywiki/innerwiki

Insert link

17th August 2016 at 12:57pm

Editor toolbar

This will give you a dialog to search for and pick existing tiddlers in the wiki. When you pick a tiddler it will be inserted as a WikiText link where the cursor is in the text field.

Pressing 🔗 will give you `[[Some link]]`

It wil **not** insert: external web links or picture links.

Insert picture

17th August 2016 at 1:02pm

Editor toolbar

This will give you a dialog to search for and pick existing image tiddlers in the wiki. When you pick a tiddler it will be inserted as a WikiText image link where the cursor is in the text field.

Pressing will give you `[img[$:/favicon.ico]]`

insertafter Operator

23rd February 2022 at 1:44am

Filter Operators Listops Operators Order Operators

purpose	insert an item **T** into a list immediately after an item **A**
input	a selection of titles
suffix	(optional) the name of a variable containing the title of the tiddler after which this one should be inserted
parameter	the **insertafter** operator accepts 1 or 2 parameters, see below for details
output	the input tiddler list with the new entry inserted

Learn more about how to use Filters

Introduced in v5.2.3

The **insertafter** operator requires at least one parameter which specifies the title to insert into the input list. A second parameter can be used to specify the title after which the new title should be inserted.

A suffix can also be used to specify **A**, the title after which the new title should be inserted, but this form is deprecated. Instead, the two-parameter form is recommended. If the two-parameter form is used, the suffixes **start** and **end** can be used to specify where the item should be inserted if **A** is not found.

```
insertafter:<after-title-variable>[<title>]
insertafter:<missing-location>[<title>],[<after-title>]
```

- **title** : a title **T** to insert in the input list.
- **after-title** : (optional). Insert **T** after this title **A** in the input list.
- **after-title-variable** : (optional). The name of a variable specifying **A** instead of the `after-title` parameter.
- **missing-location** : (optional). Either `start` or `end` : where to insert **T** if **A** is not found in the list.

If the item **A** isn't present in the input list then the new item is inserted at the end of the list. ⓘ **Introduced in v5.2.3** The suffixes **start** and **end** can be spedified to control where the new item is inserted when **A** is not found. The suffix **end** is the default, inserting the new item at the end of the list. The suffix **start** will cause the new item to be inserted at the start of the list when **A** is not found.

Either parameter can be a string, a text reference or a variable

If **A** is specified as both a suffix and a parameter, the parameter takes precedence

Examples

insertafter Operator (Examples)

23rd February 2022 at 1:44am

insertafter Operator Operator Examples

These examples use the following predefined variables:

- **after-title**: `Friday`
- **missing-title**: `Yesterday`

```
[list[Days of the Week]insertafter[Today]]
```

Try it

```
[list[Days of the Week]insertafter[Today],[Tuesday]]
```

Try it

```
[list[Days of the Week]insertafter[Today],<after-title>]
```

Try it

```
[list[Days of the Week]insertafter:after-title[Today]]
```

Try it

```
[list[Days of the Week]insertafter[Today],<missing-title>]
```

Try it

```
[list[Days of the Week]insertafter:missing-title[Today]]
```

Try it

insertbefore Operator

23rd February 2022 at 1:44am

Filter Operators Listops Operators Order Operators

purpose	insert an item **T** into a list immediately before an item **B**
input	a selection of titles
suffix	ⓘ **Introduced in v5.2.3** (optional) the name of a variable containing the title of the tiddler before which this one should be inserted
parameter	ⓘ **Introduced in v5.2.2** the **insertbefore** operator accepts 1 or 2 parameters, see below for details
output	the input tiddler list with the new entry inserted

Learn more about how to use Filters

ⓘ **Introduced in v5.2.2**

The **insertbefore** operator requires at least one parameter which specifies the title to insert into the input list. A second parameter can be used to specify the title before which the new title should be inserted.

Introduced in v5.2.3

Using the suffix to specify **B**, the title before which the new title should be inserted, is deprecated. Instead, the two-parameter form is recommended. If the two-parameter form is used, the suffixes **start** and **end** can be used to specify where the item should be inserted if **B** is not found.

```
insertbefore:<before-title-variable>[<title>]
insertbefore:<missing-location>[<title>],[<before-title>]
```

- **title** : a title **T** to insert in the input list.
- **before-title** : (optional). Insert **T** before this title **B** in the input list.
- **before-title-variable** : (optional). The name of a variable specifying **B** instead of the `before-title` parameter.
- **missing-location** : (optional). Either `start` or `end` : where to insert **T** if **B** is not found in the list.

If the item **B** isn't present in the input list then the new item is inserted at the end of the list. **Introduced in v5.2.3** The suffixes **start** and **end** can be spedified to control where the new item is inserted when **B** is not found. The suffix **end** is the default, inserting the new item at the end of the list. The suffix **start** will cause the new item to be inserted at the start of the list when **B** is not found.



Either parameter can be a string, a text reference or a variable



If **B** is specified as both a suffix and a parameter, the parameter takes precedence

Examples

insertbefore Operator (Examples)

23rd February 2022 at 1:44am

insertbefore Operator Operator Examples

These examples use the following predefined variables:

- **before-title**: `Friday`
- **missing-title**: `Yesterday`

```
[list[Days of the Week]insertbefore[Today]]
```

Try it

```
[list[Days of the Week]insertbefore[Today],[Tuesday]]
```

Try it

```
[list[Days of the Week]insertbefore[Today],<before-title>]
```

Try it

```
[list[Days of the Week]insertbefore:before-title[Today]]
```

Try it

```
[list[Days of the Week]insertbefore[Today],<missing-title>]
```

Try it

```
[list[Days of the Week]insertbefore:missing-title[Today]]
```

Try it

Installing a plugin from the plugin library

17th June 2022 at 2:21pm

Plugins

If you want to follow the steps *side by side* you can open **this** tiddler in a new window.

Follow these instructions when using TiddlyWiki as a standalone, single file wiki.

1. Open your TiddlyWiki in a browser

2. Create a backup of your current wiki file (just in case)
3. Open the ⚙ ControlPanel
    - Click on the **Plugins** tab and then the
    - ⊙ **Get more plugins** button
4. Click ≫ **open plugin library** to open the official plugin library
5. When the library listing is loaded:
    1. Use the tab to select between **plugins**, **themes** and **languages**
    2. Use the **search** box to search the plugin details
6. Click the **install** button to install a plugin
7. Save your TiddlyWiki ⊘
8. If a yellow warning bar appears at the top of the window, refresh the window so that TiddlyWiki completes installation of the plugin ↺
    - ⓘ **Introduced in v5.1.22** It is no longer necessary to refresh TiddlyWiki when deleting plugins that support dynamic loading. See the PluginMechanism for more details
9. The plugin should now be available for use

Note

> If you use TiddlyWiki with **Node.js** have a closer look at Installing official plugins on Node.js
> An overview can be found at Plugins

Installing custom plugins on Node.js

17th June 2022 at 3:01pm

PluginsCS TiddlyWiki on Node.js

Introduction

There are several ways in which official plugins and custom plugins can be installed when using TiddlyWiki with a Node.js client-server configuration.

Note

> For instructions on installing plugins from libraries to **single-file** wikis learn more at Plugins.

Plugin Load Order

Using the Node.js client-server configuration plugins are activated in the following order:

1. Plugins found using the OS environment variables
   - See: PluginFolders
   - and: Environment Variables on Node.js
2. Plugins stored in the wiki `/plugins` path
   - See: PluginFolders
3. Plugins specified in the command line
   - See: Using TiddlyWiki on Node.js ... the notes about ⓘ **Introduced in v5.1.20**
4. Plugins imported by drag and drop as wiki content
   - See: Installing a plugin from the plugin library or Manually installing a plugin

Important:

- **Elements lower in the list take precedence**

- Including a plugin as an ordinary tiddler by drag and drop into the browser, will result in the plugin only being active in the browser
  - It will not be available under Node.js

Option 1:

- Using a Node.js configuration option 1 is the most common one
- It lowers maintenence, because all plugins can be updated in one go

Option 2:

- Option 2 is preferred if you want to work with a fixed configuration
- Plugin updates only effect the corresponding wiki

Option 3:

- This option allows to **add and start** an edition without the need to overwrite an existing `tiddlywiki.info` file
- It allows you to start any edition as a client-server edition, even if the tiddlywiki.info file didn't specify it.

Option 4:

- Is the exact same mechanism used for a single-file wiki
- This mechanism should only be used in a Node.js configuration for **testing and debugging** purposes, since the plugins are stored in the `tiddlers/` directory

Installing official plugins on Node.js

17th June 2022 at 3:23pm

PluginsCS TiddlyWiki on Node.js

Follow these instructions when using TiddlyWiki with a client-server Node.js configuration:

1. Identify the plugins you want to install using the **Plugins** tab of ⚙ ControlPanel (**don't** install the plugins from here, though)
   - Plugins are identified by their type (ie language, theme or plugin) and their publisher and title. For example, the plugin `$:/plugins/tiddlywiki/internals` is referred to as **tiddlywiki/internals**
2. Quit the server if it is running
3. Edit the `tiddlywiki.info` file (it is in JSON format) and locate the `plugins`, `themes` or `languages` section (see below)
4. Add entries corresponding to the plugins you wish to add
   - Take care to retain **commas** to separate items
   - Do **not** terminate the last item in a list with a comma
5. Restart the server

```
{
    "plugins": [
        "tiddlywiki/codemirror"
    ],
    "themes": [
        "tiddlywiki/vanilla",
        "tiddlywiki/snowwhite"
    ],
    "languages": [
        "es-ES",
```

```
        "fr-FR",
        "en-EN"
    ]
}
```

Note

> An overview of working with plugins can be found at Plugins.
> Also see: Installing custom plugins on Node.js.

Installing TiddlyWiki on Microsoft Internet Information Server

7th May 2020 at 12:58pm

Saving TiddlyWiki on Node.js Windows

Microsoft's Internet Information Server (IIS) is the built-in web server for Windows. It can be useful to host TiddlyWiki within IIS in order to take advantage of IIS features like URL rewriting, static file hosting and automatic restarts after a crash.

1. Enable IIS

1. In Windows, access the Control Panel and click **Add or Remove Programs**
2. In the Add or Remove Programs window, click **Add/Remove Windows Components**.
3. Select the **Internet Information Services (IIS)** check box
4. Click **Next**, then click **Finish**

2. Install the required tools

1. Install the IIS module HttpPlatformHandler from https://www.iis.net/downloads/microsoft/httpplatformhandler
2. Install Node.js from https://nodejs.org/
3. Install Git from https://git-scm.com/
    - *optional; only required when using the latest TiddlyWiki release pulled directly from GitHub – see below*

3. Install TiddlyWiki and setup a new wiki

1. Create a convenient directory for the wiki (e.g. `C:\MyStuff`)
2. Within it, create a file called `C:\MyStuff\package.json` with the content:

```
{
    "name": "MyStuff",
    "description": "A description of this wiki",
    "dependencies": {
        "sax": "1.2.4",
        "tiddlywiki": "*"
    }
}
```

copy to clipboard

3. Also create a file called `C:\MyStuff\web.config` with the content:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
    <system.webServer>
        <handlers>
            <add
                name="httpplatformhandler"
                path="*"
                verb="*"
                modules="httpPlatformHandler"
                resourceType="Unspecified"
                requireAccess="Script" />
        </handlers>
        <httpPlatform
           stdoutLogEnabled="true"
           stdoutLogFile=".\node.log"
           startupTimeLimit="20"
           processPath="C:\Program Files\nodejs\node.exe"
           arguments=".\node_modules\tiddlywiki\tiddlywiki.js ./wiki --listen port=PORT
path-prefix=/MyApp">
            <environmentVariables>
                <environmentVariable name="PORT" value="%HTTP_PLATFORM_PORT%" />
                <environmentVariable name="NODE_ENV" value="Production" />
            </environmentVariables>
        </httpPlatform>
    </system.webServer>
```

```
</configuration>
```

copy to clipboard

4. Create a subdirectory called "wiki" (i.e. `C:\MyStuff\wiki`)
5. Create a file called `C:\MyStuff\wiki\tiddlywiki.info` with the content:

```
{
    "description": "My wiki",
    "plugins": [
        "tiddlywiki/tiddlyweb",
        "tiddlywiki/filesystem"
    ],
    "themes": [
        "tiddlywiki/vanilla",
        "tiddlywiki/snowwhite"
    ]
}
```

copy to clipboard

6. Create a subdirectory called "tiddlers" (i.e. `C:\MyStuff\wiki\tiddlers`)
7. Within it, create a file called `C:\MyStuff\wiki\tiddlers\config-tiddlyweb-host.tid` with the content:

```
title: $:/config/tiddlyweb/host
text: $protocol$//$host$/MyApp/
```

copy to clipboard

- (See the documentation for details of the configuration fields for HttpPlatformHandler

8. Execute the command `npm install` within the `C:/MyStuff` directory

4. Setup the application in IIS

- In Windows, run the IIS Manager application (use the start menu to run `inetmgr.exe`)
- Locate the server in the **Connections** column on the left, and click the disclosure triangle to show its content
- Open the **Sites** folder
- Right click on the **Default Web Site** entry, and select **Add application..** from the menu
- Enter the following information in the dialogue box:
  1. **Alias**: `MyApp`
  2. **Physical path**: `C:\MyStuff`
- Click **OK**

5. Test the application

Test the app by visiting http://localhost/MyApp/ in a browser.

Notes

- If you require authentication, specify a username and password in the `--listen` command in `web.config`. For example:
  - `arguments=".\node_modules\tiddlywiki\tiddlywiki.js ./wiki-server --listen username=joe "password=bloggs" port=PORT path-prefix=/MyApp">`
  - Take note of the need to use double quotes around non-alphanumeric passwords, and to HTML encode them into `"`
- If you change the settings in the `web.config` file, or modify the app code, then you'll need to restart the server using the IIS manager application

Installing TiddlyWiki on Node.js

16th November 2022 at 12:08am

Linux Mac Saving TiddlyWiki on Node.js Windows

1. Install Node.js
   - Linux:

> *Debian/Ubuntu*:
> `apt install nodejs`
> May need to be followed up by:
> `apt install npm`
>
> *Arch Linux*
> `yay -S tiddlywiki`

> (installs node and tiddlywiki)

- Mac

  > `brew install node`

- Android

  > Termux for Android

- Other

  > See http://nodejs.org

2. Open a command line terminal and type:

   > `npm install -g tiddlywiki`
   >
   > If it fails with an error you may need to re-run the command as an administrator:
   >
   > `sudo npm install -g tiddlywiki` (Mac/Linux)

3. Ensure TiddlyWiki is installed by typing:

   > `tiddlywiki --version`

   - In response, you should see TiddlyWiki report its current version (eg "5.3.3". You may also see other debugging information reported.)

4. Try it out:
    1. `tiddlywiki mynewwiki --init server` to create a folder for a new wiki that includes server-related components
    2. `tiddlywiki mynewwiki --listen` to start TiddlyWiki
    3. Visit http://127.0.0.1:8080/ in your browser
    4. Try editing and creating tiddlers
5. Optionally, make an offline copy:
    - click the **save changes** button in the sidebar, **OR**
    - `tiddlywiki mynewwiki --build index`

The `-g` flag causes TiddlyWiki to be installed globally. Without it, TiddlyWiki will only be available in the directory where you installed it.



If you are using Debian or Debian-based Linux and you are receiving a `node: command not found` error though node.js package is installed, you may need to create a symbolic link between `nodejs` and `node`. Consult your distro's manual and `whereis` to correctly create a link. See github issue 1434.

Example Debian v8.0: `sudo ln -s /usr/bin/nodejs /usr/bin/node`



You can also install prior versions like this:
`npm install -g tiddlywiki@5.1.13`

Installing TiddlyWiki Prerelease on Node.js

22nd October 2019 at 11:55am

TiddlyWiki on Node.js

1. Clone a local copy of the TiddlyWiki5 GitHub repository from https://github.com/Jermolene/TiddlyWiki5
2. Open a command line terminal and change the current working directory to the root of the TiddlyWiki5 repo
3. Type `npm link` (Windows) or `sudo npm link` (Mac/Linux) to tell *npm* to use this copy of the repo as the globally installed one
4. Inside the root, you can launch TiddlyWiki like this:
   `tiddlywiki editions/tw5.com-server --listen`

After this procedure you can work with TiddlyWiki5 via *npm* as though it had been installed in the usual way with `npm install -g tiddlywiki`.

Update the clone from time to time in order to ensure that you have the latest code.

Instruction Tiddlers

17th January 2015 at 4:25pm

Improving TiddlyWiki Documentation

Instruction tiddlers talk directly to the reader and guide them through a process. The reader is likely to be a beginner or an intermediate user.

Such tiddlers can be subcategorised as:

Welcome

- What is TiddlyWiki and why should I care?
- Demonstrations of key features and benefits
- Frequently asked questions
- Examples of TiddlyWiki in the field

- Information about the project itself

Tutorial

- An ordered presentation of material for beginners
- Each tiddler introduces one new point or concept
- Its main content contains very few links
- A revealable "Find out more" section at the end can offer related links

Exercise

- Accompanying a tutorial tiddler
- Solution revealed on demand

How-to

- A list of numbered steps for performing a small specific task
- Concise, with links to reference tiddlers where appropriate
- Often has a preamble to clarify the nature of the task

Example

- Accompanying a reference tiddler
- Can contain explanations and similar commentary
- Kept separate to keep the reference tiddler pure

Instruction tiddlers talk directly to the reader as "you". They can be reasonably chatty.

But they avoid excessively colloquial language, cultural or topical references and attempts at humour, as these can baffle or even offend the international readership. They also avoid potentially frustrating the reader with descriptions of features as "convenient" or "easy".

Interactive Git Documentation by Devin Weaver

21st June 2015 at 11:45pm

Examples

A git choose-your-own-adventure!ⓡ for walking yourself though fixing a *broken* Git repository.

http://sukima.github.io/GitFixUm/

> This document is an attempt to be a fairly comprehensive guide to recovering from what you did not mean to do when using git. It isn't that git is so complicated that you need a large document to take care or your particular problem, it is more that the set of things that you might have done is so large that different techniques are needed depending on exactly what you have done and what you want to have happen.

Interchangeable Filter Run Prefixes

27th March 2023 at 3:06pm

Filter Run Prefix

Interchangeable Filter Run Prefixes

In technical / logical terms:

Run	Equivalent named prefix	Interpretation	Output
`[run]`	`:or[run]`	de-duplicated union of sets	... OR run
`+[run]`	`:and[run]`	accumulation of filter steps	... AND run
`-[run]`	`:except[run]`	difference of sets	... AND NOT run
`~[run]`	`:else[run]`	else	... ELSE run
`=[run]`	`:all[run]`	union of sets without de-duplication	... OR run

The input of a run is normally a list of all the non-shadow tiddler titles in the wiki (in no particular order).
But the `+` prefix can change this:

Prefix	Input
`-`, `~`, `=`, `:intersection` or none	`[all[]]` tiddler titles, unless otherwise determined by the first filter operator
`+`, `:filter`, `:map`, `:reduce`, `:sort`	the filter output of all previous runs so far

Precisely because of varying inputs, be aware that both prefixes `-` and `+` do not behave inverse to one another!

For example, in both of the following, `$:/baz` will only be removed if it actually exists:

- `foo bar $:/baz -[is[system]]`
- `foo bar $:/baz -[prefix[$:/]]`

To understand why, consider the input for both final runs with their `-` prefix.

In order to remove `$:/baz` in any case, existing or not, simply use the `+` prefix with negated filter operators:

- `foo bar $:/baz +[!is[system]]`

- `foo bar $:/baz +[!prefix[$:/]]`

Difference between + and intersection

For the difference between `+` and `:intersection`, see Intersection Filter Run Prefix (Examples).

For Developers

To create a new filter run prefix, create a Javascript module with a module-type of `filterrunprefix`.

Intersection Filter Run Prefix

10th July 2023 at 9:33am

Named Filter Run Prefix

purpose	find the intersection of titles from previous runs with titles in this filter
input	all titles from previous filter runs
output	the titles that are present in both the result of this filter run and the output from previous runs

Introduced in v5.1.23
TypeError: setting getter-only property "parentNode"

The filter output from previous runs is set aside. The `:intersection` filter run is started with all tiddler titles as input. Once this latest filter run has completed, the latest output is compared to the set-aside output. A new output is produced that contains only titles that appeared in both the set-aside output and the latest output.

Intersection Filter Run Prefix (Examples)

5th March 2023 at 1:53pm

Intersection Filter Run Prefix

```
A B C D :intersection[enlist[C D E F]]
```

Try it

```
A B C D :intersection[enlist[A B C D]]
```

Try it

```
A B C D :intersection[enlist[E F G H]]
```

Try it

```
:intersection[enlist[A B C D]]
```

→ Using intersection in the first filter run is of little use. The result will always be empty.

Try it

Difference between `+` and `:intersection`

The `+` prefix should be thought of as an "AND" in formal logic, e.g. "give me all titles that satisfy condition A **and** condition B". But it's not suitable for all cases; if condition B uses a filter operator that replaces its input, then it will be difficult to use the `+` prefix. For example, if you wanted to find out what tags two tiddlers have in common, you might try to write a filter expression like:

- `[[field Operator]tags[]] +[[compare Operator]tags[]]`

But that won't work, because the second filter run will end up throwing away its input and replacing it with an input consisting of the single title `[[compare Operator]]`. So the result you'd get from that filter expression would be just the tags of the `compare Operator` tiddler.

For cases like this, the `:intersection` prefix is what you need. It takes the filter output so far, *sets it aside*, and starts the next filter run with all tiddler titles as input. Then once the latest filter run has completed, it takes the latest output, compares it to the set-aside output, and produces a new output that contains only titles that appeared in both the set-aside output and the latest output. So to get only the tags that the `field Operator` and `compare Operator` tiddlers have in common, you would write a filter expression like this:

- `[[field Operator]tags[]] :intersection[[compare Operator]tags[]]`

The following examples use the field Operator and compare Operator tiddlers and their respective tags.

```
[[field Operator]tags[]]
```

Try it

```
[[compare Operator]tags[]]
```

Try it

```
[[field Operator]tags[]] +[[compare Operator]tags[]]
```

→ The "field Operator" tiddler's tags are lost, so only the "compare Operator" tiddler's tags are returned

Try it

```
[[field Operator]tags[]] :intersection[[compare Operator]tags[]]
```

→ Returns the tags that both tiddlers have in common

Try it

Introducing TiddlyDesktop Video

12th September 2014 at 5:04pm

TiddlyDesktop Videos

This brief introduction shows how to install and use TiddlyDesktop:

Introduction to filter notation

3rd November 2020 at 12:14pm

Filters Learning

This explains the basics of writing a filter to select a set of tiddlers. For a more technical presentation, see Filter Syntax.



Filters do nothing if you just type them into a tiddler on their own. They need a context. An easy way to experiment with filters is to type them into the **Filter** tab of **Advanced Search**.

The simplest case is where you already know exactly which tiddlers you want. Type each title in double square brackets, with a space between each one and the next:

```
[[Recipe book]] [[ScrambledEggs]] [[Mom's apple pie]]
```

You can omit the square brackets when a title doesn't contain any spaces:

```
[[Recipe book]] ScrambledEggs [[Mom's apple pie]]
```

The double square brackets are actually a shorthand for this:

```
[title[ScrambledEggs]]
```

... which gives us the **general model** for any filter:

```
[operator[parameter]]
```

For instance, here's how to select all the tiddlers that have been tagged **`Recipe`**:

```
[tag[Recipe]]
```

We can reverse the meaning by adding an exclamation mark `!` just before the operator. For example, we can select any tiddlers that do *NOT*

have the **Recipe** tag:

> `[!tag[Recipe]]`

Tiddlers can be filtered by other fields than just title and tags:

> `[field:serving[4]]`

That example will select any tiddlers that have **4** in their **serving** field.

As the word "serving" isn't a standard filter operator (and isn't likely to become one), you can safely omit the `field:` prefix:

> `[serving[4]]`

Combinations

The filters we've looked at so far have involved just one step each. But you can **run** several steps together like this:

> `[tag[Vegetarian]!tag[soup]serving[4]]`

Notice how the entire run is contained in a single pair of square brackets.

A tiddler has to match *ALL* of the steps in a run. So the example above retrieves vegetarian recipes (other than soups) for 4 people.

A sequence of separate runs will select the tiddlers that match *ANY* of the runs. We can use this to find recipes that serve either 3, 4 or 5 people:

> `[serving[3]] [serving[4]] [serving[5]]`

If we want to ignore vegetarian recipes that serve 4, we can say this:

```
[serving[3]] [serving[4]!tag[Vegetarian]] [serving[5]]
```

By default, each run considers every tiddler in the wiki. But we can use a `+` sign to force a run to consider only the tiddlers that were selected by the preceding runs:

```
[serving[3]] [serving[4]] [serving[5]] +[tag[Vegetarian]] +[sort[title]]
```

This selects recipes for 3, 4 or 5 people, then filters *THOSE* to keep only the vegetarian ones, and finally sorts any that are left into alphabetical order of title.

In a similar way, we can use a `-` sign to *REMOVE* a run's tiddlers from the result so far. Here we select all vegetarian recipes apart from two:

```
[tag[Vegetarian]] -[title[ScrambledEggs]] -BeansOnToast
```

Special parameters

The parameter of each step we've seen so far has been in square brackets. It means that TiddlyWiki will filter for the exact string found between the brackets. But two other kinds of bracket are possible:

Curly brackets `{}` mean that the parameter is a TextReference, and it will be replaced with content from another tiddler. For example, if we have a tiddler with the title **`Preference`** whose content is the single word **`Vegetarian`**, we can say

```
[tag{Preference}]
```

In this simplest form the TextReference will take the full content of the tiddler (in technical terms, the text field of the tiddler) and substitute it in place of the TextReference. This way the tiddler's content will become the filter parameter, just like if you have written `[tag[Vegetarian]]`. But it gives you the added flexibility to change the parameter by changing the content of the Preference tiddler.

Angle brackets `<>` mean that the parameter is the name of a variable whose value is to be used instead. Here we use the built-in **`currentTiddler`** variable in a filter that selects any tiddlers whose text contains the title of the current one:

> `[search<currentTiddler>]`

Note that these special brackets cannot be nested. It is not possible for example to write `[search{<currentTiddler>}]`.

Multiple parameters

ⓘ **Introduced in v5.1.23** Some steps accept multiple parameters which are separated by a `,` character.

> Example: `[param1],[param2]` or `<param1>,{param2}`

Introduction to Lists

12th December 2017 at 6:51pm

Learning

Making lists of items plays a central role in working with TiddlyWiki. The following is a brief run-through of some of the more common techniques.

TiddlyWiki uses lists internally for many purposes so the word can have several meanings. Here we are concerned with displaying sequences of

items, but not necessarily presented as a conventional bullet list.

Manually Typed Lists

WikiText

WikiText lists are manually typed lists that use a special character to specify what sort of list it is and how it should be displayed.

Examples include bullet lists created with asterisks (*) and numbered lists (#). Behind the scenes, WikiText lists are based on the standard `<ul>` and `<li>` HTML elements. For more information see Lists in WikiText.

An example of a typed list in WikiText:

copy to clipboard

```
* Greatest Movies of All Time
** Casa Blanca
** Pride and Prejudice and Zombies
```

That renders as:

- Greatest Movies of All Time
  - Casa Blanca
  - Pride and Prejudice and Zombies

Generated Lists

Lists can be automatically generated with the ListWidget using filters in which filter operators specify criteria for selecting which tiddlers are desired as output. Shortcut macros are provided for some common types of list.

ListWidget

The ListWidget is the most powerful tool for creating lists. It allows the filtered output to be manipulated and styled into forms that may not seem to resemble lists at all, for example tables or complex texts. For more details, see ListWidget.

An example to show all tiddlers tagged with "HelloThere" might look like:

copy to clipboard

```
<$list filter="[tag[HelloThere]]"><$view field="title"/><br/></$list>
```

That renders as:

A Gentle Guide to TiddlyWiki
Discover TiddlyWiki
Some of the things you can do with TiddlyWiki
Ten reasons to switch to TiddlyWiki
Examples
What happened to the original TiddlyWiki?
Funding TiddlyWiki
Open Collective



Even tiddlers themselves are made with the ListWidget. The **ViewTemplate** makes use of the ListWidget to fetch all specified templates that are used to show a tiddlers title, tags, text and more.

Filtered transclusion

The syntax for filtered transclusion `{{{...}}}` takes a filter as input and outputs a linked list of matching titles. You can also apply a template, for example:

copy to clipboard

```
{{{ [tag[HelloThere]] || $:/core/ui/TagTemplate }}}
```

That renders as:

A Gentle Guide to TiddlyWiki Discover TiddlyWiki Some of the things you can do with TiddlyWiki Ten reasons to switch to TiddlyWiki Examples What happened to the original TiddlyWiki? Funding TiddlyWiki Open Collective

list-links Macro

The list-links macro gives a preformatted list, typically a bullet list, in a more simplified way than by using the ListWidget. Behind the scenes it really is the ListWidget applying a default template to each list item.

copy to clipboard

```
<<list-links "[tag[HelloThere]]">>
```

That renders as:

- A Gentle Guide to TiddlyWiki
- Discover TiddlyWiki
- Some of the things you can do with TiddlyWiki
- Ten reasons to switch to TiddlyWiki
- Examples
- What happened to the original TiddlyWiki?
- Funding TiddlyWiki
- Open Collective

Other “list related” features

list, list-before and list-after are all field names to control the position of tiddlers in a list. Fields are a way to add additional bits of structured information to a tiddler such as date, quantity, category, etc.

list and listed are *filter operators* to, respectively, select and find titles in lists.

Introduction Video

26th November 2014 at 4:30pm

Videos Working with TiddlyWiki

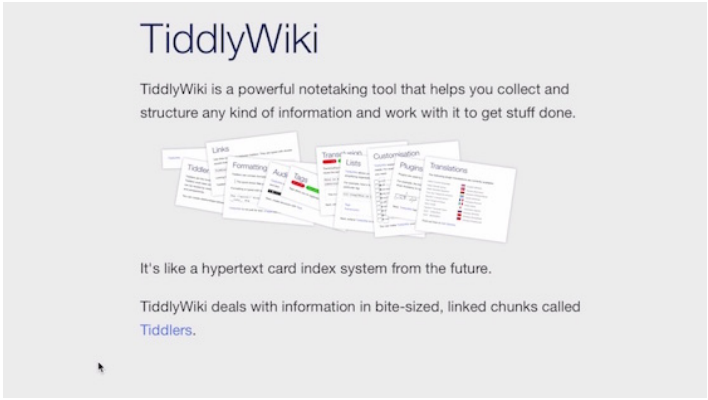
This brief presentation explains the basic principles of TiddlyWiki.

The TiddlyWiki used to produce the video can be found here:

https://tiddlywiki.com/editions/introduction/

Introduction Video Thumbnail.jpg

picture



is Operator

7th November 2020 at 12:09pm

Common Operators Filter Operators Negatable Operators

purpose	filter the input by fundamental category
input	a selection of titles
parameter	**C** = a category
output	those input tiddlers that belong to category **C**
! **output**	those input tiddlers that do *NOT* belong to category **C**

Learn more about how to use Filters

The parameter **c** is one of the following fundamental categories:

Category	Matches any tiddler title that...
`current`	is the current tiddler
`image`	has an image ContentType
`binary`	has a binary ContentType
`missing`	does not exist (other than possibly as a shadow tiddler), regardless of whether there are any links to it
`orphan`	has no hard links to it
`shadow`	is a shadow tiddler, regardless of whether it has been overridden with a non-shadow tiddler
`system`	is a system tiddler, i.e. its title starts with `$:/`
`tag`	is in use as a tag
`tiddler`	exists as a non-shadow tiddler
`variable`	ⓘ **Introduced in v5.1.20** exists as a variable (whether or not that variable has a non-blank value)
`blank`	ⓘ **Introduced in v5.1.20** is blank (i.e. is a zero length string)
`draft`	ⓘ **Introduced in v5.1.23** is a draft of another tiddler. Synonym for `[has[draft.of]]`

If **c** is anything else an error message is returned. ⓘ **Introduced in v5.1.14** if **c** is blank, the output is passed through unchanged (in earlier versions an error message was returned).

`!is[tiddler]` is a synonym for `is[missing]`, and vice versa.

When **is[missing]** is the first operator in a run, its output is always empty. And when **is[shadow]** comes first, it outputs only those shadow tiddlers that have been overridden. This is because the initial input to a run contains only *NON*-shadow tiddlers.

Use `has[title]` to match any tiddler title that is either a tiddler *or* a shadow tiddler.

The **all** operator is similar to **is**, but its scope is the whole wiki.

Examples

is Operator (Examples)

19th January 2015 at 2:46pm

is Operator Operator Examples

```
[is[tag]]
```

Try it

```
[!is[tag]]
```

Try it

`[all[shadows]!is[system]]`

→ shadow tiddlers that don't start with `$:/`

Try it

`[is[system]tag[$:/tags/Stylesheet]]`

→ system stylesheets

Try it

`[all[shadows]is[system]tag[$:/tags/Stylesheet]]`

→ shadow system stylesheets

Try it

`[is[shadow]]`

→ overridden shadow tiddlers

Try it

```
[is[missing]]
```

→ empty because its input contains only tiddlers that exist

Try it

Japanese (Japan) Edition

27th December 2020 at 2:00am

Languages

TiddlyWikiの日本語バージョンはこちら:

- **空白のコピー** : https://tiddlywiki.com/languages/ja-JP/empty.html

JavaScript

11th February 2014 at 8:53pm

Definitions

JavaScript is a computer language that was originally introduced by browsers as a way of scripting web pages. At first it was considered a poorly designed toy, but over the years has become recognised as a powerful language in its own right, and has been adopted widely beyond the browser.

JavaScript looks like this:

```
function circleArea(radius) {
    return radius * 2 * 3.141592653;
}
```

JavaScript Object Notation

27th April 2022 at 7:09pm

Definitions

JSON is a standard plain-text format used for modelling hierarchical structures of objects that contain named fields. See JSON in TiddlyWiki for an overview.

JeremyRuston

20th September 2017 at 3:11pm

Definitions

I'm the original inventor of TiddlyWiki. You can hire me through Federatial, and find me on these services:

- jeremy (at) jermolene (dot) com
- Jermolene on GitHub
- Jermolene on GitTip, a micropayment service
- @Jermolene on Twitter
- Jermy on LinkedIn
- Jermy on Flickr

Further information:

- An interview with me in The Inquirer by Wendy Grossman
- A hilarious interview with me from British television in 1983
- Here's a video of a presentation I did in 2007 called "How to Start an Open Source Project".

Jermolene

28th August 2013 at 9:02pm

Definitions

Alias for JeremyRuston.

join Operator

13th June 2019 at 5:46pm

Filter Operators

purpose	join a list of strings together with the separator **S**
input	a selection of titles
parameter	**S** = the separator to insert between each of the items
output	the input titles joined together with the separator **S**

Learn more about how to use Filters

ⓘ Introduced in v5.1.20

Examples

join Operator (Examples)

13th June 2019 at 5:47pm

join Operator Operator Examples

```
a b c d e +[join[,]]
```

Try it

JSON in TiddlyWiki

11th June 2022 at 12:47pm

Features

Introduction

JSON (JavaScript Object Notation) is a standardised text representation for data structures that is widely used for the storage and transfer of data.

JSON is used in several different contexts in TiddlyWiki. For example:

- Tiddlers are represented as JSON data within TiddlyWiki HTML files

- Groups of tiddlers can be exported and imported as JSON files
- Plugin tiddlers store their constituent shadow tiddlers as JSON data
- The client-server configuration uses JSON messages to communicate between the client and the server
- Arbitrary JSON data within DataTiddlers can be processed and manipulated using a set of filter operators and action widgets

About JSON

The technical description of JSON at the official website https://json.org/ is terse. Here we summarise the main features.

JSON supports two basic data structures:

Arrays are lists of items. The items are identified by their numeric index (starting at zero)

An example of an array is:

```
["one","two","three\"four"]
```

Note the following features of arrays:

- The array is signified by square brackets surrounding the list of items
- Each item is a string in double quotes. Double quotes can be included within the strings by preceding them with a backslash (\ )
- The items are separated by commas

Objects are collections of name/value pairs. Each item is a value that is identified by a unique name

An example of an object is:

```
{
  "first": "This is the first value",
  "second": "This is the second value",
  "third": "This is the third value"
}
```

Note the following features of objects:

- The object is signified by curly braces surrounding the list of name/value pairs
- Each name/value pair consists of the name in double quotes, a colon, and then the value
- The name/value pairs are separated by commas

The examples above all show string values. JSON actually supports several different types of value. Any of these types can be used as a value:

- String values, as shown above
- Numeric values, represented as signed decimals such as `1`, `3.14`. Exponential notation can also be used e.g. `-1E10`
- Boolean values, represented by the keywords `true` and `false`
- The special value `null`, which is often used to represent data that is missing or incomplete
- Objects and arrays are also values, allowing complex nested structures to be represented

Working with Data Tiddlers

- Reading data from JSON tiddlers
- Constructing JSON tiddlers
- Modifying JSON tiddlers

Json Mangler plugin by Joshua Fontany

6th January 2021 at 4:10pm

Community Plugins

Extend tiddlywiki to parse complex ("nested") json data tiddlers.

Json Mangler introduces a new path syntax for indexes of json data tiddlers , and includes many supporting tools, filters, widgets, etc.

Example Wiki: https://joshuafontany.github.io/TW5-JsonMangler/

Source: https://github.com/joshuafontany/TW5-JsonMangler

jsonextract Operator

11th June 2022 at 12:47pm

Filter Operators JSON Operators

purpose	retrieve the JSON string of a property from JSON strings
input	a selection of JSON strings
parameter	one or more indexes of the property to retrieve
output	the JSON string values of each of the retrieved properties

Learn more about how to use Filters

ⓘ **Introduced in v5.2.4** See JSON in TiddlyWiki for background.

The **jsonextract** operator is used to retrieve values from JSON data as JSON substrings. See also the following related operators:

- **jsonget** to retrieve the values of a property in JSON data
- **jsontype** to retrieve the type of a JSON value
- **jsonindexes** to retrieve the names of the fields of a JSON object, or the indexes of a JSON array

Properties within a JSON object are identified by a sequence of indexes. In the following example, the value at `[a]` is `one`, and the value at `[d][f][0]` is `five`.

```
{
    "a": "one",
    "b": "",
    "c": "three",
    "d": {
        "e": "four",
        "f": [
            "five",
            "six",
            true,
            false,
            null
        ],
        "g": {
            "x": "max",
            "y": "may",
```

```
            "z": "maize"
        }
    }
}
```

The following examples assume that this JSON data is contained in a variable called `jsondata`.

The **`jsonextract`** operator uses multiple parameters to specify the indexes of the property to retrieve. Values are returned as literal JSON strings:

```
[<jsondata>jsonextract[a]] --> "one"
[<jsondata>jsonextract[d],[e]] --> "four"
[<jsondata>jsonextract[d],[f],[0]] --> "five"
[<jsondata>jsonextract[d],[f]] --> ["five","six",true,false,null]
[<jsondata>jsonextract[d],[g]] --> {"x":"max","y":"may","z":"maize"}
```

New in v5.3.2 Negative indexes into an array are counted from the end, so -1 means the last item, -2 the next-to-last item, and so on:

```
[<jsondata>jsonextract[d],[f],[-1]] --> null
[<jsondata>jsonextract[d],[f],[-2]] --> false
[<jsondata>jsonextract[d],[f],[-4]] --> "six"
```

Indexes can be dynamically composed from variables and transclusions:

```
[<jsondata>jsonextract<variable>,{!!field},[0]]
```

A subtlety is that the special case of a single blank parameter is used to identify the root object. Thus:

```
[<jsondata>jsonextract[]] --> {"a":"one","b":"","c":"three","d":{"e":"four","f":
["five","six",true,false,null],"g":{"x":"max","y":"may","z":"maize"}}}
```

jsonget Operator

11th June 2022 at 12:47pm

Filter Operators JSON Operators

purpose	retrieve the value of a property from JSON strings
input	a selection of JSON strings
parameter	one or more indexes of the property to retrieve
output	the values of each of the retrieved properties

Learn more about how to use Filters

Introduced in v5.2.4 See JSON in TiddlyWiki for background.

The **`jsonget`** operator is used to retrieve values from JSON data as strings. See also the following related operators:

- **`jsontype`** to retrieve the type of a JSON value
- **`jsonindexes`** to retrieve the names of the fields of a JSON object, or the indexes of a JSON array
- **`jsonextract`** to retrieve a JSON value as a string of JSON

Properties within a JSON object are identified by a sequence of indexes. In the following example, the value at `[a]` is `one`, and the value at `[d][f][0]` is `five`.

```
{
    "a": "one",
    "b": "",
    "c": "three",
    "d": {
        "e": "four",
        "f": [
            "five",
            "six",
            true,
            false,
            null
        ],
        "g": {
            "x": "max",
            "y": "may",
            "z": "maize"
        }
    }
}
```

The following examples assume that this JSON data is contained in a variable called `jsondata`.

The **`jsonget`** operator uses multiple parameters to specify the indexes of the property to retrieve:

```
[<jsondata>jsonget[a]] --> "one"
[<jsondata>jsonget[d],[e]] --> "four"
[<jsondata>jsonget[d],[f],[0]] --> "five"
```

ⓘ **New in v5.3.2** Negative indexes into an array are counted from the end, so -1 means the last item, -2 the next-to-last item, and so on:

```
[<jsondata>jsonget[d],[f],[-1]] --> null
[<jsondata>jsonget[d],[f],[-2]] --> false
[<jsondata>jsonget[d],[f],[-4]] --> "six"
```

Indexes can be dynamically composed from variables and transclusions:

```
[<jsondata>jsonget<variable>,{!!field},[0]]
```

Boolean values and null are returned as normal strings. The **jsontype** operator can be used to retrieve a string identifying the original type. Thus:

```
[<jsondata>jsontype[a]] --> "string"
[<jsondata>jsontype[d]] --> "object"
[<jsondata>jsontype[d],[f]] --> "array"
[<jsondata>jsontype[d],[f],[2]] --> "boolean"
```

Using the **jsonget** operator to retrieve an object or an array returns a list of the values. For example:

```
[<jsondata>jsonget[d],[f]] --> "five","six","true","false","null"
[<jsondata>jsonget[d],[g]] --> "max","may","maize"
```

The **jsonindexes** operator retrieves the corresponding indexes:

```
[<jsondata>jsonindexes[d],[f]] --> "0", "1", "2", "3", "4"
[<jsondata>jsonindexes[d],[g]] --> "x", "y", "z"
```

If the object or array contains nested child objects or arrays then the values are retrieved recursively and returned flattened into a list. For example:

```
[<jsondata>jsonget[d]] --> "four","five","six","true","false","null","max","may","maize"
```

A subtlety is that the special case of a single blank parameter is used to identify the root object. Thus:

```
[<jsondata>jsonindexes[]] --> "a", "b", "c", "d"
```

jsonindexes Operator

11th June 2022 at 12:47pm

Filter Operators JSON Operators

purpose	retrieve the value of a property from JSON strings
input	a selection of JSON strings
parameter	one or more indexes of the property to retrieve
output	the values of each of the retrieved properties

Learn more about how to use Filters

ⓘ **Introduced in v5.2.4** See JSON in TiddlyWiki for background.

The **jsonindexes** operator is used to retrieve the property names of JSON objects or the index names of JSON arrays. See also the following related operators:

- **jsonget** to retrieve the values of a property in JSON data
- **jsontype** to retrieve the type of a JSON value
- **jsonextract** to retrieve a JSON value as a string of JSON

Properties within a JSON object are identified by a sequence of indexes. In the following example, the value at `[a]` is `one`, and the value at `[d][f][0]` is `five`.

```
{
    "a": "one",
    "b": "",
    "c": "three",
    "d": {
        "e": "four",
        "f": [
            "five",
            "six",
            true,
            false,
            null
        ],
        "g": {
            "x": "max",
```

```
            "y": "may",
            "z": "maize"
        }
    }
}
```

The following examples assume that this JSON data is contained in a variable called `jsondata`.

The **`jsonindexes`** operator uses multiple parameters to specify the indexes of the property to retrieve:

```
[<jsondata>jsonindexes[d],[f]] --> "0", "1", "2", "3", "4"
[<jsondata>jsonindexes[d],[g]] --> "x", "y", "z"
```

Indexes can be dynamically composed from variables and transclusions:

```
[<jsondata>jsonindexes<variable>,{!!field}]
```

Retrieving the indexes of JSON properties that are not objects or arrays will return nothing.

A subtlety is that the special case of a single blank parameter is used to identify the root object. Thus:

```
[<jsondata>jsonindexes[]] --> "a", "b", "c", "d"
```

jsonset Operator

4th December 2023 at 12:52pm

Filter Operators JSON Operators

purpose	set the value of a property in JSON objects
input	a selection of JSON objects
suffix	data type of the value to be assigned to the property
parameter	one or more indexes of the property to modify, if applicable followed by the value to be assigned
output	the JSON objects with the specified value assigned to the specified property

Learn more about how to use Filters

ⓘ **New in v5.3.2** The `jsonset` operator is used to set a property value in JSON strings. See JSON in TiddlyWiki for background. See also the following related operators:

- `jsonget` to retrieve the values of a property in JSON data
- `jsontype` to retrieve the type of a JSON value
- `jsonindexes` to retrieve the names of the fields of a JSON object, or the indexes of a JSON array
- `jsonextract` to retrieve a JSON value as a string of JSON

The type of the value to be assigned to the property can be optionally specified with a suffix:

- **string**: default, the string is specified as the final operand
- **boolean**: the boolean value is true if the final operand is the string "true" and false if the final operand is the string "false", any other value for the final string results prevents the property from being assigned
- **number**: the numeric value is taken from the final operand, invalid numbers are interpreted as zero
- **json**: the JSON string value is taken from the final operand, invalid JSON prevents the property from being assigned
- **object**: an empty object is assigned to the property, the final operand is ignored
- **array**: an empty array is assigned to the property, the final operand is ignored

- **null**: the special value null is assigned to the property, the final operand is ignored

Properties within a JSON object are identified by a sequence of indexes. In the following example, the value at `[a]` is `one`, and the value at `[d][f][0]` is `five`.

```
{
    "a": "one",
    "b": "",
    "c": "three",
    "d": {
        "e": "four",
        "f": [
            "five",
            "six",
            true,
            false,
            null
        ],
        "g": {
            "x": "max",
            "y": "may",
            "z": "maize"
        }
    }
}
```

The **jsonset** operator uses multiple parameters to specify the indexes of the property to set. When used to assign strings (default behaviour if no suffix is specified) the final operand is interpreted as the value to assign.

Negative indexes are counted from the end, so -1 means the last item, -2 the next-to-last item, and so on.

Indexes can be dynamically composed from variables and transclusions, e.g. `[<jsondata>jsonset<variable>,{!!field},[0],{CurrentResult}]`.

In the special case where only a single parameter is defined, the operator replaces the entire input object with the the value of that parameter. If the single parameter is blank, the operation is ignored and no assignment takes place.

If the input consists of multiple JSON objects with matching properties, the value is set for all of them.

Examples

jsonset Operator (Examples)

4th December 2023 at 12:50pm

jsonset Operator Operator Examples

The examples below assume the following JSON object is contained in the variable `object-a`:

```
{
    "a": "one",
    "b": "",
    "c": "three",
    "d": {
```

```
        "e": "four",
        "f": [
            "five",
            "six",
            true,
            false,
            null
        ],
        "g": {
            "x": "max",
            "y": "may",
            "z": "maize"
        }
    }
}
```

```
[<object-a>jsonset[d],[Jaguar]]
```

Try it

```
[<object-a>jsonset[d],[f],[Panther]]
```

Try it

```
[<object-a>jsonset[d],[f],[-1],[Elephant]]
```

Try it

```
[<object-a>jsonset[d],[f],[-2],[Elephant]]
```

Try it

```
[<object-a>jsonset[d],[f],[-4],[Elephant]]
```

Try it

```
[<object-a>jsonset[Panther]]
```

→ If only a single parameter is specified, it replaces the entire JSON object

Try it

`[<object-a>jsonset[]]`

→ If only a single blank parameter is specified, no changes are made to the JSON object

Try it

The examples below assume the following JSON object is contained in the variable `object-b`:

```
{"a":"one","b":"","c":1.618,"d":{"e":"four","f":["five","six",true,false,null]}}
```

`[<object-b>jsonset[]]`

→ If only a single blank parameter is specified, no changes are made to the JSON object

Try it

`[<object-b>jsonset[],[Antelope]]`

→ If the property to be set is blank, the entire JSON object is replaced

Try it

`[<object-b>jsonset:number[],[not a number]]`

→ invalid numbers are interpreted as zero

Try it

`[<object-b>jsonset[id],[Antelope]]`

→ nonexistent top level properties are added to the object

Try it

`[<object-b>jsonset[missing],[id],[Antelope]]`

→ nonexistent nested properties are are ignored

Try it

`[<object-b>jsonset:notatype[id],[Antelope]]`

→ invalid type suffix is interpreted as the default string type

Try it

```
[<object-b>jsonset:boolean[id],[false]]
```

Try it

```
[<object-b>jsonset:boolean[id],[Antelope]]
```

→ invalid boolean value causes no assignment to be made

Try it

```
[<object-b>jsonset:number[id],[42]]
```

Try it

```
[<object-b>jsonset:null[id]]
```

Try it

```
[<object-b>jsonset:array[d],[f],[5]]
```

Try it

`[<object-b>jsonset:object[d],[f],[5]]`

Try it

`[<object-a>] [<object-b>] :and[jsonset[b],[two]]`

→ If the input consists of multiple JSON objects with matching properties, the value is set for all of them

Try it

jsonstringify Operator

22nd September 2023 at 2:14pm

Filter Operators String Operators

purpose	apply JSON string encoding to a string, see also the similar `stringify`
input	a selection of titles
suffix	`R` = ⓘ **Introduced in v5.1.23** optionally, the keyword `rawunicode`
output	the input with JSON string encodings applied

Learn more about how to use Filters

ⓘ Introduced in v5.1.14

The following substitutions are made:

Character	Replacement	Condition
`\`	`\\`	Always
`"`	`\"`	Always
Carriage return (0x0d)	`\r`	Always
Line feed (0x0a)	`\n`	Always
Backspace (0x08)	`\b`	Always
Form field (0x0c)	`\f`	Always
Tab (0x09)	`\t`	Always
Characters from 0x00 to 0x1f, except listed above	`\u####` where #### is four hex digits	Always
Characters from from 0x80 to 0xffff	`\u####` where #### is four hex digits	If `rawunicode` suffix is not present (default)
Characters from 0x80 to 0xffff	**ⓘ Introduced in v5.1.23** Unchanged	If `rawunicode` suffix is present

ⓘ Introduced in v5.1.23 If the suffix `rawunicode` is present, Unicode characters above 0x80 (such as ß, ä, ñ or 🎄) will be passed through unchanged. Without the suffix, they will be substituted with `\u` codes, which was the default behavior before 5.1.23. Characters outside the Basic Multilingual Plane, such as 🎄 and other emojis, will be encoded as a UTF-16 surrogate pair, i.e. with two `\u` sequences.

Note

> Mind the differences compared to **`stringify`** in encoding of single quotes and control characters (0x00 to 0x1f).

Examples

jsonstringify Operator (Examples)

22nd September 2023 at 2:23pm

jsonstringify Operator Operator Examples

Compare the encoding of quotes and control characters in the first example with the analogue example for the stringify operator.

```
[[Backslash \, double quote ", single quote ', tab	, line feed
]] +[jsonstringify[]]
```

Try it

```
[[Accents and emojis -> äñøßπ ⏳🎄🍪🍓 without suffix]] +[jsonstringify[]]
```

Try it

```
[[Accents and emojis -> äñøßπ ⏳🎄🍪🍓 with rawunicode suffix]] +[jsonstringify:rawunicode[]]
```

Try it

jsontiddler Macro

27th April 2022 at 7:12pm

Core Macros Macros

The **jsontiddler** macro returns the fields of a single tiddler in JSON form. See JSON in TiddlyWiki for an overview.

Parameters

title

The title of a tiddler

JSONTiddlers

27th April 2022 at 7:12pm

Concepts

A JSON tiddler is a data tiddler containing a JSON structure in its `text` field. See JSON in TiddlyWiki for an overview.

Its ContentType is `application/json`.

The history list is an example of a JSON tiddler.

jsontiddlers Macro

27th April 2022 at 7:11pm

Core Macros Macros

The **jsontiddlers** macro returns the fields of a selection of tiddlers in JSON form. See JSON in TiddlyWiki for an overview.

An example can be seen in the **template tiddler for JSON exports**.

Parameters

filter
: A filter selecting which tiddlers to include

spaces
: An optional number of spaces to use for formatting the JSON. Defaults to 4, with blank or zero resulting in packed JSON with no formatting spaces

JSONTiddlerWidget

27th April 2022 at 7:11pm

Widgets

Introduction

The jsontiddler widget renders the fields of a tiddler to display as a JSON object. The fields of the tiddler can be customised or excluded.

The jsontiddler widget is used in system templates to generate JSON representations of tiddlers for saving. See JSON in TiddlyWiki for an overview.

Content and Attributes

The content of the `<$jsontiddler>` widget is ignored.

Attribute	Description
tiddler	Optional title of the tiddler from which the fields are to be displayed
exclude	Optional space separated list of fields to be excluded
escapeUnsafeScriptChars	If set to "yes" then the `<` character will be escaped to the equivalent `\u003C` encoding
{any attributes starting with $}	Each attribute name (without the leading `$`) specifies a field to be modified with the attribute value providing the value to assign to the field

jsontype Operator

11th June 2022 at 12:47pm

Filter Operators JSON Operators

purpose	retrieve the type of a property from JSON strings
input	a selection of JSON strings
parameter	one or more indexes of the property whose type is to be retrieved
output	the types of each of the retrieved properties

Learn more about how to use Filters

ⓘ **Introduced in v5.2.4** See JSON in TiddlyWiki for background.

The **jsontype** operator is used to retrieve the type of a property in JSON data. See also the following related operators:

- **jsonget** to retrieve the values of a property in JSON data
- **jsonindexes** to retrieve the names of the fields of a JSON object, or the indexes of a JSON array
- **jsonextract** to retrieve a JSON value as a string of JSON

JSON supports the following data types:

- **string** - a Unicode string
- **number** - a floating point number
- **boolean** - Boolean value (true or false)
- **array** - an array of values
- **object** - an object of name/value pairs
- **null** - a special type representing a missing value

Properties within a JSON object are identified by a sequence of indexes. In the following example, the value at `[a]` is `one`, and the value at `[d][f][0]` is `five`.

```
{
    "a": "one",
    "b": "",
    "c": "three",
    "d": {
        "e": "four",
        "f": [
```

```
            "five",
            "six",
            true,
            false,
            null
        ],
        "g": {
            "x": "max",
            "y": "may",
            "z": "maize"
        }
    }
}
```

The following examples assume that this JSON data is contained in a variable called `jsondata`.

The **`jsontype`** operator uses multiple parameters to specify the indexes of the property whose type is to be retrieved:

```
[<jsondata>jsontype[a]] --> "string"
[<jsondata>jsontype[d]] --> "object"
[<jsondata>jsontype[d],[f]] --> "array"
[<jsondata>jsontype[d],[f],[2]] --> "boolean"
```

ⓘ **New in v5.3.2** Negative indexes into an array are counted from the end, so -1 means the last item, -2 the next-to-last item, and so on:

```
[<jsondata>jsontype[d],[f],[-1]] --> "null"
```

```
[<jsondata>jsontype[d],[f],[-2]] --> "boolean"
[<jsondata>jsontype[d],[f],[-4]] --> "string"
```

Indexes can be dynamically composed from variables and transclusions:

```
[<jsondata>jsontype<variable>,{!!field},[0]]
```

A subtlety is that the special case of a single blank parameter is used to identify the root object. Thus:

```
[<jsondata>jsontype[]] --> "object"
```

JSZip Plugin

19th October 2020 at 10:48am

OfficialPlugins

This plugin provides primitives for creating Zip files in the browser. It also makes the JSZip library available for use by other plugins.

KaTeX Plugin

28th February 2017 at 11:25am

OfficialPlugins Plugin Editions

This plugin adds the ability to display mathematical notation written in LaTeX.

See https://tiddlywiki.com/plugins/tiddlywiki/katex

kdi-demo-configtid

23rd November 2021 at 4:44am

KeyboardDrivenInput

Keyboard Codes

25th May 2021 at 12:37pm

KeyboardWidget

This is a list of all the key codes supported by the KeyboardWidget:

`cancel`, `help`, `backspace`, `tab`, `clear`, `return`, `enter`, `pause`, `escape`, `space`, `page_up`, `page_down`, `end`, `home`,
`left`, `up`, `right`, `down`, `printscreen`, `insert`, `delete`

`0`, `1`, `2`, `3`, `4`, `5`, `6`, `7`, `8`, `9`

`a`, `b`, `c`, `d`, `e`, `f`, `g`, `h`, `i`, `j`, `k`, `l`, `m`, `n`, `o`, `p`, `q`, `r`, `s`, `t`, `u`, `v`, `w`, `x`, `y`, `z`

`numpad0`, `numpad1`, `numpad2`, `numpad3`, `numpad4`, `numpad5`, `numpad6`, `numpad7`, `numpad8`, `numpad9`

`multiply`, `add`, `separator`, `subtract`, `decimal`, `divide`

`f1`, `f2`, `f3`, `f4`, `f5`, `f6`, `f7`, `f8`, `f9`, `f10`, `f11`, `f12`
`f13`, `f14`, `f15`, `f16`, `f17`, `f18`, `f19`, `f20`, `f21`, `f22`, `f23`, `f24`

`firefoxsemicolon`, `firefoxequals`, `firefoxminus`

`semicolon` , `equals` , `comma` , `dash` , `period` , `slash` , `backquote` , `openbracket` , `backslash` , `closebracket` , `quote`

`shift` , `ctrl` , `alt` , `meta`

Keyboard Shortcut Descriptor

Definitions

`Keyboard Shortcut Descriptors` are used by the KeyboardWidget and within Tiddlers tagged $:/tags/KeyboardShortcut to link actions with a keyboard combination

A `Keyboard Shortcut Descriptor` has the following format:

```
ctrl+enter
ctrl+shift+alt+A
alt-shift-T
alt-Space
```

It can have multiple Key combinations:

```
ctrl+enter alt+shift+A ctrl+alt+9 alt-Space ctrl-Backspace
```

There are **two ways** Keyboard Shortcut Descriptors can be used:

- directly
  - passing the attribute `key="ctrl-enter"` to the KeyboardWidget

    - setting `ctrl-enter` as the **value** of the `key` **field** in a Tiddler tagged $:/tags/KeyboardShortcut
- with a **reference** to a configuration Tiddler
    - A **configuration Tiddler** for Keyboard Shortcuts follows some naming rules:
        - The title starts with `$:/config/`
        - Followed by either the string `shortcuts` or one of the following **platform descriptors**:
            - **`shortcuts-linux`**
            - **`shortcuts-not-linux`**
            - **`shortcuts-mac`**
            - **`shortcuts-not-mac`**
            - **`shortcuts-windows`**
            - **`shortcuts-not-windows`**
        - The title ends with another `/` followed by a suffix that can be chosen freely
            - This suffix is the **identifier** for the `Keyboard Shortcut Descriptor` (A)
    - The **reference** to the configuration Tiddler is made of `((` + the **identifier** (A) + `))`
        - Example: `((my-shortcut))` is the **reference** to these configuration Tiddlers:
            - *`$:/config/shortcuts/my-shortcut`*
            - *`$:/config/shortcuts-linux/my-shortcut`*
            - *`$:/config/shortcuts-not-linux/my-shortcut`*
            - *`$:/config/shortcuts-mac/my-shortcut`*
            - *`$:/config/shortcuts-not-mac/my-shortcut`*
            - *`$:/config/shortcuts-windows/my-shortcut`*
            - *`$:/config/shortcuts-not-windows/my-shortcut`*

- Finally, the referenced configuration Tiddler stores the Keyboard Shortcut Descriptor in its **text** field

Keyboard Shortcut Tiddler

Concepts

A **Keyboard Shortcut Tiddler** is made of three parts:

- The **Tag** $:/tags/KeyboardShortcut
- The **field** `key` with a Keyboard Shortcut Descriptor as its **value**
- Actions in its **text** field



Introduced in v5.2.4 By default **`$keyboard`** and text editor shortcuts take priority, which can be circumvented by setting the **field** `priority` to `yes`.

If the Keyboard Shortcut Descriptor has the form `((my-shortcut))` it's a **reference** to a **configuration Tiddler** that stores the corresponding Keyboard Shortcut

In order to make a **shortcut** editable through the **Keyboard Shortcuts** Tab in the **$:/ControlPanel** it's sufficient to create a tiddler `$:/config/ShortcutInfo/my-shortcut`, where the **suffix** is the **reference** used for the Keyboard Shortcut

Notes on wiki navigation

If you want to create keyboard shortcuts for navigation, there are two things to keep in mind:

- If your shortcut uses **Ctrl**, you need to include `$scroll="yes"` in the ActionNavigateWidget's attributes otherwise the action will be ignored.
- The actions need to be wrapped in NavigatorWidget like in this **New Tiddler keyboard shortcut**.

keyboard-driven-input Macro

Core Macros Macros

The **keyboard-driven-input** macro generates an input field or textarea that lets you cycle through a given list of entries with the `Up` and `Down` keys. Doing so, an entry gets selected and can be processed with further actions

Parameters

To create the input field or textarea, the **keyboard-driven-input** macro accepts all parameters of the EditTextWidget

The additional parameters are:

parameter	purpose
storeTitle	the title of the tiddler that stores the user input
selectionStateTitle	the title of the tiddler that stores the selected entry with a -primaryList or -secondaryList suffix to make it unique
inputAcceptActions	the actions that get processed when the user hits `Enter`
inputAcceptVariantActions	the actions that get processed when the user hits `ctrl-Enter`
inputCancelActions	the actions that get processed when the user hits `Escape`
configTiddlerFilter	a **filter** that specifies the tiddler that stores the first item-filter in its **`first-search-filter`** field and the second

parameter	purpose
	item-filter in its **`second-search-filter`** field
firstSearchFilterField	the field of the configTiddler where the first search-filter is stored. Defaults to **`first-search-filter`**
secondSearchFilterField	the field of the configTiddler where the second search-filter is stored. Defaults to **`second-search-filter`**
filterMinLength	the minimum length of the user input after which items are filtered

See Demonstration: keyboard-driven-input Macro for further guidance on using this macro.

KeyboardShortcuts

10th June 2016 at 10:21am

Features Working with TiddlyWiki

Keyboard shortcuts can either be used with the **`$keyboard`** Widget or with Keyboard Shortcut Tiddlers

> The **`$keyboard`** Widget makes the shortcuts accessible within an `input` or `textarea` field.
> A shortcut defined through a Keyboard Shortcut Tiddler makes the shortcut accessible globally

See How to create keyboard shortcuts with a detailed explanation for creating your own shortcuts

Keyboard shortcuts are available for common editing operations within the Text Editor:

- Confirming changes to the draft tiddler containing the keyboard focus (defaults to `ctrl-Enter`)
- Abandoning changes to the draft tiddler containing the keyboard focus (defaults to `escape`)
- Formatting operations from the tiddler editing toolbar (see the tooltips)

ⓘ **Introduced in v5.1.18** : New **global** Keyboard shortcuts:

Action	Default Shortcut
Creating a new tiddler	`alt-N`
Creating a new journal	`alt-J`
Creating a new image	`alt-I`
Focusing sidebar search	ⓘ **Introduced in v5.1.20** `ctrl-shift-F`
Toggling the sidebar	ⓘ **Introduced in v5.1.20** `shift-alt-S`
Advanced search	ⓘ **Introduced in v5.1.20** `ctrl-shift-A`

The current shortcuts can be inspected and customised in the "Keyboard Shortcuts" tab of the **Control Panel** ⚙.

Special Keys

adding tags in the editor

ⓘ **Introduced in v5.1.14** : if you hit `Enter` in the new tag input, the tag will be added

KeyboardWidget

9th October 2021 at 2:12pm

TriggeringWidgets Widgets

Introduction

The keyboard widget allows ActionWidgets to be triggered by specific key combinations. There is also a shorthand for generating Messages via the `message` and `param` attributes.

Content and Attributes

The content of the `<$keyboard>` widget is rendered normally. The keyboard shortcuts only take effect when the focus is within the contained content.

Attribute	Description
actions	A string containing ActionWidgets to be triggered when the key combination is detected ⓘ **Introduced in v5.2.0** The variables: **event-key**, **event-code** and **modifier** are available for the actions
message	The title of the WidgetMessage to generate
param	The parameter to be passed with the WidgetMessage
key	Key string identifying the key(s) to be trapped (see below)
class	A CSS class to be assigned to the generated HTML DIV element
tag	ⓘ **Introduced in v5.1.14** The html element the widget creates to capture the keyboard event, defaults to: » `span` when parsed in inline-mode » `div` when parsed in block-mode

Variables

ⓘ **Introduced in v5.2.0** The following variables are made available to the actions:

Variables	Description
`event-key`	The **`event-key`** variable contains the character, if possible. eg: `1`. You can experiment with some settings at: *Key Codes (Example)*
`event-code`	The **`event-code`** variable contains a character description. eg: `Digit1` instead of `1`. Or `Space` instead of an empty string ` `, which is hard to see
`event-key-descriptor`	The **`event-key-descriptor`** variable is available if the keyboard event captured was configured using a keyboard shortcut descriptor of the form `((my-shortcut))` which references a configuration tiddler.
`modifier`	The modifier Variable contains the Modifier Key held during the event (can be `normal`, `ctrl`, `shift`, `alt` or combinations thereof)

Key Strings

Key strings are made up of one or more key specifiers separated by spaces. Each key specifier is zero or more of the modifiers `alt`, `shift`, `ctrl` or `meta` followed by the name of a key, all joined with "+" plus or "-" minus symbols. Key names are either the letter or digit printed on the key (eg "a" or "1"), or one of the special keys `backspace`, `tab`, `enter` or `escape`.

TypeError: setting getter-only property "parentNode"

For example:

```
A
shift+A
shift+escape
ctrl+enter
```

```
ctrl+shift+alt+A
```

More Info

- Keyboard Codes

Kill the Dragon

15th November 2021 at 2:32am

task TaskManagementExample

This is a sample task for the TaskManagementExample

Kin filter operator by bimlas

6th January 2021 at 4:10pm

Community Plugins

The purpose of the kin operator with examples:

- Finds related tags, related tiddlers in any depth
- Finds out where base tiddler originates and what other elements originate from it
- Finds the ancestors and successors of a family member
- Finds the "leaves" of the branch of the base tiddler in a tree-like structure (where the base tiddler is a leaf)
- Finds the super- and subsets / groups of a mathematical set (where the base tiddler is a set)

https://bimlas.github.io/tw5-kin-filter/

Kookma Plugin Library by Mohammad

17th November 2020 at 5:08pm

Community Plugins

This library contains most useful plugins developed under the name Kookma for Tiddlywiki 5.

https://kookma.github.io/TW-PluginLibrary/

It has two parts

The plugin library introduces a very simple mechanism like Tiddlywiki Official Plugin Library to let select among the published plugins and install any number of them you like.

Korean (Korea Republic) Edition

24th April 2016 at 1:33pm

Languages

티들리위키 한국어 번역은 다음에서 사용할 수 있습니다:

- **빈 위키** : https://tiddlywiki.com/languages/ko-KR/empty.html

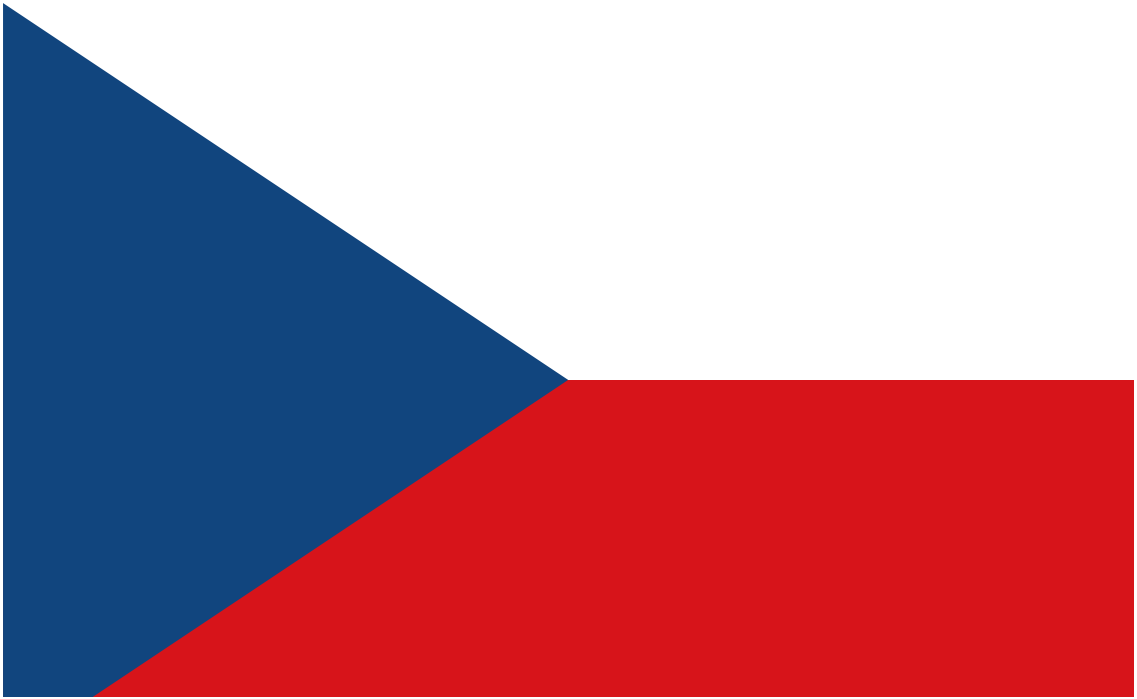
Language Icon: ca-ES

Icon Language



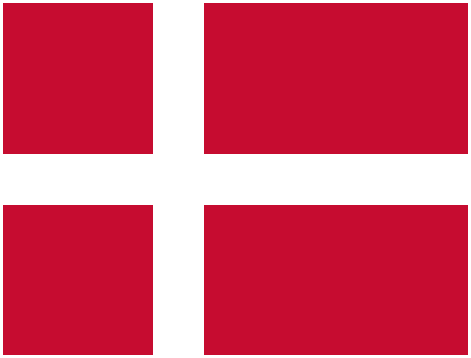
Language Icon: cs-CZ

Icon	Language



Language Icon: da-DK

Icon Language



Language Icon: de-AT

Icon Language



Language Icon: de-DE

Icon Language



Language Icon: el-GR

Icon Language



Language Icon: en-US

Icon Language



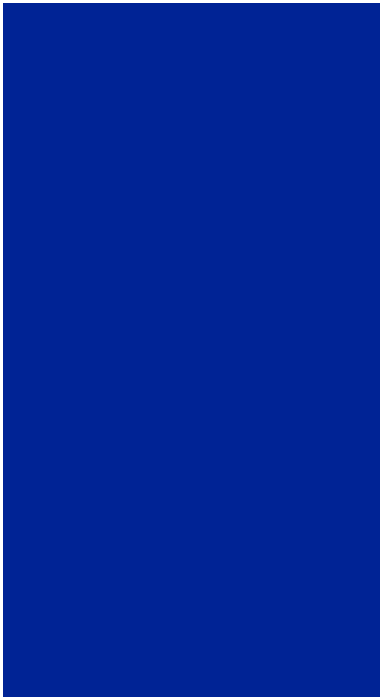
Language Icon: es-ES

Icon Language



Language Icon: fr-FR

Icon Language





Language Icon: hi-IN

Icon Language



Language Icon: ia-IA

Icon	Language



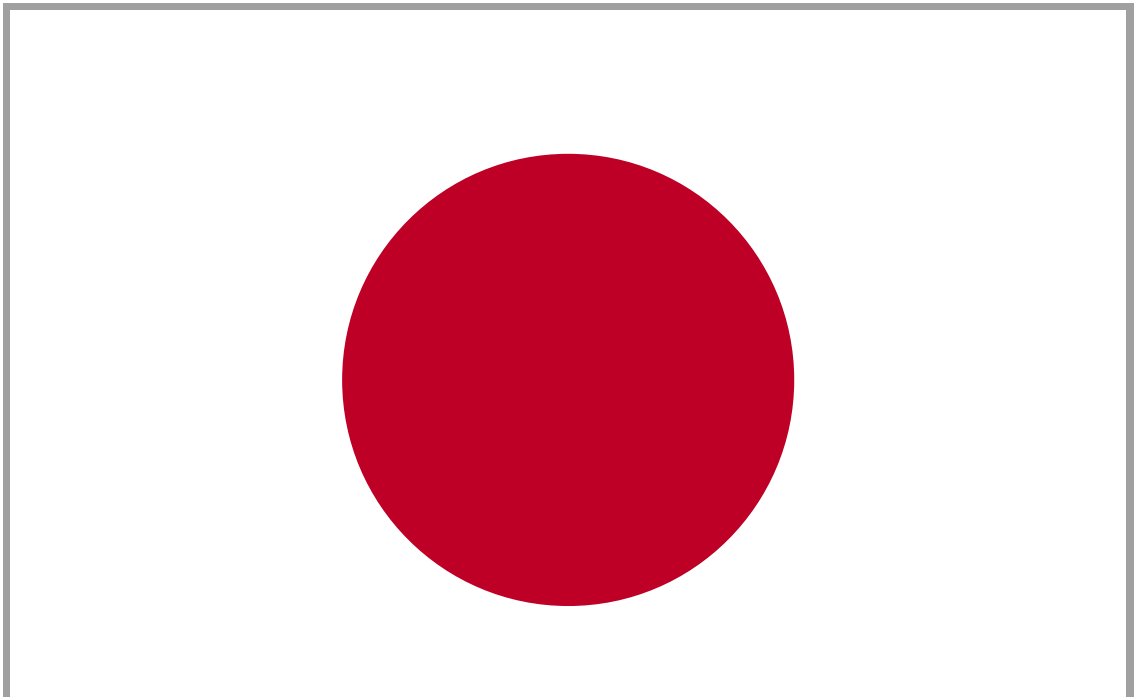
Language Icon: it-IT

Icon Language



Language Icon: ja-JP

Icon Language



Language Icon: ko-KR

Icon Language



Language Icon: nl-NL

Icon Language



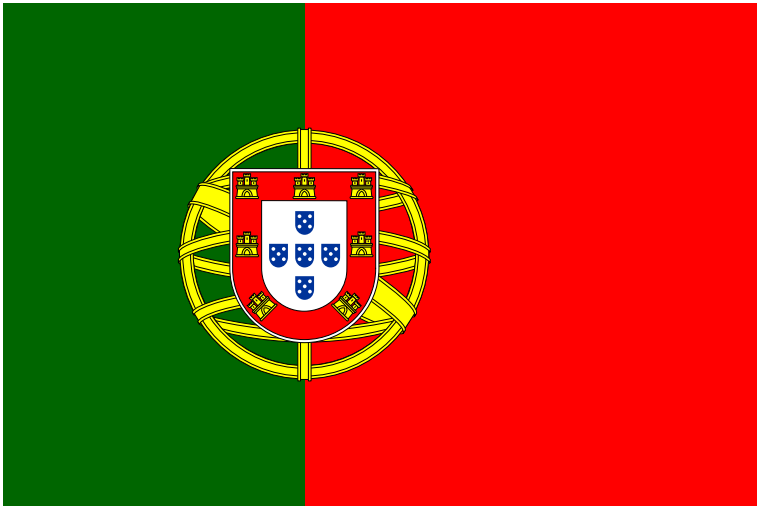
Language Icon: pa-IN

Icon	Language



Language Icon: pt-PT

Icon Language



Language Icon: ru-RU

Icon Language



Language Icon: sk-SK

Icon Language



Language Icon: sv-SE

Icon Language



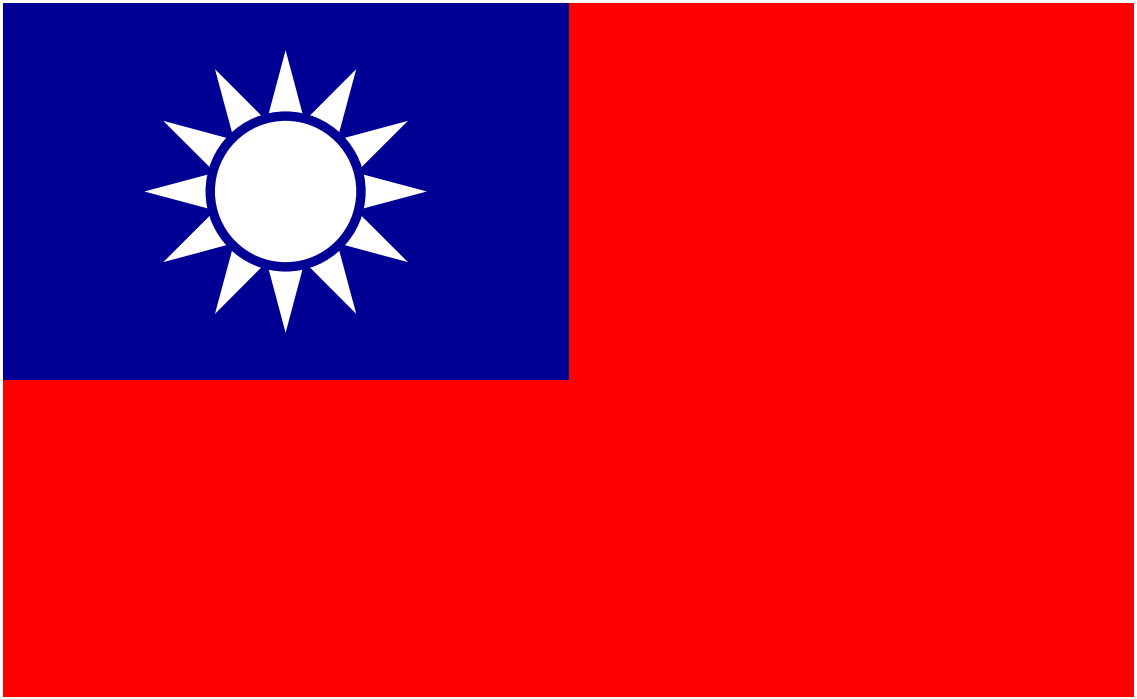
Language Icon: zh-Hans

Icon Language



Language Icon: zh-Hant

Icon Language



LanguageGallery Example

17th November 2021 at 10:27pm

Languages

Here is an example of using the ListWidget and the TranscludeWidget to show a grid of all of Language Icons which are tiddlers tagged Language and Icon



Languages

17th June 2016 at 12:12pm

TableOfContents

Language Plugins

The core TiddlyWiki user interface is available in over 20 languages. You can install language plugins following the instructions Installing a plugin from the plugin library. You can change the current language using the **language** button, found on the "Tools" tab in the sidebar.

You can contribute a new language to the library by learning how to translate TiddlyWiki into your language.

Language Editions

For some languages, there are pre-built editions with additional translated documentation:

- 中文 (简体)
- 中文 (正體)
- Castellano (España)
- Deutsch (Deutschland)
- Deutsch (Österreich)
- Français (France)
- 日本語 (日本国)
- 한국어 (대한민국)
- LanguageGallery Example

last Operator

3rd February 2015 at 8:20pm

Filter Operators Order Operators

purpose	select the last **N** input titles
input	a selection of titles
parameter	**N** = an integer, defaulting to 1
output	the last **N** input titles

Learn more about how to use Filters

Examples

last Operator (Examples)

19th January 2015 at 7:21pm

last Operator Operator Examples

These examples make use of the Days of the Week tiddler.

```
[list[Days of the Week]last[]]
```

Try it

```
[list[Days of the Week]last[5]]
```

Try it

```
[tag[Concepts]!sort[title]last[3]]
```

Try it

Latest

19th September 2014 at 7:05pm

Community

The latest news, articles, resources and examples.

"TiddlyWiki Posts" by Jeffrey Kishner

Posted 7 months ago

A collection of articles covering integration with Fargo, Font Awesome and Google Calendar, and tips for managing task lists. The original site is missing, but a link to an archive is provided.

https://web.archive.org/web/20221015011644/http://blog.jeffreykishner.com/tiddlywiki/

> I have become a regular user of TiddlyWiki and have become so accustomed to using Font Awesome icons in Fargo that I wanted to incorporate the icons into my wikis as well.

GitHub Saver Tutorial by Mohammad

Posted 7 months ago

GitHub Saver is a step by step tutorial that shows how to integrate Tiddlywiki 5 and GitHub Pages to create websites hosted on https://github.com/.

https://kookma.github.io/TW5-GitHub-Saver/

This instruction is based on Tiddlywiki single html file model, while it can use subfolder for extra materials like images, audios, videos, pdfs,... in separate folders.

Other tutorials

Tiddlywiki, Travis-CI and GitHub Pages
https://kookma.github.io/Tiddlywiki-Travis-CI/
This wiki shows how to set up websites hosted on GitHub Pages using Travis-CI and Tiddlywiki 5 on Node.js.

Tiddlywiki and GitHub Pages
https://kookma.github.io/Tiddlywiki-and-GitHub-Pages/
This instruction is based on local edit, save and push to GitHub. It does NOT use the new GitHub Saver mechanism (requires TW 5.1.20+) which lets edit and save directly from Tiddlywiki!

"TW5 Magick" by Stephan Hradek

Posted 2 years ago

Stephan Hradek's growing catalogue of tips, tutorials, and other resources. Stephan has also developed several plugins and macros that you can find at http://tiddlystuff.tiddlyspot.com/.

http://tw5magick.tiddlyspot.com/

> Some things in TiddlyWiki seem like Magic. As my preferred nick is Skeeve, I felt that TW5 Magick is a proper name for this collection of some of the "magic" tricks one can do with TiddlyWiki.

"Reveal.js" by Devin Weaver

Posted 2 years ago

Embed Reveal.js presentations inside your TiddlyWiki.

https://sukima.github.io/tiddlywiki-reveal-js/

> reveal.js enables you to create beautiful interactive slide decks using HTML.
>
> With the power of TiddlyWiki and Reveal.js one can…
>
> - …make an interactive tiddler with fancy animations
> - …Make it go full screen for professional keynote
> - …Distribute the slideshow with examples and additional resources in one HTML file

TW-Scripts by Mohammad

Posted 3 years ago

TW-Scripts is one of the most comprehensive collections of solutions for Tiddlywiki 5.

https://kookma.github.io/TW-Scripts/

TW-Scripts includes:

- Collected solutions to questions in Tiddlywiki Google groups
- Learn through examples
- Templates, stylesheets
- Wikitext, macros, and snippets
- Tips and tricks
- Search tools

TiddlyWiki Video Tutorials by Francis Meetze

Posted 3 years ago

An extensive tutorial for getting started with TiddlyWiki. Recommended

https://www.youtube.com/playlist?list=PLzZCajspPU_UjFn0uy-J9URz0LP4zhxRK

> This tutorial is aimed at giving you a basic foundation so that you can start using TiddlyWiki right away. Here we take a look at the technology behind Tiddlywiki, how to install it and create your first Tiddler.

RegExp in Tiddlywiki by Mohammad

Posted 3 years ago

RegExp in Tiddlywiki contains practical use cases of regular expression in Tiddlywiki.

http://tw-regexp.tiddlyspot.com/

The regular expression is a way to describe complex search patterns using sequences of characters .

RegExp in Tiddlywiki has four main parts

- Using regular expression to match a pattern in tiddler title
- Using regular expression to match a pattern in tiddler fields excluding tiddler text (body)
- Using regular expression to match a pattern in tiddler body (text field)
- Using regular expression for validation like username, password, etc.

"TB5 - a pocket full of tips" by Tobias Beer

Posted 3 years ago

An enormous collection of tips beautifully curated by Tobias Beer. It is actually a collection of sites, with sub-sites dedicated to topics such as filtering (http://tobibeer.github.io/tw/filters/).

http://tobibeer.github.io/tb5/

> A collection of references and sandbox for testing concepts around TiddlyWiki 5...

LazyLoading

9th October 2021 at 4:54pm

Features

Ordinarily with TiddlyWiki, the full content of all tiddlers is embedded into the main HTML file. Lazy loading refers to the technique of only embedding metadata about the tiddler (in other words all fields except the **text** field), and requesting the body from the server when required.

Lazy loading can be used in two configurations:

- When running TiddlyWiki on Node.js, just image tiddlers or all non-system tiddlers can be subject to lazy loading
- When running TiddlyWiki in the Sky for TiddlyWeb, all tiddlers are subject to lazy loading

See the Lazy Loading Mechanism for details of how lazy loading is implemented.

Lazy loading under Node.js

To start TiddlyWiki with lazy loading for image tiddlers use this command:

```
tiddlywiki --listen root-tiddler=$:/core/save/lazy-images
```

To apply lazy loading to all non-system tiddlers use this command:

```
tiddlywiki --listen root-tiddler=$:/core/save/lazy-all
```

Leaflet maps plugin by Sylvain Comte

6th January 2021 at 4:10pm

Community Plugins

A mapping plugin based on Leaflet library. Allows you to create maps based upon geographical data contained in tiddlers.

http://sycom.github.io/TiddlyWiki-Plugins/#Leaflet%20plugin

> The leaflet plugin is a (working) attempt to integrate the leaflet library in TiddlyWiki in order to display geographical purpose tiddlers.
>
> For now `<$leafmap>` widget displays an interactive map. Select size, location and zoom, clustering distance, and background. You can display data : geojson, point(s), polygon(s) and/or polyline(s) directly or calling *GeoTiddler*(s). Tiddler can be called individually, by list or with a filter.
>
> Have a look at demo page on the web.

Learning

19th September 2014 at 6:17pm

TableOfContents

Learn more about using TiddlyWiki:

- Videos
- Introduction to filter notation
- Sharing a TiddlyWiki on Dropbox
- Sharing your tiddlers with others
- Copying tiddlers between TiddlyWiki files
- Creating SubStories
- Editing Tiddlers with Emacs
- Making curved text with SVG
- TaskManagementExample
- Adding a Twitter Follow button
- Philosophy of Tiddlers
- Adopt a Titles Policy
- Concatenating text and variables using macro substitution
- Constructing JSON tiddlers
- Demonstration: keyboard-driven-input Macro
- Editing Tiddlers with Vim
- Formatting List Results as Tables (no CSS)
- Formatting List Results as Tables with CSS - Specified Columns Methods
- Formatting List Results as Tables with CSS - Variable Column Method
- How to add a banner for GitHub contributions
- How to add a new tab to the sidebar
- How to change the sort order of sub-branches in a TOC macro

- How to create a custom story tiddler template
- How to create a custom tiddler colour rule
- How to create a custom tiddler icon rule
- How to create keyboard shortcuts
- How to Customize TiddlyDesktop
- How to embed PDF and other documents
- How to export tiddlers
- How to hide the author's and other fields with CSS
- How to put the last modification date in a banner
- How to turn off camel case linking
- How to widen tiddlers (aka storyriver)
- Icon Gallery
- Introduction to Lists
- Modifying JSON tiddlers
- Reading data from JSON tiddlers
- Simple ways to write protect tiddlers
- TaskManagementExample (Draggable)
- TaskManagementExampleDraggableTemplate
- Visible Transclusions

Also see the complete Reference, including advanced WikiText, macros, widgets, filters etc.

Legacy

19th March 2023 at 2:22pm

Definitions

> Legacy code is old computer source code that is no longer supported on the standard hardware and environments, and is a codebase that is in some respect obsolete or supporting something obsolete. Legacy code may be written in programming languages, use frameworks and external libraries, or use architecture and patterns that are no longer considered modern, increasing the mental burden and ramp-up time for software engineers who work on the codebase.
>
> https://en.wikipedia.org/wiki/Legacy_system

length Operator

13th June 2019 at 5:33pm

Filter Operators

purpose	returns the number of characters of each item in the list
input	a selection of titles
output	replaces each input title with its length as an integer

Learn more about how to use Filters

ⓘ Introduced in v5.1.20

Examples

length Operator (Examples)

13th June 2019 at 5:34pm

length Operator Operator Examples

```
[[abc]length[]]
```

Try it

```
[tag[HelloThere]length[]]
```

Try it

LetWidget

1st October 2022 at 11:46am

Widgets

Introduction

Introduced in v5.2.1 The **`$let`** widget allows multiple variables to be set in one operation. In some situations it can result in simpler code

than using the more flexible **$set** widget. It differs from the **$vars** widget in that variables you're defining may depend on earlier variables defined within the same **$let**.

Content and Attributes

The content of the **$let** widget is the scope for the value assigned to the variable.

Attribute	Description
{attributes}	Each attribute name specifies a variable name. The attribute value is assigned to the variable

Attributes are evaluated in the order they are written. Attributes with the same name are allowed. Each time a duplicate attribute is encountered, it will replace the existing value set by the earlier duplicate.

Note

> ⓘ **Introduced in v5.2.4** There is no longer any restriction on using variable names that start with the $ character.

Examples

Consider a case where you need to set multiple variables, where some depend on the evaluation of others.

Using the **$let** widget, this situation may be handled in the following way:

```
\define helloworld() Hello world!
```

```
<$let target="MyTiddler" currentTiddler={{{ [<target>prefix[$:/settings/for/]] }}} settings={{!!text}}
currentTiddler=<<target>> >
  The settings for <<currentTiddler>> are: <<settings>>
</$let>
```

In contrast, here is the same example using the **$set** widget:

```
<$set name="target" value="MyTiddler" >
<$set name="currentTiddler" value={{{ [<target>prefix[$:/settings/for/]] }}} >
<$set name="settings" value={{!!text}} >
<$set name="currentTiddler" value=<<target>> >
  The settings for <<currentTiddler>> are: <<settings>>
</$set>
</$set>
</$set>
</$set>
```

Remarks

This widget differs from **$vars** in the following way:

- Each variable's definition will be immediately available to all proceeding variables in the same let widget. This differs from vars, in which definitions which depend on some variable will always look to the widget's outer scope for a value.

This widget differs from **$set** in the following ways:

- A fallback (also known as "emptyValue") cannot be specified
- Filters cannot be used to produce a conditional variable assignment
- Variable names must be literal strings

levenshtein Operator

4th March 2023 at 7:16pm

Filter Operators String Operators

purpose	determine the Levenshtein distance of the input title(s) and a given string
input	a selection of titles
parameter	**S** = a string
output	the Levenshtein distance between the input title(s) and **S**

Learn more about how to use Filters

ⓘ Introduced in v5.2.6

The Levenshtein distance is a metric for measuring the difference between two strings. Informally, the Levenshtein distance between two strings is the *minimum* number of single-character edits required to change one string into the other.

Examples

levenshtein Operator (Examples)

4th March 2023 at 7:31pm

levenshtein Operator Operator Examples

Determine the Levenshtein distance between two words:

```
[[motel]levenshtein[money]]
```

Try it

List the 10 tiddler titles with the smallest Levenstein distance to "TiddlyWiki":

copy to clipboard

```
<ul>
<$list filter="[all[tiddlers]!is[system]] :sort:number[levenshtein[TiddlyWiki]] :and[first[10]]">
<li>
<$link /> (<$text text={{{ [all[current]levenshtein[TiddlyWiki]] }}} />)
</li>
</$list>
</ul>
```

That renders as:

- TiddlyWiki (0)
- TiddlyWiki5 (1)
- TiddlyWeb (3)

- TiddlyFox (4)
- Tiddlyhost (4)
- TiddlyIE (4)
- TiddlyWiki.mp3 (4)
- TiddlerLinks (5)
- Tiddlers (5)
- TiddlySpace (5)

License

27th January 2017 at 11:26pm

About

TiddlyWiki is published under a permissive BSD 3-Clause License stored in the shadow tiddler **$:/core/copyright.txt**:

```
TiddlyWiki created by Jeremy Ruston, (jeremy [at] jermolene [dot] com)

Copyright (c) 2004-2007, Jeremy Ruston
Copyright (c) 2007-2023, UnaMesa Association
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this
```

```
  list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimer in the documentation
  and/or other materials provided with the distribution.

* Neither the name of the copyright holder nor the names of its
  contributors may be used to endorse or promote products derived from
  this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS 'AS IS'
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

In layman's terms, the license says that you can take TiddlyWiki and do anything you want with it without any license fee payment or other legal obligation to the creators of TiddlyWiki or anyone else. The quid pro quo is that there is no warranty or guarantee with open source projects like TiddlyWiki. You can't sue the contributors to TiddlyWiki for any loss or damage due to the use of TiddlyWiki: even if your data is lost due to a tragic chain of circumstances that involves TiddlyWiki.

You are respectfully requested that to make an attribution to the project, but there's no obligation to do so.

For the avoidance of doubt, any information that you choose to store within your own copy of TiddlyWiki remains yours; using TiddlyWiki to publish content doesn't change whatever rights you may have to that content.

limit Operator

3rd February 2015 at 7:54pm

Common Operators Filter Operators Negatable Operators Order Operators

purpose	select the first or last `N` input titles
input	a selection of titles
parameter	`N` = an integer, defaulting to 0
output	the first `N` input titles
`!` **output**	the last `N` input titles

Learn more about how to use Filters

`limit[N]` and `!limit[N]` are synonyms for `first[N]` and `last[N]` respectively.

Examples

limit Operator (Examples)

23rd January 2015 at 10:43pm

limit Operator Operator Examples

```
[sort[modified]!limit[20]]
```

→ the 20 most recently modified tiddlers

Try it

`[has[created]sort[created]limit[10]]`

→ the oldest 10 tiddlers in the wiki

Try it

lingo Macro

21st February 2015 at 4:57pm

Core Macros Macros

The **lingo** macro relates to the translation of TiddlyWiki's user interface into other languages. It returns a piece of text in the user's currently selected language.

Translatable text is supplied by language plugins containing tiddlers with specific titles that start with `$:/language/`.

Parameters

title

The title of the shadow tiddler that contains the text. The prefix `$:/language/` is added automatically

Examples

lingo Macro (Examples)

21st February 2015 at 5:01pm

lingo Macro Macro Examples

This example shows the text used as the basis for the title of a newly created tiddler:

copy to clipboard

```
<<lingo DefaultNewTiddlerTitle>>
```

Try it

This example shows the name of the eighth month of the year, for use in formatting dates:

copy to clipboard

```
<<lingo Date/Long/Month/8>>
```

Try it

LinkCatcherWidget

26th February 2022 at 1:11pm

MessageHandlerWidgets TriggeringWidgets Widgets

Introduction

The link catcher widget traps WidgetMessage: tm-navigate dispatched within its child content by performing any or all of these actions:

- sending a different widget message
- setting a tiddler to the title of the navigated tiddler
- setting a tiddler to a specified value
- performing a series of ActionWidgets

Content and Attributes

The content of the `<$linkcatcher>` widget is displayed normally.

Attribute	Description
to	Optional title of the tiddler to be set to the title of the navigated tiddler
message	Optional identifier for a widget message to be sent when a navigation is caught
set	Optional title of the tiddler to be set to a specified value when navigation occurs
setTo	Value to be assigned by the `set` attribute
actions	Actions to be performed when a link is caught. Within the action string, the variable "navigateTo" contains the title of the tiddler being navigated. ⓘ **Introduced in v5.1.23** the **modifier** variable lists the modifier keys that are pressed when the action is invoked. The possible modifiers are **ctrl**, **ctrl-alt**, **ctrl-shift**, **alt**, **alt-shift**, **shift** and **ctrl-alt-shift**



Introduced in v5.2.0 For more complex use cases involving trapping the **`tm-navigate`** message consider the MessageCatcherWidget which provides greater flexibility

Linking in WikiText

30th December 2021 at 3:59pm

WikiText

A key capability of WikiText is the ability to make links to other tiddlers or to external websites.

Manual Links

Link to a tiddler by title:

```
[[Tiddler Title]]
```

To link to a tiddler and specify the text of the link:

```
[[Displayed Link Title|Tiddler Title]]
```

You can also create a link from the editor toolbar. Click **link** (🔗), and search and select a tiddler.

CamelCase Links

For tiddler titles that match the CamelCase rules, just typing the title without double square brackets will automatically create a link.

You can suppress a link from being recognised by preceding it with `~`. For example:

copy to clipboard

```
* ~HelloThere is not a link
* ~http://google.com/ is not a link
```

That renders as:

- HelloThere is not a link
- http://google.com/ is not a link

... and the underlying HTML is:

```
<ul><li>HelloThere is not a link</li><li>http://google.com/ is not a link</li></ul>
```

External Links

To link to an external resource such as a website or a file, type its *full* URL, including the URI scheme such as a protocol (e.g. `http://`, `file://`) or `mailto`:

```
https://tiddlywiki.com/
```

```
[[TW5|https://tiddlywiki.com/]]

[[Mail me|mailto:me@where.net]]

[[Open file|file:///c:/users/me/index.html]]
```

For this syntax to work, the URL has to be recognisable as a URL. Otherwise, it is treated as a tiddler title. As a result, in case you want to link to a resource locatable using a relative path, use the extended syntax:

```
[ext[Open file|index.html]]

[ext[Open file|./index.html]]

[ext[Open file|../README.md]]

[ext[Open file|c:\users\me\index.html]]
```

The extended syntax still works with full URLs, although in that case it is not necessary:

```
[ext[https://tiddlywiki.com]]

[ext[TW5|https://tiddlywiki.com]]

[ext[Mail me|mailto:me@where.net]]

[ext[Open file|file:///c:/users/me/index.html]]
```

You can also use the extended syntax to force an external link:

```
[ext[Donate|bitcoin:1aabbdd....?amount=0.001]]
```

Customising Tiddler Links

See the LinkWidget for details of the underlying widget used to implement tiddler links, including macros that can be used to customise its behaviour.

Generating dynamic links

Linking in WikiText does not lend itself well to creating dynamic links. The reason is because this WikiText link:

```
[[link to myself|Linking in WikiText]]
```

is shorthand for using the LinkWidget and TextWidget like this:

```
<$link to="Linking in WikiText">
  <$text text="link to myself"/>
</$link>
```

Since both the link title ("link to myself") and the link target ("Linking in WikiText") are used as widget attributes, no WikiText expansion takes place.

For example, an attempt to use a reference and a variable to dynamically generate a link like this:

⚠ Warning: Don't do it this way!	`[[{{!!caption}}|<<currentTiddler>>]]`

is the same as trying to use the LinkWidget and TextWidget like this:

⚠ Warning:
Don't do it this way!
	`<$link to="<<currentTiddler>>">`
	`  <$text text="{{!!caption}}"/>`
	`</$link>`

This will not work as desired. Content inside of quoted widget attributes is treated as a literal value and protected from WikiText expansion, so it will render like this:

⚠ Warning: Don't do it this way!	*{{!!caption}}*

In order to get the desired result, the LinkWidget can be used directly like this:

copy to clipboard

```
<$link to=<<currentTiddler>>>{{!!caption}}</$link>
```

That renders as:

Linking

See also another example of constructing dynamic links.

Linking within tiddlers - "anchor links"

In TiddlyWiki anchor links can help us link to target points and distinct sections within rendered tiddlers. They can help the reader navigate longer tiddler content.

See Anchor Links using HTML for more information.

links Operator

3rd February 2015 at 7:55pm

Common Operators Filter Operators

purpose	find the titles linked to by each input title
input	a selection of titles
parameter	none

output	the titles to which the input tiddlers contain hard links

Learn more about how to use Filters

Each input title is processed in turn. The corresponding tiddler's list of links is generated, in the order in which they appear in the tiddler's text, and dominantly appended to the operator's overall output.

Examples

links Operator (Examples)

24th January 2015 at 9:27pm

links Operator Operator Examples

```
[[HelloThere]links[]]
```

Try it

```
[all[current]links[]]
```

→ tiddlers hard-linked from the current one

Try it

Here are some hard links:

- HelloThere
- Filter Operators
- links Operator

LinkWidget

13th November 2023 at 10:33am

Widgets

The `link` widget generates links to tiddlers. (Use the HTML `<a>` element to generate external links).

Content and Attributes

The content of the link widget is rendered within the `<a>` tag representing the link. If the content is empty then the title of the target tiddler is rendered as the default, for example:

- `<$link/>` is equivalent to `<$link to=<<currentTiddler>>><$view field="title"/></$link>`
- `<$link to="HelloThere"/>` is equivalent to `<$link to="HelloThere">HelloThere</$link>`

Attribute	Description
to	The title of the target tiddler for the link (defaults to the current tiddler)
aria-label	Optional accessibility label
tooltip	Optional tooltip WikiText
tabindex	Optional numeric tabindex

draggable	"yes" to enable the link to be draggable (defaults to "yes")
tag	Optional tag to override the default "a" element
class	Optional CSS classes *in addition to* the default classes (see below)
overrideClass	ⓘ **Introduced in v5.1.16** Optional CSS classes *instead of* the default classes
data-*	ⓘ **New in v5.3.2** Optional data attributes to be assigned to the HTML element
style.*	ⓘ **New in v5.3.2** Optional CSS properties to be assigned to the HTML element

The draggable functionality is equivalent to using the DraggableWidget with the **tiddler** attribute set to the link target.

The default value of the tooltip attribute is supplied by the **`tv-wikilink-tooltip`** variable.

This means that you can control the text of a link tooltip in several ways:

```
<$link to="HelloThere" tooltip="Custom tooltip">Link 1</$link>

<$link to="HelloThere" tooltip="Another tooltip">{{$:/core/icon}}</$link>

<$set name="tv-wikilink-tooltip" value="I'm a link to {{!!title}}">
<$link to="HelloThere">Link 2</$link>
</$set>
```

Renders as:

Link 1



Link 2

Note that the tooltip is rendered with the current tiddler set to the target of the link.

A useful convention is to set the tooltip like this:

```
\define tv-wikilink-tooltip()
<$transclude field="tooltip"><$transclude field="title"/></$transclude>
\end
```

This causes the tooltip to be the **tooltip** field of the target tiddler. If the field isn't present, then the title is used instead.

CSS Classes

The link widget automatically determines and applies the following classes to links:

- `tc-tiddlylink` - applied to all links
- `tc-tiddlylink-external` - applied to external, non-tiddler links
- `tc-tiddlylink-internal` - applied to tiddler links
- `tc-tiddlylink-missing` - applied to tiddler links where the target tiddler doesn't exist
- `tc-tiddlylink-resolves` - applied to tiddler links when the target tiddler does exist

Use the `class` attribute to specify additional css classes, or `overrideClass` to apply only that but not the above defaults, e.g. when used in a LinkCatcherWidget:

```
*<$link class="example">Here</$link> the `example` class is added.
*<$link overrideClass="example">Here</$link> only the `example` class applies.
*<$link overrideClass="">Here</$link> no class is set.
```

That renders as:

- Here the `example` class is added.
- Here only the `example` class applies.
- Here no class is set.

`href` generation

The following process is used to generate the `href` attribute of the generated HTML `<a>` element:

1. If **`tv-get-export-link`** is defined it is invoked to convert the target tiddler title to the `href` value
   - In practice, only a JavaScript macro can be used
2. ⓘ **Introduced in v5.1.15** If **`tv-filter-export-link`** is defined it is interpreted as a filter that converts the target tiddler title to the `href` value
3. If **`tv-wikilink-template`** is defined it is treated as a specialised macro body that can perform limited conversion of the target tiddler title to the `href` value
4. Otherwise, the target tiddler title is URI encoded to create the `href`

Configuration variables

- `tv-wikilinks`
- `tv-filter-export-link`
- `tv-wikilink-template`
- `tv-wikilink-tooltip`
- `tv-get-export-link`

list Operator

15th November 2021 at 10:23am

Field Operators Filter Operators Negatable Operators Selection Constructors

purpose	select titles via a list field
input	ignored
! **input**	a selection of titles
parameter	**R** = a reference to a field or property of a particular tiddler
output	the titles stored as a title list at **R**
! **output**	those input titles that are *NOT* mentioned at **R**

Learn more about how to use Filters

R can reference either a field or a property. See TextReference for the syntax.

- If neither is specified, the **list** field is used by default. So `[list[T]]` outputs the titles listed in the **list** field of tiddler T.

- If **R** consists of *ONLY* a field or a property, the tiddler part of the reference defaults to the current tiddler. So `[list[!!tags]]` outputs the titles listed in the **tags** field of the current tiddler.

Examples

list Operator (Examples)

23rd January 2015 at 10:44pm

list Operator Operator Examples

These examples make use of the Days of the Week tiddler.

```
[list[HelloThere]]
```

Try it

```
[list[Days of the Week!!short]]
```

Try it

list-links Macro

5th November 2022 at 10:08am

Core Macros Macros

The **list-links** macro returns a formatted list of links to a selection of tiddlers.

If a tiddler has a **`caption`** field, this is shown instead of the tiddler's title. If the caption field is empty, a blank entry is shown.

Note: Each first step of a filter run not given any input titles receives the output of [all[tiddlers]] as its input. This means all the existing non-shadow tiddlers.

Parameters

filter
: A filter selecting which tiddlers to include

field
: The name of the field to transclude for each list item, defaulting to `caption`

type
: An HTML element to use for the overall list element, defaulting to `ul`

subtype
: An HTML element to use for each item in the list, defaulting to `li`

class
: A CSS class for the overall list element

emptyMessage
: Optional wikitext to display if there are no tiddlers with the specified tag

Examples

list-links Macro (Examples)

21st February 2015 at 5:07pm

list-links Macro Macro Examples

This example lists all the tiddlers whose titles start with J:

copy to clipboard

```
<<list-links filter:"[prefix[J]]">>
```

Try it

This example lists the documentation tiddlers for the core macros, each of which has a **`caption`** field:

copy to clipboard

```
<<list-links filter:"[tag[Core Macros]]">>
```

Try it

list-links-draggable Macro

14th December 2021 at 3:16pm

Core Macros Macros

The **list-links-draggable** macro renders the ListField of a tiddler as a list of links that can be reordered via drag and drop.

Parameters

tiddler
The title of the tiddler containing the list

field
The name of the field containing the list (defaults to `list`)

emptyMessage
Optional wikitext to display if there is no output (tiddler is not existed, field is not existed or empty)

type
The element tag to use for the list wrapper (defaults to `ul`)

subtype
The element tag to use for the list items (defaults to `li`)

class
Optional space separated classes to add to the wrapper element

itemTemplate
Optional title of a tiddler to use as the template for rendering list items

If the `itemTemplate` parameter is not provided then the list items are rendered as simple links. Within the `itemTemplate`, the currentTiddler Variable refers to the current list item.

Examples

list-links-draggable Macro (Examples)

28th March 2017 at 11:12pm

list-links-draggable Macro Macro Examples

copy to clipboard

```
<<list-links-draggable tiddler:"Days of the Week">>
```

Try it

list-tagged-draggable Macro

9th January 2018 at 6:12pm

Core Macros Macros

The **list-tagged-draggable** macro renders the tiddlers with a particular tag as a list of links that can be reordered via drag and drop.

Parameters

tag

The title of the tag

subFilter

An optional subfilter to filter out unwanted items (eg `!tag[done]`)

itemTemplate

Optional title of a tiddler to use as the template for rendering list items

emptyMessage
Optional wikitext to display if there are no tiddlers with the specified tag

Note that the ordering is accomplished by assigning a new list to the `list` field of the tag tiddler. Any `list-before` or `list-after` fields on any of the other tiddlers carrying the tag are also removed to ensure the `list` field is respected.

If the `itemTemplate` parameter is not provided then the list items are rendered as simple links. Within the `itemTemplate`, the currentTiddler Variable refers to the current list item.

Examples

list-tagged-draggable Macro (Examples)

29th March 2017 at 11:33am

list-tagged-draggable Macro Macro Examples

copy to clipboard

```
<<list-tagged-draggable tag:"Features">>
```

Try it

list-thumbnails Macro

12th June 2020 at 7:18pm

Core Macros Macros

The **list-thumbnails** macros are used to create lists of linkable thumbnail panels.

Parameters

filter
: filter for selecting thumbnails

width
: Width of thumbnail (default 280 pixels)

height
: Height of thumbnail (default 157 pixels)

Examples

list-thumbnails Macro (Examples)

12th June 2020 at 7:14pm

Macro Examples

This example lists all the tiddlers which are tagged with "HelloThumbnail":

copy to clipboard

```
<<list-thumbnails filter:"[tag[HelloThumbnail]]" width:"168" height:"95">>
```

Try it

listed Operator

3rd February 2015 at 7:57pm

Field Operators Filter Operators

purpose	find the titles that list the input titles
input	a selection of titles
parameter	**F** = the name of a field, defaulting to **list**
output	the titles in which field **F** mentions any of the input titles

Learn more about how to use Filters

F is assumed to be a title list.

Each input title is processed in turn. A list of tiddlers whose **F** field mentions it is generated (in no particular order) and dominantly appended to the operator's overall output.

Examples

listed Operator (Examples)

23rd January 2015 at 10:46pm

listed Operator Operator Examples

```
[[HelloThere]listed[]]
```

Try it

```
[all[current]listed[my-special-list]]
```

Try it

ListenCommand

1st July 2018 at 7:10pm

Commands

Introduced in v5.1.18 See WebServer for details of TiddlyWiki's web server functionality.

Serves a wiki over HTTP.

The listen command uses NamedCommandParameters:

```
--listen [<name>=<value>]...
```

All parameters are optional with safe defaults, and can be specified in any order. The recognised parameters are:

- **host** - optional hostname to serve from (defaults to "127.0.0.1" aka "localhost")
- **path-prefix** - optional prefix for paths
- **port** - port number on which to listen; non-numeric values are interpreted as a system environment variable from which the port number is extracted (defaults to "8080")
- **credentials** - pathname of credentials CSV file (relative to wiki folder)
- **anon-username** - the username for signing edits for anonymous users
- **username** - optional username for basic authentication
- **password** - optional password for basic authentication
- **authenticated-user-header** - optional name of request header to be used for trusted authentication.
- **readers** - comma-separated list of principals allowed to read from this wiki
- **writers** - comma-separated list of principals allowed to write to this wiki
- **csrf-disable** - set to "yes" to disable CSRF checks (defaults to "no")
- **root-tiddler** - the tiddler to serve at the root (defaults to "**$:/core/save/all**")
- **root-render-type** - the content type to which the root tiddler should be rendered (defaults to "text/plain")
- **root-serve-type** - the content type with which the root tiddler should be served (defaults to "text/html")
- **tls-cert** - pathname of TLS certificate file (relative to wiki folder)
- **tls-key** - pathname of TLS key file (relative to wiki folder)
- **debug-level** - optional debug level; set to "debug" to view request details (defaults to "none")
- **gzip** - set to "yes" to enable gzip compression for some http endpoints (defaults to "no")

- **use-browser-cache** - set to "yes" to allow the browser to cache responses to save bandwidth (defaults to "no")

For information on opening up your instance to the entire local network, and possible security concerns, see the WebServer tiddler at TiddlyWiki.com.

ListField

24th January 2015 at 9:29pm

Fields

The `list` field of a tiddler is an optional feature that can be used to help structure your content. Its value is a title list, and it can be used in several ways:

- The `list` field of a tiddler that is being used as a tag determines the ordering of the tiddlers that carry that tag - see Tagging for details
- The `list` filter selects the entries from a list
- The `listed` filter selects the tiddlers that list the selected tiddler(s)
- The NavigatorWidget manipulates a *StoryList* tiddler containing a `list` field of the tiddlers that are displayed in the main story column

ListopsData

8th November 2015 at 5:18am

Operator Examples

```
DataIndex:
```

Lists in WikiText

7th June 2016 at 11:31am

Lists WikiText

Bulleted Lists

You can create bulleted (unordered) lists with `*` characters (⁝≡):

copy to clipboard

```
* First list item
* Second list item
** A subitem
* Third list item
```

That renders as:

- First list item
- Second list item
    - A subitem
- Third list item

... and the underlying HTML is:

```
<ul><li>First list item</li><li>Second list item<ul><li>A subitem</li></ul></li><li>Third list item</
li></ul>
```

Numbered Lists

Numbered (ordered) lists use `#` instead of `*` (¡≡):

1. First item
2. Second item
3. Third item

You can also mix ordered and unordered list items:

copy to clipboard

```
* To do today
*# Eat
* To get someone else to do
*# This
*# That
*## And the other
```

That renders as:

- To do today
    1. Eat
- To get someone else to do
    1. This
    2. That
        1. And the other

... and the underlying HTML is:

```
<ul><li>To do today<ol><li>Eat</li></ol></li><li>To get someone else to do<ol><li>This</
li><li>That<ol><li>And the other</li></ol></li></ol></li></ul>
```

Here's an example the other way around, with numbers as the first level:

copy to clipboard

```
# To do today
#* Eat
# To get someone else to do
#* This
#* That
#** And the other
```

That renders as:

1. To do today

    - Eat
2. To get someone else to do
    - This
    - That
        - And the other

... and the underlying HTML is:

```
<ol><li>To do today<ul><li>Eat</li></ul></li><li>To get someone else to do<ul><li>This</
li><li>That<ul><li>And the other</li></ul></li></ul></li></ol>
```

CSS Classes

You can also assign a CSS class to an individual member of a list with this notation:

copy to clipboard

```
* List One
*.MyClass List Two
* List Three
```

That renders as:

- List One

- List Two
- List Three

... and the underlying HTML is:

```
<ul><li>List One</li><li class="MyClass">List Two</li><li>List Three</li></ul>
```

Mixing Lists and Block Quotes

Note that Block Quotes in WikiText can be mixed with lists. For example:

copy to clipboard

```
* List One
** List Two
**> A quote
**> Another quote
* List Three
```

That renders as:

- List One
    - List Two

> A quote
>
> Another quote

- List Three

... and the underlying HTML is:

```
<ul><li>List One<ul><li>List Two<blockquote><div>A quote</div><div>Another quote</div></blockquote></
li></ul></li><li>List Three</li></ul>
```

Paragraphs in Lists

Entries in the list are delimited with a linebreak, making it impossible to include linebreaks within a list entry. There are a couple of workarounds.

First, you can transclude paragraph content from another tiddler. For example:

```
* First entry
* <$transclude tiddler="MyTiddler" mode="block"/>
* Third entry
```

Secondly, you can use an HTML "div" element to contain the multiline content. For example:

```
# Step 1
# Step 2
```

```
# Step 3<div>

Here is the first of several paragraphs. Note that the double linebreak preceding this paragraph is
significant.

And here is the second of several paragraphs.
</div>
# Step 4
# Step 5
# Step 6
```

ListWidget

31st August 2023 at 8:29pm

Lists Widgets

Introduction

The list widget displays a sequence of tiddlers that match a tiddler filter. It can be used for many purposes:

- Displaying custom lists of links, like in TiddlyWiki5's sidebar
- Custom lists, such as "all tiddlers tagged 'task' that are not tagged 'done'"
- Listing each of the tags applied to a tiddler
- Handling the main story river

The tiddlers are displayed by transcluding each in turn through a template. There are several ways to specify the template and for controlling the behaviour of the list.

Examples

plain list

```
<$list filter="[tag[ListWidget]sort[title]]"/>
```

Displays as:

> GroupedLists

custom item output

```
<$list filter="[tag[ListWidget]sort[title]]">
<<currentTiddler>>
{{||$:/core/ui/ViewTemplate/tags}}
</$list>
```

Displays as:

> GroupedLists
>
> Lists ListWidget

custom item template

```
<$list filter="[tag[ListWidget]sort[title]]" template="$:/core/ui/ViewTemplate/subtitle"/>
```

Displays as:

> 22nd February 2015 at 12:01am

Grouped Lists

See GroupedLists for how to generate nested and grouped lists using the ListWidget.

Content and Attributes

The content of the `<$list>` widget is an optional template to use for rendering each tiddler in the list.

New in v5.3.2 If the widgets `<$list-template>` or `<$list-empty>` are found as immediate children of the **$**`ListWidget` widget then the content of those widgets are used as the list item template and/or the empty template. Note that the **`emptyMessage`** and **`template`** attributes take precedence if they are present.

The action of the list widget depends on the results of the filter combined with several options for specifying the template:

- If the filter evaluates to an empty list, the text of the **emptyMessage** attribute is rendered, and all other templates are ignored
- Otherwise, if the **template** attribute is specified then it is taken as the title of a tiddler to use as a template for rendering each item of the list

- Otherwise, if the list widget content is not blank, it is used as a template for rendering each item of the list
- Otherwise, a default template is used consisting of a `<span>` or `<div>` element wrapped around a link to the item

Attribute	Description
filter	The tiddler filter to display
limit	ⓘ **New in v5.3.2** Optional numeric limit for the number of results that are returned. Negative values will return the results from the end of the list
template	The title of a template tiddler for transcluding each tiddler in the list. When no template is specified, the body of the ListWidget serves as the item template. With no body, a simple link to the tiddler is returned.
editTemplate	An alternative template to use for DraftTiddlers in edit mode
join	ⓘ **New in v5.3.2** Text to include between each list item
variable	The name for a variable in which the title of each listed tiddler is stored. Defaults to **currentTiddler**
counter	ⓘ **Introduced in v5.2.0** Optional name for a variable in which the 1-based numeric index of each listed tiddler is stored (see below)
emptyMessage	Message to be displayed when the list is empty
storyview	Optional name of module responsible for animating/processing the list
history	The title of the tiddler containing the navigation history

`counter` attribute

The optional `counter` attribute specifies the name of a variable to hold the 1-based numeric index of the current item in the list.

Two additional variables are also set to indicate the first and last items in the list:

- `<counter-variable-name>-first` is set to `yes` for the first entry in the list, `no` for the others

- `<counter-variable-name>-last` is set to `yes` for the last entry in the list, `no` for the others

For example:

```
<$list filter="[tag[About]sort[title]]" counter="counter">
<div>
<<counter>>: ''<$text text=<<currentTiddler>>/>'' (is first: <<counter-first>>, is last: <<counter-
last>>)
</div>
</$list>
```

Displays as:

> 1: **Acknowledgements** (is first: yes, is last: no)
>
> 2: **Contributors** (is first: no, is last: no)
>
> 3: **Funding TiddlyWiki** (is first: no, is last: no)
>
> 4: **History of TiddlyWiki** (is first: no, is last: no)
>
> 5: **License** (is first: no, is last: no)
>
> 6: **Open Collective** (is first: no, is last: no)
>
> 7: **Releases** (is first: no, is last: no)
>
> 8: **RoadMap** (is first: no, is last: no)

> 9: **TiddlyWiki Archive** (is first: no, is last: yes)

Note that using the `counter` attribute can reduce performance when working with list items that dynamically reorder or update themselves. The best advice is only to use it when it is really necessary: to obtain a numeric index, or to detect the first or last entries in the list. Note that if you are only using it to insert something (like a comma) between list items, the `join` attribute performs much better and you should use it instead of `counter`.

Setting `counter="transclusion"` is a handy way to make child elements for each list element be identified as unique. A common use case are multiple tag macros for the same tag generated by a list widget. Refer to tag macro examples for more details.

`join` attribute

ⓘ **New in v5.3.2** The optional `join` attribute allow you to insert some WikiText between each list item without needing to use the `counter` attribute, which can become quite slow if the list is updated frequently.

ⓘ **New in v5.3.2** If the widget `<$list-join>` is found as an immediate child of the **$**`ListWidget` widget then the content of that widget is used as the "join" template, included between two list items. Note that the **`join`** attribute takes precedence if it is present.

For example:

```
<$list filter="[tag[About]sort[title]]" join=", " variable="item"><<item>></$list>
```

Displays as:

> Acknowledgements, Contributors, Funding TiddlyWiki, History of TiddlyWiki, License, Open Collective, Releases, RoadMap, TiddlyWiki Archive

Edit mode

The `<$list>` widget can optionally render draft tiddlers through a different template to handle editing, see DraftMechanism.

`storyview` attribute

The `storyview` attribute specifies the name of an optional module that can animate changes to the list (including navigation). The core ships with the following storyview modules:

- `classic` : renders the list as an ordered sequence of tiddlers
- `zoomin` : just renders the current tiddler from the list, with a zoom animation for navigating between tiddlers
- `pop` : shrinks items in and out of place

In order for the storyviews to animate correctly each entry in the list should be a single block mode DOM element.

History and navigation

The optional `history` attribute specifies the name of a tiddler that is used to track the current tiddler for navigation purposes. When the history tiddler changes the list view responds by telling the listview to handle navigating to the new tiddler. See HistoryMechanism for details.

Additional Notes and Edge Cases

- If the `filter` attribute is not present then a default of `[!is[system]sort[title]]` is used
- If the list widget is completely empty (ie only whitespace between the opening and closing tags), then it behaves as if the content were a `DIV` or a `SPAN` containing a link to the current tiddler (it's a `DIV` if the list widget is in block mode, or a SPAN if it is in inline mode)

- If the `template` attribute is not present then the content of the list widget will be used as the template, unless the widget is completely empty in which case a default template is used

Literal Attribute Values

15th June 2023 at 6:54am

Widget Attributes WikiText

Literal attribute values can use several different styles of quoting:

- Single quotes (eg `attr='value'`)
- Double quotes (eg `attr="value"`)
- Tripe double quotes (eg `attr="""value"""`)
- No quoting is necessary for values that do not contain spaces (eg `attr=value`)

Literal attribute values can include line breaks. For example:

```
<div data-address="Mouse House,
Mouse Lane,
Rodentville,
Ratland."/>
```

By using triple-double quotes you can specify attribute values that contain single double quotes. For example:

```
<div data-address="""Mouse House,
"Mouse" Lane,
```

```
Rodentville,
Ratland."""/>
```

LoadCommand

19th January 2014 at 9:15pm

Commands

Load tiddlers from TiddlyWiki (`.html`), `.tiddler`, `.tid`, `.json` or other local files. The processing applied to incoming files is determined by the file extension. Use the alternative `import` command if you need to specify the deserializer and encoding explicitly.

```
--load <filepath> [noerror]
--load <dirpath> [noerror]
```

By default, the load command raises an error if no tiddlers are found. The error can be suppressed by providing the optional "noerror" parameter.

To load tiddlers from an encrypted TiddlyWiki file you should first specify the password with the PasswordCommand. For example:

```
tiddlywiki ./MyWiki --password pa55w0rd --load my_encrypted_wiki.html
```

Note that TiddlyWiki will not load an older version of an already loaded plugin.

Locator plugin by bimlas

6th January 2021 at 4:10pm

Community Plugins

For those who use many tags or store many different topics in a common wiki the Locator plugin is a table of contents widget and an enhanced search engine that gives you the opportunity to filter results by related tags. Unlike table of contents, standard search and list of tags, this plugin offers these features in an organic, collaborative way.

https://bimlas.github.io/tw5-locator/

log Operator

20th November 2020 at 5:14am

Binary Mathematics Operators Filter Operators Mathematics Operators

purpose	treating each input title as a number, return its logarithm with base equal to the numeric value of the parameter if specified, otherwise base e
input	a selection of titles
parameter	**N** = a number
output	the logarithm of each input title as numbers, with base **N** if specified otherwise base e

Learn more about how to use Filters

Introduced in v5.1.23

Examples

log Operator (Examples)

20th November 2020 at 5:20am

log Operator Operator Examples

Logarithm of `8` with base `2`:

```
[[8]log[2]]
```

Try it

Logarithm of `100` with base `10`:

```
[[100]log[10]]
```

Try it

Natural logarithm of 10 (base `e`), equivalent to `ln(10)` in mathematics:

```
[[10]log[]]
```

Try it

LogWidget

20th November 2020 at 4:49pm

Debugging Widgets Widgets

Introduction

Introduced in v5.1.23 The **log** widget can be used to output debugging information to the JavaScript console supported by most browsers.



For use with ActionWidgets see ActionLogWidget which uses identical parameters

Content and Attributes

The **log** widget is invisible. Any content within it is ignored. Note that the widget will log to the console both when it is first rendered and also every time it refreshes.

When the widget is rendered, the names and values of all attributes are logged to the JavaScript console. In addition there are optional attributes that can be used:

Optional Attribute	Description
$$filter	All variables matching this filter will also be logged.
$$message	A message to display as the title of the information logged. Useful when several `log` widgets are used in sequence.
$$all	Set to "yes" to log all variables in a collapsed table. Note that if there is nothing specified to log, all variables are always logged instead.

Example

Here is an example of logging two variables:

```
<$log myVar=<<myVar>> otherVar=<<otherVar>>/>
```

To log all variables:

```
<$log />
```

To log two variables as well as all core variables (which start with `tv-`):

```
<$log myVar=<<myVar>> other={{!!status}} $$filter="[prefix[tv-]]"/>
```

lookup Operator

16th January 2021 at 9:13am

Filter Operators

purpose	applies a prefix to each input title to yield the title of a tiddler from which the final value is retrieved. With a single parameter, the default field is "text" and the default index is "0". If a second parameter is provided, that becomes the target field or index.
input	a selection of titles
suffix	**`D, I`** = the default value to be used for missing lookups. This operator can now accept a second suffix of `:index`, see below for details
parameter	**`P, T`** = prefix applied to input titles to yield title of lookup tiddler from which value is retrieved. Now accepts 1 or 2 parameters, see below for details
output	the lookup values corresponding to each lookup tiddler

Learn more about how to use Filters

ⓘ Introduced in v5.1.15

The action of this operator is as follows with 1 parameter:

- Apply the specified prefix to each input tiddler title, yielding a new list of tiddler titles
- Transclude the value of the `text` field each of those tiddlers
    - Substitute the default value for missing or empty values
- Return the list of values

ⓘ Introduced in v5.2.2

The use of the `:index` second suffix changes the default lookup location from field: `text` to index: `0`. This is used if no 2nd parameter is provided.

The action of this operator is as follows with 2 parameters:

If there are two parameters provided, use the second parameter as the target field or index.

Note

> If there is only one parameter given, the filter checks for a second suffix equal to "index". If this suffix is found, the default target index is "0".

In all other cases, the default target field is "text".

Then:

- Apply the specified prefix to each input tiddler title, yielding a new list of tiddler titles

- Transclude the value of the target field or index
  - Substitute the default value for missing or empty values
- Return the list of values

Examples

lookup Operator (Examples)

18th November 2021 at 3:54am

Operator Examples

```
[all[shadows+tiddlers]tag[$:/tags/PageControls]lookup[$:/config/PageControlButtons/Visibility/]]
```

→ Retrieve the visibility status of each page control button

Try it

```
[all[shadows+tiddlers]tag[$:/tags/PageControls]lookup:show[$:/config/PageControlButtons/Visibility/]]
```

→ Retrieve the visibility status of each page control button, this time with a default value

Try it

```
[all[tiddlers]has[plugin-type]removeprefix[$:/plugins/tiddlywiki/]lookup:missing-description:field[$:/
```

```
plugins/tiddlywiki/],[description]]
```

→ Retrieve the description of all plugin-tiddlers that are in the `$:/plugins/tiddlywiki/` namespace.

Try it

```
OriginalTiddlerPaths +[lookup:missing-index:index[$:/config/],[HelloThere]]
```

→ Lookup the original tiddler path on disk for the *Hello There* tiddler.

Try it

```
OriginalTiddlerPaths +[lookup:missing-index:index[$:/config/],[MissingTiddler]]
```

→ Lookup the original tiddler path on disk for the *MissingTiddler* tiddler.

Try it

```
HistoryList MissingHistoryList +[lookup:missing-0:index[$:/]]
```

→ Retrieve index `0` from the `$:/HistoryList` and `$:/MissingHistoryList`.

Try it

```
OriginalTiddlerPaths MissingTiddlerPaths +[lookup:missing-tiddler:index[$:/config/],[$:/key-test]]
```

→ Retrieve index `$:/key-test` from `$:/config/OriginalTiddlerPaths` and `$:/config/MissingTiddlerPaths`.

Try it

lowercase Operator

19th June 2019 at 1:08pm

Filter Operators

purpose	returns each item in the list as lowercase
input	a selection of titles
output	the input titles with each uppercase letter replaced by the equivalent lowercase letter

Learn more about how to use Filters

ⓘ Introduced in v5.1.20

See also uppercase Operator and sentencecase Operator.

Examples

lowercase Operator (Examples)

13th June 2019 at 5:38pm

lowercase Operator Operator Examples

```
[[Abc]lowercase[]]
```

Try it

```
[tag[HelloThere]lowercase[]]
```

Try it

Macro Call Syntax

21st February 2015 at 11:23pm

Macro Syntax

What follows is a formal presentation of the syntax of the WikiText syntax for macro calls, using railroad diagrams. A simpler overview is also available.

TypeError: setting getter-only property "parentNode"

space denotes a sequence of whitespace characters.

The macro's **name** is a sequence of non-whitespace characters other than `(` or `>`.

Each individual **`param-value`** has the following syntax:

TypeError: setting getter-only property "parentNode"

The **`param-name`** is a sequence of letters (`A`–`Z`, `a`–`z`), digits (`0`–`9`), hyphens (`-`) and underscores (`_`).

The **`value`** is specified as follows:

TypeError: setting getter-only property "parentNode"

Macro Calls

19th April 2023 at 12:31pm

Macros WikiText

Introduction

This tiddler describes the different ways in which macros can be called.

Macro Call Transclusion Shortcut

To call a macro, place `<<` double angle brackets `>>` around the name and any parameter values.

```
<<mymacro param:"This is the parameter value">>
```

By default, parameters are listed in the same order as in the macro's definition. A parameter can be labelled with its name and a colon to allow them to be listed in a different order.

If no value is specified for a parameter, the default value given for that parameter in the macro definition is used instead. (If no default value was defined, the parameter is blank).

Each parameter value can be enclosed in `'` single quotes `'`, `"` double quotes `"`, `"""` triple double quotes `"""` or `[[` double square brackets `]]`. Triple double quotes allow a value to contain almost anything. If a value contains no spaces or single or double quotes, it requires no delimiters.

A more formal presentation of this syntax is also available.

See some examples and discussion about parser modes.

Macro Calls with `$transclude` Widget

The shortcut syntax expands to the `$transclude` widget with the `$variable` attribute specifying the name of the macro to transclude.

```
<$transclude $variable="mymacro" param="This is the parameter value"/>
```

The widget itself offers greater flexibility than the shortcut syntax, including the ability to specify parameter values.

Assigning Macro Calls to Attribute Values

The result of a macro can be directly assigned to an attribute of a widget or HTML element. The result of the macro is not wikified, but the parameter substitution is performed.

```
<div class=<<myclasses "Horizontal">>>
...
```

```
</div>
```

Using Macro Calls in Filters

Macro calls can be used in filters:

```
<$list filter="[<mymacro param:'value'>]">
...
</$list>
```

Macro Calls in WikiText

17th September 2022 at 9:48am

See Macro Calls.

Macro Calls in WikiText (Examples)

21st February 2015 at 11:17pm

WikiText Examples

```
\define sayhi(name:"Bugs Bunny",address:"Rabbit Hole Hill")
Hi, I'm $name$ and I live in $address$.
\end
```

copy to clipboard

```
<<sayhi>>
```

Try it

copy to clipboard

```
<<sayhi Bugs>>
```

Try it

copy to clipboard

```
<<sayhi "Donald Duck" Disneyland>>
```

Try it

copy to clipboard

```
<<sayhi "Mickey Mouse" "Mouse House">>
```

Try it

copy to clipboard

```
<<sayhi name:'Minnie Mouse' address:[[Mouse House]]>>
```

Try it

copy to clipboard

```
<<sayhi address:"Quacky Towers" name:"Donald Duck">>
```

Try it

copy to clipboard

```
<<sayhi "Mickey Mouse" """Mouse House",
Rodent's Lane,
Squeaksville,
Ratland""">>
```

Try it

Macro Definition Syntax

21st February 2015 at 11:23pm

Macro Syntax

What follows is a formal presentation of the syntax of the `\define` pragma, using railroad diagrams. A simpler overview is also available.

TypeError: setting getter-only property "parentNode"

space denotes a sequence of whitespace characters.

The macro's **name** is a sequence of non-whitespace characters other than `(` or `>`.

The parameter declaration list (**params**) has the following syntax:

TypeError: setting getter-only property "parentNode"

The parameter separator (**sep**) is any sequence of characters that does not match a **param-name** . Among other things, this includes commas, spaces and linefeeds.

A **param-name** is a sequence of letters (`A` – `Z` , `a` – `z`), digits (`0` – `9`), hyphens (`-`) and underscores (`_`).

Each individual **param** has the following syntax:

TypeError: setting getter-only property "parentNode"

The optional **default** value of a parameter is specified as follows:

TypeError: setting getter-only property "parentNode"

The **rest** of the definition has the following syntax:

TypeError: setting getter-only property "parentNode"

lf denotes a linefeed.

The **snippet** is any sequence of characters that doesn't terminate the macro definition. That is to say, a single-line snippet cannot contain a linefeed, and a multi-line snippet cannot contain `\end` on a line of its own.

The snippet can contain placeholders with the following syntax:

TypeError: setting getter-only property "parentNode"

Macro Definitions

19th April 2023 at 12:31pm

Macros WikiText

Introduction

This tiddler describes the different ways in which macros can be defined.

Macro Definition Pragma

Macros are created using the Pragma: \define at the start of a tiddler. The definitions are available in the rest of the tiddler that defines them, plus any tiddlers that it transcludes.

```
\define mymacro(param)
This is the macro text (param=$param$)
\end
```

Nested Macro Definitions

Macro definitions can be nested to any number of required levels by specifying the name of the macro in the `\end` marker. Nested macro definitions must appear at the start of the definition that contains them. For example:

copy to clipboard

```
\define special-button(caption:"Click me")
\define actions()
<$action-sendmessage $message="tm-notify" $param="HelloThere"/>
\end actions
<$button actions=<<actions>>>
$caption$
</$button>
\end special-button

<<special-button>>
```

That renders as:

Click me

Note that the textual substitution of macro parameters that occurs when the outer macro is rendered will apply to the nested definitions as well. That generally means that textual substitution of macro parameters should not be used within nested macros.

Parameters of nested macros can also be accessed via the `<<__variablename__>>` syntax. As ordinary variables, these parameters are available within nested child macros (and grandchildren etc).

Macro Definition with Set Widget

Macros are implemented as a special type of variable and so internally are actually defined with a **`$set`** widget.

```
<$set name="mymacro" value="This is the macro text">
...
</$set>
```

Note

> that it is not currently possible to specify parameters when defining a macro with the **`$set`** widget.

Importing Macro Definitions

The Pragma: \import or **`$importvariables`** widget can be used to copy macro definitions from another tiddler.

`$:/tags/Macro` Tag

Global macros can be defined using the SystemTag: $:/tags/Macro.

The tag SystemTag: $:/tags/Macro/View is used to define macros that should only be available within the main view template and the preview panel.

The tag SystemTag: $:/tags/Macro/View/Body is used to define macros that should only be available within the main view template body and the

preview panel.

JavaScript Macros

Macros can also be written as JavaScript modules.

Macro Definitions in WikiText

20th August 2018 at 6:51pm

See Macro Definitions.

Macro Parameter Handling

19th April 2023 at 12:31pm

Macros WikiText

Introduction

Macros parameters are handled in two different ways:

1. Textual substitution is always performed for each parameter before the macro contents is used
2. When the macro contents are wikified the parameters are made available as variables. The variable names are formed by wrapping the parameter name with double underscores

Somewhat confusingly, in some situations both of these mechanisms will occur; this is related to the pitfalls of using macros.

Textual Substitution of Parameters and variables

The following substitutions take place before the text of a macro is used:

- The pattern `$param$` is replaced with the value of the named parameter
- The pattern `$(variable)$` is replaced with the value of the named variable

The actual value of the parameter or variable is substituted for the placeholder whenever the macro is called:

copy to clipboard

```
\define say-hi-using-parameters(name,address)
Hi, I'm $name$ and I live in $address$.
\end

<<say-hi-using-parameters name:"Bugs" address:"Rabbit Hole Hill">>
```

That renders as:

Hi, I'm Bugs and I live in Rabbit Hole Hill.

Here's an example using variable substitution:

copy to clipboard

```
\define say-hi-using-variables()
Hi, I'm $(name)$ and I live in $(address)$.
```

```
\end

\define name() Bugs

<$let address="Rabbit Hole Hill">
<<say-hi-using-variables>>
</$let>
```

That renders as:

Hi, I'm Bugs and I live in Rabbit Hole Hill.



It is important to note that if the text being inserted contains any substitution tokens then they will in turn be processed. This can lead to unexpected results.

Accessing Parameters as Variables

When macros are wikified, the parameters can be accessed as variables with the name of the parameter wrapped with double underscores. For example, the parameter `address` would be accessed as the variable `__address__`.

Thus, the example above could also be expressed as:

copy to clipboard

```
\define say-hi-using-parameters(name,address)
Hi, I'm <<__name__>> and I live in <<__address__>>.
```

```
\end

<<say-hi-using-parameters name:"Bugs" address:"Rabbit Hole Hill">>
```

That renders as:

Hi, I'm Bugs and I live in Rabbit Hole Hill.

Accessing parameters as variables only works in macros that are wikified and not, for example, when a macro is used as an attribute value.

Advantages of Accessing Parameters as Variables

The primary advantage of the technique is that it avoids the parameter value being substituted into the macro as a literal string, which in turn can help avoid issues with parameters that contain quotes.

For example, consider this macro. It invokes another macro using the single parameter as an argument for it:

```
\define film-quote(line) <$macrocall $name="anothermacro" actor="Bugs Bunny" line="""$line$"""/>
```

The code above will fail if the macro is invoked with the argument containing triple double quotes (for example `<<film-quote 'I quote thrice """ - see!?'>>`). Using parameter variables offers a workaround:

```
\define film-quote(line) <$macrocall $name="anothermacro" actor="Bugs Bunny" line=<<__line__>>/>
```

See Macro Pitfalls for more discussion.

Macro Pitfalls

19th April 2023 at 12:31pm

Introduction

In the early days of TiddlyWiki, macros were the best way of encapsulating snippets for reuse, and so they were used extensively. However, they have always suffered from some significant disadvantages that can give rise to errors and poor performance.

ⓘ **New in v5.3.0** Macros have been joined by Procedures, Custom Widgets and Functions which together provide more robust and flexible ways to encapsulate and re-use code. It is now recommended to only use macros when textual substitution is specifically required.

Shortcomings of Textual Substitution

TiddlyWiki's handling of macro parameters is based on "textual substitution" which means that the string values of the parameters provided when calling a macro are plugged into the macro definition before it is wikified.

Here's a typical example of the approach in early versions of TiddlyWiki 5. The intention is to provide a macro that takes a single parameter of the title of the tiddler to view:

```
\define mymacro(title)
<$codeblock code={{$title$}}/>
\end
```

That works for simple cases like `<<mymacro "HelloThere">>` but is subtly brittle. For example, the macro above would fail with tiddler titles containing double closing curly braces. Trying to use it with the title `foo}}bar` would lead to the macro being expanded to the following invalid syntax:

```
<$codeblock code={{foo}}bar}}/>
```

As a result of this issue, for many years the TiddlyWiki 5 user interface failed if a variety of combinations of special characters were encountered in tiddler titles.

This issue has been mitigated over the years, particularly by providing access to the macro parameters as variables. However, for backwards compatibility, this was done without affecting the existing syntax, which required us to adopt the clumsy protocol of wrapping the parameter name in double underscores to get the name of the corresponding variable.

Performance of Global Macros

Global Macro Definitions defined with the SystemTag: $:/tags/Macro suffer from poor performance because every macro has to be parsed regardless of whether it is actually used.

Furthermore, the way that definitions are imported means that updating a tiddler tagged SystemTag: $:/tags/Macro will cause the entire page to be refreshed.

Macro Syntax

27th February 2015 at 8:31pm

Macros

- Macro Call Syntax
- Macro Definition Syntax

MacroCallWidget

19th April 2023 at 12:31pm

$:/deprecated Widgets

DEPRECATED SINCE: 5.3.0 (see TranscludeWidget)

The **$macrocall** widget is deprecated. While it will continue to work, users are now advised to use the **$transclude** widget, converting the `$name` attribute to `$variable`.

For example,

```
<$macrocall $name="my-macro" my-parameter="Elephant"/>
```

should be changed to:

```
<$transclude $variable="my-macro" my-parameter="Elephant"/>
```

Internally, the **$macrocall** widget is implemented via the **$transclude** widget.

Content and Attributes

The content of the **$macrocall** widget is ignored.

Attribute	Description
$name	Name of the macro to invoke
$type	ContentType with which the macro text should be parsed (defaults to `text/vnd.tiddlywiki`)
$output	ContentType for the output rendering (defaults to `text/html`, can also be `text/plain` or `text/raw`)
parameters	Macro parameters specified as attributes

Macros

22nd September 2023 at 11:49am

Concepts Reference

Important

ⓘ **New in v5.3.0** Macros have been superseded by Procedures, Functions and Custom Widgets which together provide more robust and flexible ways to encapsulate and re-use code.

For text substitutions it is now recommended to use: Substituted Attribute Values, substitute Operator and Transclusion and Substitution

Introduction

A **macro** is a named snippet of text. They are typically defined with the Pragma: \define:

```
\define my-macro(parameter:"Default value")
This is the macro, and the parameter is $parameter$.
\end
```

The name wrapped in double angled brackets is used a shorthand way of transcluding the snippet. Such transclusions are known as **macro calls**, and each call can supply a different set of parameters:

```
<<my-macro>>
<<my-macro "The parameter">>
```

The parameters that are specified in the macro call are substituted for special placeholders within the snippet:

- `$parameter-name$` is replaced with the value of the named parameter
- `$(variable-name)$` is replaced with the value of the named variable).

How Macros Work

Macros are implemented as a special kind of variable. The only thing that distinguishes them from ordinary variables is the way that the parameters are handled.

Using Macros

- Macro Definitions describes how to create macros
- Macro Calls describes how to use macros
- Macro Parameter Handling describes how macro parameters work
- Macro Pitfalls describes some of the pitfalls of using macros
- Core Macros lists the built-in core macros

Macros in WikiText

21st February 2015 at 10:40am

See Macros.

Make the beds

15th November 2021 at 2:32am

task TaskManagementExample

This is a sample task for the TaskManagementExample

makedatauri Macro

30th March 2020 at 12:53pm

Core Macros Macros

The **makedatauri** macro takes a piece of text and an associated ContentType, and returns a corresponding data URI.

`makedatauri` is used to implement the **`datauri`** macro.

Parameters

text

The text to be converted to a data URI

type

The ContentType of the text

_canonical_uri

The optional **_canonical_uri** address of the content

Examples

makedatauri Macro (Examples)

21st February 2015 at 11:36pm

Macro Examples makedatauri Macro

copy to clipboard

```
<<makedatauri "some example text" "text/plain">>
```

Try it

MakeLibraryCommand

3rd September 2014 at 10:52am

Commands

Constructs the `$:/UpgradeLibrary` tiddler for the upgrade process.

The upgrade library is formatted as an ordinary plugin tiddler with the plugin type `library`. It contains a copy of each of the plugins, themes

and language packs available within the TiddlyWiki5 repository.

This command is intended for internal use; it is only relevant to users constructing a custom upgrade procedure.

```
--makelibrary <title>
```

The title argument defaults to `$:/UpgradeLibrary`.

See the UpgradeMechanism for more details.

makepatches and applypatches Operator (Examples)

4th March 2023 at 5:03pm

applypatches Operator makepatches Operator Operator Examples

These examples use the example texts in Hamlet, taken from https://neil.fraser.name/software/diff_match_patch/demos/patch.html

Shakespeare's original	Hamlet: Do you see yonder cloud that's almost in shape of a camel? Polonius: By the mass, and 'tis like a camel, indeed. Hamlet: Methinks it is like a weasel. Polonius: It is backed like a weasel. Hamlet: Or like a whale? Polonius: Very like a whale. – Shakespeare
Modern English	Hamlet: Do you see the cloud over there that's almost the shape of a camel? Polonius: By golly, it is like a camel, indeed. Hamlet: I think it looks like a weasel.

	Polonius: It is shaped like a weasel. Hamlet: Or like a whale? Polonius: It's totally like a whale. – Shakespeare
Trekkie's Copy	Kirk: Do you see yonder cloud that's almost in shape of a Klingon? Spock: By the mass, and 'tis like a Klingon, indeed. Kirk: Methinks it is like a Vulcan. Spock: It is backed like a Vulcan. Kirk: Or like a Romulan? Spock: Very like a Romulan. – Trekkie

Use `makepatches` to generate the set of patches to transform Shakepeare's original into Modern English:

```
[{Hamlet##Shakespeare-old}makepatches{Hamlet##Shakespeare-new}]
```

Try it

Use `applypatches` to apply the patches to Shakespeare's original text:

```
[{Hamlet##Shakespeare-old}makepatches{Hamlet##Shakespeare-new}] :map[{Hamlet##Shakespeare-old}
applypatches<currentTiddler>]
```

Try it

In the above example, the Map Filter Run Prefix is used to pass the patches information as a parameter to `applypatches`. Inside `:map`, **`currentTiddler`** is set to the input title (i.e. the previously generated patches).

The patch information from the Shakepeare texts can also be used to transform the *Trekkie's Copy* to a Modern English version:

```
[{Hamlet##Shakespeare-old}makepatches{Hamlet##Shakespeare-new}] :map[{Hamlet##Trekkie-old}
applypatches<currentTiddler>]
```

Try it

The above examples used the character mode of `makepatches`. The `word` mode yields very similar results in this case, even when applied to the *Trekkie's Copy*.

```
[{Hamlet##Shakespeare-old}makepatches:words{Hamlet##Shakespeare-new}]
```

Try it

```
[{Hamlet##Shakespeare-old}makepatches:words{Hamlet##Shakespeare-new}] :map[{Hamlet##Trekkie-old}applypatches<currentTiddler>]
```

Try it

The `lines` mode doesn't work as well in this application:

```
[{Hamlet##Shakespeare-old}makepatches:lines{Hamlet##Shakespeare-new}]
```

Try it

```
[{Hamlet##Shakespeare-old}makepatches:lines{Hamlet##Shakespeare-new}] :map[{Hamlet##Trekkie-old}applypatches<currentTiddler>]
```

Try it

It is better suited as a very fast algorithm to detect line-wise incremental changes to texts and store only the changes instead of multiple versions of the whole texts.

makepatches Operator

4th March 2023 at 1:24pm

Filter Operators String Operators

purpose	returns a set of patches that transform the input to a given string
input	a selection of titles
suffix	**T** = `lines` to operate in line mode, `words` to operate in word mode. If omitted (default), the algorithm operates in character mode. See notes below.
parameter	**S** = a string of characters
output	a set of patch instructions per input title to be used by the applypatches Operator to transform the input title(s) into the string **S**

Learn more about how to use Filters

Introduced in v5.2.6

The difference algorithm operates in character mode by default. This produces the most detailed diff possible. In `words` mode, each word in the input text is transformed into a meta-character, upon which the algorithm then operates. In the default character mode, the filter would find two patches between "*ActionWidget*" and "Action-Widgets" (the hyphen and the plural s), while in `words` mode, the whole word is found to be changed. In `lines` mode, the meta-character is formed from the whole line, delimited by newline characters, and is found to be changed independent of the number of changes within the line.

The different modes influence the result when the patches are applied to texts other than the original, as well as the runtime.



The calculation in `words` mode is roughly 10 times faster than the default character mode, while `lines` mode can be more than 100 times faster than the default.

Examples

Making a custom journal button

14th November 2017 at 10:58pm

Customise TiddlyWiki

To get your own ***custom*** Journal button, start by cloning the **New Journal** button.

Then create your own config tiddlers, eg:

- `$:/config/myNewTiddler/Tags` and ...
- `$:/config/myNewTiddler/Title` and ...
- `$:/config/myNewTiddler/Text`

Edit your custom button and "search and replace" the config tiddler names.

Change the following lines, near the end of the tiddler from

```
<$set name="journalTitleTemplate" value={{$:/config/NewJournal/Title}}>
<$set name="journalTags" value={{$:/config/NewJournal/Tags}}>
<$set name="journalText" value={{$:/config/NewJournal/Text}}>
```

to

```
<$set name="journalTitleTemplate" value={{$:/config/myNewTiddler/Title}}>
<$set name="journalTags" value={{$:/config/myNewTiddler/Tags}}>
<$set name="journalText" value={{$:/config/myNewTiddler/Text}}>
```

if you want to see your config tiddlers in the sidebar use:

```
<$set name="journalTitleTemplate" value={{config/myNewTiddler/Title}}>
<$set name="journalTags" value={{config/myNewTiddler/Tags}}>
<$set name="journalText" value={{config/myNewTiddler/Text}}>
```

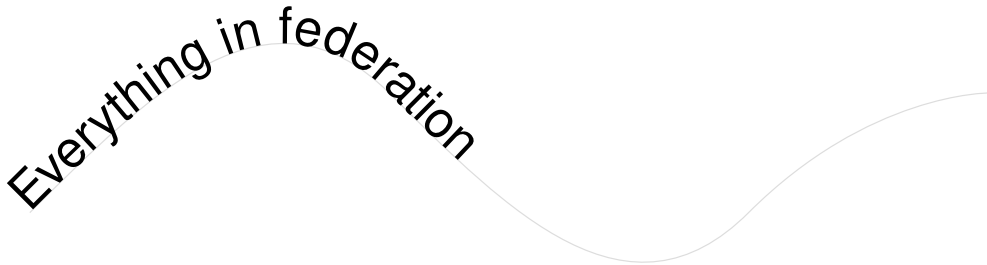
Making curved text with SVG

5th May 2015 at 8:37am

Learning

This demo shows how to use SVG to render transcluded text along a path. Enter some text in the textbox below to try it out; view the source to see how it is done.

Everything in federation



Manually installing a plugin

17th June 2022 at 1:59pm

Plugins

If you want to follow the steps *side by side* you can open the **example edition** in a new window.

1. Create a backup of your current TiddlyWiki HTML file (just in case)
2. Open your TiddlyWiki in a browser
3. In another browser window, find a link to the plugin, e.g. https://tiddlywiki.com/plugins/tiddlywiki/katex. You will typically find these links on the home page of the plugin
4. Drag the link similar to `$:/plugins/tiddlywiki/example` to the browser window containing your TiddlyWiki

5. Import the new plugin shown in the *$:/Import* tiddler
6. Save your TiddlyWiki (⊘)
7. If a yellow warning bar appears at the top of the window, refresh the window so that TiddlyWiki completes installation of the plugin ↺
   - ⓘ **Introduced in v5.1.22** It is no longer necessary to refresh TiddlyWiki when deleting plugins that support dynamic loading. See PluginMechanism for more details
8. The plugin should now be available for use

Note

> More information about installing a plugin form the official plugin library, can be found at Installing a plugin from the plugin library
> An overview can be found at Plugins

Map Filter Run Prefix

10th July 2023 at 9:33am

Named Filter Run Prefix

purpose	modify input titles by the result of evaluating this filter run for each item
input	the filter output of all previous runs so far
suffix	ⓘ **Introduced in v5.2.3** `flat` to return all results from the filter run, or omit (default) to return only the first result
output	the input titles as modified by the result of this filter run

ⓘ **Introduced in v5.2.0**
TypeError: setting getter-only property "parentNode"

Each input title from previous runs is passed to this run in turn. The filter run transforms the input titles and the output of this run replaces the input title. For example, the filter run `[get[caption]else{!!title}]` replaces each input title with its caption field, unless the field does

not exist in which case the title is preserved.

Note that within the filter run, the **currentTiddler** variable is set to the title of the tiddler being processed. This permits filter runs like `:map[{!!price}multiply{!!cost}]` to be used for computation.

The following variables are available within the filter run:

- **currentTiddler** - the input title
- **..currentTiddler** - the value of the variable `currentTiddler` outside the filter run.
- **index** - ⓘ **Introduced in v5.2.1** the numeric index of the current list item (with zero being the first item in the list).
- **revIndex** - ⓘ **Introduced in v5.2.1** the reverse numeric index of the current list item (with zero being the last item in the list).
- **length** - ⓘ **Introduced in v5.2.1** the total length of the input list.

Filter runs used with the `:map` prefix should return at least the same number of items that they are passed. Input titles for which the filter run returns no output are replaced by an empty string. In particular, when retrieving the value of a field with the get Operator it is helpful to guard against a missing field value using the else Operator. For example `[get[myfield]else[default-value]...`.

Map Filter Run Prefix (Examples)

Map Filter Run Prefix (Examples)

5th March 2023 at 1:54pm

Map Filter Run Prefix

Replace the input titles with the caption field if it exists, otherwise preserve the input title:

```
[tag[Widgets]] :map[get[caption]else{!!title}]
```

Try it



The above example is equivalent to `[tag[Widgets]] :map[{!!caption}!is[blank]else{!!title}]`. Note that referencing a field as a text reference such as `{!!caption}` returns an empty string for a non-existent or empty caption field. Therefore a check for `is[blank]` is needed before the `else` operator

For each title in a shopping list, calculate the total cost of purchasing each item:

```
[tag[shopping]] :map[get[quantity]else[0]multiply{!!price}]
```

Try it

Get the tags of all tiddlers tagged `Widget:`

```
[tag[Widgets]] :map:flat[tagging[]] :and[!is[blank]unique[]]
```

Try it



Without the `flat` suffix the `:map` filter run only returns the first result for each input title

Comparison between `:map` with and without the `flat` suffix

The `:map` filter run will return at least as many outputs as given in the input. By default one input item will result in exactly one output item. When the filter run transforms an input item into an empty result, the output for that item will be an empty string. When the filter run transforms an input item into multiple items, only the first item will appear in the output. This behavior can be overridden by providing the `flat` suffix. The `flat` suffix will cause all the items to appear in the output.

`:map`	`:map:flat`
`[range[4]] :map[match[this matches nothing]]` Try it	`[range[4]] :map:flat[match[this matches nothing]]` Try it
`[range[4]] :map[range<currentTiddler>]` Try it	`[range[4]] :map:flat[range<currentTiddler>]` Try it

`[range[4]] :map[range<currentTiddler>]` Try it	`[range[4]] :map:flat[range<currentTiddler>first[]]` Try it
`[range[4]] :map[range<currentTiddler>sum[]]` Try it	`[range[4]] :map:flat[range<currentTiddler>sum[]]` Try it
`[[1,2,3]] [[4,5]] :map[split[,]]` Try it	`[[1,2,3]] [[4,5]] :map:flat[split[,]]` Try it
`[[1,2,3]] [[4,5]] :map[split[,]]` Try it	`[[1,2,3]] [[4,5]] :map:flat[split[,]first[]]` Try it

Comparison between `:map` and `:and`/`+` filter run prefixes

The functionality of the `:map` filter run prefix has some overlap with the `:and` prefix (alias `+`). They will sometimes return the same results as each other. In at least these cases, the results will be different:

1. The `:and` filter run can modify the number of items (either increase or decrease). The `:map` run will never alter the number of items.
2. The number of items in the `:and` filter run will also decrease due to de-duplication. The `:map` run will not de-duplicate.
3. Explicit references to the "currentTiddler" variable will behave differently
4. Implicit references to the "currentTiddler" using TextReference will behave differently.

`:map`	`:and`
results are the same	
`[range[5]] :map[add[1]]` Try it	`[range[5]] :and[add[1]]` Try it
`[range[5]] :map[addsuffix[ hello]]` Try it	`[range[5]] :and[addsuffix[ hello]]` Try it

`[tag[shopping]] :map[get[quantity]]` Try it	`[tag[shopping]] :and[get[quantity]]` Try it
decrease in the number of items	
`[range[5]] :map[sum[]]` Try it	`[range[5]] :and[sum[]]` Try it
`[range[5]] :map[join[,]]` Try it	`[range[5]] :and[join[,]]` Try it
increase in the number of items	

`[[1,2,3]] [[4,5]] :map[split[,]]` Try it	`[[1,2,3]] [[4,5]] :and[split[,]]` Try it
de-duplication	
`[range[5]] :map[[hello]]` Try it	`[range[5]] :and[[hello]]` Try it
currentTiddler	
`[tag[shopping]] :map[<currentTiddler>]` Try it	`[tag[shopping]] :and[<currentTiddler>]` Try it

```
[tag[shopping]] :map[{!!quantity}]
```

Try it

```
[tag[shopping]] :and[{!!quantity}]
```

Try it

Markdown

28th July 2022 at 9:16pm

Definitions

> Markdown is a text-to-HTML conversion tool for web writers. Markdown allows you to write using an easy-to-read, easy-to-write plain text format, then convert it to structurally valid XHTML (or HTML).
>
> *https://daringfireball.net/projects/markdown/*

TiddlyWiki and Markdown share some formatting rules eg: inline code and code blocks but WikiText also offers advanced functions like transclusions and macros

Markdown Plugin

28th February 2017 at 11:25am

OfficialPlugins Plugin Editions

The Markdown plugin enables you to use tiddlers that are written in standard Markdown markup.

See https://tiddlywiki.com/plugins/tiddlywiki/markdown

match Operator

11th July 2023 at 10:24am

Filter Operators

purpose	returns each item in the list that matches the parameter string
input	a selection of titles
suffix	the **match** operator uses a rich suffix, see below for details
parameter	the string to be matched
output	each item in the input list that matches the parameter string (potentially including duplicates)
! output	each item in the input list that does not match the parameter string

Learn more about how to use Filters

ⓘ Introduced in v5.1.20

The **match** operator uses an extended syntax that permits multiple flags to be passed:

```
[match:<flag list>[<parameter>]]
```

- **flag list**: a comma delimited list of flags

The available flags are:

- **casesensitive**: (default), this flag forces a case-sensitive match, where upper and lower case letters are considered different

- **caseinsensitive**: overrides the default so that upper and lower case letters are considered identical for matching purposes

Examples

match Operator (Examples)

31st July 2019 at 10:04am

match Operator Operator Examples

```
a b c +[match[b]]
```

Try it

```
[match[HelloThere]]
```

Try it

Mathematics Operators

21st March 2023 at 2:30pm

Filters

Introduced in v5.1.20 The mathematics filter operators allow numerical calculations to be performed within filters.

Using mathematics operators

The mathematics operators interpret their arguments as numbers according to the following rules:

- Any non-numeric characters after a valid number are ignored (e.g. `28px` is interpreted as the number 28)
- If the argument cannot be interpreted as a number, the value 0 is used (e.g. `foo` is interpreted as the number 0)
- The special values `Infinity` and `-Infinity` can be used to represent positive and negative infinity respectively



Some filter operators remove duplicate items which can cause unexpected results when using the mathematics operators. See Dominant Append for details.

The mathematics operators take three different forms:

- **Unary operators** apply an operation to each number in the input list (e.g. negate, truncate, sign)
  - `=1 =2 =3 =4 +[negate[]]` evaluates to `-1` `-2` `-3` `-4`
  - `=1.2 =2.4 =3.6 =4.8 +[trunc[]]` evaluates to `1` `2` `3` `4`
  - `=1.2 =2.4 =3.6 =4.8 +[round[]]` evaluates to `1` `2` `4` `5`
- **Binary operators** apply an operation and parameter to each number in the input list (e.g. add, multiply, remainder)
  - `=1 =2 =3 =4 +[add[3]]` evaluates to `4` `5` `6` `7`
  - `=1 =2 =3 =4 +[multiply[8]]` evaluates to `8` `16` `24` `32`
- **Reducing operators** apply an operation to all of the numbers in the input list, returning a single result (e.g. sum, product)
  - `=1 =2 =3 =4 +[sum[]]` evaluates to `10`
  - `=1 =2 =3 =4 +[product[]]` evaluates to `24`
  - `=1 =2 =3 =4 +[average[]]` evaluates to `2.5`

Operations Combination

Operations can be combined by concatenating them while merging the inner `][` characters.

- `[[355]divide[113]fixed[5]]` evaluates to `3.14159`
- `[range[100]sum[]divide[100]]` evaluates to `50.5`
- `[tag[HelloThere]get[text]length[]sum[]]` evaluates to `10375`

Complex operations will sometimes need to be split up into separate filters. For example, to compute the average length of the text field of tiddlers tagged "HelloThere":

copy to clipboard

```
<$set name="number-of-tiddlers" value={{{ [tag[HelloThere]count[]] }}}>
Average length of <$text text=<<number-of-tiddlers>>/> tiddlers tagged <<tag "HelloThere">>: <$text
text={{{ [tag[HelloThere]get[text]length[]sum[]divide<number-of-tiddlers>fixed[3]] }}}/>
</$set>
```

That renders as:

Average length of 8 tiddlers tagged HelloThere : 1296.875

Mathematics Operators list

Operator	Purpose
abs	calculate the absolute value of a list of numbers
acos	calculate the arccosine value (in radians) of a list of numbers
add	treating each input title as a number, add to each the numeric value of the parameter
asin	calculate the arcsine value (in radians) of a list of numbers
atan	calculate the arctangent value (in radians) of a list of numbers
atan2	returns the angle in the plane (in radians) between the positive x-axis and the ray from (0,0) to the point (x,y), for [**Y**]atan2[**X**]
average	treating each input title as a number, compute their arithmetic mean
ceil	rounds a list of numbers up to the next largest integer
compare	filter the input by comparing each item against the parameter
cos	calculate the cosine value of a list of angles (given in radians)
divide	treating each input title as a number, divide it by the numeric value of the parameter
exponential	convert each number to exponential notation with **N** digits
fixed	convert each number to fixed point notation with **N** digits after the decimal point
floor	rounds a list of numbers to the largest integer less than or equal to each number
log	treating each input title as a number, return its logarithm with base equal to the numeric value of the parameter if specified, otherwise base e
max	treating each input title as a number, take the maximum of its value and the numeric value of the parameter
maxall	find the largest of a list of numbers
median	treating each input title as a number, compute their median value
min	treating each input title as a number, take the minimum of its value and the numeric value of the parameter
minall	find the smallest of a list of numbers
multiply	treating each input title as a number, multiply it by the numeric value of the parameter

negate	calculate the negation of a list of numbers
power	treating each input title as a number, raise it to the power of the numeric value of the parameter
precision	convert each number to a string with **N** significant digits
product	produce the product of the input numbers
remainder	treating each input title as a number, return the remainder when divided by the numeric value of the parameter
round	rounds a list of numbers to the nearest integer
sign	return -1, 0 or 1 for a list of numbers according to whether each number is negative, zero, or positive
sin	calculate the sine value of a list of angles (given in radians)
standard-deviation	treating each input title as a number, compute their standard-deviation
subtract	treating each input title as a number, subtract from each the numeric value of the parameter
sum	produce the sum of the input numbers
tan	calculate the tangent value of a list of angles (given in radians)
trunc	truncates a list of numbers to their integer part, removing any fractional part
untrunc	rounds a list of numbers to the next integer with largest absolute value, that is, away from zero
variance	treating each input title as a number, compute their variance

MathJax Plugin by Martin Kantor

6th January 2021 at 4:10pm

Community Plugins

An experimental *MathJax* plugin for TiddlyWiki version 5. As Martin says, the implementation is a bit of a hack but may be useful until we have a better alternative.

http://mathjax-tw5.kantorsite.net

> Welcome. I have created plugin for TiddlyWiki 5 which allows you to use *MathJax* (math in *TeX* and MathML) inside TiddlyWiki 5. It's unofficial plugin and it doesn't follow general policy of TiddlyWiki as stand-alone solution but it works. So you can use it if you want.

MathML

24th November 2021 at 10:53pm

Definitions

MathML is a markup language for mathematical notation that can be used with HTML.

If your browser supports it, MathML elements can be used in TiddlyWiki5 WikiText just like HTML.

Here is an example MathML equation from the W3C:

```
<math><mrow><mo>[</mo><mtable><mtr><mtd><mn>0</mn></mtd><mtd><mn>1</mn></mtd><mtd><mn>0</mn></mtd></
mtr><mtr><mtd><mn>0</mn></mtd><mtd><mn>0</mn></mtd><mtd><mn>1</mn></mtd></mtr><mtr><mtd><mn>1</mn></
mtd><mtd><mn>0</mn></mtd><mtd><mn>0</mn></mtd></mtr></mtable><mo>]</mo></mrow></math>
```

Renders as:

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

max Operator

21st March 2023 at 2:30pm

Binary Mathematics Operators Filter Operators Mathematics Operators

purpose	treating each input title as a number, take the maximum of its value and the numeric value of the parameter
input	a selection of titles
parameter	`N` = a number
output	the input as numbers, with any that are less than `N` being replaced by `N`

Learn more about how to use Filters

ⓘ **Introduced in v5.1.20** See Mathematics Operators for an overview.

See also the min Operator, and compare with the maxall Operator and the minall Operator.

Examples

max Operator (Examples)

11th June 2019 at 3:09pm

max Operator Operator Examples

```
[[23]max[32]]
```

Try it

```
=1 =2 =3 =4 +[max[3]]
```

Try it

maxall Operator

11th June 2019 at 6:09pm

Filter Operators Mathematics Operators Reducing Mathematics Operators

purpose	find the largest of a list of numbers
input	a selection of titles
output	the largest of the input numbers

Learn more about how to use Filters

ⓘ **Introduced in v5.1.20** See Mathematics Operators for an overview.

See also the minall Operator, and compare with the max Operator and the min Operator.

Examples

maxall Operator (Examples)

11th June 2019 at 6:10pm

maxall Operator Operator Examples

```
=1 =2 =3 =4 =5 +[maxall[]]
```

Try it

```
[tag[HelloThere]get[text]length[]maxall[]]
```

Try it

Mechanisms

10th September 2014 at 11:17pm

Reference

These are the internal mechanisms that fit together to make up TiddlyWiki.

- AlertMechanism

- DraftMechanism
- DragAndDropMechanism
- HistoryMechanism
- InfoMechanism
- PluginMechanism
- PopupMechanism
- RefreshMechanism
- RefreshThrottling
- SavingMechanism
- StateMechanism
- TranslationMechanism
- UpgradeMechanism
- WikificationMechanism

median Operator

26th April 2021 at 3:15pm

Filter Operators Mathematics Operators

purpose	treating each input title as a number, compute their median value
input	a selection of titles
output	the median of the input numbers

Learn more about how to use Filters

Introduced in v5.2.0 See Mathematics Operators for an overview.

Examples

median Operator (Examples)

26th April 2021 at 3:15pm

median Operator Operator Examples

```
=1 =3 =4 =5 +[median[]]
```

Try it

Note that if there is no input the operator returns `NaN`

```
[title[NotATiddler]get[price]] +[median[]]
```

Try it

Meetups

12th July 2016 at 2:15pm

Community

Local meetings around the world for TiddlyWiki people:

- The TiddlyWiki European Meetup 2016 and TiddlyWiki European Meetup 2017 in Oxford, UK
- OXTWIG, the **Oxford TiddlyWiki Interest Group** meets monthly in Oxford, UK to share experiences of using TiddlyWiki
- TiddlyWiki Camp Paris is an all day celebration and exploration of TiddlyWiki for experienced users and beginners alike

If you are a TiddlyWiki enthusiast please consider starting a local TWIG in your area, it's a great way to spread the word about using TiddlyWiki

MessageCatcherWidget

22nd December 2021 at 4:30pm

MessageHandlerWidgets TriggeringWidgets Widgets

Introduction

Introduced in v5.2.0

The message catcher widget traps messages dispatched within its child content, and allows invoking a series of ActionWidgets in response to those messages.

Content and Attributes

The content of the `<$messagecatcher>` widget is displayed normally.

Attribute	Description
{any attributes starting with $}	Each attribute name (excluding the $) specifies the name of a message, and the value specifies the action string to be invoked
type	Name of the message be trapped, for example "tm-scroll" or "tm-navigate". The `type` and `actions` attributes can be useful when the type of event needs to be specified via a variable or text reference
actions	Action string to be invoked when a matching message is trapped. Must be used in conjunction with the `type` attribute

Variables

The message catcher widget sets the following variables within each action string:

Variables	Description
`event-*`	All string-based properties of the `event` object, with the names prefixed with `event-`
`list-event`	A list of the names of each the string-based properties of the `event` object (the names are not prefixed with `event-`)
`event-paramObject-*`	All string-based properties of the `event.paramObject` object, with the names prefixed with `event-paramObject-`
`list-event-paramObject`	A list of the names of each the string-based properties of the `event.paramObject` object (the names are not prefixed with `event-paramObject-`)
`modifier`	For messages that originated with browser events, the modifier keys that were pressed when the event was fired. The possible modifiers are **normal** (no modifiers), **ctrl**, **ctrl-alt**, **ctrl-shift**, **alt**, **alt-shift**, **shift** and **ctrl-alt-shift**

Example

copy to clipboard

```
\define actions()
<$action-log/>
\end

<$messagecatcher $tm-navigate=<<actions>>>

Click on [[this link]] to fire an action. See the browser JavaScript console for the output

</$messagecatcher>
```

That renders as:

Click on *this link* to fire an action. See the browser JavaScript console for the output

MessageHandlerWidgets

13th November 2021 at 5:38pm

Widgets

Message handler widgets are those widgets which can react to one or more widget messages.



Widget messages are similar to low-level DOM events, except they are higher-level and specific to TiddlyWiki. To handle DOM events see the EventCatcherWidget

The following message handler widgets are provided:

- fieldmangler
- linkcatcher
- MessageCatcherWidget
- navigator

Messages

17th November 2021 at 5:20am

Concepts Reference

Widget **messages** are generated by Widgets in response to user actions. Messages have a **`name`**, an optional primary **`parameter`**, and one or more optional named parameters. These messages travel up the widget tree where they are handled by ancestor widgets or the core itself.

The following widget messages are implemented by the core:

- tm-add-field
- tm-add-tag
- tm-auto-save-wiki
- tm-browser-refresh
- tm-cancel-tiddler
- tm-clear-password
- tm-close-all-tiddlers
- tm-close-all-windows

- tm-close-other-tiddlers
- tm-close-tiddler
- tm-open-window
- tm-copy-to-clipboard
- tm-delete-tiddler
- tm-download-file
- tm-edit-bitmap-operation
- tm-edit-text-operation
- tm-edit-tiddler
- tm-focus-selector
- tm-fold-all-tiddlers
- tm-fold-other-tiddlers
- tm-fold-tiddler
- tm-full-screen
- tm-home
- tm-http-cancel-all-requests
- tm-http-request
- tm-import-tiddlers
- tm-load-plugin-from-library
- tm-load-plugin-library
- tm-login
- tm-logout

- tm-modal
- tm-navigate
- tm-new-tiddler
- tm-notify
- tm-open-external-window
- tm-open-window
- tm-perform-import
- tm-permalink
- tm-permaview
- tm-print
- tm-relink-tiddler
- tm-remove-field
- tm-remove-tag
- tm-rename-tiddler
- tm-save-tiddler
- tm-save-wiki
- tm-scroll
- tm-server-refresh
- tm-set-password
- tm-unfold-all-tiddlers
- tm-unload-plugin-library

Milk

6th October 2020 at 8:12pm

shopping

This is a sample shopping list item for the Shopping List Example

min Operator

11th June 2019 at 3:11pm

Binary Mathematics Operators Filter Operators Mathematics Operators

purpose	treating each input title as a number, take the minimum of its value and the numeric value of the parameter
input	a selection of titles
parameter	`N` = a number
output	the input as numbers, with any that are greater than `N` being replaced by `N`

Learn more about how to use Filters

ⓘ **Introduced in v5.1.20** See Mathematics Operators for an overview.

See also the max Operator, and compare with the maxall Operator and the minall Operator.

Examples

min Operator (Examples)

11th June 2019 at 3:11pm

min Operator Operator Examples

```
[[23]min[32]]
```

Try it

```
=1 =2 =3 =4 +[min[3]]
```

Try it

minall Operator

11th June 2019 at 6:10pm

Filter Operators Mathematics Operators Reducing Mathematics Operators

purpose	find the smallest of a list of numbers
input	a selection of titles
output	the smallest of the input numbers

Learn more about how to use Filters

ⓘ **Introduced in v5.1.20** See Mathematics Operators for an overview.

See also the maxall Operator, and compare with the max Operator and the min Operator.

Examples

minall Operator (Examples)

11th June 2019 at 6:11pm

minall Operator Operator Examples

```
=1 =2 =3 =4 =5 +[minall[]]
```

Try it

```
[tag[HelloThere]get[text]length[]minall[]]
```

Try it

minlength Operator

11th October 2016 at 9:42am

Filter Operators

purpose	filter items shorter than the specified minimum length

input	a list of items
parameter	**`minlength`** = the minimum length for items
output	those items at least as long as the specified minimum length

Learn more about how to use Filters

Introduced in v5.1.14

Examples

minlength Operator (Examples)

11th October 2016 at 9:42am

minlength Operator Operator Examples

These examples make use of the Days of the Week tiddler.

```
[list[Days of the Week]minlength[7]]
```

Try it

```
[list[Days of the Week]minlength[1]]
```

Try it

Mobile Drag And Drop Shim Plugin

28th March 2017 at 7:43pm

OfficialPlugins

The Mobile Drag And Drop Shim Plugin provides a "shim" that enables HTML 5 compatible drag and drop operations on mobile browsers, including iOS and Android. The shim was created by Tim Ruffles and is published at https://github.com/timruffles/ios-html5-drag-drop-shim.

Modals

21st December 2021 at 11:26am

Features

Modals (or "wizards") fade the main TiddlyWiki window to display an isolated tiddler that must be explicitly dismissed by the user.

The tiddler to be displayed can contain the following optional fields that are used to customise the modal:

Field	Description
footer	The footer text for the modal
subtitle	The subtitle text for a modal, displayed in a `h3` html tag
class	An additional class to apply to the modal wrapper
help	An optional external link that will be displayed at the left of the footer with the text "Help"
mask-closable	When set to **yes** or **true**, will close the modal dialog when the mask (area outside the modal) is clicked

Note that the footer and subtitle fields are not limited to plain text, and wiki text features such as widgets and transclusions can be used as well.

Modals are displayed with the WidgetMessage: tm-modal.

Open demo modal

Introduced in v5.2.4 allow using "mask-closable" field

modifier Variable

14th April 2022 at 6:28pm

Core Variables Variables

Within the **action** string of the DroppableWidget, the **startactions** and **endactions** of the DraggableWidget, the **action** string of the ButtonWidget and the **action** string of the LinkCatcherWidget and the EventCatcherWidget, the **modifier** variable contains the modifier key(s) held during the drag, click or other event. Possible key combinations are listed in the table below.

The variable contains a string that identifies the keys:

Modifier Key	Variable Content
meta	meta
ctrl	ctrl
alt	alt
shift	shift
meta+shift	meta-shift
meta+ctrl	meta-ctrl
meta+alt	meta-alt
ctrl+shift	ctrl-shift
alt+shift	alt-shift
ctrl+alt	ctrl-alt

Modifier Key	Variable Content
meta+ctrl+shift	meta-ctrl-shift
meta+ctrl+alt	meta-ctrl-alt
meta+alt+shift	meta-alt-shift
ctrl+alt+shift	ctrl-alt-shift
meta+ctrl+alt+shift	meta-ctrl-alt-shift
no modifier (normal click / drag)	normal

Some operating systems may intercept the **meta** key so it is never detected

Test modifier keys here

Modifier keys held while clicking the button above:

Modifying JSON tiddlers

27th April 2022 at 7:17pm

JSON in TiddlyWiki Learning

See JSON in TiddlyWiki for an overview of using JSON in TiddlyWiki.

Note that

Using ActionSetFieldWidget and ActionListopsWidget

The widgets ActionSetFieldWidget and ActionListopsWidget can manipulate named properties of data tiddlers by indicating the name of the

property in the $index attribute.

- To create or modify a named property with ActionSetFieldWidget, provide a $value attribute
- To delete a named property with ActionSetFieldWidget, omit the $value attribute.

ActionListopsWidget assigns the named property the list constructed through its $filter and $subfilter attributes.

moduleproperty Operator

19th September 2021 at 10:13pm

Filter Operators Special Operators

purpose	retrieve a module property
input	a selection of modules
parameter	module property to retrieve
output	the value of the module property as specified in the parameter

Learn more about how to use Filters

Examples

moduleproperty Operator (Examples)

19th September 2021 at 10:17pm

moduleproperty Operator Operator Examples

Get the name of the macro in the module `$:/core/modules/macros/qualify.js`:

```
[[$:/core/modules/macros/qualify.js]moduleproperty[name]]
```

Try it

For all macro modules retrieve their name module properties:

```
[[macro]modules[]moduleproperty[name]]
```

Try it

Modules

10th June 2016 at 10:22am

Concepts

A *module* in TiddlyWiki5 is a tiddler containing executable JavaScript, of the type `application/javascript` and with the field `module-type` set to the ModuleType of the module.

See **control panel** ⚙ > **Info** > **Advanced** > **Loaded Modules** for a list of currently loaded modules.

modules Operator

19th September 2021 at 10:11pm

Filter Operators Special Operators

purpose	select the names of all modules of the input module types
input	a selection of module types
parameter	none
output	the title of each module with any of the input types

Learn more about how to use Filters

ⓘ **Introduced in v5.2.0** The **`modules`** filter allows two optional parameters. When both are specified, it returns the modules with the module property specified in the first parameter which has the value in the second parameter.

Examples

modules Operator (Examples)

19th September 2021 at 10:10pm

modules Operator Operator Examples

```
[[filteroperator]modules[]]
```

Try it

Get the title of the macro module which a `property` name with value `version`:

```
[[macro]modules[name],[version]]
```

Try it

ModuleType

21st November 2021 at 4:13am

Definitions Modules

The `module-type` field of a JavaScript module is a string that identifies the type of the module. Here is a list of the module types used in this wiki:

allfilteroperator
: A sub-operator for the **all** filter operator.

animation
: Animations that may be used with the RevealWidget.

authenticator
: Defines how requests are authenticated by the built-in HTTP server.

bitmapeditoroperation
: A bitmap editor toolbar operation.

command
: Commands that can be executed under Node.js.

config

Data to be inserted into `$tw.config`.

filteroperator
Individual filter operator methods.

filterrunprefix

formatfilteroperator

global
Global data to be inserted into `$tw`.

indexer

info
Publishes system information via the $:/temp/info-plugin pseudo-plugin.

isfilteroperator
Operands for the **is** filter operator.

library
Generic module type for general purpose JavaScript modules.

macro
JavaScript macro definitions.

parser
Parsers for different content types.

route
Defines how individual URL patterns are handled by the built-in HTTP server.

saver

Savers handle different methods for saving files from the browser.

startup

Startup functions.

storyview

Story views customise the animation and behaviour of list widgets.

texteditoroperation

A text editor toolbar operation.

tiddlerdeserializer

Converts different content types into tiddlers.

tiddlerfield

Defines the behaviour of an individual tiddler field.

tiddlermethod

Adds methods to the `$tw.Tiddler` prototype.

upgrader

Applies upgrade processing to tiddlers during an upgrade/import.

utils

Adds methods to `$tw.utils`.

utils-node

Adds Node.js-specific methods to `$tw.utils`.

widget

Widgets encapsulate DOM rendering and refreshing.

widget-subclass

wikimethod

Adds methods to `$tw.Wiki`.

wikirule

Individual parser rules for the main WikiText parser.

moduletypes Operator

3rd February 2015 at 7:59pm

Filter Operators Selection Constructors Special Operators

purpose	select the names of all TiddlyWiki module types
input	ignored
parameter	none
output	the name of each known TiddlyWiki module type, in alphabetical order

Learn more about how to use Filters

Examples

moduletypes Operator (Examples)

23rd January 2015 at 11:16pm

moduletypes Operator Operator Examples

```
[moduletypes[]]
```

Try it

Monday

16th November 2021 at 11:11pm

Operator Examples

This example tiddler is used to illustrate some of the Filter Operators.

More actions

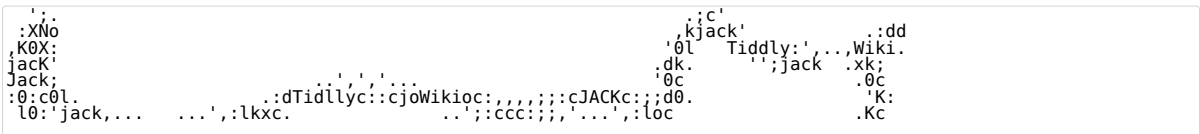
17th August 2016 at 1:01pm

Editor toolbar

Pressing ⌄ Gives you a dropdown menu with more editing options.

Motovun Jack.ascii

picture



```
,xk:...jackjackjack;'.                                                    :K,
  .lxxl;.....                                                           .0x.
     .':lTiddlyWikI.                                                   .xk.
                 ;0;                                                .  ;Od.
                .dO'                                              .0Ojackx,
                  cOo'                                          .kk'.'..
                    .k0'                                      .cOo.
                   lO;          ...                        .jack.
                 .xk'          .jackTiddlyWikijackO,      .lkd;.
                ;Oo.          .dO.               .ok.    ,Oo.
               l0;    .c,     ,0:                .kd    l0.
              .xk.   .xNx     lO.                 lO.   ;0,
               .Od.  ,XW0,    :0;                  dk.   .kx.
                .kx.  jACKx'   cOl.                .Ol     ;0:
                 .oO:  .JACKx.  .jack;'.           .0:      c0;
                   'dkc    ,lXKd'   .'jack.       .Ol       cOc.
                      .;ljack,.;xkc,...,oK,        'xxjack.   .oko.
                                'cljack'                .okdlcJack.
                                                         'JacK,'
```

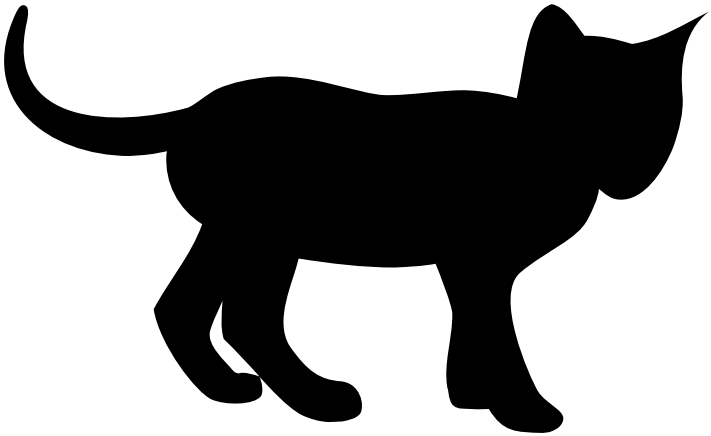
Motovun Jack.jpg

picture



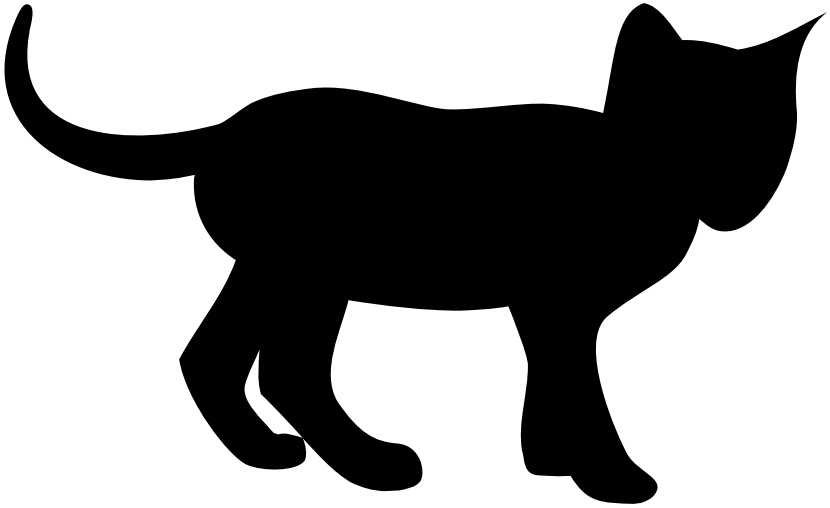
Motovun Jack.pdf

picture



Motovun Jack.svg

picture TiddlyWikiSitesMenu



move Operator

8th November 2015 at 9:24am

Filter Operators Listops Operators Order Operators

purpose	move marker N places in the list
input	a list of items
suffix	an integer N, defaulting to 1
parameter	**`marker`** = the list item to be used as a marker
output	re-ordered list of items

Learn more about how to use Filters

Examples

move Operator (Examples)

8th November 2015 at 6:16am

move Operator Operator Examples

These examples make use of the Days of the Week tiddler.

```
[list[Days of the Week]] +[move[Wednesday]]
```

Try it

```
[list[Days of the Week]] +[move:-2[Wednesday]]
```

Try it

multiply Operator

11th June 2019 at 2:58pm

Binary Mathematics Operators Filter Operators Mathematics Operators

purpose	treating each input title as a number, multiply it by the numeric value of the parameter
input	a selection of titles
parameter	`N` = a number
output	the input as numbers, but with each one multiplied by `N`

Learn more about how to use Filters

ⓘ **Introduced in v5.1.20** See Mathematics Operators for an overview.

Compare with the product Operator.

Examples

multiply Operator (Examples)

11th June 2019 at 2:57pm

multiply Operator Operator Examples

```
[[23]multiply[19]]
```

Try it

```
=1 =2 =3 =4 +[multiply[4]]
```

Try it

MultiTiddlerFiles

21st June 2015 at 8:21pm

TiddlyWiki on Node.js

MultiTiddlerFiles allow multiple tiddlers to be concisely represented in a single text file.

The goals of this format are:

- To be easy to type and easy to read
- Optimised for single line strings
- To allow common fields or tags to be shared within groups of tiddlers
- To be simple to process with external tools

MultiTiddlerFiles have the extension `multids`. The file is structured as a block of shared fields followed by a blank line. The rest of the file is a

sequence of comments and tiddlers. Tiddlers are specified by their title, followed by a colon, at least one space character, and then the rest of the line is the text field for the tiddler.

For example:

```
title: $:/language/ControlPanel/
tags: strings
modifier: JoeBloggs

Basics/Caption: Basics
# This is a comment
Basics/Version: ~TiddlyWiki Version
```

This example defines two tiddlers, **$:/language/ControlPanel/Basics/Caption** and *$:/language/ControlPanel/Basics/Version*.

If a `title` field is specified in the header then it is treated as a prefix for the individual tiddlers defined in the title.

Syntax Specification

MultiTiddlerFiles

TypeError: setting getter-only property "parentNode"

Header Line

TypeError: setting getter-only property "parentNode"

Tiddler Descriptor Line

TypeError: setting getter-only property "parentNode"

Comment Line

TypeError: setting getter-only property "parentNode"

MultiTiddlerFileSyntax

21st June 2015 at 8:14pm
TiddlyWiki on Node.js

MultiTiddlerFiles

TypeError: setting getter-only property "parentNode"

Header Line

TypeError: setting getter-only property "parentNode"

Tiddler Descriptor Line

TypeError: setting getter-only property "parentNode"

Comment Line

TypeError: setting getter-only property "parentNode"

Named Filter Run Prefix

11th July 2023 at 11:08am

Filter Run Prefix

In ⓘ **Introduced in v5.1.23** the named filter run prefixes where implemented. `:cascade`, `:map` and `:sort` have been added later as shown in the diagrams.

A named filter run prefix can precede any run of a filter expression in place of a shortcut run prefix.

TypeError: setting getter-only property "parentNode"

> Compare named filter run prefix `:filter` with filter Operator which applies a subfilter to every input title, removing the titles that return an empty result from the subfilter

> Compare named filter run prefix `:reduce` with reduce Operator which is used to used to flatten a list of items down to a single item by repeatedly applying a subfilter

> Within the filter runs prefixed with `:reduce`, `:sort`, `:map` and `:filter`, the **`currentTiddler`** variable is set to the title of the tiddler being processed.
> The value of currentTiddler outside the subfilter is available in the variable **`..currentTiddler`** ⓘ **Introduced in v5.2.0**

Also see: Interchangeable Filter Run Prefixes

NamedCommandParameters

18th November 2021 at 12:42am

TiddlyWiki on Node.js

Introduced in v5.1.18 Most TiddlyWiki Commands use a position-based system for their parameters where each parameter must be listed in the precise order defined by the command. Some of the more complex commands offer an alternative scheme of named command parameters. For example, here we provide two parameters named "port" and "host":

```
--listen port=8090 host=0.0.0.0
```

Note that the order of the parameters does not matter.

Using special characters within a parameter requires quoting. Unix, Linux and the Mac use single quotes, and Windows uses double quotes:

```
--listen port=8090 username=joe 'password=s3cret(!'
--listen port=8090 username=joe "password=s3cret(!"
```

Note that the quotes are applied to the entire name=value pair, not just to the value part.

namespace Variable

28th February 2015 at 2:37pm

Core Variables Variables

The **namespace** variable is used internally by HTML widgets to keep track of the correct namespace for the DOM elements they generate.

An `svg` or `math` element supplies a distinct namespace for itself and any child elements it contains.

Naming of System Tiddlers

2nd February 2024 at 1:10pm

SystemTiddlers

The system tiddlers provided as part of the core are named according to the following rules:

Namespace	Format	Description
`$:/*`	CamelCase	Root user interface tiddlers (eg control panel, advanced search)
`$:/config/*`	CamelCase	User-oriented configuration setting
`$:/core/images/*`	hyphen-case	Core images
`$:/core/modules/*`	lowercase	JavaScript module tiddlers
`$:/core/save/*`	lowercase	Saving templates for creating TiddlyWiki documents
`$:/core/templates/*`	*inconsistent*	Templates needed for TiddlyWiki to operate. Currently uses a mix of dashes and periods to separate words
`$:/core/ui/*`	*inconsistent*	Tiddlers comprising the default user interface of TiddlyWiki. Currently uses a mix of CamelCase and lowercase naming conventions
`$:/core/wiki/*`	lowercase	Metadata about the entire wiki
`$:/docs/*`	lowercase	Documentation tiddlers
`$:/messages/*`	CamelCase	System messages
`$:/plugins/*`	lowercase	Plugin tiddlers, and plugin content
`$:/snippets/*`	*inconsistent*	Reusable snippets (will be replaced by macros)
`$:/state/*`	lowercase	User interface state tiddlers (see StateMechanism)
`$:/tags/*`	CamelCase	User interface configuration tags

`$:/temp/*`	lowercase	Temporary tiddlers that shouldn't be saved
`$:/themes/*`	lowercase	Theme plugins

In the format column:

- **hyphen-case** refers to joining multiple lowercase words with hyphens
- **CamelCase** refers to directly joining multiple lowercase words with initial uppercase letters
- **lowercase** refers to directly joining multiple lowercase words
- **inconsistent** marks namespaces that are currently titled inconsistently

Navigating between open tiddlers

7th June 2016 at 4:52pm

Working with TiddlyWiki

In the default "classic" storyview mode, open tiddlers are displayed in a vertical column called the "story river". There are a number of ways you can navigate the story river - that is, how you can jump back and forth between the open tiddlers.

- The best approach is to **use the Open tab** in the sidebar to click on the open tiddler to which you wish to navigate.
  - The **Open** tab contains a list of all the open tiddlers. You can click on any tiddler in the list to jump to it, or click the "×" next to a tiddler link to close it. There is also a handy ⛓ **close all** button at the bottom of the list of open tiddlers.
- Another way is to simply **scroll the page up and down** using the story river scrollbar to the right.
  - Note that when both the story river and the sidebar extend below the visible screen, there will be *two* scrollbars. The outer or far-right scrollbar controls the story river. The inner scrollbar controls the sidebar.
- A clunky approach that many new users try is to **close tiddlers one by one** until they get to the tiddler they are looking for.

- Closing any tiddler at the top or in the middle of the story river will cause all the tiddlers below it to slide up the river. The tiddler immediately below the tiddler you closed will slide up to take its place.
- Closing the bottom tiddler will close it and the bottom of the tiddler above it will come into view.

NavigatorWidget

27th November 2021 at 3:39am

MessageHandlerWidgets Widgets

Introduction

The navigator widget manipulates the current store, the story list and history lists in response to various Messages.

Content and Attributes

The navigator widget displays any contained content, and responds to Messages dispatched within it.

Attribute	Description
story	Name of the tiddler containing the story list to be manipulated
history	Name of the tiddler containing the history list to be manipulated
openLinkFromInsideRiver	Determines the location for opening new tiddlers from links within the story river: at the **top** or **bottom** of the story river, or **above** or **below** the current tiddler
openLinkFromOutsideRiver	Determines the location for opening new tiddlers from links outside the story river: at the **top** or **bottom** of the

	story river
relinkOnRename	ⓘ **Introduced in v5.1.14** Determines if tags in other tiddlers are renamed if the tiddler title changes. **yes** or **no**. default: **no**

Widget Messages

The following Messages are handled by the navigator widget:

- WidgetMessage: tm-cancel-tiddler
- WidgetMessage: tm-close-all-tiddlers
- WidgetMessage: tm-close-other-tiddlers
- WidgetMessage: tm-close-tiddler
- WidgetMessage: tm-delete-tiddler
- WidgetMessage: tm-edit-tiddler
- WidgetMessage: tm-import-tiddlers
- WidgetMessage: tm-new-tiddler
- WidgetMessage: tm-perform-import
- WidgetMessage: tm-relink-tiddler
- WidgetMessage: tm-rename-tiddler
- WidgetMessage: tm-save-tiddler

Variables

The following Variables are assigned by the navigator widget:

- **tv-story-list**, giving the name of the tiddler containing the story list
- **tv-history-list**, giving the name of the tiddler containing the history list

Examples

See Creating SubStories.

negate Operator

11th June 2019 at 7:40pm

Filter Operators Mathematics Operators Unary Mathematics Operators

purpose	calculate the negation of a list of numbers
input	a selection of titles
output	the negation of the input numbers

Learn more about how to use Filters

Introduced in v5.1.20 See Mathematics Operators for an overview.

Examples

negate Operator (Examples)

11th June 2019 at 7:41pm

negate Operator Operator Examples

```
[[-2000]negate[]]
```

Try it

```
=1 =2 =3 =4 =5 +[negate[]]
```

Try it

New Release Banner

picture



Newnham Horizon.jpg

external-image picture

next Operator

3rd February 2015 at 7:59pm

Field Operators Filter Operators Order Operators

purpose	find which titles in a **`list`** field follow the input ones
input	a selection of titles
parameter	**`T`** = a tiddler title
output	a selection containing each title that immediately follows each of the input titles in the **`list`** field of **`T`**

Learn more about how to use Filters

Each input title is processed in turn, and its successor is located in the **`list`** field and appended to the output. If a title is not listed in the field, or is the last title there, then it contributes nothing to the output.

Examples

next Operator (Examples)

23rd January 2015 at 11:18pm

next Operator Operator Examples

These examples make use of the Days of the Week tiddler.

```
[[Monday]next[Days of the Week]]
```

Try it

```
[[Sunday]next[Days of the Week]]
```

Try it

```
Tuesday Wednesday Thursday +[next[Days of the Week]]
```

Try it

node-webkit

20th November 2021 at 11:17pm

Definitions

"node-webkit" was the original name for the project now known as NW.js.

Node.js

12th June 2022 at 11:18am

Definitions

Node.js is a downloadable application for your PC, Mac or Linux computer that lets it run JavaScript applications. Unlike JavaScript applications running in a web browser, Node.js code has full access to the file system and other resources of the computer, enabling it to perform the roles that have traditionally been the preserve of languages like Java, PHP and Python.

For TiddlyWiki, Node.js means that we can have a single code base that can run in the browser or on the server, giving great flexibility in how it is used.

For end users, Node.js is no more complicated to install than a web browser, but unlocks powerful capabilities such as the ability to run TiddlyWiki as a web server that you can connect to from other devices.

Node.js is a **trademark** of the OpenJS Foundation. See: https://nodejs.org fore more details.

TiddlyWiki on Node.js will give an overview about the possibilities using a client-server configuration.

Notes for upgrading to 5.0.11-beta

16th May 2014 at 5:02pm

Release 5.0.11-beta

Version 5.0.11-beta includes some changes that can break content from earlier releases of TiddlyWiki 5.

Command line changes

Previously, commands that generate output files would interpret the specified path to the file as being relative to the current working directory. So, for example, the following command would write `index.html` to the current directory:

```
tiddlywiki mywiki --rendertiddler $:/core/save/all index.html text/plain
```

In 5.0.11-beta this behaviour has changed, and now the specified filename is resolved relative to an `output` folder within the *TiddlyWikiFolder*. So the command above will now write the file `index.html` to `mywiki/output/index.html`.

You can override this behaviour with the OutputCommand. For example, to generate the `index.html` file within the current directory:

```
tiddlywiki mywiki --output . --rendertiddler $:/core/save/all index.html text/plain
```

A further change is that the `--rendertiddlers` command now clears the output folder before it writes any files. This means that any previous `--rendertiddler` commands to the same folder will have their output deleted.

Notes for upgrading to 5.0.8-beta

23rd February 2014 at 8:55pm

Release 5.0.8-beta

Version 5.0.8-beta includes some changes that can break content from earlier releases of TiddlyWiki 5.

Change to $:/SiteTitle and $:/SiteSubtitle

You should rename any existing *SiteTitle* and *SiteSubtitle* tiddlers to $:/SiteTitle and $:/SiteSubtitle respectively.

Changed parsing rules for content of HTML elements

Version 5.0.8-beta marks a change in the way that TiddlyWiki determines whether to parse the content of an HTML element or widget in *block mode* or *inline mode*.

- In block mode, TiddlyWiki parses text into paragraphs, creating `<p>` tags to wrap them. It also recognises block syntax like headings, lists and tables.
- In inline mode, TiddlyWiki ignores paragraph formatting, and just recognises character formatting, like bold and italic.

It's important to be able to control which type of parsing is performed for different situations.

Prior to 5.0.8-beta, TiddlyWiki parsed the content of an element in inline mode unless the opening tag of the element were immediately followed by a line break. This meant that much of the time element tags would be shunted together into a long line, hindering readability.

The new behaviour for 5.0.8-beta is to parse the content of an element in inline mode unless the opening tag is immediately followed by two line breaks.

To adjust existing content for 5.0.8-beta you will need to manually add the additional line break after the opening tag of elements and widgets whose content should be parsed in block mode.

The positive aspect of the change is that it removes the need to pack multiple HTML tags onto a single line, improving readability.

Examples

Consider the difference between these two examples. First, here's an HTML tag that starts with two line breaks:

copy to clipboard

```
<blockquote>

! This is a heading

And a paragraph of text.

</blockquote>
```

That renders as:

> # This is a heading
>
> And a paragraph of text.

... and the underlying HTML is:

```
<blockquote><h1 class="">This is a heading</h1><p>And a paragraph of text.</p></blockquote>
```

Secondly, here's an HTML tag with just a single line break. Notice how the heading is no longer recognised as a heading

copy to clipboard

```
<blockquote>
! This is a heading

And a paragraph of text.
</blockquote>
```

That renders as:

> ! This is a heading And a paragraph of text.

... and the underlying HTML is:

```
<p><blockquote>
! This is a heading

And a paragraph of text.
</blockquote>
</p>
```

Changed commands for TiddlyWiki on Node.js

The handling of wiki folders has changed. Previously, if the `tiddlywiki` command was run against a wiki folder that didn't have the necessary `tiddlywiki.info` file then it would be automatically created. Now, the wiki folder must be initialised with the InitCommand.

This is how to create and start a new server-based wiki:

```
tiddlywiki mywikifolder --init server
tiddlywiki mywikifolder --server
```

Note that the name of the **clientserver** edition has changed to **server**.

Notifications

7th January 2016 at 11:58pm

Features

Notifications are small informational messages that appear at the top right of the TiddlyWiki window, and automatically disappear after a preset time.

Display sample notification

now Macro

29th June 2021 at 11:50pm

Core Macros Macros

The **now** macro returns the current date and time in a specified format.

The value doesn't update automatically, like a ticking clock. It updates whenever the macro call is rendered, such as when the containing tiddler's display needs to be refreshed for some other reason.

Parameters

format

A string specifying the desired format, defaulting to `0hh:0mm, DDth MMM YYYY`

Note: The format string `[UTC]YYYY0MM0DD0hh0mm0ssXXX` will return a date string representing the UTC time-stamp as it is used in the TiddlyWiki `created` and `modified` time-stamp fields.

Introduced in v5.2.0

You can now pass literal parameters to the `now` macro in filters. For example, this filter will return all tiddlers created today:
`[all[tiddlers]] :filter[get[created]format:date[YYYY0MM0DD]match<now YYYY0MM0DD>]`

Examples

now Macro (Examples)

21st February 2015 at 6:04pm

Macro Examples now Macro

copy to clipboard

```
<<now>>
```

Try it

copy to clipboard

```
<<now YYYY-0MM-0DD>>
```

Try it

copy to clipboard

```
<<now "hh:0mm:0sspm">>
```

Try it

nsort Operator

3rd February 2015 at 8:00pm

Field Operators Filter Operators Negatable Operators Order Operators

purpose	sort the input by number field
input	a selection of titles
parameter	**F** = the name of a field, defaulting to **title**
output	the input, sorted into ascending order by field **F**, treating field values as numbers
! **output**	the input, likewise sorted into descending order

Learn more about how to use Filters

Non-numeric values are treated as having a higher value than any number, and the difference between capital and lowercase letters is ignored. Compare **nsortcs**.

Examples

nsort Operator (Examples)

23rd January 2015 at 11:21pm

nsort Operator Operator Examples

```
10 010 1000 100 +[nsort[]]
```

Try it

```
10 010 ALPHA beta alpha 1000 100 +[nsort[]]
```

Try it

nsortcs Operator

17th April 2015 at 2:57pm

Field Operators Filter Operators Negatable Operators Order Operators

purpose	sort the input titles by number field, treating upper and lower case as different
input	a selection of titles
parameter	**F** = the name of a field, defaulting to **title**
output	the input, sorted into ascending order by field **F**, treating field values as numbers
! **output**	the input, likewise sorted into descending order

Learn more about how to use Filters

Non-numeric values are treated as having a higher value than any number, and capital and lowercase letters are treated as different. Compare **nsort**.

Examples

nsortcs Operator (Examples)

23rd January 2015 at 11:21pm

nsortcs Operator Operator Examples

```
10 010 1000 100 +[nsortcs[]]
```

Try it

```
10 010 ALPHA beta alpha 1000 100 +[nsortcs[]]
```

Try it

nth Operator

29th October 2021 at 4:37am

Filter Operators Order Operators

purpose	select the **N** th input title
input	a selection of titles
parameter	**N** = an integer, defaulting to 1

output	the **N** th input title

Learn more about how to use Filters

N is one-based. In other words, `nth[1]` has the same effect as the **first** operator.

See **zth** for an equivalent operator with a 0 based parameter

Examples

nth Operator (Examples)

23rd January 2015 at 11:29pm

nth Operator Operator Examples

These examples make use of the Days of the Week tiddler.

```
[list[Days of the Week]nth[]]
```

Try it

```
[list[Days of the Week]nth[5]]
```

Try it

NW.js

10th March 2015 at 5:50pm

Definitions

NW.js is an OpenSource application that fuses the functionality of a web browser with that of Node.js. It makes it possible to use web applications as though they were native apps, with full access to the file system and other native facilities. Learn more from the official site or this introductory blog post.

See *TiddlyWiki on NW.js* for details of how to use TiddlyWiki with NW.js.

OfficialPlugins

12th June 2022 at 10:23am

Plugins

The official TiddlyWiki plugin library contains the plugins that are part of the main TiddlyWiki 5 GitHub repository. Install them from the plugin library. Plugins from the official plugin library are automatically updated as part of the upgrade process.

- Amazon Web Services Plugin
- BrowserStorage Plugin
- CodeMirror Plugin
- Consent Banner Plugin
- D3 Plugin
- Dynaview Plugin

- External Attachments Plugin
- Highlight Plugin
- Innerwiki Plugin
- JSZip Plugin
- KaTeX Plugin
- Markdown Plugin
- Mobile Drag And Drop Shim Plugin
- Railroad Plugin
- SaveTrail Plugin
- Share Plugin
- TW2Parser Plugin
- Twitter Plugin

Note

> An overview, how to handle plugins can be found at: Plugins

Open Collective

4th December 2022 at 5:56pm

About HelloThere

Open Collective is a platform for transparent fundraising and expenses for projects like TiddlyWiki. It is the official TiddlyWiki community fundraising space.

https://opencollective.com/tiddlywiki

You can make a fixed one-time donation, or setup a recurring contribution.

The main goals listed for donations are to cover basic costs for community infrastructure like the Discourse forum, and a new goal for supporting the TiddlyWiki Core.

Additionally, the community can make use of the platform for special projects - to pool funds to pay for development, design, or anything else. The File Upload Plugin is the first of these, and we want to welcome others to launch projects here.

Open Collective Logo

picture



OpenSource

25th August 2013 at 3:07pm

Definitions

OpenSource is defined by Wikipedia as *a philosophy, or pragmatic methodology that promotes free redistribution and access to an end*

product's design and implementation details.

Or Filter Run Prefix

22nd March 2023 at 3:07pm

Named Filter Run Prefix

purpose	de-duplicated union of tiddler sets
input	all titles from previous filter runs
output	output titles are dominantly appended to the output of previous filter runs

Introduced in v5.1.23

TypeError: setting getter-only property "parentNode"

The :or prefix is equivalent to using no prefix at all. See `run` at Shortcut Filter Run Prefix

Order of Tagged Tiddlers

29th July 2022 at 4:15pm

Tagging

When TiddlyWiki generates a list of the tiddlers that have a particular tag (e.g. the dropdown list of a tag pill), it orders the tiddlers using the following rules:

1. Start with any that are declared in the **`list`** field of the tag tiddler, in the order given there.
2. In each remaining tiddler **`T`**, look for a **`list-before`** field. If this has a tiddler title as its value, place **`T`** just *BEFORE* that one.
    - As a special case, if the field exists but its value is empty, place **`T`** at the very start of the list.

3. In each remaining tiddler **T**, look for a **list-after** field. If this has a tiddler title as its value, place tiddler **T** just *AFTER* that one.
    - As a special case, if the field exists but its value is empty, place **T** at the very end of the list.
4. If any tiddlers still remain, place them at the end of the list in ascending alphabetical order of title. The difference between capital and lowercase letters is ignored.

The ordering of tiddlers with a particular tag can be directly modified using drag and drop within a tag pill dropdown. The underlying list-tagged-draggable Macro can also be used elsewhere.

order Operator

28th November 2016 at 8:12pm

Common Operators Field Operators Filter Operators Order Operators

purpose	selectively reverse the input list
input	a selection of titles
parameter	**F** = Either the string `reverse` or another value
output	the input, with the order reversed if **F** is the special value `reverse`

Learn more about how to use Filters

Either reverses or preserves the order of the input list according to whether the parameter is the special value `reverse`.

Examples

Osmosoft

19th November 2021 at 1:46am

Definitions

Founded in 2004 by JeremyRuston, Osmosoft was originally a consultancy for software services around TiddlyWiki. Notable engagements included working with Socialtext on Socialtext Unplugged.

In 2007, Osmosoft was acquired by BT and became the champions for open source within the enterprise. As part of BT, Osmosoft has worked on a diverse range of projects within BT and for BT's customers.

See http://osmosoft.com/

Other Resources

1st January 2021 at 4:25pm

Community

This list contains any usefull resource that have been shared with the TiddlyWiki Community that doesn fit the previous categories. They may be, macros, widgets, code snippts, ideas or more.

"TiddlyWiki Posts" by Jeffrey Kishner

Posted 7 months ago

A collection of articles covering integration with Fargo, Font Awesome and Google Calendar, and tips for managing task lists. The original site is missing, but a link to an archive is provided.

https://web.archive.org/web/20221015011644/http://blog.jeffreykishner.com/tiddlywiki/

> I have become a regular user of TiddlyWiki and have become so accustomed to using Font Awesome icons in Fargo that I wanted to

> incorporate the icons into my wikis as well.

"TW5 Magick" by Stephan Hradek

Posted 2 years ago

Stephan Hradek's growing catalogue of tips, tutorials, and other resources. Stephan has also developed several plugins and macros that you can find at http://tiddlystuff.tiddlyspot.com/.

http://tw5magick.tiddlyspot.com/

> Some things in TiddlyWiki seem like Magic. As my preferred nick is Skeeve, I felt that TW5 Magick is a proper name for this collection of some of the "magic" tricks one can do with TiddlyWiki.

TiddlyMap Plugin by Felix Küppers

Posted 2 years ago

An interactive network visualisation plugin based on Vis.js. A demo that also contains installation instructions can be found here: http://tiddlymap.org. The plugin's GitHub repository can be found here.

> TiddlyMap is a TiddlyWiki plugin that allows you to link your wiki-topics (tiddlers) in order to create clickable graphs. By creating relations between your topics you can easily do the following:
>
> - Create concept maps and quickly manifest your ideas in tiddlers.
> - Create task-dependency graphs to organise and describe your tasks.
> - Visualise your topic structures to get an immediate grasp of topics and relations.
>
> In general you may create, visualise and describe any network-structure you have in mind.

"TiddlyServer" by Matt Lauber

Posted 2 years ago

Note: This is an older version of TiddlyServer. A modern version can be referenced here.

TiddlyServer is a special purpose Desktop app, designed to facilitate managing multiple instances of TiddlyWiki running as a server. It does not require internet acess to access the wikis.

https://github.com/mklauber/TiddlyServer/releases/

> *TiddlyServer* can import both TiddlyWiki files and *TiddlyFolder* wikis. For each wiki, you specify a prefix to serve it with and the source to import from. It will copy the wikis to its own internal store and begin serving them up at http://localhost:8080/{prefix}/. The export button for each wiki will convert it to a single file wiki.

"muritest" by Simon Huber

Posted 2 years ago

A terrific demo of drag and drop, pan and zoom, and other sophisticated touch effects. Based on the *muuri plugin by Uwe Stuehler*.

http://muritest.tiddlyspot.com

"Reveal.js" by Devin Weaver

Posted 2 years ago

Embed Reveal.js presentations inside your TiddlyWiki.

https://sukima.github.io/tiddlywiki-reveal-js/

> reveal.js enables you to create beautiful interactive slide decks using HTML.
>
> With the power of TiddlyWiki and Reveal.js one can…
>
> - …make an interactive tiddler with fancy animations
> - …Make it go full screen for professional keynote
> - …Distribute the slideshow with examples and additional resources in one HTML file

"TW Icons" by morosanuae

Posted 2 years ago

> A large collection of icons for TiddlyWiki.

https://morosanuae.github.io/tw-icons

"Noteself" by Danielo Rodríguez

Posted 2 years ago

NoteSelf is your personal, private, customisable, Evernote-like experience. You want cloud? Fine! You don't? Fine too! It's all yours, It's your decision! Why not be a bit selfish?

https://noteself.github.io/

> NoteSelf is the perfect place to store your ideas, notes, thoughts, tips, tricks, recipes... whatever you want to put into it! NoteSelf will store, index, and make it easily searchable, so you will find it instantly whenever you need it!
>
> You already know and love Evernote, we know it. It is comfortable, it syncs, has search capabilities... Wow, it's almost perfect, but what about:

> - Privacy - Everything is stored on the Evernote's servers!
> - Customisation - If you don't like the interface of Evernote, there's nothing you can do
>
> NoteSelf is built on top of TiddlyWiki, a powerful, free, highly customisable and open-source personal wiki. We took the best of it, it's powerful customisation system, and mixed it with one of the best embedded databases available, PouchDb, for bringing the synchronisation capabilities you need.

"Dynamic Tables" by Jed Carty

Posted 2 years ago

Dynamic Tables is a set of macros that allow you to create tables where the rows are created from tiddlers and the columns from the tiddler fields, or where the columns are created from tiddlers and the rows from their fields. Individual cells can be edited on the fly. Row tables can be sorted by columns and column tables can be sorted by rows.

http://ooktech.com/jed/ExampleWikis/DynamicTables/

"Grok TiddlyWiki" by Soren Bjornstad

Posted 2 years ago

This new textbook from Soren Bjornstad is highly recommended for learning TiddlyWiki. The presentation and design are also a first class example of using TiddlyWiki.

From the site:

> Grok TiddlyWiki is a textbook that helps you build a deep, lasting understanding of and proficiency with TiddlyWiki through a combination of detailed explanations, practical exercises, and spaced-repetition reviews of prompts called takeaways.

https://groktiddlywiki.com/read/

TW5-firebase: TiddlyWiki5 for Google Firebase by Peter Neumark

Posted 3 years ago

Google-Firebase hosted version of TiddlyWiki5.

https://github.com/neumark/tw5-firebase

> I've been using TiddlyWiki5 with Google Firebase for over a year now. It's stable enough to use on a daily basis. There's a detailed walkthrough of how to create your own instance with lots of screenshots. The entire process takes less than 30 minutes: https://neumark.github.io/tw5-firebase/
>
> Please consider this version a proof of concept rather than a polished product! My focus was on multi-device or small-team collaboration:
>
> - Basic authorisation (bags have an access policy determining who can read / write them).
> - Multiple wikis can be hosted under a single firebase account.
> - It uses Firebase's built-in social auth to log in users.
> - Tiddlers are written individually with locking, preventing users overwriting each others' updates.

Widdly by Opennota

Posted 3 years ago

A cross platform server application that can save tiddlers to a local database

https://gitlab.com/opennota/widdly

> a minimal self-hosted app, written in Go, that can serve as a backend for a personal TiddlyWiki.

twproxy by Steve Gattuso

Posted 3 years ago

An authentication proxy for your TiddlyWiki.

https://github.com/stevenleeg/twproxy

> One of my favorite aspects of TiddlyWiki has been that it's web-based, making it possible for me to access my wiki from everywhere (especially my phone). That being said, I'm a bit paranoid, so I was left a bit unsatisfied with the tiddlywiki server's HTTP basic auth for protecting my wiki. ~23 commits later, I've created something called twproxy that I'd like to share with you all today, as I'm hoping somebody other than myself will find it useful.
>
> Essentially it is a simple proxy that puts your wiki behind a username, password, and optional 2-factor auth prompt. This gives you added security in addition to the ability to remember your credentials past one browsing session (I was getting sick and tired of typing my username/password in over and over using basic auth).

Twexe: Single File Tiddlywiki5 executable

Posted 3 years ago

Single File Tiddlywiki5 executable

https://ihm4u.github.io/twexe/

Features

- Automatic backups
- Automatic saving
- No browser add-ons needed!!
- Works on Linux and Windows
- Single file executable, can be moved, copied, etc.
- Wiki is compressed, occupying less space on your disk (sometimes even a third of the space!)

- Converts *any* Tiddlywiki5 file to a single file executable , no need for special tiddlywiki plugins
- Automatically uses the saving tab in the control panel (there you can specify a backup directory if you don't like the default)
- Easily get back the html wiki file from the executable twexe (if you need it for some reason)
- Opens up the possibility to run external executables from your wiki (to draw charts, etc.) - stay tuned!!
- Supports external images (i.e. with _canonical_uri field)

TWCommunitySearch

Posted 3 years ago

The CommunitySearch wiki aggregates many public wikis from the community. It is updated automatically every day. At the time of writing, more than 5000 tiddlers are indexed, thus giving access to the largest known collection of TW content.

It lets you search for some specific content among a preselected list of wikis. The indexed content is mostly focused on TW usage, so you would typically use this search system to find information about a particular aspect of TiddlyWiki. Say for example that you want to learn how to make a table of content, type "table of content" in the CommunitySearch box (also accessible in the default tiddler GettingStarted). The search results are links to various wikis, and point directly to the specific tiddlers containing your request.

The CommunitySearch wiki also lets you:

- Browse the list of wikis
- Browse the list of author
- Browse the list of plugins
- Access content by tag (see also BookmarkingTags)
- Follow the latest news from the community (and publish your own news if you are a wiki author)

TW5-SingleExecutable by Jed Carty

Posted 3 years ago

TiddlyWiki5 Packaged with the multi-user plugin in a single executable file.

https://github.com/OokTech/TW5-SingleExecutable

> Tiddlywiki, node js, multiuser plugin - all packaged into a single file so user can just download the file and run it and not have to install anything else. Versions available for windows, osx, and linux.

TW-Scripts by Mohammad

Posted 3 years ago

TW-Scripts is one of the most comprehensive collections of solutions for Tiddlywiki 5.

https://kookma.github.io/TW-Scripts/

TW-Scripts includes:

- Collected solutions to questions in Tiddlywiki Google groups
- Learn through examples
- Templates, stylesheets
- Wikitext, macros, and snippets
- Tips and tricks
- Search tools

Timimi: WebExtension and Native Host by Riz

Posted 3 years ago

Timimi is a web-extension accompanied by a native host that allows you to save and backup your standalone HTML tiddlywiki files **anywhere in your hard-drive**. Once installed, you can save the tiddlywiki files without any extra steps, like the original Tiddlyfox addon.

- The native host requires a component installed on the host computer, outside the browser.

https://ibnishak.github.io/Timimi/

As of version 2.1, Timimi supports the following browsers

- Chrome/Chromium
- Firefox
- Opera
- Microsoft Edge (Chromium)

It is also reported to work seamlessly in chrome based browsers like Brave and Vivaldi.

Timimi also provides users with 4 backup strategies, viz:

- Create a backup Every n^{th} save
- Create a backup every n^{th} minute
- Customised Tower of Hanoi
- First in First Out

TiddlyWiki extensions for ViM

Posted 3 years ago

Vi iMproved syntax file for TiddlyWiki5 `*.tid` files.

https://github.com/roma0104/vim-tid

TiddlyWiki extensions for Sublime Text 3 by roma0104

Posted 3 years ago

Sublime Text 3 syntax file for TiddlyWiki5 `*.tid` files.

https://github.com/roma0104/sublime-tid

TiddlyServer by Arlen Beiler

Posted 3 years ago

An extension to the Node.js configuration of TiddlyWiki that adds support for static attachments and for working with multiple wikis at the same time.

https://github.com/Arlen22/TiddlyServer

> *TiddlyServer* 2.0 takes the server command of TiddlyWiki on *NodeJS* and adds it to a static file server. This means you can load and serve any TiddlyWiki data folder in the same way you can serve a single file TiddlyWiki.
>
> But you don't need to serve files and folders from just one place, you can serve them from multiple places anywhere on your harddrive (literally anywhere *NodeJS* can stat, readdir, and readFile). You can even organise them into virtual folders (aka aliases in Apache and mounts in Express).
>
> The main point, of course, is that you can actually edit your files, not just look at them. Single file *TiddlyWikis* use the put saver, which needs to be patched using a bookmarklet included on the index page. The instructions for this are below under the heading "One thing that needs to be noted".
>
> And, of course, you can edit data folder tiddlywikis just like you were running node tiddlywiki.js data –server, except that you run it on the path that you found it at (e.g. http://localhost/personal/notes/). You can have as many data folders open as you want, they don't conflict (though they will each take memory).
>
> Data folders store individual tiddlers instead of entire wikis. They take less disk space as they also do not store the core and plugins. This means they also save much quicker, especially over the internet. They also save immediately (within 10 seconds or so) and they save drafts.

TiddlyDrive Add-on for Google Drive by Joshua Stubbs

Posted 3 years ago

An add-on for Google Drive that allows TiddlyWiki files stored there to be opened and saved directly

To find more information, go to the github page of this project: https://github.com/tiddlydrive/tiddlydrive.github.io

> I made an app that lets one edit TiddlyWiki files saved in your Google drive and then saving them back automagically. The page is here https://chrome.google.com/webstore/detail/tiddly-drive/oaphhjhbbabdjnpjpilieppphpmnioolo but I think you need to add it from the Google Drive web interface.
>
> I even added an optional ability to save with ctrl + s hotkeys.
>
> *Joshua's launch post*

TiddlyChrome by Arlen Beiler

Posted 3 years ago

An extension application for Google Chrome that enables TiddlyWiki to save changes directly to the file system.

https://github.com/Arlen22/tiddly-chrome-app/

Semantic Colors by Mohammad

Posted 3 years ago

Semantic colors are set of CSS to apply colorful theme to individual tiddlers for categorisation purpose.

https://kookma.github.io/TW-Semantic-Colors/

Tiddlywiki "semantic colors" has two objectives:

1. How it is simply possible to apply theme to an individual tiddler
2. Use semantic colors for categorisation

Some use cases

Learning materials (each topic, category can have dedicated semantic color, like learning a foreign language)
Slideshow (section divider, each part can have a semantic color)
Books and tutorials (different semantic colors can be applied to chapters, specific subjects, like example tiddlers)
Visual tags (instead of looking for tag bar and see to what category this subject (tiddler) belongs, you can identify it by its semantic color)

SeeAlso by Matias Goldman

Posted 3 years ago

A simple technique for adding tag-based cross references to the default view template.

http://seealso.tiddlyspot.com

> See also: - is a feature to generate a "See also:" section after the tiddler text, somewhat like in Wikipedia articles.

Code styles and auto format settings for IDEs

Posted 3 years ago

Implementation of TiddlyWiki Coding Style Guidelines for different IDEs.

https://gitlab.com/bimlas/tw5-code-style-auto-format

"X3DOM for TiddlyWiki 5" by Jamal Wills

Posted 3 years ago

An experiment to embed interactive 3D graphics within TiddlyWiki by Jamal Wills. It is based X3DOM, an open source JavaScript library.

http://x3dom.tiddlyspot.com/

"Wills Q&D gTD" by Matabele

Posted 3 years ago

Matabele's stylish and clean system for personal task management.

http://wills.tiddlyspot.com

> This is my work in progress version of TW5, incorporating various features and ideas as and when they intrigue me. To grab any feature you may wish to try, drag the listed tiddlers across to the dropzone of an empty TiddlyWiki 5.0.10-beta

"Wikilabs" by PMario

Posted 3 years ago

A collection of references for testing concepts around TiddlyWiki 5...

https://wikilabs.github.io

> This is PMario's playground with ramblings about TiddlyWiki and related stuff.
>
> It basically is a "jump off" page to various TiddlyWiki sites, that show editions, themes and plugins.

"TWeb.at" by Mario Pietsch

Posted 3 years ago

A collection of TiddlyWiki resources from Mario Pietsch, with a focus on cloud deployments. Mario also maintains the German translation of TiddlyWiki.

https://tweb.at

> My name is Mario Pietsch from Austria. I'm living near Salzburg.
>
> This page, will be the portal to my TiddlyWiki, TiddlyWeb related content.

"TiddlyWiki5^2 documenting while learning TiddlyWiki5" by Iannis Zannos

Posted 3 years ago

A wealth of hints, tips and notes about using TiddlyWiki on Node.js:

> TiddlyWiki is different from other wikis because of its principle of dynamically customizeable "storyline" based on tiddlers as basic units of

> information. That is, the user "composes" their own version of the webpage by clicking on tiddler links, which add tiddlers to the page in order to compose a storyline.
>
> The Node.js implementation in TiddlyWiki5 adds all the advantages of flat-file markup language based type of site. This makes TiddlyWiki an excellent alternative to flat-file based CMS/webpage/blog authoring systems for the web.
>
> Also very cool is the treatment of tags as menus everywhere.

http://larigot.avarts.ionio.gr/users/iani/wikis/tw5square.html

Also available on GitHub (download and save index.html and open it in your browser).

"TiddlyWiki5 Playground" by Ton Gerner

Posted 3 years ago

Huge collection of customisations and tweaks from Ton Gerner.

http://tongerner.tiddlyspot.com/

> I am a longtime user of TiddlyWiki Classic and still use it daily since you can't do everything in TiddlyWiki 5 (yet) ;)
>
> I started experimenting with TW5 at the end of September 2013 (alpha10). Since I missed a few things in the layout I got used to in TiddlyWiki Classic 1, I started with modifying the layout of TW5. To share my knowledge, I made a few guides about these 'modifications'. The guides started as a non-linear personal web notebook (yeah, the subtitle of TiddlyWiki!). I started experimenting with layout things and used TW5 to document my experiments.

"TiddlyWiki5 Forum on Reddit" by Riz

Posted 3 years ago

A TiddlyWiki5 forum on Reddit.

https://www.reddit.com/r/TiddlyWiki5/

> A TiddlyWiki5 forum on Reddit. As of 2016, it claims to have the most up-to-date list of plugins available for TW5.

"TiddlyWiki5 Bourbon" by mkt_memory

Posted 3 years ago

An extensive collection of tips and resources for users of TiddlyWiki in Japanese.

http://bacchus.ivory.ne.jp/bourbon/

> TiddlyWikiをHTML5にしたTiddlyWiki5が正式にリリースされました。このTiddlyWiki5の使い方について、自分で試した機能を中心に解説します。
>
> TiddlyWiki5は、非常に多機能なローカルWikiです。このブログに書いてあることは、TiddlyWiki5で出来ることのごく一部です。

"TiddlyWiki 舞" by Bram Chen

Posted 3 years ago

A translation of the tiddlywiki.com documentation from Bram Chen, TiddlyWiki's Chinese translator.

http://tw5-zh.tiddlyspot.com

"TiddlyWiki Toolmap" by David Gifford

Posted 3 years ago

A topical index of known plugins, tutorials, themes, and other helpful tools for TiddlyWiki.

https://dynalist.io/d/zUP-nIWu2FFoXH-oM7L7d9DM

Some of the topics indexed so far include:

- Background images and background colors
- Breadcrumbs
- Business - examples
- CSS / Styles, general
- Calendars, dates and timelines
- Checklists
- Collaboration
- Color palettes
- Colors
- Data visualizations
- Developer / advanced tools
- Diplay tiddlers horizontally
- Dropdowns
- Educational
- External files
- Fields
- Fonts
- Fun and games
- Home and lifestyle
- IFrames and embedding media
- Image galleries, modals and lightboxes
- Images, icons, emojis
- Importing and exporting between TiddlyWiki files
- Includes scaffolding within a tiddler, dragging to reorder tiddlers, choosing the number of columns
- Interfacing with, importing and exporting to and from other formats
- Knowledge bases, dictionaries and glossaries

- Layout/UI
- Lightboxes
- Links, linking
- Lists and filters
- Maps and geography
- Masonry, scaffolding
- Math
- MindMapping
- Mobile tools
- Music and sound
- Navigation
- node.js
- Note-taking
- Open the "filter" tab
- Partially hidden tiddlers
- Personal websites (examples)
- Presentations and slideshows
- Printing
- Project management
- Publishing and hosting *TiddlyWikis* on the web
- Searching tools
- Security and backup tools
- Social media, sharing, comments
- Spreadsheets and math
- Startup
- Sticky titles
- Story and screenplay creation
- Table of contents
- Tables, charts and graphs (cf data visualizations)

- Tabs
- Tags, tagging
- Task management and productivity - to do lists
- Themes
- Understanding TiddlyWiki
- Writing and editing

"TiddlyWiki Notes" by James Anderson

Posted 3 years ago

Notes and tips by a developer working on writing TiddlyWiki plugins – including TWExe, a widget for running Windows scripts and executables.

http://welford.github.io

"TiddlyWiki Knowledge Network" by Dmitry Sokolov

Posted 3 years ago

A Knowledge Network about TiddlyWiki.

http://confocal-manawatu.pbworks.com/w/page/113574373/TiddlyWiki

> Interconnected with the visual taxonomy at every node: http://debategraph.org/Stream.aspx?nid=457381&vt=bubble&dc=focus
>
> TWKN will be transferred into TWederation/TiddlyMap format as soon as I am convinced in reliable and secure versioning and multi-user operation in TW format. Your assistance and support would be highly appreciated.

"TiddlyWiki guide FR" by Sylvain Naudin

Posted 3 years ago

A collection of tips and guides in French, introduced by this blog post:

http://sylvain.naud.in/post/2014/10/29/Ressources-pour-TiddlyWiki-5-%28FR%29

> Ce guide de prise en main de TiddlyWiki vous permettra de débuter sur ce logiciel : rédiger du texte avec la syntaxe wiki, insérer des images, créer des Tiddler, faire une recherche, ajouter des fonctionnalités, et bien d'autres choses.. Bonne lecture ! Sylvain

"TB5 - a pocket full of tips" by Tobias Beer

Posted 3 years ago

An enormous collection of tips beautifully curated by Tobias Beer. It is actually a collection of sites, with sub-sites dedicated to topics such as filtering (http://tobibeer.github.io/tw/filters/).

http://tobibeer.github.io/tb5/

> A collection of references and sandbox for testing concepts around TiddlyWiki 5...

"savetiddlers" Extension for Chrome and Firefox by buggyj

Posted 3 years ago

An extension for Google Chrome and Mozilla Firefox that smoothes out some of the friction from TiddlyWiki's built-in HTML5 saver, making it almost as easy to use as TiddlyFox once it is set up correctly.

https://github.com/buggyj/savetiddlers

"MathCell" by Stephen Kimmel

Posted 3 years ago

Mathcell allows spreadsheet-like abilities in a completely contained TiddlyWiki.

http://mathcell.tiddlyspot.com/

> Mathcell allows spreadsheet-like abilities in a completely contained TiddlyWiki. Additional abilities allow it work as a general math engine.

"In My Socks" by Jed Carty

Posted 3 years ago

An extensive collection of hints and tips from Jed Carty.

http://inmysocks.tiddlyspot.com

> I made this so that some of my friends with absolutely no coding experience could use some of the better features of TW5 without having to learn much. I am hopefully going to continue updating it as I learn new things. There isn't really anything here that isn't available in plenty of other places, but I figured that since I spent a while putting it together and people keep saying that TW needs more

> documentation I would post it here.

"Hacks" by Thomas Elmiger

Posted 3 years ago

A collection of helpers by Thomas Elmiger, among them a **tweet button** macro, a **text-stretch** solution, a **snippet extraction** macro and a tiddler that installs a **scroll-to-top button** via drag-and-drop.

http://tid.li/tw5/hacks.html

> Quick Extensions for TiddlyWiki 5 – […] As I take from the community, I am happy if I can give back. You can use my stuff to do whatever you like, but remember to save a back-up first.
>
> *Thomas*

"Font Awesome 5 Free SVGs for TiddlyWiki" by morosanuae

Posted 3 years ago

The **Font Awesome 5 Free** SVG images collection converted to tiddlers that you can easily import (drag'n'drop) in your wiki.

http://fa5-free-svg.tiddlyspot.com

"file-backups" Extension for Firefox by pmario

Posted 3 years ago

An extension for Mozilla Firefox that smoothes out some of the friction from TiddlyWiki's built-in HTML5 saver, making it almost as easy to use as

TiddlyFox. The workflow is intended to work out of the box, without configuration.

https://github.com/pmario/file-backups which contains links to the documentation and introduction video(s).

A chrome version is planned!

"Dropboard" by Reid Gould

Posted 3 years ago

A TiddlyWiki plugin for kanban-like organisation in the style of Trello™ using boards, lists, and cards.

https://github.com/reidgould/tiddlywiki-dropboard

"CouchDB Adaptor" by William Shallum

Posted 3 years ago

An adaptor to enable TiddlyWiki to sync changes with a CouchDB database.

https://github.com/wshallum/couchadaptor

> *CouchDB* sync adaptor for TiddlyWiki 5. Requires TiddlyWiki >= 5.1.2.
>
> Works in limited testing. Not sure how well it handles conflicts.

"BrainTest - tools for a digital brain" by Danielo Rodriguez

Posted 3 years ago

A collection of customisations and macros from Danielo Rodriguez.

http://braintest.tiddlyspot.com/

> This is my personal tiddlywiki file for testing.
>
> Here I will post the tools and Macros that I develop for this awesome tool. That way, all the help the community gave me can came back to the community again.

"ATWiki" by Lamusia Project

Posted 3 years ago

A desktop application for TiddlyWiki, running on Windows.

https://lamusia.github.io/#ATWiki

> TiddlyWiki for Windows. A unique non-linear notebook for capturing, organising and sharing complex information.

TWaddle by Matias Goldman

Posted 8 years ago

A collection of hints and tips, musings and proposals from long-time TiddlyWiki contributor Mat Goldman.

http://twaddle.tiddlyspot.com/

> This is Mat's, a.k.a <:-) little man-cave in the TiddlyVerse.
>
> Most TW development is, understandably, based on the premise that *"Improve code ➔ Better TW"*. **TWaddle** is also about developing TW but from the perspective that *Bigger community ⇄ Better TW*. Thus, TWaddle looks more to people issues - how to attract them, how to make them stick with TW etc.

> I'm not a programmer but I *am* a TW enthusiast so I tiddlefiddle enough to make the occasional discovery of something cool. Given the amount of words I utter, it is also a mere numbers game before I say something that makes sense. TWaddle is intended to capture these eventualities.

OutputCommand

25th April 2014 at 10:57am

Commands

Sets the base output directory for subsequent commands. The default output directory is the `output` subdirectory of the edition directory.

```
--output <pathname>
```

If the specified pathname is relative then it is resolved relative to the current working directory. For example `--output .` sets the output directory to the current working directory.

OXTWIG

12th September 2014 at 5:03pm

Meetups Videos

The **Oxford TiddlyWiki Interest Group** meets monthly for discussions and demonstrations about TiddlyWiki.

See https://oxtwig.eventbrite.co.uk/ for details of our next meeting.

We have an email discussion list, too: https://groups.google.com/forum/#!members/oxtwig

OXTWIG #2

The second OXTWIG meeting was held on Thursday 16th January 2014:

OXTWIG #1

The first OXTWIG meeting was held on Thursday 21st November 2013:

pad Operator

29th November 2020 at 6:53pm

Filter Operators String Operators

purpose	returns each item in the list padded to the specified length
input	a selection of titles
suffix	(optional). Whether to pad by adding a "suffix" or "prefix". Defaults to "prefix".
parameter	the **pad** operator accepts 1 or more parameters, see below for details

output	the input titles padded to the specified length

Learn more about how to use Filters

Introduced in v5.1.23

The **pad** operator requires at least one parameter which specifies the desired length of the input titles. A second optional parameter can be used to specify the string to pad with (defaults to "0").

```
[pad[<length>],[<padding-string>]]
```

- **length** : the desired final length of the input titles.
- **padding-string**: (optional). The string to use to pad to the desired length. Defaults to "0".

Examples

pad Operator (Examples)

29th November 2020 at 7:00pm

Operator Examples pad Operator

```
[[2]pad[3]]
```

Try it

```
[[2]pad[3],[a]]
```

Try it

```
[[12]pad[9],[abc]]
```

Try it

```
[[12]pad:suffix[9],[abc]]
```

Try it

Paragraphs in WikiText

14th December 2013 at 6:04pm

WikiText

To mark the end of a paragraph in TiddlyWiki you need to type `enter` twice to create a double line break:

```
This is the first paragraph.
```

```
And this is the second paragraph.
```

Single line breaks are ignored within paragraphs. For example:

copy to clipboard

```
This is a
paragraph made
up of
short lines
```

That renders as:

This is a paragraph made up of short lines

... and the underlying HTML is:

```
<p>This is a
paragraph made
up of
short lines</p>
```

For situations where this behaviour isn't convenient, you can also use Hard Linebreaks in WikiText.

ParametersWidget

18th May 2023 at 3:40pm

Widgets

New in v5.3.0 The `$parameters` widget is used within transcluded content to declare the parameters to be made available to the `$transclude` widget.

There are shortcuts for common scenarios that can often make it unnecessary to use the `$parameters` widget directly:

- the Pragma: \parameters
- the Pragma: \procedure for declaring procedure
- the Pragma: \widget for declaring custom widgets

The `$parameters` widget must be used directly in the following situations:

- When the default value of a parameter must be computed dynamically
- When the `$depth` attribute is used to retrieve parameters from a parent transclusion (see below)

Content and Attributes

The content of the `$parameters` widget is the scope within which the values of the parameters can be accessed as ordinary variables.

Attribute	Description
$depth	The index of the parent transclusion from which to obtain the parameters (defaults to 1). See below
$parseMode	Optional name of a variable in which is made available the parse mode of the content of the parent transclusion (the parse mode can be "inline" or "block")

$parseTreeNodes	Optional name of a variable in which is made available the JSON representation of the parse tree nodes contained within the parent transclusion
$slotFillParseTreeNodes	Optional name of a variable in which is made available the JSON representation of the parse tree nodes corresponding to each fill widget contained within the parent transclusion (as an object where the keys are the slot names and the values are the parse tree nodes)
$params	Optional name of a variable in which is made available the JSON representation of the parameters passed to the parent transclusion (as an object where the keys are the parameter names and the values are the coresponding values)
{attributes not starting with $}	Any attributes that do not start with a dollar are used as parameters, with the value specifying the default to be used for missing parameters
{other attributes starting with $}	Other attributes starting with a single dollar sign are reserved for future use
{attributes starting with $$}	Attributes starting with two dollar signs are used as parameters to the transclusion, but with the name changed to use a single dollar sign. The value specifies the default to be used for missing parameters

Note

> Note the special treatment required for parameters names that start with a `$`; this can be avoided by using one of the pragmas

`$depth` Attribute

By default, the **`$parameters`** widget retrieves parameters from the immediate parent transclusion. The `$depth` attribute permits access to the parameters of parent transclusions by specifying an index to the parent to be inspected ("1" is the immediate parent, "2" is the parent of that parent, etc.). This is useful in some situations where an intervening transclusion prevents immediate access to desired parameters.

`$parseMode`, `$parseTreeNodes`, `$slotFillParseTreeNodes` and `$params` Attributes

These attributes provide low level access to the contents of the transcluding widget:

- The `$params` attribute provides access to the raw parameters provided to the transcluding widget. Represented in JSON as an object with keys of the parameter names and values of the corresponding parameter values
- The `$parseMode` attribute contains `block` or `inline` to indicate whether the contents was parsed in block or inline mode
- The `$parseTreeNodes` attribute provides access to the raw parse tree nodes that represent the contents of the transcluding widget. Represented in JSON as an array of parse tree nodes
- The `$slotFillParseTreeNodes` attribute provides access to the raw parse tree nodes corresponding to the filled slots within the contents of the transcluding widget. Represented in JSON as an object with keys of the slot name and values being an array of parse tree nodes

Examples

Example i: Shows transclusion of Sample Tiddler Template. The template tiddler has two parameters `name` and `age` and here their default values are used.

copy to clipboard

```
<$transclude $tiddler="Sample Tiddler Template" />
```

Try it

Example ii: Shows, another transclusion of Sample Tiddler Template, here the value of `age` is passed, but `name` uses its default value.

copy to clipboard

```
<$transclude $tiddler="Sample Tiddler Template" age=33/>
```

Try it

Example iii: Shows, another transclusion of Sample Tiddler Template, here the value of both `name` and `age` are passed.

copy to clipboard

```
<$transclude $tiddler="Sample Tiddler Template" age=45 name="Jeremy Ruston" />
```

Try it

In the simple form the above transclusion is equivalent to

copy to clipboard

```
{{Sample Tiddler Template|Jeremy Ruston|45}}
```

Try it

In this simple form, parameters passed by position not by name. So the first value is passed to the first parameter, here `name` and the second value is passed to the second parameter, here `age`.

Remarks

1. Passing parameter by name is good practice and is recommended for clarity. So for parameterized transclusions, the use of **`$transclude`** is recommended over simple form transclusion.
2. When passing parameters value by position, you cannot pass the second parameter while the first one has not been passed.

Example iv: Here the **`$parameters`** widget is used to declare a parameter whose default value is transcluded from another tiddler.

copy to clipboard

```
\procedure myproc()
<$parameters name={{$:/SiteTitle}} age="21">
My name is <<name>> and my age is <<age>>.
</$parameters>
\end

<$transclude $variable="myproc" age="19"/>
```

Try it

ParametersWidget (Examples)

18th May 2023 at 3:51pm

ParametersWidget

Example i: Shows transclusion of Sample Tiddler Template. The template tiddler has two parameters `name` and `age` and here their default values are used.

copy to clipboard

```
<$transclude $tiddler="Sample Tiddler Template" />
```

Try it

Example ii: Shows, another transclusion of Sample Tiddler Template, here the value of `age` is passed, but `name` uses its default value.

copy to clipboard

```
<$transclude $tiddler="Sample Tiddler Template" age=33/>
```

Try it

Example iii: Shows, another transclusion of Sample Tiddler Template, here the value of both `name` and `age` are passed.

copy to clipboard

```
<$transclude $tiddler="Sample Tiddler Template" age=45 name="Jeremy Ruston" />
```

Try it

In the simple form the above transclusion is equivalent to

copy to clipboard

```
{{Sample Tiddler Template|Jeremy Ruston|45}}
```

Try it

In this simple form, parameters passed by position not by name. So the first value is passed to the first parameter, here `name` and the second value is passed to the second parameter, here `age`.

Remarks

1. Passing parameter by name is good practice and is recommended for clarity. So for parameterized transclusions, the use of **`$transclude`** is recommended over simple form transclusion.
2. When passing parameters value by position, you cannot pass the second parameter while the first one has not been passed.

Example iv: Here the **`$parameters`** widget is used to declare a parameter whose default value is transcluded from another tiddler.

copy to clipboard

```
\procedure myproc()
<$parameters name={{$:/SiteTitle}} age="21">
My name is <<name>> and my age is <<age>>.
</$parameters>
\end

<$transclude $variable="myproc" age="19"/>
```

Try it

PasswordCommand

Commands

Set a password for subsequent crypto operations

```
--password <password>
```

Note: This should not be used for serving TiddlyWiki with password protection. Instead, see the password option under the ServerCommand.

PasswordWidget

30th October 2013 at 9:37am

Widgets

Introduction

The password widget displays a password input box that is bound to a named entry in the TiddlyWiki5 *PasswordVault*. Passwords are currently stored in the browsers local storage and are not themselves encrypted.

Content and Attributes

The content of the `<$password>` widget is ignored.

Attribute	Description
name	Name of the password vault entry

Percent Encoding

28th February 2015 at 2:15pm

Definitions

Percent encoding is a notation that allows otherwise invalid characters to be included in a URI.

Such characters are represented as a percent sign `%` followed by two additional characters.

For example, a space is represented as `%20` and an actual percent sign is represented as `%25`.

Performance

14th October 2019 at 11:19am

Working with TiddlyWiki

TiddlyWiki ships with defaults that are designed to get the best out of modern devices from smartphones to desktop computers. If you need to work on older, less powerful devices, or work with large amounts of content, there are a few steps you can take to improve performance.

Usage

- **Avoid the "Recent" tab**. It is computationally slow to generate and update in response to tiddler changes.
- **Use the "Vanilla" theme**. The default "Snow White" theme includes visual effects like shadows, transparency and blurring that can be slow to render on older devices
- **Avoid large tiddlers**. Large bitmaps can significantly slow TiddlyWiki's performance. For example, an image taken with a modern smartphone will often be 5MB or more. Use ExternalImages whenever possible
- **Don't have too many tiddlers open at once**. Every tiddler you have open will require processing to keep it up to date as the store changes (for example, while you type into a draft tiddler). It is particularly easy when using zoomin story view to end up with dozens of tiddlers listed in the **Open** tab in the sidebar. Get into the habit of periodically closing all open tiddlers with the **close all** button

WikiText

- **Use the built-in performance instrumentation**. Studying the performance instrumentation results can help highlight performance problems
- Take advantage of indexed filter operators. The following constructions at the start of a filter run will be optimised to run many times faster than otherwise:
  - `[all[tiddlers]tag[x]...`
  - `[all[shadows]tag[x]...`
  - `[all[tiddlers+shadows]tag[x]...`
  - `[all[shadows+tiddlers]tag[x]...`
  - `[all[tiddlers]field:y[x]...`
  - `[all[shadows]field:y[x]...`

    - `[all[tiddlers+shadows]field:y[x]...`
    - `[all[shadows+tiddlers]field:y[x]...`
    - Note that the field indexer currently defaults to indexing field values of less than 128 characters; longer values can still be searched for, but no index will be constructed
    - Also note that the "field" operator is also used when the operator name is a fieldname, so, for example, `[all[shadows+tiddlers]caption[x]...` is optimised.
- Use the throttling feature of the RefreshMechanism judiciously
- Keep in mind that **transcluding separate tiddlers is more performant than heavy use of macros** and the difference can be significant in some situations. The result of parsing each tiddler is cached and reused the next time if the tiddler has not changed. The same technique cannot be used for macros and they have to be re-parsed every time, as they are not global but local to the widget tree.
    - ⓘ **Introduced in v5.1.23** Parse trees are now cached for macros that do **not** perform any text substitution either via parameters or variables (i.e. `$parameter$` or `$(variable)$`).
- Where possible **use the** SetWidget **or** VarsWidget **with filters instead of the** WikifyWidget for declaring variables and string concatenation. The performance of the wikify mechanism is relatively poor as there is no opportunity to cache the parse tree or widget tree.

Performance Instrumentation

10th May 2019 at 3:34pm

Features

The core has built-in performance instrumentation for the refresh cycle. See "Performance Instrumentation" on the "Settings" tab of control panel. The wiki will need to be reloaded before it takes effect.

With performance instrumentation switched on, there is additional timing information displayed the browser developer console each time a

refresh cycle is triggered. For example:

```
performance: mainRender: 327.83ms
performance: styleRefresh: 9.14ms
performance: mainRefresh: 68.10ms
```

The information above should be interpreted as follows:

- mainRender is the time taken for the initial display of the page template
- styleRefresh is the time taken to refresh the page stylesheet
- mainRefresh is the time taken to refresh the main page template

As an example, try switching between the sidebar tabs to compare how long they take to render.

More detailed information on filter execution timings is also available. With performance instrumentation enabled, type the following JavaScript command in the browser developer console:

```
$tw.perf.log()
```

A table is displayed with the following information:

- **name** - the string "Filter: " plus the text of the filter
- **invocations** - the number of invocations of the filter since startup
- **totalTime** - the total time in milliseconds spent evaluating this filter since startup
- **percentTime** - the percentage of the execution time of this filter against the total execution time of all filters

PerlinLight.jpg

PermaLinks

6th April 2021 at 3:12pm

Concepts Features

Permalinks allow direct links to individual tiddlers within a TiddlyWiki.

Simple Permalinks

The simplest form of permalink (#) is a single target tiddler title appended to the base URL with `#`:

https://tiddlywiki.com/#HelloThere

The tiddler title can contain spaces if required:

https://tiddlywiki.com/#Using TiddlyWiki on Node.js

Note that double square brackets are not required around the target tiddler title; however, if present they are silently removed.

Story Permalinks

The permalink can also specify the story list of tiddlers that should be opened alongside the target tiddler as a TiddlerFilter:

https://tiddlywiki.com/#TiddlerFields:Tiddlers TiddlerTags TiddlerFields ContentType

If the target tiddler isn't present in the story list then it is automatically inserted at the top. This means that the following two examples both target the tiddler `Tiddlers` within the story sequence `Tiddlers`, `Tags`, `TiddlerFields`:

https://tiddlywiki.com/#Tiddlers:Tags TiddlerFields

https://tiddlywiki.com/#Tiddlers:Tiddlers Tags TiddlerFields

It is also possible to specify a story filter without specifying a target tiddler for navigation:

https://tiddlywiki.com/#:[tag[task]]

About URL encoding

There are technical restrictions on the legal characters in an URL fragment. To allow all tiddler titles to be addressed, illegal characters are subject to a process called "URL encoding" whereby problematic characters are replaced by their numeric code. For example, the space character is replaced with `%20`.

Both the target tiddler title and the story filter should be URL encoded (but not the separating colon). TiddlyWiki generates properly encoded URLs which can look quite ugly. However, in practice browsers will usually perfectly happily process arbitrary characters in URL fragments. Thus when creating permalinks manually you can choose to ignore URL encoding.

Permalink Behaviour

Two important aspects of TiddlyWiki's behaviour with permalinks can be controlled via options in the **control panel** ⚙ **Settings** tab:

- Whether to automatically update the address bar at each navigation, and if so whether to include the story sequence as well as the target tiddler
- Whether the updates to the address bar should affect browser history. The default is **no**; when switched to **yes** you can rewind navigation between tiddlers using the browser back and forward buttons

Note that typing or navigating to a permalink will always cause the permalink to be processed, and tiddlers opened and closed as appropriate.

Technical Details

When TiddlyWiki starts up it processes permalinks according to the following steps; the same steps are repeated if the permalink changes dynamically (this happens in response to the user editing the address bar, for example).

1. If the permalink contains a colon, treat the string before as the **target** and the string after it as the **story filter**
2. If the permalink doesn't contain a colon, treat the entire string as the **target** and mark the **story filter** as *unspecified*
3. If the **story filter** was unspecified and we're in the process of starting up, then set the **story filter** to the empty string if the **target** is specified, or to the default tiddlers if the **target** is unspecified
4. If the **story filter** was unspecified and we're not starting up, then set the **story filter** to the current story list
5. Evaluate the **story filter** as the **story list**
6. If the **target** is specified and not present in the **story list** then add it at the top
7. If the **target** is specified then navigate to it, otherwise navigate to the first tiddler in the **story list**

PESpot Lesson Planner by Patrick Detzner

3rd April 2015 at 12:52pm

Examples

A system for lesson planning built with TiddlyWiki.

http://pespot.tiddlyspot.com

> I built a program to help enter and store lesson plan information and then facilitate/automate some of the tedious work involved in creating the actual lesson plan document.

Philosophy of Tiddlers

19th September 2014 at 6:06pm

Learning

The purpose of recording and organising information is so that it can be used again. The value of recorded information is directly proportional to the ease with which it can be re-used.

The philosophy of tiddlers is that we maximise the possibilities for re-use by slicing information up into the smallest semantically meaningful units with rich modelling of relationships between them. Then we use aggregation and composition to weave the fragments together to present narrative stories.

TiddlyWiki aspires to provide an algebra for tiddlers, a concise way of expressing and exploring the relationships between items of information.

Pinstripe.gif

Places where the parser ignores WikiText

22nd January 2022 at 7:28pm

WikiText Parser Modes

Text enclosed by these constructs is skipped by the parser and WikiText punctuation will be ignored:

Code	One of the main purposes of code blocks is to suppress wikitext expansion. Once the code block starts, the parser will ignore all

Blocks in WikiText	WikiText punctuation until the code block ends.
Images in WikiText	`[[img|literal image link text]]` - the text enclosed by square braces will be ignored. This means, for example, transclusions and macro calls cannot be used to dynamically construct the link text
Linking in WikiText	`[[literal link target|literal link text]]` - the text enclosed by square braces will be ignored. This means, for example, transclusions and macro calls cannot be used to dynamically construct the link target or the link text
Macro Calls	`<<mymacro ''notbold'' "literal text" "<<macro_expansion_ignored>>" {{transclusion_ignored}}>>` - while processing the text enclosed by a macro call, the parser will follow special rules for detecting macro parameters. These rules do not include detection of WikiText. However, after the parameters are substituted into the macro definition, the result will be parsed using *normal rules*. This will likely result in the detection of any WikiText.

Platforms

12th April 2015 at 8:54pm

TableOfContents

TiddlyWiki can be used on several platforms:

- TiddlyDesktop
- TiddlyWiki on Node.js

Plugin Dependencies

28th September 2019 at 10:27am

PluginMechanism

The simplest form of dependency between plugins is that one plugin can declare itself to be a sub-plugin of another plugin by specifying that plugin in the `parent-plugin` field. Sub-plugins are displayed in the control panel plugin chooser within the expandable area of their parent plugin. For example, the official plugin library includes the CodeMirror Plugin and also more than a dozen sub-plugins that extend the *CodeMirror* plugin with optional functionality. Note that sub-plugins cannot themselves have further sub-plugins.

Plugins can also use their `dependents` field to list the titles of any other plugins that should also be installed. Note that the list is not resolved recursively: dependents of dependents will not be installed.

When installing a plugin from the plugin library, both the `parent-plugin` and `dependents` fields are used to determine any additional plugins to be installed.

Note that the `dependents` field is also used by the *ThemeMechanism* and the *LanguageMechanism* to switch in dependent theme/language plugins.

Plugin Editions

7th January 2016 at 11:37pm

Editions

These editions contain simple demos of the original set of TiddlyWiki plugins. They were prepared for earlier versions of TiddlyWiki that lacked the plugin library.

- CodeMirror Plugin
- D3 Plugin
- External Attachments Plugin
- Highlight Plugin

- KaTeX Plugin
- Markdown Plugin
- Railroad Plugin
- TW2Parser Plugin

Plugin Fields

28th September 2019 at 10:30am

PluginMechanism

Plugins are stored as tiddlers with the following fields:

Field	Description
author	*(optional)* Author of plugin
dependents	List of dependencies: plugins that should be installed in order for this one to work
description	Description of plugin
list	Names of exposed plugin information tiddlers (see below)
name	Brief name of the plugin
plugin-priority	*(optional)* Numerical priority (higher numbers install later, the core plugin has a priority of zero
plugin-type	Generally **plugin**, **language** or **theme**; see PluginMechanism for details
parent-plugin	*(optional)* ⓘ **Introduced in v5.1.22** Title of the plugin under which this plugin should be displayed as a sub-plugin in the control panel plugin chooser
source	*(optional)* Source URL of plugin
text	JSON encoding of the list of tiddlers comprising the plugin

title	Title of plugin
type	Must be **application/json**
version	*(optional)* Version string (should conform to SemanticVersioning convention)

Plugin Information Tiddlers

15th October 2016 at 2:17pm

PluginMechanism

Information Tiddlers for Plugins

Plugin authors are encouraged to provide special information and documentation tiddlers that TiddlyWiki can include as plugin information tabs in the **control panel**.

Plugins should provide an icon contained in a tiddler with the title formed of `<plugin-name>/icon` (for example, **$:/core/icon**).

Plugins expose the names of the individual information tabs that they wish to display in the `list` field of the plugin tiddler. By convention, some or all of the following should be provided:

- **readme**: basic information about the plugin
- **license**: the license under which the plugin is published

The title of the associated information tiddler must be formed as follows:

1. `$:/<plugin-name>/<current-language>/<tab-name>` (for example, *$:/core/en-GB/readme*)

2. `$:/<plugin-name>/<tab-name>` (for example, **$:/core/readme**)

Thus, plugins can provide language-specific versions of each information tiddler.

Note that information tiddlers should not reference other tiddlers within the plugin. This is because plugins containing themes or languages are dynamically switched in and out as they are selected, and so their information tiddlers may not be available for viewing. The control panel uses the 'subtiddler' attribute of the TranscludeWidget to access these tiddlers, which works independently of the plugin switching mechanism.

Plugin Ordering

28th June 2022 at 6:01pm

PluginMechanism

Using the Node.js client-server configuration plugins are activated in the following order:

1. Plugins found using the OS environment variables
   - See: PluginFolders
   - and: Environment Variables on Node.js
2. Plugins stored in the wiki `/plugins` path
   - See: PluginFolders
3. Plugins specified in the command line
   - See: Using TiddlyWiki on Node.js ... the notes about ⓘ **Introduced in v5.1.20**
4. Plugins imported by drag and drop as wiki content
   - See: Installing a plugin from the plugin library or Manually installing a plugin

Important:

- **Elements lower in the list take precedence**
- Including a plugin as an ordinary tiddler by drag and drop into the browser, will result in the plugin only being active in the browser
    - It will not be available under Node.js

Option 1:

- Using a Node.js configuration option 1 is the most common one
- It lowers maintenence, because all plugins can be updated in one go

Option 2:

- Option 2 is preferred if you want to work with a fixed configuration
- Plugin updates only effect the corresponding wiki

Option 3:

- This option allows to **add and start** an edition without the need to overwrite an existing `tiddlywiki.info` file
- It allows you to start any edition as a client-server edition, even if the tiddlywiki.info file didn't specify it.

Option 4:

- Is the exact same mechanism used for a single-file wiki
- This mechanism should only be used in a Node.js configuration for **testing and debugging** purposes, since the plugins are stored in the `tiddlers/` directory

Plugin Types

14th November 2021 at 11:26pm

PluginMechanism

Plugins have a `plugin-type` field that determines how it is treated. The following values are defined by the core:

- `plugin` - an ordinary plugin
- `theme` - a theme plugin (see *ThemeMechanism*). Only the theme plugin named in the tiddler $:/theme is activated at any one time (along with any dependent theme plugins)
- `language` - a language plugin (see TranslationMechanism). Only the language plugin named in the tiddler $:/language is activated at any one time (along with any dependent language plugins)

The following plugin types are used internally by the core:

- `import` - for the special *$:/Import* plugin created while importing is in progress
- `info` - for the special $:/temp/info-plugin tiddler created by the InfoMechanism

While not supported by the plugin library, it is also possible to use custom plugin types to build plugin-like behaviour with a custom user interface. Plugins with a custom type will automatically receive the same behaviour as plain plugins but can display a custom user interface using an appropriate view template segment.

By default, tiddlers within plugins with custom plugin types will not be registered as shadow tiddlers (the constituent tiddlers can still be addressed as subtiddlers by the TranscludeWidget, ViewWidget or the plugintiddlers Operator). Setting a configuration tiddler `$:/config/RegisterPluginType/<plugin-type>` to `yes` will cause the tiddlers within the plugin to be registered as shadow tiddlers.

PluginFolders

13th June 2022 at 2:35pm

PluginMechanism

Working with Plugin Folders

- Arrange the PluginFolders containing the plugins in a convenient shared location
- Use environment variables to tell TiddlyWiki to search those folders.
- The plugins can be referenced in `tiddlywiki.info` by their name (e.g. `tiddlytools/magic`)
- Place the PluginFolders containing the plugins in a `plugins` folder within the wiki folder. TiddlyWiki will attempt to include every subfolder as a plugin. Do not add the plugin names to `tiddlywiki.info`. Do not add the PluginFolders under a specific namespace:

```
├── plugins/
│   ├── relink/
│   │   ├── js/
│   │   ├── tiddlers/
│   │   └── plugin.info
│   └── relink-markdown/
│       ├── js/
│       ├── plugin.info
│       └── readme.tid
├── tiddlers/
└── tiddlywiki.info
```

Plugin.info File

On the server, plugins can be stored as ordinary JSON tiddlers but it is often more convenient to store them as separate tiddler files within folders. Plugin folders must contain a `plugin.info` file that contains the metadata for the plugin. It can also optionally identify files external to the plugin folder that should be loaded as tiddlers.

The `plugin.info` file should contain the following JSON structure:

The JSON structure for plugin tiddlers is as follows:

```
{
    "title": "$:/plugins/publisher/name",
```

```
    "name": "name",
    "description": "An exemplary plugin for demonstration purposes",
    "author": "JeremyRuston",
    "version": "1.2.3-alpha3",
    "core-version": ">=5.0.0",
    "source": "https://tiddlywiki.com/MyPlugin",
    "plugin-type": "plugin",
    "list": "readme license history"
}
```

By convention, the titles of the individual tiddlers are prefixed with the title of the containing plugin, but they are not restricted to do so.

Note that if the `version` field is omitted from a `plugin.info` file when the plugin folder is packed then it is automatically filled in by the core to the current core version number. This is to ensure that all the core plugins carry the correct version number. Generally plugin authors will want to ensure that they do explicitly specify a version number.

PluginMechanism

13th June 2022 at 2:44pm

Mechanisms

Introduction

Plugins are bundles of tiddlers that are distributed and managed as a single unit. Users can install them with drag and drop, or using the plugin

library.

Introduced in v5.1.22 Plugins that contain JavaScript modules require a reload of the wiki before they will work. Plugins that do not contain JavaScript modules are automatically dynamically loaded and unloaded.

Plugins can be used to package any tiddler content, including JavaScript modules that extend and enhance the core TiddlyWiki5 functionality. The tiddlers within registered plugins are ShadowTiddlers: they can be freely overwritten by creating a tiddler with the same title, but deleting that tiddler restores the underlying tiddler value from the plugin.

By convention, plugin titles have the form `$:/plugins/<publisher>/<name>`. Plugins that are part of the core TiddlyWiki distribution have titles of the form `$:/plugins/tiddlywiki/<name>`.

When running TiddlyWiki under Node.js, plugins can also be stored as individual tiddler files in PluginFolders.

Plugin Types

Plugins have a `plugin-type` field that determines how it is treated. The following values are defined by the core:

- `plugin` - an ordinary plugin
- `theme` - a theme plugin (see *ThemeMechanism*). Only the theme plugin named in the tiddler $:/theme is activated at any one time (along with any dependent theme plugins)
- `language` - a language plugin (see TranslationMechanism). Only the language plugin named in the tiddler $:/language is activated at any one time (along with any dependent language plugins)

The following plugin types are used internally by the core:

- `import` - for the special *$:/Import* plugin created while importing is in progress
- `info` - for the special $:/temp/info-plugin tiddler created by the InfoMechanism

While not supported by the plugin library, it is also possible to use custom plugin types to build plugin-like behaviour with a custom user interface. Plugins with a custom type will automatically receive the same behaviour as plain plugins but can display a custom user interface using an appropriate view template segment.

By default, tiddlers within plugins with custom plugin types will not be registered as shadow tiddlers (the constituent tiddlers can still be addressed as subtiddlers by the TranscludeWidget, ViewWidget or the plugintiddlers Operator). Setting a configuration tiddler `$:/config/RegisterPluginType/<plugin-type>` to `yes` will cause the tiddlers within the plugin to be registered as shadow tiddlers.

Plugin Dependencies

The simplest form of dependency between plugins is that one plugin can declare itself to be a sub-plugin of another plugin by specifying that plugin in the `parent-plugin` field. Sub-plugins are displayed in the control panel plugin chooser within the expandable area of their parent plugin. For example, the official plugin library includes the CodeMirror Plugin and also more than a dozen sub-plugins that extend the *CodeMirror* plugin with optional functionality. Note that sub-plugins cannot themselves have further sub-plugins.

Plugins can also use their `dependents` field to list the titles of any other plugins that should also be installed. Note that the list is not resolved recursively: dependents of dependents will not be installed.

When installing a plugin from the plugin library, both the `parent-plugin` and `dependents` fields are used to determine any additional plugins to be installed.

Note that the `dependents` field is also used by the *ThemeMechanism* and the *LanguageMechanism* to switch in dependent theme/language

plugins.

Plugin Ordering

Using the Node.js client-server configuration plugins are activated in the following order:

1. Plugins found using the OS environment variables
   - See: PluginFolders
   - and: Environment Variables on Node.js
2. Plugins stored in the wiki `/plugins` path
   - See: PluginFolders
3. Plugins specified in the command line
   - See: Using TiddlyWiki on Node.js ... the notes about **Introduced in v5.1.20**
4. Plugins imported by drag and drop as wiki content
   - See: Installing a plugin from the plugin library or Manually installing a plugin

Important:

- **Elements lower in the list take precedence**
- Including a plugin as an ordinary tiddler by drag and drop into the browser, will result in the plugin only being active in the browser
  - It will not be available under Node.js

Option 1:

- Using a Node.js configuration option 1 is the most common one
- It lowers maintenence, because all plugins can be updated in one go

Option 2:

- Option 2 is preferred if you want to work with a fixed configuration
- Plugin updates only effect the corresponding wiki

Option 3:

- This option allows to **add and start** an edition without the need to overwrite an existing `tiddlywiki.info` file
- It allows you to start any edition as a client-server edition, even if the tiddlywiki.info file didn't specify it.

Option 4:

- Is the exact same mechanism used for a single-file wiki
- This mechanism should only be used in a Node.js configuration for **testing and debugging** purposes, since the plugins are stored in the `tiddlers/` directory

Plugin Fields

Plugins are stored as tiddlers with the following fields:

Field	Description
author	*(optional)* Author of plugin

dependents	List of dependencies: plugins that should be installed in order for this one to work
description	Description of plugin
list	Names of exposed plugin information tiddlers (see below)
name	Brief name of the plugin
plugin-priority	*(optional)* Numerical priority (higher numbers install later, the core plugin has a priority of zero
plugin-type	Generally **plugin**, **language** or **theme**; see PluginMechanism for details
parent-plugin	*(optional)* ⓘ **Introduced in v5.1.22** Title of the plugin under which this plugin should be displayed as a sub-plugin in the control panel plugin chooser
source	*(optional)* Source URL of plugin
text	JSON encoding of the list of tiddlers comprising the plugin
title	Title of plugin
type	Must be **application/json**
version	*(optional)* Version string (should conform to SemanticVersioning convention)

More information

- Disabling Plugins
- Plugin Dependencies
- Plugin Fields
- Plugin Information Tiddlers
- Plugin Ordering
- Plugin Types

- PluginFolders

Plugins

17th June 2022 at 3:38pm

Concepts TableOfContents

Introduction

Plugins in TiddlyWiki5 can be used to distribute optional components that customise and extend wiki. You can install them from the official plugin library or from community sites.

Internally, plugins are a bundle of tiddlers packaged together as a single tiddler that can be installed, copied, disabled or deleted as a unit. The individual tiddlers within a plugin appear as shadow tiddlers.

Plugins can contain JavaScript modules, style sheets, and templates. Plugins can also be used to distribute ordinary text, images or any other content.

Handling Plugins with a Single File Wiki

- Installing a plugin from the plugin library
- Uninstalling a plugin
- Manually installing a plugin

- OfficialPlugins

Handling Plugins with a Client - Server Configuration (Node.js)

- Installing custom plugins on Node.js
- Installing official plugins on Node.js
- Uninstalling a plugin with Node.js

Plugin Mechanism

The PluginMechanism tiddler contains more details about how plugins are implemented internally.

You can open the plugin details in the ⚙ ControlPanel on the **Plugins** sub-tab.

There is a plugin called $:/core that contains the main core code of TiddlyWiki. It is always present, and it is the source of default ShadowTiddlers.

Plugins by TheDiveO

6th January 2021 at 4:10pm

Community Plugins

A collection of plugins from *TheDiveO*.

http://thediveo.github.io

TheDiveO's Third Flow plugin construction system:

> The ThirdFlow plugin brings to you another way to develop customisation plugins for TiddlyWiki 5. It is not enforcing a specific development flow, it simply tries to help you. Otherwise, it tries to stay out of your way.

TheDiveO's FontAwesome plugin:

> The *FontAwesome* plugin supports embedding Font Awesome in TiddlyWiki 5. There is no need to install this font into your operating system in order to use it with TiddlyWiki 5. The font is already embedded in this TiddlyWiki 5 customisation plugin instead, so nothing else to install.

PluginsCS

17th June 2022 at 2:45pm

This tiddler is used as a tag in the Plugins tiddler to create a sorted list

- Installing custom plugins on Node.js
- Installing official plugins on Node.js
- Uninstalling a plugin with Node.js

pluginтiddlers Operator

3rd February 2015 at 8:03pm

Filter Operators Special Operators

purpose	select all shadow titles in the input plugins
input	a selection of plugin tiddler titles
parameter	none
output	all shadow titles contained in the input plugins

Learn more about how to use Filters

Examples

plugintiddlers Operator (Examples)

23rd January 2015 at 11:30pm

Operator Examples plugintiddlers Operator

```
[[$:/core]plugintiddlers[]]
```

Try it

PopupMechanism

23rd December 2015 at 11:27pm

Mechanisms

The popup mechanism allows blocks of content to be selectively displayed and positioned with respect to an anchor. It has several parts:

- StateTiddlers to record whether a popup is currently displayed or not
- The RevealWidget to selectively display the popup content
  - ⓘ **Introduced in v5.2.4** For positioning the popups relative or absolute coordinates can be used. See Coordinate Systems for more information about usage and format.
  - For "sticky" popups — those that don't close when clicking inside one — set the **class** attribute to `tc-popup-keep`
- The ButtonWidget to trigger the display of the popup by setting the state tiddler appropriately

power Operator

20th November 2020 at 5:08am

Binary Mathematics Operators Filter Operators Mathematics Operators

purpose	treating each input title as a number, raise it to the power of the numeric value of the parameter
input	a selection of titles
parameter	**N** = a number
output	the input as numbers, but with each one raised to the power of **N**

Learn more about how to use Filters

ⓘ **Introduced in v5.1.23**

Examples

power Operator (Examples)

20th November 2020 at 5:15am

Operator Examples power Operator

```
[[3]power[3]]
```

Try it

```
9 +[power[2]]
```

Try it

Pragma

22nd January 2022 at 7:28pm

See Pragmas.

Pragma: \define

17th December 2023 at 7:55pm

Pragmas

The **define** pragma is used to define macros. It is a shortcut syntax for the SetVariableWidget.

The usual form allows macros to span multiple lines.

```
\define <macro-name>(<param-name>[:<param-default-value>],<param-name>[:<param-default-value>]...)
<multiple-line-definition-text>
\end [<macro-name>]
```

Note that the `\end` marker can optionally specify the name of the macro to which it relates which allows macro definitions to be nested.

There is also a single line form for shorter macros:

```
\define <macro-name>(<param-name>[:<param-default-value>],<param-name>[:<param-default-value>]...)
<single-line-definition-text>
```

The first line of the definition specifies the macro name and any parameters. Each parameter has a name and, optionally, a default value that is used if no value is supplied on a particular call to the macro.

The lines that follow contain the text of the macro text (i.e. the snippet represented by the macro name), until `\end` appears on a line by itself:

```
\define sayhi(name:"Bugs Bunny",address:"Rabbit Hole Hill")
Hi, I'm $name$ and I live in $address$.
\end
```

Alternatively, the entire definition can be presented on a single line without an `\end` marker:

```
\define sayhi(name:"Bugs Bunny") Hi, I'm $name$.
```

Macro definitions can be nested by specifying the name of the macro in the `\end` marker. For example:

copy to clipboard

```
\define special-button(caption:"Click me")
\define actions()
<$action-sendmessage $message="tm-notify" $param="HelloThere"/>
\end actions
<$button actions=<<actions>>>
$caption$
</$button>
\end special-button

<<special-button>>
```

That renders as:

Click me



If macros are nested, textual substitution will only occur for the outermost macro. Thi is because by the time the inner macros are processed all the substitutions will have already occurred

A more formal presentation of this syntax is also available.

Pragma: \function

19th April 2023 at 12:31pm

Pragmas

New in v5.3.0 The **\function** pragma is used to define custom functions. It is a shortcut syntax for the SetVariableWidget.

The usual form allows custom functions to span multiple lines:

```
\function <function-name>(<param-name>[:<param-default-value>],<param-name>[:<param-default-value>]...)
<multiple-line-definition-text>
\end [<function-name>]
```

Note that the `\end` marker can optionally specify the name of the function to which it relates, enabling function definitions to be nested inside procedures, macros or widget definitions.

There is also a single line form for shorter functions:

```
\function <function-name>(<param-name>[:<param-default-value>],<param-name>[:<param-default-value>]...)
<single-line-definition-text>
```

The first line of the definition specifies the function name and any parameters. Each parameter has a name and, optionally, a default value that is used if no value is supplied on a particular call to the function. The lines that follow contain the text of the function (i.e. the snippet represented by the function name), until `\end` appears on a line by itself:

See Functions for more details.

Pragma: \import

19th April 2023 at 12:31pm

Pragmas

The **import** pragma is used to import definitions from other tiddlers that are identified with a filter. It is a shortcut syntax for the ImportVariablesWidget.

```
\import <filter>
```

For example:

```
\import [all[shadows+tiddlers]tag[$:/tags/Global]]
```

Pragma: \parameters

18th May 2023 at 4:35pm

Pragmas

ⓘ **New in v5.3.0** The **parameters** pragma is used within procedure and widget definitions to declare the parameters that are expected, and their default values. It is a shortcut syntax for the ParametersWidget.

```
\parameters (<name>[:<default-value>],<name>[:<default-value>]...)
```

For example:

```
\parameters (firstname:"Joe",lastname:"Blogs")
```

To illustrate the use of **parameters** pragma, see Core Icons which are parameterised. The first parameter `size` specified the size at which the icon should be rendered. For example see the text of **$:/core/images/print-button** tiddler. The first line defines the size parameter as

```
\parameters (size:"22pt")
```

copy to clipboard

```
{{$:/core/images/print-button|16px}}
<$transclude $tiddler="$:/core/images/print-button" size="32px"/>
```

That renders as:





In the above example, the first line shows a simple transclusion of **$:/core/images/print-button** icon with `size` parameter passed by position and is set to 16px. The second line is a transclusion of image with `size` parameter passed by name and is set to 32px.

Pragma: \parsermode

17th January 2023 at 12:22pm

Pragma

Introduced in v5.2.4

The **\parsermode** pragma adjusts whether the remaining text is parsed in block mode or inline mode. See WikiText Parser Modes for details of parser modes.

As with all pragmas, setting the parser mode this way can only be done at the start of text, not within the main body of the text.

`\parsermode block` **or** `\parsermode inline`

Pragma: \procedure

24th July 2023 at 8:40pm

Pragmas

New in v5.3.0 The **\procedure** pragma is used to define procedures. It is a shortcut syntax for the SetVariableWidget with an implicit ParametersWidget.

The usual form allows procedures to span multiple lines:

```
\procedure <procedure-name>(<param-name>[:<param-default-value>],<param-name>[:<param-default-value>]...)
<multiple-line-definition-text>
\end [<procedure-name>]
```

Note that the `\end` marker can optionally specify the name of the procedure to which it relates which allows procedure definitions to be nested.

There is also a single line form for shorter procedures:

```
\procedure <procedure-name>(<param-name>[:<param-default-value>],<param-name>[:<param-default-value>]...)
<single-line-definition-text>
```

The first line of the definition specifies the procedure name and any parameters. Each parameter has a name and, optionally, a default value that is used if no value is supplied on a particular call to the procedure. The lines that follow contain the text of the procedure text (i.e. the

snippet represented by the procedure name), until `\end` appears on a line by itself:

For example:

```
\procedure sayhi(name:"Bugs Bunny")
Hi, I'm <<name>>.
\end

<<sayhi "Jeremy">>
```

Alternatively, the entire definition can be presented on a single line without an `\end` marker:

```
\procedure sayhi(name:"Bugs Bunny") Hi, I'm <<name>>.
```

Procedure definitions can be nested by specifying the name of the procedure in the `\end` marker. For example:

copy to clipboard

```
\procedure special-button(caption:"Click me")
\procedure actions()
<$action-sendmessage $message="tm-notify" $param="HelloThere"/>
\end actions
<$button actions=<<actions>>>
<<caption>>
</$button>
\end special-button
```

```
<<special-button>>
```

That renders as:

Click me

Use of Parameters Inside Procedures

The parameters can be declared inside procedures. The parameters widget is necessary in a procedure if you want to use computed default values. For example:

copy to clipboard

```
\procedure myproc()
<$parameters name={{$:/SiteTitle}} desc={{$:/SiteSubtitle}}>
This is <<name>> demonstrates <<desc>>.
</$parameters>
\end

<<myproc>>
```

That renders as:

This is TiddlyWiki demonstrates a non-linear personal web notebook.

Caution in Using Positional Parameters

Procedures are a shortcut syntax for the SetVariableWidget with an implicit ParametersWidget, so generally there is no reason to have multiple parameters widgets within a definition. In the below example when passing `x` to `myproc`, it will also be set to `a`:

copy to clipboard

```
\procedure myproc(x:10)
\parameters (a:100, b:200)

x=<<x>>, a=<<a>>, b=<<b>>
\end
```

```
<<myproc 50>>
```

That renders as:

x=50, a=50, b=200

The reason for that result is clearer if we consider an equivalent with explicit parameters widgets.

copy to clipboard

```
<$let myprog="""
\parameters (x:10)
\parameters (a:100, b:200)

x=<<x>>, a=<<a>>, b=<<b>>
""">
<<myprog 50>>
</$let>
```

That renders as:

x=50, a=50, b=200

This is because those two parameters widgets are entirely independent. They are both processed as if the other parameter widget is not there.



The positional parameters are only required when using the parameterised transclusion shortcut syntax, and that in other cases it is generally clearer to use named parameters.

To prevent such situation of above example, pass parameters by name as below.

copy to clipboard

```
\procedure myproc(x:10)
\parameters (a:100, b:200)

x=<<x>>, a=<<a>>, b=<<b>>
\end

<<myproc x:50>>
```

That renders as:

x=50, a=100, b=200

Pragma: \rules

24th July 2023 at 8:40pm

Pragmas

The **\rules** pragma adjusts the set of parser rules used to parse the remaining text.

```
\rules only|except <rule-list>
```

The list of available parser rules can be consulted in **$:/ControlPanel** -> Info -> Advanced -> Parsing.

For example, in stylesheets it is typical to only use the rules associated with macros and transclusions:

```
\rules only filteredtranscludeinline transcludeinline macrodef macrocallinline
```

Some users prefer not to use CamelCase links:

```
\rules except prettylink
```

Pragma: \whitespace

21st September 2023 at 8:03pm

Pragmas

Introduced in v5.1.15 The **whitespace** pragma determines how spaces and newlines are treated within wikitext.

- **notrim** – whitespace text is not subject to special processing (the default)
- **trim** – whitespace text is ignored

Note that the processing only applies to the printable text, and not to other text, such as the values of attributes.

The whitespace setting only applies to the parsed content in which it appears. The setting is inherited by embedded Procedure Definitions and Custom Widgets definitions, but is not inherited by *Macro definitions*.

```
\whitespace trim|notrim
```

For example:

```
\whitespace trim
```

Pragma: \widget

19th April 2023 at 12:31pm

Pragmas

New in v5.3.0 The **\widget** pragma is used to define custom widgets. It is a shortcut syntax for the SetVariableWidget with an implicit ParametersWidget.

The usual form allows custom widgets to span multiple lines:

```
\widget <widget-name>(<param-name>[:<param-default-value>],<param-name>[:<param-default-value>]...)
<multiple-line-definition-text>
\end [<widget-name>]
```

Note that the `\end` marker can optionally specify the name of the widget to which it relates which allows widget definitions to be nested.

There is also a single line form for shorter widgets:

```
\widget <widget-name>(<param-name>[:<param-default-value>],<param-name>[:<param-default-value>]...)
<single-line-definition-text>
```

The first line of the definition specifies the widget name and any parameters. Each parameter has a name and, optionally, a default value that is used if no value is supplied on a particular call to the widget. The lines that follow contain the text of the widget text (i.e. the snippet represented by the widget name), until `\end` appears on a line by itself:

See Custom Widgets for more details.

Pragmas

21st July 2023 at 8:44am

Concepts WikiText Parser Modes

A **pragma** is a special component of WikiText that provides control over the way the remaining text is parsed.

Pragmas occupy lines that start with `\`. They can only appear at the start of the text of a tiddler, but blank lines and comments are allowed between them. If a pragma appears in the main body of the text, it is treated as if it was ordinary text.

ⓘ **Introduced in v5.2.6** Pragmas can have preceding optional whitespace characters.

The following pragmas are available:

- Pragma: \define
- Pragma: \function
- Pragma: \import
- Pragma: \parameters
- Pragma: \procedure
- Pragma: \rules

- Pragma: \whitespace
- Pragma: \widget

precision Operator

11th June 2019 at 5:14pm

Binary Mathematics Operators Filter Operators Mathematics Operators

purpose	convert each number to a string with `N` significant digits
input	a selection of titles
parameter	`N` = a number
output	the input as numbers with `N` significant digits

Learn more about how to use Filters

Introduced in v5.1.20 See Mathematics Operators for an overview.

The **precision** operator returns a string representation of the input number that has exactly the specified number of significant digits, using whichever is shorter of exponential or fixed notation.

Examples

precision Operator (Examples)

11th June 2019 at 5:42pm

Operator Examples precision Operator

```
123456789 +[precision[3]]
```

Try it

```
123456789 +[precision[13]]
```

Try it

```
1.23E23 +[precision[90]]
```

Try it

prefix Operator

11th July 2023 at 10:28am

Filter Operators Negatable Operators String Operators

purpose	filter the input titles by how they start
input	a selection of titles
suffix	the **prefix** operator uses a rich suffix, see below for details
parameter	**s** = a string of characters

output	those input titles that start with **S**
! **output**	those input tiddlers that do *NOT* start with **S**

Learn more about how to use Filters

ⓘ Introduced in v5.2.2

The **prefix** operator uses an extended syntax that permits multiple flags to be passed:

```
[prefix:<flag list>[<parameter>]
```

- **flag list**: a comma delimited list of flags

The available flags are:

- **casesensitive**: (default), this flag forces a case-sensitive match, where upper and lower case letters are considered different
- **caseinsensitive**: overrides the default so that upper and lower case letters are considered identical for matching purposes

Examples

prefix Operator (Examples)

18th February 2022 at 3:34am

Operator Examples prefix Operator

```
[tag[task]!prefix[Go]]
```

Try it

`[tag[task]!prefix:caseinsensitive[go]]`

Try it

`[prefix[$:/languages/]]`

Try it

`[prefix[$:/]]`
→ same as `[is[system]]`

Try it

prepend Operator

8th November 2015 at 6:17am

Filter Operators Listops Operators Order Operators

purpose	prepend a range of items from an array to the list
input	a list of items
suffix	an integer N, defaulting to all
parameter	`list` = the array of items to be prepended to the head of the list
output	a list with items prepended from the head of the parameter array
! **output**	a list with items prepended from the tail of the parameter array

Learn more about how to use Filters

Examples

prepend Operator (Examples)

21st March 2023 at 2:04pm

Operator Examples prepend Operator

These examples make use of the Days of the Week tiddler.

The operator may be used to prepend a number of items to the list.

```
[list[Days of the Week]prepend[Yesterday Today Tomorrow]]
```

Try it

The parameter may list only items without spaces – to include items with spaces, use a reference to an array, e.g. prepend the last three short days of the week to the list

```
[list[Days of the Week]!prepend:3{Days of the Week!!short}]
```

Try it

Preserving open tiddlers at startup

12th September 2014 at 4:22pm

Customise TiddlyWiki

Ordinarily, at startup TiddlyWiki displays the tiddlers specified as a filter in the tiddler $:/DefaultTiddlers.

Sometimes it's useful to re-open the same tiddlers that were open when the file was saved. To do so, set $:/DefaultTiddlers to this filter:

```
[list[$:/StoryList]]
```

This filter returns the tiddlers specified in the $:/StoryList tiddler, which is the system tiddler that TiddlyWiki uses to store the sequence of tiddlers in the current story.

previous Operator

3rd February 2015 at 8:05pm

Field Operators Filter Operators Order Operators

purpose	find which titles in a **`list`** field precede the input ones
input	a selection of titles

parameter	**T** = a tiddler title
output	a selection containing each title that immediately precedes each of the input titles in the **list** field of **T**

Learn more about how to use Filters

Each input title is processed in turn, and its predecessor is located in the **list** field and appended to the output. If a title is not listed in the field, or is the first item there, then it contributes nothing to the output.

Examples

previous Operator (Examples)

23rd January 2015 at 11:34pm

Operator Examples previous Operator

These examples make use of the Days of the Week tiddler.

```
[[Wednesday]previous[Days of the Week]]
```

Try it

```
[[Monday]previous[Days of the Week]]
```

Try it

```
Tuesday Wednesday Thursday +[previous[Days of the Week]]
```

Try it

Procedure Calls

19th April 2023 at 12:31pm

Procedures WikiText

Introduction

This tiddler describes the different ways in which procedure can be called.

Procedure Call Transclusion Shortcut

To call a procedure, place `<<` double angle brackets `>>` around the name and any parameter values.

```
<<my-procedure param:"This is the parameter value">>
```

By default, parameters are listed in the same order as in the procedure definition. A parameter can be labelled with its name and a colon to allow them to be listed in a different order.

If no value is specified for a parameter, the default value given for that parameter in the procedure definition is used instead. (If no default value was defined, the parameter is blank).

Each parameter value can be enclosed in `'` single quotes `'`, `"` double quotes `"`, `"""` triple double quotes `"""` or `[[` double square brackets `]]`. Triple double quotes allow a value to contain almost anything. If a value contains no spaces or single or double quotes, it requires no delimiters.

See the discussion about parser modes

Procedure Calls with `$transclude` Widget

The shortcut syntax expands to the `$transclude` widget with the `$variable` attribute specifying the name of the procedure to transclude.

```
<$transclude $variable="my-procedure" param="This is the parameter value"/>
```

The widget itself offers greater flexibility than the shortcut syntax, including the ability to specify dynamic parameter values.

Assigning Procedure Calls to Attribute Values

The text of a procedure can be directly assigned to an attribute of a widget or HTML element. The result of the procedure is not wikified, which means that parameter handling does not take place.

```
<div class=<<myclasses>>>
...
</div>
```

Using Procedure Calls in Filters

Procedure calls can be used in filters. The text is not wikified which again means that the parameters will be ignored.

```
<$list filter="[<my-procedure>]">
...
</$list>
```

Procedure Definitions

21st September 2023 at 8:03pm

Procedures WikiText

Introduction

This tiddler describes the different ways in which Procedures can be defined.

Procedure Definition Pragma

Procedures are created using the Pragma: \procedure at the start of a tiddler. The definitions are available in the rest of the tiddler that defines them, plus any tiddlers that it transcludes.

```
\procedure my-procedure(param)
This is the procedure text (param=<<param>>)
\end
```

Note that the Pragma: \whitespace setting is inherited from the parsing context in which the procedure definition occurs. That means that a tiddler containing multiple procedure definitions only needs a single whitespace pragma at the top of the tiddler, and the setting will be automatically inherited by the procedure definitions without needing the pragma to be repeated.

Procedure Definition with Set Widget

Procedures are implemented as a special kind of variable and so internally are actually defined with a **$set** widget.

```
<$set name="my-procedure" value="This is the procedure text">
...
</$set>
```

Note

> that it is not currently possible to specify parameters when defining a procedure with the **$set** widget.

Importing Procedure Definitions

The Pragma: \import or **$importvariables** widget can be used to copy procedure definitions from another tiddler.

$:/tags/Global Tag

Global procedures can be defined using the SystemTag: $:/tags/Global.

The tag SystemTag: $:/tags/Global/View is used to define procedures that should only be available within the main view template and the preview panel.

The tag SystemTag: $:/tags/Global/View/Body is used to define procedures that should only be available within the main view template body and the preview panel.

Procedure Parameter Handling

19th April 2023 at 12:31pm

Procedures WikiText

Introduction

Procedure parameters are made available as variables when the procedure contents are wikified.

Accessing Parameters as Variables

When procedures are wikified, the parameters can be accessed as variables.

For example:

copy to clipboard

```
\procedure say-hi(name,address)
Hi, I'm <<name>> and I live in <<address>>.
\end

<<say-hi name:"Bugs" address:"Rabbit Hole Hill">>
```

That renders as:

Hi, I'm Bugs and I live in Rabbit Hole Hill.

Accessing parameters as variables only works in procedures that are wikified and not, for example, when a procedure is used as an attribute value.

Procedures

19th April 2023 at 12:31pm

Concepts Reference

Introduction

New in v5.3.0 A **procedure** is a named snippet of text. They are typically defined with the Pragma: \procedure:

```
\procedure my-procedure(parameter:"Default value")
This is the procedure, and the parameter is <<parameter>>.
```

```
\end
```

The name wrapped in double angled brackets is used a shorthand way of transcluding the snippet. Each of these **procedure calls** can supply a different set of parameters:

```
<<my-procedure>>
<<my-procedure "The parameter">>
```

The parameters that are specified in the procedure call are made available as variables.

How Procedures Work

Procedures are implemented as a special kind of variable. The only thing that distinguishes them from ordinary variables is the way that the parameters are handled.

Using Procedures

- Procedure Definitions describes how to create procedures
- Procedure Calls describes how to use procedures
- Procedure Parameter Handling describes how procedure parameters work

Product Hunt Link

HelloThere Badge



product Operator

11th June 2019 at 6:02pm

Filter Operators Mathematics Operators Reducing Mathematics Operators

purpose	produce the product of the input numbers
input	a selection of titles
output	the result of multiplying together the input as numbers

Learn more about how to use Filters

Introduced in v5.1.20 See Mathematics Operators for an overview.

Compare with the multiply Operator.

Examples

product Operator (Examples)

11th June 2019 at 6:02pm

Operator Examples product Operator

```
=1 =2 =3 =4 =5 +[product[]]
```

Try it

```
[tag[HelloThere]get[text]length[]product[]]
```

Try it

ProductHunt-Badge.svg

HelloThere Badge



Projectify by Nicolas Petton

7th April 2022 at 8:12pm

Community Editions Community Plugins

Project & todo management for TiddlyWiki.

https://github.com/ThaddeusJiang/Projectify

Projectify is a TiddlyWiki plugin for managing projects & todo lists, inspired by products like Todoist and Basecamp.

Projectify features:

- An inbox for collecting tasks and thoughts
- Projects to structure tasks together into separate todo-lists
- Support for scheduling tasks
- A dashboard to quickly view all projects, the inbox, or scheduled tasks

`Note:` The original site, https://projectify.wiki, no longer exists. The provided site is a maintained fork, though not by the original creator.

putafter Operator

8th November 2015 at 6:18am

Filter Operators Listops Operators Order Operators

purpose	move N trailing items after the marker
input	a list of items
suffix	an integer N, defaulting to 1
parameter	**`marker`** = the list item to be used as a marker
output	re-ordered list of items

Learn more about how to use Filters

Examples

putafter Operator (Examples)

8th November 2015 at 6:18am

Operator Examples putafter Operator

These examples make use of the Days of the Week tiddler.

```
[list[Days of the Week]] +[putafter[Wednesday]]
```

Try it

```
[list[Days of the Week]] +[putafter:2[Tuesday]]
```

Try it

```
[list[Days of the Week]] [[Yesterday]] [[Today]] [[Tomorrow]] +[putafter:3[Tuesday]]
```

Try it

putbefore Operator

Filter Operators Listops Operators Order Operators

purpose	move N trailing items before the marker
input	a selection of titles
suffix	an integer N, defaulting to 1
parameter	`marker` = an integer, defaulting to 1
output	the first `N` input titles

Learn more about how to use Filters

Examples

putbefore Operator (Examples)

8th November 2015 at 6:19am

Operator Examples putbefore Operator

These examples make use of the Days of the Week tiddler.

```
[list[Days of the Week]] +[putbefore[Wednesday]]
```

Try it

```
[list[Days of the Week]] +[putbefore:2[Tuesday]]
```

Try it

```
[list[Days of the Week]] [[Yesterday]] [[Today]] [[Tomorrow]] +[putbefore:3[Tuesday]]
```

Try it

putfirst Operator

8th November 2015 at 6:19am

Filter Operators Listops Operators Order Operators

purpose	move N trailing items to the head of the list
input	a list of items
suffix	an integer N, defaulting to 1
parameter	ignored
output	re-ordered list of items

Learn more about how to use Filters

Examples

putfirst Operator (Examples)

8th November 2015 at 6:19am

Operator Examples putfirst Operator

These examples make use of the Days of the Week tiddler.

```
[list[Days of the Week]] +[putfirst[]]
```

Try it

```
[list[Days of the Week]] +[putfirst:2[]]
```

Try it

```
[list[Days of the Week]] [[Yesterday]] [[Today]] [[Tomorrow]] +[putfirst:-2[]]
```

Try it

putlast Operator

8th November 2015 at 6:20am

Filter Operators Listops Operators Order Operators

purpose	move N leading items to the tail of the list
input	a list of items
suffix	an integer N, defaulting to 1
parameter	ignored
output	re-ordered list of items

Learn more about how to use Filters

Examples

putlast Operator (Examples)

8th November 2015 at 6:20am

Operator Examples putlast Operator

These examples make use of the Days of the Week tiddler.

```
[list[Days of the Week]] +[putlast[]]
```

Try it

```
[list[Days of the Week]] +[putlast:2[]]
```

Try it

```
one two three [list[Days of the Week]] +[putlast:-3[]]
```

Try it

Q: How can I style a tiddler if it has "this" AND "that" tag?

11th April 2018 at 7:39pm

Custom Styles FAQ

Answer: ⓘ **Introduced in v5.1.16**

If tiddler has both: `this` AND tag `that` create an orange border.

```
[data-tags*="bar"][data-tags*="froz"] {
  border: 2px solid orange;
}
```

If your CSS looks like this:

```
[data-tags~="this"] {
  border: 2px solid blue;
}
```

```
[data-tags~="that"] {
  border: 2px solid red;
}
```

`this` creates a blue border `that` creates a red border If you swap positions, you get it the other way around.

Learn more about possible attributes!

Q: How can I style a tiddler if it has "this" OR "that" tag?

11th April 2018 at 7:39pm

Custom Styles FAQ

Answer: ⓘ **Introduced in v5.1.16**

Either tag `this` or tag `that` create a red border.

```
[data-tags~="this"],
[data-tags~="that"] {
  border: 2px solid red;
}
```

Learn more about possible attributes!

Q: How can I use a custom field to style a tiddler?

11th April 2018 at 7:39pm

Custom Styles FAQ

Consider the following usecase: ⓘ **Introduced in v5.1.16**

There is a field named: `rank`, which can hold different values eg: `species`.

Answer:

The idea is now: We dynamically create the stylesheet. The style-sheet can look like this:

title: `myStyles`
tags: `$:/tags/Stylesheet`

```
<$list filter="[rank[species]]">
[data-tiddler-title^="<$view field=title/>"] .tc-tiddler-body {
    column-count: 2;
}
</$list>
```

This creates a CSS rule for every tiddler-title, that has a field `rank` and value `species`.

Learn more about possible attributes!

Q: Is there a way to create dynamic stylesheets?

11th April 2018 at 7:39pm

Custom Styles FAQ

Answer: ⓘ **Introduced in v5.1.16**

Yes, see: Q: How can I use a custom field to style a tiddler?

Q: What if a tiddler has no tags?

11th April 2018 at 7:39pm

Custom Styles FAQ

Answer: ⓘ **Introduced in v5.1.16**

- If a tiddler has no tags, but needs styling use: `data-tiddler-title` as CSS selector
  - There is only one tiddler
- If a user wants to style system tiddlers in a custom way: use: `[data-tiddler-title^="$:"/]` as a selector
  - using the TW namespace functionality
- If the user wants special behaviour for tagged tiddlers. eg: Learning
  - use: `[data-tags*="Learning"]` as CSS selector.

The names I'm using are only used for documentation purpose, without changing the existing wiki. I don't want, that the docs has side effects.

Learn more about possible attributes!

qualify Macro

21st February 2015 at 11:38pm

Core Macros Macros

The **qualify** macro is an important part of the StateMechanism.

It returns a unique string that encodes its position within the widget tree, as identified by the stack of transcluded tiddlers that lead to that position.

It is implemented using the **`transclusion`** variable.

Parameters

title
: The prefix for the returned string, normally a tiddler title

Examples

qualify Macro (Examples)

Macro Examples qualify Macro

The results returned by the **`qualify`** macro depend on its location in the transclusion stack.

The two examples below are identical, but produce different results because the second one has been transcluded from another tiddler.

Each example shows the result of calling **`qualify`** and then the value of the **`transclusion`** variable.

copy to clipboard

```
<ul>
<li><code><<qualify "BaseTiddler">></code></li>
<li><code><<transclusion>></code></li>
</ul>
```

Try it

copy to clipboard


```
<ul>
<li><code><<qualify "BaseTiddler">></code></li>
<li><code><<transclusion>></code></li>
</ul>
```


Try it

QualifyWidget

25th January 2022 at 5:30pm

Widgets

Introduction

The **$qualify** widget sets a variable to a unique string that encodes its position within the widget tree, as identified by the stack of transcluded tiddlers that lead to that position.

Internally, it is implemented using the transclusion variable.

Content and Attributes

The content of the **$qualify** widget is the scope for the value assigned to the **name** variable.

Attribute	Description
title	The prefix for the string, normally a tiddler title
name	The name of the variable to be set to the unique string

Examples

The two examples below are identical, but produce different results because the second one has been transcluded from another tiddler.

Each example shows the result of calling **qualify** and then the value of the **transclusion** variable.

copy to clipboard

```
<$qualify title="BaseTiddler" name="iamnotanumber">
<ul>
<li><code><<iamnotanumber>></code></li>
<li><code><<transclusion>></code></li>
```

```
</ul>
</$qualify>
```

Try it

copy to clipboard

```
<$qualify title="BaseTiddler" name="iamnotanumber">
<ul>
<li><code><<iamnotanumber>></code></li>
<li><code><<transclusion>></code></li>
</ul>
</$qualify>
```

Try it

Remarks

In most cases it will be simpler to use the qualify Macro.

Quick Start

Quick Start: Desktop

Quick Start

Download the official desktop application for macOS, Windows and Linux

Quick Start: DIY

Quick Start

Find the configuration that is right for you to get the full benefits of TiddlyWiki

Quick Start: Tiddlyhost

Quick Start

The easiest way to get started with an online TiddlyWiki

Quick Start: Xememex

Quick Start

For companies and teams, a multiuser TiddlyWiki from Federatial

Quine

10th November 2017 at 11:01am

Definitions

Wikipedia defines a Quine as *a computer program which takes no input and produces a copy of its own source code as its only output*.

TiddlyWiki is an unusual example of a practical quine: it is this ability to produce a copy of its own source code that lies at the heart of TiddlyWiki's ability to independently save changes to itself.

"Quine" is also the name of the unofficial TiddlyWiki application for iPhone/iPad

RadioWidget

13th November 2023 at 10:33am

TriggeringWidgets Widgets

Introduction

The radio widget displays an HTML `<input type="radio">` that reflects whether a given tiddler field has a specified value. Selecting the radio button sets to the tiddler field to the value.

Content and Attributes

The content of the `<$radio>` widget is displayed within an HTML `<label>` element also containing the radio button. This means that clicking on the content will have the same effect as clicking on the button itself.

Attribute	Description
tiddler	Title of the tiddler to manipulate (defaults to the current tiddler)
field	The field of the *tiddler* bound to the radio button
index	ⓘ **Introduced in v5.1.14** The index of the *tiddler* being DataTiddler bound to the radio button (takes precedence over *field*)
default	ⓘ **Introduced in v5.2.0** The default value to be used for matching if the tiddler, field or index are missing
value	The value for the *field* or *index* of the *tiddler*
class	The CSS classes assigned to the label around the radio button ⓘ **Introduced in v5.1.14** `tc-radio` is always applied by default, as well as `tc-radio-selected` when selected
actions	ⓘ **Introduced in v5.1.23** Optional string containing ActionWidgets to be triggered when the value changes. The variable: **actionValue** is available for the actions
disabled	ⓘ **Introduced in v5.1.23** Optional. Set to "yes" to disable the radio input. Defaults to "no"
data-*	ⓘ **New in v5.3.2** Optional data attributes to be assigned to the `<input>` HTML element
style.*	ⓘ **New in v5.3.2** Optional CSS properties to be assigned to the `<input>` HTML element

Field Mode

This example uses the radio widget to change the `modifier` field of this tiddler:

copy to clipboard

```
<$radio field="modifier" value="JoeBloggs"> Joe Bloggs</$radio>
<$radio field="modifier" value="JaneBloggs"> Jane Bloggs</$radio>
```

That renders as:

○ Joe Bloggs ○ Jane Bloggs

Index Mode

Using the radio widget in index mode requires the *index* attribute to specify the name of the index of a DataTiddler to which the specified *value* is assigned.

This example sets the `Tree Frog` index in the tiddler *AnimalColours*:

copy to clipboard

```
<$tiddler tiddler="AnimalColours">
<$radio index="Tree Frog" value="green"> green</$radio>
<$radio index="Tree Frog" value="brown"> brown</$radio>
</$tiddler>
```

That renders as:

○ green ○ brown

Using Actions

When you select an option below the following acitons will occur:

- the value is written to *$:/temp/test/value*
- the **id** field of tiddler: *$:/temp/test* will be updated
- a notification showing the **actionValue** will be shown

○ Option a

○ Option b

○ Option c

The wikitext of this example is shown below:

```
\define newText() value: $(actionValue)$

\define radioActions()
<$action-sendmessage $message="tm-notify" $param="RadioWidget ExampleTemplate" value=<<actionValue>>
current=<<currentTiddler>> />
<$action-createtiddler $basetitle="$:/temp/test/value" $overwrite="yes" text=<<newText>>/>
<$action-navigate $to="$:/temp/test/value"/>
\end

!! Using Actions

When you select an option below the following acitons will occur:
```

```
* the value is written to $:/temp/test/value
* the ''id'' field of tiddler: $:/temp/test will be updated
* a notification showing the ''actionValue'' will be shown

<$tiddler tiddler="$:/temp/test">

<$radio field="id" value="a" actions=<<radioActions>> > Option a </$radio>

<$radio field="id" value="b" actions=<<radioActions>> > Option b </$radio>

<$radio field="id" value="c" actions=<<radioActions>> > Option c </$radio>

</$tiddler>

The wikitext of this example is shown below:

<$codeblock code={{RadioWidget Example!!text}}/>
```

RadioWidget Example

7th March 2021 at 11:20am

RadioWidget

Using Actions

When you select an option below the following acitons will occur:

- the value is written to *$:/temp/test/value*
- the **id** field of tiddler: *$:/temp/test* will be updated
- a notification showing the **actionValue** will be shown

○ Option a

○ Option b

○ Option c

The wikitext of this example is shown below:

```
\define newText() value: $(actionValue)$

\define radioActions()
<$action-sendmessage $message="tm-notify" $param="RadioWidget ExampleTemplate" value=<<actionValue>>
current=<<currentTiddler>> />
<$action-createtiddler $basetitle="$:/temp/test/value" $overwrite="yes" text=<<newText>>/>
<$action-navigate $to="$:/temp/test/value"/>
\end

!! Using Actions

When you select an option below the following acitons will occur:
```

```
* the value is written to $:/temp/test/value
* the ''id'' field of tiddler: $:/temp/test will be updated
* a notification showing the ''actionValue'' will be shown

<$tiddler tiddler="$:/temp/test">

<$radio field="id" value="a" actions=<<radioActions>> > Option a </$radio>

<$radio field="id" value="b" actions=<<radioActions>> > Option b </$radio>

<$radio field="id" value="c" actions=<<radioActions>> > Option c </$radio>

</$tiddler>

The wikitext of this example is shown below:

<$codeblock code={{RadioWidget Example!!text}}/>
```

RadioWidget ExampleTemplate

30th November 2020 at 7:24pm

RadioWidget

value =

Railroad Diagrams

17th November 2021 at 10:24pm

Concepts

Railroad diagrams, sometimes called **syntax diagrams**, are a visual way of explaining the syntax rules of a computer language. Reading one is like reading a public transport map.

Each diagram starts on the left and ends on the right. Simply follow any line from the startpoint to the endpoint. All the alternative lines are equally valid. A line will sometimes jump over an item that is optional, or loop back to indicate that an item can be repeated.

TypeError: setting getter-only property "parentNode"

In the example above, a comma appears between each occurrence of the `repeated` item. The comma path runs from right to left, and can only be reached by first passing through `repeated`.

Characters in round boxes are literal, i.e. they denote themselves. A name in a rectangular box denotes a further railroad diagram.

The railroad diagrams in TiddlyWiki's documentation are generated with the Railroad Plugin.

Railroad Plugin

28th February 2017 at 11:25am

OfficialPlugins Plugin Editions

This plugin provides a `<$railroad>` widget for generating railroad diagrams as SVG images.

Alternatively, the **diagram notation** can be stored in a dedicated tiddler with its `type` field set to `text/vnd.tiddlywiki.railroad`, and that tiddler can simply be transcluded to wherever it is needed.

The plugin is based on a library by Tab Atkins, and has been extended to make it more flexible, including allowing components of a diagram to function as links or be transcluded from other tiddlers.

The railroad widget uses a special notation to construct the components defined below.

`x` and `y` here stand for any component.

Names (as opposed to quoted strings) are available when a value starts with a letter and contains only letters, digits, underscores, dots and hyphens.

---

sequence

TypeError: setting getter-only property "parentNode"

- A sequence of components
- The `<-` and `->` delimiters allow you to force a single component to be treated as a sequence. This is occasionally useful for spacing a diagram out

examples

```
x y z           sequence
```

TypeError: setting getter-only property "parentNode"

```
<-x y z->       explicit sequence
```

TypeError: setting getter-only property "parentNode"

---

optional

TypeError: setting getter-only property "parentNode"

- A component that can be omitted
- The colon makes `x` appear straight ahead

examples

```
[:x]            optional, normally included
```

TypeError: setting getter-only property "parentNode"

```
[x]             optional, normally omitted
```

TypeError: setting getter-only property "parentNode"

---

repeated

TypeError: setting getter-only property "parentNode"

- A list of one or more `x`
- The `+` suffix adds `y` as a separator between each `x` and the next

examples

```
{x}             one or more
```

TypeError: setting getter-only property "parentNode"

```
{x +","}        one or more, comma-separated
```

TypeError: setting getter-only property "parentNode"

---

optional repeated

TypeError: setting getter-only property "parentNode"

- An optional list of `x`, i.e. a list of zero or more `x`

examples

```
[{:x}]          zero or more, normally included
```

TypeError: setting getter-only property "parentNode"

```
[{:x +","}]     zero or more, comma-separated, normally included
```

TypeError: setting getter-only property "parentNode"

```
[{x}]           zero or more, normally omitted
```

> TypeError: setting getter-only property "parentNode"
>
> ```
> [{x +","}]      zero or more, comma-separated, normally omitted
> ```
>
> TypeError: setting getter-only property "parentNode"

---

choice

TypeError: setting getter-only property "parentNode"

- A set of alternatives
- The colon indicates which branch appears straight ahead. By default, it's the first branch

examples

> ```
> (x|y|z)         alternatives
> ```
>
> TypeError: setting getter-only property "parentNode"
>
> ```
> (x|:y|z)        alternatives, normally y
> ```
>
> TypeError: setting getter-only property "parentNode"

---

string / **terminal**

TypeError: setting getter-only property "parentNode"

- A literal or terminal component

- This follows the normal TiddlyWiki rules for quoted strings

examples

```
"x"             terminal
```

TypeError: setting getter-only property "parentNode"

---

nonterminal
TypeError: setting getter-only property "parentNode"

- A nonterminal component, i.e. the name of another diagram

examples

```
<"x">           nonterminal
```

TypeError: setting getter-only property "parentNode"

---

comment
TypeError: setting getter-only property "parentNode"

- A comment

examples

```
/ "comment" /       comment
```

TypeError: setting getter-only property "parentNode"

---

dummy

TypeError: setting getter-only property "parentNode"

- The absence of a component

examples

```
-       dummy
```

TypeError: setting getter-only property "parentNode"

---

link

TypeError: setting getter-only property "parentNode"

- A link to the tiddler title or URI given by the string or name

examples

```
[[x|"tiddler"]]     link
```

TypeError: setting getter-only property "parentNode"

---

transclusion

TypeError: setting getter-only property "parentNode"

- Treats the content of another tiddler as diagram syntax and transcludes it into the current diagram

examples

```
{{"$:/plugins/tiddlywiki/railroad/example-transclusion"}}       transclusion

// "railroad transclusion example" text     <- text in the tidlder
```

TypeError: setting getter-only property "parentNode"

---

arrow pragma

TypeError: setting getter-only property "parentNode"

- Controls whether repeat paths have an arrow on them
- Can be toggled on and off in mid-diagram, if desired

examples

```
<$railroad text="""
\arrow yes
[{:x}]
"""/>
```

> TypeError: setting getter-only property "parentNode"
>
> ```
> <$railroad text="""
> \arrow no
> [{:x}]
> """/>
> ```
>
> TypeError: setting getter-only property "parentNode"

---

debug pragma

TypeError: setting getter-only property "parentNode"

- Causes the diagram to display its parse tree

examples

> ```
> \debug
> x
> ```
>
> ```
> Root
>   Nonterminal: x
> ```

---

start/end pragma

TypeError: setting getter-only property "parentNode"

- Controls the style of the diagram's startpoint or endpoint

examples

```
\start none x
```

TypeError: setting getter-only property "parentNode"

```
\start double
\end double
 x
```

TypeError: setting getter-only property "parentNode"

range Operator

21st March 2023 at 2:38pm

Filter Operators Negatable Operators Selection Constructors

purpose	generate a range of numbers
input	ignored
parameter	**N** = a range specification, like `[1],[5]`
output	a series of evenly spaced numbers ranging from `<begin>` to `<end>`
! **output**	a series of evenly spaced numbers ranging from `<begin>` to `<end>` in reverse order

Learn more about how to use Filters

The `range` operator produces a list of numbers counting up or down. It is useful for counting and numbering.

ⓘ **Introduced in v5.2.0** The range operator has been updated to use multiple parameters. Prior to this version, the range operator only had one parameter, with the three parts delimited by `,`, `;` or `:`.

```
[range[<begin>]]
[range[<begin>],[<end>]]
[range[<begin>],[<end>],[<step>]]
```

The behaviour depends on the number of parameters:

Parameter	Output
`<end>`	Whole numbers up to `<end>`
`<begin>,<end>`	Numbers from `<begin>` to `<end>`, spaced by whole numbers
`<begin>,<end>,<step>`	Numbers from `<begin>` to `<end>` spaced out by `<step>`

Each part must be a number, and works as follows:

- `<begin>`: start counting at this number.
  - Defaults to 1 if `<end>` is at least 1 (or -1 if `<end>` is at most -1).
- `<end>`: stop counting at this number.
  - This number will appear in the list unless it falls between two steps.
- `<step>`: count up (or down) by this amount.
  - Defaults to 1.
  - Cannot be zero.

- We always count from `<begin>` toward `<end>`, whether `<step>` is positive or negative.

The number of decimal points in the output is fixed, and based on the parameter with the *most* decimal points.

To prevent the browser from freezing, `range` is currently limited to 10,000 values.

Examples

```
[range[7]]
```

1 2 3 4 5 6 7

```
[range[1],[10]]
```

1 2 3 4 5 6 7 8 9 10

```
[range[17],[13]]
```

17 16 15 14 13

```
[range[1.001],[5],[1]]
```

1.001 2.001 3.001 4.001

```
[range[.5],[1.4],[.004]]
```

0.500 0.504 0.508 0.512 0.516 0.520 0.524 0.528 0.532 0.536 0.540 0.544 0.548 0.552 0.556 0.560 0.564 0.568 0.572 0.576 0.580 0.584

0.588 0.592 0.596 0.600 0.604 0.608 0.612 0.616 0.620 0.624 0.628 0.632 0.636 0.640 0.644 0.648 0.652 0.656 0.660 0.664 0.668 0.672 0.676 0.680 0.684 0.688 0.692 0.696 0.700 0.704 0.708 0.712 0.716 0.720 0.724 0.728 0.732 0.736 0.740 0.744 0.748 0.752 0.756 0.760 0.764 0.768 0.772 0.776 0.780 0.784 0.788 0.792 0.796 0.800 0.804 0.808 0.812 0.816 0.820 0.824 0.828 0.832 0.836 0.840 0.844 0.848 0.852 0.856 0.860 0.864 0.868 0.872 0.876 0.880 0.884 0.888 0.892 0.896 0.900 0.904 0.908 0.912 0.916 0.920 0.924 0.928 0.932 0.936 0.940 0.944 0.948 0.952 0.956 0.960 0.964 0.968 0.972 0.976 0.980 0.984 0.988 0.992 0.996 1.000 1.004 1.008 1.012 1.016 1.020 1.024 1.028 1.032 1.036 1.040 1.044 1.048 1.052 1.056 1.060 1.064 1.068 1.072 1.076 1.080 1.084 1.088 1.092 1.096 1.100 1.104 1.108 1.112 1.116 1.120 1.124 1.128 1.132 1.136 1.140 1.144 1.148 1.152 1.156 1.160 1.164 1.168 1.172 1.176 1.180 1.184 1.188 1.192 1.196 1.200 1.204 1.208 1.212 1.216 1.220 1.224 1.228 1.232 1.236 1.240 1.244 1.248 1.252 1.256 1.260 1.264 1.268 1.272 1.276 1.280 1.284 1.288 1.292 1.296 1.300 1.304 1.308 1.312 1.316 1.320 1.324 1.328 1.332 1.336 1.340 1.344 1.348 1.352 1.356 1.360 1.364 1.368 1.372 1.376 1.380 1.384 1.388 1.392 1.396 1.400

RangeWidget

13th November 2023 at 10:33am

TriggeringWidgets Widgets

Introduction

The range widget displays an HTML `<input type="range">` that reflects a given tiddler field numeric value. Adjusting the range slider sets the tiddler field to the value.

Content and Attributes

The content of the `<$range>` widget is ignored.

Attribute	Description
tiddler	Title of the tiddler to manipulate (defaults to the current tiddler)
field	The field of the *tiddler* bound to the range slider
index	ⓘ **Introduced in v5.1.22** The index of the *tiddler*, a DataTiddler, to which the radio button is bound
min	The minimum value to be able to be set by the `<$range>` widget
max	The maximum value to be able to be set by the `<$range>` widget
increment	The minimum amount by which a value may be changed. Defaults to 1
default	If the field is missing or empty this is the default position for the widget handle relative to the min and max values
class	CSS classes to be assigned to the label around the range slider
actions	ⓘ **Introduced in v5.1.23** Optional, A string containing ActionWidgets to be triggered when the value changes. The variable: **actionValue** is available for **all** actions. The variable: **actionValueHasChanged** is always `yes` here.
actionsStart	ⓘ **Introduced in v5.1.23** Optional, A string containing ActionWidgets to be triggered when the "handle" is **clicked**. The variable: **actionValueHasChanged** is always `no` here
actionsStop	ⓘ **Introduced in v5.1.23** Optional, A string containing ActionWidgets to be triggered when the "handle" is **released**. The variable: **actionValueHasChanged** is `yes`, **if** the new-value is different to the start-value
disabled	ⓘ **Introduced in v5.1.23** Optional, disables the range input if set to "yes". Defaults to "no"
data-*	ⓘ **New in v5.3.2** Optional data attributes to be assigned to the `<input>` HTML element
style.*	ⓘ **New in v5.3.2** Optional CSS properties to be assigned to the `<input>` HTML element

Examples

Range -1 to 10

copy to clipboard

```
<$range tiddler="$:/_RangeDemo/1" min="-1" max="10" default="1" increment="1"/> {{$:/_RangeDemo/1}}
```

That renders as:



Range 0 to 1

copy to clipboard

```
<$range tiddler="$:/_RangeDemo/2" min="0" max="1" default=".01" increment=".01"/> {{$:/_RangeDemo/2}}
```

That renders as:



Using Actions





☐ Show log Clear Log

The wikitext of this example is shown below:

```
\define getText() $(text)$ - value: $(value)$ - changed: $(actionValueHasChanged)$

\define range-log(text:"", value:"", tiddler:"$:/temp/range/log")
<$vars text="""$text$""" value="""$value$""" >
<$action-setfield $tiddler="""$tiddler$""" $index=<<now "0hh-0mm-0ss:0XXX">> $value=<<getText>>/>
</$vars>
\end

\define clear-range-log(tiddler:"$:/temp/range/log")
<$action-setfield $tiddler="""$tiddler$""" text="{}" show=show/>
\end

\define logTiddler()$:/temp/range/log

\define clearLogButton()<$button actions=<<clear-range-log>>>Clear Log</$button>

\define actionsStop()
<$macrocall $name="range-log" text="                    actionsStop" value=<<actionValue>> />
```

```
\end

\define actionsStart()
<$macrocall $name="range-log" text="                    actionsStart" value=<<actionValue>> />
\end

\define actions()
<$macrocall $name="range-log" text="actions" value=<<actionValue>> />
\end

!! Using Actions

<$range tiddler="$:/temp/range/1__" min="-1" max="10" default="1" increment="1"
  actionsStart=<<actionsStart>> actionsStop=<<actionsStop>> actions=<<actions>>/> {{$:/temp/range/1__}}

<$range tiddler="$:/temp/range/__2" min="0" max="1" default=".01" increment=".01"
  actionsStart=<<actionsStart>> actionsStop=<<actionsStop>> actions=<<actions>> /> {{$:/temp/range/__2}}

<$checkbox tiddler=<<logTiddler>> field="show" checked="show" unchecked="hide"> Show log</$checkbox>
<<clearLogButton>>

<$reveal stateTitle=<<logTiddler>> stateField="show" type="match" text="show">
<$transclude tiddler=<<logTiddler>> />
</$reveal>
```

```
---

The wikitext of this example is shown below:

<$codeblock code={{RangeWidget Example!!text}}/>
```

RangeWidget Example

7th March 2021 at 11:20am

RangeWidget

Using Actions





☐ Show log Clear Log

The wikitext of this example is shown below:

```
\define getText() $(text)$ - value: $(value)$ - changed: $(actionValueHasChanged)$

\define range-log(text:"", value:"", tiddler:"$:/temp/range/log")
<$vars text="""$text$""" value="""$value$""" >
```

```
<$action-setfield $tiddler="""$tiddler$""" $index=<<now "0hh-0mm-0ss:0XXX">> $value=<<getText>>/>
</$vars>
\end

\define clear-range-log(tiddler:"$:/temp/range/log")
<$action-setfield $tiddler="""$tiddler$""" text="{}" show=show/>
\end

\define logTiddler()$:/temp/range/log

\define clearLogButton()<$button actions=<<clear-range-log>>>Clear Log</$button>

\define actionsStop()
<$macrocall $name="range-log" text="                    actionsStop" value=<<actionValue>> />
\end

\define actionsStart()
<$macrocall $name="range-log" text="                    actionsStart" value=<<actionValue>> />
\end

\define actions()
<$macrocall $name="range-log" text="actions" value=<<actionValue>> />
\end

!! Using Actions
```

```
<$range tiddler="$:/temp/range/1__" min="-1" max="10" default="1" increment="1"
  actionsStart=<<actionsStart>> actionsStop=<<actionsStop>> actions=<<actions>>/> {{$:/temp/range/1__}}

<$range tiddler="$:/temp/range/__2" min="0" max="1" default=".01" increment=".01"
  actionsStart=<<actionsStart>> actionsStop=<<actionsStop>> actions=<<actions>> /> {{$:/temp/range/__2}}

<$checkbox tiddler=<<logTiddler>> field="show" checked="show" unchecked="hide"> Show log</$checkbox>
<<clearLogButton>>

<$reveal stateTitle=<<logTiddler>> stateField="show" type="match" text="show">
<$transclude tiddler=<<logTiddler>> />
</$reveal>

---

The wikitext of this example is shown below:

<$codeblock code={{RangeWidget Example!!text}}/>
```

rboue's plugins for TiddlyWiki

6th January 2021 at 4:10pm

Community Plugins

Featuring plugins that integrate two interesting new capabilities:

- SocialCalc providing a powerful spreadsheet
- Three.js providing interactive 3D graphics

http://rboue.tiddlyspot.com

> I developed some plugins related to engineering tasks.
>
> After using them for personal projects I'm glad to share with TiddlyWiki community.

Reading data from JSON tiddlers

11th June 2022 at 12:47pm

JSON in TiddlyWiki Learning

See JSON in TiddlyWiki for an overview of using JSON in TiddlyWiki.

Filter Operators for Accessing JSON Data

The following filter operators allow values to be read from JSON data:

- **`jsonget`** to retrieve the values of a property in JSON data
- **`jsontype`** to retrieve the type of a JSON value
- **`jsonindexes`** to retrieve the names of the fields of a JSON object, or the indexes of a JSON array
- **`jsonextract`** to retrieve a JSON value as a string of JSON

Text References for Accessing JSON Data

Text references are a simple shortcut syntax to look up the value of a named property. For example, if a DictionaryTiddler called `MonthDays` contains:

```
oct:31
nov:30
dec:31
```

... then `{{MonthDays##nov}}` will resolve to the value `30`.

The same is true if `MonthDays` is a JSONTiddler with the following content:

```
{"oct":31,"nov":30,"dec":31}
```

ReadMe

20th September 2014 at 3:52pm

Welcome to TiddlyWiki, a non-linear personal web notebook that anyone can use and keep forever, independently of any corporation.

TiddlyWiki is a complete interactive wiki in JavaScript. It can be used as a single HTML file in the browser or as a powerful Node.js application. It is highly customisable: the entire user interface is itself implemented in hackable WikiText.

Learn more and see it in action at https://tiddlywiki.com/

Developer documentation is in progress at https://tiddlywiki.com/dev/

Join the Community

Official Forums

The new official forum for talking about TiddlyWiki: requests for help, announcements of new releases and plugins, debating new features, or just sharing experiences. You can participate via the associated website, or subscribe via email.

https://talk.tiddlywiki.org/

Note that talk.tiddlywiki.org is a community run service that we host and maintain ourselves. The modest running costs are covered by community contributions.

For the convenience of existing users, we also continue to operate the original TiddlyWiki group (hosted on Google Groups since 2005):

https://groups.google.com/group/TiddlyWiki

Developer Forums

There are several resources for developers to learn more about TiddlyWiki and to discuss and contribute to its development.

- tiddlywiki.com/dev is the official developer documentation
- Get involved in the development on GitHub
    - Discussions are for Q&A and open-ended discussion
    - Issues are for raising bug reports and proposing specific, actionable new ideas
- The older TiddlyWikiDev Google Group is now closed in favour of GitHub Discussions but remains a useful archive: https://

groups.google.com/group/TiddlyWikiDev
    - An enhanced group search facility is available on mail-archive.com
- Follow @TiddlyWiki on Twitter for the latest news
- Chat at https://gitter.im/TiddlyWiki/public (development room coming soon)

Other Forums

- TiddlyWiki Subreddit
- Chat with Gitter at https://gitter.im/TiddlyWiki/public !
- Chat on Discord at https://discord.gg/HFFZVQ8

Documentation

There is also a discussion group specifically for discussing TiddlyWiki documentation improvement initiatives: https://groups.google.com/group/tiddlywikidocs

Installing TiddlyWiki on Node.js

1. Install Node.js
    - Linux:

> *Debian/Ubuntu*:
> `apt install nodejs`
> May need to be followed up by:

> `apt install npm`
>
> *Arch Linux*
> `yay -S tiddlywiki`
> (installs node and tiddlywiki)

- Mac

> `brew install node`

- Android

> Termux for Android

- Other

> See http://nodejs.org

2. Open a command line terminal and type:

> `npm install -g tiddlywiki`
>
> If it fails with an error you may need to re-run the command as an administrator:
>
> `sudo npm install -g tiddlywiki` (Mac/Linux)

3. Ensure TiddlyWiki is installed by typing:

```
tiddlywiki --version
```

- In response, you should see TiddlyWiki report its current version (eg "5.3.3". You may also see other debugging information reported.)

4. Try it out:
   1. `tiddlywiki mynewwiki --init server` to create a folder for a new wiki that includes server-related components
   2. `tiddlywiki mynewwiki --listen` to start TiddlyWiki
   3. Visit http://127.0.0.1:8080/ in your browser
   4. Try editing and creating tiddlers
5. Optionally, make an offline copy:
   - click the **save changes** button in the sidebar, **OR**
   - `tiddlywiki mynewwiki --build index`

The `-g` flag causes TiddlyWiki to be installed globally. Without it, TiddlyWiki will only be available in the directory where you installed it.



If you are using Debian or Debian-based Linux and you are receiving a `node: command not found` error though node.js package is installed, you may need to create a symbolic link between `nodejs` and `node`. Consult your distro's manual and `whereis` to correctly create a link. See github issue 1434.

Example Debian v8.0: `sudo ln -s /usr/bin/nodejs /usr/bin/node`



You can also install prior versions like this:
`npm install -g tiddlywiki@5.1.13`

Using TiddlyWiki on Node.js

TiddlyWiki5 includes a set of commands for use on the command line to perform an extensive set of operations based on TiddlyWikiFolders, TiddlerFiles.

For example, the following command loads the tiddlers from a TiddlyWiki HTML file and then saves one of them in static HTML:

```
tiddlywiki --verbose --load mywiki.html --rendertiddler ReadMe ./readme.html
```

Running `tiddlywiki` from the command line boots the TiddlyWiki kernel, loads the core plugins and establishes an empty wiki store. It then sequentially processes the command line arguments from left to right. The arguments are separated with spaces.

Introduced in v5.1.20 First, there can be zero or more plugin references identified by the prefix `+` for plugin names or `++` for a path to a plugin folder. These plugins are loaded in addition to any specified in the TiddlyWikiFolder.

The next argument is the optional path to the TiddlyWikiFolder to be loaded. If not present, then the current directory is used.

The commands and their individual arguments follow, each command being identified by the prefix `--`.

```
tiddlywiki [+<pluginname> | ++<pluginpath>] [<wikipath>] [--<command> [<arg>[,<arg>]]]
```

For example:

```
tiddlywiki --version
tiddlywiki +plugins/tiddlywiki/filesystem +plugins/tiddlywiki/tiddlyweb mywiki --listen
tiddlywiki ++./mygreatplugin mywiki --listen
```

Introduced in v5.1.18 Commands such as the ListenCommand that support large numbers of parameters can use NamedCommandParameters to make things less unwieldy. For example:

```
tiddlywiki wikipath --listen username=jeremy port=8090
```

See Commands for a full listing of the available commands.

Upgrading TiddlyWiki on Node.js

If you've installed TiddlyWiki on Node.js on the usual way, when a new version is released you can upgrade it with this command:

```
npm update -g tiddlywiki
```

On Mac or Linux you'll need to add **sudo** like this:

```
sudo npm update -g tiddlywiki
```

Also see

- Building TiddlyWikiClassic
- Customising Tiddler File Naming
- Environment Variables on Node.js
- Generating Static Sites with TiddlyWiki

- How to build a TiddlyWiki5 from individual tiddlers
- Installing custom plugins on Node.js
- Installing official plugins on Node.js
- Internet Information Services
- Installing TiddlyWiki Prerelease on Node.js
- MultiTiddlerFiles
- MultiTiddlerFileSyntax
- NamedCommandParameters
- Scripts for TiddlyWiki on Node.js
- Node.js on Termux
- TiddlerFiles
- tiddlywiki.files Files
- tiddlywiki.info Files
- TiddlyWikiFolders
- Uninstalling a plugin with Node.js
- Using a custom path prefix with the client-server edition
- Using TiddlyWiki for GitHub project documentation
- Working with the TiddlyWiki5 repository

This readme file was automatically generated by TiddlyWiki

ReadMeBinFolder

8th September 2014 at 5:08pm

Script Files

The TiddlyWiki5 repository contains several scripts in the `bin` folder that you can use to automate common tasks, or as a useful starting point for your own scripts. See *Scripts for building tiddlywiki.com* for details of the scripts used to build and release https://tiddlywiki.com/.

All the scripts expect to be run from the root folder of the repository.

`serve`: serves tw5.com

```
./bin/serve.sh -h
./bin/serve.sh [edition dir] [username] [password] [host] [port]
```

Or:

```
./bin/serve.cmd -h
./bin/serve.cmd [edition dir] [username] [password] [host] [port]
```

This script starts TiddlyWiki5 running as an HTTP server, defaulting to the content from the `tw5.com-server` edition. By default, the Node.js serves on port 8080. If the optional `username` parameter is provided, it is used for signing edits. If the `password` is provided then HTTP basic authentication is used. Run the script with the `-h` parameter to see online help.

To experiment with this configuration, run the script and then visit `http://127.0.0.1:8080` in a browser.

Changes made in the browser propagate to the server over HTTP (use the browser developer console to see these requests). The server then syncs changes to the file system (and logs each change to the screen).

`test`: build and run tests

This script runs the `test` edition of TiddlyWiki on the server to perform the server-side tests and to build `test.html` for running the tests in the browser.

`lazy`: serves tw5.com with lazily loaded images

```
./bin/lazy.sh <username> [<password>]
```

Or:

```
./bin/lazy.cmd <username> [<password>]
```

This script serves the `tw5.com-server` edition content with LazyLoading applied to images.

Reddit Logo



Reduce Filter Run Prefix

10th July 2023 at 9:33am

Named Filter Run Prefix

purpose	replaces all filter output so far with a single item by repeatedly applying a filter run to each input title
input	the filter output of all previous runs so far
output	the accumulated single item

Introduced in v5.1.23
TypeError: setting getter-only property "parentNode"

Each input title from previous runs is passed to this run in turn. The result of each previous call to this run is made available in the next call via the variable named **`accumulator`**. The result of the last call to this run is returned as the output. A typical use is to add up the values in a given field of each input title.

Replaces all filter output so far with a single item by repeatedly applying a formula, as described above, to each input title.

The following variables are available within the filter run:

- **`accumulator`** - the result of the previous filter run
- **`currentTiddler`** - the input title
- **`..currentTiddler`** - the value of the variable `currentTiddler` outside the filter run. **Introduced in v5.2.0**
- **`index`** - the numeric index of the current list item (with zero being the first item in the list)
- **`revIndex`** - the reverse numeric index of the current list item (with zero being the last item in the list)
- **`length`** - the total length of the input list



Compare named filter run prefix `:reduce` with reduce Operator which is used to flatten a list of items down to a single item by

> repeatedly applying a subfilter.

```
[tag[shopping]] :reduce[get[quantity]add<accumulator>]
```

is equivalent to:

```
\define num-items() [get[quantity]add<accumulator>]

[tag[shopping]reduce<num-items>]
```

Reduce Filter Run Prefix (Examples)

Reduce Filter Run Prefix (Examples)

5th March 2023 at 1:54pm

Reduce Filter Run Prefix

The following examples use these data tiddlers:

- Brownies quantity: 1, price: 4.99
- Chick Peas quantity: 5, price: 1.32
- Milk quantity: 12, price: 0.46
- Rice Pudding quantity: 4, price: 2.66

```
[tag[shopping]] :reduce[get[quantity]add<accumulator>]
```

Try it

```
[tag[shopping]] :reduce[get[price]multiply{!!quantity}add<accumulator>]
```

Try it

```
[tag[shopping]] :reduce[<index>compare:number:gt[0]then<accumulator>addsuffix[, ]addsuffix<currentTiddler>else<currentTiddler>]
```

→ Uses `<index>` to act differently on the first item than the rest. Just for demonstration. Better to use the join Operator to accomplish this task

Try it

```
[tag[non-existent]] :reduce[get[price]multiply{!!quantity}add<accumulator>]
```

→ Empty input produces empty output

Try it

```
[tag[non-existent]] :reduce[get[price]multiply{!!quantity}add<accumulator>] :else[[0]]
```

→ Use `:else` to ensure output if input was empty

Try it



Unlike the reduce Operator, the `:reduce` prefix cannot specify an initial value for the accumulator, so its initial value will always be empty (which is treated as 0 by mathematical operators). So `=1 =2 =3 :reduce[multiply<accumulator>]` will produce 0, not 6. If you need to specify an initial accumulator value, use the reduce Operator.

```
=1 =2 =3 :reduce[multiply<accumulator>]
```

→ Empty initial value is treated as 0 by mathematical operators

Try it

```
=1 =2 =3 +[reduce<multiply-input>,[1]]
```

→ Setting initial value is sometimes necessary for correct results

Try it

reduce Operator

21st March 2023 at 2:39pm

Filter Operators

purpose	apply a subfilter to each input title in turn, accumulating a single value
input	a selection of titles passed as input to the filter
parameter	**S** = a filter expression. Optional second parameter for initial value for accumulator
output	the final result of running the subfilter **S**

Learn more about how to use Filters

Introduced in v5.1.23 The **reduce** operator runs a subfilter for each input title, passing the result of the previous subfilter run as a variable. The initial value of the accumulator can optionally be specified. It returns the result of the final subfilter run.

The **reduce** operator is used to flatten a list of items down to a single item by repeatedly applying a formula. A typical use is to add up the values in a given field of a list of tiddlers.

The following variables are available within the subfilter:

- **accumulator** - the result of the previous subfilter run
- **currentTiddler** - the input title
- **..currentTiddler** - the value of the variable `currentTiddler` outside the subfilter. **Introduced in v5.2.0**
- **index** - the numeric index of the current list item (with zero being the first item in the list)
- **revIndex** - the reverse numeric index of the current list item (with zero being the last item in the list)
- **length** - the total length of the input list

If the **reduce** operator receives no input, its output will be empty. The else Operator can be useful in such cases.



Literal filter parameters cannot contain square brackets but you can work around the issue by using a variable:

```
<$set name="sum-input" value="[add<accumulator>]">
```

```
{{{ =1 =2 =3 +[reduce<sum-input>] }}}
</$set>
```

Compare with the analogous named filter run prefix `:reduce`

```
\define num-items() [get[quantity]add<accumulator>]

[tag[shopping]reduce<num-items>]
```

is equivalent to:

```
[tag[shopping]] :reduce[get[quantity]add<accumulator>]
```

If the optional second parameter is not given, the initial accumulator value will be empty. Numerical operators treat empty input as if it was the number 0. See the multiply-input examples for how this can affect the result of **reduce** in some cases.

Examples

reduce Operator (Examples)

4th December 2020 at 2:11pm

Operator Examples reduce Operator

These examples use the following predefined variables:

- **add-price**: `[get[price]multiply{!!quantity}add<accumulator>]`
- **num-items**: `[get[quantity]add<accumulator>]`

- **join-with-commas**: `[<index>compare:number:gt[0]then<accumulator>addsuffix[, ]addsuffix<currentTiddler>else<currentTiddler>]`
- **multiply-input**: `[multiply<accumulator>]`

They also use the following data tiddlers:

- Brownies quantity: 1, price: 4.99
- Chick Peas quantity: 5, price: 1.32
- Milk quantity: 12, price: 0.46
- Rice Pudding quantity: 4, price: 2.66

`[tag[shopping]reduce<num-items>]`

Try it

`[tag[shopping]reduce<add-price>]`

Try it

`[tag[shopping]reduce<join-with-commas>]`

→ Uses `<index>` to act differently on the first item than the rest

Try it

`[tag[non-existent]reduce<add-price>]`

→ Empty input produces empty output

Try it

`[tag[non-existent]reduce<add-price>else[0]]`

→ Use `else` to ensure output if input was empty

Try it

`=1 =2 =3 +[reduce<multiply-input>]`

→ Empty initial value is treated as 0 by mathematical operators

Try it

`=1 =2 =3 +[reduce<multiply-input>,[1]]`

→ Setting initial value is sometimes necessary for correct results

Try it

Reference

27th September 2018 at 10:06am

TableOfContents

The following topics provide the canonical reference documentation for TiddlyWiki:

- Concepts
- Definitions
- WikiText
- Macros
- Variables
- SystemTags
- Widgets
- Filters
- Messages
- Commands
- Mechanisms
- Developers
- Custom Widgets
- Functions
- Procedures

Reference Tiddlers

18th March 2023 at 5:08pm

Improving TiddlyWiki Documentation

Reference tiddlers offer raw information in a comprehensive interlinked way. The reader is likely to be an intermediate or expert user.

There are several subcategories:

Concepts

- With definitions, together forming a glossary

User manual

- Presenting technical details of WikiText features
- Subcategorised: messages, operators, widgets, etc
    - Widget documentation should follow the Widget Documentation Style Guide

Developer manual

- Presenting technical details of TiddlyWiki's internal architecture

Reference material is written in a terse, formal style that avoids referring to the reader, and instead focuses on how TiddlyWiki itself behaves. The passive voice is often suitable:

- "the template is specified as a tiddler" rather than "specify the template as a tiddler"

- "the widget can be used for various purposes" rather than "you can use the widget for various purposes"
- But "this widget has several possible uses" is better, because it is less convoluted and more succinct

Most contracted verb forms are avoided in reference tiddlers. But those ending in "n't" ("aren't", "doesn't", "hasn't", "isn't", etc) are acceptable, as they make it less easy to accidentally overlook the word "not".

Refnotes by Mohammad

17th November 2020 at 5:21pm

Community Plugins

Refnotes plugin is a set of macros and stylesheets for creating abbreviations, footnotes and citations. It also makes tables of footnotes, abbreviations (glossary) and references (bibliography table).

https://kookma.github.io/TW-Refnotes/

Refnotes contains codes and elements to

Create and insert

- Abbreviation
- Footnote
- Reference

Create tables of

- Abbreviations or glossary
- Footnotes and endnotes
- References (bibliography) using different output style

RefreshMechanism

14th October 2019 at 11:17am

Mechanisms

The refresh mechanism is the part of the WikificationMechanism concerned with updating a rendering when there are changes in the tiddler store.

The refresh mechanism is notified of changes to the tiddler store asynchronously. This is done so that multiple consecutive changes can be coalesced into a single change notification. Thus, a series of action widgets modifying several different tiddlers will only trigger a single refresh cycle.

When changes occur, the rendering is updated by calling the "refresh" method of the root widget. The refresh method determines whether the widget needs to be updated to reflect the incoming changes, and then recursively calls into the refresh methods of each child widget

The refresh cycle is inherently fairly slow because it involves visiting every node in the render tree. To maintain performance there is a RefreshThrottling mechanism that enables refresh processing to be deferred when rapid changes occur to the same tiddler.

RefreshThrottling

18th November 2021 at 12:26am

Mechanisms RefreshMechanism

The RefreshMechanism allows the refresh cycle to be throttled (or deferred) when rapid changes occur to the same tiddler. It is used to maintain responsiveness while editing a draft tiddler, but can also be used on other tiddlers.

The rules governing refresh throttling are:

- When a change notification occurs, throttling will only take place if all of the modified tiddlers meet at least one of these criteria:
  - Has the field `draft.of`
  - Has the field `throttle.refresh`
  - Has a title prefixed with `$:/temp/volatile/`
- If the refresh cycle is to be throttled, a timer is set for the internal specified in $:/config/Drafts/TypingTimeout (cancelling any previously set timer)
  - When the timer fires, the refresh cycle is triggered, passing the aggregated titles of all the deferred refresh cycles

RegExp in Tiddlywiki by Mohammad

6th January 2021 at 4:10pm

Tutorials

RegExp in Tiddlywiki contains practical use cases of regular expression in Tiddlywiki.

http://tw-regexp.tiddlyspot.com/

The regular expression is a way to describe complex search patterns using sequences of characters .

RegExp in Tiddlywiki has four main parts

- Using regular expression to match a pattern in tiddler title
- Using regular expression to match a pattern in tiddler fields excluding tiddler text (body)
- Using regular expression to match a pattern in tiddler body (text field)
- Using regular expression for validation like username, password, etc.

regexp Operator

3rd February 2015 at 8:06pm

Field Operators Filter Operators Negatable Operators

purpose	filter the input by pattern-matched field
input	a selection of titles
suffix	**F** = the name of a field, defaulting to **title**
parameter	**X** = a regular expression
output	those input tiddlers in which field **F** matches **X**
! **output**	those input tiddlers in which field **F** does *NOT* match **X**

Learn more about how to use Filters

Regular expressions are concise strings of characters that denote patterns of text to search for. The format used in TiddlyWiki is fully defined in this Mozilla reference.

The filter syntax makes it impossible to directly specify a regular expression that contains square brackets. The solution is to store the expression in a variable. See the examples.

The parameter **X** can optionally start or end with a string of flags:

TypeError: setting getter-only property "parentNode"

Only the `i` flag is generally useful: it forces the different between capital and lowercase letters to be ignored.

If **X** is empty, **regexp** will match all of the input tiddlers.

Examples

regexp Operator (Examples)

23rd January 2015 at 11:53pm

Operator Examples regexp Operator

These examples make use of the Days of the Week tiddler.

`[!is[system]regexp[Wiki]]`

→ non-system tiddlers with `Wiki` in their title

Try it

`[!is[system]regexp[(?i)Wiki]]`

→ non-system tiddlers with `Wiki` in their title, ignoring case

Try it

`[!is[system]regexp[Wiki(?i)]]`

→ same again

Try it

`[regexp[(?i)\.jpe?g$]]`

→ tiddlers with titles ending in `.jpg` or `.jpeg`, ignoring case

Try it

`[regexp:created[^201408]]`

→ tiddlers created in August 2014

Try it

The regular expression `[0-9]{2}` matches two consecutive digits. Because it contains square brackets, the way to use it with the **`regexp`** operator is via a variable, as follows:

copy to clipboard

```
<$set name="digit-pattern" value="[0-9]{2}">
<<list-links "[regexp:title<digit-pattern>]" field:"title">>
```

```
</$set>
```

That renders as:

- "デザイン刷新。次の25年を目指すTiddlyWiki" by Moongift
- $:/state/tab/sidebar--1835078512
- $:/state/tab/sidebar--595412856
- Base64
- decodebase64 Operator
- encodebase64 Operator
- Notes for upgrading to 5.0.11-beta
- Release 5.0.0-alpha.11
- Release 5.0.0-alpha.12
- Release 5.0.0-alpha.13
- Release 5.0.0-alpha.14
- Release 5.0.0-alpha.15
- Release 5.0.0-alpha.16
- Release 5.0.0-alpha.17
- Release 5.0.10-beta
- Release 5.0.11-beta
- Release 5.0.12-beta
- Release 5.0.13-beta
- Release 5.0.14-beta

- Release 5.0.15-beta
- Release 5.0.16-beta
- Release 5.0.17-beta
- Release 5.0.18-beta
- Release 5.1.10
- Release 5.1.11
- Release 5.1.12
- Release 5.1.13
- Release 5.1.14
- Release 5.1.15
- Release 5.1.16
- Release 5.1.17
- Release 5.1.18
- Release 5.1.19
- Release 5.1.20
- Release 5.1.21
- Release 5.1.22
- Release 5.1.23
- sha256 Operator
- sha256 Operator (Examples)
- TiddlyWiki European Meetup 2016
- TiddlyWiki European Meetup 2017

- TiddlyWiki extensions for Sublime Text 3 by roma0104

Release 5.0.0-alpha.11

8th November 2013 at 6:54pm

AlphaReleaseNotes

This release is the culmination of a large scale refactoring of the widget mechanism of TiddlyWiki5. There are several changes to be aware of if upgrading from earlier versions:

- The following widgets have been removed:
  - `<$setstyle>` - use `<div style=<<macroName Param>>>` instead
  - `<$video>` - will return in a later release
  - `<$datauri>` - use the `<<makedatauri>>` built-in macro instead
  - `<$error>` - may return in a later release
  - `<$import>` - use the BrowseWidget, DropzoneWidget and NavigatorWidget instead
  - `<$info>` - use the `<<changecount>>` built-in macro instead
  - `<$version>` - use the `<<version>>` built-in macro instead
- The following widgets have had significant changes:
  - EditWidget
  - ListWidget - the list widget itself no longer generates HTML nodes, so you'll often need to wrap the template in a `<div>` or a `<span>` to be able to style the content
  - ViewWidget - has several changes:

    - `<$view format="link"/>` is no longer available; use an explicit `<$link>` widget instead
    - `<$view format="link"/>` is no longer available; use the TranscludeWidget instead
- The following new widgets have been added:
    - BrowseWidget
    - DropzoneWidget
    - EditTextWidget and EditBitmapWidget
    - SetVariableWidget
- Widget attribute names have been made more consistent. In particular, `tiddler` is used to reference a tiddler by title, not `title`
- It is no longer possible to import the macro definitions within another tiddler by transcluding that tiddler
- The `body` element now has the class `tw-body`, which will need to be specified in any overrides
    - This was done due to the new support for HTML foreign objects, which makes it possible to have multiple `<body>` elements in a document

Release 5.0.0-alpha.12

8th November 2013 at 7:54pm

AlphaReleaseNotes

See GitHub for detailed change history of this release

This minor release just contains fixes for the syncer and related adaptor modules for the TiddlyWeb edition and for TiddlyWiki5's integrated server.

Release 5.0.0-alpha.13

9th November 2013 at 12:22pm

AlphaReleaseNotes

See GitHub for detailed change history of this release

This release has several bug fixes:

- Fixes to *DaveGifford*'s themes **blanca**, **blue** and **rocker**
- Fix an issue with the `<$edit-text>` widget
- Documentation updates
- Improved deployment scripts
- Made the modifier field in the subtitle be a link
- Styling improvements for the *CodeMirror* plugin
- Improved the ViewWidget so that it falls back to displaying its content if the field/property is missing or empty
- Extend use of the built-in `<<tabs>>` macro to the tiddler info panel, the control panel and the sidebar
    - This means that you can add new tabs by creating tiddlers with these tags, and optionally a `caption` field for the text of the tab:
        - **$:/tags/TiddlerInfo** for tiddler info panel tabs
        - **$:/tags/ControlPanel** for control panel tabs
        - **$:/tags/SideBar** for sidebar tabs
        - **$:/tags/MoreSideBar** for tabs in the "more" sidebar
        - Change the order of tabs by adjusting the `list` field of the corresponding tag tiddler

Release 5.0.0-alpha.14

10th November 2013 at 8:22pm

AlphaReleaseNotes

See GitHub for detailed change history of this release

This release has several bug fixes:

- Improved the layout of the **control panel**
- Fixed problem with using the CheckboxWidget to apply tags to tiddlers that don't have any existing tags
- Fixed problem with default password for the PasswordWidget being the string "null"

Release 5.0.0-alpha.15

12th November 2013 at 9:50pm

AlphaReleaseNotes

See GitHub for detailed change history of this release

New Features

- Added a dropdown to the edit template for setting the tiddler type
- A saver module for Microsoft Internet Explorer version 10 and above. Clicking save in the sidebar causes the browser to pull up a bar at the bottom of the window where you can click **save**. You then get a new copy of your wiki in the downloads folder.
- Support for new `tw-close-other-tiddlers` message (eg, close others)
- For http://five.tiddlywiki.com, add a tiddler info tab with a link to the static representation of the tiddler

- Make more UI elements extensible via system tags:
  - **$:/tags/ViewToolbar** for the view mode tiddler toolbar
  - **$:/tags/EditTemplate** for the edit template
  - **$:/tags/EditToolbar** for the edit mode tiddler toolbar
  - **$:/tags/PageControls** for the page control tools in the sidebar

Improvements

- Rename the `<$setvariable>` to `<$set>`
  - `<$setvariable>` will temporarily remain as a synonym for `<$set>` for the next few releases
- Improve the popup mechanism so that the tiddler info panel doesn't close so easily
- Various improvements for working with TiddlyWeb, including:
  - Updated control panel

Bug fixes

- Fixed bug when creating a tiddler title starting or ending with a space
- Fixed behaviour of tags editor dropdown when search box is empty
- Fixed problem with interpretation of `fields` and `index` attributes of the TranscludeWidget
- Fixed the module type names in the internal tab of the control panel
- Improved styling for embedded PDFs
- Fixed bug with second being omitted from serialised date formats

Contributors to this release include @jermolene and @grayeul.

Release 5.0.0-alpha.16

30th November 2013 at 10:55am

AlphaReleaseNotes

See GitHub for detailed change history of this release

New Features and Improvements

- Improved appearance of tags editor
- Improved generation of qualified tiddler titles for state storage
  - Instead of `$:/state/tab/sidebar-{$:/core/ui/SideBar|$:/core/ui/SideBar||}{$:/core/ui/PageTemplate|$:/core/ui/PageTemplate||}` one gets `$:/state/tab/sidebar-{1743827719}`
- Significant updates to the structure and content of the user documentation
- Added a new plugin for displaying corner ribbons; currently used for a version banner on tw5.com
- Split **Snow White** theme into a base **Vanilla** theme with all the basic formatting, leaving the decorative bits to **Snow White**
  - Existing TiddlyWikiFolders will need updating to include **Vanilla** as an additional theme
- The page building blocks are now driven by the system tag `$:/tags/PageTemplate`
  - *$:/TopSideBar* and *$:/LeftSideBar* are no longer specially treated; use the new tag instead
- Fixed problem that prevented tag configured items from shadow tiddlers interleaving with items from ordinary tiddlers
- Refactored control panel to add **Saving** tab that includes *TiddlySpot* options
- Improved notifications when saving to *TiddlySpot*

- Added backup URL to *TiddlySpot* control panel tab
- Extended the ServerCommand to add primitive support for basic authentication when running under Node.js

Bug fixes

- Fixed problem with displaying tiddler titles that contain WikiText syntax
- No longer crashes when encountering an error in a filter string
- Fixed a crash when dragging and dropping tiddlers within TiddlyWiki5
- No longer update the modified date for imported tiddlers
- Automatically forces new field names to be lower case

Internal changes

- Added Windows-compatible build scripts
- Changes to the SavingMechanism to allow the tiddlers that are saved to be selected
  - This enables the wiki at http://five.tiddlywiki.com/index.html to generate an empty wiki
- The main HTML file template used for saving TiddlyWiki5 documents has changed from `$:/core/templates/tiddlywiki5.template.html` to `$:/core/save/all`
- Moved tw5.com's Google Analytics integration into a proper plugin

Contributors to this release include @jermolene, @asampal and @pmario.

Release 5.0.0-alpha.17

30th November 2013 at 4:16pm

AlphaReleaseNotes

See GitHub for detailed change history of this release

Bug fixes

- Fixes a small but important bug that prevented downloading an empty wiki with alpha.16

Release 5.0.1-alpha

6th December 2013 at 5:13pm

AlphaReleaseNotes

See GitHub for detailed change history of this release

Improvements

- Changes to the importing process to enable a smoother Upgrading process
  - Ignores attempts to import plugins that are older than currently installed plugins
  - System tiddlers are now imported as usual
- If `$:/theme` isn't defined or refers to a missing tiddler, then fallback through **Snow White** to **Vanilla**. This means that `empty.html` now defaults to **Snow White**
- Added support for Block Quotes in WikiText
  - Contributed by *StephanHradek* (@Skeeve on GitHub)

Bug fixes

- Fixed bug that was preventing `$:/tags/PageControls` tiddlers from being reordered

Internal changes

- Changed the TiddlyWiki5 Versioning policy
  - Beta releases will be `5.0.x-beta` and the final release will be `5.1.x`

Release 5.0.10-beta

18th January 2015 at 7:24pm

BetaReleaseNotes

See GitHub for detailed change history of this release

Highlights

- Added new syntax for Images in WikiText and a new ImageWidget
- Added support for a SafeMode that disables customisations

Documentation Improvements

- Added DateFormat documentation

Usability Improvements

- Refactor sorting to respect accented characters
- Support embedded images in Markdown tiddlers
- Refactored sidebar hiding action so that the story river border is maintained

Hackability Improvements

- Added `[is[tag]]` to the **is** operator
- Hide the top bars in the print stylesheet

Bug Fixes

- Added support for saving changes on Windows network drives
- Refactored configuration processing so that ordinary tiddlers are processed after shadow tiddlers. This resolves an issue whereby user stylesheets were being overridden by shadow stylesheets
- Fixed issue with offline snapshot of server edition erroneously including shadow tiddlers
- Fixed problem with corrupted upgrades from 5.0.x-prerelease to 5.0.x-beta
- Fixed bug with `[untagged[]]` filter operator
- Fixed incorrect background colour for tag pills in the sidebar
- Fixed issues with null fields under TiddlyWeb
- Fixed problem with hamburger overlapping scrollbars

Release 5.0.11-beta

21st February 2015 at 11:57pm

BetaReleaseNotes

See GitHub for detailed change history of this release

Incompatible Changes

See Notes for upgrading to 5.0.11-beta for more details of these changes:

- The default output location for command line operations has changed to the `/output` folder within TiddlyWikiFolders (it used to be current directory)
- Note that the shadow tiddler *$:/core/ui/PageMacros* has changed with this release. If you've made modifications to it you should copy your modifications to a new tiddler, delete *$:/core/ui/PageMacros*, perform the upgrade and then manually update the new copy of *$:/core/ui/PageMacros*

Documentation Improvements

- Added more warnings about taking care to backup your data

Usability Improvements

- Many performance optimisations, particularly for filter operations
- Added support for PermaLinks and browser back and forward buttons

- Added support for *WikiLinks* in Markdown via `[link text](#TiddlerTitle)`
- Added support for explicit external Linking in WikiText (eg `[ext[tooltip|url]]`)
- Replaced hamburger menu icon with double chevron icon
- Enhance plain text parsing to use the CodeBlockWidget, and hence use syntax highlighting if the plugin is installed. This gives us syntax highlighting for JavaScript shadow tiddlers, amongst other things
- Improvements to the German, French, Italian, Japanese and Chinese translations
- Removed WikiText tiddlers from the sidebar "Types" tab
- Added various warnings when editing shadow tiddlers
- Allow digits in field names
- Added warning banner and when attempting to edit binary tiddlers
- Added automatic refreshing of the browser window title from **$:/core/wiki/title**

Hackability Improvements

- Improved vertical layouts of **`tabs`** macro
- Added "standard" tab to **advanced search**
- Added new filter operators **`before`** and **`after`**
- Added new **`get`** operator
- Added BuildCommand, OutputCommand and ClearPasswordCommand
- Added new extensible *StartupMechanism* for orchestrating startup tasks
- Updated to new version of *CodeMirror*
- Added https://github.com/Jermolene/TiddlyWiki5NodeWebkit, a demo of using TiddlyWiki as a library in a node-webkit application
- Extended `.tid` TiddlerFiles to allow single line text fields

- Renamed `$:/ShowEditPreview` to `$:/state/showeditpreview`

Bug Fixes

- Fixed issue with hovering of the topbar icons in Firefox
- Fixed problem with dragging and dropping within a text edit control

Contributors

I (@Jermolene) would like to thank the contributors to this release who have generously given their time to helping improve TiddlyWiki:

@BramChen, @mwfogleman, @nameanyone, @natecain, @pekopeko1, @pmario, @sukima, @xcazin.

Release 5.0.12-beta

17th May 2014 at 3:02am

BetaReleaseNotes

See GitHub for detailed change history of this release

This is an emergency release that fixes a single significant bug with shadow tiddler handling in 5.0.11. See the discussion thread for more details.

Release 5.0.13-beta

18th January 2015 at 7:25pm

BetaReleaseNotes

See GitHub for detailed change history of this release

Accessibility Improvements

This release includes a number of features designed to improve the experience of TiddlyWiki with a screen reader. Particular thanks to @domasofan for his feedback

- Added label and title for show/hide sidebar buttons
- Added label and title for advanced search link
- Added label and title for close buttons in "Open" sidebar tab
- Added **aria-label** support to the LinkWidget
- Switched to H1 tags for the page title and H2 tags for tiddler titles
- Added label and title for tiddler and page toolbar buttons

Usability Improvements

- Added support for global Macros in WikiText via the new ImportVariablesWidget
- Added new SelectWidget
- Improved behaviour of `text/html` tiddlers so that they are displayed within an HTML iframe element
- Added group headings to the content type dropdown in edit mode
- Improved advanced navigation settings in **$:/ControlPanel** to use radio buttons
- Improved (and here) behaviour of PermaLinks so that now https://tiddlywiki.com/#HelloThere just opens the single specified tiddler

- Added support for triple-quoted multi-line attributes with macros, widgets and HTML elements
- Added TableOfContents tab for tw5.com edition

Hackability Improvements

- Restored **escape** as a shortcut to abandon edits
- Added support for ExternalImages that are referenced by URI and not stored in the TiddlyWiki HTML file
- Added new SetFieldCommand
- Added new SaveTiddlersCommand
- Added support for system tag *$:/tags/RawMarkup*
- Added editor type mappings to control panel advanced tab
- Refactored *CodeMirror* (https://tiddlywiki.com/codemirrordemo.html) plugin for simpler configuration
- Added support for Environment Variables on Node.js
- Extended ServerCommand to allow a path prefix
- Extended the `listed` operator to use any list field

Bug Fixes

- Fixed here and here the problem with state tiddlers being inadvertently saved
- Fixed bug with TextWidget not refreshing correctly
- Fixed problem with vertical tabs in Safari
- Fixed issue with images in Markdown

Contributors

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:

- @BramChen
- @buggyj
- @csugden
- @danielo515
- @IreneKnapp
- @jayfresh
- @mwfogleman
- @xcazin

Release 5.0.14-beta

21st February 2015 at 11:57pm

BetaReleaseNotes

See GitHub for detailed change history of this release

Major Changes

Improved Control Panel Plugins Tab

The **Plugins** tab of the **control panel** has been improved with the addition of documentation tabs that plugins can use to show documentation.

There is also a larger draggable area for dragging tiddlers across to other wikis.

Configurable Button Toolbars

The page toolbar and the tiddler toolbars can now be configured with several new buttons - see the **control panel** under the **Appearance** and **Toolbars** tabs.

The sidebar tools tab now shows all the available page controls, allowing them to be invoked or checked to appear in the toolbar. The tiddler info area now includes a tools tab that includes buttons for all the tiddler actions.

Upgrade Mechanism

There are two components:

- A more flexible *ImportMechanism* that:
  - Presents incoming tiddlers as a pending import list that allows the user to inspect them and, if necessary, explicitly deselect them from the actual import
  - Provides *UpgraderModules* with an opportunity to process each incoming tiddler
    - The **plugin upgrader** module handles version checking of plugins and upgrading them from a special *UpgradeLibrary* plugin tiddler
    - The **system upgrader** module is responsible for suppressing the importing of certain system tiddlers (currently $:/StoryList and $:/HistoryList)
    - The **themetweak upgrader** module handles migrating theme tweaks from their pre-5.0.14-beta format (see below)
- An *UpgradePlugin* and associated edition that provides a custom, single-purpose user interface for upgrading standalone TiddlyWiki files - see https://tiddlywiki.com/upgrade.html

Improvements to CamelCase Recognition

TiddlyWiki now takes a much more conservative approach to recognising CamelCase terms that should be automatically linked. Previously, the dash and underscore were treated as lower case letters, leading to a number of false positives. See the GitHub bug #337 for details.

Automatic Permalinking Off by Default

With previous beta releases of TiddlyWiki the browser address bar is automatically updated so that it dynamically reflects the tiddlers that are currently open. This makes it easier to get a permalink for copying and pasting elsewhere, but it leads to much confusion for casual users who don't always understand why unexpected tiddlers are being displayed after they have refreshed the page in the browser.

For 5.0.14-beta, the setting has been changed. Visit **control panel** **Advanced**/**Settings** to switch the setting back to "Include the target tiddler and the current story sequence".

Usability Improvements

- Added previews to the icon dropdown in the **tag manager**
- Added an indication of unsaved changes by changing the colour of the save changes button
- Split the wikitext emphasis parsers into separate modules so that they can be independently controlled with the `\rules` pragma
- Refactored (and here) *ReleaseHistory* to place the releases into vertical tabs
- Stopped saving $:/HistoryList, thus avoiding it uncontrollably increasing in size
- Improved (and here) print stylesheet to remove page background
- Updated display of plugin tiddlers so that their constituent tiddlers are shown, rather than the raw JSON
- Moved the functionality of the fullscreen plugin into the core

Hackability Improvements

- Extended TiddlerWidget to add a CSS class corresponding to each tag present on displayed tiddlers
- Added support for hiding specified fields in the tiddler editor
- Added support for variable operands in filters (see Filter Parameter)
- Added support for widget messages *tw-permalink* and *tw-permaview*
- Added browser sniffing plugin so that tiddlywiki.com can present the correct browser-specific documentation
- Added a configuration option for specifying the default location for saving new tiddlers in the client-server configuration
- Added support for the InfoMechanism
- Extended CheckboxWidget to allow it to toggle fields as well as tags
- Refactored **control panel** theme tweaks to be stored in individual tiddlers
- Extend the **tabs** macro to allow tabs to be templated
- Added (and here) SystemTags support for inserting content above and below the story river
- Added support for transcluding plugin subtiddlers with the TranscludeWidget
- Added (and here) support for importing from `*.htm` and `*.hta` files as well as the existing support for `*.html` files
- Extended the "sticky titles" theme so that it works when tiddlers are in edit mode

Bug Fixes

- Fixed bug with execution order of BuildCommand targets
- Fixed bug with missing hover colours for external links
- Fixed problem with refreshing modal dialogues
- Fixed issue with cookies disabled on Firefox

- Relax the requirement for a newline immediately the closing `\end` of a macro definition

Contributors

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:

- @BramChen
- @pmario
- @ssokolow
- @xcazin

Release 5.0.15-beta

21st February 2015 at 8:40pm

BetaReleaseNotes

See GitHub for detailed change history of this release

Usability Improvements

- Simplified shadow default tiddlers
- Added the ability to disable plugins (see PluginMechanism) with a user interface in **control panel**
- Added page control button for invoking the **tag manager**
- Simplified downloading an offline copy of a client-server wiki
- Blocked temporary state tiddlers from import/upgrade

Hackability Improvements

- Added a banner to the edit template encouraging improvements to the documentation
- Refactored `saver-handler.js` out of `syncer.js`
- Added Table-of-Contents Macros
- Simplified startup module organisation (see *Startup Modules.svg*)
- Changed the overlay text for the DropzoneWidget to make it translateable
- Introduced a new *refresh button* and reverted *home button* behaviour
- Added **alt** attribute to the ImageWidget

Bug Fixes

- Fixed problem with building TiddlyWiki under Windows
- Fixed unclickable download ink in upgrade wizard
- Fixed missing language flag in **empty.html**
- Fixed problem with switching the type of a tiddler between the bitmap and text editor
- Fixed foreground colour for tag pills in the sidebar
- Fixed problem with github source links for tiddlywiki.com not working for titles containing colons

Contributors

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:

- @BramChen

- @xcazin

Release 5.0.16-beta

21st February 2015 at 8:41pm

BetaReleaseNotes

See GitHub for detailed change history of this release

Incompatible Changes

5.0.16-beta brings more incompatible changes than any previous release. These changes are likely to break almost all plugins written for previous releases of TiddlyWiki 5, and will break many customisations.

- Changed all CSS class prefixes from `tw-` to `tc-` (eg `tw-tiddler-frame` has become `tc-tiddler-frame`; missing prefixes have also been added, so `btn-invisible` has become `tc-btn-invisible`)
- Changed all message prefixes from `tw-` to `tm-` (eg `tw-close-tiddler` has become `tm-close-tiddler`)
- Changed all variable prefixes from `tw-` to `tv-` (eg `tw-config-toolbar-icons` has become `tv-config-toolbar-icons`)
- Changed prefix/removeprefix filter operators to be case-sensitive

Compatibility with TiddlyWikiClassic

This release includes a preliminary version of a new plugin that provides compatibility with content created for TiddlyWikiClassic:

https://tiddlywiki.com/classicparserdemo.html

Subsequent releases will include end-user documentation with step-by-step instructions for migrating content.

Many thanks to @buggyj for his work on this plugin.

Usability Improvements

- Amended behaviour of the *tm-browser-refresh* message so that it no longer clears the location hash
- Hide toolbar buttons from static renderings

Hackability Improvements

- Extend the **`all`** operator to be able to select all the source tiddlers
- Fixed problem with single line macro definitions incorrectly including whitespace in the value. (For example, `\define mymacro() yes` would set the macro value to " yes", with a leading space)
- Extend Table-of-Contents Macros use the caption field if present
- Configurability for the default tab shown in the tiddler info panel (see Configuring the default TiddlerInfo tab)
- Added **`suffix`** and **`removesuffix`** operators

Bug Fixes

- Reverted incorrect refreshing of the tiddler widget
- Fixed indentation of selective expandable Table-of-Contents Macros

Contributors

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:

- @BramChen
- @buggyj
- @gernert
- @pmario
- @xcazin

Release 5.0.17-beta

21st February 2015 at 11:52pm

BetaReleaseNotes

See GitHub for detailed change history of this release

Highlights

This release includes major improvements from @giffmex to the welcome and tutorial documentation, and new development docs incorporating the work of @cheigele and @cjrk.

Incompatible Changes

Change System Tag `$:/tags/stylesheet` with `$:/tags/Stylesheet`

The issue is that the capitalisation of `$:/tags/stylesheet` is not consistent with other system tags. This release adds support for `$:/`

`tags/Stylesheet`, and adds a deprecation warning if `$:/tags/stylesheet` is used. Support for `$:/tags/stylesheet` will be removed before the beta.

- Added support for `$:/tags/Stylesheet` and a deprecation warning when `$:/tags/stylesheet` is used

Change ButtonWidget `title` attribute to `tooltip`

The use of the `title` attribute in the ButtonWidget was not consistent with the `tooltip` attribute used by the ImageWidget and LinkWidget. This release adds support for the `tooltip` attribute to the ButtonWidget alongside the existing support for the `title` attribute, but with a deprecation warning.

- Added `tooltip` attribute to ButtonWidget

Removing Support for *RegExp* Filter Operands

As discussed in the associated ticket, the support for regular expression filter operands is incomplete and inconsistent. The plan is to remove support for this feature in the next release. A new filter operator has been provided to replace it, along with the addition of a deprecation warning that is displayed whenever regular expression filter operators are used.

- Added new **`regexp`** operator
- Added deprecation warning for regular expression filter operators

Repository Reorganisation

The goal is to make the TiddlyWiki repository easier to comprehend for newcomers.

- Moved scripts into a separate `bin` folder

Usability Improvements

- Reorganised the **control panel** to make it more logical and usable

Hackability Improvements

- Moved developer documentation to https://tiddlywiki.com/dev
- Improvements (and here) to `serve.sh` / `serve.cmd` scripts
- Disable plugins when in SafeMode
- Added new **`timeline`** macro
- Added new **`dumpvariables`** macro
- Extended Table-of-Contents Macros so that individual links can be disabled
- Removed (and here) unnecessary `<p>` tag from edit tags and types dropdowns
- Added **multiple** attribute to the BrowseWidget
- Extended view template to hide bodies of tiddlers with field **hide-body** set to **yes**

Bug Fixes

- Fixed problem with draft tiddlers not counting as dirty
- Fixed problem with keyboard shortcuts introduced in 5.0.16-beta
- Fixed problem with stylesheets being parsed in inline mode
- Fixed bug with selective expandable Table-of-Contents Macros

Contributors

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:

- @BramChen
- @buggyj
- @cheigele
- @cjrk
- @giffmex
- @pmario
- @xcazin

Release 5.0.18-beta

22nd February 2015 at 12:01am

BetaReleaseNotes

See GitHub for detailed change history of this release

This is a minor release prior to the full release of TiddlyWiki on September 20th. The documentation has been cleaned up and improved (with more improvements to come).

File Layout of tiddlywiki.com

The layout of files on tiddlywiki.com has been adjusted to make it more logical. See the ticket for a discussion. You can see the source files that

make up tiddlywiki.com at https://github.com/Jermolene/jermolene.github.com

Hackability Improvements

- Added first iteration of a `list-links` macro (further improvements are planned)
- Added support for importing `.markdown` and `.md` files
- Extended `timeline` macro to support a subfilter

Bug Fixes

- Fixed problem with digits being classified as lower case letters for wiki link matching
- Fixed crash when sorting missing tiddlers by fields other than title
- Fixed problem with handling `.jpeg` file extensions
- Fixed problem with RadioWidget and missing tiddlers
- Fixed problem with dragging a partially selected link

Contributors

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:

- @BramChen
- @buggyj
- @Eucaly
- @pmario

- @xcazin

Release 5.0.2-beta

15th December 2013 at 3:24pm

BetaReleaseNotes

See GitHub for detailed change history of this release

Improvements

- Added support for Hard Linebreaks in WikiText
- Added a new RadioWidget, contributed by *StephanHradek* (@Skeeve on GitHub)
- Two new savers for *InternetExplorer*, both contributed by *DavidJade* (@davidjade on GitHub)
  - **tiddlyie** for saving with TiddlyIE, an equivalent of TiddlyFox for *InternetExplorer*
  - **fsosaver** that can *save changes directly with InternetExplorer* but requires the HTML file to be renamed `*.hta`

Bug fixes

- Fixed problem deleting tiddlers under the filesystemadaptor
- Fixed problem with transcluding fields containing lists or dates (eg, `{{!!tags}}`)
- Fixed problem with re-ordering page control tiddlers (ie tiddlers tagged `$:/tags/PageControls`)
- Update template used for deploying to TiddlyWeb
- Fixed problem with tiddler deletions via the file system adaptor

Internal changes

- Allowed variable number of arguments to *JavaScript Macros*
  - Contributed by *StephanHradek* (@Skeeve on GitHub)
- Changed the format of plugin tiddlers to remove duplicated information (see the PluginMechanism)

Release 5.0.3-beta

15th December 2013 at 6:00pm

BetaReleaseNotes

See GitHub for detailed change history of this release

This release includes minor bug fixes and documentation updates.

Release 5.0.4-beta

21st December 2013 at 9:55pm

BetaReleaseNotes

See GitHub for detailed change history of this release

Improvements

- Switched to using an HTML5 placeholder for the default text of a new tiddler
- Refactored stylesheet handling so that theme tweaks are now applied immediately, and some ordering issues have been resolved

- Several improvements for running TiddlyWiki on Node.js:
  - The `tiddlywiki.info` file in the root of wiki folder file is now optional, falling back to a default configuration suitable for serverside use (commit)
  - The file system plugin now automatically creates the `tiddlers` subfolder in the wiki folder
- Added *favicon.ico support*
- Added SaveTiddlerCommand

Bug Fixes

- Fixed problem with tiddlers sometimes wrongly opening at the top of the story (commit)
- Fixed problem with refreshing RadioWidget
- Fixed problem with the dragger image being visible in some circumstances
- Fixed drag and drop support on *InternetExplorer*, contributed by *DavidJade* (@davidjade on GitHub)
- Fixed crash on modifying page template (commit)
- Updated fullscreen plugin to latest version of the API (commit)
- Improved handling of double square brackets within tags (thanks to *StephanHradek*)

This release includes minor bug fixes and documentation updates.

Release 5.0.5-beta

24th December 2013 at 3:20pm

BetaReleaseNotes

See GitHub for detailed change history of this release

Improvements

- Added support for Setting a favicon

Bug Fixes

- Fixed problem with modal wizard positioning on narrow screens
- Fix problem with static content being included in empty.html when downloaded from the full wiki
- Improved performance of tiddler import with large tiddlers on Firefox

Release 5.0.6-beta

3rd January 2014 at 5:49pm

BetaReleaseNotes

See GitHub for detailed change history of this release

Improvements

- Added support for numeric sorting to *TiddlerFilters* (thanks to *StephanHradek*)
- Extended Tables in WikiText to allow for vertical alignment of cells (thanks to *StephanHradek*)
- Added experimental support for running *TiddlyWiki on node-webkit*
- Added an **$:/AdvancedSearch**

- Added support for specifying hostname for the ServerCommand
- Added a first pass at generating an all-in-one static HTML representation of a wiki, complete with internal anchor links for wiki links
  - See https://tiddlywiki.com/alltiddlers.html
- Hide encryption features when running on the server
- Added double backticks as an alternative for delimitting inline code (thanks to *StephanHradek*)

Bug Fixes

- Fixed import **browse** button
- Fixed crash when attempting full screen mode on browsers that don't support it
- Fixed requirement for a newline immediately after a horizontal rule, table or typed block (thanks to *StephanHradek*)
- Fixed problem with **placeholder** attibute on *InternetExplorer* (thanks to *DavidJade*)

Release 5.0.7-beta

25th January 2014 at 8:10pm

BetaReleaseNotes

See GitHub for detailed change history of this release

Incompatible Changes

These are changes that might affect users upgrading from previous betas.

- The node-webkit saver has been moved into a plugin - $:/plugins/tiddlywiki/nodewebkitsaver. The plugin is only needed when embedding

a single TiddlyWiki in node-webkit and is not required for TiddlyDesktop, which as of v.0.0.2 uses the existing TiddlyFox saver.

Documentation updates

- A new video tutorial: TiddlyWiki on Firefox for Android Video
- A first pass at *TiddlyWiki Coding Style Guidelines*
- Added an explanation of TemplateTiddlers
- Added documentation for saving on a custom PHP server
- Added documentation for TextWidget
- Summary of the approach for Naming of System Tiddlers

Improvements

- Added support for importing encrypted TiddlyWiki documents
- Added several new *filter operators*, including **reverse**, **first**, **last**, **butlast**, **rest**, **nth**
- Extend **list** filter operator to allow other fields to be used via a TextReference
- Added shortcut in **$:/ControlPanel Basics** tab for setting $:/DefaultTiddlers to retain tiddler story ordering
- Added emacs and vim keymapping support to the CodeMirror plugin, (thanks to João Bolila, @jbolila on GitHub)
- Added the highlight.js syntax highlighting plugin: https://tiddlywiki.com/highlightdemo.html (thanks to João Bolila, @jbolila on GitHub)
- Added the first export option to the **Tools** tab of the **control panel**
- Added an ellipsis for **advanced search** next to the search box in the sidebar
- Adjusted the password dialogue to make it narrower for smaller screens
- Adjusted the display of tiddler dictionaries to make them display as plain text

Bug Fixes

- Fixed problem with default format of ViewWidget substituting seconds for minutes in displayed times
- Fixed problem that was causing several animations to fail in Safari
- Fixed problem with unclickable sidebar under the Centralised theme
- Fixed problem with non-system tiddlers showing up in system tiddler **advanced search**
- Fixed problem with notifications not always disappearing in Firefox
- Fixed problem with wiki folders including wiki files with tiddlers that do not have a title
- Fixed problem with ServerCommand logging "Serving on undefined:8080"

Release 5.0.8-beta

28th February 2014 at 4:49pm

BetaReleaseNotes

See GitHub for detailed change history of this release

Incompatible Changes

See Notes for upgrading to 5.0.8-beta for more details of these changes:

- Changed rules for parsing content of HTML elements
- Switched *SiteTitle* and *SiteSubtitle* to $:/SiteTitle and $:/SiteSubtitle
- Changes to commands used with TiddlyWiki on Node.js

- Changes to naming of some editions
  - Notably **clientserver** has changed to **server**

Documentation Improvements

- Improved documentation for *TiddlerFilters*, including a notation of the *TiddlerFilter Formal Grammar* contributed by @Tikkoneus
- Improved documentation for *WidgetMessages*
- Tiddlers on tiddlywiki.com now have a link to the original source on GitHub for pull requests (see the "Sources" tab of the tiddler info panel)

Usability Improvements

- Added automatic saving on editing a tiddler and a warning when attempting to close the window with unsaved changes
- Added confirmation dialogue when deleting tiddlers
- Add support for switchable (and editable) ColourPalettes
- Added TranslationMechanism and translations for:
  - Deutsch (Österreich and Deutschland) by @pmario
  - Français (France) by @xcazin
  - Chinese (Simplified and Traditional) by @*BramChen*
- Add error alerts when syncing to a server
- Rejigged **$:/ControlPanel** to use nested tabs
- Added **$:/TagManager** for easy management of tags
- Added experimental KeyboardWidget, including support for **ctrl-enter** (or **cmd-enter**) to finish editing a tiddler

- Added number of tags to control panel **Basics** tab
- Enhanced link handling so that control/command clicking a link opens the target tiddler without navigating to it
- Importing tiddlers via drag and drop no longer opens all the tiddlers

Scalability Improvements

- Added support for LazyLoading of images
- Extended RevealWidget to allow control over content retention

Hackability Improvements

- Added HelpCommand and InitCommand for TiddlyWiki on Node.js
- Extended ButtonWidget to allow navigating to a tiddler
- Added experimental support for building plugins in the browser: *How to create plugins in the browser*
- Extend the TranscludeWidget to display its content as a fallback if the tiddler or field is missing
- Add logging and AlertMechanism
- Added a **Filter** tab to **advanced search**
- Add **indexes** *filter operator*
- Control over line width and colour for EditBitmapWidget
- Add support for `.multids` MultiTiddlerFiles
- Extend ViewWidget so that it works with indexes
- Added support for Tank, a new service built on TiddlyWeb: https://tank.peermore.com
- Extend relative dates to work in the future

Bug Fixes

- Fixed problem with pasting items into the browser
- Fixed problem with colour pickers not showing correct colour in Chrome
- Fixed problem with tag pills not working inside table cells
- Fixed problem in client server configuration with **%** in tiddler titles
- Fixed problem with not using placeholder text when editing missing tiddlers
- Fixed problem with test data in certain time zones
- Fixed problem with non-breaking space characters being converted into '@' symbols
- Fixed problem with highlighting plugin only working in the browser
- Fixed crash when wiki/themes folder contains files that are not themes

Release 5.0.9-beta

2nd March 2014 at 7:09pm

BetaReleaseNotes

See GitHub for detailed change history of this release

Highlights

- Improved layout, including a **hamburger** icon for dismissing the sidebar and expanding the story river to fill the space
- Added new **Seamless** theme
- New **Filter** tab in **$:/AdvancedSearch**

- Initial implementation of *CecilyView*
- Overhaul of inconsistencies in *TiddlerFilters* (see Changes to filters in 5.0.9-beta)
- New translations for Italian and Japanese
- Performance improvements, particularly during editing

Documentation Improvements

- Improved and reorganised documentation for *TiddlerFilters*
- Demo of Making curved text with SVG
- Community links are now broken up into individual tiddlers
- Added overview diagram of *TiddlyWiki Architecture*
- Added documentation from buggyj: *Developing plugins using Node.js and GitHub*

Usability Improvements

- Made the dropdown arrow icon skinnier
- Added validation of legal characters for fieldnames
- Added blacklisting of unsafe HTML elements and attributes
- Added a warning indicator to tiddlers in TiddlyWikiClassic format
- Add tiddler info **Advanced** panel with information about plugins and shadow tiddlers
- Improved layout of **$:/ControlPanel** **Plugins** tab
- Enhance viewing of system tiddlers by fading down the `$:/` prefix
- Extend **$:/TagManager** to allow icons to be assigned to tags

- Added support for `list-before` and `list-after` fields for controlling tag ordering (see *TiddlerTags* for details)
- Added request for confirmation before abandoning edits to a tiddler

Hackability Improvements

- Updated Transclusion in WikiText syntax to allow translusion of a template without affecting the current tiddler
- Added `sourceURL` handling to `eval()` so that tiddler modules can be properly debugged in Chrome
- New ScrollableWidget giving better control over scrollable regions
- Added new CSS class `tw-site-title` for the site title
- Disable the TiddlyWeb sync adaptor unless the wiki is loaded over HTTP
- Added a high resolution timer mechanism and a performance measurement mechanism
- Added a basic CSV parser
- Several measures to enforce the TiddlyWiki programming model:
  - Refactor wiki store object to make members be private
  - Freeze tiddler object and fields tiddler fields to enforce their immutability
- Extend the ButtonWidget to be able to set text references
- Add a class to tiddler frames in view mode
- Added support for *WidgetMessage: tw-home*
- Hidden system tags from the sidebar **Tags** tab
- Allow pasting and import of HTML content
- Add support for a tooltip on the LinkWidget

Bug Fixes

- Fixed problem with occasional freezes of the sync mechanism - thanks to buggyj
- Fixed problem with tiddlers or fields called `__proto__`
- Fixed with refreshing the D3 plugin
- Fixed problem with "null" message when unloading under Internet Explorer 11

Release 5.1.0

20th September 2014 at 2:40pm

ReleaseNotes

See GitHub for detailed change history of this release

This is the first full release of TiddlyWiki. Documentation updates made up the bulk of the changes since the previous 5.0.18-beta release.

Hackability Improvements

- Removed curly braces from qualified identifiers

Bug Fixes

- Fixed problem with Markdown Maruku mode metadata

Contributors

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:

- @BramChen
- @malgam
- @gernert
- @pmario
- @xcazin

Release 5.1.1

21st September 2014 at 2:40pm

ReleaseNotes

See GitHub for detailed change history of this release

This is a minor bug fix release to improve the documentation and correct some issues with the first full release of TiddlyWiki.

Incompatible Changes

Certain features that were deprecated in Release 5.0.17-beta have now been removed:

- Removed deprecated `title` attribute on the ButtonWidget
- Removed support for deprecated `$:/tags/stylesheet` system tag

Note that the next release will also remove support for regular expression filter operands, which was also deprecated in 5.0.17-beta.

Hackability Improvements

- Added first implementation of the KaTeX Plugin for mathematical typesetting

Contributors

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:

- @pmario
- @TheDiveO

Release 5.1.10

8th January 2016 at 12:16am

ReleaseNotes

See GitHub for detailed change history of this release

Major Improvements

Performance Optimisations

This release includes several low-level performance optimisations that should improve speed and responsiveness in many common operations.

- Caching the results of parsing a tiddler
- Loop optimisations

- Tiddler iteration optimisations
- State qualifier generation
- Caching data tiddlers
- Removed expensive visual effects

There is a new setting in the control panel to enable performance instrumentation, making it much easier to measure and improve performance issues.

New Editions

Three new editions of TiddlyWiki are included in this release. These can be used as starting points for creating your own wiki for several purposes:

- The Résumé Builder Edition by @inmysocks is a custom edition to guide you through the process of using TiddlyWiki to create a good looking résumé (or curriculum vitæ)
- The Blog Edition provides tools for using TiddlyWiki under Node.js to create a static HTML blog that can be published on GitHub pages (or similar)
- The Text-Slicer Edition is a custom edition with tools to help advanced users slice longer texts up into individual tiddlers.

External Text Tiddlers

Limited support for tiddlers stored in external `.tid` files:

- standalone TiddlyWiki HTML files with external text tiddlers can be built under Node.js
- wikis with external text tiddlers can be worked with in the browser, automatically lazily loading the content of external text tiddlers when it is

first referenced

- saving changes in the browser doesn't work as expected: if edited, the external text tiddler is replaced with an ordinary tiddler
- lazy loading of external text tiddlers doesn't work in Chrome when viewing the TiddlyWiki HTML file on a **file:** URI; it works OK in Firefox
- lazy loading works on an HTTP URI on all browsers as long as the target file is either stored on the same domain or is served by a server that supports CORS

See Alice in Wonderland for an example. Try opening it without a network connection.

New "Tight" Theme

A new theme called "Tight" uses reduced padding, margins and font-sizes to pack as much text as possible:

Try it out Revert

Other Improvements

Translation Improvements

- Added Swedish translation by @Superdos
- Added Korean (Korean Republic) translation by @araname
- Improved Catalan, Chinese, Danish, Dutch, French, German, Italian, Russian and Spanish translations

Usability Improvements

- Enhanced the tiddler editor to accept new fields and tags when clicking confirm, without having to click "Add tag" or "Add field"
- Added new **fold** and **fold others** buttons to the tiddler toolbar to temporarily hide the body of a tiddler, and also added new **fold all** and **unfold all** page buttons
- Added ability to disable individual WikiText parser rules (see the **Advanced** tab of **$:/ControlPanel**). Also added a simple setting for disabling automatic linking of CamelCase words
- Extended support for automatically linked system tiddler titles to include digits and underscore
- Added warning when trying to use plugins designed for TiddlyWiki Classic
- Upgraded to version v8.8.0 of highlight.js for the Highlight Plugin
- Removed the need to manually confirm deleting alerts
- Added optional tooltips to tabs Macro

Hackability Improvements

- Updated to *KaTeX* v0.5.1, with several new features
- Added `publishFilter` to default save template
- Allow **delete** button to be used in the tiddler view-mode toolbar
- Added a new *hidden setting for keyboard shortcuts*
- Update ActionSetFieldWidget to allow timestamp to be preserved
- Added option to suppress navigation for edit/cancel/save
- Enhance SaveTiddlersCommand with do not delete option
- Enhance RenderTiddlerCommand with template option
- Extend tm-open-window to pass variables
- Added new utility method `$tw.utils.tagToCssSelector(tagName)` for converting tag titles into a form suitable for matching the

`tc-tagged-*` classes generated by TiddlyWiki

- Added support for tiddlers containing mp4 videos
- Extended the ActionSetFieldWidget to enable it to be used to delete a field or indexed property
- Made the tw5.com documentation available as a plugin and an edition, simplifying some scenarios for re-using it
- Improved rendering of "More" sidebar tab gradients in the "Snow White" theme
- Added option in **control panel** to enable the display of detailed performance instrumentation in the browser developer console
- Added a LazyLoading template for all non-system tiddlers
- Improved handling of TiddlerLinks in Markdown tiddlers
- Extended FieldManglerWidget to create the target tiddler if it does not exist
- Extended KeyboardWidget to trigger ActionWidgets
- Extended SelectWidget to support multiple selections
- Added new days Operator
- Added official plugin wrappers for the JavaScript libraries async.js and JSZip
- Added new ActionListopsWidget
- Added **rows** attribute to EditTextWidget

Bug Fixes

- Fixed problem introduced in 5.1.9 with processing `tiddlywiki.info` files
- Fixed problem with invisible icons in **$:/AdvancedSearch**
- Improved check for required plugins in ServerCommand
- Fixed problem with queueing change events for deleting non-existent tiddlers
- Ensure EditTextWidget fixes height even if refresh isn't required

- Fixed the gradient on the sidebar tabs divider
- Fixed problem with tiddler titles such as `$:/templates/something` being erroneously classed as `$:/temp` tiddlers, and so excluded from syncing
- Switched to absolute date format in tiddler subtitles

Node.js Improvements

- Fixed problem with the filesystem adaptor saving new image files as base64-encoded .tid files instead of as native binary files with an accompanying `.meta` file
- Extended `tiddlywiki.files` support in TiddlyWikiFolders to allow directories to be loaded recursively, as well as individual files

Contributors

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:

- @afeldspar
- @araname
- @BramChen
- @danielo515
- @Drakor
- @erwanm
- @felixhayashi
- @Evolena
- @hegart-dmishiv

- @idoine
- @inmysocks
- @IreneKnapp
- @gernert
- @kixam
- @Marxsal
- @matabele
- @mklauber
- @nameanyone
- @pmario
- @senevoldsen90
- @Spangenhelm
- @spelufo
- @SuperDOS
- @tgrosinger
- @tobibeer
- @xcazin
- @zahlman

Release 5.1.11

30th January 2016 at 1:41pm

ReleaseNotes

See GitHub for detailed change history of this release

This is a bug-fix release for Release 5.1.10 that fixes a serious issue affecting the operation of the official plugin library.

Release 5.1.12

13th July 2016 at 12:47pm

ReleaseNotes

See GitHub for detailed change history of this release

Major Improvements

New Editor Toolbars

Added support for extensible formatting toolbars for the editor widgets.

- Formatting operations for WikiText, with support for Markdown:
  - Character formatting operations for bold, italic, strikethrough, underline, superscript, subscript, inline monospace
  - Line formatting operations for monospaced blocks, quoting, headings and both numbered and bulleted lists
  - Insert picture from a dropdown chooser
  - Stamp snippet of pre-programmed text
  - Excise selected text into a new tiddler
  - Optional automatic resizing to fit content

- Full keyboard shortcut editor in control panel. Shortcuts are also shown in the toolbar button tooltips

Improved Bitmap Editor

The bitmap editor has been enhanced with a toolbar supporting:

- Painting with selectable colour, width and opacity
- Clearing the image
- Resizing the image
- Keeping track of recently chosen colours

Improved Plugins

Several of the official plugins available in the plugin library have been updated for this release. Upgrading will automatically update any installed plugins.

- The *KaTeX* plugin has been updated to version v0.60.0
- The *CodeMirror* plugin has been updated to version 5.13.2, and integrated with the new editor toolbars. The default configuration has been updated to include syntax highlighting for HTML, JavaScript, CSS, XML, TiddlyWiki Classic and Markdown.
- The Markdown plugin includes formatting toolbar buttons and a new "new markdown tiddler" button in the Tools tab of the sidebar.
- The experimental Evernote plugin allows notes and images from `.enex` files to be imported into TiddlyWiki.
- The new Internals plugin provides features to help understand the internal operation of TiddlyWiki, including new preview modes in the editor showing both the parse and widget trees of the current tiddler
- Improved language translations:

    - Korean
    - Chinese (Simplified) and Chinese (Traditional)
    - German
    - French
    - Danish
    - Dutch

Text-Slicer Plugin Improvements

- Added support for sticky notes within documents
- Extended support for slicing documents to Node.js
- Improved document metadata display
- Added `list-children` filter operator
- Fixed bugs

Other Improvements

Usability Improvements

- Added a setting for disabling linking to missing tiddlers
- Added a setting for forcing word wrapping in monospaced code blocks
- Added a warning when creating or editing tiddler titles containing troublesome characters

- Addition of “new image” button to the “Tools” tab of the sidebar
- Introduction of new general purpose colour-picker Macro
- Introduction of new general purpose image-picker Macro
- Added a bulk delete button to **advanced search** "Filter" tab

Hackability Improvements

- Introduction of the new WikifyWidget
- Introduction of new *EditShortcutWidget* for direct entry of keyboard shortcut codes
- Extended editor preview pane to allow plugins to add further types of preview (see the new `tiddlywiki/internals` plugin for an example)
- Added a simple HTTP PUT saver for use with *WebDav* or REST servers
- Added build targets to the server edition
- Improved days Operator to better cope with operations on today and yesterday
- Extended ActionWidgets mechanism to allow them to be used with the LinkCatcherWidget and SelectWidget
- Added `mobile-web-app-capable` meta tag for Android Chrome
- Added new `plainwikified` format for the ViewWidget

Bug Fixes

- Fixed problem with tiddler titles containing single quotes in Table-of-Contents Macros
- Fixed problem with z-ordering of modals in zoomin story view
- Fixed problem with has Operator

- Fixed problem with double byte entities in the EntityWidget
- Fixed problem with ScrollableWidget in static renderings
- Fixed problem with repeatedly lazy loading of tiddlers without a text field
- Fixed problem with titles containing URLs being interpreted as external links

Node.js Improvements

- Added ability to customise the paths used for tiddler saving
- Fixed problem saving non-JPG binary files

Contributors

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:

- @araname
- @BramChen
- @brentmaxwell
- @Drakor
- @edrex
- @Evolena
- @felixhayashi
- @FND
- @Infurnoape

- @kixam
- @Marxsal
- @maxthomax
- @mklauber
- @nameanyone
- @nome
- @PauDeLuca
- @pmario
- @rcrath
- @r1chard5mith
- @sukima
- @telmiger
- @twMat
- @webninjasi
- @xcazin

Release 5.1.13

25th July 2016 at 10:48am

ReleaseNotes

See GitHub for detailed change history of this release

This is a minor bug fix release.

Bug Fixes

- Fixed (and refixed) problematic stricter external link matching rules introduced in 5.1.12
- Fixed problem with tiddlers titled "undefined"
- Fixed rendering problem with KaTeX Plugin
- Removed unneeded border in print stylesheet

Node.js Bug Fixes

- Fixed problem with extraneous `.tid` extension added to tiddler filenames

Other Improvements

- Updated Portuguese (Portugal) translation

Contributors

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:

- @sukima

Release 5.1.14

26th April 2017 at 6:00pm

ReleaseNotes

See GitHub for detailed change history of this release

New and Improved Plugins

- Added SaveTrail Plugin that continuously downloads edited tiddlers in the background, providing a trail of backups
- Added XLSX Utilities plugin for importing Excel-compatible spreadsheets. See the XLSX Utilities Edition for details
- Added QR code generator plugin
- Added BibTeX importer plugin
- Updated Google Analytics plugin to the latest version
- Added Twitter Plugin for embedding tweets etc
- Updated KaTeX Plugin to v0.7.1 of KaTeX

Improved Support for Drag and Drop

There are several improvements to Drag and Drop:

- Ability to rearrange the "Open" sidebar panel and the tag dropdown (and new tiddlers can be opened by dragging them to the list)
- Ability to reorder entries in tag dropdowns
- Dragging a tag pill is now equivalent to dragging all of the tagged tiddlers together
- Entries in the control panel "Appearance"/"Toolbars" tab can be reordered by drag and drop
- The image shown while dragging now reflects the dragged item, instead of being a generic drag pill

New "Heavier" Theme

A new theme called "Heavier" thickens TiddlyWiki's default lightweight fonts. A variant "Tight-Heavier" combines the "Heavier" theme with the existing "Tight" theme:

Try out the "Heavier" theme | Try out the "Tight-Heavier" theme | Revert

New Tiddler Manager

Added new **tiddler manager** to support tasks such as:

- Reviewing and applying tags to a large number of tiddlers
- Tweaking the custom colours and icons for a large number of tiddlers

The tiddler manager will be extended in the future to increase flexibility and to add support for bulk operations.

Performance Improvements

- Extended search mechanism to require the search string to be a minimum length

Translation Improvements

- Improvements to Danish, Dutch, French, German, Portuguese and Chinese translations
- Added new Hebrew translation
- Added preliminary support for right-to-left (RTL) languages

Usability Improvements

- Added a select all/none checkbox to the header of the import panel
- Added "close plugin library" button
- Added support for adding tags using the `enter` key
- Improved handling of `ctrl-Enter` to add partially completed tags and fields
- Added support for relinking references to renamed tiddlers
- Added new "print page" page control button (and added associated **icon**)
- Added new page control button to temporarily suspend timestamps
- Improved access to plugin information:
  - Refactored the display of plugin tiddlers to use the same format as the control panel, adding the shadow tiddler listing as an overridable extension tab
  - Added a new tab in the “More” sidebar providing quick access to all installed plugins
- Fixed height of preview pane to use a scrollbar when fixed height layout is selected
- Extended editor "link" button to create external links and missing links
- Increased size of icons in the "Filter" tab of **$:/AdvancedSearch**
- Fixed problem with multiple copies of a tiddler appearing in the story river
- Improved dropdown for new field names with typeahead search
- Improved the ordering of groups in the tiddler editor content type dropdown
- Added "sticky" mode for tiddler info panel. See "Settings" in **control panel**
- Increased boldness of tag pill text
- Added documentation marker to indicate the version in which a new feature was released

- Added an entry to the advanced search filter drop down for exporting the current story (minus advanced search)

New and Improved Filter Operators

- Added new enlist Operator
- Added several new filter operators for string encoding/decoding:
  - decodehtml Operator
  - decodeuri Operator
  - decodeuricomponent Operator
  - encodehtml Operator
  - encodeuri Operator
  - encodeuricomponent Operator
  - escaperegexp Operator
  - stringify Operator
- Added new minlength Operator
- Added new order Operator
- Added strict mode to tag Operator
- Added new **tags** option to the all Operator
- Changed is Operator to pass through arguments if the operand is blank
- Added "field" suffix for has Operator
- Improved *wikiparserrules Operator* to return all available parse rules

Hackability Improvements

- Added support for "formattedtext" output type to the WikifyWidget
- Added a saver for the peer-to-peer Beaker browser; see *TiddlyWiki on Beaker Browser* for details
- Refactored the tag templates and macros to make them more reusable
- Improved modals and notifications so that global macros are available
- Extended the SetWidget to allow a single result to be selected from a filtered result list
- Added new `$:/info/url/*` information tiddlers providing document location information
- Added support for the **actions** attribute to the CheckboxWidget
- Extended CheckboxWidget to support data tiddlers via the `index` attribute
- Added new entry to the advanced search filtered export menu for exporting the current story river
- Added `tag` attribute to the KeyboardWidget
- Added overwrite protection to the WebDAV saver
- Added new "big block quote" style for Block Quotes in WikiText
- Added customisable template text for new journal tiddlers, accessible the control panel
- Added support for `.json` tiddler files containing a single tiddler object, instead of an array of tiddlers

Node.js Improvements

- Fixed multiple issues the creation of new tiddler files by the filesystem adaptor
- Added new FetchCommand for retrieving tiddlers over http/https
- Enhanced RenderTiddlerCommand to add support for an additional variable

Bug Fixes

- Improved Node.js support for the *HighlightPlugin*
- Fixed problem with recursion errors in the Table-of-Contents Macros
- Prevented email addresses from being mis-recognised as HTML tags
- Fixed issue with `$tw.utils.strEndsWith()`
- Partially fixed problem with action widgets being invoked independently of refresh cycle
- Revised handling of `syncAdaptor.isReady()` method to improve compatibility with older adaptors
- Fixed problem with unsafe use of `String.prototype.replace()`, corrupting fields containing dollar signs
- Fixed problem with page background colour not showing through transparent background images
- Fixed sync problem with skinny tiddlers not being loaded
- Fixed problem with "Cancel" button in login dialogue also submitting the form
- Fixed typo affecting created/creator fields when deleting a field
- Fixed issue with system links containing international characters not being recognised
- Fixed issue with single line macro definitions without a terminating line break
- Fixed issue with empty strings and the CheckboxWidget
- Fixed problem with calculation of week numbers
- Fixed problem with move Operator not wrapping correctly
- Fixed problem preventing customisation of textarea background colours

Developer Bug Fixes and Improvements

- Updated Stanford JavaScript Library to version v1.0.6
- Updated Google Analytics plugin to latest version of Google's code
- Improved Common/JS compatibility by adding support for implicitly adding `/index.js` to a require target

- Improved support for bulk loading tiddler files via tiddlywiki.files Files within TiddlyWikiFolders
- Refactored server mechanism to make it simpler to integrate TiddlyWiki into ExpressJS apps
- Refactored the TiddlyFox saver module to make it easier to use it with other saving mechanisms
- Added (and here) several new UI hooks, and improved the hook mechanism; see developer docs

Contributors

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:

- @beniquezsd
- @BramChen
- @crypdick
- @danielo515
- @dedioste
- @DoronTzur
- @duarteframos
- @FND
- @gernert
- @hchaase
- @ibnishak
- @jest
- @jrgetsin

- @nameanyone
- @Marxsal
- @mklauber
- @nmota
- @pmario
- @roma0104
- @Serj-Aleks
- @stevesunypoly
- @sukima
- @sycom
- @tejjyid
- @telmiger
- @tobibeer
- @twMat
- @xcazin
- @zakrec

Release 5.1.15

13th November 2017 at 5:11pm

ReleaseNotes

See GitHub for detailed change history of this release

New and Improved Plugins

- New Amazon Web Services Plugin with primitives for running TiddlyWiki in the cloud
- New External Attachments Plugin that allows automatic creation of ExternalImages for TiddlyDesktop

Translation Improvements

New and improved translations:

- New Persian translation
- Updated Chinese and Greek translations

Another improvement that benefits users of translations is the addition of about 800 new *transliteration pairs*. These are substitutions that are performed when converting a tiddler title into a filename, such as the two characters "oe" for the single character "œ". See the commit for more details. The result is that automatically generated filenames are much more readable.

Usability Improvements

- Added new "Explorer" tab to "More" sidebar and underlying tree Macro
- Added configuration option in **theme tweaks** for the editor font

Hackability Improvements

- Added new lookup Operator to simplify use of lookup tables
- Extended the SetWidget with **tiddler**, **field** and **index** attributes

- Added new **whitespace** Pragma allowing whitespace to be selectively ignored within wikitext
- Updated support for saving in *Beaker Browser*
- Extended date formatting to support milliseconds and UTC dates
- Extended LinkCatcherWidget to make the target tiddler available within the action string
- Extended (and here) the DropzoneWidget and BrowseWidget to allow the deserializer to be manually specified
- Added new `tc-btn-*` classes to toolbar buttons allowing easier custom styling
- Added new **mode** attribute to the ViewWidget, allowing control over the parsing mode for wikified output formats
- Added new jsonstringify Operator

Node.js Improvements

- Added SaveCommand and RenderCommand, intended as more flexible replacements for RenderTiddlerCommand, RenderTiddlersCommand, SaveTiddlerCommand and SaveTiddlersCommand
- Added (and here) more colour to command line output
- Added new "raw" options to the FetchCommand
- Added new ImportCommand

Bug Fixes

- Fixed problem with "new journal" button clearing the text field
- Removed extraneous URI encoding of filename with PUT saver
- Restored ability to use sub-filter with image-picker macro
- Fixed problem with the download saver and filenames containing spaces, accents or other characters that need URL encoding

- Fixed problem with importing JSON tiddlers with leading or trailing spaces in the title
- Fixed plugin display so that "contents" tab is shown even if no other tabs are provided
- Fixed problem with `allbefore:include` filter operator
- Updated Update Stanford JavaScript Library to version 1.0.7

Developer Bug Fixes and Improvements

- Extended `$tw.boot.boot()` to add a callback parameter
- Extended navigation handling to pass original event data
- Added new `th-importing-files` hook

Contributors

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:

- @0mp
- @Arlen22
- @BramChen
- @cjhunt
- @danielo515
- @hoelzro
- @Marxsal
- @maxthomax

- @morosanuae
- @pmario
- @rubaboo
- @saqimtiaz
- @telmiger
- @twMat
- @xcazin

Release 5.1.16

25th April 2018 at 5:56pm

ReleaseNotes

See GitHub for detailed change history of this release

Improved CodeMirror Plugins

The CodeMirror plugin has undergone a substantial rewrite, with many new improvements:

- Upgrade to CodeMirror version v5.37.0
- Now modularised into:
  - Base plugin: basic lightweight configuration, approximately 240KB
  - Add-on plugins:
    - Search and Replace

        - Auto-completion
        - Fullscreen editing
        - Auto-close brackets
        - Auto-close tags
        - SublimeText keymap
        - Emacs keymap
        - Vim keymap
        - Highlighting modes: Javascript, Xml, Html (htmlmixed), Html (htmlembedded), Css, Markdown, X-Tiddlywiki
- Automatic configuration (the configuration tiddler is not needed anymore)
    - CodeMirror (engine.js) builds the configuration using information within the installed CodeMirror plugins
- New control panel configuration tab:
    - Line numbers
    - Theme:
        - Editor font family (same as native TiddlyWiki)
        - Cursor settings
        - Line wrapping
    - Active line highlighting
        - Autocompletion on/off (if installed)
    - Keymap

Other New and Improved Plugins

- New Dynaview Plugin makes it possible to build user interfaces that dynamically respond to changes in scrolling or zooming

- Improved Text-Slicer Plugin with customisable JSON rules
- Updated KaTeX Plugin to latest v0.9.0

Translation Improvements

New and improved translations:

- New Slovenian translation
- New Português (Brasil) translation
- Improved Dutch translation
- Improved French translation
- Improved German translation
- Improved Traditional and Simplified Chinese translations
    - These translations now also reset the minimum search length to one character

Privacy Improvements

Prompted by the General Data Protection Regulation (GDPR), we've removed all third party cookies from tiddlywiki.com.

Usability Improvements

- Added new DiffTextWidget, and integrated it into the import listing and as a preview pane
- Added selectable previews to the import listing
- Added new editor toolbar buttons for wrapping the selection with double square and curly brackets

- Added new sortan Operator for sorting alphanumeric values such as "Apple 6", "Apple 7", "Apple 8", "Apple 9", "Apple 10", "Apple 20", "Apple 30", "Apple 100"
- Improved default fonts for readability as advised by GitHub
- Added an editor toolbar button for stamping fragments of formulae for the KaTeX Plugin
- Simplified the page scrolling behaviour: navigating to a tiddler now scrolls the top of the tiddler to the top of the viewport
- Added WidgetMessage: tm-copy-to-clipboard and also copy-to-clipboard Macro for copying text directly to the clipboard
- Added a hint to the field list dropdown
- Added a control panel setting for the default tab in the More sidebar
- Added a control panel setting for the titles of new tiddlers
- Added "rotate left" button to bitmap editor toolbar
- Improved criteria under which certain editor toolbar buttons are displayed
- Added support for a minimum length for the new tag field before the dropdown is shown
- Improved consistency and reusability of the palette, language and theme switchers
- Added support for retaining tab content in the tabs Macro
- Added support for adding links via the `enter` key

Hackability Improvements

- Added support for comparison operators to the RevealWidget
- Added support for StartupActions to execute action widgets at startup
- Extended ServerCommand to allow the port number to be specified as an environment variable
- Added screen width/height to platform information exposed at startup
- Extended DroppableWidget to pass modifier keys in a variable

- Extended DraggableWidget with actions for drag start and drag end
- Added a new TaskManagementExample (Draggable)
- Added new 'Stylesheets' tab to control panel
- Added new 'emptyMessage' parameter to list-links Macro and list-tagged-draggable Macro
- Added new `th-page-refreshed` hook
- Enhanced macro evaluation by making parameters available as variables - see "Parameters as Variables" in Macro Definitions in WikiText
- Enhanced LinkWidget to allow better customisation of CSS classes
- Added new "value" suffix for the each Operator
- Added new **subtiddler** attribute to the SetWidget
- Added new subtiddlerfields Operator for extracting the fields from tiddlers within plugins
- Added ability to trigger actions on unchecking the CheckboxWidget
- Added support for a checked class to the RadioWidget
- Extended the is Operator to allow multiple types to be specified
- Added data attributes to *ViewTemplate* (see Custom data-styles)
- Added new RangeWidget

Bug Fixes

- Removed superfluous list item at the end of list-links-draggable macro
- Fixed problem with the content area of vertical tabs sometimes overflowing their right hand margin
- Fixed problem with double quotes within the new journal template
- Fixed problem with dragging inside a textarea
- Fixed usage of "default" attribute of SelectWidget when "index" attribute is used

- Fixed problem with crash when dragging over TiddlyWiki on some browsers
- Fixed issue with new journal button
- Fixed problem with **filter** and **value** attributes of SetWidget
- Fixed problem with numeric operands in the Extended Listops Filters
- Fixed problem with "put" saver and 412 return codes
- Fixed problem with variable references in field values in the edit template
- Extended behaviour of list-after field so that the tiddler is placed at the end of the list if the list-after field is present but empty

Developer Bug Fixes and Improvements

- Added **plugin-priority** field to core plugins
- Fixed problems with plugin ordering
- Several optimisations to the tiddler store:
  - Avoiding slow Object.keys calls - saving 50% of some operations
  - Optimising the tag filter - saving 50% on some tag operations
  - Minor improvements to tiddler deletion
- Added support for custom boot paths
- Added new navigation hooks: `th-cancelling-tiddler`, `th-new-tiddler` and `th-renaming-tiddler`
- Added new hook when the HTTP server starts `th-server-command-post-start`

Contributors

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:

- @alexhough
- @anavarre
- @BramChen
- @BurningTreeC
- @diego898
- @flibbles
- @furicle
- @gernert
- @ibnishak
- @inmysocks
- @Marxsal
- @metebalci
- @mklauber
- @pmario
- @Skeeve
- @TheDiveO
- @tobibeer
- @twMat
- @xcazin

Release 5.1.17

12th May 2018 at 12:43pm

ReleaseNotes

See GitHub for detailed change history of this release

Reversions

After encountering significant problems we've reluctantly decided to remove support for multiple sub-operators with the is Operator that was added in v5.1.16.

Usability Improvements

- Improved appearance of TiddlyWiki's "red screen of embarrassment"

Bug Fixes

- Fixed problem with extraneous space in new journal titles
- Fixed problem with repeated use of "new journal" with text set
- Fixed (and here) problem with Etag handling
- Fixed problem with is Operator introduced in 5.1.16
- Restored default of the preview pane being hidden
- Fixed incorrect plugin library location

- Fixed animation timing problem with reveal widget that sometimes caused the sidebar not to show when requested
- Fixed problem with saving with non-Dat URI schemes
- Fixed deletion of `$:/status/UserName` when logged out

Node.js Improvements

- Stopped clearing *$:/status/UserName* when logged out
- Fixed bug with attempting to lazy load tiddlers that do not qualify for syncing
- Added support for `/*\` style metadata comments at the top of `*.CSS` files

Contributors

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:

- @BramChen
- @BurningTreeC
- @danielo515
- @icarito
- @ldorigo
- @mklauber
- @sukima
- @tobibeer
- @twMat

Release 5.1.18

6th December 2018 at 10:00am

ReleaseNotes

See GitHub for detailed change history of this release

Major Improvements

Global Keyboard Shortcuts

This version introduces new infrastructure to handle global keyboard shortcuts. The core ships with the following shortcuts:

- `alt-I` (or `ctrl-I` on Mac) - New image
- `alt-J` (or `ctrl-J` on Mac) - New journal
- `alt-N` (or `ctrl-N` on Mac) - New tiddler

See How to create keyboard shortcuts

Node.js HTTP Server

The HTTP WebServer functionality has been significantly improved:

- Adds ability to make a wiki that is read-only for anonymous users (with the editing UI hidden), but allows editing for users who log in
- Hides UI features like the new tiddler button and the edit button when the user only has read-only access to the wiki
- Adds support for serving static file attachments over HTTP

- Adds support for multiple user credentials for basic authentication via a CSV file containing usernames/passwords
- Adds support for serving over HTTPS as well as HTTP
- Adds support for authentication via a trusted header, making it easier to integrate with corporate single sign on (and Windows authentication)
- Adds support for more fine grained authorisation (ie granting/denying individual users read/write permission to resources)
- Better debugging information
- Experimental support for a new, plain HTML single tiddler per page view
- Adds a simple level of CSRF protection
- Deprecating the existing ServerCommand in favour of a new ListenCommand using the new NamedCommandParameters support for named parameters that can be specified in any order
- Refactoring routes as separate modules, making it easier to extend the server with new functionality

See WebServer for more details, or see the changes on GitHub.

Special Characters in Tiddler Titles

In previous versions of TiddlyWiki, there were some special character sequences in tiddler titles that would break certain features. For example, a tiddler title ending in double quotes would cause the Table-of-Contents Macros to render incorrectly.

As of 5.1.18, many of these problems have been fixed by refactoring the features to use the latest abilities of wikitext:

- Table-of-Contents Macros
- tree Macro
- tag Macro
- list-tagged-draggable Macro

- **$:/TagManager**

Note that using certain characters in tiddler titles will still prevent some features from being used correctly. For example, two vertical bar characters are used in the double curly braces transclusion syntax to separate the tiddler title from the template title: `{{title||template}}`. However, although it would be impossible to transclude a tiddler called `title||template` using that syntax, one could still transclude it using `<$transclude tiddler="title||template"/>`.

Plugin Improvements

- *KaTeX* **Plugin** – the KaTeX Plugin has been updated to KaTeX v0.10.0. The plugin now incorporates an extension for chemical formulae
- **Comment Plugin** – this release adds a new commenting plugin – see *$:/plugins/tiddlywiki/comments*
- **AWS Plugin** – support for running TiddlyWiki as a Lambda function has been improved with support for GZip compression of the JSON payload

Translation Improvements

New and improved translations:

- Catalan
- Chinese Simplified and Chinese Traditional
- Dutch
- French
- German

Usability Improvements

- Added support for splash screens to be shown while TiddlyWiki is loading – see Creating a splash screen
- Added display of backlinks when renaming a tiddler
- Added persistent indicators at the bottom of the window showing any hidden drafts
- Added current username (if set) to the title of draft tiddlers, making them easier to identify in multiuser scenarios
- Added a warning for binary tiddlers in view mode
- Extended permalink/permaview to copy the URL to the clipboard as well as updating the address bar. This behaviour can be customised in **$:/ControlPanel** settings
- Improved saving to use the $:/SiteTitle as the basis of the filename
- Added support for webp, heic and heif image formats
- Fixed tiddler positioning when creating new tiddlers
- Fixed smooth scrolling in new windows
- Fixed problem dismissing popups within new windows
- Fixed handling of disabled missing links so that core user interface features are not broken (see also here)

Hackability Improvements

- Added a new filter prefix `~` allowing if-then-else logic within filters – see Filter Expression
- Added a new subfilter Operator to allow nested filters to be used
- Added new **stateTitle**, **stateField** and **stateIndex** attributes to the ButtonWidget and the RevealWidget, making it possible to use state tiddlers whose titles contain `!!` or `##`
- Enhanced the search Operator to support searching arbitrary lists of fields, literal searches, whitespace-tolerant searches and regexp

searches

- Added new contains Operator for searching within list fields
- Added support for a navigation scroll offset to avoid navigating to tiddlers resulting in them being obscured under a `position:fixed` top menu bar – see Hidden Setting: Scroll Top Adjustment
- Extended WidgetMessage: tm-full-screen message to support forcing the status instead of just toggling it
- Refactored the sidebar to make it be dynamically built from the tag $:/tags/SideBarSegment
- Refactored the tiddler info panel to make it be dynamically built from the tag $:/tags/TiddlerInfoSegment
- Added new range Operator for generating ranges of numbers
- Added palette support for `<select>` element colouring
- Added new text editor operation **save-selection** – see WidgetMessage: tm-edit-text-operation
- Added support for editor toolbar buttons to have action widgets
- Added support for rebasing headings at render time – see tv-adjust-heading-level Variable
- Added experimental support for packaging TiddlyWiki's JavaScript core into an external script for use with the new Web Server enhancements
- Added new `\import` Pragma as a more concise alternative to using the ImportVariablesWidget directly
- Added (and here) new tv-show-missing-links Variable for tighter control over rendering of missing links by the LinkWidget

Bug Fixes

- Fixed with WidgetMessage: tm-new-tiddler overwriting creation fields of existing tiddlers
- Fixed problem with unwanted variable substitutions by restricting variable substitutions to macros defined with the **define** pragma
- Fixed problem with the 'excise' editor toolbar button not showing if the tiddler type is "text/vnd.tiddlywiki"
- Fixed problem with generation of wifi QR codes

- Fixed *$tw.platform.isLinux* on the latest Firefox
- Fixed problem with using full screen on secondary windows
- Fixed subtle bug whereby the RevealWidget was not refreshing properly when the state tiddler of a popup changed
- Fixed problem with malformed HTML entities
- Fixed crash when passing illegal arguments to the decodeuri Operator and decodeuricomponent Operator
- Fixed errors when a popup blocker blocks opening a new window
- Fixed problems using a lazily loaded tidler as a state tiddler with the RevealWidget
- Fixed problem with modals not appearing in the correct window
- Fixed the animations for adding and removing tags in edit mode, and a similar fix for the animations in the "Open" sidebar tab

Developer Bug Fixes and Improvements

- Refactored `$tw.utils.parseStringArray()` to optionally allow non-unique entries

Contributors

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:

- @abesamma
- @AnthonyMuscio
- @Arlen22
- @bimlas
- @BramChen

- @BurningTreeC
- @Drakor
- @EvanBalster
- @flibbles
- @G0erman
- @hchaase
- @hoelzro
- @iblech
- @jho1965us
- @ldorigo
- @Lioric
- @Marxsal
- @morosanuae
- @pmario
- @talha131
- @TechLifeWeb
- @tobibeer
- @twMat
- @yrashk
- @xcazin
- @xcodinas

Release 5.1.19

20th December 2018 at 5:34pm

ReleaseNotes

See GitHub for detailed change history of this release

This release consists of fixes for bugs discovered in Release 5.1.18.

Hackability Improvements

- Added new hidden setting for the filename used when saving

Bug Fixes

- Fixed incompatible change to the construction of filename used for saving
- Fixed problem with fonts for the KaTeX Plugin
- Fixed problem with forcing basic authentication for the WebServer
- Fixed problem with the ButtonWidget not refreshing when the `popupTitle` attribute changes
- Fixed problem with `empty.html` accidentally containing a splash screen
- Fixed regression with search matches across multiple fields

Contributors

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:

- @BurningTreeC
- @dlund4
- @hoelzro

Release 5.1.2

27th September 2014 at 6:24pm

ReleaseNotes

See GitHub for detailed change history of this release

This is another small release with documentation updates and a few bug fixes and improvements.

Usability Improvements

- Improved error handling for malformed or unrecognised LaTeX content with KaTeX Plugin

Hackability Improvements

- Improved styling of separator before the untagged item in the tags sidebar
- Improved error handling for the upload saver (which is used for saving to *TiddlySpot*)
- Improved storyview error detection

Bug Fixes

- Fixed crash caused by incorrectly formatted plugins
- Fixed problem with tiddler info not being removed by the syncer after deleting a tiddler
- Fixed HTTP handling to treat response code 201 as success

Contributors

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:

- @Evolena
- @pmario
- @simonbaird
- @TheDiveO

Release 5.1.20

9th August 2019 at 4:13pm

ReleaseNotes

See GitHub for detailed change history of this release

New Conditional Operators

Added new Conditional Operators then Operator and else Operator

New Mathematics Operators

Added several new Mathematics Operators for working with numbers: negate, abs, ceil, floor, round, trunc, untrunc, sign, add, subtract, multiply, divide, remainder, max, min, fixed, precision, exponential, sum, product, maxall and minall.

New String Operators

Added several new string operators: match, length, uppercase, lowercase, titlecase, sentencecase, trim, split, splitregexp and join.

Improved Handling of Duplicates in Filters

There is now finer control over TiddlyWiki's default behaviour of removing duplicates from filter results, making it much easier to work with the new mathematics and string operators:

- Added support for new `=` prefix for merging filter runs without removing duplicates
- Added option for the enlist Operator to not remove duplicates

Improved Keyboard Shortcuts

Keyboard shortcut handling has been significantly improved with the following new global shortcuts (configurable via **$:/ControlPanel**):

- `ctrl-shift-F` - focus the sidebar search field
- `alt-shift-S` - toggle the visibility of the sidebar
- `ctrl-shift-A` - open the advanced search tiddler

Saving to GitHub and GitLab

Added a Git service saver for saving directly to GitHub and GitLab

Plugin Improvements

New and improved plugins:

- New BrowserStorage Plugin for saving tiddlers in browser local storage
- New Innerwiki Plugin for rendering TiddlyWiki screenshots
- Updated KaTeX Plugin to KaTeX v0.10.2
- Added an editor toolbar button for the KaTeX Plugin
- Updated Highlight Plugin to use highlight.js v9.15.6
- Added support for language aliases with the Highlight Plugin
- Updates to the CodeMirror Plugin, including adding support for smart indent
- Updates to the Dynaview Plugin, including adding support for restoring the scroll position at startup from local storage
- Updates to the Amazon Web Services Plugin, including adding support for specifying credentials

Translation Improvements

New and improved translations:

- Catalan
- Chinese Simplified and Chinese Traditional

- French
- German
- US English

Performance Improvements

- Added pluggable index modules to accelerate the field Operator and the tag Operator (in tests on a wiki with 60K tiddlers, startup time is reduced by 25% with these optimisations, and refresh time is reduced by a factor of three.)
- Optimised the "Classic" storyview to perform much faster when the animation duration is set to zero (approximtely 50% speed improvement was observed in tests opening a storyview with 8,000 entries)
- Optimised the RevealWidget to avoid using the relatively slow `localeCompare()` method to compare strings
- Updated Performance Instrumentation to show the timings for individual filters
- Optimised rendering and refreshing child widgets

Usability Improvements

- Improved the **palette manager** with the ability to delete entries, and directly edit indirect entries
- Several new palettes: "Nord", "Solarized Light", "Spartan Day", "Spartan Night" and "Twilight"
- Fixed the plugin library to search all plugin fields
- Added a warning when core JavaScript modules are overwritten via import
- Fixed prompt for references when renaming a tiddler
- Added a new filter preset for advanced search for "tiddlers modified this session"
- Added a number of animations (some of were already present but had been broken over the years) for changes in:

- Tags in the view template
- Page controls
- Top left and top right bar
- Tabs
- Fields in the edit template

Hackability Improvements

- Added support for configurable view and edit templates
- Improved the dumpvariables Macro
- Added new variables Operator and getvariable Operator
- Added new WidgetMessage: tm-focus-selector
- Added new escapecss Operator
- Added new unusedtitle Macro
- Added new `[is[variable]]` option to the is Operator
- Added new `[is[blank]]` option to the is Operator
- Added optional autofocus for the text inputs in **$:/AdvancedSearch** (controlled by the same Hidden Setting: Search AutoFocus as the sidebar search)
- Added optional **tabindex** attribute to simple, framed and *CodeMirror* text editor engines
- Added **tabindex** attributes to edit templates
- Added outline colour to button, textarea, input
- Added **tooltip** attribute to SelectWidget
- Added an option for plugins to disable the initial navigation at startup

- Added **$:/info/browser/language** to the InfoMechanism
- Added an example of using StartupActions to set the default language
- Improved importing of JSON files to verify whether they are valid tiddler files, and import them as plain JSON if not
- Improved the LinkWidget to use the title of the target tiddler as a default if it has no content
- Improved WidgetMessage: tm-open-window to allow a window title to be specified
- Added **checkActions** attribute to the CheckboxWidget
- Added support for searches anchored to the start of a field

Bug Fixes

- Fixed a long standing problem with the main standalone HTML file template whereby raw markup tiddlers would be included even if not selected by the save filter
- Fixed overlapping dropdowns with sticky titles enabled
- Removed apparently obsolete limitation of selection wrapping in text editor
- Fixed bug in RevealWidget when stateTitle tiddler is missing
- Fixed bug with animation duration being blank
- Fixed Firefox problem with drag and drop placeholders in the "open" tab of the sidebar
- Ameliorated Firefox problem with sticky placeholder during drag and drop operations
- Fixed CodeMirror Plugin to use **tiddler-editor-border** colour
- Fixed problem using the *TiddlySpot* saver with cookies disabled
- Fixed framed text editor to use correct background colour
- Fixed problem with the class `tc-sidebar-lists` being used on two different areas of the sidebar, making it impossible to distinguish them in CSS selectors

- Fixed problem with unnecessary "list" field created when renaming tiddlers
- Fixed crash with EditBitmapWidget and missing tiddlers

Node.js Bug Fixes and Improvements

- Added new SaveWikiFolderCommand to save a wiki as a wiki folder
- Added new DeleteTiddlersCommand
- Added (and also here) ability to specify plugins on the command line (see Using TiddlyWiki on Node.js)
- Added support for deflate and gzip compression to the WebServer ListenCommand
- Added an option to the LoadCommand to ignore loading errors
- Improved handling of JSON files by the file system adaptor:
  - Use `.json` files (instead of `.tid`) for any tiddler whose fields contain values that can't be stored as a .tid file
  - Save `application/json` tiddlers as `.json` files

Developer Bug Fixes and Improvements

- Added support for widget subclassing – see https://tiddlywiki.com/prerelease/dev/#WidgetSubclassingMechanism

Contributors

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:

- @00SS

- @admls
- @benwebber
- @bimlas
- @BramChen
- @BurningTreeC
- @diego898
- @dnebauer
- @inmysocks
- @flibbles
- @jdjdjdjdjdjd
- @JesseWeinstein
- @joshuafontany
- @Kartiku
- @kyrias
- @MarxSal
- @MidnightLightning
- @pacoriviere
- @pmario
- @rmunn
- @talha131
- @telmiger
- @the-kenny

- @xcazin

Release 5.1.21

10th September 2019 at 5:23pm

ReleaseNotes

See GitHub for detailed change history of this release



Congratulations to Sylvain Comte for his winning design for the banner for this release (here are the other entries).

This is a bug fix release that resolves issues introduced in the recent Release 5.1.20.

Bug Fixes

- Fixed crash with ActionDeleteFieldWidget and a missing **tiddler** attribute
- Fixed edge cases where some tiddler lists were not valid when interpreted as a filter
- Fixed problem with too many alerts when upgrading a wiki containing an overwritten core module
- Reverted inadvertently switching on performance instrumentation
- Fixed bug with WikiText within field names
- Fixed problem with classes in the Table-of-Contents Macros
- Fixed missing foreground colour for buttons with class `tc-btn-invisible`

- Fixed whitespace around field editor inputs
- Fixed layout problems with the translators edition
- Fixed crash with the External Attachments Plugin on Windows
- Removed inconsistent shadow for tag pills in the sidebar
- Fixed regression with the field Operator and blank operands
- Reverted erroneous change in v5.1.20 in the way that lists are stringified
- Fixed the **Open** sidebar tab to use the `tv-story-list` variable
- Fixed problem saving to GitLab

Node.js Bug Fixes

- Fixed bug with customised tiddler file naming

Contributors

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:

- @bimlas
- @BurningTreeC
- @flibbles
- @heronils
- @hoelzro
- @markstegeman

- @StefanSTFG
- @xcazin

Release 5.1.22

15th April 2020 at 6:08pm

ReleaseNotes

See GitHub for detailed change history of this release



Congratulations to Thomas Elmiger for his winning design for the banner for this release (here are the other entries).

Major Improvements

Dynamic Plugin Loading

Plugins can now be loaded or deleted dynamically, without requiring a reload – as long as they don't contain any JavaScript modules. Plugins that require a reload are indicated in the plugin chooser in **$:/ControlPanel**.

The **plugin listing** and the **plugin chooser** in the control panel feature several improvements:

- A new **"Updates" tab** listing all upgrades available from a plugin library, and a button to update all of them with one click

- Tabs are now used to display plugin libraries within the plugin chooser
- The plugin chooser now displays plugins that provide add-on functionality to another plugin as nested **"sub-plugins"**, making the main listing easier to navigate
- Plugins may now declare dependencies that are **automatically installed** without user intervention
- All plugins are now displayed more consistently, including a concise, informal name (in addition to the description and title)

Client-Server Improvements

This release sees several improvements to the client-server configuration:

- Improved resilience to connection errors. Errors encountered by the browser are automatically retried, and when they succeed the associated error messages are automatically removed
- Changes to system tiddlers are now synchronised from the server to the browser
- Deletions of tiddlers on the server are now synchronised to the browser
- New "cloud" page control button with a dropdown menu showing status and additional actions:
  - "Refresh from server" to force an immediate check for changes from the server
  - "Logout"
  - "Save snapshot for offline use" to save a copy of the wiki without the client-server components
  - "Copy syncer logs to clipboard" to make it easier to ask end users for debugging information
- Added "filter" and "exclude" parameters to the GET /recipes/default/tiddlers.json endpoint to allow the returned tiddlers and fields to be filtered
- Enhanced LazyLoading to include skinny versions of tiddlers in the HTML file, and to avoid unnecessary loads from the server
- The official plugin library is now disabled when the tiddlyweb adaptor is active

- Internal file system details are no longer exposed to browsers when a file request fails (see https://github.com/Jermolene/TiddlyWiki5/issues/3724)

See the pull request on GitHub for more details.

New compare Operator

Added a new filter operator for comparing numbers, strings, dates and version numbers. For example:

```
[[2]compare:number:eq[3]]
```

returns nothing

```
[[2]compare:number:lt[3]]
```

returns "2"

```
[[2]compare:number:eq[2]]
```

returns "2"

```
=1.2 =1.8 =2.3 =3.4 =7.2 +[compare:number:gt[1.5]compare:number:lt[2.5]]
```

returns "1.8" "2.3"

```
[[1]compare:number:gt[2]then[yes]else[no]]
```

returns "no"

New Plugins

- Added new **menubar** plugin that creates a menu bar that is responsive on mobile devices, and supports search, dropdowns and links
- Added new **freelinks** plugin to automatically create links from any tiddler title, without having to type double square brackets or use

CamelCase.

- Added new **dynannotate** plugin containing primitives for overlaying annotations or highlights over other content
- Added new **share** plugin for sharing tiddlers via the URL location hash
- Switched the **markdown** plugin to the newer and improved remarkable library

Translation Improvements

- Improved Chinese translation
- Improved Dutch translation
- Improved French translation
- Improved German translations
- Improved Portuguese translation

Performance Improvements

- Optimised ImportVariablesWidget to improve performance of global macros
- Optimised core SVG images to save approximately 100KB from the core plugin
- Added indexer for backlinks
- Optimised sort Operator

Usability Improvements

- Added download button for binary tiddlers
- Fixed (and here) spacing for page control buttons
- Improved display of icons in documentation
- Added support in control panel for specifying the field that should receive focus when editing or creating a tiddler
- Added support in control panel for specifying the tags for new tiddlers and journals
- Added language attribute to HTML document to improve accessibility
- Fixed scaling of embedded videos
- Added "none" option in icon dropdown in **$:/TagManager**
- Fixed a number of layout issues relating to the edit template and tag picker (see 18151cc1, 793d84bc, 1cf2d079, 3c365a25, 45def4de, 89728d8a, 4cf96e73, bfa3ddd0 and 74172b35)
- Added new **Gruvbox Dark** palette
- Improved (and *https://github.com/Jermolene/TiddlyWiki5/pull/4517*)) Vanilla and Nord palettes
- Fixed Railroad plugin to use standard palette colours
- Improved Comment plugin to add a link on each comment to the original post

Hackability Improvements

- Added new saver for https://gitea.io/
- Added new ActionPopupWidget for triggering popups
- Added **tabindex** attribute to the ButtonWidget
- Added **index** attribute to the RangeWidget enabling it to be used with DataTiddlers
- Added support for RefreshThrottling via the `throttle.refresh` field

- Added (and here) new `[is[binary]]` operand for the is Operator
- Fixed usage of `publishFilter` in save templates – see SavingMechanism
- Added CSS class identifying the tiddler body editor
- Added CSS classes to identify sidebar tabsets
- Updated normalize.css from v3.0.0 to v8.0.1
- Updated Highlight plugin to use highlight.js v9.18.1
- Updated Jasmine to v3
- Refactored *AndTidWiki* saver
- Added new SystemTag: $:/tags/Macro/View for creating macros that are only available within the view template
- Added support for embedding `.webm` and `.ogg` video files
- Added support for embedding `.woff2` font files
- Added **spaces** parameter to jsontiddlers Macro for controlling the formatting of the output
- Updated the colour Macro to add support for specifying a fallback for colours not in the current colour palette
- Added new Hidden Setting: Default Tiddler Icon
- Added new Hidden Setting: Disable Drag and Drop
- Added new **tc-tiddler-overridden-shadow** class to tiddler frames for tiddlers that override a shadow tiddler
- Added support for a custom class to modal wrapper
- Fixed datauri Macro to work with tiddlers with a **_canonical_uri** field
- Fixed invaldi HTML produced in ">" in blockquote lists
- Added new "index" option to the has Operator for detecting the presence of data tiddler items
- Extended the FieldsWidget with an "include" attribute and more sorting options, and the fields Operator with new "include" and "exclude" suffixes

- Extended ActionCreateTiddlerWidget with attributes for specifying a template and for controlling overwriting existing tiddlers

Bug Fixes

- Fixed bug with deletes failing when lazy loading used under Node.js
- Fixed problem with the GitLab saver
- Fixed bug with permaview button when placed above the search box
- Fixed usage of **count** parameter of WidgetMessage: tm-edit-text-operation
- Fixed bug with toc-selective-expandable macro still showing disclosure arrow despite excluding tiddlers
- Removed (and here) erroneous word break setting for vertical tabs
- Fixed problem with programmatically deselecting entries from the SelectWidget in multiple selection mode
- Fixed usage of spans in colour picker
- Fixed refreshing of RangeWidget when underlying value changes
- Fixed problem with splash screen being shown when JavaScript is disabled
- Fixed problem with millisecond 0XXX DateFormat
- Improve compatibility of single tiddler window vs. main window
- Fixed problems with some core icons in Internet Explorer
- Fixed GitHub and GitLab savers to use default path of `/` if none specified
- Fixed support for embedding ZIP files in some environments
- Fixed lack of refresh when ButtonWidget **actions** attribute changes
- Fixed the RangeWidget to work in Internet Explorer 10/11
- Fixed CountWidget to display "0" for an empty filter, instead of "undefined"

- Fixed MakeLibraryCommand to respect environment variables
- Fixed not to return "null" for an empty list
- Fixed z-index for *CodeMirror* autocomplete dropdowns
- Fixed unwanted whitespace within sidebar links

Contributors

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:

- @activescott
- @Arlen22
- @BramChen
- @BurningTreeC
- @donmor
- @ento
- @flibbles
- @gernert
- @heronils
- @hoelzro
- @Janno
- @joshuafontany
- @inmysocks

- @kookma
- @linonetwo
- @m42e
- @MidnightLightning
- @mklauber
- @pmario
- @s-l-lee
- @s-light
- @saqimtiaz
- @xcazin

Release 5.1.23

24th December 2020 at 2:29pm

ReleaseNotes

See GitHub for detailed change history of this release



Congratulations to Atronoush for his winning design for the banner for this release (here are the other entries).

Performance Improvements

- **improved** filter execution to use a more efficient linked list structure for intermediate results
- **updated** widget mechanism to cache the result of parsing macros which don't use text substitution
- **updated** page templates to use a single VarsWidget instead of several SetVariableWidgets, for improved performance and easier debugging
- **updated** ListWidget to not initialize parsers for blank `emptyMessage` attributes

Usability Improvements

- **renamed** "references" to "backlinks" in the tiddler info panel
- **added** warning message about using the online plugin library with the client-server configuration
- **updated** favicon for the prerelease edition
- **removed** normalize.css's styling of search input fields and **updated** to a modern fork of normalize.css
- **removed** unneeded editor toolbar buttons when editing SVG tiddlers
- **added** global keyboard shortcut for switching layouts
- **hide** the `.tid` exporter when more than one tiddler to export

Import Mechanism Improvements

- **added** support for renaming tiddlers and a warning about overwriting existing tiddlers
- **added** colour coding for import items that will overwrite existing tiddlers, and for other warnings
- **improved** other warnings in the import listing

- **fixed** content type of imported `.tid` files that do not have a `type` field
- **fixed** extraneous system tiddlers created during import process

Palette Improvements

- **fixed** the **$:/PaletteManager** to show "indirect" colours (ie, colours defined by another `<<colour>>` macro)
- **added** new "Desert Sand" palette
- **added** new "Cupertino Dark" palette
- **added** "Solarized Dark" palette
- **updated** (and **here**) the Vanilla theme to optionally use palette colours for the browser selection outline
- **updated** "Gruvbox" palette readability of toolbar buttons

Keyboard Handling Improvements

- **added** support for navigating the search and new tag dropdowns via the keyboard. The keyboard-driven-input Macro can be used to add this capability to other dropdowns
- **added** a **tabIndex** property to the tiddler editor input area to make it easier to use the `tab` key to move between edit controls
- **added** keyboard support for cycling through the tabs in **$:/AdvancedSearch**
- **added** keyboard support for navigating the field name dropdown in the Edit Template
- **added** keyboard support or navigating the `type` field input in the Edit Template
- **added** keyboard support for using the **insert wikilink** toolbar dropdown in the Edit Template

- **added** a keyboard shortcut for saving the wiki
- **added** a keyboard shortcut for deleting a field in the edit template
- **added** a keyboard shortcut to change the sidebar layout

Widget Improvements

- **added** EventCatcherWidget for low level event handling
- **added** new LogWidget and ActionLogWidget to help debugging
- **added** new ActionConfirmWidget to prompt the user for simple yes/no questions
- **extended** ButtonWidget and DroppableWidget so that changing the class attribute does not trigger a refresh. This makes it easier to use classes to trigger CSS animations
- **extended** the RevealWidget to optionally dynamically refresh popup positions when the state tiddler changes
- **added** support for **disabled** attribute to EditWidget, EditTextWidget, CheckboxWidget, RadioWidget and RangeWidget
- **added** support for **disabled** attribute to ButtonWidget
- **extended** the MacroCallWidget to be able to optionally render the raw text of the macro (previously the output was always wikified)
- **updated** the LinkWidget to work within SVG elements
- **added** **accept** attribute to the BrowseWidget
- **extended** the ActionPopupWidget to allow popups to be dismissed
- **extended** the EditWidget to pass all attributes through to the sub-widget
- **extended** the LinkCatcherWidget to pass keyboard modifier status to actions
- **extended** the RadioWidget to support actions
- **extended** the RangeWidget to support actions

- **added** **autocomplete** attribute to the EditTextWidget and EditWidget
- **fixed** incorrect behaviour of default values with lookup Operator
- **fixed** has Operator when used with the **tags** field
- **fixed** crash when using *edition Operator* in the browser
- **fixed** **class** attribute of DroppableWidget

Filter Improvements

New capabilities:

- **added** support for multiple operands for filter operators

New filter run prefixes:

- **added** support for named filter run prefixes
- **added** new :filter filter run prefix which is analagous to the new filter Operator but applies to a filter run
- **added** new :intersection filter run prefix to get the intersection of two filter runs
- **added** new :reduce filter run prefix

New operators:

- **added** new filter and reduce operators for processing lists of items
- **added** new power Operator and log Operator
- **added** new draft Operator to check if a tiddler is a draft of another tiddler

- **added** new search-replace Operator to search and replace in strings
- **added** new enlist-input Operator to parse its input titles as a title lists
- **added** new slugify Operator and duplicateslugs Operator for generating human readable filenames/URLs
- **added** new sortsub Operator for sorting by a user defined subfilter
- **added** new format Operator for formatting dates
- **added** new reduce Operator to apply a subfilter to each input title in turn, accumulating a single value
- **added** new filter Operator to apply a subfilter to each input title and return the titles that return a non-empty result from the subfilter
- **added** new toggle Operator to toggle a title in a list
- **added** new cycle Operator
- **added** new pad Operator

Other improvements:

- **extended** the trim Operator to optionally trim a given string instead of whitespace, and trim from front, back, or both sides of input tiddlers
- **extended** jsonstringify Operator and stringify Operator to support "rawunicode" suffix

Tiddler File Naming Improvements

extended the mechanism for Customising Tiddler File Naming to:

- Save drafts to a special folder for each user
- Allow filters in *$:/config/FileSystemPaths* to change the path of a tiddler file on disk
- Allow filters in *$:/config/FileSystemExtensions* to change the extension of a tiddler file on disk

- **added** `th-make-tiddler-path` hook for low-level control of tiddler path construction
- **extended** tiddlywiki.files Files specification with `isEditableFile` attribute allowing files to be saved back to their original location

Hackability Improvements

- **added** basic support for switching page templates
- **extended** the WidgetMessage: tm-scroll message to allow the target element to be specified by a CSS selector, making it possible to scroll to positions within a tiddler
- **fixed** modals to incorporate a NavigatorWidget so that links work as expected
- **extended** the Table-of-Contents Macros to support custom link targets
- **added** new Hyperdrive saver for use with Beaker Browser v1.0
- **extended** favicon mechanism to support **_canonical_uri** images
- **added** support for recording the startup timestamp in **$:/info/startup-timestamp** (see InfoMechanism)
- **added** support for SVG favicons
- **extended** `th-saving-tiddler` hook to include information about the draft tiddler (see https://tiddlywiki.com/dev/#Hook%3A%20th-saving-tiddler)
- **extended** WidgetMessage: tm-rename-tiddler to control whether relinking occurs
- **extended** WidgetMessage: tm-import-tiddlers to override the title *$:/Import* and to better control whether the import tiddler is opened automatically
- **extended** the internal `<$element>` widget to add a hook so that plugins can intercept DOM node creation
- **added** original event to `tm-navigate` event
- **extended** the password prompt to enable it to be customised

- **extended** syncer to enable syncadaptors to customise the login prompt
- **extended** the tabs Macro to support `actions` and `explicitState` attributes
- **updated** (and **here**) filters used for syncing on node.js and saving the single file version to exclude multiple story lists and history lists based on their prefix, as well as multiple tiddlers that might be used for the import process
- **added** post-render startup actions
- **added** support for username/password parameters for `tm-login` message
- **added** support for the content type **image/jpg** (the correct type is **image/jpeg** but the misspelling is common so most browsers now support it)
- **added** support for an override saver
- **added** utility CSS classes to replace use of ` ` to introduce visual separation
- **added** option to configure the tag used for TableOfContents in the menubar
- **modified** the KeyboardWidget to not trap keys if there are no actions to be invoked
- **added** buttons to the Edit Template toolbar for the editor-height and the stamp tool for tiddlers of type **application/javascript**,**application/json** and **application/x-tiddler-dictionary**
- **updated** all instance of the tabs Macro in the core to use the explicitState attribute
- **added** support for the `meta` key as a modifier in actions
- **added** support for **$:/info/darkmode** to the InfoMechanism, reflecting the browser dark mode vs. light mode setting
- **updated** `story.js` to remove dependency on `wiki.js` for story start up and navigator

Plugin Improvements

JSZip Plugin

extended the JSZip Plugin with the ability to dynamically create Zip files, and thus to conveniently build multi-file static sites within the browser

Consent Banner Plugin

added the new Consent Banner Plugin plugin helps make websites that are compliant with "cookie legislation" such as the EU General Data Protection Regulation. It presents a banner inviting the user to accept or reject cookies, keeping track of their consent in local storage so that the banner can be hidden on subsequent visits.

By default, content embedded with <iframe>, <embed> and <object> is blocked unless the user consents to accept cookies.

Consent status is available via a configuration tiddler so that it is possible to construct content that behaves differently depending upon whether consent has been granted. As an example, a macro is provided for embedding YouTube videos that automatically uses the youtube-nocookie.com variant of video URLs unless the user has accepted cookies.

Please note that using this plugin does not guarantee compliance with any particular legislation. You will need to understand the technical issues specific to your situation, and if necessary seek legal advice.

Freelinks Plugin

- **added** (and **here**) support for ignoring case when matching titles
- **fixed** bug with autolinking within HTML `<a>` elements

Twitter Plugin

- **added** warning if wiki needs to be saved and reloaded

Dynaview Plugin

- **fixed** examples (and **here**)

CodeMirror Plugin

- **fixed** issue with `e.toggleComment` being undefined with *CodeMirror* Sublime keymap
- **updated** *CodeMirror* plugins to version 5.58.3

Markdown Plugin

- **fixed** encoding of Markdown image files
- **fixed** issue with whitespace and linebreaks
- **added** **tc-tiddlylink-external** class to external links
- **fixed** to add `rel="noopener noreferrer"` to external links
- **updated** to use palette colours

Amazon Web Services Plugin

- **added** a new **aws-encodeuricomponent** filter that also encodes single quotes

BibTeX Plugin

- **updated** to a later fork of the underlying third party BibTeX parsing library

Menubar Plugin

- **updated** so the top margin of the side bar adjusts to the height of the menu

Dynannotate Plugin

- **added** examples of usage in the View Template

External Attachments Plugin

- **fixed** a bug with relative paths in the External Attachments Plugin

Highlight Plugin

- **updated** to use palette colours

Translation Plugins

- **added** new Arabic (Palestine) translation
- **updated** Catalan translation
- **updated** (and **here**) Chinese translation
- **updated** Dutch translation
- **updated** French translation
- **updated** German translation

Other Bug Fixes

- **fixed** parsing of dates from 0 to 100CE
- **fixed** handling of negative dates (see Date Fields and DateFormat)
- **fixed** "409 conflict" errors with the GitHub saver when saving within 60 seconds of the last save
- **fixed** behaviour of system tiddler syncing with the client server configuration. By default, changes to system tiddlers are not synced from the server to the client, restoring the behaviour from v5.1.21 and earlier. Bidirectional syncing of system tiddlers can be enabled with the configuration tiddler **$:/config/SyncSystemTiddlersFromServer**
- **fixed** hover effect for search dropdown items
- **fixed** restored missing parameter to `saveTiddler()` method of syncadaptors
- **fixed** MakeLibraryCommand to skip non-directories
- **fixed** erroneous use of `$tw.wiki`
- **fixed** high CPU usage with animated syncing icon introduced in v5.1.22
- **fixed** **$:/config/NewJournal/Tags** appearing in tag dropdowns
- **fixed** popups not being cancelled when clicking within a text editor
- **fixed** issue with timeline Macro and invalid date values
- **fixed** hover colours of tag dropdowns in the sidebar
- **fixed** alignment of tag pill icons
- **fixed** crash with droppable widget
- **fixed** issue with adding tags in **$:/Manager**
- **fixed** edit template "type" dropdown positioning
- **fixed** edit template field dropdown positioning

- **fixed** (with **addendum**) syntax error in ScrollableWidget
- **fixed** problem with headers not being treated case insensitively
- **fixed** problem with `Content-Type` HTTP header sent as `Content-type`
- **fixed** EntityWidget not refreshing correctly
- **fixed** Markdown Plugin to honour alignment directives
- **fixed** syncing issues with external JS template
- **fixed** incorrect base64 encoding of astral plane Unicode text
- **fixed** server error when saving a new tiddler created by following a tiddler link
- **added** a `plugin-priority` field to the TiddlyWeb plugin so that language plugins can override its language strings
- **fixed** bug whereby joining an empty list would not return an empty list
- **fixed** bug exporting tiddlers with double quoted titles
- **fixed** bug with syncing plugin tiddlers
- **fixed** bug with the position of the tiddler title when there is no icon in use
- **fixed** unwanted autosave in the upgrade wizard
- **fixed** problem whereby `$:/temp` tiddlers were being saved in single file wikis
- **fixed** a bug with the `sortan` filter operator when used with date fields
- **fixed** a bug for location hashes that contain a `#` character
- **fixed** default branch to **main** for saving to GitHub
- **fixed** shadow tiddlers not refreshing when their plugin is deleted or modified
- **fixed** `tc-dirty` class not appearing on external windows
- **fixed** `static.tiddler.html` template to make rendered tiddlers full-width

Contributors

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:


@adithya…

@Arlen22

@bimlas

@BramC…

@Burnin…

@danielo…

@default-…

@ento

@favadi

@fkohrt

@flibbles

@gera2ld

@ibnishak

@idotobi

@jdangerx

@jjduhamel

@joshuaf…

@Kamal-…

@kookma

@Marxsal

@mocsa

@Nicolas…

@Ombra…

@passuf

@pmario

@rmunn

@saqimtiaz

@SmilyOrg

@twMat

@xcazin

Release 5.1.3

22nd February 2015 at 12:01am

ReleaseNotes

See GitHub for detailed change history of this release

Usability Improvements

- Added "new journal" button; see Creating journal tiddlers
- Added "new here" button; see Creating and editing tiddlers
- Added "new journal here" button; see Creating journal tiddlers
- Made date format strings be translateable
- Added Russian translation
- Added confirmation when clearing password
- Added additional prompt when setting password
- Increased size of the clear search button
- Upgraded to KaTeX v0.1.1, with support for several additional *LaTeX* features

Hackability Improvements

- Added audio "parser" for handling Audio content
- Added support for ActionWidgets
- Added new **`addprefix`** and **`addsuffix`** operators
- Added support for custom password prompts
- Extended the **`list-links`** macro to display **caption** field if present
- Extended ImageWidget to allow percentage width and height to be specified

Bug Fixes

- Fixed the **each** operator to work with missing tiddlers
- Fixed problem with tiddler width in zoomin storyview with the sidebar hidden
- Fixed bug whereby the `tm-home` message wasn't navigating to a tiddler, causing problems in zoomin storyview
- Fixed problem with jumping toolbar icons under Firefox
- Fixed problem with untagged label being incorrectly coloured
- Fixed problem with title background colours with the "Sticky Titles" theme
- Fixed problem with subfilter parameter of **timeline** macro
- Exclude search string tiddler from search results

Node.js Changes

These changes are only relevant to people using TiddlyWiki under Node.js

- Extended KaTeX plugin to be able to work under Node.js to generate static HTML
- Extended "includeWikis" to merge build targets
- Refactored the build scripts for tiddlywiki.com into a separate repository at https://github.com/Jermolene/build.jermolene.github.io
- Extended boot kernel to raise an error if a missing wiki folder is specified on the command line

Contributors

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:

- @andrey013
- @BramChen

- @buggyj
- @Eucaly
- @Evolena
- @fghhfg
- @Marxsal
- @pmario
- @simonbaird
- @TheDiveO
- @xcazin

Release 5.1.4

22nd October 2014 at 5:55pm

ReleaseNotes

See GitHub for detailed change history of this release

This is a small release with documentation improvements and some important bug fixes from Release 5.1.3.

Usability Improvements

- Added a toolbar button for advanced search
- Changed dragging behaviour so that you can drag a tiddler title into an edit box to insert the title, making link creation easier. (This change doesn't affect the behaviour of dragging tiddler links between TiddlyWiki windows)

Hackability Improvements

- Added additional theme tweaks for the body text size and line height
- Extended the ViewWidget to work with subtiddlers

Bug Fixes

- Fixed problem with search counts

Node.js Changes

These changes are only relevant to people using TiddlyWiki under Node.js

- Fixed problem with missing wiki folder warnings by removing and re-implementing the fix from Release 5.1.3
- Added coloured warnings and errors under Node.js

Contributors

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:

- @danielo515
- @gernert

Release 5.1.5

21st February 2015 at 11:50pm

ReleaseNotes

See GitHub for detailed change history of this release

Translation Improvements

- Added new translators edition making it much easier to create and maintain translations of TiddlyWiki
- Added improved Japanese translation
- Added new Danish translation
- Added new Greek translation

Usability Improvements

- Improvements to tooltips for accessibility
- Added export button to tiddler toolbar, page controls and **advanced search** "filter" tab
- Improved layout of **$:/TagManager**
- Simplified default tiddler toolbar buttons
- Added advanced search button to sidebar search results
- Improved search results by listing title matches at the top
- Added input box for specifying new field value in edit template
- Improved tag pill rendering so that the foreground colour is dynamically chosen for maximum contrast with the background colour
- Enhancements to many of the core icons
- Removed (and here, here and here) unnecessary confirmations when abandoning or deleting unmodified tiddlers

- Added a subtle rounded corner to tiddlers and tabs
- Added automatic focusing of the title of newly created tiddlers
- Added advanced search links on tiddler counts in **$:/ControlPanel**

Hackability Improvements

- Introduced new ActionSetFieldWidget, ActionDeleteFieldWidget and ActionDeleteTiddlerWidget for manipulating tiddler fields and values
- Enhanced WidgetMessage: tm-new-tiddler to allow a skeleton to be specified with additional fields
- Enhanced the EditTextWidget to allow more control over the sizing of textarea editors
- Added new canned filter for recently modified system tiddlers in **$:/AdvancedSearch**
- Added new system image for videos: **$:/core/images/video**
- Extended search results to allow custom visualisations - see Customising search results
- Extended `timeline` macro to be able to use different date fields
- Updated the `search` operator to allow a field to be specified
- Updated the highlight plugin to display tabs as spaces
- Extended modal handling to permit variables to be passed to the modal (see WidgetMessage: tm-modal)
- Added simple logging to help track down drop/paste issues (see https://tiddlywiki.com/dev/#ImportLogging)
- Updated WidgetMessage: tm-download-file and WidgetMessage: tm-save-wiki to allow variables to be specified for the rendering
- Added Example Table of Contents: Tabbed Internal
- Extended RevealWidget so that it can be used without specifying a state tiddler
- Added "hooks" mechanism so that plugins can modify the default tiddler list
- Added support for nested popups

Bug Fixes

- Fixed problem with *CodeMirror* editor and missing tiddlers
- Exclude drafts from Table-of-Contents Macros
- Fixed problem with encoding of HTML tiddlers
- Fixed problem with sticky titles being covered by vertical tabs
- Removed flicker when automatically resizing textareas
- Fixed JSON deserialiser so that all fields can be imported

Node.js Changes

These changes are only relevant to people using TiddlyWiki under Node.js

- Extended **includeWikis** mechanism of TiddlyWikiFolders to allow read-only wikis to be included
- Improved error handling under Node.js
- Added UnpackPluginCommand
- Added EditionsCommand to list the available editions

Changes affecting plugin authors

- Disabled wiki change events for shadow tiddlers
- Updated structure of widget messages that have a hashmap parameter
- The support for nested popups has necessitated a change in the way that popups are styled that affects popups that are triggered with the focusPopup attribute of the EditTextWidget: it is now necessary for the widget to be given the class `tc-popup-handle`.

Contributors

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:

- @andrey013
- Birthe C
- @Braincoke
- @BramChen
- @danielo515
- @erwanm
- @Evolena
- @giffmex
- @inmysocks
- Makoto Hirohashi
- @pmario
- @Skeeve
- @tobibeer
- @welford

Release 5.1.6

21st February 2015 at 8:42pm

ReleaseNotes

See GitHub for detailed change history of this release

Translation Improvements

- Added Dutch translation
- Updated Danish and Japanese translations

Hackability Improvements

- Allow buttons to be styled to look like internal links
- Disabled linking when transcluding **caption** field in Table-of-Contents Macros
- Added support for templates to tabbed Table-of-Contents Macros
- Allow shadow tiddlers to appear in Table-of-Contents Macros
- Added tooltip to BrowseWidget
- Improved handling of missing tiddlers by the **`has`** and **`field`** operators

Bug Fixes

- Fixed problem with download saver using incorrect filename
- Fixed problem with processing of date format templates
- Fixed problem with **storytop** theme tweak not being respected when the narrow responsive design kicks in

Node.js Changes

These changes are only relevant to people using TiddlyWiki under Node.js

- Fixed problem with "all tiddlers" static HTML export
- Fixed incorrect checking for an empty folder with the InitCommand

Changes affecting plugin authors

- Extended `$tw.utils.each` so that the loop can be broken out of

Contributors

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:

- @anavarre
- @Drakor
- @dullroar
- @erwanm
- @malgam
- @nameanyone
- @pekopeko1
- @Spangenhelm
- @tobibeer
- @xcazin

Release 5.1.7

19th December 2014 at 10:50pm

ReleaseNotes

See GitHub for detailed change history of this release

This is an hot fix release with the following change over Release 5.1.6:

- Fixed typo in GettingStarted

Release 5.1.8

17th April 2015 at 6:33pm

ReleaseNotes

See GitHub for detailed change history of this release

This release includes many improvements to the documentation for TiddlyWiki. Many thanks to everyone who has helped out, but especially to our prodigious new contributor Astrid Elocson.

Translation Improvements

- Improvements to French, Danish, Chinese and Japanese translations
- Added Czech translation
- Added Interlingua translation

- Added Portuguese translation
- Added Punjabi and Hindu translations
- Added Slovak translation
- Added Spanish translation
- Added localisation for encryption prompt

Usability Improvements

- Added user interface for setting a page background image
- Fixed problem with unnecessary scrolling at startup
- Upgraded the KaTeX Plugin to use KaTeX v0.2.0, with better symbol support
- Added *Help Plugin* providing floating help panel
- Added support for an online plugin library
- Added automatic focusing of the search box at startup
- Introduced the Railroad Plugin by Astrid Elocson (see it in action in the new Filter Syntax documentation)
- Migrated functionality of "Sticky Titles" theme into an optional tweak for the "Vanilla"/"Snow White" themes. This means that tiddler titles will stick to the top of the window during scrolling on browsers that support `position: sticky` (such as Safari and Firefox)
- Added icons to **$:/AdvancedSearch**, **$:/ControlPanel** and **$:/TagManager**
- Changed wording of tiddler "save" button to "confirm"
- Added automatic linking of system tiddlers such as **$:/ControlPanel**
- Improved new tag dropdown to only list title matches
- Added a warning banner notifying the user when modified plugins require the page to be reloaded
- Added first pass at "stacked" story view

- Changed modal headers and footers to be parsed in inline mode by default (avoiding unneeded `<p>` tags)

Hackability Improvements

- Added resolvepath Macro
- Added **class** attribute to CheckboxWidget
- Added support for a special message to be displayed when the main story river is empty
- Improved font rendering under Mac OS X
- Added support for embedding ZIP files
- Added support for custom elements with the RevealWidget
- Added **style** attribute to RevealWidget
- Fixed text inputs to use palette colours
- Several new core icons: 

Bug Fixes

- Fixed the sameday Operator and eachday Operator to accept TW5 date strings
- Fixed version number compatibility checks for plugins
- Fixed problem with repeatedly cancelling a draft
- Improved sandboxing of generated `<iframe>` elements
- Fixed problem with modal backgrounds on Firefox
- Fixed problem with accidentally dragging a tiddler within its originating window

- Fixed problem with dragging certain plugins with Safari
- Partially fixed problem with SVG icons being clipped by 1 pixel on the right and bottom
- Fixed problem with double quotes in a new field value

Node.js Changes

These changes are only relevant to people using TiddlyWiki under Node.js

- Fixed Highlight Plugin to work when generating static files under Node.js
- Fixed problem with case-insensitive filing systems (eg Windows)
- Added mobile metadata to static page templates
- Added added "noclean" parameter to RenderTiddlersCommand
- Added editions Operator and editiondescription Operator for enumerating available editions

Contributors

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:

- @aelocson
- @andreasabeck
- @alexhough
- @Arlen22
- @BramChen
- @danielo515

- @dpdannemiller
- @Drakor
- @evgeniy-gryaznov
- @felixhayashi
- @gernert
- @Infurnoape
- @Jc-L
- @le-pako
- @Mathobal
- @ng110
- @pmario
- @reflectionist
- @roma0104
- @simonbaird
- @Spangenhelm
- @sukima
- @tobibeer
- @twMat
- @welford
- @xcazin

Release 5.1.9

3rd July 2015 at 5:37pm

ReleaseNotes

See GitHub for detailed change history of this release

Major Improvements

New "Powered by TiddlyWiki" plugin

An experimental plugin to add a simple "Powered by TiddlyWiki" banner to your site. It is included in this prerelease but will not be on the main tiddlywiki.com wiki.

New "Fluid story, fixed sidebar" mode

It is now possible to arrange the main window so that the sidebar has a fixed width and the story river expands to fill the remaining space.

To switch it on, visit **Theme Tweaks** in the **$:/ControlPanel** **Appearance** tab and use the following options:

- Sidebar layout:
  - **Fixed story, fluid sidebar** (default) - the story river has a fixed width and the sidebar fills the remaining space
  - **Fluid story, fixed sidebar** - the story river expands to fill horizontal space remaining after the fixed width sidebar
- Sidebar width:
  - The width of the sidebar. Can be specified in pixels (eg **350px**), a percentage (eg **25%**) or other CSS unit

Toolbar Button Style Setting

A new setting in **$:/ControlPanel** **Settings** tab allows the toolbar button style to be chosen from the following options:

- borderless the existing style
- boxed, which adds a thin border box around the button
- rounded, which rounds the button corners and inverts the button colours

Plugins can add new style options.

New Tiddler Toolbar Button: "Open in new window"

An experimental new tiddler toolbar button opens a single tiddler in a separate pop-up browser window. The tiddler will be dynamically updated just as in the main window. There are several uses:

- Arranging reference tiddlers for easy access while editing in the main window
- Making good use of multi-screen layouts
- Printing the content of a single tiddler
- Running presentations in a separate window while maintaining notes in the main window

Improvements for visually impaired users

- Improved monochrome palette variants:
  - Dark palette. Foreground: , Background:

- Light palette. Foreground: , Background:
- Revert

New "Solar Flare" palette

There's a new core palette "Solar Flare", contributed by Rustem Akbulatov (@nameanyone).

Try it out Revert

Options for Story Control

There are new options for controlling the position in the story river of newly opened tiddlers. Visit the **Settings** tab of **$:/ControlPanel** and look for "Tiddler Opening Behaviour".

New Hidden Settings

Two new hidden settings for advanced configuration:

- Hidden Setting: Search AutoFocus
- Hidden Setting: Typing Refresh Delay

New Vars widget

Added VarsWidget for setting multiple variables in one operation:

```
<$vars greeting="Hi" me={{!!title}} sentence=<<helloworld>>>
  <<greeting>>! I am <<me>> and I say: <<sentence>>
</$vars>
```

Other Improvements

Translation Improvements

- Updates to the Dutch, Chinese, Italian and German translations
- New Catalan (Spain) translation

Usability Improvements

- Added new page control icon for setting the palette: palette
- Added new icons for palette  and "open new window" 
- Improved the theme icon to better match the new palette icon: 
- Avoid glitch with autofocusing the search box in the plugin library
- Added information about TiddlyDesktop to Releases
- Stopped autosaving after cancelling an edit
- Added dropdown to edit template to allow existing field names to be selected

Hackability Improvements

- Updated to KaTeX version 0.4.3, with the following editions since the previous 0.2.0 release that was distributed with TiddlyWiki 5.1.8:
  - Added support for some `\begin` / `\end` environments, including support for `array`, `matrix`, `pmatrix`, `bmatrix`, `vmatrix`, and `Vmatrix`
  - Added support for optional `\sqrt` arguments, e.g. `\sqrt[3]{x}`
  - Add support for `\phantom`
  - Add `\#`, `\&`
  - See the KaTeX GitHub page for full release details
- Extended ActionSendMessageWidget for more flexible passing of named parameters via new `$name` and `$value` attributes
- Added styling for `<kbd>` elements such as `escape` and `enter`
- Added subtle new "pin stripe" pattern for page background - Try it out
- Added support for custom elements to the ButtonWidget
- Improved *CodeMirror* plugin to automatically refresh when a tiddler type changes
- Added new setting in **$:/ControlPanel** to render tiddler titles as links, which can be convenient for drag and drop
- Simplified detecting the preview pane with the tv-tiddler-preview Variable
- Improved automatic linking of system tiddler titles by restricting them to letters and the characters `/.-`

Bug Fixes

- Fixed problem with special characters in field names
- Fixed static banner layout on mobile devices
- Fixed popups within the scrollable widget by adding `position: relative;` to example scrollable widget styles

Contributors

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:

- @BramChen
- @buggyj
- @cehmke
- @danielo515
- @Drakor
- @ericshulman
- @felixhayashi
- @gernert
- @heroboy
- @idoine
- @infurnoape
- @inmysocks
- @kixam
- @lolole13
- @marcor
- @Marxsal
- @nameanyone
- @pmario
- @r1chard5mith

- @roma0104
- @silvyn
- @slovuj
- @StevenMcD
- @sukima
- @vouko
- @welford
- @xcazin

Release 5.2.0

3rd October 2021 at 5:15pm

ReleaseNotes

See GitHub for detailed change history of this release



Congratulations to Frank. B for his winning design for the banner for this release (here are the other entries).

Highlights

Unrestricted Fieldnames and the New JSON Store Area

The major change in this release is that the format used to store tiddlers in TiddlyWiki's HTML file has changed from an HTML DIV-based format to a new JSON-based format. The consequences of this change are far-reaching. They are thus the primary rationale for bumping the version number from v5.1.x to v5.2.0:

Firstly, the **characters that can be used in field names are now unrestricted** (just like tiddler titles), making it possible to use fieldnames like `My^Field` or . This has become possible because every other part of TiddlyWiki was already capable of dealing with unrestricted field names apart from saving as an HTML file.

Secondly, external tools that read, write or otherwise process TiddlyWiki HTML files will need modification to understand the new store area format. The details are documented in the /dev wiki.

For more details, see ticket #5708 on GitHub.

Dragging and Dropping Images into the Editor

It is now possible to drag and drop an image file directly into the tiddler text editor. A popup allows the incoming image tiddler to be renamed, and then `[img[title]]` is automatically inserted into the tiddler text at the insertion point.

- **added** (and again here) ability to drag and drop images directly in the editor

Simplified Access to Tiddler Title with ActionCreateTiddlerWidget

It is now possible to embed actions within the body of the ActionCreateTiddlerWidget, and those actions can directly access the title of the newly created tidddler via the variable `createTiddler-title`, and the title of the corresponding draft in the variable `createTiddler-draftTitle`. The old `$savetitle` and `$savedrafttitle` attributes are now deprecated.

```
<$button>
  <$action-createtiddler $basetitle="Homemade Button" tags="$:/tags/PageControls" text={{$:/state/new-
button-caption}}>
    <$action-navigate $to=<<createTiddler-title>>/>
  </$action-createtiddler>
  Create a new non-functional page control button and open the tiddler
</$button>
```

For more details:

- **extended** ActionCreateTiddlerWidget to make the new title available as a variable

More Flexible Parsing of Macros

Two long-standing limitations of TiddlyWiki's macro syntax have been resolved.

First, it is now possible to nest macro calls within the parameters to other macros. For example:

```
<<mymacro arg:"""nested <<macro>> call""">>
```

Second, macro parameters can now be used within filter expressions. For example:

```
[<qualify "$:/state/mydata">addsuffix[-]addsuffix<now "YYYY MM DDD hh:mm:ss">]
```

- **improved** macro calls to use the same parser as that used for widget attributes
- **added** support for macro parameters within filters (see Filter Parameter)

Optimised Refreshing of Transclusions

This simple-sounding change has several important consequences:

The first consequence is that it makes the TranscludeWidget much more efficient, and improves the performance of the refresh cycle.

The second consequence resolves a long-standing difficulty for many users: it is now possible to use an EditWidget to edit another field of the same tiddler. Previously, the edit widget would not work properly in such cases because it would be refreshed on every keypress, leading to the current editing caret/selection position being lost.

As sometimes happens, the third consequence is that constructions that relied on the old, buggy behaviour may no longer work as expected. A common example of such a construction is an invocation of a JavaScript macro. The problem stems from the fact that JavaScript macros are designed to be as simple as possible from the developers perspective, being pure functions that have no side effects, and that always return the same value for the same parameters. In particular, JavaScript macros cannot participate in TiddlyWiki's refresh mechansim, so they cannot force themselves to be updated when underlying data changes (for situations where such processing is needed, JavaScript widgets should be used instead of JavaScript macros).

However, developers have sometimes worked around this limitation by using the TranscludeWidget to force refreshing of a macro invocation when a particular tiddler changes. The newer, more selective behaviour means that sometimes such widgets will avoid a refresh that was previously relied upon to update the JavaScript macro.

For end users, if an upgrade to v5.2.0 causes problems then consult the discussion Forums to see if other users have encountered the same problem, and make a new posting if needed.

- **improved** the TranscludeWidget to avoid triggering a refresh unless the transcluded field changes (previously it was also triggered for changes to other fields)

Other Notable Improvements

extended (and here) the ListWidget with an optional `counter` attribute specifying a variable to contain the numeric index of each list item. This is useful for creating a list of numbered items, and for detecting the first and last entries in a list

extended action widget execution with an optional new mode that ensures that all widgets are refreshed before invocation – see ActionWidget Execution Modes. This resolves a long-standing inconsistency in the way that ActionWidgets are executed, making them behave much more like a conventional programming language

improved the HTML comment syntax to enable it to be used before and in between pragmas such as macro definitions. This makes it much easier to provide documentation within complex wikitext tiddlers

fixed size of dropzone when story river is empty. For a long time, TiddlyWiki confusingly wouldn't recognise a drag and drop operation if the story river was empty

Performance Improvements

- **improved** (and again here) the efficiency of the linked list implementation used in filter processing
- **improved** the all Operator and here the links Operator, and here the backlinks Operator to use the new linked list implementation
- **improved** ETag-based browser-side caching for all resources and GZip or Deflate compression for all responses to GET requests
- **improved** plugin internal formatting to remove unneeded whitespace, saving about 6% on the size of the $:/core plugin
- **improved** performance of sort Operator through optimising common cases

Usability Improvements

- **fixed** ability to import previously blocked system tiddlers
- **improved** warning message when cancelling a pending import
- **updated** the default font weights in the "Snow White" theme to be thicker and more legible
- **fixed** modal mechanism to prevent page from scrolling while modals are displayed
- **fixed** fill colour for "remove tag" button
- **fixed** page title so that the separating em-dash is only used if the site subtitle is present
- **fixed** broken aria-label in **$:/PaletteManager**
- **updated** SVG optimiser script
- **added** plus/minus SVG icons: ⊕ and ⊖
- **added** link to the existing tiddler when the warning "Target tiddler already exists" is displayed in the edit template
- **added** class to make tag pills in the edit template look like those in the *ViewTemplate*
- **fixed** overflow of long field values that do not contain whitespace
- **improved** GitHub, Gitea and GitLab savers to wikify the commit message, enabling macros to be used in the message
- **improved** the text editor stamp toolbar button dropdown to allow sorting by drag and drop

Widget Improvements

- **added** (and here) new MessageCatcherWidget to catch arbitrary messages and execute action strings
- **modified** FieldManglerWidget to ensure it doesn't propogate events that it traps

- **extended** DropzoneWidget to optionally invoke actions after the `tm-import-tiddlers` message has been sent, and to specify an optional `contentTypesFilter` which determines which content types are accepted by the dropzone.
- **extended** DropzoneWidget to also use the specified deserializer for strings either dropped or pasted on to the dropzone
- **fixed** SelectWidget refreshing
- **fixed** ActionListopsWidget bug by avoiding concatenating fragments of filter expressions
- **extended** ActionPopupWidget to create floating popups that must be manually cleared
- **extended** (and here) KeyboardWidget to provide additional information variables
- **extended** DraggableWidget with option to hide the drag image

Filter improvements

- **added** new :map filter run prefix
- **added** new :sort filter run prefix
- **added** average Operator, median Operator, variance Operator and standard-deviation Operator for calculating averages of a list of numbers
- **added** deserializers filter Operator for obtaining a list of the available deserializers
- **added** format:titlelist operator for selectively adding double square brackets around titles
- **added** charcode Operator for easily accessing special characters like tab and linefeed
- **added** support for suffixes to filter run prefixes
- **extended** search-replace Operator with **m** flag for multiline mode
- **extended** range Operator to allow parameters to be specified as separate operands
- **added** moduleproperty Operator and extended modules Operator to provide access to internal JavaScript modules

Hackability Improvements

- **extended** unusedtitle Macro to add **separator** and **template** parameters
- **added** **color-scheme** field to all themes to differentiate between light and dark themes
- **added** support for `image/vnd.microsoft.icon` content type
- **added** support for throttling refreshes for changed tiddlers with the title prefix *$:/temp/volatile/*
- **added** a hidden setting to control sandboxing of tiddlers of type `text/html`
- **added** text operation insert-text
- **added** **data-tag-title** attribute to tag pills
- **added** support for dynamic toolbar buttons
- **added** new SystemTag: $:/tags/ControlPanel/Plugins for customising the tabs shown under the "Plugins" tab of control panel
- **added** support for DateFormat tokens for weekday number and day of year
- **added** further transliteration pairs for the slugify Operator

Client-server Improvements

- **fixed** crash running in client server configuration when 'ETag' header is missing
- **fixed** blank favicon when using lazily loaded images
- **fixed** web server issue with custom path prefix and basic authentication
- **fixed** crash on Node.js with ActionSetFieldWidget when type field is given a value upon new tiddler creation
- **fixed** issue with saving tiddler files with titles that already end in the required extension

- **fixed** several consistency issues with the filesystem plugin
- **fixed** issue with encoding of $:/config/OriginalTiddlerPaths outside the wiki folder
- **updated** the *TiddlySpot* Saver settings form
- **added** 401 and 403 error messages for PUT saver
- **extended** the external JavaScript template to be easier to use and work more smoothly when offline
- **fixed** incorrect usage of X-Requested-With header for simple HTTP requests
- **fixed** security issue allowing access to files outside the `./files/` folder

Node.js Improvements

- **extended** RenderCommand to allow multiple variables to be passed

Plugin Improvements

Markdown Plugin

- **added** support for using the KaTeX Plugin in Markdown
- **fixed** extraneous paragraph tags in lists

XLSX Utilities Plugin

- **fixed** crash with the XLSX Utils plugin

KaTeX Plugin

- **updated** to *KaTeX* v0.13.18
- **added** custom macro editor
- **added** automatic numbering reset for each displayed tiddler

Freelinks Plugin

- **extended** the Freelinks plugin with a filter to determine which tiddlers can be the targets of freelinks

Menubar Plugin

- **fixed** Menu plugin to support optional **dropdown-position** field

BibTeX Plugin

- **fixed** *BibTeX* Plugin to report errors more sensibly
- **modified** *BibTex* Plugin to force fieldnames to be lowercase

Developer Experience Improvements

- **improved** the widget subclassing mechanism to work with widgets that add event listeners in their constructor
- **improved** the Jasmine test suite output

- **extended** wikitext parser with a subclassing mechanism
- **added** added support for **utils-browser** modules
- **added** **th-before-importing** hook to allow plugins to inspect or modify the `importTiddler` object before any tiddlers are imported
- **added** **th-closing-tiddler** hook when tiddlers are closed

Translation improvements

- **added** Polish translation
- **improved** Catalan translation
- **improved** Chinese translations
- **improved** French translation
- **improved** German translations
- **improved** Spanish translation

Other Bug Fixes

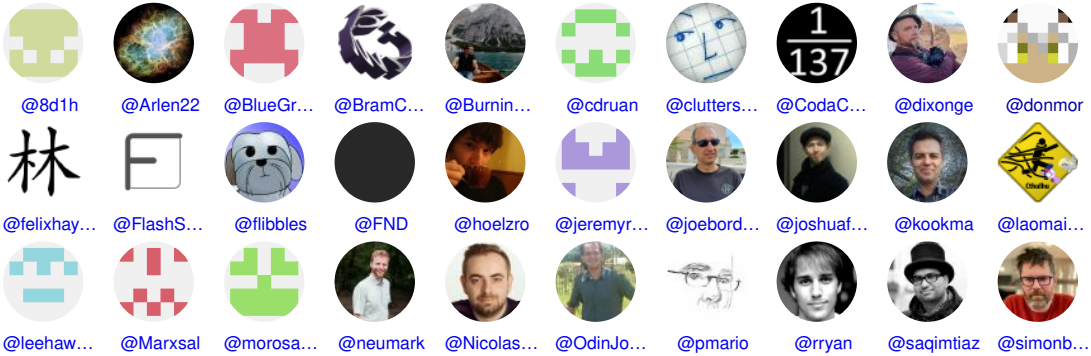
- **fixed** crash with malformed input to `decodeURIComponent()`
- **fixed** issue with lookup Operator returning "undefined" under some circumstances
- **fixed** crash with unterminated wikitext comments
- **fixed** tiddler info area content bleeding on close animation
- **fixed** inline/block widget parsing glitch

- **fixed** runaway regexp when parsing filters
- **fixed** right margin of tag pill when used outside of the tags wrapper
- **extended** upload saver to optionally work without a username or password
- **fixed** RadioWidget to refresh selectively, and to use the **checked** attribute correctly
- **fixed** "invert" option of `wiki.search()` method
- **fixed** ES5 compatibility issue
- **fixed** exporting of tiddlers that begin and end with double quotes
- **improved** accessibility of button widget when controlling a popup
- **fixed** EditTextWidget to use default text for missing fields
- **fixed** css-escape-polyfill to work under Node.js
- **fixed** (and here) crash when sorting by non-string fields
- **fixed** (and here) some bugs in the EventCatcherWidget, introduced new `stopPropagation` attribute and new syntax for specifying actions
- **fixed** *CurrentTiddler* variable consistency in subfilters and prefixes
- **fixed** crash when accessing variables in filters that don't have a widget context
- **fixed** unnecessary triggering of reload warning when JavaScript tiddlers are not subsequently imported
- **fixed** minor issue with import pragma
- **fixed** leading and trailing whitespace in themes
- **fixed** configuration list of HTML5 block elements
- **fixed** shape and color for disabled button to work with `tc-btn-invisible` class
- **fixed** inconsistent spacing of view toolbar items
- **fixed** crash when assigning new value to file input controls

- **fixed** text editor stamp button to work with prefixes and suffixes in shadow tiddlers
- **fixed** Internet Explorer crash when pasting text using an HTA
- **fixed** bug with `[all[shadows]]` sometimes returning overridden shadow tiddlers
- **fixed** refresh bug with radio widget where the tc-radio-selected class is not correctly updated
- **fixed** crash with reading invalid JSON files

Acknowledgements

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:








@slavma… @sobiorn… @twMat @xcazin

Release 5.2.1

8th December 2021 at 12:58pm

ReleaseNotes

See GitHub for detailed change history of this release



Congratulations to Mohammad Rahmani and Elise Springer for their winning design for the banner for this release (here are the other entries).

Highlights

added New Filter Cascade Mechanism

The cascade mechanism provides a flexible way to choose between multiple options. Crucially, it is done in a way that makes it simple for plugins to tweak the logic or add their own options.

Some of the things that can be done with cascades are:

- Replacing the default tiddler body template with a custom one for a certain type of tiddler
- Giving all journal tiddlers a custom icon
- Implementing a custom editor for certain types of tiddler

There are two parts to these changes. The underpinning is a new Cascade Filter Run Prefix that takes a list of filters and runs them in order, returning the result of the first one to return a value.

The second part is a series of improvements based on this new filter cascade mechanism. Conditional logic that was previously hidden within core templates can now be extended and tweaked much more easily than before.

Choosing the following elements of the core user interface is now handled with cascades:

- Story Tiddler Templates
- Tiddler Colour
- Tiddler Icons
- View Template Body
- View Template Title
- Edit Template Body

The cascades can be inspected in **$:/ControlPanel** under **Info** -> **Advanced** -> **Cascades**.

See Cascades for more information.

added New LetWidget

The LetWidget is an improved alternative to the existing VarsWidget. It is recommended to use the new LetWidget instead of the VarsWidget in

all circumstances.

The chief advantage is that the LetWidget performs the variable assignments in the same order as they are written, and permits references to earlier assignments. For example, here we swap the values of two variables:

```
<$let temp=<<foo>> foo=<<bar>> bar=<<temp>>>
...
</$let>
```

Bug Fixes

- **fixed** fixed issue with drag and drop in Chrome 96
- **fixed** fixed issue with drag and drop and the CodeMirror Plugin in Chrome 96
- **fixed** refreshing of LinkWidget attributes
- **fixed** crash when using "source=basename-uri-decoded" in tiddlywiki.files
- **fixed** **$timestamp** attribute ignored when using ActionSetFieldWidget to set the value of an index
- **fixed** crash during static rendering of CodeMirror Plugin

Usability Improvements

- **fixed** image picker in theme tweaks to not dismiss when an image is selected
- **added** a new hidden setting for controlling the visibility of the editor preview pane on a per-tiddler basis

- **improved** sidebar plugin listing to show icon and more details

Widget Improvements

- **fixed** refreshing of the CodeBlockWidget
- **extended** EditTextWidget to provide a new **actionValue** variable to action strings that contains the value of the input

Filter improvements

- **added** new trigonometric operators: acos, asin, atan, cos, sin and tan
- **added** new zth Operator that works like nth Operator but counts from zero instead of one
- **extended** Map Filter Run Prefix to provide additional variables to the filter
- **fixed** macro parameters in filter run prefixes

Hackability Improvements

- **added** new ActionSetMultipleFieldsWidget, new SetMultipleVariablesWidget and extended ActionSendMessageWidget for working with multiple variables/fields/indexes/parameters in one operation
- **added** new `focus-editor` operation to WidgetMessage: tm-edit-text-operation

Developer Improvements

- **added** support for widgets to access the order in which attributes are defined

Node.js Improvements

- **added** support for recursively loading subdirectories with tiddlywiki.files Files

Translation improvements

- Polish
- Chinese

Acknowledgements

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:


@bmann

@BramC…

@btheado

@Burnin…

@eiro10

@Evident…

@flibbles

@joshuaf…

@Marxsal

@pmario






@saqimtiaz @Telumire @tw-FRed @twMat

Release 5.2.2

25th March 2022 at 2:08pm

ReleaseNotes

See GitHub for detailed change history of this release



Congratulations to Elise Springer for her winning design for the banner for this release (here is the competition thread).

Highlights

updated Highlight Plugin to use highlight.js v11.4.0

This is a major upgrade to the latest version of highlight.js. The new version has many improvements, including better support for Fortran.



> The new version of the Highlight Plugin requires a modern browser that fully supports JavaScript ES6 (released in 2015). The older version is still available as the **highlight-legacy** plugin for users who need to use an older browser.

extended support for secondary windows

The WidgetMessage: tm-open-window now supports an optional `windowID` parameter so that opened windows can be closed with the new WidgetMessage: tm-close-window message. There is also a new WidgetMessage: tm-close-all-windows message which closes all secondary windows.

added support for line breaks within filtered transcluded attributes

To improve readability, it is now possible to use newlines as whitespace within filtered transcluded attributes of HTML elements and widgets. For example:

```
<span class={{{
    [<currentTiddler>addsuffix[-primaryList]]
    :except[<searchListState>get[text]]
    :and[then[]else[tc-list-item-selected]]
}}}>
```

fixed inconsistent ordering of tagged tiddlers

This was a long standing bug that manifested itself in several ways.

The root cause was that the order in which tiddlers were enumerated depended upon the order in which they had been added to the store.

The effect was that lists based on enumerating tiddlers would show different results depending upon whether new tiddlers were added since the wiki was reloaded.

For example, adding a new tiddler with a given tag previously caused the new tiddler to appear at the bottom of the tag pill listing for that tag. Saving and reloading the wiki would reorder the list to put the new tiddler in the correct position.

The fix ensures that the enumeration order remains consistent.

fixed some incorrectly nested `<span>` and `<div>` elements

TiddlyWiki's core page layout has historically included several instances of inline `<span>` elements containing block `<div>` elements, something that is technically invalid HTML. In practice, browsers have always handled these cases leniently to make the page work but it has also been unhelpful for developers to encountering invalid HTML constructions in their first interactions with TiddlyWiki.

The immediate prompt for starting to fix these issue now is that Chrome v100 includes a change that causes some of these misnested elements to be rendered correctly visually but to be unclickable in some situations. (The change is reverted in Chrome v102, but it still makes sense to fix it)

Plugin Improvements

- **fixed** issue with LaTeX content within Markdown tiddlers
- **fixed** incorrect handling of dropdown classes in the menu bar plugin
- **updated** the Dynannotate plugin to use the new View Template Body Cascade to avoid modifying the default view template.

Translation improvements

- Polish
- Chinese
- French

Usability Improvements

- **improved** wording of drag and drop banner (from "drop here" to "drop now")
- **fixed** wrapping and wikification of field names in field viewer
- **fixed** missing whitespace between description and MIME type in edit template dropdown for the **type** field
- **fixed** incorrect usage of code view for certain system tiddlers
- **fixed** sidebar plugin listing to be sorted by name rather than description

Widget Improvements

- **added** classes to the ImageWidget to indicate whether it is loading, loaded or has encountered an error
- **fixed** RadioWidget not using default value if the field or index is missing
- **fixed** issue with ImportVariablesWidget when importing block mode widgets
- **extended** DraggableWidget to support an optional drag handle

Filter improvements

- **added** support for case insensitive matching for the prefix Operator and suffix Operator
- **added** sha256 Operator
- **fixed** crash when using deprecated regexp operands for filter operators
- **extended** lookup Operator to support indexes as well as fields
- **extended** search Operator with new 'some' flag
- **fixed** untagged Operator to consider non-existent tiddlers to be untagged
- **extended** insertbefore Operator to accept the position title as a parameter, instead of as a variable name in the suffix
- **fixed** butlast Operator to be consistent with the rest Operator

Hackability Improvements

- **added** Field Editor Cascade for customising how fields are displayed in the edit template
- **added** support for directly specifying style properties on HTML elements (for example, `<div style.color={{!!color}}>`)
- **added** support for **code-body** field set to **yes** to trigger display of a tiddler in the code view
- **added** support to WidgetMessage: tm-scroll for scrolling without animating
- **added** support for WidgetMessage: tm-relink-tiddler
- **extended** WidgetMessage: tm-open-window to support 'top' and 'left' parameters
- **added** new **matchSelector** attribute to the EventCatcherWidget

Developer Improvements

- **added** support for **renderEnd()** method for storyviews
- **added** sourceURL tags to $:/boot/boot.js and $:/boot/bootprefix.js, enabling them to be accessed in the browser debugger more easily

Node.js Improvements

- **added** several new web server options: admin, required-plugins and tls-passphrase

Performance Improvements

- **added** support for caching the listed Operator
- **improved** stylesheet handling to avoid unnecessary refreshes

Bug Fixes

- **fixed** importing/upgrading encrypted single file wikis
- **fixed** WidgetMessage: tm-edit-text-operation crash with **wrap-lines** operation if prefix or suffix is missing
- **fixed** processing of *$:/tags/RawMarkupWikified/TopHead* tiddlers
- **fixed** issue whereby renaming tags could result in duplicate tags
- **fixed** search results obscured on narrow screens
- **fixed** ActionListopsWidget bug with input list items starting with `-`, `+`, `~`, `:` or other character with a special meaning in filters
- **fixed** (and here) refreshing of the ButtonWidget when the `default` attribute changes

Acknowledgements

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:


@benwe…

@BramC…

@btheado

@cdruan

@CodaC…

@damscal

@davout…

@Evident…

@FlashS…

@flibbles

@FSpark

@ibnishak

@jc-ose

@joshuaf…

@linonetwo

@Marxsal

@nilslind…

@oflg

@pmario

@rryan

@saqimtiaz

@slayma…

@tw-FRed

@twMat

Release 5.2.3

2nd August 2022 at 2:25pm

ReleaseNotes

See GitHub for detailed change history of this release



Congratulations to feedthegood for their winning design for the banner for this release (here is the competition thread and the voting thread).

Plugin Improvements

- **updated** (and **here**) KaTeX Plugin to KaTeX v0.16
- **extended** BrowserStorage Plugin to be able to delete existing tiddlers as well as modify or add tiddlers
- **extended** (and **here**) Markdown Plugin to add the **link** and **linkify** editor toolbar buttons
  - The linkify button just inserts `[]()`. If any text is selected, it will be inside the square brackets: `[text]()`
  - The link button opens a popup menu in which you can either paste a URL or search for an existing tiddler
- **extended** Markdown Plugin to add the **codeblock** editor toolbar button
- **extended** Markdown Plugin to add `ctrl-M` (Mac) or `alt-M` (other platforms) to create a new Markdown tiddler
- **extended** Browser Sniff Plugin to expose **$:/info/browser/is/mobile**
- **fixed** BrowserStorage Plugin crashing if local storage not available
- **improved** (and **here**) Dynannotate Plugin selection tracker, making it easier to add a popup menu to text selections

Translation improvements

- Chinese
- French

- German
- Japanese
- Polish

Accessibility Improvements

- **improved** ARIA support for the sidebar and story river
- **improved** ARIA support for notifications so that screen readers will automatically read them
- **improved** ARIA support for the edit template

Usability Improvements

- **updated** ActionSetFieldWidget to avoid inadvertent changes to the current tiddler
- **updated** "put" and "upload" savers (as used by TiddlyHost to display error responses from the server
- **updated** (and **extended**) various palettes to work with **color-scheme: dark**
- **extended** the monospaced blocks and block quotes editor buttons so that they can be undone by clicking the button again
- **updated** field and tag editors to trim whitespace
- **improved** formatting of stylesheet tiddlers to use syntax highlighting if the Highlight Plugin is installed

Widget Improvements

- **extended** CheckboxWidget to support the indeterminate state
- **extended** CheckboxWidget to support the **listField** and **filter** attributes
- **extended** DraggableWidget to support an **enabled** attribute
- **extended** DraggableWidget to pass additional context variables to the **dragstartactions** action string
- **improved** refreshing of DraggableWidget
- **fixed** brittle selector implementation for the DraggableWidget
- **updated** the ImageWidget to support lazy loading of images

Filter improvements

- **added** new insertafter Operator to match the existing insertbefore Operator
- **fixed** handling of `{!!title}` in a filter with no currentTiddler variable set
- **added** new "flat" option to the Map Filter Run Prefix enabling multiple items to be returned from each invocation of the filter run

Hackability Improvements

- **updated** tiddler subtitle rendering to allow individual elements to be controlled via the SystemTag: $:/tags/ViewTemplate/Subtitle
- **added** support for SystemTag: $:/tags/Macro/View/Body
- **extended** colour Macro to allow for palette-specific fallback colours to be specified
- **improved** whitespace and indentation of tabs Macro to improve readability
- **added** **color-scheme** CSS property to the root of the Vanilla base theme

- **extended** EventCatcherWidget to support `tv-widgetnode-width` and `tv-widgetnode-height`
- **extended** list-links-draggable Macro to support an message to be displayed if the list is empty

Bug Fixes

- **fixed** issue with drag and drop running on mobile Chrome browsers
- **fixed** problem when switching fields in the editor causing their values to be cleared
- **fixed** incorrect **color-scheme** metatag for iframe content with the framed editor
- **fixed** crash when using the SaveCommand to attempt to save missing fields
- **fixed** bug with formatting UTC date strings
- **fixed** SaveCommand crash when attempting to save missing tiddlers
- **fixed** fix broken style block behaviour
- **fixed** incorrect display of image system tiddlers as text
- **fixed** erroneous link rendering within captions in list-links Macro
- **fixed** erroneous link rendering within captions in list-links-draggable Macro
- **fixed** bug with JavaScript modules and lazy loading
- **fixed** fixed tiddler title indentation discrepancy
- **fixed** problem with numbered list editor button not undoing markers in Markdown tiddlers
- **fixed** palette manager showing duplicate entries
- **fixed** crash with missing palette tiddlers
- **fixed** (and **here**) search inputs not to trigger Chrome's password autocomplete popup
- **fixed** embedded SVG foreignObject namespace

- **fixed** anchor links not working when address bar is updated with a permalink, and animation duration is set to zero
- **improved** positioning of server page control dropdown
- **fixed** "save changes" button in the menu bar not turning red when the wiki is dirty
- **fixed** hardcoded spaces in Table-of-Contents Macros
- **fixed** state handling in tabs Macro
- **fixed** fixed formatting of IPv6 URLs by ListenCommand and ServerCommand
- **fixed** crash when closing multiple plugin libraries

Developer Improvements

- A number of core tiddlers have been refactored to use `\whitespace trim` for improved readability. The work was split into a number of PRs: #6257, #6265, #6269, #6270, #6272, #6275, #6276, #6587, #6600, #6604, #6611

Node.js Improvements

- **extended** RenderCommand to support the `storyTiddler` variable

Performance Improvements

- **updated** filter processing to allow compiled filters to be cached

Acknowledgements

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:


@Arlen22

@BramC…

@btheado

@Burnin…

@damscal

@es-kha

@Evident…

@FlashS…

@flibbles

@FSpark

@fu-sen

@ibnishak

@jeremyr…

@joshuaf…

@kookma

@linonetwo

@Marxsal

@MaxGy…

@ndarilek

@oflg

@pmario

@rmunn

@saqimtiaz

@simonb…

@Telumire

@tobibeer

@tw-FRed

@twMat

@xcazin

Release 5.2.4

13th December 2022 at 5:31pm

ReleaseNotes

See GitHub for detailed change history of this release





Congratulations to dmikh for their winning design for the banner for this release (here is the competition thread and the voting thread).

Important Update

After the release of v5.2.5, we found some backwards compatibility issues with the stylesheet changes in #7039 and #6910. We have therefore decided to make a rapid bug fix Release 5.2.5 to resolve these issues, and all users should upgrade to the new version.

Major Improvements

New Twitter Archivist plugin to import the tweets and associated media from a Twitter Archive as individual tiddlers.

added new GenesisWidget that allows the dynamic construction of another widget, where the name and attributes of the new widget can be dynamically determined, without needing to be known in advance

added (and **here**) new operators for reading and formatting JSON data: jsonget Operator, jsonindexes Operator, jsontype Operator, jsonextract Operator and format Operator

Translation Improvements

Improvements to the following translations:

- Chinese
- French
- German
- Polish
- Spanish
- Japanese

Improvements to the translation features of TiddlyWiki itself:

- **improved** the Translators Edition to add an option to display the original English text underneath the text area
- **updated** "delete" button text in **$:/AdvancedSearch** so that it is translatable

Usability Improvements

- **fixed** problem with long presses on tiddler links triggering a preview on iOS/iPadOS
- **improved** consistency of button and input elements across browsers
- **fixed** edit preview to use the View Template Body Cascade
- **fixed** opening a tiddler in a new window to use the View Template Body Cascade
- **improved** detection of infinite recursion errors in widgets and filters
- **extended** default styles for styled runs
- **improved** upgrade wizard to make the version number more prominent
- **improved** parsing of tiddlers containing CSV data for greater compatibility
- **added** new page control button to summon the layout switcher

- **fixed** folded tiddlers to ensure that the unfold button is always visible
- **improved** handling of Modals to optionally allow them to be dismissed by clicking on the background

Widget Improvements

- **fixed** EditTextWidget to remove the default text "Type the text for the tiddler 'foo'"
- **added** **focus** attribute to SelectWidget
- **removed** restriction of the LetWidget being unable to create variables whose names begin with a dollar sign

Filter improvements

- **fixed** issue with availability of variables within filter runs
- **fixed** issue with removing multiple items from a linked list during filter processing

Hackability Improvements

- **added** support for nested macro definitions
- **added** support for SystemTag: $:/tags/ClassFilters/TiddlerTemplate and SystemTag: $:/tags/ClassFilters/PageTemplate to assign dynamic CSS classes to both tiddler frames and the page template
- **improved** safe mode to disable wiki store indexers
- **fixed** safe mode to prevent globally disabling parser rules

- **extended** keyboard shortcut handling to allow to global shortcuts to override all other shortcuts
- **extended** list-links Macro to allow the rendered field to be overriden
- **improved** Table-of-Contents Macros to allow the default icons to be overridden
- **added** **data-tags-*** and **data-tiddler-title** attributes to the edit preview area
- **extended** timeline Macro to override the link template
- **added** support for Unix epoch timestamps in date format strings
- **fixed** the "big green download button" to use the defined palette colour
- **added** new hidden setting to use horizontal tabs for the "more" sidebar tab
- **extended** the external JavaScript templates to allow the URL of the external script file to be configured

Bug Fixes

- **fixed** truncated search results on small screens
- **fixed** table contents overflow on small screens
- **fixed** "new image" keyboard shortcut not to assign journal tags
- **fixed** SelectWidget class to update if it uses a filter
- **fixed** issue with wikification within the advanced search filter dropdown
- **fixed** the table in *$:/Import* to avoid creating hidden empty rows
- **fixed** advanced search keyboard shortcut not navigating correctly
- **fixed** erroneous display of drafts within the advanced search filter dropdown
- **fixed** backwards compatibility of new field editor cascade introduced in v5.2.3

Node.js Improvements

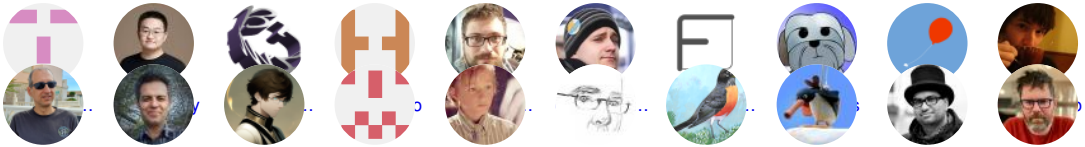
- **added** new CommandsCommand to enable command tokens to be dynamically generated from a filter
- **improved** console logging to avoid spaces and `<empty string>` message
- **fixed** problem with lazy loading deleting tiddler bodies under certain circumstances
- **fixed** handling of ".mp4" file extension so that it defaults to video not audio
- **added** test server to the plugin library edition
- **added** Hidden Setting: Sync Logging to control logging of sync-related messages
- **updated** Jasmine plugin to require the explicit use of the `--test` command in order to cause the tests to be run

Performance Improvements

- **improved** performance of `wiki.getTiddler()`
- **improved** performance of variable prototype chain handling
- **improved** performance of list handling during filter processing

Acknowledgements

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:



Release 5.2.5

19th December 2022 at 7:45pm

ReleaseNotes

See GitHub for detailed change history of this release



Congratulations to dmikh for their winning design for the banner for this release (here is the competition thread and the voting thread).

This is a bug fix release intended to resolve backwards compatibility issues discovered in v5.2.4. See this GitHub ticket for the background.

The issues are entirely cosmetic stylesheet changes, and do not affect the functionality of TiddlyWiki. However, we encourage all users to upgrade to this new version for consistency.

Since v5.2.5 replaces v5.2.4 that was only released for a week, here is the release note for v5.2.4.

Release Note for v5.2.4

See GitHub for detailed change history of this release



Congratulations to dmikh for their winning design for the banner for this release (here is the competition thread and the voting thread).

Important Update

After the release of v5.2.5, we found some backwards compatibility issues with the stylesheet changes in #7039 and #6910. We have therefore decided to make a rapid bug fix Release 5.2.5 to resolve these issues, and all users should upgrade to the new version.

Major Improvements

New Twitter Archivist plugin to import the tweets and associated media from a Twitter Archive as individual tiddlers.

added new GenesisWidget that allows the dynamic construction of another widget, where the name and attributes of the new widget can be dynamically determined, without needing to be known in advance

added (and **here**) new operators for reading and formatting JSON data: jsonget Operator, jsonindexes Operator, jsontype Operator, jsonextract Operator and format Operator

Translation Improvements

Improvements to the following translations:

- Chinese
- French
- German
- Polish
- Spanish
- Japanese

Improvements to the translation features of TiddlyWiki itself:

- **improved** the Translators Edition to add an option to display the original English text underneath the text area
- **updated** "delete" button text in **$:/AdvancedSearch** so that it is translatable

Usability Improvements

- **fixed** problem with long presses on tiddler links triggering a preview on iOS/iPadOS
- **improved** consistency of button and input elements across browsers
- **fixed** edit preview to use the View Template Body Cascade
- **fixed** opening a tiddler in a new window to use the View Template Body Cascade
- **improved** detection of infinite recursion errors in widgets and filters
- **extended** default styles for styled runs
- **improved** upgrade wizard to make the version number more prominent
- **improved** parsing of tiddlers containing CSV data for greater compatibility
- **added** new page control button to summon the layout switcher
- **fixed** folded tiddlers to ensure that the unfold button is always visible
- **improved** handling of Modals to optionally allow them to be dismissed by clicking on the background

Widget Improvements

- **fixed** EditTextWidget to remove the default text "Type the text for the tiddler 'foo'"
- **added** **focus** attribute to SelectWidget
- **removed** restriction of the LetWidget being unable to create variables whose names begin with a dollar sign

Filter improvements

- **fixed** issue with availability of variables within filter runs

- **fixed** issue with removing multiple items from a linked list during filter processing

Hackability Improvements

- **added** support for nested macro definitions
- **added** support for SystemTag: $:/tags/ClassFilters/TiddlerTemplate and SystemTag: $:/tags/ClassFilters/PageTemplate to assign dynamic CSS classes to both tiddler frames and the page template
- **improved** safe mode to disable wiki store indexers
- **fixed** safe mode to prevent globally disabling parser rules
- **extended** keyboard shortcut handling to allow to global shortcuts to override all other shortcuts
- **extended** list-links Macro to allow the rendered field to be overriden
- **improved** Table-of-Contents Macros to allow the default icons to be overridden
- **added** **data-tags-*** and **data-tiddler-title** attributes to the edit preview area
- **extended** timeline Macro to override the link template
- **added** support for Unix epoch timestamps in date format strings
- **fixed** the "big green download button" to use the defined palette colour
- **added** new hidden setting to use horizontal tabs for the "more" sidebar tab
- **extended** the external JavaScript templates to allow the URL of the external script file to be configured

Bug Fixes

- **fixed** truncated search results on small screens
- **fixed** table contents overflow on small screens
- **fixed** "new image" keyboard shortcut not to assign journal tags
- **fixed** SelectWidget class to update if it uses a filter
- **fixed** issue with wikification within the advanced search filter dropdown
- **fixed** the table in *$:/Import* to avoid creating hidden empty rows
- **fixed** advanced search keyboard shortcut not navigating correctly
- **fixed** erroneous display of drafts within the advanced search filter dropdown
- **fixed** backwards compatibility of new field editor cascade introduced in v5.2.3

Node.js Improvements

- **added** new CommandsCommand to enable command tokens to be dynamically generated from a filter
- **improved** console logging to avoid spaces and `<empty string>` message
- **fixed** problem with lazy loading deleting tiddler bodies under certain circumstances
- **fixed** handling of ".mp4" file extension so that it defaults to video not audio
- **added** test server to the plugin library edition
- **added** Hidden Setting: Sync Logging to control logging of sync-related messages
- **updated** Jasmine plugin to require the explicit use of the `--test` command in order to cause the tests to be run

Performance Improvements

- **improved** performance of `wiki.getTiddler()`
- **improved** performance of variable prototype chain handling
- **improved** performance of list handling during filter processing

Acknowledgements

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:


@Anthon…

@bestony

@BramC…

@btheado

@carlo-c…

@Evident…

@FlashS…

@flibbles

@fu-sen

@hoelzro

@joebord…

@kookma

@linonetwo

@Marxsal

@oflg

@pmario

@rmunn

@roma0…

@saqimtiaz

@simonb…

@talha131

@Telumire

@tw-FRed

@twMat

@xcazin

Release 5.2.6

20th March 2023 at 7:43pm

ReleaseNotes

See GitHub for detailed change history of this release



Congratulations to StS for their winning design for the banner for this release (here is the competition thread and the voting thread).

Major Improvements

Improved Markdown Plugin

- **updated** the Markdown Plugin to use the newer and better maintained markdown-it library. The previous Markdown plugin remains available as "markdown-legacy"

Better Handling of Loss of Network Connectivity with Client Server Configuration

- **added** integration between the BrowserStorage Plugin and the client-server configuration to allow changes to be made while offline and then later resynchronised with the server

New Diff-Match-Patch Primitives

- **added** three new operators exposing previously hidden features of the Diff-Match-Patch library that is integrated with TiddlyWiki:

- new levenshtein Operator to calculate the similarity of two strings as the number of characters that need to be inserted, deleted or modified in order to turn one into the other
- new makepatches Operator and applypatches Operator that can be used to make and apply patches that represent the difference between two different texts. See the examples

Translation Improvements

Improvements to the following translations:

- Chinese
- German
- Italian
- Polish

Plugin Improvements

- **fixed** Innerwiki Plugin to allow the `<$data>` widget to override existing tiddler fields
- **fixed**, **here** and **here** problem preventing Share Plugin from working properly
- **updated** XLSX Plugin to handle importing numeric fields
- **fixed** Dynannotate plugin crash when using the fake DOM used for static rendering
- **fixed** bug where pasting text into the *CodeMirror* editor also opened an `$:/Import` tiddler
- **improved** documentation for the Railroad Plugin

Accessibility Improvements

- **improved** appearance of save wiki button so that it is accessible to users without colour vision

Usability Improvements

- **improved** consistency of tiddler deletion by allowing missing tiddlers to be 'deleted', which just results in them being closed
- **improved** layout switcher to include an optional icon for each layout
- **removed** improvements to table layout from v5.2.5 that have proved to not be backwards compatible
- **added** support for `$` HTML entity

Widget Improvements

- **extended** EditTextWidget to support `focusSelectFromStart` and `focusSelectFromEnd` attributes to give better control over text selection

Filter improvements

- **extended** encodeuricomponent Operator to encode characters such as `*` that are illegal in Windows filenames
- **added** encodebase64 Operator and decodebase64 Operator

Hackability Improvements

- **extended** the pragma syntax to allow them to be indented with whitespace
- **improved** readability of tag-picker Macro and keyboard-driven-input Macro
- **improved** reliability of list-links-draggable Macro and list-tagged-draggable Macro by using the new GenesisWidget instead of textual substitution
- **improved** external JavaScript core support to make it possible to save an external JS wiki as a standard single file wiki
- **added** a warning message in the browser console when the Zoomin storyview fails due to the tiddler view template lacking a single containing element

Bug Fixes

- **fixed** bug with the DropzoneWidget that prevented all the available formats being pasted
- **updated** PDF rendering to use an `<iframe>` instead of the older `<embed>`
- **fixed** issue with `.tid` files with fieldnames containing hash characters by saving such tiddlers in `.json` format
- **fixed** (and **here**) overriding `toc-caption` macro for Table-of-Contents Macros
- **fixed** rendering of **$:/core/ui/MoreSideBar/Tags** when viewed within the story river
- **fixed** undefined variable crash in reduce Operator, filter Operator and sortsub Operator
- **improved** GenesisWidget not to create anything if the `$type` attribute is blank or missing
- **fixed** GenesisWidget to pass `isBlock` flag to the generated widget
- **fixed** crash using jsonget Operator with "fixed" as key name

- **fixed** importing of MP3 files to match current browser implementations
- **fixed** crash when attempting to create an element with a blank tag name
- **fixed** layout issue with "save tiddler" button
- **fixed** (and **here**) handling of whitespace in lists to cover all Unicode whitespace characters
- **fixed** test spacing of parser rule checkboxes in control panel
- **fixed** CSS classes assigned to edit toolbar buttons
- **fixed** usage of broken CSS class `tc-storyview-zoomin-tiddler` in Zoomin story view
- **fixed** usage of `importTitle` and `autoOpenOnImport` options for WidgetMessage: tm-import-tiddlers
- **fixed** fixed text editor refresh when the palette is changed
- **fixed** crash with LetWidget when referencing a variable that exists but has an undefined value
- **fixed** unwanted double braces around tags specified in *$:/config/NewTiddler/Tags* when using "new here" button

Node.js Improvements

- **fixed** duplicate fields in internal templates used in client-server configuration
- **fixed** lazy loading not triggering a sync from the server
- **fixed** crash on creating a new tiddler if anonymous users manage to create syncable tiddlers in a read only wiki
- **improved** handling of logout in the client-server configuration to avoid 404 errors
- **fixed** problem with saving tiddlers with `_canonical_uri` field as `.tid` files
- **fixed** missing meta viewport to static river template

Performance Improvements

- **improved** field indexer to more efficiently process lookups

Acknowledgements

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:


@andriga…

@Anthon…

@BramC…

@btheado

@cdruan

@CrossEye

@cs8425

@Evident…

@fkmiec

@flibbles

@Game…

@hffqyd

@jeffrey4l

@joebord…

@kookma

@linonetwo

@m42e

@Marxsal

@mateus…

@michsa

@newme…

@pippep

@pmario

@saqimtiaz

@Telumire

@twMat

@wincen…

@yaisog

Release 5.2.7

26th March 2023 at 10:32am

ReleaseNotes

See GitHub for detailed change history of this release



Congratulations to StS for their winning design for the banner for this release (here is the competition thread and the voting thread).

This is a bug fix release intended to resolve a backwards compatibility issue discovered in v5.2.6. See this GitHub ticket for the background.

The issue is that trailing newlines within a macro definitions were being ignored, which affected whether the content were parsed in inline or block mode.

Since v5.2.7 replaces v5.2.6 that was only released for less than a week, here is the release note for v5.2.6.

Release Note for v5.2.6

See GitHub for detailed change history of this release



Congratulations to StS for their winning design for the banner for this release (here is the competition thread and the voting thread).

Major Improvements

Improved Markdown Plugin

- **updated** the Markdown Plugin to use the newer and better maintained markdown-it library. The previous Markdown plugin remains available as "markdown-legacy"

Better Handling of Loss of Network Connectivity with Client Server Configuration

- **added** integration between the BrowserStorage Plugin and the client-server configuration to allow changes to be made while offline and then later resynchronised with the server

New Diff-Match-Patch Primitives

- **added** three new operators exposing previously hidden features of the Diff-Match-Patch library that is integrated with TiddlyWiki:
  - new levenshtein Operator to calculate the similarity of two strings as the number of characters that need to be inserted, deleted or modified in order to turn one into the other
  - new makepatches Operator and applypatches Operator that can be used to make and apply patches that represent the difference between two different texts. See the examples

Translation Improvements

Improvements to the following translations:

- Chinese
- German
- Italian
- Polish

Plugin Improvements

- **fixed** Innerwiki Plugin to allow the `<$data>` widget to override existing tiddler fields
- **fixed**, **here** and **here** problem preventing Share Plugin from working properly
- **updated** XLSX Plugin to handle importing numeric fields
- **fixed** Dynannotate plugin crash when using the fake DOM used for static rendering
- **fixed** bug where pasting text into the *CodeMirror* editor also opened an `$:/Import` tiddler
- **improved** documentation for the Railroad Plugin

Accessibility Improvements

- **improved** appearance of save wiki button so that it is accessible to users without colour vision

Usability Improvements

- **improved** consistency of tiddler deletion by allowing missing tiddlers to be 'deleted', which just results in them being closed
- **improved** layout switcher to include an optional icon for each layout
- **removed** improvements to table layout from v5.2.5 that have proved to not be backwards compatible
- **added** support for `$` HTML entity

Widget Improvements

- **extended** EditTextWidget to support `focusSelectFromStart` and `focusSelectFromEnd` attributes to give better control over text selection

Filter improvements

- **extended** encodeuricomponent Operator to encode characters such as `*` that are illegal in Windows filenames
- **added** encodebase64 Operator and decodebase64 Operator

Hackability Improvements

- **extended** the pragma syntax to allow them to be indented with whitespace
- **improved** readability of tag-picker Macro and keyboard-driven-input Macro
- **improved** reliability of list-links-draggable Macro and list-tagged-draggable Macro by using the new GenesisWidget instead of textual substitution

- **improved** external JavaScript core support to make it possible to save an external JS wiki as a standard single file wiki
- **added** a warning message in the browser console when the Zoomin storyview fails due to the tiddler view template lacking a single containing element

Bug Fixes

- **fixed** bug with the DropzoneWidget that prevented all the available formats being pasted
- **updated** PDF rendering to use an `<iframe>` instead of the older `<embed>`
- **fixed** issue with `.tid` files with fieldnames containing hash characters by saving such tiddlers in `.json` format
- **fixed** (and **here**) overriding `toc-caption` macro for Table-of-Contents Macros
- **fixed** rendering of **$:/core/ui/MoreSideBar/Tags** when viewed within the story river
- **fixed** undefined variable crash in reduce Operator, filter Operator and sortsub Operator
- **improved** GenesisWidget not to create anything if the `$type` attribute is blank or missing
- **fixed** GenesisWidget to pass `isBlock` flag to the generated widget
- **fixed** crash using jsonget Operator with "fixed" as key name
- **fixed** importing of MP3 files to match current browser implementations
- **fixed** crash when attempting to create an element with a blank tag name
- **fixed** layout issue with "save tiddler" button
- **fixed** (and **here**) handling of whitespace in lists to cover all Unicode whitespace characters
- **fixed** test spacing of parser rule checkboxes in control panel
- **fixed** CSS classes assigned to edit toolbar buttons
- **fixed** usage of broken CSS class `tc-storyview-zoomin-tiddler` in Zoomin story view

- **fixed** usage of `importTitle` and `autoOpenOnImport` options for WidgetMessage: tm-import-tiddlers
- **fixed** fixed text editor refresh when the palette is changed
- **fixed** crash with LetWidget when referencing a variable that exists but has an undefined value
- **fixed** unwanted double braces around tags specified in *$:/config/NewTiddler/Tags* when using "new here" button

Node.js Improvements

- **fixed** duplicate fields in internal templates used in client-server configuration
- **fixed** lazy loading not triggering a sync from the server
- **fixed** crash on creating a new tiddler if anonymous users manage to create syncable tiddlers in a read only wiki
- **improved** handling of logout in the client-server configuration to avoid 404 errors
- **fixed** problem with saving tiddlers with `_canonical_uri` field as `.tid` files
- **fixed** missing meta viewport to static river template

Performance Improvements

- **improved** field indexer to more efficiently process lookups

Acknowledgements

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:


@andriga…

@Anthon…

@BramC…

@btheado

@cdruan

@CrossEye

@cs8425

@Evident…

@fkmiec

@flibbles

@Game…

@hffqyd

@jeffrey4l

@joebord…

@kookma

@linonetwo

@m42e

@Marxsal

@mateus…

@michsa

@newme…

@pippep

@pmario

@saqimtiaz

@Telumire

@twMat

@wincen…

@yaisog

Release 5.3.0

1st July 2023 at 2:34pm

ReleaseNotes

See GitHub for detailed change history of this release



Congratulations to vilc for their winning design for the banner for this release (here is the competition thread).

Overview of v5.3.0

This release introduces a number of significant improvements and new features related to some of TiddlyWiki's most fundamental components: macros, widgets, operators and transclusion. v5.3.0 also contains several other bug fixes and improvements.

Introduction to Parameterised Transclusion, Procedures, Functions and Custom Widgets

added The motivation of these changes is to fix one of TiddlyWiki 5's early design flaws: the reliance on macros using textual substitution as the primary way to modularise and reuse wikitext and filters.

Experience has shown that while macros are a good match for a small number of tasks, they are brittle and error prone for many common operations. See Macro Pitfalls for a discussion of the problems that accompany this approach. Over the years we have introduced mitigations for the worst problems but these have come at a cost of increased complexity.

The changes in this release provide powerful new ways to achieve common tasks, and unlock completely new capabilities that were previously impossible in wikitext.

- Procedures, which are essentially what macros should have been; they work in exactly the same way except that parameters are exposed as simple variables (without the double underscores) and no textual substitution takes place
- Custom Widgets, allowing the creation of widgets in wikitext, and the redefinition of built-in widgets
- Functions, a new way to encapsulate filter expressions with named parameters, including the ability to make custom filter operators
- Parameterised Transclusions, allowing strings and wikitext trees to be passed to transclusions

The approach taken by this release is to add new functionality by extending and augmenting the system without disturbing existing functionality. All of these changes are thus intended to be backwards compatible. While they represent a new field of opportunities for wikitext authors, it is possible for authors to ignore all these new features and continue to use TiddlyWiki 5 in the way that they have always done.

These changes lay the groundwork for macros and related features to be deprecated (which is the point at which users are advised not to use old features, and instead given clear pointers to the equivalent modern functionality).

Text Substitution Improvements

added The new transclusion architecture is not by itself sufficient to enable us to fully deprecate macros yet. To handle most of the remaining use cases this release adds convenient new ways of using textual substitution without having to create a macro:

Firstly, the new text substitution syntax for widget attributes allows widget attributes to be assigned the value of a string with certain placeholders being replaced by their processed contents. For example:

- Substitute variable names with the value:

```
attr=`Current tiddler is $(currentTiddler)$`
```

- Substitute filter expressions with the first value they return:

```
attr=```There are ${ [tag[Done]count[]] }$ completed tasks```
```

Secondly, the new substitute operator allows the same textual substitutions to be performed via a filter operator with the addition of positional parameters that use placeholders of the form `$1$`, `$2$`, `$3$` etc.

```
[[https://$1$/$(currentTiddler)$]substitute<domain-name>]
```

HTTP Requests in WikiText

added new WidgetMessage: tm-http-request for performing HTTP requests in WikiText. This opens up some exciting new opportunities:

- Integration with Web-based APIs. The documentation includes an example of using the Zotero API to retrieve academic citation data
- Dynamic content loading: additional tiddlers can be imported dynamically after the main wiki has loaded

Defaulting to Disabling CamelCase Links

updated CamelCase linking is now disabled by default for new wikis. (Note that this documentation wiki has CamelCase linking explicitly enabled because much of the old content was written relying on them).

Plugin Improvements

- **updated** Google Analytics plugin to use new GA4 code. Note that the update requires manual configuration to use the new "measurement ID" instead of the old "account ID"
- **extended** Dynannotate pugin to support three additional search modes
- **fixed** problem with BrowserStorage Plugin unnecessarily saving shadow tiddlers
- **improved** BrowserStorage Plugin to request that browser storage be persisted without eviction

- **improved** CodeMirror Plugin to add an option to make trailing spaces visible

Translation improvement

Improvements to the following translations:

- French
- German
- Polish
- Chinese

Usability Improvements

- **improved** consistency of layout of "Settings" tab in **$:/ControlPanel**

Filter improvements

- **added** format Operator to support converting Unix timestamps to TiddlyWiki's native date format
- **added** new ':then' filter run prefix

Hackability Improvements

- **extended** Core Icons to allow the size to be controlled with a parameter
  - 
    This change can cause problems with non-standard usage of the core icons where the text is directly rendered instead of being transcluded
- **added** new thisTiddler Variable that refers to the tiddler currently being rendered
- **added** `data-tag-title` attribute to all tag pills, allowing easier Custom tag pill styles
- **improved** Story Tiddler Template Cascade handling to fall back to the default template if the output of the cascade is not valid
- **fixed** missing file extensions for "audio/mpeg" files
- **extended** Table-of-Contents Macros to add consistent support for an **exclude** parameter
- **added** tv-config-static Variable for indicating static rendering

Bug Fixes

- **fixed** importing tiddlers by pasting tiddler data
- **fixed** unneeded 14px bottom border for textareas with autoheight enabled
- **fixed** crashes when using an invalid CSS selector for WidgetMessage: tm-focus-selector and WidgetMessage: tm-scroll
- **fixed** bug whereby scrolling occurs if the linkcatcher widget triggers an action-navigate and the $scroll attribute is set to "no"
- **fixed** problem switching between LTR and RTL text
- **fixed** bug when the listField attribute of the CheckboxWidget was given the name of a date field (like **`created`** or **`modified`**)
- **fixed** size of buttons in dropdown for editor "link" toolbar button
- **fixed** crash when transcluding a lazily loaded tiddler as an attribute value
- **fixed** DiffTextWidget crash with missing or empty attributes

Node.js Improvements

- **extended** WebServer Parameter: authenticated-user-header to require URI encoding of authenticated username header, permitting non-ASCII characters in usernames
- **added** support for `filepath` source attribute to tiddlywiki.files Files
- **added** a warning message for JSON errors in tiddlywiki.info Files or plugin.info Files
- **added** new "explodePlugins" option to SaveWikiFolderCommand

Performance Improvements

- **fixed** module execution to reuse the same sandbox, saving memory and improving performance

Acknowledgements

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:


@Anthon…

@Arlen22

@BramC…

@btheado

@buggyj

@carlo-c…

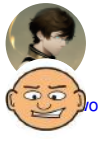
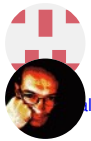
@cdruan

@donmor

@Evident…

@flibbles








@rmunn @saqimtiaz @tavin @twMat @xcazin @yaisog @Zachari… @muzim… @oeyoews @pmario

Release 5.3.1

20th August 2023 at 1:28pm

ReleaseNotes

See GitHub for detailed change history of this release



Congratulations to vilc for their winning design for the banner for this release (here is the competition thread).

Overview of v5.3.1

This release comes only a few weeks after v5.3.0. The motivation for the release is to swiftly fix some issues that have emerged with v5.3.0. There are also some other improvements included in this release, notably the ability to access binary resources over HTTP – the demo downloads a random image or video of a dog.

Bug Fixes and Reversions of v5.3.0 Changes

- Reverted adding the `widget.destroy()` method because of performance concerns (see https://github.com/Jermolene/TiddlyWiki5/pull/7468)
- **fixed** inefficiency when transcluding with the **$output** attribute set to `text/plain` that manifested itself as extremely slow export times
- **fixed** unwanted error message "Global assignment is not allowed within modules on node"

Translation improvement

Improvements to the following translations:

- Chinese
- Polish

Widget Improvements

- **added** **tabindex** attribute to SelectWidget

Hackability Improvements

- **improved** ImportVariablesWidget and Pragma: \import to trim whitespace when parsing tiddlers

- **extended** WidgetMessage: tm-http-request to handle binary responses (demo)

Bug Fixes

- **fixed** table of contents indentation
- **fixed** bindStatus and bindProgress parameters of WidgetMessage: tm-http-request
- **fixed** attribute substitution to handle variables containing non-word characters
- **fixed** the pragmas introduced in v5.3.0 so that they can be indented with whitespace
- **fixed** size of tiddler icons
- **fixed** drag and drop from Chrome-like browsers to Firefox
- **fixed** listIndex mode of checkbox widgets

Node.js Improvements

- **improved** console reporting of JavaScript errors

Developer Improvements

- **fixed** overeager onload handler in Jasmine plugin
- **fixed** ordering of shadow tiddler listings to not reflect order of insertion

Acknowledgements

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:


@Anthon…

@btheado

@catter-fly

@cmo-po…

@CrossEye

@flibbles

@hffqyd

@lilscribby

@linonetwo

@Marxsal

@mateus…

@pille1842

@pmario

@rmunn

@saqimtiaz

@steves…

@TiddlyT…

@twMat

@yaisog

Release 5.3.2

13th December 2023 at 9:06am

ReleaseNotes

See GitHub for detailed change history of this release



Congratulations to catter-fly for their winning design for the banner for this release (here is the competition thread).

Major Improvements

Conditional Shortcut Syntax

added a new shortcut syntax for concisely expressing if-then-else logic. This is the first of a new type of wikitext syntax based on tokens delimited with `<%` and `%>`. We plan to introduce other structures using the same format such as a "case" statement.

These new token-based shortcuts allow a richer structure and expressivity than existing features such as widgets or pragmas. For example:

```
<% if [<animal>match[Elephant]] %>
  It is an elephant
<% elseif [<animal>match[Giraffe]] %>
    It is a giraffe
<% else %>
    It is completely unknown
<% endif %>
```

Behind the scenes, the conditional shortcut syntax is rendered as the equivalent ListWidgets.

Explicit Templates for the ListWidget

added support for `<$list-template>` and `<$list-empty>` as immediate children of the **$**`ListWidget` widget to specify the list item template and/or the empty template.

This new feature is designed to replace a common pattern of using the `emptyMessage` attribute of the ListWidget to render complex wikitext

that thus has to be quoted. Working with wikitext within quotes is awkward and error prone. The new structure can be somewhat faster because it allows the empty message to be parsed in advanced of rendering.

For example:

```
<$list filter=<<filter>>>
    <$list-template>
        <$text text=<<currentTiddler>>/>
    </$list-template>
    <$list-empty>
        None!
    </$list-empty>
</$list>
```

Note that the **emptyMessage** and **template** attributes take precedence if they are present.

Joiners for the ListWidget

added a **join** attribute to the **$**ListWidget widget to insert a short piece of text between list items. This is both easier to use and faster than using the **counter** attribute for the same purpose. So if your list looked like this:

```
<$list filter=<<filter>> counter="counter" variable="item">
<$text text=<<item>>/><$list filter="[<counter-last>match[no]]" variable="ignore"><$text text=", "/></
$list>
</$list>
```

You can replace it with:

```
<$list filter=<<filter>> variable="item" join=", "><$text text=<<item>>/></$list>
```

If the joiner text that you need is long and awkward to write in an attribute, you can use the new `<$list-join>` widget. Like `<$list-template>` and `<$list-empty>`, it must be an immediate child of the **$**`ListWidget`:

```
<$list filter=<<filter>> variable="item"><$text text=<<item>>/><$list-join>, and <em>also</em> let's not forget </$list-join></$list>
```

jsonset operator

added jsonset Operator for setting values within JSON objects

QR Code Reader

extended QR Code plugin to be able to read QR codes and a number of other bar code formats

Translation improvements

Improvements to the following translations:

- Chinese
- Polish

- Spanish

Plugin Improvements

- **fixed** Comments Plugin to use predefined palette colours
- **improved** Evernote Importer Plugin to support images and other attachments
- **added** `$floating` attribute to Dynannotate Plugin to support popups that do not disappear when another part of the screen is clicked. Instead they have to dismissed manually

Widget Improvements

- **fixed** ImageWidget encoding for more image types
- **extended** ImageWidget to add a "usemap" attribute
- **improved** the ScrollableWidget to allow the scroll position to be bound to a tiddler, so that changes to the tiddler affect the scroll position, and vice versa

Usability Improvements

- **updated** editor preview button to automatically focus the editor
- **improved** file type names in the export menu

Hackability Improvements

- **extended** all the relevant core widgets to allow arbitrary `data-*` attributes and `style.*` attributes to be applied to the generated DOM nodes. This is useful for passing data to the EventCatcherWidget
- **extended** jsonextract Operator, jsonget Operator, jsonset Operator and jsontype Operator to allow negative indexes into arrays to be counted from the end of the array
- **improved** the default page layout to better support CSS grid and flexbox layouts
- **improved** the editor to use grid layout, simplifying customisation

Bug Fixes

- **fixed** ordering of Vanilla stylesheets
- **fixed** missing closing tag in tag-pill-inner macro
- **removed** invalid "type" attribute from textarea elements generated by the EditTextWidget
- **fixed** editor "type" dropdown state tiddlers
- **fixed** handling of "counter-last" variable when appending items with the ListWidget
- **fixed** upgrade download link in Firefox
- **fixed** refreshing of transcluded functions
- **fixed** resizing of height of textareas in control panel
- **fixed** encodebase64 Operator and decodebase64 Operator to work properly with binary data
- **fixed** WidgetMessage: tm-open-window when opening an existing window to bring it to the front and focus it

- **fixed** behaviour of last Operator when zero items selected
- **fixed** incorrectly setting focus on field name input field when deleting field using the delete field button
- **fixed** Table-of-Contents Macros to not show expander icon for a sublist that has all children excluded
- **fixed** overflow of CodeMirror Plugin editor within grid container
- **fixed** wikitest parser removing whitespace when parsing pragmas
- **fixed** tooltip for editor add field button
- **fixed** plain text parser being susceptible to the CodeBlockWidget being redefined
- **fixed** pragmas not working within the action string of several core macros

Node.js Improvements

- **fixed** a significant flaw in the synchronisation algorithm used by the client-server configuration. The flaw could lead to tiddlers temporarily disappearing from the browser

Performance Improvements

- **improved** performance of predefined patterns with all Operator
- **updated** favicon format to PNG

Developer Improvements

- **improved** global hook handling to support removing hooks
- **added** some useful npm scripts to `package.json`

Infrastructure Improvements

- **improved** Continuous Integration tests to use Playwright to run our browser-based tests
- **added** an automatic build of the external core TiddlyWiki at https://tiddlywiki.com/empty-external-core.html

Acknowledgements

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:


@Anthon…

@BramC…

@Buckar…

@Burnin…

@CrossEye

@Evident…

@Gk0Wk

@joebord…

@kookma

@linonetwo

@mateus…

@oflg

@pille1842

@pmario

@rmunn

@saqimtiaz

@simonb…

@T1mL3…

@yaisog

Release 5.3.3

23rd December 2023 at 11:22am

ReleaseNotes

See GitHub for detailed change history of this release



Congratulations to catter-fly for their winning design for the banner for this release (here is the competition thread).

This is a bug fix release to address a number of bugs that were introduced with Release 5.3.2.

- **fixed** handling of a list widget with an empty paragraph as inline template
- **fixed** broken per-tiddler previews
- **fixed** missing comma before skinny tiddlers in JSON store area
- **fixed** handling of whitespace immediately after pragmas
- **fixed** SelectWidget handling of classes and rendering typo

Since v5.3.3 replaces v5.3.2 after only a couple of weeks, here is the release note for v5.3.2.

Release Note for v5.3.2

See GitHub for detailed change history of this release



Congratulations to catter-fly for their winning design for the banner for this release (here is the competition thread).

Major Improvements

Conditional Shortcut Syntax

added a new shortcut syntax for concisely expressing if-then-else logic. This is the first of a new type of wikitext syntax based on tokens delimited with `<%` and `%>`. We plan to introduce other structures using the same format such as a "case" statement.

These new token-based shortcuts allow a richer structure and expressivity than existing features such as widgets or pragmas. For example:

```
<% if [<animal>match[Elephant]] %>
  It is an elephant
<% elseif [<animal>match[Giraffe]] %>
    It is a giraffe
<% else %>
    It is completely unknown
<% endif %>
```

Behind the scenes, the conditional shortcut syntax is rendered as the equivalent ListWidgets.

Explicit Templates for the ListWidget

added support for `<$list-template>` and `<$list-empty>` as immediate children of the **$**`ListWidget` widget to specify the list item template and/or the empty template.

This new feature is designed to replace a common pattern of using the `emptyMessage` attribute of the ListWidget to render complex wikitext that thus has to be quoted. Working with wikitext within quotes is awkward and error prone. The new structure can be somewhat faster because it allows the empty message to be parsed in advanced of rendering.

For example:

```
<$list filter=<<filter>>>
    <$list-template>
        <$text text=<<currentTiddler>>/>
    </$list-template>
    <$list-empty>
        None!
    </$list-empty>
</$list>
```

Note that the **`emptyMessage`** and **`template`** attributes take precedence if they are present.

Joiners for the ListWidget

added a **`join`** attribute to the **$**`ListWidget` widget to insert a short piece of text between list items. This is both easier to use and faster than using the **`counter`** attribute for the same purpose. So if your list looked like this:

```
<$list filter=<<filter>> counter="counter" variable="item">
```

```
<$text text=<<item>>/><$list filter="[<counter-last>match[no]]" variable="ignore"><$text text=", "/></
$list>
</$list>
```

You can replace it with:

```
<$list filter=<<filter>> variable="item" join=", "><$text text=<<item>>/></$list>
```

If the joiner text that you need is long and awkward to write in an attribute, you can use the new `<$list-join>` widget. Like `<$list-template>` and `<$list-empty>`, it must be an immediate child of the **$**`ListWidget`:

```
<$list filter=<<filter>> variable="item"><$text text=<<item>>/><$list-join>, and <em>also</em> let's not
forget </$list-join></$list>
```

jsonset operator

added jsonset Operator for setting values within JSON objects

QR Code Reader

extended QR Code plugin to be able to read QR codes and a number of other bar code formats

Translation improvements

Improvements to the following translations:

- Chinese
- Polish
- Spanish

Plugin Improvements

- **fixed** Comments Plugin to use predefined palette colours
- **improved** Evernote Importer Plugin to support images and other attachments
- **added** `$floating` attribute to Dynannotate Plugin to support popups that do not disappear when another part of the screen is clicked. Instead they have to dismissed manually

Widget Improvements

- **fixed** ImageWidget encoding for more image types
- **extended** ImageWidget to add a "usemap" attribute
- **improved** the ScrollableWidget to allow the scroll position to be bound to a tiddler, so that changes to the tiddler affect the scroll position, and vice versa

Usability Improvements

- **updated** editor preview button to automatically focus the editor

- **improved** file type names in the export menu

Hackability Improvements

- **extended** all the relevant core widgets to allow arbitrary `data-*` attributes and `style.*` attributes to be applied to the generated DOM nodes. This is useful for passing data to the EventCatcherWidget
- **extended** jsonextract Operator, jsonget Operator, jsonset Operator and jsontype Operator to allow negative indexes into arrays to be counted from the end of the array
- **improved** the default page layout to better support CSS grid and flexbox layouts
- **improved** the editor to use grid layout, simplifying customisation

Bug Fixes

- **fixed** ordering of Vanilla stylesheets
- **fixed** missing closing tag in tag-pill-inner macro
- **removed** invalid "type" attribute from textarea elements generated by the EditTextWidget
- **fixed** editor "type" dropdown state tiddlers
- **fixed** handling of "counter-last" variable when appending items with the ListWidget
- **fixed** upgrade download link in Firefox
- **fixed** refreshing of transcluded functions
- **fixed** resizing of height of textareas in control panel

- **fixed** encodebase64 Operator and decodebase64 Operator to work properly with binary data
- **fixed** WidgetMessage: tm-open-window when opening an existing window to bring it to the front and focus it
- **fixed** behaviour of last Operator when zero items selected
- **fixed** incorrectly setting focus on field name input field when deleting field using the delete field button
- **fixed** Table-of-Contents Macros to not show expander icon for a sublist that has all children excluded
- **fixed** overflow of CodeMirror Plugin editor within grid container
- **fixed** wikitest parser removing whitespace when parsing pragmas
- **fixed** tooltip for editor add field button
- **fixed** plain text parser being susceptible to the CodeBlockWidget being redefined
- **fixed** pragmas not working within the action string of several core macros

Node.js Improvements

- **fixed** a significant flaw in the synchronisation algorithm used by the client-server configuration. The flaw could lead to tiddlers temporarily disappearing from the browser

Performance Improvements

- **improved** performance of predefined patterns with all Operator
- **updated** favicon format to PNG

Developer Improvements

- **improved** global hook handling to support removing hooks
- **added** some useful npm scripts to `package.json`

Infrastructure Improvements

- **improved** Continuous Integration tests to use Playwright to run our browser-based tests
- **added** an automatic build of the external core TiddlyWiki at https://tiddlywiki.com/empty-external-core.html

Acknowledgements

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:


@Anthon…

@BramC…

@Buckar…

@Burnin…

@CrossEye

@Evident…

@Gk0Wk

@joebord…

@kookma

@linonetwo

@mateus…

@oflg

@pille1842

@pmario

@rmunn

@saqimtiaz

@simonb…

@T1mL3…

@yaisog

Releases

5th October 2023 at 10:56pm

About

New releases of TiddlyWiki and TiddlyDesktop are announced via the official discussion groups and Twitter

See the TiddlyWiki Archive to download older versions.

TiddlyWiki | TiddlyDesktop

Here are the details of recent releases of TiddlyWiki5. See TiddlyWiki5 Versioning for details of how releases are named.

If you are using node.js, you can also install prior versions like this:

> npm install -g tiddlywiki@5.1.13

(BetaReleases and AlphaReleases are listed separately).

5.3.3
5.3.2
5.3.1
5.3.0
5.2.7
5.2.6
5.2.5
5.2.4
5.2.3
5.2.2
5.2.1
5.2.0
5.1.23
5.1.22
5.1.21
5.1.20
5.1.19
5.1.18

Release 5.3.3

Released 23rd December 2023 at 11:22

See GitHub for detailed change history of this release



Congratulations to catter-fly for their winning design for the banner for this release (here is the competition thread).

This is a bug fix release to address a number of bugs that were introduced with Release 5.3.2.

- **fixed** handling of a list widget with an empty paragraph as inline template
- **fixed** broken per-tiddler previews
- **fixed** missing comma before skinny tiddlers in JSON store area
- **fixed** handling of whitespace immediately after pragmas
- **fixed** SelectWidget handling of classes and rendering typo

Since v5.3.3 replaces v5.3.2 after only a couple of weeks, here is the release note for v5.3.2.

Release Note for v5.3.2

5.1.17
5.1.16
5.1.15
5.1.14
5.1.13
5.1.12
5.1.11
5.1.10
5.1.9
5.1.8
5.1.7
5.1.6
5.1.5
5.1.4
5.1.3
5.1.2
5.1.1
5.1.0

See GitHub for detailed change history of this release



Congratulations to catter-fly for their winning design for the banner for this release (here is the competition thread).

Major Improvements

Conditional Shortcut Syntax

added a new shortcut syntax for concisely expressing if-then-else logic. This is the first of a new type of wikitext syntax based on tokens delimited with `<%` and `%>`. We plan to introduce other structures using the same format such as a "case" statement.

These new token-based shortcuts allow a richer structure and expressivity than existing features such as widgets or pragmas. For example:

```
<% if [<animal>match[Elephant]] %>
  It is an elephant
<% elseif [<animal>match[Giraffe]] %>
    It is a giraffe
<% else %>
    It is completely unknown
<% endif %>
```

Behind the scenes, the conditional shortcut syntax is rendered as the equivalent ListWidgets.

Explicit Templates for the ListWidget

added support for `<$list-template>` and `<$list-empty>` as immediate children of the **$**`ListWidget` widget to specify the list item template and/or the empty template.

This new feature is designed to replace a common pattern of using the `emptyMessage` attribute of the ListWidget to render complex wikitext that thus has to be quoted. Working with wikitext within quotes is awkward and error prone. The new structure can be somewhat faster because it allows the empty message to be parsed in advanced of rendering.

For example:

```
<$list filter=<<filter>>>
    <$list-template>
        <$text text=<<currentTiddler>>/>
    </$list-template>
    <$list-empty>
        None!
    </$list-empty>
</$list>
```

Note that the **`emptyMessage`** and **`template`** attributes take precedence if they are present.

Joiners for the ListWidget

added a **`join`** attribute to the **$**`ListWidget` widget to insert a short piece of text between list items. This is both easier to use and faster than using the **`counter`** attribute for the same purpose. So if your list looked like this:

```
<$list filter=<<filter>> counter="counter" variable="item">
<$text text=<<item>>/><$list filter="[<counter-last>match[no]]" variable="ignore"><$text text=",
"/></$list>
</$list>
```

You can replace it with:

```
<$list filter=<<filter>> variable="item" join=", "><$text text=<<item>>/></$list>
```

If the joiner text that you need is long and awkward to write in an attribute, you can use the new `<$list-join>` widget. Like `<$list-template>` and `<$list-empty>`, it must be an immediate child of the **$**`ListWidget`:

```
<$list filter=<<filter>> variable="item"><$text text=<<item>>/><$list-join>, and <em>also</em>
let's not forget </$list-join></$list>
```

jsonset operator

added jsonset Operator for setting values within JSON objects

QR Code Reader

extended QR Code plugin to be able to read QR codes and a number of other bar code formats

Translation improvements

Improvements to the following translations:

- Chinese
- Polish
- Spanish

Plugin Improvements

- **fixed** Comments Plugin to use predefined palette colours
- **improved** Evernote Importer Plugin to support images and other attachments
- **added** `$floating` attribute to Dynannotate Plugin to support popups that do not disappear when another part of the screen is clicked. Instead they have to dismissed manually

Widget Improvements

- **fixed** ImageWidget encoding for more image types
- **extended** ImageWidget to add a "usemap" attribute
- **improved** the ScrollableWidget to allow the scroll position to be bound to a tiddler, so that changes to the tiddler affect the scroll position, and vice versa

Usability Improvements

- **updated** editor preview button to automatically focus the editor
- **improved** file type names in the export menu

Hackability Improvements

- **extended** all the relevant core widgets to allow arbitrary `data-*` attributes and `style.*` attributes to be applied to the generated DOM nodes. This is useful for passing data to the EventCatcherWidget
- **extended** jsonextract Operator, jsonget Operator, jsonset Operator and jsontype Operator to allow negative indexes into arrays to be counted from the end of the array
- **improved** the default page layout to better support CSS grid and flexbox layouts
- **improved** the editor to use grid layout, simplifying customisation

Bug Fixes

- **fixed** ordering of Vanilla stylesheets
- **fixed** missing closing tag in tag-pill-inner macro
- **removed** invalid "type" attribute from textarea elements generated by the EditTextWidget
- **fixed** editor "type" dropdown state tiddlers
- **fixed** handling of "counter-last" variable when appending items with the ListWidget

- **fixed** upgrade download link in Firefox
- **fixed** refreshing of transcluded functions
- **fixed** resizing of height of textareas in control panel
- **fixed** encodebase64 Operator and decodebase64 Operator to work properly with binary data
- **fixed** WidgetMessage: tm-open-window when opening an existing window to bring it to the front and focus it
- **fixed** behaviour of last Operator when zero items selected
- **fixed** incorrectly setting focus on field name input field when deleting field using the delete field button
- **fixed** Table-of-Contents Macros to not show expander icon for a sublist that has all children excluded
- **fixed** overflow of CodeMirror Plugin editor within grid container
- **fixed** wikitest parser removing whitespace when parsing pragmas
- **fixed** tooltip for editor add field button
- **fixed** plain text parser being susceptible to the CodeBlockWidget being redefined
- **fixed** pragmas not working within the action string of several core macros

Node.js Improvements

- **fixed** a significant flaw in the synchronisation algorithm used by the client-server configuration. The flaw could lead to tiddlers temporarily disappearing from the browser

Performance Improvements

- **improved** performance of predefined patterns with all Operator
- **updated** favicon format to PNG

Developer Improvements

- **improved** global hook handling to support removing hooks
- **added** some useful npm scripts to `package.json`

Infrastructure Improvements

- **improved** Continuous Integration tests to use Playwright to run our browser-based tests
- **added** an automatic build of the external core TiddlyWiki at https://tiddlywiki.com/empty-external-core.html

Acknowledgements

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:


@Anthon…

@BramC…

@Buckar…

@Burnin…

@CrossEye

@Evident…

@Gk0Wk

@joebord…

@kookma












'two @mateus… @oflg @pille1842 @pmario @rmunn @saqimtiaz @simonb… @T1mL3…

@yaisog

ReleaseTemplate

20th November 2021 at 12:11am

Release Template

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

ReleaseTemplate

Released TBA

Recursive transclusion error in transclude widget

Released Recursive transclusion error in transclude widget

Recursive transclusion error in transclude widget

Recursive transclusion error in transclude widget

Recursive transclusion error in transclude widget

Recursive transclusion error in transclude widget

Recursive transclusion error in transclude widget

remainder Operator

11th June 2019 at 3:01pm

Binary Mathematics Operators Filter Operators Mathematics Operators

purpose	treating each input title as a number, return the remainder when divided by the numeric value of the parameter
input	a selection of titles
parameter	`N` = a number
output	the input as numbers, but with each replaced by the remainder when dividing it by `N`

Learn more about how to use Filters

Introduced in v5.1.20 See Mathematics Operators for an overview.

Examples

remainder Operator (Examples)

11th June 2019 at 3:01pm

Operator Examples remainder Operator

```
[[23]remainder[3]]
```

Try it

```
=1 =2 =3 =4 +[remainder[2]]
```

Try it

remove Operator

25th January 2017 at 9:00pm

Filter Operators Listops Operators Order Operators

purpose	remove a list of titles specified in the parameter from the input
input	a list of items
suffix	an integer N, defaulting to all
parameter	**`array`** = an array of items to remove
output	items removed from current list that appear at the head of the parameter array
! output	items removed from current list that appear at the tail of the parameter array

Learn more about how to use Filters

Examples

remove Operator (Examples)

8th November 2015 at 6:20am

Operator Examples remove Operator

These examples make use of the Days of the Week tiddler.

```
[list[Days of the Week]] +[remove[Tuesday Wednesday Thursday]]
```

Try it

```
[list[Days of the Week]] +[remove:3[Monday Wednesday Friday Saturday]]
```

Try it

```
[list[Days of the Week]] +[!remove:2{Days of the Week!!list}]
```

Try it

removeprefix Operator

11th July 2023 at 10:28am

Filter Operators String Operators

purpose	filter the input titles by how they start, deleting that prefix
input	a selection of titles
suffix	the **`removeprefix`** operator uses a rich suffix, see below for details
parameter	**`S`** = a string of characters
output	those input titles that start with **`S`**, but with those characters discarded

Learn more about how to use Filters



This filters out input titles that do not start with S. For removing S without filtering out input titles that don't start with S, see trim.

Introduced in v5.2.2

The **removeprefix** operator uses an extended syntax that permits multiple flags to be passed:

```
[removeprefix:<flag list>[<parameter>]]
```

- **flag list**: a comma delimited list of flags

The available flags are:

- **casesensitive**: (default), this flag forces a case-sensitive match, where upper and lower case letters are considered different
- **caseinsensitive**: overrides the default so that upper and lower case letters are considered identical for matching purposes

Examples

removeprefix Operator (Examples)

18th February 2022 at 3:34am

Operator Examples removeprefix Operator

```
[[My Cat]] [[Your Garden]] [[My Favourite Armchair]] +[removeprefix[My ]]
```

Try it

```
[[My Cat]] [[Your Garden]] [[My Favourite Armchair]] +[removeprefix:caseinsensitive[my ]]
```

Try it

removesuffix Operator

11th July 2023 at 10:28am

Filter Operators String Operators

purpose	filter the input titles by how they end, deleting that suffix
input	a selection of titles
suffix	the **`removesuffix`** operator uses a rich suffix, see below for details
parameter	**`S`** = a string of characters
output	those input titles that end with **`S`**, but with those characters discarded

Learn more about how to use Filters



This filters out input titles that do not end with S. For removing S without filtering out input titles that don't end with S, see trim.

ⓘ Introduced in v5.2.2

The **`removesuffix`** operator uses an extended syntax that permits multiple flags to be passed:

```
[removesuffix:<flag list>[<parameter>]]
```

- **flag list**: a comma delimited list of flags

The available flags are:

- **casesensitive**: (default), this flag forces a case-sensitive match, where upper and lower case letters are considered different
- **caseinsensitive**: overrides the default so that upper and lower case letters are considered identical for matching purposes

Examples

removesuffix Operator (Examples)

18th February 2022 at 3:34am

Operator Examples removesuffix Operator

```
SIMPLEX Googolplex Complex +[removesuffix[plex]]
```

Try it

```
SIMPLEX Googolplex Complex +[removesuffix:caseinsensitive[plex]]
```

Try it

RenderCommand

19th September 2017 at 3:18pm

Commands

Render individual tiddlers identified by a filter and save the results to the specified files.

Optionally, the title of a template tiddler can be specified. In this case, instead of directly rendering each tiddler, the template tiddler is rendered with the "currentTiddler" variable set to the title of the tiddler that is being rendered.

A name and value for an additional variable may optionally also be specified.

```
--render <tiddler-filter> [<filename-filter>] [<render-type>] [<template>] [ [<name>] [<value>] ]*
```

- **tiddler-filter**: A filter identifying the tiddler(s) to be rendered
- **filename-filter**: Optional filter transforming tiddler titles into pathnames. If omitted, defaults to `[is[tiddler]addsuffix[.html]]`, which uses the unchanged tiddler title as the filename
- **render-type**: Optional render type: `text/html` (the default) returns the full HTML text and `text/plain` just returns the text content (ie it ignores HTML tags and other unprintable material)
- **template**: Optional template through which each tiddler is rendered
- **name**: Name of optional variables
- **value**: Value of optional variables

By default, the filename is resolved relative to the `output` subdirectory of the edition directory. The `--output` command can be used to direct output to a different directory.

Notes:

- The output directory is not cleared of any existing files
- Any missing directories in the path to the filename are automatically created.
- When referring to a tiddler with spaces in its title, take care to use both the quotes required by your shell and also TiddlyWiki's double

square brackets: `--render "[[Motovun Jack.jpg]]"`

- The filename filter is evaluated with the selected items being set to the title of the tiddler currently being rendered, allowing the title to be used as the basis for computing the filename. For example `[encodeuricomponent[]addprefix[static/]]` applies URI encoding to each title, and then adds the prefix `static/`
- Multiple **name**/**value** pairs can be used to pass more than one variable
- The `--render` command is a more flexible replacement for both the `--rendertiddler` and `--rendertiddlers` commands, which are deprecated

Examples:

- `--render '[!is[system]]' '[encodeuricomponent[]addprefix[tiddlers/]addsuffix[.html]]'` – renders all non-system tiddlers as files in the subdirectory "tiddlers" with URL-encoded titles and the extension HTML
- `--render '.' 'tiddlers.json' 'text/plain' '$:/core/templates/exporters/JsonFile' 'exportFilter' '[tag[HelloThere]]'` – renders the tiddlers tagged "HelloThere" to a JSON file named "tiddlers.json"

Show available rendering templates

RenderTiddlerCommand

Commands

DEPRECATED SINCE: 5.1.15 (see RenderCommand).

(Note: The `--rendertiddler` command is deprecated in favour of the new, more flexible `--render` command)

Render an individual tiddler as a specified ContentType, defaulting to `text/html` and save it to the specified filename.

Optionally the title of a template tiddler can be specified, in which case the template tiddler is rendered with the "currentTiddler" variable set to the tiddler that is being rendered (the first parameter value).

A name and value for an additional variable may optionally also be specified.

```
--rendertiddler <title> <filename> [<type>] [<template>] [<name>] [<value>]
```

By default, the filename is resolved relative to the `output` subdirectory of the edition directory. The `--output` command can be used to direct output to a different directory.

Any missing directories in the path to the filename are automatically created.

For example, the following command saves all tiddlers matching the filter `[tag[done]]` to a JSON file titled `output.json` by employing the core template `$:/core/templates/exporters/JsonFile`.

```
--rendertiddler "$:/core/templates/exporters/JsonFile" output.json text/plain "" exportFilter
"[tag[done]]"
```

RenderTiddlersCommand

Commands

DEPRECATED SINCE: 5.1.15 (see RenderCommand).

(Note: The `--rendertiddlers` command is deprecated in favour of the new, more flexible `--render` command)

Render a set of tiddlers matching a filter to separate files of a specified ContentType (defaults to `text/html`) and extension (defaults to `.html`).

```
--rendertiddlers '<filter>' <template> <pathname> [<type>] [<extension>] ["noclean"]
```

For example:

```
--rendertiddlers '[!is[system]]' $:/core/templates/static.tiddler.html ./static text/plain
```

By default, the pathname is resolved relative to the `output` subdirectory of the edition directory. The `--output` command can be used to direct output to a different directory.

Any files in the target directory are deleted unless the **noclean** flag is specified. The target directory is recursively created if it is missing.

replace Operator

8th November 2015 at 6:21am

Filter Operators Listops Operators Order Operators

purpose	replace marker with N trailing items
input	a list of items
suffix	an integer N, defaulting to 1
parameter	**`marker`** = the item to be used as a marker
output	re-ordered list of items

Learn more about how to use Filters

Examples

replace Operator (Examples)

8th November 2015 at 6:21am

Operator Examples replace Operator

These examples make use of the Days of the Week tiddler.

```
[list[Days of the Week]] Last +[replace[Wednesday]]
```

Try it

```
[list[Days of the Week]] Last +[replace:2[Tuesday]]
```

Try it

```
[list[Days of the Week]] [[Yesterday]] [[Today]] [[Tomorrow]] +[replace:3[Tuesday]]
```

Try it

ReportingBugs

10th September 2014 at 11:25pm

Community

You can report bugs or problems with TiddlyWiki via our discussion groups. If you have a GitHub account then you can raise an issue there:

https://github.com/Jermolene/TiddlyWiki5/issues/new

Unless you are already familiar with GitHub, it's usually easiest to report problems through the discussion groups.

TiddlyWiki on GitHub

We use GitHub Issues to manage bug reports and feature requests for TiddlyWiki. To maintain their effectiveness we endeavour to have as few open issues as possible.

Policies for Managing Issues

Open issues should be actionable: generally either a reproducible bug report, or a specific feature request. From the perspective of the core developers, the issues list behaves like a shared todo list. Every item on the list requires a little bit of attention each time we check the list.

GitHub Issues are not very good for managing ideas that are not immediately actionable. Better to use the TiddlyWiki discussion groups for open ended questions, or speculative discussions of new features.

Creating Issues

Before creating a GitHub issue it is good etiquette to search through the existing issues to see whether the problem has already been reported. If a search isn't practical, don't worry too much; GitHub makes it easy to merge existing issues.

When you do create an issue, remember that for effective debugging, we need as much information as possible. At a minimum, please try to include:

- A descriptive title
- A summary
- Steps to reproduce
- Expected behaviour
- Context (OS, browser etc.)

Consider also adding screenshots if it makes things clearer.

There's a lot of good material on the web about bug reports:

- http://mhay68.tumblr.com/post/1648223018/what-makes-a-good-bug-report
- http://www.chiark.greenend.org.uk/~sgtatham/bugs.html

resolvepath Macro

21st February 2015 at 11:38pm

Core Macros Macros

The **resolvepath** macro takes a relative path and an absolute path. It interprets the former relative to the latter, and returns the absolute equivalent of the former.

Folders in the paths are delimited by `/`.

The special folder name `.` denotes the current folder, and `..` denotes the parent folder.

If the absolute path indicates a folder, it needs to end with `/`. Anything after the final `/` is treated as an arbitrary filename within the intended path, and is discarded.

If no absolute path is supplied, the relative path is returned unchanged, except that anything after the final `/` is discarded.

The return value is a path, but does *NOT* end with `/`.

Parameters

source
: the relative path

root
: the absolute path

Examples

resolvepath Macro (Examples)

21st February 2015 at 7:34pm

Macro Examples resolvepath Macro

copy to clipboard

```
<<resolvepath "./backup" "http://example.com/store.php">>
```

Try it

copy to clipboard

```
<<resolvepath "backup" "http://example.com/store.php">>
```

Try it

copy to clipboard

```
<<resolvepath "../backup" "http://example.com/store.php">>
```

Try it

copy to clipboard

```
<<resolvepath "../jpg/Motovun_Jack.jpg" "http://example.com/resources/images/png/">>
```

Try it

copy to clipboard

```
<<resolvepath "../jpg/../png/Motovun_Jack.png">>
```

Try it

copy to clipboard

```
<<resolvepath "jpg/Motovun_Jack.jpg">>
```

Try it

rest Operator

3rd February 2015 at 8:08pm

Filter Operators Order Operators

purpose	discard the first `N` input titles
input	a selection of titles
parameter	`N` = an integer, defaulting to 1
output	all but the first `N` input titles

Learn more about how to use Filters

`butfirst` and **`bf`** are synonyms for **`rest`**.

Examples

rest Operator (Examples)

26th February 2022 at 5:33am

Operator Examples rest Operator

These examples make use of the Days of the Week tiddler.

```
[list[Days of the Week]rest[]]
```

Try it

```
[list[Days of the Week]rest[0]]
```

Try it

```
[list[Days of the Week]rest[3]]
```

Try it

```
Z Y X W V U T S R Q P O +[rest[5]]
```

Try it

Résumé Builder Edition

2nd June 2016 at 7:15am

Editions

The **Résumé Builder** edition of TiddlyWiki gets you started with a simple way to create a professional resume that you can save to the web or print.

https://tiddlywiki.com/editions/resumebuilder/

RevealWidget

21st November 2020 at 11:09am

Widgets

Introduction

The reveal widget hides or shows its content depending upon the value of a *state tiddler*. The type of the widget determines the condition for the content being displayed:

- type=**match**: the content is displayed if the state tiddler matches the text attribute value
- type=**nomatch**: the content is displayed if the state tiddler doesn't match the text attribute value

- type=**popup**: the content is displayed as a popup as described in the PopupMechanism
- type=**lt**: the content is displayed if the state tiddler contains an integer with a value **less than** the text attribute value
- type=**gt**: the content is displayed if the state tiddler contains an integer with a value **greater than** the text attribute value
- type=**lteq**: the content is displayed if the state tiddler contains an integer with a value **less than or equal to** the text attribute value
- type=**gteq**: the content is displayed if the state tiddler contains an integer with a value **greater than or equal to** the text attribute value

Content and Attributes

The content of the `<$reveal>` widget is displayed according to the rules given above.

Attribute	Description
state	A TextReference containing the state
stateTitle	A title containing the state, **without** TextReference. Gets preferred over the **`state`** attribute
stateField	A **field name** which is used to look for the state, if the attribute **`stateTitle`** is present
stateIndex	An **index** which is used to look for the state, if the attribute **`stateTitle`** is present
tag	Overrides the default HTML element tag (`<div>` in block mode or `<span>` in inline mode)
type	The type of matching performed: **match**, **nomatch**, **popup**, **lt**, **gt**, **lteq** or **gteq**
text	The text to match when the type is **match**, **nomatch**, **lt**, **gt**, **lteq** or **gteq**
class	An optional CSS class name to be assigned to the HTML element » Set to `tc-popup-keep` to make a popup "sticky", so it won't close when you click inside of it
style	An optional CSS style attribute to be assigned to the HTML element
position	The position used for the popup when the type is **popup**. Can be **left**, **above**, **aboveleft**, **aboveright**, **right**, **belowleft**, **belowright** or **below**

positionAllowNegative	Set to "yes" to prevent computed popup positions from being clamped to be above zero
default	Default value to use when the state tiddler is missing
animate	Set to "yes" to animate opening and closure (defaults to "no"; requires "retain" to be set to "yes")
retain	Set to "yes" to force the content to be retained even when hidden (defaults to "no")
updatePopupPosition	ⓘ **Introduced in v5.1.23** Set to "yes" to update the popup position when the state tiddler is updated (defaults to "no")

ⓘ **Introduced in v5.1.18** **`stateTitle`**, **`stateField`** and **`stateIndex`** attributes allow specifying Tiddler states **directly**, without interpreting them as TextReferences. This is useful for edge-cases where titles may contain characters that are used to denote Tiddler fields or indices (`!!` , `##`)

Retaining the content when hidden can give poor performance since the hidden content requires refresh processing even though it is not displayed. On the other hand, the content can be revealed much more quickly. Note that setting **animate="yes"** will also require **retain="yes"**

Examples

Simple content reveal

Here's a simple example of showing and hiding content with buttons:

copy to clipboard

```
<$button set="$:/state/SampleReveal1" setTo="show">Show me</$button>
<$button set="$:/state/SampleReveal1" setTo="hide">Hide me</$button>
```

```
<$reveal type="match" state="$:/state/SampleReveal1" text="show">

! This is the revealed content
And this is some text

</$reveal>
```

That renders as:

Show me Hide me

Accordion or Slider

An "accordion" or "slider" is a button that can be used to toggle the display of associated content.

copy to clipboard

```
<$reveal type="nomatch" state="$:/state/SampleReveal2" text="show">

<$button set="$:/state/SampleReveal2" setTo="show">Show me</$button>

</$reveal>
<$reveal type="match" state="$:/state/SampleReveal2" text="show">

<$button set="$:/state/SampleReveal2" setTo="hide">Hide me</$button>
```

```
! This is the revealed content
And this is some text

</$reveal>
```

That renders as:

Show me

Popup

Here is a simple example of a popup built with the RevealWidget:

copy to clipboard

```
<$button popup="$:/SamplePopupState">Pop me up!</$button>

<$reveal type="popup" state="$:/SamplePopupState">
<div class="tc-drop-down">

! This is the popup
And this is some text

</div>
```

```
</$reveal>
```

That renders as:

Pop me up!

How to use text references with field content

Here is a simple example how to use text references with field content to control the RevealWidget. If the field `jeremy` is populated with text `tiddlywiki`, a message will be displayed.

copy to clipboard

```
<$reveal type="match" state="!!jeremy" text="tiddlywiki">
~TiddlyWiki!
</$reveal>
```

That renders as:

TiddlyWiki!

reverse Operator

3rd February 2015 at 8:08pm

Filter Operators Order Operators

purpose	reverse the order of the input titles
input	a selection of titles
parameter	none
output	the input, in reverse order

Learn more about how to use Filters

Examples

reverse Operator (Examples)

23rd January 2015 at 10:27pm

Operator Examples reverse Operator

These examples make use of the Days of the Week tiddler.

```
[list[Days of the Week]reverse[]]
```

Try it

Rice Pudding

6th October 2020 at 8:12pm

shopping

This is a sample shopping list item for the Shopping List Example

RoadMap

16th May 2020 at 9:42pm

About

TiddlyWiki 5 is now a mature, stable project that is relied upon by many people. Simultaneously, it is rapidly evolving in many directions thanks to the broad community of developers and users. This paradoxical situation is possible because the project strictly maintains backwards compatibility, adding new features alongside the existing ones.

There is no formal roadmap, but quite a few areas that have yet to be fully implemented, such as search and replace, and rich text editing. Current work can be found on GitHub at https://github.com/Jermolene/TiddlyWiki5/

round Operator

13th June 2019 at 10:46am

Filter Operators Mathematics Operators Unary Mathematics Operators

purpose	rounds a list of numbers to the nearest integer
input	a selection of titles
output	rounds each of the input numbers to the nearest integer

Learn more about how to use Filters

Introduced in v5.1.20 See Mathematics Operators for an overview.

Examples

round Operator (Examples)

13th June 2019 at 10:46am

Operator Examples round Operator

```
[[1.6]round[]]
```

Try it

```
[[-1.6]round[]]
```

Try it

```
=-1.2 =-2.4 =3.6 =4.8 =5.1 +[round[]]
```

Try it

SafeMode

12th September 2019 at 11:31am

Features

Introduction

Safe mode provides a way to disabling most customisations in TiddlyWiki. This is useful because if TiddlyWiki is customised incorrectly it can be rendered inoperable. A particular issue is that some customisations break when upgrading to a newer core version of TiddlyWiki (especially during the beta).



Safe mode should only be used with the single file configuration of TiddlyWiki, and then only with savers that do not autosave. Using safe mode in the client server configuration can lead to data loss.

Enabling Safe Mode

Safe mode is enabled in the browser by starting TiddlyWiki with the URL hash set to the string `#:safe`. For example:

https://tiddlywiki.com/#:safe

How Safe Mode Works

Safe mode triggers two changes:

- All plugins are temporarily disabled. You can use the **control panel** ⚙ to disable individual plugins
- Any tiddlers that override shadow tiddlers are renamed to give them the prefix `SAFE:` , thus restoring the underlying shadow tiddler
- Certain configuration options are ignored, and the default settings used instead:
  - *WikiParserRuleConfiguration*

A report tiddler is displayed that allows you to inspect the tiddlers that were renamed.

sameday Operator

3rd February 2015 at 8:09pm

Date Operators Filter Operators

purpose	filter the input by date
input	a selection of titles
suffix	**F** = the name of a date field, defaulting to **modified**
parameter	**D** = a date, in the format `YYYYMMDD`
output	those input tiddlers in which field **F** has the value **D** , ignoring time

Learn more about how to use Filters

If **D** is not a valid date, the output is empty.

D may include a time of day, but this is ignored.

Examples

sameday Operator (Examples)

24th January 2015 at 11:17am

Operator Examples sameday Operator

```
[sameday[20140410]]
```

→ tiddlers modified on 10 April 2014

Try it

```
[sameday:created[20140410]]
```

→ tiddlers created on 10 April 2014

Try it

Sample Headings 1,2,3

17th November 2021 at 3:46am

Variable Examples

This is a Level 1 Heading

This is a paragraph.

This is written as a Level 3 heading

This is written as a Level 4 heading

This is written as a Level 5 heading

This is a Level 2 heading

This is written as a Level 4 heading

This is written as a Level 5 heading

This is written as a Level 6 heading

Sample Headings 3,4,5

17th November 2021 at 3:46am

Variable Examples

This is written as a Level 3 heading

This is written as a Level 4 heading

This is written as a Level 5 heading

Sample Headings 4,5,6

17th November 2021 at 3:46am

Variable Examples

This is written as a Level 4 heading

This is written as a Level 5 heading

This is written as a Level 6 heading

Sample Tiddler Template

18th May 2023 at 2:53pm

Demonstrations ParametersWidget Examples

My name is Jeremy and my age is 21.

SampleModal

17th November 2021 at 11:43pm

Message Examples

Hello,

This is an example modal containing the following message:

SampleNotification

19th November 2021 at 8:23pm

Demonstrations

This is a notification!

SampleTabFour

21st February 2015 at 10:16pm

sampletab

This is the fourth of our sample tabs.

SampleTabOne

sampletab

This is the first of our sample tabs.

SampleTabThree

sampletab

This is the third of our sample tabs.

SampleTabTwo

sampletab

This is the second of our sample tabs.

sampletag1

17th November 2021 at 10:46pm

Widget Examples

This is `sampletag1`.

sampletag2

17th November 2021 at 10:47pm

Widget Examples

This is `sampletag2`.

SampleTemplate

6th May 2023 at 7:05pm

Demonstrations thisTiddler

thisTiddler: SampleTemplate

currentTiddler: SampleTemplate

SampleTiddlerFirst

17th November 2021 at 1:36am

sampletag1 sampletag2 Widget Examples

This is a test tiddler called SampleTiddlerFirst.

SampleTiddlerSecond

17th November 2021 at 1:37am

sampletag1 sampletag2 Widget Examples

This test tiddler is called SampleTiddlerSecond.

SampleTiddlerThird

17th November 2021 at 1:37am

sampletag1 sampletag2 Widget Examples

This tiddler SampleTiddlerThird is the third test tiddler.

SampleWindowTemplate

27th February 2022 at 10:13pm

Message Examples

Name:	
Rank:	

Variable 'something' contains:

SampleWizard

19th November 2021 at 9:51pm

Demonstrations

This is a modal wizard dialogue, stored in the tiddler SampleWizard.



You can nest wizards.

SampleWizard2

19th November 2021 at 9:51pm

Demonstrations

This is another modal wizard dialogue, stored in the tiddler SampleWizard2.

You can nest wizards.

Saturday

16th November 2021 at 11:11pm

Operator Examples

This example tiddler is used to illustrate some of the Filter Operators.

SaveCommand

19th September 2017 at 3:18pm

Commands

Saves individual tiddlers identified by a filter in their raw text or binary format to the specified files.

```
--save <tiddler-filter> <filename-filter>
```

- **tiddler-filter**: A filter identifying the tiddler(s) to be saved
- **filename-filter**: Optional filter transforming tiddler titles into pathnames. If omitted, defaults to `[is[tiddler]]`, which uses the unchanged tiddler title as the filename

By default, the filename is resolved relative to the `output` subdirectory of the edition directory. The `--output` command can be used to direct output to a different directory.

Notes:

- The output directory is not cleared of any existing files
- Any missing directories in the path to the filename are automatically created.
- When saving a tiddler with spaces in its title, take care to use both the quotes required by your shell and also TiddlyWiki's double square brackets: `--save "[[Motovun Jack.jpg]]"`
- The filename filter is evaluated with the selected items being set to the title of the tiddler currently being saved, allowing the title to be used as the basis for computing the filename. For example `[encodeuricomponent[]addprefix[static/]]` applies URI encoding to each title, and then adds the prefix `static/`
- The `--save` command is a more flexible replacement for both the `--savetiddler` and `--savetiddlers` commands, which are deprecated

Examples:

- `--save "[!is[system]is[image]]" "[encodeuricomponent[]addprefix[tiddlers/]]"` – saves all non-system image tiddlers as files in the subdirectory "tiddlers" with URL-encoded titles

SaveTiddlerCommand

18th December 2013 at 1:16pm

Commands

DEPRECATED SINCE: 5.1.15 (see SaveCommand).

(Note: The `--savetiddler` command is deprecated in favour of the new, more flexible `--save` command)

Saves an individual tiddler in its raw text or binary format to the specified filename.

```
--savetiddler <title> <filename>
```

By default, the filename is resolved relative to the `output` subdirectory of the edition directory. The `--output` command can be used to direct output to a different directory.

Any missing directories in the path to the filename are automatically created.

SaveTiddlersCommand

9th June 2014 at 2:16pm

Commands

DEPRECATED SINCE: 5.1.15 (see SaveCommand).

(Note: The `--savetiddlers` command is deprecated in favour of the new, more flexible `--save` command)

Saves a group of tiddlers in their raw text or binary format to the specified directory.

```
--savetiddlers <filter> <pathname> ["noclean"]
```

By default, the pathname is resolved relative to the `output` subdirectory of the edition directory. The `--output` command can be used to direct output to a different directory.

The output directory is cleared of existing files before saving the specified files. The deletion can be disabled by specifying the **noclean** flag.

Any missing directories in the pathname are automatically created.

SaveTrail Plugin

28th March 2017 at 7:39pm

OfficialPlugins

This plugin causes TiddlyWiki to continuously download (as a JSON file) the contents of any tiddler that is manually changed by any of several means:

- Confirming an edit
- Deleting tiddlers
- Imports
- Renames/relinks
- Optionally, typing in draft tiddlers can trigger a download

Where appropriate, separate 'before' and 'after' files are downloaded. Configured correctly, the browser will download the files silently in the background, and they can be used as a backup in case of accidental data loss.

SaveWikiFolderCommand

14th April 2019 at 1:01pm

Commands

Introduced in v5.1.20 Saves the current wiki as a wiki folder, including tiddlers, plugins and configuration:

```
--savewikifolder <wikifolderpath> [<filter>] [ [<name>=<value>] ]*
```

- The target wiki folder must be empty or non-existent
- The filter specifies which tiddlers should be included. It is optional, defaulting to `[all[tiddlers]]`
- Plugins from the official plugin library are replaced with references to those plugins in the `tiddlywiki.info` file
- Custom plugins are unpacked into their own folder

The following options are supported:

- **filter**: a filter expression that defines the tiddlers to include in the output.
- **explodePlugins**: defaults to "yes"
  - **yes** will "explode" plugins into separate tiddler files and save them to the plugin directory within the wiki folder
  - **no** will suppress exploding plugins into their constituent tiddler files. It will save the plugin as a single JSON tiddler in the tiddlers folder

Note that both **explodePlugins** options will produce wiki folders that build the exact same original wiki. The difference lies in how plugins are represented in the wiki folder.

A common usage is to convert a TiddlyWiki HTML file into a wiki folder:

```
tiddlywiki --load ./mywiki.html --savewikifolder ./mywikifolder
```

Save the plugin to the tiddlers directory of the target wiki folder:

```
tiddlywiki --load ./mywiki.html --savewikifolder ./mywikifolder explodePlugins=no
```

Saving

12th August 2022 at 4:45pm

Working with TiddlyWiki

OS

- [ ] Windows
- [ ] Mac
- [ ] Linux
- [ ] Android
- [ ] iOS

Browser

- [ ] Firefox
- [ ] Chrome
- [ ] Edge
- [ ] Internet Explorer
- [ ] Safari
- [ ] Opera
- [ ] Standalone App

Use the checkboxes to explore the methods of saving that work with your platform(s)

Saving on a PHP Server

7th May 2020 at 1:03pm

PHP Saving

The built-in `TiddlySpot` saver can also be used to save changes to a simple PHP script that you can run on most hosting providers.

This code hasn't been updated in several years. If you have difficulty with it, consider using TW Receiver instead

1. Download a copy of `TiddlyHome_0.1.2.zip` from https://code.google.com/archive/p/bidix/downloads
2. Unzip.
3. Extract a copy of `store.php` from under the unzipped subdirectory `_th\lib`
4. Edit your copy of **store.php** to add your username(s) and password(s). Find the line `$USERS = array( 'UserName1'=>'Password1', etc)` and replace Username1 and Password1 with your desired username and password
    - Make sure you leave all the punctuation and code, such as the single quotes, intact
5. Save the file
6. Using FTP or your web interface, upload **store.php** to your server. Make sure that the filename is correct

    - If you've uploaded the file correctly you should be able to view it in your browser (eg, http://example.com/store.php)
7. In TiddlyWiki, go to the **Saving** tab of the **control panel** ⚙ and enter the following information:
    - Your username as the wiki name
    - Your password
    - The URL of the **store.php** file (***not*** *the URL of the wiki, this must the full URL to the* ***store.php*** *file*)

The control panel **Saving** tab includes the following configuration options:

Name	Description
Server URL	The full URL to the **store.php** file on your server
Upload filename	The filename used to save the TiddlyWiki (defaults to **index.html**)
Upload directory	The relative path from **store.php** to the directory used for saving the file
Backup directory	The relative path from **store.php** to the directory used for backups

Note about maximum size

As your TW file increases in size, you may need to modify your `.htaccess` or `htaccess` file (depending on system) to increase the value of the maximum upload and/or post size. Check with your Web Host Provider for your particular settings. Be sure also to make backups before experiementing. On some systems, the settings might look like:

```
php_value upload_max_filesize 4M
php_value post_max_size 6M
```

Note about possible error message

If you get an error message regarding `split()`, you may need to change references to `split` in **store.php** to function `explode`.

Saving on Android

7th May 2020 at 12:39pm

Android Saving Standalone App

The Tiddloid and Tiddloid Lite app are Android apps that makes it possible to edit and save changes to TiddlyWiki HTML files.

- For more information: Tiddloid Tiddloid Lite.
- Download apks: Tiddloid, TiddloidLite

Features

- Create new TiddlyWiki importing latest edition from internet
- Import existing TiddlyWikis stored on device/internal storage. (TiddloidLite supports external SD card too)
- Fork interesting ~Tiddlywikis from internet (Supports TW5 only)
- TiddlyWiki detection
- Locally stored ~Tiddlywikis are updated simultaneously on saving changes to TiddlyWikis imported to the app
- Backup system that is compatible with TiddlyDesktop, the desktop TiddlyWiki saver
- Creating shortcuts to existing TiddlyWiki on Android Homepage
- TiddloidLite supports cloud storages like GDrive and OneDrive

Please note

- Tiddloid Lite feature better support for devices running Android Q or later. It also supports cloud storages like GDrive and OneDrive, while Tiddloid keeps the compatibility to TiddlyWikiClassic. To know more about differences between Tiddloid and Tiddloid Lite, please visit

Tiddloid's homepage.

- You should keep the `.html` or `.htm` extension of the files to be imported.

Saving on Browser with File System Access API

6th February 2022 at 4:57am

Chrome Edge Linux Mac Opera Saving Windows

Tiddlywiki can save changes to the filesystem without *downloading a new file each time* each time using the File System Access API.



> This API is not yet fully standardized hence this method of saving is somewhat experimental.

The API used by this saver works in most Chromium based browsers. See caniuse for up to date information on browser support.

- Plugin Wiki
- GitHub Repo

Saving on Browser with TiddlyStow

3rd April 2023 at 8:30pm

Chrome Edge Linux Mac Opera Saving Windows

Link: https://github.com/btheado/tiddlystow

Tiddlystow saves TiddlyWiki files locally using the browser file system API (Chrome-based browsers currently). This is a simple web page for loading a local TiddlyWiki file and storing it back to the same local file requiring no plugins or extensions.

Saving on iPad/iPhone

7th October 2020 at 10:53pm

iOS Saving Standalone App

The iPad/iPhone app **Quine 2** makes it possible to view, edit and then save changes to TiddlyWiki5 on iOS. Download it here.

Instructions for use:

1. Open Quine 2
2. Tap the + toolbar button to create and open a new TiddlyWiki
3. From the file list tap an existing TiddlyWiki file to open it
4. Edit the TiddlyWiki as normal, and save as normal using either Autosave or the TiddlyWiki save button
5. Tap the left hand "Documents" toolbar button to close an open TiddlyWiki

- Quine 2 works natively in iOS with the local file system and the iCloud file system
- Quine 2 also allows you to open, edit and save TiddlyWiki files stored with cloud file providers
- Quine 2 allows you to follow embedded WikiText links and canonical links to external files for cloud-like file providers which support "folder level sharing".
    - This includes the apps "Secure Shellfish" and "Working Copy". Most providers, though, do not allow apps like Quine 2 to access linked files this way.
    - If you wish to enable such links for "well behaved" file providers, toggle "on" the "Enable folder selection for out-of-sandbox links" setting in iOS Settings for Quine 2

Note that Quine is published independently of TiddlyWiki

Saving on TiddlyDesktop

7th May 2020 at 12:43pm

Linux Mac Saving Standalone App Windows

See the Introducing TiddlyDesktop Video

1. Install the latest release of TiddlyDesktop from https://github.com/TiddlyWiki/TiddlyDesktop/releases
2. Run TiddlyDesktop
3. Use the browse button to open TiddlyWiki files
4. Save changes within TiddlyWiki in the usual way

Saving on TiddlyHost

23rd April 2021 at 2:39am

Android Chrome Edge Firefox Internet Explorer iOS Linux Mac Opera PHP Safari Saving Windows



TiddlyHost.com is a hosting service for TiddlyWiki created by Simon Baird. Once you sign up and confirm your email you can create "sites", (i.e. TiddlyWikis), with support for online saving. Sites can be private or public, and you can optionally list them on the taggable and searchable TiddlyHost Hub where they'll be discoverable by others.

Unlike TiddlySpot, TiddlyHost is secure, open source, and has proper support for TiddlyWiki5. It also allows uploading existing TiddlyWiki files, supports TiddlyWikiClassic, and lets you claim ownership of your TiddlySpot sites. For more information see the FAQ and the About page.

Saving on TiddlySpot

23rd April 2021 at 2:40am

Please note: TiddlySpot is in maintenance mode and no longer allows new sites to be created. Instead of TiddlySpot you can now use TiddlyHost, a new service from the creator of TiddlySpot.

TiddlySpot.com is a hosting service for TiddlyWiki created in 2006 by Simon Baird and Daniel Baird.

In early 2021 it was superseded by TiddlyHost, a new, secure, modern reimagining of TiddlySpot. Creating new sites on TiddlySpot is no longer supported, (though sites created in 2020 or earlier are still functional).

Security warning for TiddlySpot

Note that your password is sent unencrypted when using TiddlySpot. From the FAQ:

> **Is Tiddlyspot secure?**

> No. Tiddlyspot does not use SSL/https, so all TiddlySpot web traffic is vulnerable to packet sniffing. This means your password and site data could be intercepted by a malicious third party. For this reason, please don't keep sensitive information in your TiddlySpot site, and don't use a password that you use for other web sites.

Problems with saving on TiddlySpot

In case you run into this error when uploading a freshly upgraded local TiddlyWiki to TiddlySpot:

> Error: NS_ERROR_DOM_BAD_URI: Access to restricted URI denied

The upgrade operation falls foul of a security restriction in Firefox. Until this can be resolved, we suggest using Chrome.

- Use Chrome to open the local TiddlyWiki document you want to upload to TiddlySpot and follow the steps 1 through 5 described at Upgrading.
- Once you've checked the TiddlySpot-hosted TiddlyWiki loads properly in Chrome, you should be able to access, edit and save using TiddlyFox again.
- After you've uploaded your local document once, further editing and saving of the online version hosted on TiddlySpot should work with any modern browser of your choice. (Don't forget to fill in the TiddlySpot password in your TiddlySpot TiddlyWiki control panel for any new browser you want to use for saving changes.)

See also: Upgrading

Saving on TidGi Desktop

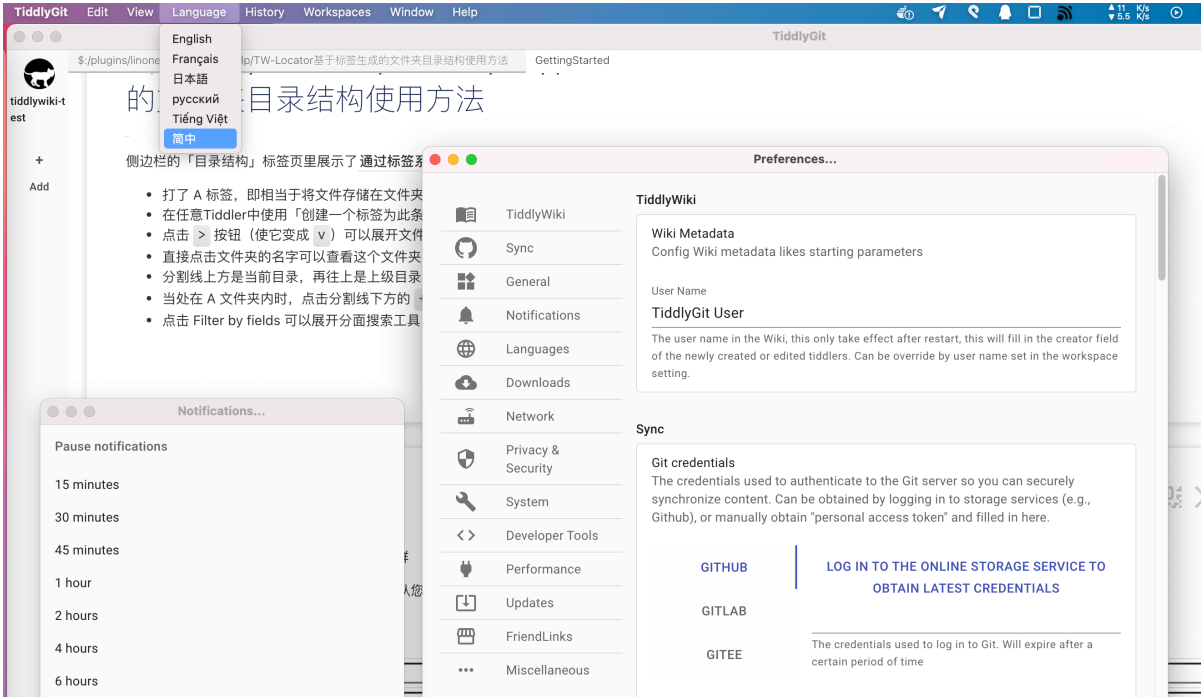
Linux Mac Saving Standalone App Windows

See its readme about details of *TidGi*'s design decisions, here is a brief summary:

1. *TidGi* runs TiddlyWiki on Node.js while TiddlyDesktop focus on single HTML wiki
2. *TidGi* has built-in git-sync backup script
3. *TidGi* provides some electron based feature, such as a menubar mini window and a shell executor.

How to install

1. Install the latest release of *TidGi* Desktop from https://github.com/tiddly-gittly/TidGi-Desktop/releases/latest
2. Run *TidGi*
3. Create a new wiki, or open an existing nodejs tiddlywiki folder
4. Changes are automatically saved, you can optionally configue it to sync to the github every 30min


TiddlyGit
Edit
View
Language
History
Workspaces
Window
Help
English
Français
日本語
русский
Tiếng Việt
简中
TiddlyGit
tiddlywiki-test
Add
GettingStarted
目录结构使用方法
侧边栏的「目录结构」标签页里展示了通过标签系
打了 A 标签，即相当于将文件存储在文件夹
在任意Tiddler中使用「创建一个标签为此条
点击 > 按钮（使它变成 v）可以展开文件
直接点击文件夹的名字可以查看这个文件夹
分割线上方是当前目录，再往上是上级目录
当处在 A 文件夹内时，点击分割线下方的
点击 Filter by fields 可以展开分面搜索工具
Notifications...
Pause notifications
15 minutes
30 minutes
45 minutes
1 hour
2 hours
4 hours
6 hours
Preferences...
TiddlyWiki
Sync
General
Notifications
Languages
Downloads
Network
Privacy & Security
System
Developer Tools
Performance
Updates
FriendLinks
Miscellaneous
TiddlyWiki
Wiki Metadata
Config Wiki metadata likes starting parameters
User Name
TiddlyGit User
The user name in the Wiki, this only take effect after restart, this will fill in the creator field of the newly created or edited tiddlers. Can be override by user name set in the workspace setting.
Sync
Git credentials
The credentials used to authenticate to the Git server so you can securely synchronize content. Can be obtained by logging in to storage services (e.g., Github), or manually obtain "personal access token" and filled in here.
GITHUB
GITLAB
GITEE
LOG IN TO THE ONLINE STORAGE SERVICE TO OBTAIN LATEST CREDENTIALS
The credentials used to log in to Git. Will expire after a certain period of time

Saving to a Git service

23rd July 2023 at 9:42am

Android Chrome Edge Firefox Internet Explorer iOS Linux Mac Opera Safari Saving Windows

TiddlyWiki can save changes directly to a GitHub repository in the single file configuration.

Saving to a Git service is configured in the **$:/ControlPanel** in the **Git Service Saver** tab under the **Saving** tab. The following settings are supported:

- **Type** - (mandatory) the type of the service (e.g. GitHub, GitLab)
- **Username** - (mandatory) the username for the Git service account used for saving changes
- **Password** - (mandatory) the OAUTH token or personal access token for the specified account. Note that GitHub deprecated password

authentication, permitted authentication methods are shown in the API documentation.

- **Repository** - (mandatory) the name of the Git repository. Both the owner name and the repository name must be specified. For example `Jermolene/TiddlyWiki5`
- **Branch** - (optional) the name of the branch to be used within the Git repository. Defaults to `main` (GitHub) or `master` (GitLab)"
- **Path** - (optional) the path to the target file. Defaults to `/`
- **Filename** - (mandatory) the filename of the target file

Notes

- The Git service token or password is stored persistently in browser local storage. Be sure to clear the password if using a shared machine. Using a personal access token for authentication offers an extra layer of security: if the access token is accidentally exposed it can be revoked without needing to reset the account password

---

For a more detailed info about the GitHub saver see: GitHub Saver Tutorial by Mohammad

Saving via a Minimal Ruby Server

7th May 2020 at 10:31pm

Android Chrome Edge Firefox Internet Explorer iOS Linux Mac Opera Safari Saving Windows

This is a minimal server using Ruby Webrick. It just serves up files from the same folder and handles saving.

Once running, just point your web browser at http://localhost:8000 to view the folder listing, and click on your wiki html file.

See the script at:

https://gist.github.com/jimfoltz/ee791c1bdd30ce137bc23cce826096da

Saving via a Minimal Web Server

2nd March 2023 at 6:59am

Android Chrome Edge Firefox Internet Explorer iOS Linux Mac Opera Safari Saving Windows

A local mini binary server for TiddlyWiki5 that saves and backups wikis and uploads files for TiddlyWiki, inspired by *tw5-server.rb* .

tw5server provides features of:

- Server for TiddlyWiki5, as well as other files (e.g. images used in TW5 `[img[images/*.png]]`);
- Easy to save wiki via browsers;
- Backup wiki in compress format (.gz), to save disk space;
- Auto clean backups: keep one newest per previous month, keep all backups in current month.
- Upload files/images to server via pressing button or drag-and-drop, for use in tiddlywiki as external links.
- Offer binary executable for Linux, macos, Android and windows.

Download executable binary at the github.com tw5-server.

Usage

```
tw5server -a:192.168.0.10 -p:8000 -d:./ -b:backup

-h usage help
```

```
-a address, defautl localhost
-p port, default 8000
-d directory to servering, default `current dir`
-b backup directory name, default `backup`
-l log saving messages to stdout

Backups auto-clean strategy:
Keep all backups in current month, keep only the newest one for previous months.
```

In Unix/Linux, maybe first `chmod +x tw5server`), then run it. For Android version, run it in Termux, or some other terminals.

Then go to http://localhost:8000 (or other address:port specified in command) in your web browser, and click on your wiki html file.

Saving via WebDAV

15th June 2022 at 5:50pm

Android Chrome Edge Firefox Internet Explorer iOS Linux Mac Opera PHP Safari Saving Windows

If hosted on a WebDAV-enabled server, TiddlyWiki will automatically save changes via HTTP for TiddlyWiki's created after 2016. If you created your wiki before Feb 16 2016 you'll need to Upgrade to enable WebDAV.

Cross platform lightweight servers

Lightweight, portable and easy to use solutions

- rclone

    - Running it can be as simple as:
      `rclone serve webdav some_directory_containing_tiddlywiki_files`
- micromata dave - the simple webdav server
- dav-server is a quick way to serve up a folder of HTML TiddlyWikis.
- hacdias webdav server
    - Guide for using this WebDAV server by Mohammad Scripts in Tiddlywiki — codes, macros, and solutions in TW

Windows

- IIS
    - Video guide by pmario: 01 Intro - How to use TiddlyWiki with IIS and WebDAV - YouTube
    - Step-by-step HowTo video by MagoArcade on using a “real” IIS server + SSL + WindowsAuth with self-signed certificate: TiddlyWiki Install - Windows IIS WebDAV Method - YouTube
- SharePoint / OneDrive for Windows
    - Save your TiddlyWiki file with an `.aspx` extension
    - Copy it to a SharePoint/OneDrive for Business sync'd library
    - As long as WebDAV has not been disabled, renaming single-file wiki from .html to .aspx “just works” when stored in one of these folders
    - This *might* work with Mac also.

OSX

- WebDavNav Server via the Mac App store.

Android

- RCX is an open source file manager for Android based on *rclone*. It is available on both *F-Droid* and *Google Play*. Thanks to its integrated *WebDAV* server, it lets you edit the wikis that you keep in your pocket. You can share them with other devices on the local network too.

Servers

Many NAS or Subversion servers support WebDAV out of the box. Setting up your own server might take some effort though:

Free Hosting

Several WebDAV hosting services tested (~~box.com~~, ~~swissdisk.com~~) don't support accessing HTML files via a webbrowser, so they won't work with TiddlyWiki. However, The GMX mediacenter (www.gmx.net) has been reported as working with *WebDAV* with a free account.

Koofr

Koofr is a cloud service that emphasizes privacy. To use their webdav services, sign up for an account at `koofr.eu` . Then go to `preferences` on their web app. Then `password` from the menu on the left. Scroll down to `App Passwords`. Use some name like 'tiddlywiki' and generate a password. Despite what it says, save the password somewhere.

You will probably need to rename your file extension to `.aspx`. Then upload your file on some path, preferably without spaces to their site. Then in the browser link to the site with an address like:

```
https://app.koofr.net/dav/Koofr/myfile.aspx
```

When you do, an authentication dialog will come up. Use your email address that you used to sign up and the password you just generated to

login.

You should be able to load the file and save changes back in place.

Paid Hosting

pCloud

pCloud is cloud service with servers in Europe and the United States. When you sign up, you need to select which server location is best for you. Then in your web browser, sign in to the service. Upload the file you wish to access via the browser interface. Then, in another tab, open either

```
https://webdav.pcloud.com
```

if your servers are in the U.S. or

```
https://ewebdav.pcloud.com
```

if your servers are in Europe.

A file navigation page should come up. Navigate to your file and open in your browser. You should be able create and save changes.

Saving with FireFox

3rd August 2023 at 11:32pm

Firefox Saving

1. Restart *Firefox*
2. *Download* an empty TiddlyWiki by clicking this button:

   > Download

3. Locate the file you just downloaded
   - You may rename it, but be sure to keep the `.html` or `.htm` extension
4. Open the file in *Firefox*
5. Try creating a new tiddler using the **new tiddler** + button in the sidebar. Type some content for the tiddler, and click the ✓ **ok** button
6. Save your changes by clicking the **save changes** button in the sidebar
   - Look for the yellow notification **Saved wiki** at the top right of the window
7. Refresh the browser window to verify that your changes have been saved correctly

Saving with Polly

23rd February 2022 at 5:04pm

Chrome Edge Firefox Internet Explorer Linux Mac Opera Safari Saving Windows

Polly is a batch file system using Windows *PowerShell* to restore TiddlyWiki files from a specified download directory to their original home directory.

Effectively, this becomes a new way to save your files, but with these features:

- No binary executables
- Human-readable batch script.

- No special plugins in your TiddlyWiki file
- No special browser required
- No browser extension required.
- No need for node.exe running in background
- Total size expanded package only 100k
- Backups as regular file and/or zip to specified directories
- The ability to "parrot" extra copies to target directories (e.g. a Dropbox folder)

It should be able to run anywhere that *PowerShell* runs, including Windows, Linux, and Mac.

https://github.com/Marxsal/polly

Saving with the HTML5 saver

10th December 2022 at 10:57pm

Android Chrome Edge Firefox Internet Explorer iOS Linux Mac Opera Safari Saving Windows

This is the default method of saving if no other method is installed. It uses your browser's built-in "download a file" handler, and has the advantage of working on almost all desktop browsers, and many mobile browsers.

1. *Download* an empty TiddlyWiki by clicking this button:

   > Download
   >
   > If the button doesn't work save this link: https://tiddlywiki.com/empty.html

> Your browser may ask you to accept the download before it begins

2. Locate the file you just downloaded
    - You may rename it, but be sure to keep the `.html` or `.htm` extension
3. Open the file in your browser
4. Try creating a new tiddler using the **new tiddler** + button in the sidebar. Type some content for the tiddler, and click the ✓ **ok** button
5. Save your changes by clicking the ⊘ **save changes** button in the sidebar
6. Your browser will download a new copy of the wiki incorporating your changes
7. Locate the newly downloaded file and open it in your browser
8. Verify that your changes have been saved correctly

Tip: most browsers have an option to prompt each time for the download location. This allows you to select the existing version of the file and replace it.

Saving with TiddlyFox

6th August 2023 at 2:14am

Firefox Saving

DEPRECATED SINCE: FIREFOX 57 (see Saving with FireFox)

Saving with TiddlyFox on Android

13th November 2017 at 2:16pm

Saving with TiddlyFox

(Alternatively, see the video tutorial)

1. Ensure you have the latest version of *Firefox for Android*
   - http://getfirefox.com
2. Install the latest release of the TiddlyFox extension from:
   - https://addons.mozilla.org/en-GB/firefox/addon/tiddlyfox/
3. Install this extension to be able to save the TiddlyWiki file locally:
   - https://addons.mozilla.org/en-GB/android/addon/save-link-menus/
4. *Download* an empty TiddlyWiki by saving this link:
   - https://tiddlywiki.com/empty.html

   > (to save the link, ensure you've installed the "save-link-menus" extension and then long-press on the link and choose "Save link")

5. When the file has downloaded, click on it within the notification tray or the download manager application
6. Choose to open the file in Firefox (rather than the default Android viewer)
7. Click **OK** in response to the prompt from TiddlyFox that asks whether to enable saving for this file
8. Try creating a new tiddler using the **new tiddler** + button in the sidebar. Type some content for the tiddler, and click the ✓ **ok** button
9. Save your changes by clicking the **save changes** button in the sidebar
   - Look for the yellow notification **Saved wiki** at the top right of the window
10. Refresh the browser window to verify that your changes have been saved correctly

Saving with TiddlyIE

7th May 2020 at 10:14pm

Internet Explorer Saving

1. Install the TiddlyIE add-on from:
   - https://github.com/davidjade/TiddlyIE/releases
2. Restart Internet Explorer. IE will prompt you to enable the TiddlyIE add-on.

   > You may also see a prompt to enable the *Microsoft Script Runtime*

3. *Download* an empty TiddlyWiki by saving this link:

   > https://tiddlywiki.com/empty.html

4. Locate the file you just downloaded
   - You may rename it, but be sure to keep the `.html` or `.htm` extension
5. Open the file in Internet Explorer
6. Try creating a new tiddler using the **new tiddler** + button in the sidebar. Type some content for the tiddler, and click the ✓ **ok** button
7. Save your changes by clicking the ⊘ **save changes** button in the sidebar. Internet Explorer will ask for your consent to save the file locally by presenting a file **Save As** dialog.
8. Refresh the browser window to verify that your changes have been saved correctly

Saving with TW Receiver

13th June 2020 at 1:43am

PHP Saving

TW Receiver is a TiddlyWiki plugin and PHP script used for saving to a PHP based server.

It's features include simple automated backups, stale Instance Overwrite Protection, challenge digest authentication (enhanced security), and data integrity signing (enhanced security).

- Visit TW-Receiver for more information, plugin, and code.

SavingMechanism

9th July 2016 at 1:27pm

Mechanisms

The SavingMechanism is the mechanism by which TiddlyWiki generates a new HTML file and stores it. It is different from the sync mechanism, which is concerned with synchronising changes to individual tiddlers back to a server (see https://tiddlywiki.com/dev/#SyncAdaptorModules for more details).

The following steps are involved:

1. The saver mechanism uses the filter defined in **$:/config/SaverFilter** to specify which modified tiddlers trigger the dirty state for the wiki. (The dirty state is reflected in the red colouring of the "save changes" button in the sidebar).
2. The ButtonWidget is used to generate a tm-save-wiki message to trigger the save operation
3. The optional parameter for the message specifies the template that will be used for generating the HTML file, defaulting to **$:/core/save/all**
    - The plugin *$:/plugins/tiddlywiki/tiddlyweb* in the client-server configuration replaces the default template with *$:/plugins/tiddlywiki/tiddlyweb/save/offline*
4. The template includes a variable assignment that specifies a filter to select the tiddlers that should be included in the saved file. It then transcludes the main page template `$:/core/templates/tiddlywiki5.html`
    - The template includes a reference to `$(publishFilter)$` that allows the filter to be customised via a global variable

5. The tm-save-wiki handler renders the template to generate the HTML file
6. The message handler chooses the highest priority "saver" module that can handle saving the file
    - See https://tiddlywiki.com/dev/#Saver for more details

Scalability

24th November 2021 at 10:56pm

Features

You might expect that TiddlyWiki's architecture as a SinglePageApplication would make it unsuitable for large amounts of data. In fact, TiddlyWiki users regularly work with files that are tens or even over a hundred megabytes without problems - and successful experiments have been done up into the gigabytes.

See performance tips for large wikis.

Scripts for TiddlyWiki on Node.js

15th October 2014 at 6:53pm

TiddlyWiki on Node.js

Script Files

The TiddlyWiki5 repository contains several scripts in the `bin` folder that you can use to automate common tasks, or as a useful starting point for your own scripts. See *Scripts for building tiddlywiki.com* for details of the scripts used to build and release https://tiddlywiki.com/.

All the scripts expect to be run from the root folder of the repository.

`serve`: serves tw5.com

```
./bin/serve.sh -h
./bin/serve.sh [edition dir] [username] [password] [host] [port]
```

Or:

```
./bin/serve.cmd -h
./bin/serve.cmd [edition dir] [username] [password] [host] [port]
```

This script starts TiddlyWiki5 running as an HTTP server, defaulting to the content from the `tw5.com-server` edition. By default, the Node.js serves on port 8080. If the optional `username` parameter is provided, it is used for signing edits. If the `password` is provided then HTTP basic authentication is used. Run the script with the `-h` parameter to see online help.

To experiment with this configuration, run the script and then visit `http://127.0.0.1:8080` in a browser.

Changes made in the browser propagate to the server over HTTP (use the browser developer console to see these requests). The server then syncs changes to the file system (and logs each change to the screen).

`test`: build and run tests

This script runs the `test` edition of TiddlyWiki on the server to perform the server-side tests and to build `test.html` for running the tests in the browser.

`lazy`: serves tw5.com with lazily loaded images

```
./bin/lazy.sh <username> [<password>]
```

Or:

```
./bin/lazy.cmd <username> [<password>]
```

This script serves the `tw5.com-server` edition content with LazyLoading applied to images.

ScrollableWidget

31st July 2023 at 12:09pm

Widgets

Introduction

The scrollable widget wraps its content in a scrollable frame. The user can scroll the contents with the mouse or with touch gestures. Code can use the WidgetMessage: tm-scroll to programmatically scroll specific DOM nodes into view.

Content and Attributes

The content of the `<$scrollable>` widget is displayed within a pair of wrapper DIVs. If the inner DIV is larger then it scrolls within the outer one. CSS is used to specify the size of the outer wrapper.

Attribute	Description
class	The CSS class(es) to be applied to the outer DIV
fallthrough	See below
bind	ⓘ **New in v5.3.2** Optional title of tiddler to which the scroll position should be bound

Binding the scroll position to a tiddler automatically copies the scroll coordinates into the `scroll-left` and `scroll-top` fields after scrolling occurs. Conversely, setting those field values will automatically cause the scrollable to scroll if it can.

Note

> If a scrollable widget can't handle the `tm-scroll` message because the inner DIV fits within the outer DIV, then by default the message falls through to the parent widget. Setting the **fallthrough** attribute to `no` prevents this behaviour.

Examples

These examples require the following CSS definitions from $:/_tw5.com-styles:

```
.tc-scrollable-demo {
    border: 1px solid <<colour message-border>>;
    background-color: <<colour message-background>>;
    padding: 1em;
    height: 400px;
    position: relative;
}
```

Simple Usage

This wiki text shows how to display a list within the scrollable widget:

copy to clipboard

```
<$scrollable class='tc-scrollable-demo'>
<$list filter='[tag[Reference]]'>

<$view field='title'/>: <$list filter='[all[current]links[]sort[title]]' storyview='pop'>
<$link><$view field='title'/></$link>
</$list>

</$list>
</$scrollable>
```

That renders as:

Concepts: TiddlyWiki

Definitions: Concepts TiddlyWiki

WikiText: Editor toolbar Formatting in WikiText Formatting text in TiddlyWiki Markdown

Macros: Brackets Core Macros Custom Widgets Functions Macro Calls Macro Definitions Macro Parameter Handling Macro Pitfalls Pragma: \define Procedures substitute Operator Substituted Attribute Values Transclusion Transclusion and Substitution Variables

Variables: Core Variables Custom Widgets Filter Operators Filters Functions ImportVariablesWidget Macro Definitions Macro Parameter Handling Macros Procedures Variable Usage Widgets

SystemTags:

Widgets: HTML in WikiText Widgets in WikiText WikiText

Filters: **$:/AdvancedSearch** CountWidget ListWidget TiddlyWiki Title Selection Transclusion in WikiText

Messages: Widgets

Binding scroll position to a tiddler

Current scroll position: ,

Set current scroll position to 100,100

Scroll up by 10 pixels

Scroll down by 10 pixels

copy to clipboard

```
<$scrollable class='tc-scrollable-demo' bind='$:/my-scroll-position'>
<$list filter='[tag[Reference]]'>
```

```
<$view field='title'/>: <$list filter='[all[current]links[]sort[title]]' storyview='pop'>
<$link><$view field='title'/></$link>
</$list>

</$list>
</$scrollable>
```

That renders as:

Concepts: TiddlyWiki

Definitions: Concepts TiddlyWiki

WikiText: Editor toolbar Formatting in WikiText Formatting text in TiddlyWiki Markdown

Macros: Brackets Core Macros Custom Widgets Functions Macro Calls Macro Definitions Macro Parameter Handling Macro Pitfalls Pragma: \define Procedures substitute Operator Substituted Attribute Values Transclusion Transclusion and Substitution Variables

Variables: Core Variables Custom Widgets Filter Operators Filters Functions ImportVariablesWidget Macro Definitions Macro Parameter Handling Macros Procedures Variable Usage Widgets

SystemTags:

Widgets: HTML in WikiText Widgets in WikiText WikiText

Filters: **$:/AdvancedSearch** CountWidget ListWidget TiddlyWiki Title Selection Transclusion in WikiText

Messages: Widgets

search Operator

11th July 2023 at 10:43am

Common Operators Field Operators Filter Operators Negatable Operators

purpose	filter the input by searching tiddler content
input	a selection of titles
suffix	the **search** operator uses a rich suffix, see below for details
parameter	one or more search terms, separated by spaces, or a literal search string
output	those input tiddlers in which *ALL* of the search terms can be found in the value of field **F**
! **output**	those input tiddlers in which *NOT* all of the search terms can be found

Learn more about how to use Filters

Introduced in v5.1.18 The search filter operator was significantly enhanced in 5.1.18. Earlier versions do not support the extended syntax and therefore do not permit searching multiple fields, or the **literal** or **casesensitive** options.

The **search** operator uses an extended syntax that permits multiple fields and flags to be passed:

```
[search:<field list>:<flag list>[<parameter>]]
```

- **field list**: a comma delimited list of field names to restrict the search

    - defaults to **`tags`**, **`text`** and **`title`** if blank
    - an asterisk `*` instead of the field list causes the search to be performed across all fields available on each tiddler
    - preceding the list with a minus sign `-` reverses the order so that the search is performed on all fields except the listed fields
- **flag list**: a comma delimited list of flags (defaults to `words` if blank)
- **parameter**: filter parameter

This example searches the fields **`title`** and **`caption`** for a case-sensitive match for the literal string :

```
[search:title,caption:literal,casesensitive[The first]]
```

The available flags are:

- Search mode - the first to be set of the following flags determines the type of search that is performed:
    - **literal**: considers the search string to be a literal string, and requires an exact match
    - **whitespace**: considers the search string to be a literal string, but will consider all runs of whitespace to be equivalent to a single space. Thus `A B` matches `A  B`
    - **regexp**: treats the search string as a regular expression. Note that the **regexp** option obviates the need for the older **`regexp`** operator.
    - **words**: (the default) treats the search string as a list of tokens separated by whitespace, and matches if **all of the tokens** appear in the string (regardless of ordering and whether there is other text in between)
    - **some**: ⓘ **Introduced in v5.2.2** treats the search string as a list of tokens separated by whitespace, and matches if **at least one** of the tokens appear in the string
- **casesensitive**: if present, this flag forces a case-sensitive match, where upper and lower case letters are considered different. By default, upper and lower case letters are considered identical for matching purposes.

- **anchored**: ⓘ **Introduced in v5.1.20** anchors the search to the start of the string (applies to **whitespace**, **literal** and **words** modes)

Examples

search Operator (Examples)

29th November 2021 at 1:27pm

Operator Examples search Operator

```
[!is[system]search[table]]
```

→ non-system tiddlers containing the word "table"

Try it

```
[all[shadows]search[table]]
```

→ shadow tiddlers containing the word "table"

Try it

`[search:caption[arch]]`

→ tiddlers containing `arch` in their **`caption`** field

Try it

`[search:*[arch]]`

→ tiddlers containing `arch` in any field

Try it

`[search:-title,caption[arch]]`

→ tiddlers containing `arch` in any field except **`title`** and **`caption`**

Try it

`[!is[system]search[the first]]`

→ non-system tiddlers containing a case-insensitive match for both the "the" and "first"

Try it

`[!is[system]search::literal[the first]]`

→ non-system tiddlers containing a case-insensitive match for the literal phrase "the first"

Try it

`[!is[system]search::literal,casesensitive[The first]]`

→ non-system tiddlers containing a case-sensitive match for the literal phrase "The first"

Try it

`[search:caption,description:casesensitive,words[arch]]`

→ any tiddlers containing a case-sensitive match for the word `arch` in their **`caption`** or **`description`** fields

Try it

`[search:tags:some[how test]]`

→ any tiddlers containing at least 1 of the search terms in the field: **`tags`**

Try it

`[search:tags:some,casesensitive[how test]]`

→ any tiddlers containing at least 1 of the case-sensitive search terms in the field: **tags**

Try it

`[search:tags,title:some,anchored[how test]]`

→ any tiddlers containing at least 1 of anchored search terms in the fields: **tags** and **title**

Try it

search-replace Operator

29th August 2021 at 7:45pm

Filter Operators String Operators

purpose	returns each item in the list, replacing within each title the string specified by the first parameter with the second parameter
input	a selection of titles
suffix	the **search-replace** operator uses a rich suffix, see below for details
parameter	the **search-replace** operator requires 2 parameters, see below for details
output	the input titles with the search string replaced with the replacement string

Learn more about how to use Filters

Introduced in v5.1.23

The **search-replace** operator uses an extended syntax that allows for multiple suffixes and parameters to be passed:

```
[search-replace:<flag list>:<regexp-mode>[<search-term>],[<replacement>]]
```

- **flag-list**: **g** for global mode to replace all matches, **i** for case-insensitive mode, **Introduced in v5.2.0** **m** for multiline mode, "gim" for all. (optional)
- **regexp-mode**: **regexp** to treat the first parameter as a regular expression (optional).
- **search-term**: string or regular expression that should be replaced
- **replacement**: string that should replace the search-term



Either parameter can be a string, a text reference or a variable

Examples

search-replace Operator (Examples)

1st November 2021 at 1:52pm

Operator Examples search-replace Operator

Replace one string with another:

```
[[The quick brown fox jumps over the lazy dog]search-replace[dog],[cat]]
```

Try it

Replace all matches of a regular expression:

```
\define myregexp() e|o
```

```
[[Hello There]search-replace:g:regexp<myregexp>,[]]
```

Try it

Replace all matches of a string irrespective of case:

```
[[Hello There]search-replace:gi[H],[]]
```

Try it

You can also use regular expression capture groups in the replacement string: `\define names() (\w+)\s(\w+)`

```
[[John Smith]search-replace::regexp<names>,[$2,$1]]
```

Try it

You can reference the portion of the input that matches the regular expression with `$&`:

```
[[John Smith]search-replace::regexp[John .*],[His name is $&]]
```

Try it

```
[[This is an exciting feature]search-replace::regexp[exciting],[amazing and $&]]
```

Try it

To replace everything but a match using a regular expression and the **multiline** (m) flag: `\define myregexp2() ^(?!Unlike).*$`

```
[[HelloThere]get[text]search-replace:gm:regexp<myregexp2>,[]]
```

Try it

You can use the search-replace Operator in combination with the sortsub Operator to ignore stop words like "A" and "The" at the beginning of titles when sorting. (Note for example that "A Tale of Two Cities" sorts under "T" rather than "A")

```
[enlist<book-list>sortsub<compare-without-stopwords>]
```

Try it

You can also customise the listing so that "A Tale of Two Cities" is listed as "Tale of Two Cities, A" but still links to the correct tiddler.

copy to clipboard

```
<ul>
<$list filter="[enlist<book-list>sortsub<compare-without-stopwords>]">
<li><$link><$text text={{{[<currentTiddler>search-replace:i:regexp[(^The |A )(.*)],[$2, $1]]}}}/></
$link></li>
</$list>
</ul>
```

That renders as:

- *Animal Farm*
- *Dice Man, The*
- *Great Expectations*
- *New Kind of Science, A*
- *Recursive Universe, The*
- *Tale of Two Cities, A*

Searching in TiddlyWiki

7th June 2016 at 6:30pm

Features Working with TiddlyWiki

Searching is fast and flexible in TiddlyWiki.

Standard search

Typing text into the search box in the sidebar will turn up a list of all the tiddlers that contain that text.

> *(Note that the sidebar tabs will be hidden by the search results until you click the 'x' to the right of the search box to remove the search results)*

Advanced searching

Clicking on the magnifying glass icon to the right of the search box will open **$:/AdvancedSearch**. This tiddler contains four tabs:

- The **standard** tab contains another instance of the search box found in the sidebar
- The **system** tab allows you to limit your search to system tiddlers
- The **shadows** tab allows you to limit your search to shadow tiddlers
- The **filter** tab is not a search box, per se, but a way to obtain a list of all tiddlers that meet the specific criteria described by that filter, for example, "All tags except system tags"

Searchwikis by Mohammad

17th November 2020 at 5:04pm

Community Plugins

Searchwikis plugin uses index (dataTiddler) to search multiple external Tiddlywikis.

https://kookma.github.io/TW-Searchwikis/

It has two parts

1. An indexer, to build an index of all tiddlers in an external wiki
2. A search tool to search indexes and display a link to a tiddler found in an external wiki

Then one master wiki can host many index tiddlers and lets to search several external wikis through index tiddlers.

Searchwikis enable to have a central wiki and search all other wikis from one place.

Second

14th November 2021 at 2:36am

Contents Table-of-Contents Demonstrations

This is an example tiddler. See Table-of-Contents Macros (Examples).

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate

velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

SecondOne

14th November 2021 at 2:36am

Second Table-of-Contents Demonstrations

This is an example tiddler. See Table-of-Contents Macros (Examples).

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

SecondThree

14th November 2021 at 2:36am

Second Table-of-Contents Demonstrations

This is an example tiddler. See Table-of-Contents Macros (Examples).

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

SecondThreeOne

14th November 2021 at 2:36am

SecondThree Table-of-Contents Demonstrations

This is an example tiddler. See Table-of-Contents Macros (Examples).

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

SecondThreeThree

14th November 2021 at 2:36am

SecondThree Table-of-Contents Demonstrations

Important

It's important that this tiddler has no "child" to be able to visually test every possible toc code-path.

- This tiddler has a field **toc-link: no**
- Do not tag any other tiddler with the title of this one

SecondThreeTwo

14th November 2021 at 2:36am

SecondThree Table-of-Contents Demonstrations

This is an example tiddler. See Table-of-Contents Macros (Examples).

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

SecondTwo

14th November 2021 at 2:36am

Second Table-of-Contents Demonstrations

This is an example tiddler. See Table-of-Contents Macros (Examples).

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

SeeAlso by Matias Goldman

6th January 2021 at 4:10pm

Other Resources

A simple technique for adding tag-based cross references to the default view template.

http://seealso.tiddlyspot.com

> See also: - is a feature to generate a "See also:" section after the tiddler text, somewhat like in Wikipedia articles.

Selection Constructors

17th September 2015 at 9:37pm

Filters

The output of a Filter Step depends on its operator:

- Most operators derive their output from their input. For example, many of them output a subset of their input, and thus truly live up to the name of "filters", narrowing down the overall output of the containing run. These operators are called **selection modifiers**.
- A few operators ignore their input and generate an independent output instead. These are called **selection constructors**: they construct an entirely new selection.

A good example of a constructor is **`title`**. The output of `[title[A]title[B]]` is just **B**. But the **`field`** operator is a modifier, so `[title[A]field:title[B]` outputs nothing at all.

SelectWidget

13th November 2023 at 10:33am

TriggeringWidgets Widgets

Introduction

The select widget displays a popup menu based on a HTML select element. The popup (or dropdown) contains a list of items defined by `<option>` and `<optgroup>` elements. Every time the user selects a new value in the menu, the selected value is written to the text of a specified tiddler field or index and any ActionWidgets within the `actions` attribute are triggered. Conversely, if the tiddler value is changed independently the select widget is automatically updated to reflect the new value.

In multiple selection mode, the list of selected values is bound to the specified tiddler field or index. Browsers generally use the `ctrl` or `cmd` keys for multiple selection.

For example, this select widget displays a list of the tags in this wiki:

copy to clipboard

```
<$select tiddler=<<qualify 'select-demo'>> default='HelloThere'>
<$list filter='[all[shadows+tiddlers]tags[]sort[title]]'>
<option value=<<currentTiddler>>><$view field='title'/></option>
</$list>
</$select>
```

That renders as:

$:/deprecated

The *state tiddler* currently contains:

See the text change as you switch entries in the select widget. Try changing the value of the state tiddler and see the select widget change. Notice how the select widget only displays an entry if there is a precise match with the tiddler text.

Content and Attributes

The content of the `<$select>` widget should be one or more HTML `<option>` or `<optiongroup>` elements that provide the available values.

Attribute	Description
tiddler	The title of the tiddler containing the value to be displayed/modified by the select widget (defaults to the current tiddler)
field	The field name for the value in the current tiddler (defaults to "text")
index	The index of a property in a DataTiddler (takes precedence over the field attribute)
class	CSS classes to be assigned to the HTML select element
tooltip	Optional tooltip
default	Default value to be used if the tiddler, field or index specifies a missing value
multiple	If present, switches to multiple selection mode
size	The number of rows to display in multiple selection mode
actions	A string containing ActionWidgets to be triggered when the key combination is detected
focus	ⓘ **Introduced in v5.2.4** Optional. Set to "yes" to automatically focus the HTML select element after creation
tabindex	ⓘ **New in v5.3.1** Optional. Sets the `tabindex` attribute of the HTML select element to the given value
data-*	ⓘ **New in v5.3.2** Optional data attributes to be assigned to the HTML element
style.*	ⓘ **New in v5.3.2** Optional CSS properties to be assigned to the HTML element

Examples

Simple Lists

This example sets the title of the current wiki $:/SiteTitle to one of a list of book titles:

copy to clipboard

```
<$select tiddler='$:/SiteTitle' tooltip='Choose a new site title'>
<option>A Tale of Two Cities</option>
<option>A New Kind of Science</option>
<option>The Dice Man</option>
</$select>
```

That renders as:

A Tale of Two Cities

Simple List with Placeholder Value

To display a default value that is also disabled, effectively functioning as a placeholder, the following form can be used. Note that the targeted field must be empty, or not exist, for the placeholder to show in the widget:

copy to clipboard

```
<$select tiddler='New Tiddler' field='text' default='Choose a new text'>
```

```
<option disabled>Choose a new text</option>
<option>A Tale of Two Cities</option>
<option>A New Kind of Science</option>
<option>The Dice Man</option>
</$select>
```

That renders as:

A Tale of Two Cities

Value lists

In this example the `value` attribute has been used to specify the text that should be used as the value of the entry instead of the display text.

copy to clipboard

```
<$select tiddler='$:/SiteTitle'>
<option value='cities'>A Tale of Two Cities</option>
<option value='science'>A New Kind of Science</option>
<option value='dice'>The Dice Man</option>
</$select>
```

That renders as:

A Tale of Two Cities

Option Groups

Entries in the list can be grouped together with the `<optgroup>` element

copy to clipboard

```
<$select tiddler='$:/SiteTitle'>
<optgroup label='Fiction'>
<option value='cities'>A Tale of Two Cities</option>
<option value='dice'>The Dice Man</option>
</optgroup>
<optgroup label='Non-fiction'>
<option value='science'>A New Kind of Science</option>
<option value='recursive'>The Recursive Universe</option>
```

```
</optgroup>
</$select>
```

That renders as:

A Tale of Two Cities

Generated Lists

The ListWidget can be used to generate the options for a select widget. For example, here we combine a select widget listing all the tiddlers tagged TableOfContents with a transclusion to display the text of the selected one.

copy to clipboard

```
<$select tiddler='$:/generated-list-demo-state'>
<$list filter='[tag[TableOfContents]]'>
<option><$view field='title'/></option>
</$list>
</$select>
<$tiddler tiddler={{$:/generated-list-demo-state}}>
<$transclude mode='block'/>
</$tiddler>
```

That renders as:

HelloThere

Nested Lists

This example uses a nested pair of list widgets. The outer one generates the `<optgroup>` elements, and the inner one generates `<option>` elements:

copy to clipboard

```
<$select tiddler='$:/generated-list-demo-nestedstate' field='type' default='text/vnd.tiddlywiki'>
<$list filter='[all[shadows+tiddlers]prefix[$:/language/Docs/Types/]each[group]sort[group]]'>
<optgroup label={{!!group}}>
<$list filter='[all[shadows+tiddlers]prefix[$:/language/Docs/Types/]group{!!group}] +
[sort[description]]'>
<option value={{!!name}}><$view field='description'><$view field='title'/></$view> (<$view field='name'/
>)</option>
</$list>
</optgroup>
</$list>
</$select>
```

That renders as:

Data dictionary (application/x-tiddler-dictionary)

Multiple Selections

This example uses the `multiple` keyword to specify that we should be able to select multiple items.

copy to clipboard

```
<$select tiddler='$:/generated-list-demo-state' field='testing' multiple size='8'>
<$list filter='[tag[TableOfContents]]'>
<option><$view field='title'/></option>
</$list>
</$select><br />
<$list filter='[list[$:/generated-list-demo-state!!testing]]'>
<$view field='title' /><br />
</$list>
```

That renders as:

HelloThere
Learning
Working with TiddlyWiki
Customise TiddlyWiki
Features
Filters
Languages
Editions

Actions

This example uses the actions attribute to apply days of the week as tags to the current tiddler.

copy to clipboard

```
<$select tiddler='$:/generated-list-demo-state' field='actions-test'
actions='<$action-listops $field="myfield" $tags={{$:/generated-list-demo-state!!actions-test}}/>'
>
<$list filter='[list[Days of the Week]]'>
<option><$view field='title'/></option>
</$list>
</$select>
```

That renders as:

Monday

Semantic Colors by Mohammad

6th January 2021 at 4:10pm

Other Resources

Semantic colors are set of CSS to apply colorful theme to individual tiddlers for categorisation purpose.

https://kookma.github.io/TW-Semantic-Colors/

Tiddlywiki "semantic colors" has two objectives:

1. How it is simply possible to apply theme to an individual tiddler
2. Use semantic colors for categorisation

Some use cases

Learning materials (each topic, category can have dedicated semantic color, like learning a foreign language)
Slideshow (section divider, each part can have a semantic color)
Books and tutorials (different semantic colors can be applied to chapters, specific subjects, like example tiddlers)
Visual tags (instead of looking for tag bar and see to what category this subject (tiddler) belongs, you can identify it by its semantic color)

sentencecase Operator

19th June 2019 at 1:07pm

Filter Operators

purpose	returns each item in the list with the first letter capitalised
input	a selection of titles
output	the input titles with each first letter capitalised

Learn more about how to use Filters

ⓘ Introduced in v5.1.20

See also lowercase Operator, uppercase Operator, titlecase Operator.



This operator does not change anything except the first letter of each title, so `[[aBcD eFgH]]` would become `[[ABcD eFgH]]`. If you want `[[Abcd efgh]]`, then use a filter like `[lowercase[]sentencecase[]]`.

Examples

sentencecase Operator (Examples)

20th June 2019 at 4:03pm

Operator Examples sentencecase Operator

```
[[abc def ghi jkl]sentencecase[]]
```

Try it

```
[[abc deF gHi jKL]sentencecase[]]
```

Try it

```
[[abc deF gHi jKL]lowercase[]sentencecase[]]
```

Try it

```
[tag[HelloThere]sentencecase[]]
```

Try it

ServerCommand

26th June 2018 at 5:05pm

Commands

Note that the `--server` command is now deprecated in favour of the new ListenCommand.

See WebServer for details of TiddlyWiki's web server functionality.

Legacy command to serve a wiki over HTTP.

```
--server <port> <root-tiddler> <root-render-type> <root-serve-type> <username> <password> <host> <path-prefix> <debug-level>
```

The parameters are:

- **port** - port number on which to listen; non-numeric values are interpreted as a system environment variable from which the port number is extracted (defaults to "8080")
- **root-tiddler** - the tiddler to serve at the root (defaults to "**$:/core/save/all**")
- **root-render-type** - the content type to which the root tiddler should be rendered (defaults to "text/plain")
- **root-serve-type** - the content type with which the root tiddler should be served (defaults to "text/html")
- **username** - the default username for signing edits
- **password** - optional password for basic authentication

- **host** - optional hostname to serve from (defaults to "127.0.0.1" aka "localhost")
- **path-prefix** - optional prefix for paths
- **debug-level** - optional debug level; set to "debug" to view request details (defaults to "none")

If the password parameter is specified then the browser will prompt the user for the username and password. Note that the password is transmitted in plain text so this implementation should only be used on a trusted network or over HTTPS.

For example:

```
--server 8080 $:/core/save/all text/plain text/html MyUserName passw0rd
```

The username and password can be specified as empty strings if you need to set the hostname or pathprefix and don't want to require a password.

```
--server 8080 $:/core/save/all text/plain text/html "" "" 192.168.0.245
```

Using an address like this exposes your system to the local network. For information on opening up your instance to the entire local network, and possible security concerns, see the WebServer tiddler at TiddlyWiki.com.

To run multiple TiddlyWiki servers at the same time you'll need to put each one on a different port. It can be useful to use an environment variable to pass the port number to the Node.js process. This example references an environment variable called "MY_PORT_NUMBER":

```
--server MY_PORT_NUMBER $:/core/save/all text/plain text/html MyUserName passw0rd
```

Serving TW5 from Android

7th May 2020 at 1:29pm

Android Saving TiddlyWiki on Node.js

Termux is and open source android application providing limited Unix environment enabling users to install Node.js and npm modules in android. Users can install and run TiddlyWiki on Node.js using Termux.

Instructions

- Download and install Termux from Google Play Store or Fdroid
- Open termux and run the following commands one by one

```
apt update
apt upgrade
apt install nodejs
npm install -g tiddlywiki
```

- If you need to create/serve **TiddlyWiki on Node.js** from the internal storage, you need to give termux storage permission by running the following command in termux

```
termux-setup-storage
```

- Now you can create and serve **TiddlyWiki on Node.js** from internal storage. In the example given below, user is creating a new wiki called "mynewwiki" in his internal folder.

```
cd storage/shared
tiddlywiki mynewwiki --init server
tiddlywiki mynewwiki --listen
```

- Visit http://127.0.0.1:8080/ in your browser
- From then on, as long as *Termux* is not closed, you may access your wiki anytime from your favourite Web browser pointing on the expected address and port.

- For more information regarding TiddlyWiki on Node.js, please see Installing TiddlyWiki on Node.js
- For more information and tips regarding termux, please refer to Termux wiki



In *Termux*, you may as well install *git*, *emacs* or *vi*, in order to edit and maintain individual tiddler files. This would probably require that you also attach a more powerful keyboard to your Android, like the Hacker's Keyboard application or a Bluetooth external device.

SetFieldCommand

9th June 2014 at 2:16pm

Commands

Note that this command is experimental and may change or be replaced before being finalised

Sets the specified field of a group of tiddlers to the result of wikifying a template tiddler with the `currentTiddler` variable set to the tiddler.

```
--setfield <filter> <fieldname> <templatetitle> <rendertype>
```

The parameters are:

- **filter** - filter identifying the tiddlers to be affected
- **fieldname** - the field to modify (defaults to "text")

- **templatetitle** - the tiddler to wikify into the specified field. If blank or missing then the specified field is deleted
- **rendertype** - the text type to render (defaults to "text/plain"; "text/html" can be used to include HTML tags)

SetMultipleVariablesWidget

23rd October 2021 at 6:24pm

Widgets

Introduction

The setmultiplevariables widget assigns values to multiple variables where the names and values of the variables are specified as separate filter.

Content and Attributes

The content of the `<$setmultiplevariables>` widget is the scope for the value assigned to the variable.

Attribute	Description
$names	Filter evaluating to the names of a list of variables to assign
$values	Filter evaluating to the values corresponding to the list of names specified in `$names`

Examples

Here is an example of a convoluted way to display the values of the fields of the tiddler "HelloThere":

copy to clipboard

```
<$setmultiplevariables $names="[[HelloThere]fields[]addprefix[demo-]]" $values="[[HelloThere]fields[]]
:map[[HelloThere]get<currentTiddler>]">
<ol>
<$list filter="[variables[]prefix[demo-]sort[]]">
<li>
''<$text text={{{ [<currentTiddler>removeprefix[demo-]] }}}/>'': <$text text={{{
[<currentTiddler>getvariable[]] }}}/>
</li>
</$list>
</ol>
</$setmultiplevariables>
```

That renders as:

1. **created**: 20130822170200000
2. **list**: [[A Gentle Guide to TiddlyWiki]] [[Discover TiddlyWiki]] [[Some of the things you can do with TiddlyWiki]] [[Ten reasons to switch to TiddlyWiki]] Examples [[What happened to the original TiddlyWiki?]]
3. **modified**: 20231223102201587
4. **tags**: TableOfContents
5. **text**: !!.tc-hero-heading "Welcome to TiddlyWiki, a unique [[non-linear|Philosophy of Tiddlers]] notebook for [[capturing|Creating and editing tiddlers]], [[organising|Structuring TiddlyWiki]] and [[sharing|Sharing your tiddlers with others]] complex information" Use it to keep your [[to-do list|TaskManagementExample]], to plan an [[essay or novel|"TiddlyWiki for Scholars" by Alberto Molina]], or to organise your

wedding. Record every thought that crosses your brain, or build a flexible and responsive website. TiddlyWiki lets you choose where to keep your data, guaranteeing that in the decades to come you will [[still be able to use|Future Proof]] the notes you take today. !! "Quick Start" <$list filter="[tag[Quick Start]]"> <$macrocall $name="flex-card" bordercolor={{!!color}} textcolor={{!!text-color}} backgroundcolor={{!!background-color}} captionField="caption" descriptionField="text"/> </$list>   <$link to="中文社区 - Chinese Community" class="tc-btn-big-green tc-card"> 中文社区
Chinese Community </$link> !! "Find Out More" <$list filter="[tag[HelloThumbnail]]"> <$macrocall $name="flex-card" captionField="caption" descriptionField="text"/> </$list> </div> !! "Community" <div class="tc-cards" style="font-size:0.7em;text-align:center;margin:3em auto;"> {{$:/core/images/help}} ~TalkTW   {{$:/core/images/video}} ~YouTube   {{$:/core/images/twitter}} Twitter   {{$:/core/images/mastodon}} Mastodon   {{$:/core/images/github}} ~GitHub   {{$:/core/images/gitter}} Gitter   {{Reddit Logo}} Reddit   {{Open Collective Logo}} Open Collective !! "Testimonials & Reviews" <$list filter="[tag[Testimonial]]"> <$macrocall $name="flex-card" class="tc-card-quote" captionField="caption" descriptionField="text"/> </$list>

6. **title**: HelloThere
7. **type**: text/vnd.tiddlywiki

Setting a favicon

10th May 2020 at 1:57pm

Customise TiddlyWiki

"favicons" are small icons that most browsers display to help users identify websites. Note that all browsers support bitmap images, but only certain modern browsers support SVG format icons.

favicons in the Browser

When TiddlyWiki starts up in the browser it looks for a tiddler called $:/favicon.ico and dynamically uses it as the favicon for the page. If you modify the image then the favicon changes instantly to reflect it.

ⓘ **Introduced in v5.1.23** To set an external image as the favicon, assign the URL to the **_canonical_uri** field of the $:/favicon.ico tiddler.

favicons on the Server

On the server, the ServerCommand will serve the tiddler $:/favicon.ico at the path `/favicon.ico`.

Setting a page background image

10th June 2016 at 10:27am

Customise TiddlyWiki

1. Import your image as a tiddler (see Images in WikiText)
    - You can also use an external image
2. Open the **control panel** ⚙ and switch to the **Appearance**/**Theme Tweaks** tab
3. Select your image from the dropdown labelled "Page background image"
4. Set "Page background image attachment" to "Fixed to window" to have the background stay stationary and the content to scroll over the top of it, or "Scroll with tiddlers" to have it move (note that the iPhone/iPad doesn't support the fixed setting for performance reasons)
5. Set "Page background image size" as follows:
    - **Auto** causes the background image to be tiled over the page background
    - **Cover** causes the background image to be sized so that it completely covers the page. Some of the image may be clipped
    - **Contain** causes the background image to be sized so that it fits within the page

Note that the palette *DarkPhotos* is provided to make the sidebar more readable on dark background images.

SetVariableWidget

25th October 2014 at 1:56pm

Widgets

The `<$setvariable>` widget is a synonym for `<$set/>`; see SetWidget for more details.

SetWidget

20th July 2023 at 7:47pm

Widgets

Introduction

The set variable widget assigns a value to a specified variable. The new value of the variable is available to the content within the set variable widget.

Content and Attributes

The content of the `<$set>` widget is the scope for the value assigned to the variable.

Attribute	Description
name	The name of the variable to assign (defaults to "currentTiddler")
value	The value to assign to the variable if the **`filter`** attribute is missing or not empty
tiddler	ⓘ **Introduced in v5.1.15** Optional title of the tiddler from which to read the value
subtiddler	ⓘ **Introduced in v5.1.16** Optionally specifies the title of a subtiddler within a plugin tiddler identified by the **tiddler** attribute
field	ⓘ **Introduced in v5.1.15** Optional field of the tiddler from which to read the value (only used if **tiddler** is used)
index	ⓘ **Introduced in v5.1.15** Optional index of the tiddler from which to read the value (only used if **tiddler** is used)
filter	An optional filter to be evaluated and assigned to the variable (see "Filtered List Variable Assignment" below). The variable can be used with the **`enlist`** operator
select	ⓘ **Introduced in v5.1.14** An optional zero-based index of the item to return from the filter output (see below)
emptyValue	The value to assign to the variable if the specified value is missing or empty (see below)



If the value of your variable is enclosed in double square brackets this might indicate that you are returning a list of values from the filter. To use a single title from the filter output without the double square brackets see **Filtered Item Variable Assignment** below.

Examples

Simple Variable Assignment

The simplest way of using set variable widget assigns a string to a variable. The following example assigns a literal string

copy to clipboard

```
<$set name="myVariable" value="Some text">
<$text text=<<myVariable>>/>
</$set>
```

That renders as:

Some text

Both the name and value attributes can be transcluded. For example:

copy to clipboard

```
<$set name=anotherVariable value="myVariable">
  <$set name=<<anotherVariable>> value={{$:/language/DefaultNewTiddlerTitle}}>
    <$text text=<<myVariable>>/>
  </$set>
</$set>
```

That renders as:

New Tiddler

Conditional Variable Assignment

This form of the set variable widget chooses one of two specified values according to whether a filter evaluates to an empty list. Here's an example that sets a variable according to whether the current tiddler is called "myMagicTitle":

copy to clipboard

```
<$set name="myVariable" filter="[all[current]field:title[myMagicTitle]]" value="It's magic"
emptyValue="It's not magic">
<$text text=<<myVariable>>/>
</$set>
```

That renders as:

It's not magic

Filtered List Variable Assignment

This form of the set variable widget evaluates the filter and assigns the result to the variable as a space-separated list (using double square brackets for titles containing spaces).



The Title List format cannot reliably represent items that contain certain specific character sequences such as `]]`. Thus it should not be used where there is a possibility of such sequences occurring.

copy to clipboard

```
<$set name="myVariable" filter="[tag[HelloThere]]">
<$text text=<<myVariable>>/>
</$set>
```

That renders as:

[[A Gentle Guide to TiddlyWiki]] [[Discover TiddlyWiki]] [[Some of the things you can do with TiddlyWiki]] [[Ten reasons to switch to TiddlyWiki]] Examples [[What happened to the original TiddlyWiki?]] [[Funding TiddlyWiki]] [[Open Collective]]

Filtered Item Variable Assignment Single Element

Introduced in v5.1.14 This form of the set variable widget evaluates the filter and assigns the specified result to the variable as a single item (ie, not using double square brackets for titles containing spaces).

copy to clipboard

```
<$set name="myVariable" filter="[tag[HelloThere]]" select="0">
<$text text=<<myVariable>>/>
</$set>
```

That renders as:

> A Gentle Guide to TiddlyWiki

Transcluded Variable Assignment

Introduced in v5.1.15 This form of the set variable widget obtains the value to assign to the variable from a value in a tiddler field or index. For example:

copy to clipboard

```
<$set name="myVariable" tiddler="HelloThere" field="list">
<$text text=<<myVariable>>/>
</$set>
```

That renders as:

[[A Gentle Guide to TiddlyWiki]] [[Discover TiddlyWiki]] [[Some of the things you can do with TiddlyWiki]] [[Ten reasons to switch to TiddlyWiki]] Examples [[What happened to the original TiddlyWiki?]]

The example above could also be written as `<$set name="myVariable" value={{HelloThere!!list}}>`. The advantage of using the **tiddler** attribute is that the tiddler title and field or index can themselves be computed. For example:

copy to clipboard

```
<$set name="myTiddler" value="HelloThere">
  <$set name="myVariable" tiddler=<<myTiddler>> field={{$:/docs/anyField!!field}}>
```

```
    <$text text=<<myVariable>>/>
  </$set>
</$set>
```

That renders as:

[[A Gentle Guide to TiddlyWiki]] [[Discover TiddlyWiki]] [[Some of the things you can do with TiddlyWiki]] [[Ten reasons to switch to TiddlyWiki]] Examples [[What happened to the original TiddlyWiki?]]

Using the Set Widget to Create Global Variables

There are times when it makes sense to use the features of the SetWidget rather than procedures or functions to create global variables. This can be accomplished by placing the set variable widget in a tiddler that is tagged $:/tags/Global. If multiple variables are required, the set variable widget can be nested as shown here:

```
<$set name="myGlobalVariable" value="I am global">
    <$set name="myOtherGlobalVariable" value="I am also a global variable.">
    </$set>
</$set>
```

sha256 Operator

29th January 2022 at 11:55am

Filter Operators String Operators

purpose	apply sha256 hash to a string
input	a selection of titles
parameter	**L** = the number of characters to return, up to a maximum of 64
output	the base64 encoded sha256 of the input, truncated to the specified length

Learn more about how to use Filters

Introduced in v5.1.14

Hashes are a way of turning strings of arbitrary length into obfuscated strings of fixed length. They are often used in situations where items need to be stored by name but it is inconvenient to allow arbitrary length strings.

See Wikipedia for details of the sha256 operation.

Examples

sha256 Operator (Examples)

8th March 2024 at 1:29pm

Operator Examples sha256 Operator

```
[[test]sha256[]]
```

Try it

```
[[test]sha256[64]]
```

Try it

shadowsource Operator

3rd February 2015 at 8:11pm

Filter Operators Special Operators

purpose	select the plugin titles that contain the input shadows
input	a selection of shadow titles
parameter	none
output	the plugins that contain the input tiddlers

Learn more about how to use Filters

Each input title is processed in turn. If it denotes a shadow tiddler, the title of its plugin tiddler is dominantly appended to the output. Non-shadow tiddlers contribute nothing to the output.

Examples

shadowsource Operator (Examples)

24th January 2015 at 10:07pm

Operator Examples shadowsource Operator

```
[[$:/core/copyright.txt]shadowsource[]]
```

Try it

```
$:/core/copyright.txt $:/plugins/tiddlywiki/railroad/parser.js +[shadowsource[]]
```

Try it

ShadowTiddlers

Concepts

ShadowTiddlers are tiddlers that are loaded from Plugins at the wiki startup. Unlike ordinary tiddlers, they don't appear in most lists.

Overriding Shadow Tiddlers to modify plugins

A *ShadowTiddler* can be overridden with an ordinary tiddler of the same name. This leaves the shadow tiddler intact but the plugin will use the overriding tiddler in its place, effectively allowing users to modify the behaviour of plugins.

Users are cautioned against overriding shadow tiddlers because if the shadow tiddler is changed in a plugin update, the overriding tiddler may no longer perform as intended. To remedy this, the overriding tiddler may be modified or deleted. If the overriding tiddler is deleted, then the plugin falls back to using the original shadow tiddler.

Overridden Shadow Tiddlers

- $:/config/WikiParserRules/Inline/wikilink
- $:/core/templates/static.content
- $:/DefaultTiddlers
- $:/language
- $:/SiteSubtitle
- $:/SiteTitle
- $:/tags/MenuBar
- $:/themes/tiddlywiki/vanilla/options/sidebarlayout
- GettingStarted

Shadow Tiddlers

A full list of shadow tiddlers can be found in the sidebar in the "More" -> "Shadows" tab

Share Plugin

28th December 2020 at 3:31pm

OfficialPlugins

> Do not install this plugin unless you understand exactly what it does

This experimental plugin provides tools to share tiddlers via URLs, comprising:

- The ability to load a group of tiddlers from the browser location hash at startup
- Wizard and templates to create URLs from group of tiddlers

Sharing a TiddlyWiki on Dropbox

8th June 2015 at 5:20am

Learning

You can work on a TiddlyWiki file in Dropbox and publish a URL that anyone can use to see a read-only view of the file.

1. Save your TiddlyWiki file within your Dropbox folder
2. Select Dropbox's "Share Link" option to obtain a public URL for the file
    - In the Dropbox web interface, this is done by clicking the link icon that appears when you hover over a file. Dropbox also adds a "Share Link" item to the file context menu in Finder on OS X and Explorer on Windows
3. The URL generated by Dropbox will have this form:

    ```
    https://www.dropbox.com/s/<gobbledegook>/mywiki.html
    ```

4. Modify the URL like this, carefully preserving the gobbledegook:

    ```
    https://dl.dropboxusercontent.com/s/<gobbledegook>/mywiki.html
    ```

The result is a "secret" URL that you can send to other people to enable them to see the wiki.

Enter a generated URL here and you can copy and paste the secret URL:

https://www.dropbox.com/s/<gobbledegook>/mywiki.html

`https://dl.dropboxusercontent.com/s/<gobbledegook>/mywiki.html` (open)

Sharing your tiddlers with others

28th December 2020 at 3:34pm

Learning

There are a number of ways that you can share tiddlers or your whole TiddlyWiki:

- You can attach a TiddlyWiki to an e-mail
- You can publish your TiddlyWiki online and grab a link to send or message to others:
  - A link to the web address of the whole TiddlyWiki file
  - A permalink (#) to a specific tiddler
  - A *permaview* (⌗) link of all the currently open tiddlers
- You can share a Dropbox link to your TiddlyWiki
- You can export tiddlers (⍐) in a variety of formats including text, static HTML and comma separated values (ie spreadsheet compatible)
- You can also share tiddlers merely by making your TiddlyWiki accessible to others, for example by publishing it online, so that they can import tiddlers from it
- There is an experimental sharing plugin that you can use to share tiddlers via the URL location hash.

Shiraz by Mohammad

16th November 2020 at 10:06pm

Community Plugins

Shiraz plugin is a very small in size and framework to create stylish contents in Tiddlywiki.

https://kookma.github.io/TW-Shiraz/

Shiraz plugin contains extended markups, macros, styles, and many customisation to empty Tiddlywiki and can be used as a **starter kit**.

Some of Shiraz features are:

- Customized elements like, alerts, cards, panels, badges
- Dynamic tables
- Sortable tables
- Display on demand, slider, and details
- Images, basic image macros, slidein, and overlay images, polaroid and pretty images
- Table customisation tools
- Sticky footer
- Multi columns tiddler
- Multi column story river
- List search
- Badge status
- Notebook style
- Test utilities
- Stylish buttons

Adding Shiraz plugin to any Tiddlywiki converts it to a full production tool. Shiraz uses modified CSS classes from Bootstrap.

Shortcut Filter Run Prefix

10th July 2023 at 9:44am

Filter Run Prefix

Shortcut prefixes are commonly used by advanced users because they are fast to type but they are harder to read by less experienced users. That's why named prefixes have been created, which are more verbose. Shortcut and named filter run prefixes are interchangeable as shown in the table below.

TypeError: setting getter-only property "parentNode"

If a run has:

- no prefix, its output titles are dominantly appended to the filter's output
- the prefix `+`, it receives the filter output so far as its input; its output then *REPLACES* all filter output so far and forms the input for the next run
- the prefix `-`, output titles are *REMOVED* from the filter's output (if such tiddlers exist)
- the prefix `~`, if the filter output so far is an empty list then the output titles of the run are dominantly appended to the filter's output. If the filter output so far is not an empty list then the run is ignored. ⓘ **Introduced in v5.1.18**
- the prefix `=`, output titles are appended to the filter's output without de-duplication. ⓘ **Introduced in v5.1.20**

Interchangeable Filter Run Prefixes

In technical / logical terms:

Run	Equivalent named prefix	Interpretation	Output
`[run]`	`:or[run]`	de-duplicated union of sets	... OR run
`+[run]`	`:and[run]`	accumulation of filter steps	... AND run
`-[run]`	`:except[run]`	difference of sets	... AND NOT run
`~[run]`	`:else[run]`	else	... ELSE run
`=[run]`	`:all[run]`	union of sets without de-duplication	... OR run

The input of a run is normally a list of all the non-shadow tiddler titles in the wiki (in no particular order).
But the `+` prefix can change this:

Prefix	Input
`-`, `~`, `=`, `:intersection` or none	`[all[]]` tiddler titles, unless otherwise determined by the first filter operator
`+`, `:filter`, `:map`, `:reduce`, `:sort`	the filter output of all previous runs so far

Precisely because of varying inputs, be aware that both prefixes `-` and `+` do not behave inverse to one another!

For example, in both of the following, `$:/baz` will only be removed if it actually exists:

- `foo bar $:/baz -[is[system]]`
- `foo bar $:/baz -[prefix[$:/]]`

To understand why, consider the input for both final runs with their `-` prefix.

In order to remove `$:/baz` in any case, existing or not, simply use the `+` prefix with negated filter operators:

- `foo bar $:/baz +[!is[system]]`

- `foo bar $:/baz +[!prefix[$:/]]`

Difference between + and intersection

For the difference between `+` and `:intersection`, see Intersection Filter Run Prefix (Examples).

For Developers

To create a new filter run prefix, create a Javascript module with a module-type of `filterrunprefix`.

sign Operator

13th June 2019 at 10:50am

Filter Operators Mathematics Operators Unary Mathematics Operators

purpose	return -1, 0 or 1 for a list of numbers according to whether each number is negative, zero, or positive
input	a selection of titles
output	replaces each input number with -1, 0 or 1 according to whether the number is negative, zero, or positive

Learn more about how to use Filters

Introduced in v5.1.20 See Mathematics Operators for an overview.

Examples

sign Operator (Examples)

13th June 2019 at 10:51am

Operator Examples sign Operator

```
[[1.6]sign[]]
```

Try it

```
[[-1.6]sign[]]
```

Try it

```
[[0]sign[]]
```

Try it

Signing the Contributor License Agreement

26th February 2022 at 6:55pm

Create a GitHub pull request to add your name to `cla-individual.md` or `cla-entity.md`, with the date in the format (YYYY/MM/DD).

step by step

1. Navigate to licenses/CLA-individual or licenses/CLA-entity according to whether you are signing as an individual or representative of an organisation
2. Ensure that the "branch" dropdown at the top left is set to `tiddlywiki-com`
3. Click the "edit" button at the top-right corner (clicking this button will fork the project so you can edit the file)
4. Add your name at the bottom
   - eg: `Jeremy Ruston, @Jermolene, 2011/11/22`
5. Below the edit box for the CLA text you should see a box labelled **Propose file change**
6. Enter a brief title to explain the change (eg, "Signing the CLA")
7. Click the green button labelled **Propose file change**
8. On the following screen, click the green button labelled **Create pull request**

Simple ways to write protect tiddlers

6th November 2021 at 3:16am

How to apply custom styles Learning

Sometimes you want to protect individual tiddlers from accidental changes, either from yourself or guests to your TiddlyWiki file. Here's a couple simple approaches. Note that these do not provide hardened protection against a wilful attack.

Using a stylesheet to lock tiddlers by tag name

Create a tiddler with the following contents and tag it as `$:/tags/Stylesheet` :

```
[data-tags*="Locked"] button[title="Edit this tiddler"]   {display: none;}
```

If your TW language isn't English, then you'll need to change the text in 'title="...."' to whatever the text is when you hover over the edit button.

Now any tiddler you tag as "Locked" will have the edit button hidden. To get it back you will need to open the stylesheet tiddler and change "none" to "inline".

Using transclusion to hide your tiddler in a system tiddler

- Open your tiddler for editing.
- Select the title and copy.
- Select all the tiddler text.
- Click on the excise tool.
- In the title box, type `$:/` and then paste the original title.
- Use the defaults to excise the text into a new tiddler and replace it with a transclusion

Now you can view the contents of your original tiddler, but to edit them you will have to find the the tiddler with the `$:/` prefix.

Simple Zork-like Game by Jed Carty

18th November 2014 at 3:36pm

Examples

A simple game built with TiddlyWiki, introduced in this post.

http://zorklike.tiddlyspot.com

> I decided to see if it was possible to make some sort of game using only the core tiddlywiki with no plugins or javascript. I made a very bare bones zork/interactive fiction type game. It is currently the simplest thing I could make and claim it was a game, but I may add on to it in the future. It uses the 5.1.5 prerelease because the action-setfield widget saved lots of work making macros.
>
> One goal of this is to use nothing besides what is contained in the core tiddlywiki, so there is no javascript and no plugins.

sin Operator

22nd October 2021 at 12:28am

Filter Operators Mathematics Operators Trigonometric Operators Unary Mathematics Operators

purpose	calculate the sine value of a list of angles (given in radians)
input	a selection of titles
output	the sine of the input angles (numeric value between -1 and 1)

Learn more about how to use Filters

ⓘ **Introduced in v5.1.21** See Mathematics Operators for an overview.

Examples

sin Operator (Examples)

20th October 2021 at 4:21pm

Operator Examples sin Operator

```
[[2]sin[]]
```

Try it

```
=1 =2 =3 =4 +[sin[]]
```

Try it

SingleFileApplication

26th January 2014 at 5:03pm

Definitions

A single file application is a web application where all of the assets are delivered in a single static file. This means that it can be downloaded and used offline without losing functionality. It can also be hosted on almost any type of web server.

TiddlyWiki is an unusual single file application because it stores its data within the same file, and is capable of saving changes to itself.

Every single file application must also by definition be a SinglePageApplication, but the converse is not true.

SinglePageApplication

23rd August 2013 at 7:51pm

Definitions

As defined by Wikipedia, a SinglePageApplication is a web application that presents itself as a single HTML document that dynamically loads additional content as the user browses the site. Frequently, single page applications are carefully designed to hide their nature, appearing and functioning as an ordinary static website by respecting the address bar and the back/forward navigation controls.

Unusually, TiddlyWiki is also a SingleFileApplication.

Slider by Mohammad

17th November 2020 at 5:29pm

Community Plugins

Slider is a plugin to create an ordered set of tiddlers also called Trail.

https://kookma.github.io/slider/

A trail can be used to create a sequence of selected contents like step by step tutorial, guided help, lessons and similar.

The slider plugin user interface contains three sections

- A sidebar tab called Trails to manage trails
- A dashboard for each trail (where trail structure is managed)
- Tiddlers in a trail, also called slides

SlotWidget

11th May 2023 at 2:39pm

Widgets

Introduction

New in v5.3.0 The `$slot` widget is used within transcluded content to mark "slots" that the transcluding widget can fill with the `$fill` widget.

See the `$transclude` widget for details.

Attributes

The content of the `$slot` widget is used as a fallback for the slot content if the corresponding `$fill` widget is not found.

Attribute	Description
$name	The name of the slot being defined
$depth	Optional number indicating how deep the `$slot` widget is compared to the matching `$fill` widget as measured by the number of nested transclude widgets (defaults to 1). Transclude widgets whose **`$fillignore`** attribute is set to **yes** are ignored, and do not affect the depth count

Examples

Quoted content

When content contains quotes, passing it through attributes and parameters can be challenging. However, passing the content using the **$fill** widget content eliminates the need to wrap it in quotes, making the process easier.

If a variable named **bold_slot** contains the following **$slot** definition:

```
<b>
<$slot $name="body"/>
</b>
```

then the slot can be filled using this variable transclusion:

copy to clipboard

```
<$transclude $variable=bold_slot>
<$fill $name=body>

"""
some text
using [[Hard Linebreaks in WikiText]]
syntax
"""
</$fill>
</$transclude>
```

That renders as:

some text

using Hard Linebreaks in WikiText
syntax

Depth

If a variable named **table_slot** contains the following **$slot** definition:

```
|!depth|!slot1|!slot2|
|1|<$slot $name=slot1/>|<$slot $name=slot2/>|
|2|<$slot $name=slot1 $depth=2>missing</$slot>|<$slot $name=slot2 $depth=2>missing</$slot>|
```

then the slot values can be filled at different transclusion depths:

copy to clipboard

```
<$transclude $variable=table_slot  $mode=block>
  <$fill $name=slot1>outer1</$fill>
  <$fill $name=slot2>outer2
    <$transclude $variable=table_slot $mode=block>
      <$fill $name=slot1>inner1</$fill>
      <$fill $name=slot2>inner2</$fill>
    </$transclude>
  </$fill>
</$transclude>
```

That renders as:

<table>
<tr><th>depth</th><th>slot1</th><th>slot2</th></tr>
<tr><td>1</td><td>outer1</td><td>outer2

<table>
<tr><th>depth</th><th>slot1</th><th>slot2</th></tr>
<tr><td>1</td><td>inner1</td><td>inner2</td></tr>
<tr><td>2</td><td>outer1</td><td>outer2

<table>
<tr><th>depth</th><th>slot1</th><th>slot2</th></tr>
<tr><td>1</td><td>inner1</td><td>inner2</td></tr>
<tr><td>2</td><td>inner1</td><td>inner2</td></tr>
</table></td></tr>
</table></td></tr>
<tr><td>2</td><td>missing</td><td>missing</td></tr>
</table>

The **`slot1`** slot is filled at both depths with a simple string (outer1 and outer2). For **`slot2`**, the outer instance is a simple string but the inner instance recursively transcludes the same **`table_slot`** variable again. The recursion ends at the third transclusion call since both "inner" slots are filled with simple strings.

slugify Operator

9th May 2020 at 4:17pm

Filter Operators

purpose	returns each item in the list in a human-readable form for use in URLs or filenames
input	a selection of titles
output	the input titles transformed so that they only contain lower case letters, numbers, periods, dashes and underscores

Learn more about how to use Filters

ⓘ **Introduced in v5.1.23** The transformation applied by the slugify operator follows these steps:

- If there is a tiddler with the same title that has a **slug** field, then return that field instead of running the following steps
- Replace uppercase letters with lowercase equivalents
- Transliterate diacritics to their basic lowercase ASCII equivalents (for example, "Æ" is transliterated to "AE")
- Replace spaces with dashes
- Remove all non-word characters except dashes and periods
- Replace multiple sequential dashes with a single dash
- Trim dashes from start and end
- If the result is the empty string then character codes are used instead (eg. "&£@" transforms to "38-163-64")

Note that it is possible for more than one title to slugify to the same string. The `duplicateslugs` can be used to alert authors to any clashes.

Examples

Some of the things you can do with TiddlyWiki

18th October 2021 at 12:23pm

HelloThere

- Take notes, and use tags and hyperlinking to form relationships between your notes
- Use tabs, tables, tag-based lists, and tables of contents to get organised
- Bookmark your favorite websites (see an example at http://giffmex.org/experiments/tidmarks.html)
- Keep track of tasks and appointments, and organise them by multiple tags (see our TaskManagementExample)
- Inventory just about anything: your recipes, personal library, contacts, music collection, and more
- Create a blog or website
- Write a book
- Organise your images into galleries (see Icon Gallery)
- Share the information in your TiddlyWiki with others, as an online file, as a file attachment, as a tiddler file, or as a link to a specific online tiddler (try clicking and dragging a tiddler from one TiddlyWiki file to another to see what happens)
- Draw a sketch (Edit Motovun Jack.jpg and start drawing over the image to see what happens)
- Use familiar web user interface elements such as lightboxes - see SampleWizard
- Create a slideshow presentation
- Set up an entire local or online knowledgebase, with a central TiddlyWiki file linking to other TiddlyWiki files (http://recursos.giffmex.org is a Spanish online example of this)
- Set up a data visualisation using tiddlers as data (see the visualisations at https://tiddlywiki.com/plugins/tiddlywiki/d3/)

Sort Filter Run Prefix

22nd March 2023 at 3:07pm

Named Filter Run Prefix

purpose	sort the input titles by the result of evaluating this filter run for each item
input	the filter output of all previous runs so far

suffix	the **`:sort`** filter run prefix uses a rich suffix, see below for details
output	output titles replace the output of previous filter runs

Introduced in v5.2.0
TypeError: setting getter-only property "parentNode"

Each input title from previous runs is passed to this run in turn. The filter run transforms the input titles into the form needed for sorting. For example, the filter run `[length[]]` transforms each input title in to the number representing its length, and thus sorts the input titles according to their length.

Note that within the filter run, the **`currentTiddler`** variable is set to the title of the tiddler being processed. This permits filter runs like `:sort:number[{!!value}divide{!!cost}]` to be used for computation. The value of **`currentTiddler`** outside the run is available in the variable **`..currentTiddler`**.

The `:sort` filter run prefix uses an extended syntax that allows for multiple suffixes, some of which are required:

```
:sort:<type>:<flaglist>[...filter run...]
```

- **type**: Required. Determines how the items are compared and can be any of: **string**, **alphanumeric**, **number**, **integer**, **version** or **date**.
- **flaglist**: comma separated list of the following flags:
  - **casesensitive** or **caseinsensitive** (required for types `string` and `alphanumeric`).
  - **reverse** to invert the order of the filter run (optional).

Note that filter runs used with the `:sort` prefix should return the same number of items that they are passed. Any missing entries will be treated as zero or the empty string. In particular, when retrieving the value of a field with the get Operator it is helpful to guard against a missing field value using the else Operator. For example `[get[myfield]else[default-value]...`.

Examples

Sort Filter Run Prefix (Examples)

15th March 2023 at 5:53pm

Sort Filter Run Prefix

Sort by title length:

```
[all[tiddlers]] :sort:number[length[]] +[limit[10]]
```

Try it

Sort by title length reversed:

```
[all[tiddlers]] :sort:number:reverse[length[]] +[limit[10]]
```

Try it

Sort by text length:

```
[all[tiddlers]] :sort:number[get[text]length[]] +[limit[10]]
```

Try it

Sort by newest of modified dates:

```
[tag[Field Operators]] :sort:date[get[modified]else[19700101]] +[limit[10]]
```

Try it

Sort by title:

```
[tag[Field Operators]] :sort:string:casesensitive[get[caption]] +[limit[10]]
```

Try it

Sort by title in reverse order:

```
[tag[Field Operators]] :sort:string:casesensitive,reverse[get[caption]] +[limit[10]]
```

Try it

Sort as text with case sensitivity:

```
Apple Banana Orange Grapefruit guava DragonFruit Kiwi apple orange  :sort:string:casesensitive[{!!title}]
```

Try it

Sort as text ignoring case:

```
Apple Banana Orange Grapefruit guava DragonFruit Kiwi apple orange :sort:string:caseinsensitive[{!!title}]
```

Try it

sort Operator

3rd February 2015 at 8:12pm

Common Operators Field Operators Filter Operators Negatable Operators Order Operators

purpose	sort the input by text field
input	a selection of titles
parameter	**F** = the name of a field, defaulting to **title**
output	the input, sorted into ascending order by field **F**, treating field values as text
! **output**	the input, likewise sorted into descending order

Learn more about how to use Filters

The difference between capital and lowercase letters is ignored. Compare **sortcs**.

Examples

sort Operator (Examples)

24th January 2015 at 12:32pm

Operator Examples sort Operator

These examples make use of the Days of the Week tiddler.

```
[list[Days of the Week]sort[]]
```

Try it

```
[list[Days of the Week]!sort[]]
```

Try it

```
one two Three four +[sort[]]
```

Try it

```
[prefix[Tiddl]sort[text]]
```

Try it

```
[has[created]sort[created]limit[10]]
```

→ the oldest 10 tiddlers in the wiki

Try it

sortan Operator

23rd February 2018 at 2:25am

Common Operators Field Operators Filter Operators Negatable Operators Order Operators

purpose	sort the input by text field considering them as alphanumerics
input	a selection of titles
parameter	**F** = the name of a field, defaulting to **`title`**
output	the input, sorted into ascending order by field **F**, treating field values as alphanumerics
! **output**	the input, likewise sorted into descending order

Learn more about how to use Filters

The difference between uppercase and lowercase letters is ignored.

Examples

sortan Operator (Examples)

27th February 2022 at 10:09pm

Operator Examples sortan Operator

These examples make use of the Apple tiddler.

Try the following example with `sort` instead of `sortan`

```
[list[Apple]sortan[]]
```

Try it

```
[list[Apple]!sortan[]]
```

Try it

```
App-10-A App-9-B App-10-B App-9-A +[sortan[]]
```

Try it

sortby Operator

8th November 2015 at 6:21am

Filter Operators Listops Operators Order Operators

purpose	sort the current list in the order of the list referenced in the parameter
input	a list of items
parameter	**`order`** = a list specifying the order in which to sort the current list
output	all items sorted by lookup list

Learn more about how to use Filters

Examples

sortby Operator (Examples)

8th November 2015 at 6:21am

Operator Examples sortby Operator

These examples make use of the Days of the Week tiddler.

```
10 6 4 9 3 2 8 +[sortby[1 2 3 4 5 6 7 8 9 10]]
```

Try it

```
Friday Tuesday Monday Thursday Sunday +[sortby{Days of the Week!!list}]
```

Try it

```
1 Mon 5 Fri 4 Tue Sun 2  +[sortby{Days of the Week!!short}]
```

Try it

sortcs Operator

17th April 2015 at 2:57pm

Field Operators Filter Operators Negatable Operators Order Operators

purpose	sort the input by text field, treating upper and lower case as different
input	a selection of titles
parameter	**F** = the name of a field, defaulting to **title**
output	the input, sorted into ascending order by field **F**, treating field values as text

! **output**	the input, likewise sorted into descending order

Learn more about how to use Filters

Capital and lowercase letters are treated as different. Compare **sort**.

Examples

sortcs Operator (Examples)

24th January 2015 at 12:33pm

Operator Examples sortcs Operator

These examples make use of the Days of the Week tiddler.

```
one two Three four +[sortcs[]]
```

Try it

```
one two Three four +[!sortcs[]]
```

Try it

sortsub Operator

22nd May 2021 at 6:25pm

Field Operators Filter Operators Negatable Operators Order Operators

purpose	sort the input by the result of evaluating a subfilter for each item
input	a selection of titles
suffix	**T** = the type used for the comparison (string, number, integer, date, version), defaulting to string
parameter	**S** = a subfilter to be evaluated
output	the input, sorted into ascending order by the result of evaluating subfilter **S**
! **output**	the input, sorted into reverse order by the result of evaluating subfilter **S**

Learn more about how to use Filters

Each item in the list of input titles is passed to the subfilter in turn. The subfilter transforms the input titles into the form needed for sorting. For example, the subfilter `[length[]]` transforms each input title in the number representing its length, and thus sorts the input titles according to their length.

Note that within the subfilter, the "currentTiddler" variable is set to the title of the tiddler being processed. This permits subfilters like `[{!!value}divide{!!cost}]` to be used for computation.

The value of currentTiddler outside the subfilter is available in the variable "..currentTiddler". ⓘ **Introduced in v5.2.0**

The suffix **T** determines how the items are compared and can be:

- "string" (the default)

- "number" - invalid numbers are interpreted as zero
- "integer" - invalid integers are interpreted as zero
- "date" - invalid dates are interpreted as 1st January 1970
- "version" - invalid versions are interpreted as "v0.0.0"
- "alphanumeric" - treat items as alphanumerics ⓘ **Introduced in v5.2.0**

Note that subfilters should return the same number of items that they are passed. Any missing entries will be treated as zero or the empty string. In particular, when retrieving the value of a field with the get Operator it is helpful to guard against a missing field value using the else Operator. For example `[get[myfield]else[default-value]...`.

Examples

sortsub Operator (Examples)

18th November 2020 at 11:33am

Operator Examples sortsub Operator

These examples make use of the following variables:

- **compare-by-title-length**: `[length[]]`
- **compare-by-text-length**: `[get[text]else[]length[]]`
- **compare-by-newest-of-modified-and-created-dates**: `[get[modified]else[19700101]]`

```
[sortsub:number<compare-by-title-length>limit[10]]
```

Try it

```
[!sortsub:number<compare-by-title-length>limit[10]]
```

Try it

```
[sortsub:number<compare-by-text-length>limit[10]]
```

Try it

```
[!sortsub:number<compare-by-text-length>limit[10]]
```

Try it

```
[tag[Field Operators]sortsub:date<compare-by-newest-of-modified-and-created-dates>]
```

Try it

You can use the search-replace Operator in combination with the sortsub Operator to ignore stop words like "A" and "The" at the beginning of

titles when sorting. (Note for example that "A Tale of Two Cities" sorts under "T" rather than "A")

```
[enlist<book-list>sortsub<compare-without-stopwords>]
```

Try it

You can also customise the listing so that "A Tale of Two Cities" is listed as "Tale of Two Cities, A" but still links to the correct tiddler.

copy to clipboard

```
<ul>
<$list filter="[enlist<book-list>sortsub<compare-without-stopwords>]">
<li><$link><$text text={{{[<currentTiddler>search-replace:i:regexp[(^The |A )(.*)],[$2, $1]]}}}/></
$link></li>
</$list>
</ul>
```

That renders as:

- *Animal Farm*
- *Dice Man, The*
- *Great Expectations*
- *New Kind of Science, A*
- *Recursive Universe, The*
- *Tale of Two Cities, A*

Spelling

17th January 2015 at 4:25pm

Improving TiddlyWiki Documentation

Because TiddlyWiki is of British origin, its English documentation uses British spelling in preference to US spelling.

Words like "customise" are spelled "-ise", not "-ize". Words like "colour" will also be spelled using UK English unless they are being used for reserved words in code, such as in CSS or JavaScript.

Standard technical acronyms are written in upper case, without dots: "HTML", not "html" or "H.T.M.L."

Avoid arbitrarily abbreviating words and sentences. But the following abbreviations are acceptable:

Abbreviation	Meaning	Notes
e.g.	for example	with a dot after each letter
i.e.	that is to say	with a dot after each letter
etc	and so on	without a dot

split Operator

13th June 2019 at 5:45pm

Filter Operators

purpose	returns each item in the list split into separate strings according to the specified separator **S** ; duplicates are *NOT* removed
input	a selection of titles
parameter	**S** = The substring at which to split each title
output	the input titles split into separate items according to the specified separator **S**

Learn more about how to use Filters

Introduced in v5.1.20

Note that in some circumstances the **split** operator will include blank items in the list of results. For example,

```
[[the band thethe are the best the]split[the]]
```

The following results are returned:

```
["", " band ", "", " are ", " best ", ""]
```

Where it might be expected that the results would be:

```
[" band ", " are ", " best "]
```

The blank items mark the boundaries between matches. If they are not required they can be removed with the **blank** category of the is Operator: `[[the band thethe are the best the]split[the]!is[blank]]`.

The reason that the blank items can be useful is that they allow search and replace operations to be constructed from a combination of the split Operator or splitregexp Operator and the join Operator. For example:

```
[[the band thethe are the best the]split[the]join[every]]
```

Try it

Examples

split Operator (Examples)

13th June 2019 at 5:44pm

Operator Examples split Operator

```
[[a b c d e]split[ ]]
```

Try it

```
[[Mississippi]split[i]]
```

Try it

splitbefore Operator

21st February 2015 at 8:37pm

Filter Operators String Operators

purpose	select a delimited prefix from each input title
input	a selection of titles
parameter	**S** = a string of characters marking the end of the prefix
output	the prefix, up to and including **S**, of each input title

Learn more about how to use Filters

Each input title is processed in turn.

- A title that contains **S** contributes everything up to and including **S**.
- A title that doesn't contain **S** simply contributes itself to the output.

Contributions are dominantly appended to the output.

The table-of-contents macros use this operator with `/` as the parameter.

Examples

splitbefore Operator (Examples)

26th January 2015 at 3:39pm

Operator Examples splitbefore Operator

```
2015-01-26 2014-07-19 2013 +[splitbefore[-]]
```

Try it

```
[[green trees]splitbefore[ee]]
```

Try it

splitregexp Operator

13th June 2019 at 5:49pm

Filter Operators

purpose	returns each item in the list split into separate strings according to the specified regular expression **R**
input	a selection of titles
suffix	flags: **m** for multiline mode, **i** for case-insensitive mode
parameter	**R** = The regular expression at which to split each title
output	the input titles split into separate items according to the specified regular expression **R**

Learn more about how to use Filters

ⓘ Introduced in v5.1.20

Note that in some circumstances the **splitregexp** operator will include blank items in the list of results. For example,

```
[[the band thethe are the best the]splitregexp[the]]
```

The following results are returned:

```
["", " band ", "", " are ", " best ", ""]
```

Where it might be expected that the results would be:

```
[" band ", " are ", " best "]
```

The blank items mark the boundaries between matches. If they are not required they can be removed with the **blank** category of the is Operator: `[[the band thethe are the best the]splitregexp[the]!is[blank]]`.

The reason that the blank items can be useful is that they allow search and replace operations to be constructed from a combination of the split Operator or splitregexp Operator and the join Operator. For example:

```
[[nobody, really; wants; to see -- all this \punctuation]splitregexp[,|;|-|\\]join[...]]
```

Try it

Syntax errors in the regular expression will cause the filter to return an error message. For example:

```
[[the cat sat on the mat]splitregexp[\]]
```

Try it

Examples

splitregexp Operator (Examples)

13th June 2019 at 5:50pm

Operator Examples splitregexp Operator

```
a=b-c=d-e +[splitregexp[=|-]]
```

Try it

standard-deviation Operator

26th April 2021 at 3:15pm

Filter Operators Mathematics Operators Reducing Mathematics Operators

purpose	treating each input title as a number, compute their standard-deviation
input	a selection of titles
output	the standard-deviation of the input as numbers

Learn more about how to use Filters

Introduced in v5.2.0 See Mathematics Operators for an overview.



The `standard-deviation` operator treats the input as a complete population and not a sample

Examples

standard-deviation Operator (Examples)

26th April 2021 at 3:15pm

Operator Examples standard-deviation Operator

```
=1 =3 =4 =5 +[standard-deviation[]]
```

Try it

Note that if there is no input the operator returns `NaN`

```
[title[NotATiddler]get[price]] +[standard-deviation[]]
```

Try it

Stanford JavaScript Crypto Library

25th June 2015 at 1:54am

Definitions

The Stanford Javascript Crypto Library is a project by the Stanford Computer Security Lab to build a secure, powerful, fast, small, easy-to-use,

cross-browser library for cryptography in Javascript.

http://bitwiseshiftleft.github.io/sjcl/

StartupActions

25th October 2020 at 5:17pm

Features

TiddlyWiki executes any ActionWidgets found in tiddlers with the following system tags during startup:

- Executed during initial startup:
  - $:/tags/StartupAction (executed on all platforms)
  - $:/tags/StartupAction/Browser (only executed when running in the browser)
  - $:/tags/StartupAction/Node (only executed when running under Node.js)
- Executed after startup rendering:
  - ⓘ **Introduced in v5.1.23** $:/tags/StartupAction/PostRender (only executed when running in the browser)

Initial Startup Actions

The initial startup actions are useful for customising TiddlyWiki according to environmental factors such as the screen size. For example, the following action widgets when placed in a tiddler tagged `$:/tags/StartupAction/Browser` will cause the sidebar to be hidden by default when the screen width is less than 1000 pixels:

```
<$reveal type="lt" state="$:/info/browser/screen/width" text="3000">
<$action-setfield $tiddler="$:/state/sidebar" $value="no"/>
```

```
</$reveal>

<$reveal type="gteq" state="$:/info/browser/screen/width" text="3000">
<$action-setfield $tiddler="$:/state/sidebar" $value="yes"/>
</$reveal>
```

Introduced in v5.1.20 This example switches to the language plugin matching the language reported by the browser:

```
<$action-setfield $tiddler="$:/language" text={{{ [[$:/languages/en-GB]] [plugin-type[language]sort[description]removeprefix[$:/languages/]] +[prefix{$:/info/browser/language}] ~[[en-GB]] +[addprefix[$:/languages/]] }}}/>
```

Note that global macros are not available within initial startup action tiddlers by default. If you need to access them then you'll need to explicitly include them with an Pragma: \import at the top of the tiddler:

```
\import [subfilter{$:/core/config/GlobalImportFilter}]
```

Post-Render Startup Actions

Introduced in v5.1.23 Post-render startup actions occur after the TiddlyWiki user interface has been rendered. This makes it possible to perform actions that depend on the rendered display (such as displaying modals).

StateMechanism

27th November 2021 at 4:39am

Mechanisms

The StateMechanism in TiddlyWiki is at the heart of how complex user interfaces can be built from WikiText.

In the browser, the TiddlyWiki display is produced by dynamically rendering the tiddler **$:/core/ui/PageTemplate**. Through various transclusions and other widgets it renders the entire user interface. The dynamic rendering is accomplished by a mechanism called "binding": any changes to the tiddlers in the store are dynamically reflected in the browser display.

The stack of templates that make up the TiddlyWiki display are complex but we'll focus on the line that displays the main story column:

```
<$list filter="[list[$:/StoryList]]" history="$:/HistoryList" template="$:/core/ui/ViewTemplate"
editTemplate="$:/core/ui/EditTemplate" storyview={{$:/view}} />
```

Breaking down the attributes applied to the list widget:

- **filter**: selects the list of tiddlers in the `list` field of the tiddler $:/StoryList. If a tiddler is added or removed from that list then it is automatically reflected in the displayed list
- **history**: references the tiddler to be used to store the history stack (see the HistoryMechanism)
- **template**: identifies a template tiddler to be used for rendering each tiddler in the list
- **editTemplate**: identifies a different template tiddler to be used for rendering tiddlers that are in draft mode
- **storyview**: specifies the story view to be used (eg classic, or zoomin)

The $:/StoryList tiddler is an example of a *StateTiddler*: a tiddler that is used to hold the state of the user interface. Changes to the user interface are made indirectly, by changing the underlying state tiddlers, and letting TiddlyWiki ripple the changes through the user interface.

Note how this approach makes the **open** tab in the sidebar very easy to implement: it is just another list widget referencing the same state tiddler, but with a different template:

```
<$list filter="[list[$:/StoryList]]" history="$:/HistoryList" storyview="pop">
<$button message="tm-close-tiddler" class="tc-btn-invisible tc-btn-mini">&times;</$button> <$link/>
</$list>
```

Now consider the implementation of the info panel within the tiddler template. We want to be able to toggle the info panel open and closed, which means that we must track its current state in a tiddler.

However, we can't track the state in a tiddler called, say, *$:/InfoPanelState* because every tiddler would share the same state; changing the value of the tiddler would affect all tiddlers displayed in the story.

The solution is to dynamically generate a unique title for each state tiddler that we need. We need to ensure that the same state tiddler title is generated each time a user interface element is rendered. To do that, we append together tokens representing each of the stack of transclusions that led to the current rendering location. Then that string of symbols is hashed to a simple numeric value.

The process of generating a state tiddler title is encapsulated in the **`qualify`** macro.

Story River

17th September 2015 at 9:40pm

Concepts

The "story river" (sometimes abbreviated to "story") is the sequence of tiddlers in the main column of the display.

Story Tiddler Template

6th December 2021 at 2:37pm

Concepts

"Story tiddler template" refers to the template used to display a tiddler within the story river.

The core story tiddler template can be found in **$:/core/ui/StoryTiddlerTemplate**

The Story Tiddler Template Cascade is used to choose the template to be used for a particular tiddler. By default, the edit template is used for tiddlers in draft mode, and the view template used otherwise.

See also:

- How to create a custom story tiddler template

Story Tiddler Template Cascade

6th December 2021 at 4:58pm

Cascades

The story tiddler template cascade is a cascade used to choose which Story Tiddler Template should be used for a particular tiddler.

The default story tiddler template cascade consists of:

1. If the tiddler is a draft tiddler, use the template title given in the tiddler **$:/config/ui/EditTemplate**
2. Otherwise, use the template title given in the tiddler **$:/config/ui/ViewTemplate**

You can see the current settings for the story tiddler template cascade in **$:/ControlPanel** under the **Info** -> **Advanced** -> **Cascades** -> **Story Tiddler** tab.

- How to create a custom story tiddler template

storyTiddler Variable

21st February 2015 at 11:25pm

Core Variables Variables

When a tiddler is viewed or edited, then within its branch of the widget tree, the **storyTiddler** variable contains the title of that tiddler.

The default **view template** and **edit template** initialise **`storyTiddler`** to the value of the **`currentTiddler`** variable. This in turn will have been set by a **`$list`** widget in **the relevant part of the page template**.

`storyTiddler` is undefined outside the story river, such as in the sidebar.

Examples

storyTiddler Variable (Examples)

21st February 2015 at 3:14pm

storyTiddler Variable Variable Examples

copy to clipboard

```
<ol>
<$list filter="[prefix[J]]">
<li><<currentTiddler>>, <<storyTiddler>></li>
</$list>
</ol>
```

Try it

storyviews Operator

3rd February 2015 at 8:14pm

Filter Operators Selection Constructors Special Operators

purpose	select the names of all the story views
input	ignored
parameter	none
output	the name of each story view

Learn more about how to use Filters

The names are those exported by modules whose **`module-type`** is **`storyview`**.

Examples

storyviews Operator (Examples)

26th January 2015 at 3:21pm

Operator Examples storyviews Operator

```
[storyviews[]]
```

Try it

stringify Operator

22nd September 2023 at 2:14pm

Filter Operators String Operators

purpose	apply JavaScript string encoding to a string, see also the similar `jsonstringify`
input	a selection of titles
suffix	`R` = ⓘ **Introduced in v5.1.23** optionally, the keyword `rawunicode`
output	the input with JavaScript string encodings applied

Learn more about how to use Filters

ⓘ **Introduced in v5.1.14**

The following substitutions are made:

Character	Replacement	Condition
`\`	`\\`	Always
`"`	`\"`	Always
`'`	`\'`	Always
Line feed (0x0a)	`\n`	Always
Carriage return (0x0d)	`\r`	Always
Characters from 0x00 to 0x1f, except listed above	`\x##` where ## is two hex digits	Always

Characters from 0x80 to 0xffff	`\u####` where #### is four hex digits	If `rawunicode` suffix is not present (default)
Characters from 0x80 to 0xffff	ⓘ **Introduced in v5.1.23** Unchanged	If `rawunicode` suffix is present

ⓘ **Introduced in v5.1.23** If the suffix `rawunicode` is present, Unicode characters above 0x80 (such as ß, ä, ñ or 🎄) will be passed through unchanged. Without the suffix, they will be substituted with `\u` codes, which was the default behavior before 5.1.23. Characters outside the Basic Multilingual Plane, such as 🎄 and other emojis, will be encoded as a UTF-16 surrogate pair, i.e. with two `\u` sequences.

Note

> Mind the differences compared to **`jsonstringify`** in encoding of single quotes and control characters (0x00 to 0x1f).

Examples

stringify Operator (Examples)

22nd September 2023 at 2:23pm

Operator Examples stringify Operator

Compare the encoding of quotes and control characters in the first example with the analogue example for the jsonstringify operator.

```
[[Backslash \, double quote ", single quote ', tab	, line feed
]] +[stringify[]]
```

Try it

```
[[Accents and emojis -> äñøßπ ⏳🎄🍪🍓 without suffix]] +[stringify[]]
```

Try it

```
[[Accents and emojis -> äñøßπ ⏳🎄🍪🍓 with rawunicode suffix]] +[stringify:rawunicode[]]
```

Try it

Structuring TiddlyWiki

24th January 2015 at 10:15pm

Working with TiddlyWiki

TiddlyWiki5 provides several features to help you structure information as tiddlers, and model the relationships between them:

- TiddlerLinks
- Tagging
- Title Lists
- DataTiddlers

Styles and Classes in WikiText

26th July 2023 at 12:57pm

How to apply custom styles WikiText

CSS styles and classes can be applied to inline or block content wrapped in `@@double at signs@@`. Classes can be applied to certain block WikiText elements.

Inline content wrapped in `@@double at signs@@` without specifying style or class will be assigned the `tc-inline-style` class and displayed as highlighted text. The foreground and background colours of the highlighted text are defined as `highlight-background` and `highlight-foreground` in the current palette.

copy to clipboard

```
@@Highlighted text@@
```

That renders as:

Highlighted text

... and the underlying HTML is:

```
<p><span class="tc-inline-style">Highlighted text</span></p>
```

Styles

Multiple style attributes, e.g. `color`, each followed by `;` semicolon can be introduced immediately after the opening `@@`, without spaces in between.

```
@@color:steelblue;background-color:lightcyan;Text with custom style@@
```

That renders as:

Text with custom style

... and the underlying HTML is:

```
<p><span style="color:steelblue;background-color:lightcyan;">Text with custom style</span></p>
```

Similarly, styles can be applied to *block content*. Wrapping block content in `@@` without specifying style or class has no effect.

copy to clipboard

```
@@background-color:lightcyan;
* Item one
* Item two
@@
```

That renders as:

- Item one
- Item two

... and the underlying HTML is:

```
<ul style="background-color:lightcyan;"><li>Item one</li><li>Item two</li></ul>
```

Classes

The following `coloured-text` and `coloured-bg` classes are defined in this tiddler for demonstration purposes:

```
.coloured-text {color: darkkhaki;}
.coloured-bg {background-color: cornsilk;}
```

Multiple classes, each prefixed with `.`, can be introduced immediately after the opening `@@`, followed by a ` ` space. This works both for inline and block content:

copy to clipboard

```
@@.coloured-text.coloured-bg Inline content with two assigned classes@@
```

That renders as:

Inline content with two assigned classes

... and the underlying HTML is:

```
<p><span class=" coloured-text coloured-bg ">Inline content with two assigned classes</span></p>
```

copy to clipboard

```
@@.coloured-bg
* Block content
* With one assigned class
@@
```

That renders as:

- Block content
- With one assigned class

... and the underlying HTML is:

```
<ul class="coloured-bg"><li>Block content</li><li>With one assigned class</li></ul>
```

Multiple classes and styles can be applied simultaneously. In case of inline content, the styles have to be defined first, followed by the classes.

copy to clipboard

```
@@font-size:1.5em;.coloured-text.coloured-bg Text with custom style and classes@@
```

That renders as:

Text with custom style and classes

... and the underlying HTML is:

```
<p><span class=" coloured-text coloured-bg " style="font-size:1.5em;">Text with custom style and classes</span></p>
```

In case of block content, the styles and classes can be defined in a single line after the opening `@@` identically as for the inline content, or in separate lines, each beginning with `@@`:

copy to clipboard

```
@@font-size:1.5em;
@@.coloured-text
@@.coloured-bg
* Block content
* With custom style and classes
@@
```

That renders as:

- Block content
- With custom style and classes

... and the underlying HTML is:

```
<ul class="coloured-text coloured-bg" style="font-size:1.5em;"><li>Block content</li><li>With custom style and classes</li></ul>
```

In a similar way classes, but not styles, can be applied to those block WikiText elements that are introduced through characters on the beginning of the line. The classes prefixed with `.` are specified immediately after the special characters, followed by a ` ` space.

copy to clipboard

```
!!!.coloured-bg Heading with a custom background class.

* Standard list element.
*.coloured-bg List element with a custom background colour class.
*.coloured-text List element with  a custom text colour class.
*.coloured-bg.coloured-text List element with both of the custom classes.
```

That renders as:

Heading with a custom background class.

- Standard list element.
- List element with a custom background colour class.
- List element with a custom text colour class.
- List element with both of the custom classes.

... and the underlying HTML is:

```
<h3 class="coloured-bg">Heading with a custom background class.</h3><ul><li>Standard list element.</
li><li class="coloured-bg">List element with a custom background colour class.</li><li class="coloured-
text">List element with  a custom text colour class.</li><li class="coloured-bg coloured-text">List
```

```
element with both of the custom classes.</li></ul>
```

Stylesheet Macros

21st February 2015 at 11:39pm

Core Macros Macros

The following core macros make it easy to specify alternative browser-specific properties when constructing a stylesheet tiddler:

`<<box-shadow shadow>>`
: for the `x-box-shadow` properties

`<<filter filter>>`
: for the `x-filter` properties

`<<transition transition>>`
: for the `x-transition` properties

`<<transform-origin origin>>`
: for the `x-transition-origin` properties

`<<background-linear-gradient gradient>>`
: for the `x-linear-gradient` values of the `background-image` property

The following macros are documented separately:

- **colour**

- `datauri`

All these macros are defined in the **$:/core/macros/CSS** tiddler.

subfilter Operator

17th December 2021 at 2:57pm

Field Operators Filter Operators Negatable Operators Selection Constructors Selection Constructors: Conditional

purpose	select titles from the parameter interpreted as a filter expression
input	a selection of titles passed as input to the subfilter
! **input**	a selection of titles passed as input to the subfilter
parameter	**S** = a filter expression
output	the selection of titles returned from the subfilter **S**
! **output**	those input titles that are *NOT* returned from the subfilter **S**

Learn more about how to use Filters

Introduced in v5.1.18 Note that the **`subfilter`** operator was introduced in version 5.1.18 and is not available in earlier versions.



Literal filter parameters cannot contain square brackets but you can work around the issue by using a variable:

```
<$set name="myFilter" value="[tag[one]]">
<$list filter="[all[tiddlers]subfilter<myFilter>]">
...
```



Compare with the similar filter operator which runs a subfilter against each title, returning those titles that return a non-empty list (and

> discards the results of the subfilter)

The **subfilter** operator will act as a constructor whenever the filter defined by its parameter is a constructor. Otherwise, it will act as a modifier.

Examples

subfilter Operator (Examples)

31st October 2021 at 12:34am

Operator Examples subfilter Operator

Literal filter parameters can be used, but such cases are better rewritten without using **subfilter**:

`[subfilter[one two three]addsuffix[!]]`

→ same as `one two three +[addsuffix[!]]`

Try it

The **subfilter** operator can be used to dynamically define parts of a filter run. This is useful for sharing a common pieces of a filter across multiple filters.

For example, this variable:

- **recent-mods**: `[has[modified]!sort[modified]limit[5]]`

can be used in one filter like this:

```
[tag[Filter Operators]subfilter<recent-mods>addsuffix[!]]
```

→ same as `[tag[Filter Operators]has[modified]!sort[modified]limit[5]addsuffix[!]]`

Try it

and in another similar filter like this:

```
[tag[ActionWidgets]subfilter<recent-mods>addprefix[!]]
```

Try it

Variables are not the only way to define dynamic subfilters. Text references can be used to load a subfilter from a tiddler field (see also Filter Parameter):

```
[subfilter{$:/StoryList!!list}limit[5]]
```

Try it

```
[subfilter{$:/StoryList!!list}subfilter<recent-mods>]
```

Try it

substitute Operator

15th June 2023 at 7:30pm

Filter Operators String Operators

purpose	returns each item in the list, replacing within each title placeholders for filters, parameters and variables with their corresponding values
input	a selection of titles
parameter	the **substitute** operator optionally accepts a variable number of parameters, see below for details
output	the input titles with placeholders for filter expressions, parameter and variables replaced with their corresponding values

Learn more about how to use Filters

New in v5.3.0

The **substitute** operator replaces any placeholders in the input titles in the following order:

1. filter expressions
2. parameters to the **substitute** operator
3. variables

placeholder syntax	description
`$n$`	Text substitution of a parameter provided to the operator, where n is the position of the parameter starting with 1 for the first parameter. Unmatched placeholders pass through unchanged.

placeholder syntax	description
`$(varname)$`	Text substitution of a variable. Undefined variables are replaced with an empty string.
`${ filter expression }$`	Text substitution of the first result of evaluating a filter expression. In other words, if the filter returns multiple titles only the first one will be used.



Placeholders that contain square bracket characters are not valid filter syntax when used directly in a filter expression. However they can be provided as input to the **`substitute`** operator as text references or variables

Examples

substitute Operator (Examples)

15th June 2023 at 1:34am

Operator Examples substitute Operator

Substitute **`substitute[]`** operator parameters

```
[[Hi, I'm $1$ and I live in $2$]substitute[Bugs Bunny],[Rabbit Hole Hill]]
```

Try it

Substitute variables

This example uses the following variables:

- name:

```
Bugs Bunny
```

- address:

```
Rabbit Hole Hill
```

```
[[Hi, I'm $(name)$ and I live in $(address)$]substitute[]]
```

Try it

Substitute variables and operator parameters

This example uses the following variable:

- time:

```
morning
```

```
[[Something in the $(time)$ at $2$ about $1$ ]substitute[Maths],[the Library]]
```

Try it

Substitute a filter expression and a variable

This example uses the following variables:

- field:

```
modified
```

- sentence:

```
This tiddler was last $(field)$ on ${[{!!modified}format:date[DDth MMM YYYY]]}$
```

```
[<sentence>substitute[]]
```

Try it

Substituted Attribute Values

15th June 2023 at 7:30pm

Widget Attributes WikiText

New in v5.3.0

Substituted attribute values can use two different styles of quoting:

- Single backticks

```
attr=`value`
```

- Triple backticks

```
attr=```value```
```

The value of the attribute will be the text denoted by the backticks with any of the placeholders for filter expressions and variables substituted with their corresponding values. Filter expression placeholders are substituted before variable placeholders, allowing for further variable substitution in their returned value.



Any other wiki syntax in that text will be left as-is.

placeholder syntax	description
`$(varname)$`	Text substitution of a variable. Undefined variables are replaced with an empty string.
`${ filter expression }$`	Text substitution with the first result of evaluating the filter expression.

Examples

Substituting a variable value into a string

copy to clipboard

```
<$text text=`Hello there this is the tiddler "$(currentTiddler)$"`/>
```

That renders as:

Hello there this is the tiddler "Substituted Attribute Values"

Substituting a variable value and the result of evaluating a filter expression into a string

copy to clipboard

```
<$text text=`This tiddler is titled "$(currentTiddler)$" and was last modified on ${[{!!modified}
format:date[DDth MMM YYYY]]}$`/>
```

That renders as:

This tiddler is titled "Substituted Attribute Values" and was last modified on 15th June 2023

Concatenating strings and variables to create a URL

copy to clipboard

```
<$let hash={{{ [<currentTiddler>encodeuricomponent[]] }}}>
<a href=`http://tiddlywiki.com/#$(hash)$`>this tiddler on tiddlywiki.com</a>
```

```
</$let>
```

That renders as:

this tiddler on tiddlywiki.com

Concatenating variables and a text reference to create a URL

copy to clipboard

```
<$let hash={{{ [<currentTiddler>encodeuricomponent[]] }}}>
<a href=`${ [{!!base-url}] }$#$(hash)$`>this tiddler on tiddlywiki.com</a>
</$let>
```

That renders as:

this tiddler on tiddlywiki.com

subtiddlerfields Operator

3rd April 2018 at 6:46pm

Filter Operators Special Operators

purpose	select all fields present in the selected tiddlers within a plugin
input	a selection of tiddler titles
parameter	**P** = the title of a plugin
output	all field names present in the selected tiddlers within the plugin named **P**

Learn more about how to use Filters

Examples

subtiddlerfields Operator (Examples)

3rd April 2018 at 6:42pm

Operator Examples plugintiddlers Operator

```
[[$:/core/copyright.txt]subtiddlerfields[$:/core]]
```

Try it

subtract Operator

11th June 2019 at 2:56pm

Binary Mathematics Operators Filter Operators Mathematics Operators

purpose	treating each input title as a number, subtract from each the numeric value of the parameter
input	a selection of titles

parameter	N = a number
output	the input as numbers, but with N subtracted from each one

Learn more about how to use Filters

Introduced in v5.1.20 See Mathematics Operators for an overview.

Examples

subtract Operator (Examples)

11th June 2019 at 2:56pm

Operator Examples subtract Operator

```
[[23]subtract[19]]
```

Try it

```
=1 =2 =3 =4 +[subtract[4]]
```

Try it

suffix Operator

11th July 2023 at 10:30am

Filter Operators Negatable Operators String Operators

purpose	filter the input titles by how they end
input	a selection of titles
suffix	the **suffix** operator uses a rich suffix, see below for details
parameter	**S** = a string of characters
output	those input titles that end with **S**
! **output**	those input tiddlers that do *NOT* end with **S**

Learn more about how to use Filters

ⓘ Introduced in v5.2.2

The **suffix** operator uses an extended syntax that permits multiple flags to be passed:

```
[suffix:<flag list>[<parameter>]]
```

- **flag list**: a comma delimited list of flags

The available flags are:

- **casesensitive**: (default), this flag forces a case-sensitive match, where upper and lower case letters are considered different
- **caseinsensitive**: overrides the default so that upper and lower case letters are considered identical for matching purposes

Examples

suffix Operator (Examples)

18th February 2022 at 3:34am

Operator Examples suffix Operator

```
[suffix[.jpg]]
```

Try it

```
[suffix:caseinsensitive[.JPG]]
```

Try it

```
[tag[task]!suffix[ing]]
```

Try it

sum Operator

11th June 2019 at 5:59pm

Filter Operators Mathematics Operators Reducing Mathematics Operators

purpose	produce the sum of the input numbers
input	a selection of titles
output	the result of adding together the input as numbers

Learn more about how to use Filters

Introduced in v5.1.20 See Mathematics Operators for an overview.

Examples

sum Operator (Examples)

11th June 2019 at 6:01pm

Operator Examples sum Operator

```
=1 =2 =3 =4 =5 +[sum[]]
```

Try it

```
[tag[HelloThere]get[text]length[]sum[]]
```

Try it

Sunday

16th November 2021 at 11:07pm

Operator Examples

This example tiddler is used to illustrate some of the Filter Operators.

SystemTag: $:/tags/AboveStory

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/AboveStory` marks elements to be placed at the top of the story river

The following tiddlers are tagged with *$:/tags/AboveStory*

	caption
$:/core/ui/AboveStory/tw2-plugin-check	$:/core/ui/AboveStory/tw2-plugin-check

SystemTag: $:/tags/AdvancedSearch

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/AdvancedSearch` marks search elements

The following tiddlers are tagged with **$:/tags/AdvancedSearch**

	caption
$:/core/ui/AdvancedSearch/Filter	Filter
$:/core/ui/AdvancedSearch/Shadows	Shadows
$:/core/ui/AdvancedSearch/Standard	Standard
$:/core/ui/AdvancedSearch/System	System

SystemTag: $:/tags/AdvancedSearch/FilterButton

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/AdvancedSearch/FilterButton` marks filter buttons

The following tiddlers are tagged with **$:/tags/AdvancedSearch/FilterButton**

	caption
$:/core/ui/AdvancedSearch/Filter/FilterButtons/clear	$:/core/ui/AdvancedSearch/Filter/FilterButtons/clear
$:/core/ui/AdvancedSearch/Filter/FilterButtons/delete	$:/core/ui/AdvancedSearch/Filter/FilterButtons/delete
$:/core/ui/AdvancedSearch/Filter/FilterButtons/dropdown	$:/core/ui/AdvancedSearch/Filter/FilterButtons/dropdown
$:/core/ui/AdvancedSearch/Filter/FilterButtons/export	$:/core/ui/AdvancedSearch/Filter/FilterButtons/export

SystemTag: $:/tags/Alert

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/Alert` marks alerts

(No tiddlers are currently tagged with *$:/tags/Alert*)

SystemTag: $:/tags/BelowStory

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/BelowStory` marks elements to be placed at the bottom of the story river

(No tiddlers are currently tagged with *$:/tags/BelowStory*)

SystemTag: $:/tags/ClassFilters/PageTemplate

20th October 2022 at 5:59am

SystemTags

The system tag `$:/tags/ClassFilters/PageTemplate` marks filters marks filters evaluated to dynamically add their output as CSS classes to the page template.

(No tiddlers are currently tagged with *$:/tags/ClassFilters/PageTemplate*)

SystemTag: $:/tags/ClassFilters/TiddlerTemplate

26th March 2023 at 5:30pm

SystemTags

The system tag `$:/tags/ClassFilters/TiddlerTemplate` marks filters evaluated to dynamically add their output as CSS classes to the tiddler template.

The variables available are within each filter:

- **storyTiddler**: the tiddler in the story for which the filter should be evaluated.
- **currentTiddler**: the tiddler holding the filter definition being evaluated.

(No tiddlers are currently tagged with *$:/tags/ClassFilters/TiddlerTemplate*)

SystemTag: $:/tags/ControlPanel

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/ControlPanel` marks control panel tabs

The following tiddlers are tagged with **$:/tags/ControlPanel**

	caption
$:/core/ui/ControlPanel/Appearance	Appearance
$:/core/ui/ControlPanel/Info	Info
$:/core/ui/ControlPanel/KeyboardShortcuts	Keyboard Shortcuts
$:/core/ui/ControlPanel/Plugins	Plugins
$:/core/ui/ControlPanel/Saving	Saving
$:/core/ui/ControlPanel/Settings	Settings

SystemTag: $:/tags/ControlPanel/Advanced

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/ControlPanel/Advanced` marks control panel advanced tabs

The following tiddlers are tagged with *$:/tags/ControlPanel/Advanced*

	caption
$:/core/ui/ControlPanel/Cascades	Cascades
$:/core/ui/ControlPanel/EditorTypes	Editor Types
$:/core/ui/ControlPanel/LoadedModules	Loaded Modules
$:/core/ui/ControlPanel/Parsing	Parsing
$:/core/ui/ControlPanel/Stylesheets	Stylesheets
$:/core/ui/ControlPanel/TiddlerFields	Tiddler Fields

SystemTag: $:/tags/ControlPanel/Appearance

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/ControlPanel/Appearance` marks control panel appearance tabs

The following tiddlers are tagged with *$:/tags/ControlPanel/Appearance*

	caption
$:/core/ui/ControlPanel/Palette	Palette
$:/core/ui/ControlPanel/StoryView	Story View
$:/core/ui/ControlPanel/Theme	Theme
$:/core/ui/ControlPanel/Toolbars	Toolbars
$:/snippets/LayoutSwitcher	Layout
$:/themes/tiddlywiki/vanilla/themetweaks	Theme Tweaks

SystemTag: $:/tags/ControlPanel/Info

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/ControlPanel/Info` marks control panel info tabs

The following tiddlers are tagged with **$:/tags/ControlPanel/Info**

	caption
$:/core/ui/ControlPanel/Advanced	Advanced
$:/core/ui/ControlPanel/Basics	Basics

SystemTag: $:/tags/ControlPanel/Plugins

7th August 2021 at 2:36pm

SystemTags

The system tag `$:/tags/ControlPanel/Plugins` marks elements to be placed under "Plugins" tab in **$:/ControlPanel**

The following tiddlers are tagged with **$:/tags/ControlPanel/Plugins**

	caption
$:/core/ui/ControlPanel/Plugins/Installed/Languages	Languages (0)
$:/core/ui/ControlPanel/Plugins/Installed/Plugins	Plugins (8)
$:/core/ui/ControlPanel/Plugins/Installed/Themes	Themes (9)

SystemTag: $:/tags/ControlPanel/Saving

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/ControlPanel/Saving` marks saving configurations

The following tiddlers are tagged with *$:/tags/ControlPanel/Saving*

	caption
$:/core/ui/ControlPanel/Saving/DownloadSaver	Download Saver
$:/core/ui/ControlPanel/Saving/General	General
$:/core/ui/ControlPanel/Saving/Gitea	Gitea Saver
$:/core/ui/ControlPanel/Saving/GitHub	GitHub Saver
$:/core/ui/ControlPanel/Saving/GitLab	GitLab Saver
$:/core/ui/ControlPanel/Saving/TiddlySpot	TiddlySpot Saver

SystemTag: $:/tags/ControlPanel/Settings

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/ControlPanel/Settings` marks control panel settings tabs

The following tiddlers are tagged with *$:/tags/ControlPanel/Settings*

	caption
$:/core/ui/ControlPanel/Settings/CamelCase	Camel Case Wiki Links

$:/core/ui/ControlPanel/Settings/DefaultMoreSidebarTab	Default More Sidebar Tab
$:/core/ui/ControlPanel/Settings/DefaultSidebarTab	Default Sidebar Tab
$:/core/ui/ControlPanel/Settings/EditorToolbar	Editor Toolbar
$:/core/ui/ControlPanel/Settings/InfoPanelMode	Tiddler Info Panel Mode
$:/core/ui/ControlPanel/Settings/LinkToBehaviour	Tiddler Opening Behaviour
$:/core/ui/ControlPanel/Settings/MissingLinks	Wiki Links
$:/core/ui/ControlPanel/Settings/NavigationAddressBar	Navigation Address Bar
$:/core/ui/ControlPanel/Settings/NavigationHistory	Navigation History
$:/core/ui/ControlPanel/Settings/NavigationPermalinkviewMode	Permalink/permaview Mode
$:/core/ui/ControlPanel/Settings/PerformanceInstrumentation	Performance Instrumentation
$:/core/ui/ControlPanel/Settings/TitleLinks	Tiddler Titles
$:/core/ui/ControlPanel/Settings/ToolbarButtons	Toolbar Buttons
$:/core/ui/ControlPanel/Settings/ToolbarButtonStyle	Toolbar Button Style

SystemTag: $:/tags/ControlPanel/Toolbars

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/ControlPanel/Toolbars` marks control panel toolbar customisation tabs

The following tiddlers are tagged with *$:/tags/ControlPanel/Toolbars*

	caption
$:/core/ui/ControlPanel/Toolbars/EditorToolbar	Editor Toolbar
$:/core/ui/ControlPanel/Toolbars/EditToolbar	Edit Toolbar
$:/core/ui/ControlPanel/Toolbars/PageControls	Page Toolbar

$:/core/ui/ControlPanel/Toolbars/ViewToolbar	View Toolbar
$:/plugins/tiddlywiki/menubar/config	Menu Bar

SystemTag: $:/tags/EditorToolbar

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/EditorToolbar` marks the editor toolbar buttons

The following tiddlers are tagged with **$:/tags/EditorToolbar**

	caption
$:/core/ui/EditorToolbar/bold	bold
$:/core/ui/EditorToolbar/clear	clear
$:/core/ui/EditorToolbar/editor-height	editor height
$:/core/ui/EditorToolbar/excise	excise
$:/core/ui/EditorToolbar/heading-1	heading 1
$:/core/ui/EditorToolbar/heading-2	heading 2
$:/core/ui/EditorToolbar/heading-3	heading 3
$:/core/ui/EditorToolbar/heading-4	heading 4
$:/core/ui/EditorToolbar/heading-5	heading 5
$:/core/ui/EditorToolbar/heading-6	heading 6
$:/core/ui/EditorToolbar/italic	italic
$:/core/ui/EditorToolbar/line-width	line width
$:/core/ui/EditorToolbar/link	link
$:/core/ui/EditorToolbar/linkify	wikilink

$:/core/ui/EditorToolbar/list-bullet	bulleted list
$:/core/ui/EditorToolbar/list-number	numbered list
$:/core/ui/EditorToolbar/mono-block	monospaced block
$:/core/ui/EditorToolbar/mono-line	monospaced
$:/core/ui/EditorToolbar/more	more
$:/core/ui/EditorToolbar/opacity	opacity
$:/core/ui/EditorToolbar/paint	paint colour
$:/core/ui/EditorToolbar/picture	picture
$:/core/ui/EditorToolbar/preview	preview
$:/core/ui/EditorToolbar/preview-type	preview type
$:/core/ui/EditorToolbar/quote	quote
$:/core/ui/EditorToolbar/rotate-left	rotate left
$:/core/ui/EditorToolbar/size	image size
$:/core/ui/EditorToolbar/stamp	stamp
$:/core/ui/EditorToolbar/strikethrough	strikethrough
$:/core/ui/EditorToolbar/subscript	subscript
$:/core/ui/EditorToolbar/superscript	superscript
$:/core/ui/EditorToolbar/transcludify	transclusion
$:/core/ui/EditorToolbar/underline	underline

SystemTag: $:/tags/EditorTools

19th May 2021 at 6:05pm

SystemTags

The system tag `$:/tags/EditorTools` can be used to include non-button UI elements in the edit mode tiddler toolbar. For buttons see

SystemTag: $:/tags/EditToolbar.

The following tiddlers are tagged with *$:/tags/EditorTools*

	caption
$:/core/ui/EditorToolbar/file-import	$:/core/ui/EditorToolbar/file-import

SystemTag: $:/tags/EditPreview

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/EditPreview` marks custom preview panes

The following tiddlers are tagged with *$:/tags/EditPreview*

	caption
$:/core/ui/EditTemplate/body/preview/diffs-current	differences from current
$:/core/ui/EditTemplate/body/preview/diffs-shadow	differences from shadow (if any)
$:/core/ui/EditTemplate/body/preview/output	output
$:/plugins/tiddlywiki/internals/EditTemplate/body/preview/parse-tree	parse tree
$:/plugins/tiddlywiki/internals/EditTemplate/body/preview/raw	raw HTML
$:/plugins/tiddlywiki/internals/EditTemplate/body/preview/widget-tree	widget tree

SystemTag: $:/tags/EditTemplate

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/EditTemplate` marks the edit template

The following tiddlers are tagged with **$:/tags/EditTemplate**

	caption
$:/ContributionBanner	$:/ContributionBanner
$:/core/ui/EditTemplate/body	$:/core/ui/EditTemplate/body
$:/core/ui/EditTemplate/controls	$:/core/ui/EditTemplate/controls
$:/core/ui/EditTemplate/fields	$:/core/ui/EditTemplate/fields
$:/core/ui/EditTemplate/shadow	$:/core/ui/EditTemplate/shadow
$:/core/ui/EditTemplate/tags	$:/core/ui/EditTemplate/tags
$:/core/ui/EditTemplate/title	$:/core/ui/EditTemplate/title
$:/core/ui/EditTemplate/type	$:/core/ui/EditTemplate/type
$:/core/ui/ViewTemplate/classic	$:/core/ui/ViewTemplate/classic

SystemTag: $:/tags/EditToolbar

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/EditToolbar` marks the edit mode tiddler toolbar

The following tiddlers are tagged with **$:/tags/EditToolbar**

	caption
$:/core/ui/Buttons/cancel	⊗ cancel
$:/core/ui/Buttons/delete	🗑 delete

$:/core/ui/Buttons/save	✓ ok

SystemTag: $:/tags/Exporter

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/Exporter` marks the exporters

The following tiddlers are tagged with *$:/tags/Exporter*

	caption
$:/core/templates/exporters/CsvFile	CSV file
$:/core/templates/exporters/JsonFile	JSON file
$:/core/templates/exporters/StaticRiver	Static HTML
$:/core/templates/exporters/TidFile	TID text file

SystemTag: $:/tags/Filter

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/Filter` marks filters in advanced search sample filter dropdown

The following tiddlers are tagged with *$:/tags/Filter*

	caption
$:/core/Filters/AllTags	All tags except system tags

$:/core/Filters/AllTiddlers	All tiddlers except system tiddlers
$:/core/Filters/Drafts	Draft tiddlers
$:/core/Filters/Missing	Missing tiddlers
$:/core/Filters/Orphans	Orphan tiddlers
$:/core/Filters/OverriddenShadowTiddlers	Overridden shadow tiddlers
$:/core/Filters/RecentSystemTiddlers	Recently modified tiddlers, including system tiddlers
$:/core/Filters/RecentTiddlers	Recently modified tiddlers
$:/core/Filters/SessionTiddlers	Tiddlers modified since the wiki was loaded
$:/core/Filters/ShadowTiddlers	Shadow tiddlers
$:/core/Filters/StoryList	Tiddlers in the story river, excluding $:/AdvancedSearch
$:/core/Filters/SystemTags	System tags
$:/core/Filters/SystemTiddlers	System tiddlers
$:/core/Filters/TypedTiddlers	Non wiki-text tiddlers

SystemTag: $:/tags/Global

19th April 2023 at 12:31pm

SystemTags

The system tag `$:/tags/Global` marks global definitions that are made available everywhere.

The following tiddlers are tagged with *$:/tags/Global*

	caption
WidgetMessage: tm-http-request Example - Random Dog	WidgetMessage: tm-http-request Example - Random Dog
WidgetMessage: tm-http-request Example - Zotero	WidgetMessage: tm-http-request Example - Zotero

SystemTag: $:/tags/Global/View

19th April 2023 at 12:31pm

SystemTags

The system tag `$:/tags/Global/View` marks definitions that are only made available within the main view templates and the preview panel.

(No tiddlers are currently tagged with *$:/tags/Global/View*)

SystemTag: $:/tags/Global/View/Body

19th April 2023 at 12:31pm

SystemTags

The system tag `$:/tags/Global/View/Body` marks definitions that are only made available within the main view template bodies and the preview panel.

(No tiddlers are currently tagged with *$:/tags/Global/View/Body*)

SystemTag: $:/tags/Image

18th October 2021 at 12:23pm

SystemTags

The system tag `$:/tags/Image` marks core images. An overview of system icons can be found in the Icon Gallery

The following tiddlers are tagged with *$:/tags/Image*

	caption

$:/core/icon	$:/core/icon
$:/core/images/add-comment	$:/core/images/add-comment
$:/core/images/advanced-search-button	$:/core/images/advanced-search-button
$:/core/images/auto-height	$:/core/images/auto-height
$:/core/images/blank	$:/core/images/blank
$:/core/images/bold	$:/core/images/bold
$:/core/images/cancel-button	$:/core/images/cancel-button
$:/core/images/chevron-down	$:/core/images/chevron-down
$:/core/images/chevron-left	$:/core/images/chevron-left
$:/core/images/chevron-right	$:/core/images/chevron-right
$:/core/images/chevron-up	$:/core/images/chevron-up
$:/core/images/clone-button	$:/core/images/clone-button
$:/core/images/close-all-button	$:/core/images/close-all-button
$:/core/images/close-button	$:/core/images/close-button
$:/core/images/close-others-button	$:/core/images/close-others-button
$:/core/images/copy-clipboard	$:/core/images/copy-clipboard
$:/core/images/delete-button	$:/core/images/delete-button
$:/core/images/done-button	$:/core/images/done-button
$:/core/images/down-arrow	$:/core/images/down-arrow
$:/core/images/download-button	$:/core/images/download-button
$:/core/images/edit-button	$:/core/images/edit-button
$:/core/images/erase	$:/core/images/erase
$:/core/images/excise	$:/core/images/excise
$:/core/images/export-button	$:/core/images/export-button
$:/core/images/file	$:/core/images/file
$:/core/images/fixed-height	$:/core/images/fixed-height

$:/core/images/fold-all-button	$:/core/images/fold-all-button
$:/core/images/fold-button	$:/core/images/fold-button
$:/core/images/fold-others-button	$:/core/images/fold-others-button
$:/core/images/folder	$:/core/images/folder
$:/core/images/full-screen-button	$:/core/images/full-screen-button
$:/core/images/github	$:/core/images/github
$:/core/images/gitter	$:/core/images/gitter
$:/core/images/globe	$:/core/images/globe
$:/core/images/heading-1	$:/core/images/heading-1
$:/core/images/heading-2	$:/core/images/heading-2
$:/core/images/heading-3	$:/core/images/heading-3
$:/core/images/heading-4	$:/core/images/heading-4
$:/core/images/heading-5	$:/core/images/heading-5
$:/core/images/heading-6	$:/core/images/heading-6
$:/core/images/help	$:/core/images/help
$:/core/images/home-button	$:/core/images/home-button
$:/core/images/import-button	$:/core/images/import-button
$:/core/images/info-button	$:/core/images/info-button
$:/core/images/italic	$:/core/images/italic
$:/core/images/layout-button	$:/core/images/layout-button
$:/core/images/left-arrow	$:/core/images/left-arrow
$:/core/images/line-width	$:/core/images/line-width
$:/core/images/link	$:/core/images/link
$:/core/images/linkify	$:/core/images/linkify
$:/core/images/list	$:/core/images/list
$:/core/images/list-bullet	$:/core/images/list-bullet

$:/core/images/list-number	$:/core/images/list-number
$:/core/images/locked-padlock	$:/core/images/locked-padlock
$:/core/images/mail	$:/core/images/mail
$:/core/images/mastodon	$:/core/images/mastodon
$:/core/images/menu-button	$:/core/images/menu-button
$:/core/images/minus-button	$:/core/images/minus-button
$:/core/images/mono-block	$:/core/images/mono-block
$:/core/images/mono-line	$:/core/images/mono-line
$:/core/images/network-activity	$:/core/images/network-activity
$:/core/images/new-button	$:/core/images/new-button
$:/core/images/new-here-button	$:/core/images/new-here-button
$:/core/images/new-image-button	$:/core/images/new-image-button
$:/core/images/new-journal-button	$:/core/images/new-journal-button
$:/core/images/opacity	$:/core/images/opacity
$:/core/images/open-window	$:/core/images/open-window
$:/core/images/options-button	$:/core/images/options-button
$:/core/images/paint	$:/core/images/paint
$:/core/images/palette	$:/core/images/palette
$:/core/images/permalink-button	$:/core/images/permalink-button
$:/core/images/permaview-button	$:/core/images/permaview-button
$:/core/images/picture	$:/core/images/picture
$:/core/images/plugin-generic-language	$:/core/images/plugin-generic-language
$:/core/images/plugin-generic-plugin	$:/core/images/plugin-generic-plugin
$:/core/images/plugin-generic-theme	$:/core/images/plugin-generic-theme
$:/core/images/plus-button	$:/core/images/plus-button
$:/core/images/preview-closed	$:/core/images/preview-closed

$:/core/images/preview-open	$:/core/images/preview-open
$:/core/images/print-button	$:/core/images/print-button
$:/core/images/quote	$:/core/images/quote
$:/core/images/refresh-button	$:/core/images/refresh-button
$:/core/images/right-arrow	$:/core/images/right-arrow
$:/core/images/rotate-left	$:/core/images/rotate-left
$:/core/images/save-button	$:/core/images/save-button
$:/core/images/save-button-dynamic	$:/core/images/save-button-dynamic
$:/core/images/size	$:/core/images/size
$:/core/images/spiral	$:/core/images/spiral
$:/core/images/stamp	$:/core/images/stamp
$:/core/images/star-filled	$:/core/images/star-filled
$:/core/images/storyview-classic	$:/core/images/storyview-classic
$:/core/images/storyview-pop	$:/core/images/storyview-pop
$:/core/images/storyview-zoomin	$:/core/images/storyview-zoomin
$:/core/images/strikethrough	$:/core/images/strikethrough
$:/core/images/subscript	$:/core/images/subscript
$:/core/images/superscript	$:/core/images/superscript
$:/core/images/tag-button	$:/core/images/tag-button
$:/core/images/theme-button	$:/core/images/theme-button
$:/core/images/timestamp-off	$:/core/images/timestamp-off
$:/core/images/timestamp-on	$:/core/images/timestamp-on
$:/core/images/tip	$:/core/images/tip
$:/core/images/transcludify	$:/core/images/transcludify
$:/core/images/twitter	$:/core/images/twitter
$:/core/images/underline	$:/core/images/underline

$:/core/images/unfold-all-button	$:/core/images/unfold-all-button
$:/core/images/unfold-button	$:/core/images/unfold-button
$:/core/images/unlocked-padlock	$:/core/images/unlocked-padlock
$:/core/images/up-arrow	$:/core/images/up-arrow
$:/core/images/video	$:/core/images/video
$:/core/images/warning	$:/core/images/warning
$:/plugins/tiddlywiki/qrcode/icon	$:/plugins/tiddlywiki/qrcode/icon

SystemTag: $:/tags/ImportPreview

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/ImportPreview` marks preview types for the import listing

The following tiddlers are tagged with *$:/tags/ImportPreview*

	caption
$:/core/ui/ImportPreviews/Diff	Diff
$:/core/ui/ImportPreviews/DiffFields	Diff (Fields)
$:/core/ui/ImportPreviews/Fields	Fields
$:/core/ui/ImportPreviews/Text	Text
$:/core/ui/ImportPreviews/TextRaw	Text (Raw)

SystemTag: $:/tags/KeyboardShortcut

SystemTags

The system tag `$:/tags/KeyboardShortcut` defines the tagged Tiddler as a Keyboard Shortcut Tiddler, a **container for actions** that get triggered when its corresponding Keyboard Shortcut Descriptor in its `key` field matches a keyboard combination executed by the User

The following tiddlers are tagged with *$:/tags/KeyboardShortcut*

	caption
$:/core/ui/KeyboardShortcuts/advanced-search	$:/core/ui/KeyboardShortcuts/advanced-search
$:/core/ui/KeyboardShortcuts/change-sidebar-layout	$:/core/ui/KeyboardShortcuts/change-sidebar-layout
$:/core/ui/KeyboardShortcuts/new-image	$:/core/ui/KeyboardShortcuts/new-image
$:/core/ui/KeyboardShortcuts/new-journal	$:/core/ui/KeyboardShortcuts/new-journal
$:/core/ui/KeyboardShortcuts/new-tiddler	$:/core/ui/KeyboardShortcuts/new-tiddler
$:/core/ui/KeyboardShortcuts/save-wiki	$:/core/ui/KeyboardShortcuts/save-wiki
$:/core/ui/KeyboardShortcuts/sidebar-search	$:/core/ui/KeyboardShortcuts/sidebar-search
$:/core/ui/KeyboardShortcuts/switcher	$:/core/ui/KeyboardShortcuts/switcher
$:/core/ui/KeyboardShortcuts/toggle-sidebar	$:/core/ui/KeyboardShortcuts/toggle-sidebar

SystemTag: $:/tags/Layout

23rd November 2020 at 8:21pm

SystemTags

The system tag `$:/tags/Layout` marks alternative page layouts that can be switched to.

(No tiddlers are currently tagged with *$:/tags/Layout*)

SystemTag: $:/tags/Macro

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/Macro` marks global macros. It is now deprecated in favour of SystemTag: $:/tags/Global.

The following tiddlers are tagged with *$:/tags/Macro*

	caption
$:/core/macros/colour-picker	$:/core/macros/colour-picker
$:/core/macros/copy-to-clipboard	$:/core/macros/copy-to-clipboard
$:/core/macros/CSS	$:/core/macros/CSS
$:/core/macros/diff	$:/core/macros/diff
$:/core/macros/dumpvariables	$:/core/macros/dumpvariables
$:/core/macros/export	$:/core/macros/export
$:/core/macros/image-picker	$:/core/macros/image-picker
$:/core/macros/keyboard-driven-input	$:/core/macros/keyboard-driven-input
$:/core/macros/lingo	$:/core/macros/lingo
$:/core/macros/list	$:/core/macros/list
$:/core/macros/tabs	$:/core/macros/tabs
$:/core/macros/tag	$:/core/macros/tag
$:/core/macros/tag-picker	$:/core/macros/tag-picker
$:/core/macros/thumbnails	$:/core/macros/thumbnails
$:/core/macros/timeline	$:/core/macros/timeline
$:/core/macros/toc	$:/core/macros/toc
$:/core/macros/translink	$:/core/macros/translink

$:/core/macros/tree	$:/core/macros/tree
$:/edition/tw5.com/dropbox-url.js	$:/edition/tw5.com/dropbox-url.js
$:/editions/tw5.com/doc-macros	$:/editions/tw5.com/doc-macros
$:/editions/tw5.com/operator-macros	$:/editions/tw5.com/operator-macros
$:/editions/tw5.com/style-guide-macros	$:/editions/tw5.com/style-guide-macros
$:/editions/tw5.com/variable-macros	$:/editions/tw5.com/variable-macros
$:/editions/tw5.com/version-macros	$:/editions/tw5.com/version-macros
$:/editions/tw5.com/wikitext-macros	$:/editions/tw5.com/wikitext-macros

SystemTag: $:/tags/Macro/View

3rd January 2020 at 10:26am

SystemTags

The system tag `$:/tags/Macro/View` marks macros that are only made available within the main view templates and the preview panel. It is now deprecated in favour of *SystemTag $:/tags/Global/View*.

(No tiddlers are currently tagged with *$:/tags/Macro/View*)

SystemTag: $:/tags/Macro/View/Body

5th May 2022 at 10:29am

SystemTags

The system tag `$:/tags/Macro/View/Body` marks macros that are only made available within the main view template bodies and the preview panel. It is now deprecated in favour of *SystemTag $:/tags/Global/View/Body*.

(No tiddlers are currently tagged with *$:/tags/Macro/View/Body*)

SystemTag: $:/tags/Manager/ItemMain

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/Manager/ItemMain` marks the tiddler manager. **$:/Manager**

The following tiddlers are tagged with **$:/tags/Manager/ItemMain**

	caption
$:/Manager/ItemMain/Fields	Fields
$:/Manager/ItemMain/RawText	Raw text
$:/Manager/ItemMain/WikifiedText	Wikified text

SystemTag: $:/tags/Manager/ItemSidebar

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/Manager/ItemSidebar` marks the tiddler manager

The following tiddlers are tagged with **$:/tags/Manager/ItemSidebar**

	caption
$:/Manager/ItemSidebar/Colour	Colour
$:/Manager/ItemSidebar/Icon	Icon
$:/Manager/ItemSidebar/Tags	Tags
$:/Manager/ItemSidebar/Tools	Tools

SystemTag: $:/tags/MoreSideBar

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/MoreSideBar` marks tabs in the 'more' sidebar

The following tiddlers are tagged with **$:/tags/MoreSideBar**

	caption
$:/core/ui/MoreSideBar/All	All
$:/core/ui/MoreSideBar/Drafts	Drafts
$:/core/ui/MoreSideBar/Explorer	Explorer
$:/core/ui/MoreSideBar/Missing	Missing
$:/core/ui/MoreSideBar/Orphans	Orphans
$:/core/ui/MoreSideBar/Plugins	Plugins
$:/core/ui/MoreSideBar/Recent	Recent
$:/core/ui/MoreSideBar/Shadows	Shadows
$:/core/ui/MoreSideBar/System	System
$:/core/ui/MoreSideBar/Tags	Tags
$:/core/ui/MoreSideBar/Types	Types

SystemTag: $:/tags/MoreSideBar/Plugins

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/MoreSideBar/Plugins` marks Sidebar: More: Plugins tab

The following tiddlers are tagged with *$:/tags/MoreSideBar/Plugins*

	caption
$:/core/ui/MoreSideBar/Plugins/Languages	Languages
$:/core/ui/MoreSideBar/Plugins/Plugins	Plugins
$:/core/ui/MoreSideBar/Plugins/Theme	Themes

SystemTag: $:/tags/PageControls

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/PageControls` marks the page control tools in the sidebar

The following tiddlers are tagged with **$:/tags/PageControls**

	caption
$:/core/ui/Buttons/advanced-search	advanced search
$:/core/ui/Buttons/close-all	close all
$:/core/ui/Buttons/control-panel	control panel
$:/core/ui/Buttons/encryption	encryption
$:/core/ui/Buttons/export-page	export all
$:/core/ui/Buttons/fold-all	fold all tiddlers
$:/core/ui/Buttons/full-screen	full-screen
$:/core/ui/Buttons/home	home

$:/core/ui/Buttons/import	import
$:/core/ui/Buttons/language	language
$:/core/ui/Buttons/layout	layout
$:/core/ui/Buttons/manager	tiddler manager
$:/core/ui/Buttons/more-page-actions	more
$:/core/ui/Buttons/network-activity	network activity
$:/core/ui/Buttons/new-image	new image
$:/core/ui/Buttons/new-journal	new journal
$:/core/ui/Buttons/new-tiddler	new tiddler
$:/core/ui/Buttons/palette	palette
$:/core/ui/Buttons/permaview	permaview
$:/core/ui/Buttons/print	print page
$:/core/ui/Buttons/refresh	refresh
$:/core/ui/Buttons/save-wiki	save changes
$:/core/ui/Buttons/storyview	storyview
$:/core/ui/Buttons/tag-manager	tag manager
$:/core/ui/Buttons/theme	theme
$:/core/ui/Buttons/timestamp	timestamps
$:/core/ui/Buttons/unfold-all	unfold all tiddlers

SystemTag: $:/tags/PageTemplate

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/PageTemplate` marks the main page elements

The following tiddlers are tagged with **$:/tags/PageTemplate**

	caption
$:/core/ui/PageTemplate/alerts	$:/core/ui/PageTemplate/alerts
$:/core/ui/PageTemplate/drafts	$:/core/ui/PageTemplate/drafts
$:/core/ui/PageTemplate/pluginreloadwarning	$:/core/ui/PageTemplate/pluginreloadwarning
$:/core/ui/PageTemplate/sidebar	$:/core/ui/PageTemplate/sidebar
$:/core/ui/PageTemplate/story	$:/core/ui/PageTemplate/story
$:/plugins/tiddlywiki/menubar/menu	$:/plugins/tiddlywiki/menubar/menu

SystemTag: $:/tags/Palette

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/Palette` marks colour palettes

The following tiddlers are tagged with *$:/tags/Palette*

	caption
$:/palettes/Blanca	A clean white palette to let you focus
$:/palettes/Blue	A blue theme
$:/palettes/ContrastDark	High contrast and unambiguous (dark version)
$:/palettes/ContrastLight	High contrast and unambiguous (light version)
$:/palettes/CupertinoDark	A macOS inspired dark palette
$:/palettes/DarkPhotos	Good with dark photo backgrounds
$:/palettes/DesertSand	A desert sand palette

$:/palettes/GruvboxDark	Retro groove color scheme
$:/palettes/Muted	Bright tiddlers on a muted background
$:/palettes/Nord	An arctic, north-bluish color palette.
$:/palettes/Rocker	A dark theme
$:/palettes/SolarFlare	Warm, relaxing earth colours
$:/palettes/SolarizedDark	Precision dark colors for machines and people
$:/palettes/SolarizedLight	Precision colors for machines and people
$:/palettes/SpartanDay	Cold, spartan day colors
$:/palettes/SpartanNight	Dark spartan colors
$:/palettes/Twilight	Delightful, soft darkness.
$:/palettes/Vanilla	Pale and unobtrusive

SystemTag: $:/tags/PluginLibrary

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/PluginLibrary` marks the plugin library

The following tiddlers are tagged with **$:/tags/PluginLibrary**

	caption
$:/config/OfficialPluginLibrary	Official TiddlyWiki Plugin Library

SystemTag: $:/tags/RawMarkup

27th September 2018 at 10:43am

SystemTags

The system tag `$:/tags/RawMarkup` marks plain text tiddlers to be included as raw markup at the bottom of the `<head>` section in the generated HTML file.

Note that the wiki must be saved and reloaded in order for a raw markup tiddler to take effect.

(No tiddlers are currently tagged with *$:/tags/RawMarkup*)

SystemTag: $:/tags/RawMarkupWikified

27th September 2018 at 10:43am

SystemTags

The system tag `$:/tags/RawMarkupWikified` marks wikified tiddlers to be included as raw markup at the bottom of the `<head>` section in the generated HTML file.

Note that the wiki must be saved and reloaded in order for a raw markup tiddler to take effect. Also note that global macros are not available by default within raw markup tiddlers; you can use the ImportVariablesWidget to explicitly import them.

(No tiddlers are currently tagged with *$:/tags/RawMarkupWikified*)

SystemTag: $:/tags/RawMarkupWikified/BottomBody

28th September 2018 at 4:58pm

SystemTags

ⓘ **Introduced in v5.1.18** The system tag `$:/tags/RawMarkupWikified/BottomBody` marks wikified tiddlers to be included as raw markup at the bottom of the `<body>` section in the generated HTML file.

Note that the wiki must be saved and reloaded in order for a raw markup tiddler to take effect. Also note that global macros are not available by default within raw markup tiddlers; you can use the ImportVariablesWidget to explicitly import them.

(No tiddlers are currently tagged with *$:/tags/RawMarkupWikified/BottomBody*)

SystemTag: $:/tags/RawMarkupWikified/TopBody

28th September 2018 at 4:58pm

SystemTags

Introduced in v5.1.18 The system tag `$:/tags/RawMarkupWikified/TopBody` marks wikified tiddlers to be included as raw markup at the top of the `<body>` section in the generated HTML file.

Note that the wiki must be saved and reloaded in order for a raw markup tiddler to take effect. Also note that global macros are not available by default within raw markup tiddlers; you can use the ImportVariablesWidget to explicitly import them.

The following tiddlers are tagged with *$:/tags/RawMarkupWikified/TopBody*

	caption
$:/SplashScreen	$:/SplashScreen

SystemTag: $:/tags/RawMarkupWikified/TopHead

28th September 2018 at 4:57pm

SystemTags

Introduced in v5.1.18 The system tag `$:/tags/RawMarkupWikified/TopHead` marks wikified tiddlers to be included as raw markup at the top of the `<head>` section in the generated HTML file.

Note that the wiki must be saved and reloaded in order for a raw markup tiddler to take effect. Also note that global macros are not available by default within raw markup tiddlers; you can use the ImportVariablesWidget to explicitly import them.

(No tiddlers are currently tagged with *$:/tags/RawMarkupWikified/TopHead*)

SystemTag: $:/tags/RawStaticContent

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/RawStaticContent` marks raw content to be saved into the TW code

(No tiddlers are currently tagged with *$:/tags/RawStaticContent*)

SystemTag: $:/tags/RemoteAssetInfo

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/RemoteAssetInfo` marks plugin import handling

(No tiddlers are currently tagged with *$:/tags/RemoteAssetInfo*)

SystemTag: $:/tags/SearchResults

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/SearchResults` marks customised search results

The following tiddlers are tagged with *$:/tags/SearchResults*

	caption
$:/core/ui/DefaultSearchResultList	List

SystemTag: $:/tags/ServerConnection

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/ServerConnection` marks plugin import handling

(No tiddlers are currently tagged with *$:/tags/ServerConnection*)

SystemTag: $:/tags/SideBar

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/SideBar` marks sidebar tabs

The following tiddlers are tagged with **$:/tags/SideBar**

	caption
$:/core/ui/SideBar/More	More
$:/core/ui/SideBar/Open	Open
$:/core/ui/SideBar/Recent	Recent
$:/core/ui/SideBar/Tools	Tools
TableOfContents	Contents

SystemTag: $:/tags/SideBarSegment

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/SideBarSegment` marks sidebar segments

Note: You can rearrange the segments by drag-and-dropping the titles inside the tag pill $:/tags/SideBarSegment

The following tiddlers are tagged with **$:/tags/SideBarSegment**

	caption
$:/core/ui/SideBarSegments/page-controls	$:/core/ui/SideBarSegments/page-controls
$:/core/ui/SideBarSegments/search	$:/core/ui/SideBarSegments/search
$:/core/ui/SideBarSegments/site-subtitle	$:/core/ui/SideBarSegments/site-subtitle
$:/core/ui/SideBarSegments/site-title	$:/core/ui/SideBarSegments/site-title
$:/core/ui/SideBarSegments/tabs	$:/core/ui/SideBarSegments/tabs

SystemTag: $:/tags/StartupAction

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/StartupAction` marks actions executed on all platforms

(No tiddlers are currently tagged with *$:/tags/StartupAction*)

SystemTag: $:/tags/StartupAction/Browser

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/StartupAction/Browser` marks actions only executed when running in the browser

(No tiddlers are currently tagged with *$:/tags/StartupAction/Browser*)

SystemTag: $:/tags/StartupAction/Node

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/StartupAction/Node` marks actions only executed when running under Node.js

(No tiddlers are currently tagged with *$:/tags/StartupAction/Node*)

SystemTag: $:/tags/Stylesheet

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/Stylesheet` marks that a tiddler should be applied as a CSS stylesheet

The following tiddlers are tagged with *$:/tags/Stylesheet*

	caption
$:/_tw_shared/styles	$:/_tw_shared/styles

$:/_tw5.com-styles	$:/_tw5.com-styles
$:/_tw5.com/CustomStoryTiddlerTemplateDemo/Styles	$:/_tw5.com/CustomStoryTiddlerTemplateDemo/Styles
$:/editions/tw5.com/doc-styles	$:/editions/tw5.com/doc-styles
$:/plugins/tiddlywiki/internals/styles	$:/plugins/tiddlywiki/internals/styles
$:/plugins/tiddlywiki/menubar/styles	$:/plugins/tiddlywiki/menubar/styles
$:/plugins/tiddlywiki/qrcode/styles	$:/plugins/tiddlywiki/qrcode/styles
$:/plugins/tiddlywiki/railroad/railroad-diagrams.css	$:/plugins/tiddlywiki/railroad/railroad-diagrams.css
$:/SavingThumbnailsStyles	$:/SavingThumbnailsStyles
$:/themes/tiddlywiki/snowwhite/base	$:/themes/tiddlywiki/snowwhite/base
$:/themes/tiddlywiki/vanilla/base	$:/themes/tiddlywiki/vanilla/base
Custom data-styles	Custom data-styles

SystemTag: $:/tags/TagDropdown

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/TagDropdown` marks tags dropdown sorting

(No tiddlers are currently tagged with *$:/tags/TagDropdown*)

SystemTag: $:/tags/TextEditor/Snippet

28th August 2020 at 5:41pm

SystemTags

The system tag `$:/tags/TextEditor/Snippet` marks text snippets

The following tiddlers are tagged with *$:/tags/TextEditor/Snippet*

	caption
$:/language/Snippets/ListByTag	List of tiddlers by tag
$:/language/Snippets/MacroDefinition	Macro definition
$:/language/Snippets/Table4x3	Table with 4 columns by 3 rows
$:/language/Snippets/TableOfContents	Table of Contents

SystemTag: $:/tags/TiddlerInfo

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/TiddlerInfo` marks tiddler info panel tabs

The following tiddlers are tagged with **$:/tags/TiddlerInfo**

	caption
$:/core/ui/TiddlerInfo/Advanced	Advanced
$:/core/ui/TiddlerInfo/Fields	Fields
$:/core/ui/TiddlerInfo/List	List
$:/core/ui/TiddlerInfo/Listed	Listed
$:/core/ui/TiddlerInfo/References	Backlinks
$:/core/ui/TiddlerInfo/Tagging	Tagging
$:/core/ui/TiddlerInfo/Tools	Tools
$:/editions/tw5.com/TiddlerInfo/Sources	Sources

SystemTag: $:/tags/TiddlerInfo/Advanced

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/TiddlerInfo/Advanced` marks tabs under the advanced tiddler tab

The following tiddlers are tagged with **$:/tags/TiddlerInfo/Advanced**

	caption
$:/core/ui/TiddlerInfo/Advanced/PluginInfo	$:/core/ui/TiddlerInfo/Advanced/PluginInfo
$:/core/ui/TiddlerInfo/Advanced/ShadowInfo	$:/core/ui/TiddlerInfo/Advanced/ShadowInfo

SystemTag: $:/tags/TiddlerInfoSegment

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/TiddlerInfoSegment` marks tiddler info panel segments

(No tiddlers are currently tagged with *$:/tags/TiddlerInfoSegment*)

SystemTag: $:/tags/ToolbarButtonStyle

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/ToolbarButtonStyle` marks the style of the buttons

The following tiddlers are tagged with *$:/tags/ToolbarButtonStyle*

	caption
$:/core/buttonstyles/Borderless	Borderless
$:/core/buttonstyles/Boxed	Boxed
$:/core/buttonstyles/Rounded	Rounded

SystemTag: $:/tags/TopLeftBar

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/TopLeftBar` marks the top left bar

(No tiddlers are currently tagged with *$:/tags/TopLeftBar*)

SystemTag: $:/tags/TopRightBar

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/TopRightBar` marks the top right bar

The following tiddlers are tagged with *$:/tags/TopRightBar*

	caption
$:/core/ui/TopBar/menu	$:/core/ui/TopBar/menu

SystemTag: $:/tags/ViewTemplate

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/ViewTemplate` identifies the individual segments that are displayed as part of the view template

The following tiddlers are tagged with **$:/tags/ViewTemplate**

	caption
$:/core/ui/ViewTemplate/body	$:/core/ui/ViewTemplate/body
$:/core/ui/ViewTemplate/classic	$:/core/ui/ViewTemplate/classic
$:/core/ui/ViewTemplate/subtitle	$:/core/ui/ViewTemplate/subtitle
$:/core/ui/ViewTemplate/tags	$:/core/ui/ViewTemplate/tags
$:/core/ui/ViewTemplate/title	$:/core/ui/ViewTemplate/title
$:/core/ui/ViewTemplate/unfold	$:/core/ui/ViewTemplate/unfold
$:/editions/tw5.com/filter-run-template	$:/editions/tw5.com/filter-run-template
$:/editions/tw5.com/operator-template	$:/editions/tw5.com/operator-template
$:/editions/tw5.com/systemtag-template	$:/editions/tw5.com/systemtag-template

SystemTag: $:/tags/ViewTemplate/Subtitle

14th July 2022 at 11:32am

SystemTags

The system tag `$:/tags/ViewTemplate/Subtitle` identifies the individual segments that are displayed as the tiddler subtitle

The following tiddlers are tagged with **$:/tags/ViewTemplate/Subtitle**

	caption
$:/core/ui/ViewTemplate/subtitle/modified	$:/core/ui/ViewTemplate/subtitle/modified
$:/core/ui/ViewTemplate/subtitle/modifier	$:/core/ui/ViewTemplate/subtitle/modifier

SystemTag: $:/tags/ViewToolbar

26th September 2018 at 7:14pm

SystemTags

The system tag `$:/tags/ViewToolbar` marks the view mode tiddler toolbar

The following tiddlers are tagged with **$:/tags/ViewToolbar**

	caption
$:/core/ui/Buttons/clone	clone
$:/core/ui/Buttons/close	close
$:/core/ui/Buttons/close-others	close others
$:/core/ui/Buttons/delete	delete
$:/core/ui/Buttons/edit	edit
$:/core/ui/Buttons/export-tiddler	export tiddler
$:/core/ui/Buttons/fold	fold tiddler
$:/core/ui/Buttons/fold-bar	fold-bar
$:/core/ui/Buttons/fold-others	fold other tiddlers
$:/core/ui/Buttons/info	info
$:/core/ui/Buttons/more-tiddler-actions	more

$:/core/ui/Buttons/new-here	new here
$:/core/ui/Buttons/new-journal-here	new journal here
$:/core/ui/Buttons/open-window	open in new window
$:/core/ui/Buttons/permalink	permalink
$:/core/ui/Buttons/permaview	permaview
$:/plugins/tiddlywiki/qrcode/ViewToolbarButton	QR code

SystemTags

19th May 2021 at 6:06pm

Concepts Reference

System tags are used to give special behaviour to tiddlers.

These are the system tags defined by the TiddlyWiki core:

$:/tags/AboveStory	marks elements to be placed at the top of the story river
$:/tags/AdvancedSearch	marks search elements
$:/tags/AdvancedSearch/FilterButton	marks filter buttons
$:/tags/Alert	marks alerts
$:/tags/BelowStory	marks elements to be placed at the bottom of the story river
$:/tags/ControlPanel	marks control panel tabs
$:/tags/ControlPanel/Advanced	marks control panel advanced tabs
$:/tags/ControlPanel/Appearance	marks control panel appearance tabs
$:/tags/ControlPanel/Info	marks control panel info tabs
$:/tags/ControlPanel/Saving	marks saving configurations

$:/tags/ControlPanel/Settings	marks control panel settings tabs
$:/tags/ControlPanel/Toolbars	marks control panel toolbar customisation tabs
$:/tags/EditorTools	marks the edit mode tiddler toolbar for non-button elements.
$:/tags/EditorToolbar	marks the editor toolbar buttons
$:/tags/EditPreview	marks custom preview panes
$:/tags/EditTemplate	marks the edit template
$:/tags/EditToolbar	marks the edit mode tiddler toolbar
$:/tags/Exporter	marks the exporters
$:/tags/Filter	marks filters in advanced search sample filter dropdown
$:/tags/Image	marks (core) images
$:/tags/ImportPreview	marks preview types for the import listing
$:/tags/KeyboardShortcut	uses the text of tagged tiddler as keyboard shortcut actions
$:/tags/Layout	marks alternative page layouts
$:/tags/Macro	marks global macros
$:/tags/Macro/View	marks global macros only active in View template
$:/tags/Macro/View/Body	marks global macros only active in View template body
$:/tags/Manager/ItemMain	marks the tiddler manager. **$:/Manager**
$:/tags/Manager/ItemSidebar	marks the tiddler manager
$:/tags/MoreSideBar	marks tabs in the 'more' sidebar
$:/tags/MoreSideBar/Plugins	marks Sidebar: More: Plugins tab
$:/tags/PageControls	marks the page control tools in the sidebar
$:/tags/PageTemplate	marks the main page elements
$:/tags/Palette	marks colour palettes
$:/tags/PluginLibrary	marks the plugin library

$:/tags/RawMarkup	marks tiddlers included as raw text at the bottom of `<head>`
$:/tags/RawMarkupWikified	marks tiddlers included as wikified text at the bottom of `<head>`
$:/tags/RawMarkupWikified/BottomBody	marks tiddlers included as wikified text at the bottom of `<body>`
$:/tags/RawMarkupWikified/TopBody	marks tiddlers included as wikified text at the top of `<body>`
$:/tags/RawMarkupWikified/TopHead	marks tiddlers included as wikified text at the top of `<head>`
$:/tags/RawStaticContent	marks raw content to be saved into the TW code
$:/tags/RemoteAssetInfo	marks plugin import handling
$:/tags/SearchResults	marks customised search results
$:/tags/ServerConnection	marks plugin import handling
$:/tags/SideBar	marks sidebar tabs
$:/tags/SideBarSegment	marks sidebar segments
$:/tags/StartupAction	marks actions executed on all platforms
$:/tags/StartupAction/Browser	marks actions only executed when running in the browser
$:/tags/StartupAction/Node	marks actions only executed when running under Node.js
$:/tags/Stylesheet	marks that a tiddler should be applied as a CSS stylesheet
$:/tags/TagDropdown	marks tags dropdown sorting
$:/tags/TextEditor/Snippet	marks text snippets
$:/tags/TiddlerInfo	marks tiddler info panel tabs
$:/tags/TiddlerInfo/Advanced	marks tabs under the advanced tiddler tab
$:/tags/TiddlerInfoSegment	marks tiddler info panel segments
$:/tags/ToolbarButtonStyle	marks the style of the buttons
$:/tags/TopLeftBar	marks the top left bar
$:/tags/TopRightBar	marks the top right bar
$:/tags/ViewTemplate	identifies the individual segments that are displayed as part of the view template

$:/tags/ViewToolbar	marks the view mode tiddler toolbar
$:/tags/ClassFilters/PageTemplate	marks filters evaluated to dynamically add classes to the page template.
$:/tags/ClassFilters/TiddlerTemplate	marks filters evaluated to dynamically add classes to the page template.
$:/tags/ControlPanel/Plugins	marks elements to be placed under "Plugins" tab in Control Panel
$:/tags/Global	marks global definitions
$:/tags/Global/View	marks global definitions only active in View template
$:/tags/Global/View/Body	marks global definitions only active in View template body
$:/tags/ViewTemplate/Subtitle	identifies the individual segments that are displayed as the tiddler subtitle

System tags defined by TiddlyWiki plugins

$:/tags/HelpPanel	can be seen at: tiddlywiki prerelease see: top left page control bubble
$:/tags/HelpPanel/Videos	help panel "videos" tab
$:/tags/MakeQR	can be seen at: tiddlywiki prerelease. see: Tiddler toolbar
$:/tags/TranslationGroup	used by the translators edition
$:/tags/TwitterUsage	twitter plugin
$:/tags/ViewToolbarButton/QRcode	see: tiddlywiki prerelease
$:/tags/test-spec	tiddlywiki test suite

SystemTiddlers

13th January 2018 at 4:38pm

Concepts

Tiddlers are classified as system tiddlers if their title starts with the special string `$:/`. The only difference from ordinary tiddlers is that they are

hidden from most searches and lists in the sidebar. (In particular, system tiddlers are shown in the "More/System" list).

System tiddlers provide a simple way to hide the internal components of TiddlyWiki so that they don't clutter things up, getting in the way of the users content.

Many of the system tiddlers follow the convention of using further slashes to delineate the components of the title. For example: `$:/config/BitmapEditor/LineWidth`.

You don't need to create your own system tiddlers, but it can be useful when customising TiddlyWiki. For example, if you create a custom palette you might call it `$:/_MyPalette` so that (a) it is hidden from searches/lists and (b) it will appear at the top of the system tiddler list for easy reference.

TabbedExampleType

toc-tabbed-internal-nav

Table Classes, Captions, Headers and Footers

19th August 2022 at 12:13pm

Tables in WikiText

Table CSS classes, captions, headers and footers can be specified as special pseudo-rows. The following example:

- `|myclass anotherClass|k` assigns the CSS classes "myclass" and "anotherClass" to the table
- `|This is a caption |c` gives the table the caption "This is a caption"

- `|Header|Header|h` adds a header row of cells with the text "Header"
- `|Footer|Footer|f` adds a footer row of cells with the text "Footer"

copy to clipboard

```
|myclass anotherClass|k
|This is a caption |c
|Cell1 |Cell2 |
|Cell3 |Cell3 |
|Header|Header|h
|Footer|Footer|f
```

That renders as:

Header	Header
Cell1	Cell2
Cell3	Cell3
Footer	Footer

This is a caption

table-example

18th January 2022 at 3:35am

WikiText parser mode transitions (Examples)

cell one	cell two
cell three	cell four

Table-of-Contents Macros

27th April 2023 at 2:55pm

Core Macros Macros

TiddlyWiki provides several macros for generating a tree of tiddler links by analysing tags:

`toc`
: A simple tree

`toc-expandable`
: A tree in which all the branches can be expanded and collapsed

`toc-selective-expandable`
: A tree in which the non-empty branches can be expanded and collapsed

`toc-tabbed-internal-nav` and `toc-tabbed-external-nav`
: A two-panel browser:
  - on the left, a selectively expandable tree that behaves like a set of vertical tabs
  - on the right, the content of whichever tiddler the user selects in the tree

The difference between the last two has to do with what happens when the user clicks a link in the right-hand panel:

`toc-tabbed-internal-nav`
: The target tiddler appears in the right-hand panel, replacing the tiddler that contained the link

toc-tabbed-external-nav
: The target tiddler appears in the normal way (which depends on the user's configured storyview)

Structure

The top level of the tree consists of the tiddlers that carry a particular tag, known as the **root tag**. Tiddlers tagged with any of those make up the next level down, and so on.

At each level, the tiddlers can be ordered by means of the **list** field of the parent tag tiddler. They can also be ordered by the macro's **sort** parameter.

The tree displays the **caption** field of a tiddler if it has one, or the tiddler's title otherwise.

Each tiddler in the tree is normally displayed as a link. To suppress this, give the tiddler a **toc-link** field with the the value **no**. In the examples, the SecondThree tiddler is set up like this. Clicking such a tiddler in the tree causes its branch to expand or collapse.

ⓘ **Introduced in v5.1.23** By default, the links open the tiddlers making up the table of contents. Alternatively, if the tiddler contains a **target** field then its contents will be used as the target of the link.

The table of contents is generated as an HTML ordered list. The `<ol>` elements always have the class `tc-toc`. Expandable trees have the additional class `tc-toc-expandable`. Selectively expandable trees (including those in the two-panel browser) have `tc-toc-selective-expandable`.

To make a table of contents appear in the sidebar, see How to add a new tab to the sidebar.

Parameters

tag
: The root tag that identifies the top level of the tree

sort
: An optional extra filter step, e.g. `sort[title]`

These two parameters are combined into a single filter expression like this:

> `[tag[$tag$]$sort$]`

`toc-tabbed-internal-nav` and **`toc-tabbed-external-nav`** take additional parameters:

selectedTiddler
: The title of the state tiddler for noting the currently selected tiddler, defaulting to `$:/temp/toc/selectedTiddler`. It is recommended that this be a system tiddler

unselectedText
: The text to display when no tiddler is selected in the tree

missingText
: The text to display if the selected tiddler doesn't exist

template
: Optionally, the title of a tiddler to use as a template for transcluding the selected tiddler into the right-hand panel

exclude ⓘ **New in v5.3.0**
: This optional parameter can be used to exclude tiddlers from the TOC list. It allows a Title List or a **`subfilter`**. Eg: `exclude:"HelloThere [[Title with spaces]]"` or `exclude:"[has[excludeTOC]]"`. Where the former will exclude two tiddlers and the later would exclude every tiddler that has a field **`excludeTOC`** independent of its value.

Be aware that eg: `[prefix[H]]` is a shortcut for `[all[tiddlers]prefix[H]]`, which can have a performance impact, if used carelessly. So use **$:/AdvancedSearch** -> **Filters** tab to test the **exclude** parameter

Custom Icons

Introduced in v5.2.4

To change the icons used by the Table-of-Contents macros, redefine the macros `toc-open-icon` and `toc-closed-icon`. This setting works for all toc-macro variants.

Default Settings

- **toc-open-icon**: `\define toc-open-icon() $:/core/images/down-arrow`
- **toc-closed-icon**: `\define toc-closed-icon() $:/core/images/right-arrow`

Custom Definitions

The default settings can be overwritten in your code using the define-pragma or the let-widget. The **$let** widget can be used, if you have multiple table of contents macros in one tiddler, that need different icons.

Define new icons using a pragma

```
\define toc-open-icon() $:/core/images/fold-button
\define toc-closed-icon() $:/core/images/folder
...
```

Define new icons using the let-widget

```
<$let toc-open-icon="$:/core/images/fold-button" toc-closed-icon="$:/core/images/folder">
...
</$let>
```

Examples

Learn more at Examples

Table-of-Contents Macros (Examples)

6th November 2018 at 11:14pm

Macro Examples Table-of-Contents Macros

These examples derive tables of contents from the root tag **`Contents`**. See Table-of-Contents Macros for details on how to use the Table-of-Contents Macros to make your own structured table-of-contents.

You can explore the same structure with these clickable tag pills:

- Contents
  - First
  - Second
    - SecondThree
  - Third

- Fourth

The tabbed example uses the real TableOfContents of this documentation instead.

Simple | Expandable | Sorted | Selective | Selective (custom icons) | Tabbed Internal | Tabbed External

copy to clipboard

```
<div class="tc-table-of-contents">
<<toc "Contents">>
</div>
```

Try it

TableOfContents

22nd March 2023 at 4:03pm

$:/tags/SideBar

- **HelloThere**
- **Learning**
- **Working with TiddlyWiki**
- **Customise TiddlyWiki**
- **Features**
- **Filters**

Languages
Editions
Plugins
Platforms
Reference
Community
About

Tables in WikiText

19th August 2022 at 12:34pm

WikiText

Basics

TiddlyWiki5 formats tables using vertical bar | characters. Exclamation marks ! are used to indicate header cells.

copy to clipboard

```
|!Cell1 |!Cell2 |
|Cell3 |Cell3 |
```

That renders as:

Cell1	Cell2

Cell3	Cell3

Cell Alignment

Table cell alignment is controlled by inserting space characters before and/or after the cell content. For example:

copy to clipboard

```
|Left aligned content |
| Right aligned content|
| Centred content |
|+++ a very wide column so we can see the alignment +++|
```

That renders as:

Left aligned content
Right aligned content
:---:
Centred content
+++ a very wide column so we can see the alignment +++

Cell vertical Alignment

Vertical alignment of cells is done by inserting either a `^` for top alignment or a `,` for bottom alignment as the first character of a cell. The normal horizontal alignment is still possible. For example:

```
|^top left |^ top center |^ top right|
|middle left | middle center | middle right|
|,bottom left |, bottom center |, bottom right|
```

The example renders as:

	top left	top center	top right	
	middle left	middle center	middle right	
	bottom left	bottom center	bottom right	

If you need to have a `^` or a `,` as the first character of a left aligned cell, you can use HTML escaping:

`^`	^
`,`	,

Cell Merging

To merge a table cell with the one above, use the special cell text `~`. To merge a cell with the one to its left use the text `<`. To merge one to its right use `>`. For example:

copy to clipboard

```
|Cell1 |Cell2 |Cell3 |Cell4 |
|Cell5 |Cell6 |Cell7 |<|
|Cell5 |~|Cell7 |Cell8 |
|>|Cell9 |Cell10 |Cell11 |
```

That renders as:

Cell1	Cell2	Cell3	Cell4
Cell5	Cell6	Cell7	
Cell5		Cell7	Cell8
Cell9		Cell10	Cell11

Table Classes, Captions, Headers and Footers

Table CSS classes, captions, headers and footers can be specified as special pseudo-rows. The following example:

- `|myclass anotherClass|k` assigns the CSS classes "myclass" and "anotherClass" to the table
- `|This is a caption |c` gives the table the caption "This is a caption"
- `|Header|Header|h` adds a header row of cells with the text "Header"

- `|Footer|Footer|f` adds a footer row of cells with the text "Footer"

copy to clipboard

```
|myclass anotherClass|k
|This is a caption |c
|Cell1 |Cell2 |
|Cell3 |Cell3 |
|Header|Header|h
|Footer|Footer|f
```

That renders as:

Header	Header
Cell1	Cell2
Cell3	Cell3
Footer	Footer

This is a caption

More examples can be found at: Tables in WikiText CSS Utility Classes

Tables in WikiText CSS Utility Classes

10th October 2022 at 9:43am

Tables in WikiText

WikiText tables can be styled by applying CSS classes. For basic information on wiktext tables and how to style them, see Tables in WikiText.

Table CSS classes, captions, headers and footers can be specified as special pseudo-rows. The following example:

- `|myclass anotherClass|k` assigns the CSS classes "myclass" and "anotherClass" to the table
- `|This is a caption |c` gives the table the caption "This is a caption"
- `|Header|Header|h` adds a header row of cells with the text "Header"
- `|Footer|Footer|f` adds a footer row of cells with the text "Footer"

copy to clipboard

```
|myclass anotherClass|k
|This is a caption |c
|Cell1 |Cell2 |
|Cell3 |Cell3 |
|Header|Header|h
|Footer|Footer|f
```

That renders as:

Header	Header
Cell1	Cell2
Cell3	Cell3
Footer	Footer

This is a caption

As seen above, the resulting table is left aligned and grows to fit the content. This is the browser default layout behaviour for tables. To get another behaviour, various CSS classes can be added into the "`k` row".

Utility Classes

Introduced in v5.2.4 The following outlines a few predefined CSS classes intended to make it simpler to style HTML block-elements and wikitext tables.

General Utility Classes

`tc-center`	Centres a block-element to the middle of the container
`tc-max-width`	Expands a block-element to use the maximum width of the container
`tc-max-width-80`	Sets the width of a block-element to use 80% of the maximum container width. This setting is useful with the `tc-center` class
`tc-edit-max-width`	Expands TextWidgets to use the maximum available width. See **ControlPanel -> Info -> Basics**
`tc-first-link-nowrap`	Ensures that any links in the first table column will never wrap to the next line

Table Utility Classes

`tc-table-no-border`	Removes the borders of a table
`tc-first-col-min-width`	The first column of a table will take up minimal possible width. It adapts to the content

Examples

The following examples apply the style classes to tables but the "General Utility Classes" can as well be used on DIVs or other HTML block elements

Centred Table

To center a table horizontally, use `tc-center`:

copy to clipboard

```
|tc-center|k
|This is a caption |c
|Cell1 |Cell2 |
|Cell3 |Cell4 |
|Header|Header|h
|Footer|Footer|f
```

That renders as:

Header	Header
Cell1	Cell2
Cell3	Cell4
Footer	Footer

This is a caption

Centred Table, 80% Width

To add empty left and right margins to a table that is otherwise full-tiddler-width, you can use `tc-max-width-80`

copy to clipboard

```
|tc-center tc-max-width-80|k
|This is a caption |c
|Cell1 |<<.lorem>> |
```

```
|<<.lorem>> |Cell4 |
|Header|Header|h
```

That renders as:

Header	Header
Cell1	Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.	Cell4

This is a caption

Table with Maximum Width

To expand a table to full tiddler width, use `tc-max-width`

copy to clipboard

```
|tc-max-width|k
|Header|Header|h
|Cell1 |Cell2 |
|Cell3 |Cell4 |
```

That renders as:

Header	**Header**
Cell1	Cell2
Cell3	Cell4

Table with First Column Minimum Width

The following example shows a simple "form" where the first columns width is adjusted to its content by means of `tc-first-col-min-width`

copy to clipboard

```
|tc-max-width tc-first-col-min-width|k
```

```
|Header|Header|h
|Cell1 |<$edit-text tiddler="$:/temp/test-table-input" tag="input" field="test"/> |
|Cell3 |<$edit-text tiddler="$:/temp/test-table-input" field="text"/>  |
```

That renders as:

Header	Header
Cell1	
Cell3	

Table with Maximum Width TextWidgets

Here, the previous "form" is styled further to give the TextWidgets full width by adding the class `tc-edit-max-width`

copy to clipboard

```
|tc-max-width tc-first-col-min-width tc-edit-max-width|k
|Header|Header|h
|Cell1 |<$edit-text tiddler="$:/temp/test-table-input" tag="input" field="test"/> |
|Cell3 |<$edit-text tiddler="$:/temp/test-table-input" field="text"/>  |
```

That renders as:

Header	Header

Header	Header
Cell1	
Cell3	

Table with No Borders

The following is a table with maximum width. It contains TextWidgets with maximum width. The first column is set to be minimum width. Further, all links in the first column are set to not line break (wrap) regardless of window resize.

copy to clipboard

```
|tc-max-width tc-first-col-min-width tc-edit-max-width tc-table-no-border tc-first-link-nowrap|k
| Cell1|<$edit-text tiddler="$:/temp/test-table-input" tag="input" field="test"/> |
|^ [[Link to a tiddler]]<br>some more text|<$edit-text tiddler="$:/temp/test-table-input" field="text"/>
|
```

That renders as:

Cell1	
Link to a tiddler some more text	

tabs Macro

21st July 2021 at 2:28pm

Core Macros Macros

The **tabs** macro presents a selection of tiddlers as a set of tabs that the user can switch between.

The tabs display the **caption** field of a tiddler if it has one, or the tiddler's title otherwise. If specified, the tabs display the **tooltip** field of a tiddler as the respective button tooltip.

By default the tabs are arranged horizontally above the content. To get vertical tabs, set the **class** parameter to **tc-vertical**.

Parameters

tabsList
: A filter selecting which tiddlers to include

default
: The title of the tiddler whose tab is to be selected by default, if the state tiddler doesn't exist. The state tiddler takes precedence over this setting

state
: The prefix for the title of a state tiddler for noting the currently selected tab, defaulting to `$:/state/tab`. It is recommended that this be a system tiddler

class
: Additional CSS classes for the generated `div` elements. Multiple classes can be separated with spaces

template
: Optionally, the title of a tiddler to use as a template for transcluding the content of the selected tab

buttonTemplate
: Optionally, the title of a tiddler to use as a template for transcluding the content of the button for the selected tab

retain
: Optionally, "yes" specifies that the content of the tabs should be retained when switching to another tab, avoiding re-rendering it (this can be useful to avoid video or audio sources unexpectedly resetting)

actions
: Optionally, actions can be specified that are triggered when changing a tab. Within the actions, the title of the selected tab is available in the **currentTab** variable and the `currentTiddler` variable from outside the tabs macro is available in the **save-currentTiddler**

explicitState
: Optionally, an explicit state title can be specified. It will be preferred over the internally computed (qualified) state title

Within the template, the title of the selected tab is available in the **currentTab** variable.

The **currentTiddler** variable is not affected by the **tabs** macro. This can put you in trouble if the list of tabs includes tiddlers that depend on the value of the **currentTiddler**, for example tiddlers listing children based on its own name. To overcome this problem you can use a TemplateTiddler like the following:

```
<$tiddler tiddler=<<currentTab>>>
<$transclude mode="block" />
</$tiddler>
```

Examples

tabs Macro (Examples)

21st February 2015 at 10:17pm

Macro Examples tabs Macro

copy to clipboard

```
<<tabs "SampleTabOne SampleTabTwo SampleTabThree SampleTabFour" "SampleTabOne" "$:/state/tab1">>
```

Try it

copy to clipboard

```
<<tabs "[tag[sampletab]]" "SampleTabTwo" "$:/state/tab2" "tc-vertical">>
```

Try it

copy to clipboard

```
<<tabs "[tag[sampletab]nsort[order]]" "SampleTabThree" "$:/state/tab3" "tc-vertical">>
```

Try it

tag Macro

25th July 2023 at 10:12pm

Core Macros Macros

The **tag** macro generates a tag pill for a specified tag. Clicking the tag pill opens a dropdown. This can be compared to the tag-pill Macro which also features other parameters.



If a list widget generates multiple tag macros for the same tag, clicking on one of them will open dropdowns on all of them. The simplest way to prevent this is to add the `counter="transclusion"` attribute to the list widget. See the examples below for more details.

Parameters

tag

The title of the tag, defaulting to the current tiddler

Examples

tag Macro (Examples)

25th July 2023 at 10:37pm

Macro Examples tag Macro

copy to clipboard

```
<<tag>>
```

Try it

copy to clipboard

```
<<tag Concepts>>
```

Try it

If a list widget generates multiple tag macros for the same tag, clicking any of them opens dropdowns on all of them, as in the example below. This is usually unwanted.

copy to clipboard

```
<$list filter="[tag[HelloThere]]">

* <$link/> is tagged with: <$list filter="[<currentTiddler>tags[]]"> <<tag>> </$list>

</$list>
```

Try it

Adding the `counter="transclusion"` attribute to the list widget that generates multiple identical tag macros causes each of them to be identified as a unique one. Clicking on any of them opens only a single dropdown.

copy to clipboard

```
<$list filter="[tag[HelloThere]]" counter="transclusion">
```

```
* <$link/> is tagged with: <$list filter="[<currentTiddler>tags[]]"> <<tag>> </$list>

</$list>
```

Try it

A slightly more performant option is to use the `variable="transclusion"` attribute in the list widget. In this case, the variable `<<transclusion>>` has to be used inside the list widget instead of the `<<currentTiddler>>` .

copy to clipboard

```
<$list filter="[tag[HelloThere]]" variable="transclusion">

* <$link to=<<transclusion>>/> is tagged with: <$list filter="[<transclusion>tags[]]"> <<tag>> </$list>

</$list>
```

Try it

tag Operator

26th November 2016 at 1:29pm

Common Operators Filter Operators Negatable Operators Tag Operators

purpose	filter the input by tag
input	a selection of titles
suffix	**S** = ⓘ **Introduced in v5.1.14** optional `strict` flag
parameter	**T** = the title of a tag
output	those input tiddlers that have tag **T**
! **output**	those input tiddlers that do *NOT* have tag **T**

Learn more about how to use Filters

The output is sorted using the tag's **`list`** field and the tiddlers' **`list-before`** and **`list-after`** fields.

The behaviour when **T** is empty depends on the settings of the **S** optional suffix:

- if **T** is missing and **S** is either missing or set to "loose", then the output of `tag` is empty, and the output of `!tag` is a copy of the input.
- ⓘ **Introduced in v5.1.14** if **T** is missing and **S** is set to "strict", then the output of both `tag` and `!tag` is a copy of the input

Examples

tag Operator (Examples)

24th January 2015 at 1:29pm

Operator Examples tag Operator

```
[tag[task]]
```

Try it

```
[tag[task]!tag[done]]
```

Try it

```
[!tag[task]]
```

Try it

```
[all[shadows]tag[$:/tags/Stylesheet]]
```

Try it

tag-picker Macro

28th November 2016 at 8:14pm

Core Macros Macros

The **tag-picker** macro generates a combination of a text box and a button that allows a tag to be selected and added.

Parameters

actions

Action widgets to be triggered when the pill is clicked. Within the text, the variable **tag** contains the title of the selected tag.

tagField

ⓘ **Introduced in v5.1.23** The **field** that gets updated with the selected tag. Defaults to **tags**.

Examples

tag-pill Macro

28th November 2016 at 8:12pm

Core Macros Macros

The **tag-pill** macro generates a static tag pill showing a specified tag, but without the dropdown action provided by the tag Macro.

Parameters

tag

The title of the tag

element-tag

The element name to be used for the pill (defaults to "span")

element-attributes

Additional attributes for the pill element

actions

Action widgets to be triggered when the pill is clicked. Within the text, the macro parameter **tag** contains the title of the selected tag.

Examples

Tagging

12th June 2016 at 3:20pm

Concepts Working with TiddlyWiki

Tagging is a way of organising tiddlers into categories. For example, if you had tiddlers representing various individuals, you could tag them as **friend**, **family**, **colleague** etc to indicate these people's relationships to you.

A tag is in fact just a tiddler (or a potential tiddler), and it can have tags of its own. You can add any number of tags to the same tiddler.

See Creating and editing tiddlers for instructions on how to tag.

By tagging your tiddlers, you can view, navigate and organise your information in numerous additional ways:

- The coloured tag pills on a tiddler give you quick access to all the other tiddlers with the same tag, as well as to the tiddler that represents the tag itself.
- If a tiddler is serving as a tag, then the **Tagging** tab in its InfoPanel will show you which tiddlers are currently tagged with it.
- The **More** tab of the sidebar has a **Tags** tab that presents an overview of all your tags and lets you access all your tagged tiddlers.
- You can use filters to create lists of tiddlers based on their tags. You can then display any combination of the fields of those tiddlers. For example, you could build a glossary by listing the title and text of all tiddlers tagged **Glossary**. Such lists can be formatted in any way you

wish: e.g. bulleted, numbered or comma-separated.

- There are a number of special **system tags** that control the layout of tiddlers and the entire TiddlyWiki page. See *Page and tiddler layout customisation* for instructions.

There are two more things you can do with tags:

Set a tag's colour and icon

You can use the 🏷 **tag manager**, found on the **Tags** tab under **More** in the sidebar, to change the colour of a tag's pill or add an icon to the pill.

- To change the colour, click the button in the **Colour** column to select from a colour picker. Alternatively, click the icon in the **Info** column, then type a *CSS* colour value in the **Colour** field
- To change the icon, click the ⌄ button in the **Icon** column and choose from the list of available icons

Change the order in which tags are listed

By default, tagged tiddlers are listed in alphabetical order.

If you want any other order, add a `list` field to the tag tiddler, and set its value to be a list of the tiddlers in that order.

The **list** field doesn't have to mention all of the tiddlers. See the precise rules TiddlyWiki uses to order tagged tiddlers.

tagging Operator

3rd February 2015 at 8:16pm

Filter Operators Tag Operators

purpose	find the tiddlers that have the input tags
input	a selection of tags
parameter	none
output	the titles of any tiddlers that carry the input tags

Learn more about how to use Filters

Each input tag is processed in turn. The list of tiddlers carrying that tag is generated, sorted, and then dominantly appended to the operator's overall output.

Examples

tagging Operator (Examples)

24th January 2015 at 2:36pm

Operator Examples tagging Operator

```
[[task]tagging[]]
```

→ same as `[tag[task]]`

Try it

```
Concepts task +[tagging[]]
```

→ tiddlers that are tagged **Concepts** or **task**

Try it

```
[all[current]tagging[]]
```

→ tiddlers tagged with the current one

Try it

tags Operator

3rd February 2015 at 8:16pm

Filter Operators Tag Operators

purpose	select all tags of the input tiddlers
input	a selection of titles
parameter	none
output	all the tags carried by the input tiddlers

Learn more about how to use Filters

Each input title is processed in turn. The corresponding tiddler's tags are retrieved (in order of appearance in the **tags** field) and then

dominantly appended to the operator's output.

Examples

tags Operator (Examples)

24th January 2015 at 2:42pm

Operator Examples tags Operator

```
[[Filter Operators]tags[]]
```

Try it

```
[all[shadows]tags[]]
```

Try it

```
[all[shadows+tiddlers]tags[]sort[]]
```

Try it

TagTiddlers

27th November 2021 at 2:59pm

Tagging

A tag tiddler is any tiddler that is in use as a tag.

The **Tagging** tab on the InfoPanel of a tag tiddler shows which tiddlers are tagged with the tag tiddler.

A tag can be used without a corresponding tag tiddler.

tan Operator

22nd October 2021 at 12:28am

Filter Operators Mathematics Operators Trigonometric Operators Unary Mathematics Operators

purpose	calculate the tangent value of a list of angles (given in radians)
input	a selection of titles
output	the tangent of the input angles

Learn more about how to use Filters

ⓘ **Introduced in v5.1.21** See Mathematics Operators for an overview.

Examples

tan Operator (Examples)

20th October 2021 at 4:22pm

Operator Examples tan Operator

```
[[2]tan[]]
```

Try it

```
=1 =2 =3 =4 +[tan[]]
```

Try it

task

15th November 2021 at 2:31am

TaskManagementExample

This tag identifies the tasks for the TaskManagementExample

TaskManagementExample

7th March 2018 at 4:35pm

Learning

TiddlyWiki5 can be used as a simple task management system without further customisation. The idea is that tasks be tagged `task`, with those that are completed also tagged `done`. In this way it is straightforward to generate task lists.



There is an enhanced version of this demo that adds the ability to drag and drop the tasks to re-order them.

Outstanding tasks

- [ ] Compose ballad
- [ ] Make the beds
- [ ] Kill the Dragon
- [ ] Get the Ring

Completed tasks

- [x] ~~Go to Mordor~~

TaskManagementExample (Draggable)

7th March 2018 at 4:35pm

Learning

This is a version of the TaskManagementExample enhanced with the ability to drag and drop the task list to re-order them.

Outstanding tasks

Drag the tasks to re-order them

- [ ] Make the beds
- [ ] Compose ballad
- [ ] Kill the Dragon
- [ ] Get the Ring

Completed tasks

(Listed in reverse order of completion)

- [x] ~~Go to Mordor~~

TaskManagementExampleDraggableTemplate

7th March 2018 at 4:35pm

Learning

- [ ] TaskManagementExampleDraggableTemplate

Technical Prose Style

17th January 2015 at 4:25pm

Improving TiddlyWiki Documentation

When writing an instruction tiddler, start by planning a route through the information you wish to present. This should be a simple, logical, direct progression of thoughts, with no backtracking or forward references. Use this approach even within individual sentences: always proceed from cause to effect, from the old or known to the new or unknown.

Keep sentences short and simple. A clear technical sentence seldom contains more than one idea. It therefore avoids parenthetical information. Similarly, keep paragraph structure simple. A flat presentation is often easier to understand than a hierarchical one.

It is often possible to simplify a sentence without changing its meaning, merely by adjusting its vocabulary or grammatical structure. "Execution of the macro is performed" just means "The macro runs". "Your expectation might be..." just means "You might expect...".

Prefer the active voice by default: "Jane creates a tiddler" rather than "a tiddler is created by Jane". The passive voice can be useful if you want the reader to focus on the action itself or its result: "a tiddler is created". But it can often be clearer to proceed from cause to effect and say "this creates a tiddler" in the active voice.

Documentation often presents two items that are parallel either by similarity or by difference. The reader will more easily detect such a pattern if you use the same sentence or phrase structure for both. But this must be balanced with the need to avoid monotony.

Prefer precise instructions over woolly descriptions. If something has a name, use it. If something lacks a name, give it a tiddler.

TemplateTiddlers

20th February 2015 at 5:09pm

Concepts

A template tiddler is not actually a type of tiddler, it is a role in which a tiddler can be used.

Templates are a way to re-use chunks of WikiText.

Transcluding through a template extends the basic functionality of Transclusion by combining two tiddlers:

- A template tiddler that contains the WikiText to be displayed. It can contain transclusions that reference fields in the current tiddler
- A target tiddler that identifies which tiddler is to be treated as current when resolving references to fields

The best example of templating is the main story river in TiddlyWiki. Each tiddler in the story river is rendered through a *ViewTemplate* that specifies how each field is to be rendered.

See Transclusion with Templates for details.

Temporary Tiddlers

2nd February 2024 at 1:02pm

Concepts

Temporary tiddlers are tiddlers that will be discarded when TiddlyWiki is saved. Under default configuration of the SavingMechanism (more specifically, the filter in **$:/core/save/all**), these are tiddlers prefixed with `$:/temp/`. This prefix makes them SystemTiddlers as well.

One example usage of temporary tiddlers is storing the search queries. The query typed in the **$:/AdvancedSearch** is stored in **$:/temp/advancedsearch**.

Ten reasons to switch to TiddlyWiki

14th April 2015 at 9:16am

HelloThere

1. With TiddlyWiki you can organise your notes your way, not their way. Your notes conform to your way of thinking rather than being forced into a hierarchical straightjacket of notebooks and tabs
2. TiddlyWiki's nonlinear approach will actually make you think about your information in new and helpful ways
3. Finding your notes in TiddlyWiki is lightning fast
4. There are many ways to customise and adapt every aspect of TiddlyWiki
5. TiddlyWiki is free and is compatible with all platforms. Any web browser will open it. You don't need to purchase an expensive program or pay a subscription fee to use it
6. TiddlyWiki files promote the free sharing of information. There are many ways you can share your information from TiddlyWiki
7. With TiddlyWiki, your information is yours, and you store it where you want to - on your device, on a USB stick, in Dropbox, on your server
8. TiddlyWiki features an ever-growing number of plugins, themes, widgets, and languages
9. The online TiddlyWiki community is friendly and will do their best to give you the help you need
10. If you are a programmer, you have even more ways to play with TiddlyWiki. With TiddlyWiki, the more you know, the more fun you can have with it

Testimonials - Joe Armstrong

Testimonial

The TiddlyWiki is the best software I've ever found for organising my ideas.

It's well worth spending an hour or so playing with it to see how it can help you. This will be time well-spent and will change how you think and how you organise your ideas.

Testimonials - Network World

Testimonial

TiddlyWiki gets a Gearhead rating of 6 out of 5 (it's that good).

Finding code that works flawlessly after just two or three years is magical enough but after seven years?!

Testimonials - Product Hunt

Testimonial



Text preview

29th October 2021 at 11:01am

Editor toolbar

Pressing the eye icon will open or close a preview window showing the results of rendering the WikiText (the icon-eye will also open or close).

Right to the eye there is another icon, , which when clicked, shows you a dropdown which gives you access to six different types of preview:

- *output* shows you the text as you see it when you do not edit the tiddler.
- *html* shows you the source code of that output. Your web browser uses this source code to display webpages.
- *parsetree* and *widgettree* show you the two object trees that TiddlyWiki internally generates from WikiText. The *parsetree* collects basic information about your WikiText. From the informations of the parsetree the *widgettree* then generates extended informations and finally the above html (and the widgettree also does other things).
- *differences from current* shows you the differences since the tiddler was last saved. Deleted parts are red, added parts are green.
- *differences from shadow (if any)* does the same, but this time it shows you the differences to the underlying shadow-tiddler.



By default, the preview pane button controls whether the preview pane is displayed for all tiddlers that are open for editing. There is a hidden setting to make the button work on a per-tiddler basis



The variable tv-tiddler-preview can be used to detect whether content is being displayed in the preview pane.

Text-Slicer Edition

2nd June 2016 at 7:15am

Editions

The **Text-Slicer** edition of TiddlyWiki contains tools to help advanced users slice long texts up into individual tiddlers.

https://tiddlywiki.com/editions/text-slicer/

TextReference

21st March 2023 at 2:04pm

Concepts

A TextReference is a general purpose way to describe a fragment of text as either a field of a tiddler, or an index within a data tiddler.

In different situations, text references can be used to retrieve values, or to specify a value that should be modified.

A TextReference consists of several parts:

- The title of the target tiddler. If omitted, it defaults to the Current Tiddler
- Either one of:
  - The name of a field (marked with `!!`)
  - The name of an index within a data tiddler (marked with `##`)
- If both the field and index are omitted, the text field is used as the default

Most of the parts of a text reference can be optional:

- `tiddlerTitle` - the title field of the specified tiddler

- `tiddlerTitle!!field` - a tiddler field (eg, `modified`, `modifier`, `type` etc)
- `!!field` - a field of the current tiddler
- `tiddlerTitle##propertyIndex` - extracts a named property from DataTiddlers

Text references can be used in several places:

- As indirect parameters within Filters (eg, `<$list filter="[tag{MyTag!!name}]"/>`)
- As indirect attributes of an element or widget (eg, `<$widget attrib={{Title!!description}}/>`)
- As the parameter of a shortcut transclusion (eg, `{{MyTiddler!!title}}`)
- As the `state` attribute of the RevealWidget and the LinkCatcherWidget



Note the distinction between a text reference such as `foo!!bar` and a transclusion of a text reference such as `{{foo!!bar}}`

TextWidget

15th January 2019 at 1:00am

Widgets

Introduction

The text widget displays plain text without parsing it as WikiText, opposite of WikifyWidget.

Content and Attributes

The content of the `<$text>` widget is not used.

Attribute	Description
text	The text to display

Example

copy to clipboard

```
<$vars string="//italic//">

* <<string>>
* <$text text=<<string>>/>

</$vars>
```

That renders as:

- *italic*
- //italic//

copy to clipboard

```
The text-test field of this tiddler has a field value: <$text text={{!!text-test}}/>
```

```
It displays in its normal WikiText format as: {{!!text-test}}
```

That renders as:

The text-test field of this tiddler has a field value: e=mc^^2^^

It displays in its normal WikiText format as: e=mc^2

The Extended Listops Filters

1st March 2023 at 6:28pm

See Extended Listops Filters

The First Rule of Using TiddlyWiki

19th September 2014 at 5:44pm

Working with TiddlyWiki

Horror Stories

Every so often, TiddlyWiki users report heart-rending tales of personal data loss on the discussion groups:

> My entire TiddlyWiki has just been wiped out when Firefox crashed during saving a tiddly.

https://groups.google.com/d/topic/tiddlywiki/oG2L7OXhUoI/discussion

> Last time I used it was last night at home on my Windows 7 desktop, hit the check mark to stop editing my last entry, it then saves via TiddlyFox and I eject my USB drive. I came to work this morning, plugged in my USB, enter my TW5 password and it doesn't want to open after several attempts. I browse to my TW5 html file and notice that my file size is no longer 3MB.. instead it is 80KB. This leads me to believe I lost everything.

https://groups.google.com/d/topic/tiddlywiki/SXStDJ0ntGI/discussion

Don't let it happen to you!

The first rule of using TiddlyWiki is:

Backup your data!

TiddlyWiki is a very flexible, customisable system that puts you firmly in charge of your own data. Every care is taken in TiddlyWiki's development to ensure that it is a safe place to preserve your most valuable data but the ultimate responsibility to reduce the risk of data loss falls to users.

The best way to make sure that your data is safe is to practise a rigorous system of backups:

- Consider using services like Dropbox to continuously back up your personal data to the cloud. (Dropbox has a neat feature whereby they keep track of previous versions of files)
- Retain backups before upgrading to a new version of TiddlyWiki
- Figure out and protect yourself from the worst case scenarios: what if your USB stick or hard drive fails? What if your computer is hit by a ransomeware virus?
- Be defensive and redundant: for example, take multiple backups and keep them in separate physical locations

Then Filter Run Prefix

10th July 2023 at 9:42am

Named Filter Run Prefix

purpose	replace any input to this filter run with its output, only evaluating the run when there is any input
input	`all` tiddler titles
output	the output of this filter run replaces the output of previous runs unless it is an empty list (see below)

New in v5.3.0

TypeError: setting getter-only property "parentNode"

The **`:then`** filter run prefix is used to replace the result of all previous filter runs with its output.

If the result of all previous runs is an empty list, the **`:then`** prefixed filter run is not evaluated.

If the output of a **`:then`** prefixed filter run is itself an empty list, the result of all previous filter runs is passed through unaltered.



Note that a list with a single empty string item is not an empty list.

:then run prefix versus the **then** operator

The major difference between the **then** operator and a **:then** prefixed filter run is that the operator will replace *each item* of the input Title List with its parameter while **:then** will replace the *whole input list* with the result of its run.

:then Filter Run Prefix	**then Operator**
`[tag[WikiText]] :then[[true]]` Try it	`[tag[WikiText]then[true]]` Try it To make them equivalent, additional filter steps may be added: `[tag[WikiText]count[]compare:number:gt[0]then[true]]` Try it

Then Filter Run Prefix (Examples)

Then Filter Run Prefix (Examples)

17th June 2023 at 8:37pm

Then Filter Run Prefix

Conditional Execution

The **:then** filter run prefix can be used to avoid the need for nested ListWidgets or Macro Definitions in WikiText.

copy to clipboard

```
<$edit-text field="search" placeholder="Search title"/>

<$let searchTerm={{!!search}}>
<$list filter="[<searchTerm>minlength[3]] :then[!is[system]search:title<searchTerm>]" template="$:/core/
ui/ListItemTemplate"/>
</$let>
```

That renders as:

Search title

Conditional (Sub)Filters

The **:then** filter run prefix can be combined with the **:else** prefix to create conditional filters. In this example, the fields used in

searchSubfilter for searching depend on the value of *$:/temp/searchFields* and the sort order used by **sortSubfilter** depends on the value of *$:/temp/searchSort*. Checkboxes are used to set the values of these tiddlers.



Note that each filter run of the subfilter receives the input of the **subfilter** operator as input

Since the **then** and **else** operators cannot call subfilters or perform additional filter steps, they cannot be used for such applications.

copy to clipboard

```
<$checkbox tiddler="$:/temp/searchSort"
      field="text"
      checked="chrono" unchecked="alpha" default="alpha">
  Sort chronologically (newest first)
</$checkbox><br/>
<$checkbox tiddler="$:/temp/searchFields"
      field="text"
      checked="title" unchecked="default" default="title">
  Search <<.field title>> only
</$checkbox><p/>
<$let searchSubfilter="[{$:/temp/searchFields}match[default]] :then[search[prefix]]
:else[search:title[prefix]]"
      sortSubfilter="[{$:/temp/searchSort}match[chrono]] :then[!nsort[modified]] :else[sort[title]]"
      limit=10 >
  <$list filter="[all[tiddlers]!
is[system]subfilter<searchSubfilter>subfilter<sortSubfilter>first<limit>]">
    <$link/> (<$text text={{{ [{!!modified}format:date[YYYY-0MM-0DD]] }}} />)<br/>
```

```
  </$list>
  <$list filter="[all[tiddlers]!is[system]subfilter<searchSubfilter>rest<limit>count[]]">
    ... and <<currentTiddler>> more.
  </$list>
</$let>
```

That renders as:

☐ Sort chronologically (newest first)
☑ Search **title** only

addprefix Operator (2015-02-03)
addprefix Operator (Examples) (2015-01-18)
All Filter Run Prefix (2023-07-11)
And Filter Run Prefix (2023-07-11)
Cascade Filter Run Prefix (2023-07-10)
Cascade Filter Run Prefix (Examples) (2023-03-05)
Else Filter Run Prefix (2023-03-22)
Except Filter Run Prefix (2023-03-22)
Filter Filter Run Prefix (2023-07-10)
Filter Filter Run Prefix (Examples) (2023-03-05)
... and 22 more.

then Operator

1st May 2023 at 7:43pm

Conditional Operators Filter Operators

purpose	replace input titles by a constant string
input	a selection of titles
parameter	**T** = a string
output	the input titles with each one replaced by the string **T**

Learn more about how to use Filters

Introduced in v5.1.20 See Conditional Operators for an overview.



The Then Filter Run Prefix has a similar purpose to the **then** operator. See its documentation for a comparison of usage.

Examples

then Operator (Examples)

2nd August 2019 at 1:35pm

Operator Examples then Operator

```
[[HelloThereMissing]is[missing]then[yes]]
```

Try it

```
[[HelloThere]is[missing]then[yes]]
```

Try it

Third

14th November 2021 at 2:36am

Contents Table-of-Contents Demonstrations

This is an example tiddler. See Table-of-Contents Macros (Examples).

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

ThirdOne

14th November 2021 at 2:36am

Table-of-Contents Demonstrations Third

This is an example tiddler. See Table-of-Contents Macros (Examples).

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

ThirdThree

14th November 2021 at 2:36am

Table-of-Contents Demonstrations Third

This is an example tiddler. See Table-of-Contents Macros (Examples).

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

ThirdTwo

14th November 2021 at 2:36am

Table-of-Contents Demonstrations Third

This is an example tiddler. See Table-of-Contents Macros (Examples).

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

thisTiddler Variable

5th May 2023 at 10:01pm

Core Variables Variables

ⓘ **New in v5.3.0** The **thisTiddler** variable is set by the **`$transclude`** widget to contain the title of the tiddler that was transcluded. This means that within transclusions **thisTiddler** will contain the title of the transcluded tiddler. It is not affected by macros, which means that within a macro **thisTiddler** will contain the title of the tiddler from which the macro was invoked. Intermediate, nested macro calls are ignored.

There are two key usages for **thisTiddler**:

- Within a transcluded template, **thisTiddler** points to the template itself, making it possible for the template to access data stored in other fields of the template
- Within a macro, **thisTiddler** points to the template that invoked the macro. If the macro was invoked by another macro, the template that invoked that macro is returned, and so on

Compare **`storyTiddler`** and **`currentTiddler`**.

Examples

thisTiddler Variable (Examples)

4th March 2023 at 1:41pm

thisTiddler Variable Variable Examples

The first example shows how **thisTiddler** works whithin transcluded template. Here the SampleTemplate is a template for demonstration.

copy to clipboard

```
{{||SampleTemplate}}
```

Try it

The second example shows **thisTiddler** in the body of tiddler and inside a macro.

copy to clipboard

```
<$text text=<<thisTiddler>>/>
```

Try it

copy to clipboard

```
\define example() <$text text=<<thisTiddler>>/>
<<example>>
```

Try it

thumbnail Macro

25th March 2015 at 6:23pm

Core Macros Macros

The **thumbnail** macros are used to create linkable thumbnail panels.

Parameters

(none)

Examples

thumbnail Macro (Examples)

25th March 2015 at 6:21pm

Macro Examples thumbnail Macro

Examples

Here is an example of the `thumbnail-right` macro used to create a video thumbnail that floats to the right of the text

copy to clipboard

```
<<thumbnail-right link:"Introduction Video" image:"Introduction Video Thumbnail.jpg"
caption:"Introduction to ~TiddlyWiki" icon:"{{$:/core/images/video}}" color:"red">>
```

```
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
```

Try it

Thursday

16th November 2021 at 11:11pm

Operator Examples

This example tiddler is used to illustrate some of the Filter Operators.

`before` and `after`

These examples make use of the Days of the Week tiddler.

```
[list[Days of the Week]before{!!title}]
```

Try it

```
[list[Days of the Week]after{!!title}]
```

Try it

Tiddler Colour Cascade

6th December 2021 at 5:53pm

Cascades

The tiddler colour cascade is a cascade used to choose which colour should be used for a particular tiddler.

Core tiddler colour cascades can be found in **$:/core/macros/tag**, **$:/core/ui/Components/tag-link**, **$:/core/ui/EditTemplate/tags**, **$:/core/ui/TagPickerTagTemplate**, **$:/core/ui/TagTemplate** and **$:/core/ui/ViewTemplate/title**

The default tiddler colour cascade consists of:

1. If the tiddler has a **color** field, use the value as the colour
2. If the tiddler *$:/config/DefaultTiddlerColour* exists, use the value as the colour

You can see the current settings for the tiddler colour cascade in **$:/ControlPanel** under the **Info** -> **Advanced** -> **Cascades** -> **Tiddler Colour** tab.

Tiddler Commander by Mohammad

16th November 2020 at 9:46pm

Community Plugins

Tiddler Commander plugin, in short **Commander** is a unique tool for batch operations on tiddlers.

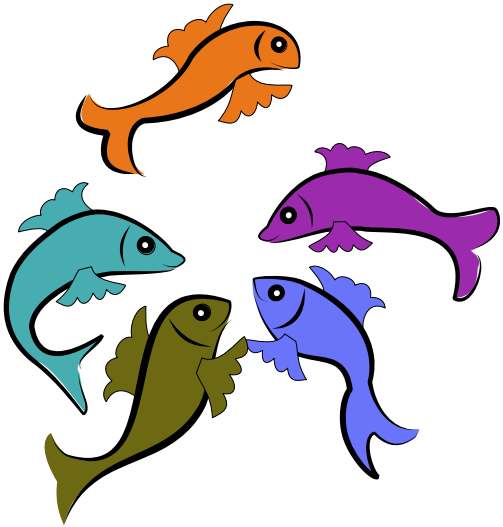
https://kookma.github.io/TW-Commander/

Commander has many features including:

- Bulk tiddler creation/deletion
- Selective operation
- Combo search to filter and select tiddlers
- Title operation: add, remove prefixes and suffixes, also remove cahras form begining and end of title (on renaming tiddlers, relink can be used to update title in other tiddlers)
- Tag operation: add, remove, replace
- Field operation: add, remove, rename, and set field value
- SnR, search and replace in all fields including text, tags, *title*, and common fields
- Inspect, to review and inspect tiddlers in one place, scroll among them and edit all fields (including common fields), tags, text (title is an exception!)
- Log, create logs of all operations
- Search, *save and load* any combination of filter search

Tiddler Fishes.svg

picture



Tiddler Icon Cascade

6th December 2021 at 5:00pm

Cascades

The tiddler icon cascade is a cascade used to choose which icon should be used for a particular tiddler.

The core tiddler icon cascade can be found in **$:/core/ui/TiddlerIcon**

The default tiddler icon cascade consists of:

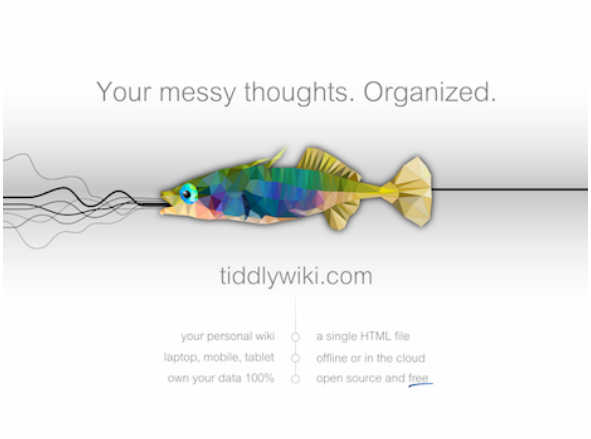
1. If the tiddler has an **icon** field, use the value as the title of the icon tiddler
2. If the tiddler *$:/config/DefaultTiddlerIcon* exists, use the value as the title of the icon tiddler

You can see the current settings for the tiddler icon cascade in **$:/ControlPanel** under the **Info** -> **Advanced** -> **Cascades** -> **Tiddler Icon** tab.

- How to create a custom tiddler icon rule

Tiddler Poster.png

picture



Tiddler Structure

7th February 2021 at 1:47pm

Improving TiddlyWiki Documentation

In accordance with the Philosophy of Tiddlers, documentation tiddlers are typically short and interlinked.

When a tiddler seems as if it needs to contain subheadings, this is often a sign that it should in fact be split into several tiddlers. But it is reasonable for a reference tiddler to consist of an untitled introductory section followed by a titled section of details.

Consistency of terminology is essential if the reader is not to become confused. Consistent typography and punctuation lend a professional quality to the documentation. Macros can improve the consistency and maintainability of the text.

Use numbered lists for step-by-step instructions, and bullet points for lists whose order is arbitrary. Use a definition list in preference to a bulleted list if each bulleted item would begin with a term and a colon. If at all possible, avoid burdening the reader with a nested list.

Use a table when information naturally falls into three or more columns, and also for lists of parameters, attributes, etc in reference tiddlers.

Avoid periods at the end of list items, headings and table cell text.

The documentation describes the current reality of TiddlyWiki. Avoid discussing future aspirations.

Tiddler Title Policy

18th January 2015 at 7:36pm

Improving TiddlyWiki Documentation

Many documentation tiddlers, especially the reference ones, are concerned with a single concept. Their titles should be succinct noun phrases like **`List Widget`** or **`Tiddler Fields`**.

Each of the main words of such a title begins with a capital letter. Minor words such as "and", "or", "the", "to" and "with" do not.

Tags also follow this pattern.

Titles of this kind are plural if they denote a category of items, e.g. **`Keyboard Shortcuts`** or **`Tiddler Fields`**. Such titles are used to tag more specific tiddlers within the category.

Where a concept is an item rather than a category, its tiddler has a singular title, e.g. **`List Widget`**, **`tag Operator`**.

Start a title with its most distinctive part. For instance, the tiddlers documenting the filter operators have titles like **`addprefix Operator`**. This helps the reader scan a list of links to find a particular tiddler.

Avoid starting a title with the word "the".

In the past, many tiddlers had CamelCase titles. This is gradually being phased out of the documentation to improve readability. CamelCase titles should no longer be used, even for tags, except in cases like **`JavaScript`** where that is the standard spelling.

Instruction tiddlers often have longer titles that can be more complicated than just a noun phrase, e.g. **`Ten reasons to switch to TiddlyWiki`**. These titles use sentence case, i.e. only the first word (and any proper names) starts with a capital letter.

How-to tiddlers have titles that begin with "How to", e.g. **`How to edit a tiddler`**. Avoid titles like **`Editing tiddlers`**, because a less fluent English speaker could misunderstand that as the name of a category of tiddlers.

TiddlerFields

9th January 2022 at 11:14am

Concepts

TiddlerFields are name:value pairs that make up a tiddler. Field names must be lowercase letters, digits or the characters `-` (dash), `_` (underscore) and `.` (period).

The standard fields are:

Field Name	Description
`title`	The unique name of a tiddler
`text`	The body text of a tiddler

`modified`	The date and time at which a tiddler was last modified
`modifier`	The tiddler title associated with the person who last modified a tiddler
`created`	The date a tiddler was created
`creator`	The name of the person who created a tiddler
`tags`	A list of tags associated with a tiddler
`type`	The content type of a tiddler
`list`	An ordered list of tiddler titles associated with a tiddler – see ListField
`caption`	The text to be displayed on a tab or button

Other fields used by the core are:

Field Name	Description
`class`	
`code-body`	ⓘ **Introduced in v5.2.1** The view template will display the tiddler as code if set to **yes**
`color`	The CSS color value associated with a tiddler
`description`	The descriptive text for a plugin, or a modal dialogue
`draft.of`	For draft tiddlers, contains the title of the tiddler of which this is a draft
`draft.title`	For draft tiddlers, contains the proposed new title of the tiddler
`footer`	The footer text for a modal
`hide-body`	The view template will hide bodies of tiddlers if set to **yes**
`icon`	The title of the tiddler containing the icon associated with a tiddler
`library`	Indicates that a tiddler should be saved as a JavaScript library if set to **yes**
`list-after`	If set, the title of the tiddler after which this tiddler should be added to the ordered list of tiddler titles, or at the end of the list if this field is present but empty

`list-before`	If set, the title of a tiddler before which this tiddler should be added to the ordered list of tiddler titles, or at the start of the list if this field is present but empty
`name`	The human readable name associated with a plugin tiddler
`plugin-priority`	A numerical value indicating the priority of a plugin tiddler
`plugin-type`	The type of plugin in a plugin tiddler
`source`	The source URL associated with a tiddler
`subtitle`	The subtitle text for a modal
`throttle.refresh`	If present, throttles refreshes of this tiddler
`toc-link`	Suppresses the tiddler's link in a Table of Contents tree if set to **no**
`_canonical_uri`	The full URI of an external image tiddler

The *TiddlyWebAdaptor* uses a few more fields:

Field Name	Description
`bag`	The name of the bag from which a tiddler came
`revision`	The revision of the tiddler held at the server
`_is_skinny`	If present, indicates that the tiddler text field must be loaded from the server

Details of the fields used in this TiddlyWiki are shown in the **control panel** under the **Info** tab >> **Advanced** sub-tab >> Tiddler Fields

TiddlerFiles

14th July 2021 at 3:07pm

TiddlyWiki on Node.js

Tiddlers can be stored in text files in several different formats. Files containing single tiddlers can also have an auxiliary `.meta` file formatted as a sequence of name:value pairs:

```
title: TheTitle
modifier: someone
```

TiddlyWeb-style .tid files

These files consist of a sequence of lines containing name:value pairs, a blank line and then the text of the tiddler. For example:

```
title: MyTiddler
modifier: Jeremy

This is the text of my tiddler.
```

Note that many text editors require that files include a terminating newline. If you want to avoid including the terminating newline in the text of the tiddler you can use this alternative syntax:

```
title: MyTiddler
modifier: Jeremy
text: This is the text of my tiddler.
```

The ContentType `application/x-tiddler` is used internally for these files

TiddlyWiki `<DIV>` .tiddler files

In TiddlyWiki 5, `*.tiddler` files look like this:

```
<div title="AnotherExampleStyleSheet" modifier="blaine" created="201102111106" modified="201102111310"
tags="examples" creator="psd">
<pre>Note that there is an embedded <pre> tag, and line feeds are not escaped.

And, weirdly, there is no HTML encoding of the body.</pre>
</div>
```

These `*.tiddler` files are not exactly the same as the tiddlers inside a TiddlyWiki HTML file where they are HTML encoded.

Older `*.tiddler` files more closely matched the store format used by TiddlyWiki at the time:

```
<div tiddler="AnotherExampleStyleSheet" modifier="JeremyRuston" modified="200508181432"
created="200508181432" tags="examples">This is an old-school .tiddler file, without an embedded
&lt;pre&gt; tag.\nNote how the body is &quot;HTML encoded&quot; and new lines are escaped to \\n</div>
```

The ContentType `application/x-tiddler-html-div` is used internally for these files

TiddlyWeb-style JSON files

These files are a straightforward array of hashmaps of `name:value` properties. All field values must be specified as strings.

For example:

```
[
```

```
    {
        "title": "First Tiddler",
        "text": "Text of first tiddler",
        "tags": "one two [[t h r e e]]"
    },{
        "title": "Second Tiddler",
        "text": "Text of second tiddler",
        "modified": "20150216171751154"
    }
]
```

The ContentType `application/json` is used internally for these files.

Note that JSON files that do not conform to the tiddler structure will instead be imported as a single tiddler containing the JSON data.

New JSON-based format for TiddlyWiki HTML files

The new format for TiddlyWiki HTML files embeds the tiddlers in JSON format within a script tag:

```
<script class="tiddlywiki-tiddler-store" type="application/json">[
{"title": "A","text": "One"},
{"title": "B","text": "Two"}
]</script>
```

Old DIV-based format for TiddlyWiki HTML files

TiddlyWiki Classic and TiddlyWiki 5 prior to version v5.2.0 stored tiddlers encoded in `<DIV>` format.

For TiddlyWiki to import an unencrypted DIV-based HTML file, it requires a `<div id="storeArea">` containing tiddler DIVs as explained above. For example:

```
<div id="storeArea">
<div created="20130302085406905" modified="20130302084548184" tags="Examples" title="A tiddler title">
<pre>HTML encoded text of tiddler
</pre>
</div>
<div created="20140315085406905" modified="20140321084548184" tags="One Two [[Three with Space]]"
title="Another title" customfield="field value">
<pre>Text of this tiddler
</pre>
</div>
</div>
```

TiddlerLinks

31st December 2014 at 10:33am

Concepts

Links are regions of a tiddler that can be clicked to cause navigation to a different tiddler. The navigation behaviour is determined by the current *StoryView*; the classic TiddlyWiki view displays the story as a linear sequence of tiddlers.

Holding the **control** or **command** key while clicking on a tiddler link opens the target tiddler but doesn't navigate to it. This can be a useful way

of queueing up tiddlers to be read later.

Links are useful for modelling organic relationships between tiddlers, and particularly for expressing the navigational paths between tiddlers.

The InfoPanel lists incoming links to a tiddler in the tab **References**.

Filters can include the following filter operators that work with links:

- `[links[]]` - returns the titles of the tiddlers that are linked from the currently selected tiddler(s)
- `[backlinks[]]` - returns the titles of the tiddlers that link to the currently selected tiddler(s)

TiddlyWiki5 alters the appearance of tiddler links to convey additional information about the target of the link:

Link description	Link appearance
To a tiddler that exists	LikeThis
To a tiddler that doesn't exist	*LikeThis*
To a shadow tiddler that has not been overridden	**LikeThis**
To a shadow tiddler that has been overridden by an ordinary tiddler	LikeThis

External links are shown like this: https://tiddlywiki.com/ or like this.

Tiddlers

21st November 2014 at 12:17pm

Concepts

Tiddlers are the fundamental units of information in TiddlyWiki. Tiddlers work best when they are as small as possible so that they can be

reused by weaving them together in different ways.

A "tiddler" is an informal British word meaning a small fish, typically a stickleback or a minnow. Other systems have analogous concepts with generic names like "items", "entries", "entities", "nodes" or "records". TiddlyWiki takes the view that it is better to be confusingly distinctive than confusingly generic.

Internally, tiddlers are a list of uniquely named values called fields. The only field that is required is the `title` field, but useful tiddlers also have a `text` field, and some or all of the standard fields listed in TiddlerFields.

Tiddlers are ubiquitous in TiddlyWiki. They are used to store everything from JavaScript code modules to the settings and state associated with the user interface.

TiddlerWidget

1st September 2017 at 12:12pm

Widgets

Introduction

The TiddlerWidget is used to set a value for the current tiddler variable, valid within the scope of the TiddlerWidget.

Content and Attributes

Attribute	Description

tiddler	The title of the tiddler to become the new current tiddler

CSS Class Variables

The tiddler widget assigns several useful CSS classes to variables that it creates:

missingTiddlerClass

`tc-tiddler-exists` or...
`tc-tiddler-missing` depending on whether the tiddler exists

shadowTiddlerClass

`tc-tiddler-shadow` if the tiddler is a shadow tiddler

systemTiddlerClass

`tc-tiddler-system` if the tiddler is a system tiddler

tiddlerTagClasses

a space separated list of CSS classes named `tc-tagged-{tagname}`,
e.g. `tc-tagged-introduction`

Note: tag names are URI encoded which means that the tag *$:/tags/Macro* appears as the CSS class `tc-tagged-%24%3A%2Ftags%2FMacro`. See How to apply custom styles by tag for more details

You can use these variables like this:

```
<$tiddler tiddler="MyOtherTiddler">
<div class=<<missingTiddlerClass>>>
```

```
<$transclude/>
</div>
</$tiddler>
```

See also **$:/core/ui/ViewTemplate**.

TiddlyBucket - Save to AWS or Google Storage

26th November 2022 at 8:28pm

Chrome Edge Firefox Internet Explorer iOS Linux Mac Opera Safari Saving Windows

TiddlyBucket - Save to AWS or Google Storage using Go

This tool replicates the TiddlyWeb backend API and can read and write the tiddler files to a local directory like the canonical TiddlyWiki5 app. But, in addition, it can do the same with a given a Google Cloud Storage bucket or AWS S3 bucket. Written in the Go programming language

TiddlyChrome by Arlen Beiler

6th January 2021 at 4:10pm

Other Resources

An extension application for Google Chrome that enables TiddlyWiki to save changes directly to the file system.

https://github.com/Arlen22/tiddly-chrome-app/

TiddlyClip by buggyjay

11th March 2022 at 7:44pm
Community Plugins

BuggyJay's TiddlyClip browser extension for Firefox permits clipping of text and graphics from web pages:

http://tiddlyclip.tiddlyspot.com

> TiddlyClip allows parts of webpages to be clipped into a TiddlyWiki, and consists of two parts, the browser addon and the TiddlyWiki plugin. The addon is completely memoryless, any configuration is determined by the current TiddlyWiki that the user has select to work with (we say that the addon is docked to the TiddlyWiki). TiddlyClip is designed to work (in a basic mode) without configuration. Once the addon and plugin are installed, all the user has to do is select which TW to dock to.

TiddlyDesktop

18th November 2017 at 8:53pm
Definitions Platforms

TiddlyDesktop is an app for working with TiddlyWiki files (both TiddlyWikiClassic and TiddlyWiki version 5). It can be installed on Windows, Mac OS X or Linux. It is compatible with TiddlyWiki version 5 and the older TiddlyWikiClassic.

See the Introducing TiddlyDesktop Video

1. Install the latest release of TiddlyDesktop from https://github.com/TiddlyWiki/TiddlyDesktop/releases
2. Run TiddlyDesktop
3. Use the browse button to open TiddlyWiki files
4. Save changes within TiddlyWiki in the usual way

This brief introduction shows how to install and use TiddlyDesktop:

Source

TiddlyDesktop is based on the OpenSource project NW.js. The source is on GitHub:

https://github.com/TiddlyWiki/TiddlyDesktop/

See TiddlyDesktop Releases and How to Customize TiddlyDesktop

TiddlyDesktop Releases

24th September 2018 at 10:24am

Releases

The latest release of TiddlyDesktop can be found at https://github.com/TiddlyWiki/TiddlyDesktop/releases/latest, or you can see a list of all releases.

TiddlyDrive Add-on for Google Drive by Joshua Stubbs

6th January 2021 at 4:10pm

Android Chrome Edge Firefox Internet Explorer iOS Linux Mac Opera Other Resources PHP plugins Safari Saving Windows

An add-on for Google Drive that allows TiddlyWiki files stored there to be opened and saved directly

To find more information, go to the github page of this project: https://github.com/tiddlydrive/tiddlydrive.github.io

> I made an app that lets one edit TiddlyWiki files saved in your Google drive and then saving them back automagically. The page is here https://chrome.google.com/webstore/detail/tiddly-drive/oaphhjhbbabdjnpjpiliepphpmnioolo but I think you need to add it from the Google Drive web interface.
>
> I even added an optional ability to save with ctrl + s hotkeys.
>
> *Joshua's launch post*

TiddlyFox

4th January 2020 at 12:19pm

Definitions

TiddlyFox is an extension for older versions of Firefox that allows standalone TiddlyWiki files to save their changes directly to the file system. TiddlyFox works on both desktop and smartphone versions of *Firefox*. See Saving with TiddlyFox or Saving with TiddlyFox on Android for detailed instructions.

TiddlyFox is now obsolete due to its incompatibility with the latest versions of Firefox - see TiddlyFox Apocalypse. There are many alternatives to TiddlyFox, but none that work in precisely the same way – see GettingStarted for details.

TiddlyFox can be downloaded from the Mozilla Addons site:

https://addons.mozilla.org/en-GB/firefox/addon/tiddlyfox/

> You can also install the latest development version of TiddlyFox direct from GitHub:
>
> https://github.com/TiddlyWiki/TiddlyFox/raw/master/tiddlyfox.xpi

TiddlyFox Apocalypse

24th November 2021 at 10:49pm

TiddlyFox

Summary

On 14th November 2017 Mozilla released Firefox 57, a major new version with many improvements and security enhancements. However, in amongst those improvements are **some fundamental changes to Firefox's security model with the unfortunate effect of making it impossible for TiddlyFox to function**.

TiddlyFox will remain available for people who continue to use older versions of Firefox, but anyone upgrading to the new version will need to choose a new way to handle saving changes with TiddlyWiki.

Happily, several new ways of working with TiddlyWiki now exist so that users have many alternative choices – see GettingStarted for details. The demise of TiddlyFox has provoked several of these recent developments and thus may well ultimately be good for the community.

There is a discussion thread on the TiddlyWiki forums about these developments.

Background

Firefox was first released in November 2004, a few months after the first version of TiddlyWiki. It was in many ways the Millenium Falcon to Microsoft's Death Star (in the shape of Internet Explorer). IE had by then enjoyed more than 5 years as the dominant browser, leading many in the web community to be frustrated that Microsoft's self-serving extensions to HTML had become de facto standards at the expense of innovation that might benefit the web community as a whole.

Firefox quickly became successful because it managed to render web pages with close enough compatibly with Internet Explorer while offering a superior user experience. A large part of the promise of that user experience was the ability for any user to customise every aspect of the browser. Two innovations were behind this:

- The entire user interface of the browser was constructed in XUL, effectively an extension of HTML that enabled it to render conventional user interfaces (at the time, HTML was largely restricted to simple document-oriented layouts). Tweaking a few lines of XUL code could

make wholesale changes to the user interface of the browser
- The Mozilla add-on architecture gave full privileges to add-ons, enabling them to observe and interact deeply with the browser engine itself, and the file system of the computer on which it was running

These two conditions enabled a vibrant ecosystem of Firefox add-ons, many of them extremely popular. In many cases, innovations that were first seen in browser add-ons later became integrated into the browser itself, most notably the web debugger Firebug which was eventually cloned by all the browser manufacturers.

Firefox continued to be extremely popular until Google joined the development of the rival WebKit browser to make Chrome. Google took a very different approach to the trade offs of making a browser, focusing on improving security at the expense of almost all other considerations. They pioneered approaches such as isolating each tab in its own process that were quickly adopted by all other major browsers.

Google's approach precluded them adopting Mozilla's free-for-all approach to add-ons. Instead of having access to the entire browser environment and filing system, add-ons in Chrome see only a restricted subset of what is going on within the browser, and enjoy little or no access to the resources of the host machine.

It was inevitable that Mozilla would eventually adopt Google's approach to browser security vis-a-vis add-ons. There is a point at which it wouldn't be responsible for Mozilla to be releasing a browser that had knowingly worse security than the market leader.

Lessons

Some of the fecundity of the TiddlyWiki ecosystem stems from the adoption of the above two principles from Firefox:

- Making the application user interface out of the same primitives as the application content
- Giving add-ons free rein to observe and interact with all of the internal logic of the application

Those two characteristics present similar security challenges to TiddlyWiki as they did to Firefox. A TiddlyWiki that was primarily focused on security would need to curtail those abilities.

The Future

Innovation on new browser-based user interfaces and capabilities has now shifted from browser extensions to a new generation of frameworks that simplify creation of a custom browser based on an off-the-shelf open source HTML rendering engine. TiddlyDesktop uses nwjs, while *Beaker Browser* uses an alternative called Electron.

TiddlyFox Apocalypse.png

picture



Tiddlyhost

10th April 2023 at 12:50pm

Definitions

Tiddlyhost.com is a hosting service for TiddlyWiki created by Simon Baird. Once you sign up and confirm your email you can create "sites", (i.e. TiddlyWikis), with support for online saving. Sites can be private or public, and you can optionally list them on the taggable and searchable Tiddlyhost Hub where they'll be discoverable by others.

Unlike TiddlySpot, Tiddlyhost is secure, open source, and has proper support for TiddlyWiki5. It also allows uploading existing TiddlyWiki files, supports TiddlyWikiClassic, and lets you claim ownership of your TiddlySpot sites. For more information see the FAQ and the About page.

Tiddlyhost Logo

picture



TiddlyIE

11th December 2013 at 11:42pm

Definitions

TiddlyIE is an extension for Internet Explorer that allows standalone TiddlyWiki files to save their changes directly to the file system. TiddlyIE works with the desktop version of Internet Explorer.

See Saving with TiddlyIE.

TiddlyMap Plugin by Felix Küppers

14th November 2021 at 9:50pm

Community Plugins Other Resources

An interactive network visualisation plugin based on Vis.js. A demo that also contains installation instructions can be found here: http://tiddlymap.org. The plugin's GitHub repository can be found here.

> TiddlyMap is a TiddlyWiki plugin that allows you to link your wiki-topics (tiddlers) in order to create clickable graphs. By creating relations between your topics you can easily do the following:
>
> - Create concept maps and quickly manifest your ideas in tiddlers.
> - Create task-dependency graphs to organise and describe your tasks.
> - Visualise your topic structures to get an immediate grasp of topics and relations.
>
> In general you may create, visualise and describe any network-structure you have in mind.

TiddlyServer by Arlen Beiler

6th January 2021 at 4:10pm

Linux Mac Other Resources Saving Windows

An extension to the Node.js configuration of TiddlyWiki that adds support for static attachments and for working with multiple wikis at the same time.

https://github.com/Arlen22/TiddlyServer

> *TiddlyServer* 2.0 takes the server command of TiddlyWiki on *NodeJS* and adds it to a static file server. This means you can load and serve any TiddlyWiki data folder in the same way you can serve a single file TiddlyWiki.
>
> But you don't need to serve files and folders from just one place, you can serve them from multiple places anywhere on your harddrive (literally anywhere *NodeJS* can stat, readdir, and readFile). You can even organise them into virtual folders (aka aliases in

> Apache and mounts in Express).
>
> The main point, of course, is that you can actually edit your files, not just look at them. Single file *TiddlyWikis* use the put saver, which needs to be patched using a bookmarklet included on the index page. The instructions for this are below under the heading "One thing that needs to be noted".
>
> And, of course, you can edit data folder tiddlywikis just like you were running node tiddlywiki.js data –server, except that you run it on the path that you found it at (e.g. http://localhost/personal/notes/). You can have as many data folders open as you want, they don't conflict (though they will each take memory).
>
> Data folders store individual tiddlers instead of entire wikis. They take less disk space as they also do not store the core and plugins. This means they also save much quicker, especially over the internet. They also save immediately (within 10 seconds or so) and they save drafts.

Tiddlyshow by Mohammad

17th November 2020 at 5:27pm

Community Plugins

Tiddlyshow is a small application of Tiddlywiki for presentation and slideshow. It can also be used as a plugin.

https://kookma.github.io/Tiddlyshow/

Tiddlyshow contains the following features

- Tools for preparing slides
- Shortcut keys for navigation (forward and backward)

- Themes to colorify and customise the slideshow

TiddlySpace

21st April 2018 at 6:26pm

Definitions

TiddlySpace was a public TiddlyWiki hosting service that operated from 2010 until 2016.

TiddlySpace, and the underlying TiddlyWeb, was originally sponsored by Osmosoft at BT.

TiddlyWeb

27th January 2017 at 11:14pm

Definitions

TiddlyWeb is a reference implementation for an interface to put Tiddlers on the web.

It was created by a team led by Chris Dent at Osmosoft under BT from 2008 to 2012.

http://tiddlyweb.com/

TiddlyWeb can be used to host TiddlyWiki and TiddlyWiki5 wikis, making the individual tiddlers available over a flexible HTTP API.

Other implementations of the API include:

- TiddlyWiki App Engine Server, a 300-line Go implementation from Russ Cox

- TiddlyWiki 5 server module, the bare-bones subset of the API implemented in TiddlyWiki version 5 for Node.js
- tiddly-wiki-server, an implementation based on Rust and SQLite

As of early 2017, none is currently as complete as TiddlyWeb itself.

TiddlyWeb JSON tiddler format

2nd October 2018 at 3:11pm

WebServer Guides

The web server API uses tiddlers in a special format originally designed for TiddlyWeb:

- Field values are represented as strings. Lists (like the **tags** and **list** fields) use double square brackets to quote values that contain spaces
- Tiddlers are represented as an object containing any of a fixed set of standard fields, with custom fields being relegated to a special property called **fields**
- The standard fields are: **bag**, **created**, **creator**, **modified**, **modifier**, **permissions**, **recipe**, **revision**, **tags**, **text**, **title**, **type**, **uri**

For example, consider the following tiddler:

```
{
    "title": "HelloThere",
    "tags": "FirstTag [[Second Tag]]",
    "my-custom-field": "Field value"
}
```

In transit over the API, the tiddler would be converted to the following format:

```
{
    "title": "HelloThere",
    "tags": "FirstTag [[Second Tag]]",
    "fields": {
        "my-custom-field": "Field value"
    }
}
```

TiddlyWiki

27th January 2017 at 11:14pm

Concepts

TiddlyWiki is a rich, interactive tool for manipulating complex data with structure that doesn't easily fit into conventional tools like spreadsheets or wordprocessors.

TiddlyWiki is designed to fit around your brain, helping you deal with the things that won't fit. The fundamental idea is that information is more useful and reusable if we cut it up into the smallest semantically meaningful chunks – tiddlers – and give them titles so that they can be structured with links, tags, lists and macros. Tiddlers use a WikiText notation that concisely represents a wide range of text formatting and hypertext features. TiddlyWiki aims to provide a fluid interface for working with tiddlers, allowing them to be aggregated and composed into longer narratives.

People love using TiddlyWiki. Because it can be used without any complicated server infrastructure, and because it is open source, it has brought unprecedented freedom to everyone to keep their precious information under their own control.

TiddlyWiki was originally created by JeremyRuston and is now a thriving open source project with a busy Community of independent

developers.

TiddlyWiki Archive

5th October 2023 at 11:05pm

About

Older versions of TiddlyWiki are available in the archive:

Version	Released	Lifetime	Summary	Download
v5.3.1	20th Aug 2023	Current	Bug fix release for v5.3.0	Complete | Empty
v5.3.0	1st Jul 2023	50 days	Parameterised transclusions, procedures, functions, custom widgets	Complete | Empty
v5.2.7	26th Mar 2023	98 days	Bug fix release for v5.2.6	Complete | Empty
v5.2.6	20th Mar 2023	6 days	Markdown improvements, indentable pragmas, accessible save wiki button	Complete | Empty
v5.2.5	19th Dec 2022	91 days	Hot fix release for v5.2.4	Complete | Empty
v5.2.4	13th Dec 2022	7 days	Hot fixes for v5.2.3, Twitter archivist plugin, GenesisWidget, JSON read operators, nested macro definitions	Complete | Empty
v5.2.3	2nd Aug 2022	134 days	Minor fixes and improvements	Complete | Empty
v5.2.2	25th Mar	130	Minor bug fix release	Complete |

	2022	days		Empty
v5.2.1	8th Dec 2021	108 days	Filter cascades, LetWidget, trigonometric operators	Complete | Empty
v5.2.0	3rd Oct 2021	66 days	JSON store area, nestable macros, counter attribute for ListWidget, MessageCatcherWidget	Complete | Empty
v5.1.23	24th Dec 2020	284 days	Switchable page templates, EventCatcherWidget, Rename during import, many plugin and filter improvements	Complete | Empty
v5.1.22	15th Apr 2020	253 days	Dynamic plugin loading, compare operator, new plugins: Menubar, Freelinks, Dynannotate, Share	Complete | Empty
v5.1.21	10th Sep 2019	219 days	Bug fix release for v5.1.20	Complete | Empty
v5.1.20	9th Aug 2019	33 days	New conditional, mathematics and string operators, GitHub Saver, save wiki folder command, Innerwiki Plugin	Complete | Empty
v5.1.19	20th Dec 2018	232 days	Bug fix release for v5.1.18	Complete | Empty
v5.1.18	6th Dec 2018	15 days	Global keyboard shortcuts, HTTP server improvements, support for splash screens, ~ filter run prefix, external JS support	Complete | Empty
v5.1.17	12th May 2018	208 days	Minor bug fix release	Complete | Empty
v5.1.16	25th Apr 2018	17 days	Dynaview Plugin, import previews, DiffTextWidget, rotate left in bitmap editor, StartupActions	Complete | Empty
v5.1.15	13th Nov 2017	163 days	Explorer tab, whitespace pragma, save and render commands	Complete | Empty
v5.1.14	26th Apr	202	Drag and drop improvements, initial RTL support, plugins for XLSX import, QR Codes,	Complete |

	2017	days	BibTeX, Google Analytics, Twitter	Empty
v5.1.13	25th Jul 2016	276 days	Bug fix release for v5.1.12	Complete | Empty
v5.1.12	13th Jul 2016	12 days	Editor toolbars, improved bitmap editor, Internals plugin, WikifyWidget	Complete | Empty
v5.1.11	30th Jan 2016	165 days	Bug fix release for v5.1.10	Complete | Empty
v5.1.10	8th Jan 2016	23 days	Text slicer, fold/unfold, performance optimisations, translations, external text tiddlers	Complete | Empty
v5.1.9	3rd Jul 2015	189 days	Fluid-fixed layout, vars widget, open in new window	Complete | Empty
v5.1.8	17th Apr 2015	77 days	Plugin library, Railroad Plugin, sticky titles, 7 new translations	Complete | Empty
v5.1.7	19th Dec 2014	119 days	Hot fix release for v5.1.7	Complete | Empty
v5.1.6	19th Dec 2014	1 day	Minor bug fix release for v5.1.5	Complete | Empty
v5.1.5	26th Nov 2014	24 days	Export button, more ActionWidgets, Danish and Greek translations	Complete | Empty
v5.1.4	22nd Oct 2014	35 days	Dragging links into text boxes, coloured errors and warnings under Node.js	Complete | Empty
v5.1.3	20th Oct 2014	2 days	Journals, ActionWidgets, **`addprefix`** and **`addsuffix`** operators, "includeWikis"	Complete | Empty
v5.1.2	27th Sep	24 days	Minor fixes	Complete |

	2014			Empty
v5.1.1	22nd Sep 2014	6 days	KaTeX Plugin	Complete | Empty
v5.1.0	20th Sep 2014	2 days	First non-beta release	Complete | Empty

TiddlyWiki Camp Paris

12th June 2015 at 12:01pm

Meetups

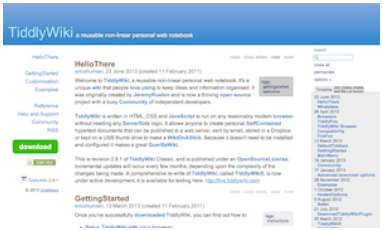
The first TiddlyWiki Camp Paris was held on Saturday 6th June 2015.

http://paris.twcamp.info

> Bienvenue sur le site du TiddlyWiki Camp. Un évènement pour rencontrer la communauté de ce logiciel Open Source, libre et gratuit. Découvrez ce bloc note personnel polyvalent et adoptez-le pour gérer votre quotidien !

TiddlyWiki Classic.png

picture



TiddlyWiki Cloud

7th May 2020 at 10:29pm

Android Chrome Edge Firefox Internet Explorer iOS Linux Mac Opera PHP Safari Saving Windows

Originally built by Jeremy Ruston and now maintained by Arlen Beiler, TiddlyWiki Cloud (formerly known as TiddlyWiki in the Sky for Dropbox) is an online service that lets you edit TiddlyWiki documents directly in your own Dropbox using just a browser.

It works with TiddlyWiki 5. As for TiddlyWiki Classic, the mainstream loader does not work, and a newer loader may work.

https://twcloud.github.io/

TiddlyWiki European Meetup 2016

12th July 2016 at 2:15pm

Meetups

https://tiddlywiki.com/tiddlywiki-eu-meetup-2016/

Here's the stream of day 1:

Here's day 2:

TiddlyWiki European Meetup 2017

26th April 2017 at 11:03pm

Meetups

The TiddlyWiki European Meetup 2017, Oxford, on June 10th and 11th 2017.

For more details:

https://tiddlywiki.com/tiddlywiki-eu-meetup-2017/

TiddlyWiki extensions for Sublime Text 3 by roma0104

6th January 2021 at 4:10pm

Other Resources

Sublime Text 3 syntax file for TiddlyWiki5 `*.tid` files.

https://github.com/roma0104/sublime-tid

TiddlyWiki extensions for ViM

6th January 2021 at 4:10pm

Other Resources

Vi iMproved syntax file for TiddlyWiki5 `*.tid` files.

https://github.com/roma0104/vim-tid

TiddlyWiki Hangouts

24th November 2021 at 10:42pm

Community Videos

The TiddlyWiki community has held many Google Hangouts over the years. They are announced in the TiddlyWiki Google group and on the TiddlyWiki Twitter account.

Past Hangouts are archived in this YouTube playlist:

TiddlyWiki in the Sky for TiddlyWeb

24th November 2021 at 10:48pm

$:/deprecated

The term "TiddlyWiki in the Sky for TiddlyWeb" was used to refer to the ability for content to be synchronised between TiddlyWiki running in the browser and a TiddlyWeb server. This configuration should still work but is no longer commonly used.

TiddlyWiki Marketplace

4th December 2022 at 5:56pm

Community



Welcome to the TiddlyWiki Marketplace. This new forum is designed to provide a space for individuals and organizations to offer commercial products and services that are built around TiddlyWiki:

- Paid hosting services
- Custom solution development
- Training courses
- One-to-one training sessions
- "Fix my wiki" debugging sessions

TiddlyWiki Marketplace

TiddlyWiki Marketplace Banner

picture



TiddlyWiki Newsletter

Subscribe to the TiddlyWiki Newsletter here:

TiddlyWiki Newsletter Badge.png

picture



TiddlyWiki on Firefox for Android Video

12th September 2014 at 5:03pm

Saving with TiddlyFox Videos

This brief screencast shows how to setup Firefox for Android so you can save changes to TiddlyWiki:

TiddlyWiki on Node.js

17th June 2022 at 1:44pm

Platforms

Running TiddlyWiki on Node.js brings several important benefits over and above the single file version:

- Individual tiddlers are stored in separate files, which you can organise as you wish
- The ability to build multiple wikis that blend different combinations of shared and unique content
- You can edit your content on any modern browser, including smartphones and tablets



Note that TiddlyWiki on Node.js doesn't currently support directly modifying the tiddler files via the file system while it is running. The server must be restarted in order to for changes to take effect. The recommended way to interact with a running wiki is via the HTTP or JavaScript APIs.

For more information see:

- Building TiddlyWikiClassic
- Customising Tiddler File Naming
- Environment Variables on Node.js
- Generating Static Sites with TiddlyWiki
- How to build a TiddlyWiki5 from individual tiddlers
- Installing custom plugins on Node.js
- Installing official plugins on Node.js
- Internet Information Services
- Node.js
- Installing TiddlyWiki Prerelease on Node.js
- MultiTiddlerFiles
- MultiTiddlerFileSyntax
- NamedCommandParameters
- Scripts for TiddlyWiki on Node.js
- Node.js on Termux
- TiddlerFiles
- tiddlywiki.files Files

- tiddlywiki.info Files
- TiddlyWikiFolders
- Uninstalling a plugin with Node.js
- Upgrading TiddlyWiki on Node.js
- Using a custom path prefix with the client-server edition
- Using TiddlyWiki for GitHub project documentation
- Using TiddlyWiki on Node.js
- Working with the TiddlyWiki5 repository

TiddlyWiki Releases

13th November 2017 at 5:14pm

Releases

Here are the details of recent releases of TiddlyWiki5. See TiddlyWiki5 Versioning for details of how releases are named.

If you are using node.js, you can also install prior versions like this:

> npm install -g tiddlywiki@5.1.13

(BetaReleases and AlphaReleases are listed separately).

5.3.3
5.3.2

Release 5.3.3

5.3.1
5.3.0
5.2.7
5.2.6
5.2.5
5.2.4
5.2.3
5.2.2
5.2.1
5.2.0
5.1.23
5.1.22
5.1.21
5.1.20
5.1.19
5.1.18
5.1.17
5.1.16
5.1.15
5.1.14

Released 23rd December 2023 at 11:22

See GitHub for detailed change history of this release



Congratulations to catter-fly for their winning design for the banner for this release (here is the competition thread).

This is a bug fix release to address a number of bugs that were introduced with Release 5.3.2.

- **fixed** handling of a list widget with an empty paragraph as inline template
- **fixed** broken per-tiddler previews
- **fixed** missing comma before skinny tiddlers in JSON store area
- **fixed** handling of whitespace immediately after pragmas
- **fixed** SelectWidget handling of classes and rendering typo

Since v5.3.3 replaces v5.3.2 after only a couple of weeks, here is the release note for v5.3.2.

Release Note for v5.3.2

See GitHub for detailed change history of this release



Congratulations to catter-fly for their winning design for the banner for this release (here is the competition thread).

5.1.13
5.1.12
5.1.11
5.1.10
5.1.9
5.1.8
5.1.7
5.1.6
5.1.5
5.1.4
5.1.3
5.1.2
5.1.1
5.1.0



Major Improvements

Conditional Shortcut Syntax

added a new shortcut syntax for concisely expressing if-then-else logic. This is the first of a new type of wikitext syntax based on tokens delimited with `<%` and `%>`. We plan to introduce other structures using the same format such as a "case" statement.

These new token-based shortcuts allow a richer structure and expressivity than existing features such as widgets or pragmas. For example:

```
<% if [<animal>match[Elephant]] %>
  It is an elephant
<% elseif [<animal>match[Giraffe]] %>
    It is a giraffe
<% else %>
    It is completely unknown
<% endif %>
```

Behind the scenes, the conditional shortcut syntax is rendered as the equivalent ListWidgets.

Explicit Templates for the ListWidget

added support for `<$list-template>` and `<$list-empty>` as immediate children of the **$**`ListWidget` widget to specify the

list item template and/or the empty template.

This new feature is designed to replace a common pattern of using the `emptyMessage` attribute of the ListWidget to render complex wikitext that thus has to be quoted. Working with wikitext within quotes is awkward and error prone. The new structure can be somewhat faster because it allows the empty message to be parsed in advanced of rendering.

For example:

```
<$list filter=<<filter>>>
    <$list-template>
        <$text text=<<currentTiddler>>/>
    </$list-template>
    <$list-empty>
        None!
    </$list-empty>
</$list>
```

Note that the **`emptyMessage`** and **`template`** attributes take precedence if they are present.

Joiners for the ListWidget

added a **`join`** attribute to the **`$`**`ListWidget` widget to insert a short piece of text between list items. This is both easier to use and faster than using the **`counter`** attribute for the same purpose. So if your list looked like this:

```
<$list filter=<<filter>> counter="counter" variable="item">
<$text text=<<item>>/><$list filter="[<counter-last>match[no]]" variable="ignore"><$text text=",
```

```
"/></$list>
</$list>
```

You can replace it with:

```
<$list filter=<<filter>> variable="item" join=", "><$text text=<<item>>/></$list>
```

If the joiner text that you need is long and awkward to write in an attribute, you can use the new `<$list-join>` widget. Like `<$list-template>` and `<$list-empty>`, it must be an immediate child of the **$**`ListWidget`:

```
<$list filter=<<filter>> variable="item"><$text text=<<item>>/><$list-join>, and <em>also</em>
let's not forget </$list-join></$list>
```

jsonset operator

added jsonset Operator for setting values within JSON objects

QR Code Reader

extended QR Code plugin to be able to read QR codes and a number of other bar code formats

Translation improvements

Improvements to the following translations:

- Chinese
- Polish
- Spanish

Plugin Improvements

- **fixed** Comments Plugin to use predefined palette colours
- **improved** Evernote Importer Plugin to support images and other attachments
- **added** `$floating` attribute to Dynannotate Plugin to support popups that do not disappear when another part of the screen is clicked. Instead they have to dismissed manually

Widget Improvements

- **fixed** ImageWidget encoding for more image types
- **extended** ImageWidget to add a "usemap" attribute
- **improved** the ScrollableWidget to allow the scroll position to be bound to a tiddler, so that changes to the tiddler affect the scroll position, and vice versa

Usability Improvements

- **updated** editor preview button to automatically focus the editor

- **improved** file type names in the export menu

Hackability Improvements

- **extended** all the relevant core widgets to allow arbitrary `data-*` attributes and `style.*` attributes to be applied to the generated DOM nodes. This is useful for passing data to the EventCatcherWidget
- **extended** jsonextract Operator, jsonget Operator, jsonset Operator and jsontype Operator to allow negative indexes into arrays to be counted from the end of the array
- **improved** the default page layout to better support CSS grid and flexbox layouts
- **improved** the editor to use grid layout, simplifying customisation

Bug Fixes

- **fixed** ordering of Vanilla stylesheets
- **fixed** missing closing tag in tag-pill-inner macro
- **removed** invalid "type" attribute from textarea elements generated by the EditTextWidget
- **fixed** editor "type" dropdown state tiddlers
- **fixed** handling of "counter-last" variable when appending items with the ListWidget
- **fixed** upgrade download link in Firefox
- **fixed** refreshing of transcluded functions
- **fixed** resizing of height of textareas in control panel

- **fixed** encodebase64 Operator and decodebase64 Operator to work properly with binary data
- **fixed** WidgetMessage: tm-open-window when opening an existing window to bring it to the front and focus it
- **fixed** behaviour of last Operator when zero items selected
- **fixed** incorrectly setting focus on field name input field when deleting field using the delete field button
- **fixed** Table-of-Contents Macros to not show expander icon for a sublist that has all children excluded
- **fixed** overflow of CodeMirror Plugin editor within grid container
- **fixed** wikitest parser removing whitespace when parsing pragmas
- **fixed** tooltip for editor add field button
- **fixed** plain text parser being susceptible to the CodeBlockWidget being redefined
- **fixed** pragmas not working within the action string of several core macros

Node.js Improvements

- **fixed** a significant flaw in the synchronisation algorithm used by the client-server configuration. The flaw could lead to tiddlers temporarily disappearing from the browser

Performance Improvements

- **improved** performance of predefined patterns with all Operator
- **updated** favicon format to PNG

Developer Improvements

- **improved** global hook handling to support removing hooks
- **added** some useful npm scripts to `package.json`

Infrastructure Improvements

- **improved** Continuous Integration tests to use Playwright to run our browser-based tests
- **added** an automatic build of the external core TiddlyWiki at https://tiddlywiki.com/empty-external-core.html

Acknowledgements

@Jermolene would like to thank the contributors to this release who have generously given their time to help improve TiddlyWiki:


@Anthon…

@BramC…

@Buckar…

@Burnin…

@CrossEye

@Evident…

@Gk0Wk

@joebord…

@kookma











wo @mateus… @oflg @pille1842 @pmario @rmunn @saqimtiaz @simonb… @T1mL3…

@yaisog

TiddlyWiki Video Tutorials by Francis Meetze

6th January 2021 at 4:10pm

Tutorials

An extensive tutorial for getting started with TiddlyWiki. Recommended

https://www.youtube.com/playlist?list=PLzZCajspPU_UjFn0uy-J9URz0LP4zhxRK

> This tutorial is aimed at giving you a basic foundation so that you can start using TiddlyWiki right away. Here we take a look at the technology behind Tiddlywiki, how to install it and create your first Tiddler.

tiddlywiki.files Files

14th November 2021 at 11:12am

TiddlyWiki on Node.js TiddlyWikiFolders

Introduction

A `tiddlywiki.files` JSON file in a sub-folder within a TiddlyWiki folder overrides the usual logic for recursively scanning the folder for tiddler files. Instead, the `tiddlywiki.files` file specifies instructions for loading tiddlers from specific files and folders.

The format of the file is an object with two optional properties:

- **tiddlers** - an array of objects describing external files with the ability to override or modify any of the fields read from the file
- **directories** - an array of objects describing external directories, a filter determining which files within those directories should be processed, and the ability to override or modify any of the fields read from the file

Note that significant enhancements to `tiddlywiki.files` processing were introduced in Release 5.1.14.

Field overrides

Both the **tiddlers** and **directories** sections of `tiddlywiki.files` files include the ability to override or customise the values of fields with a `fields` object.

Each field can be specified as either a **string** or **array** value to be assigned directly to the field, or ⓘ **Introduced in v5.1.14** an **object** describing how to generate the value for the field. The object contains the following properties:

- **source** - (optional) a string specifying the source value for the field. If not specified, the existing value is used
  - *filename* the filename of the file containing the tiddler
  - *filename-uri-decoded* the filename of the file containing the tiddler, with URI decoding applied
  - *basename* the filename of the file containing the tiddler without any extension
  - *basename-uri-decoded* the filename of the file containing the tiddler without any extension, with URI decoding applied
  - *extname* the extension of the filename of the file containing the tiddler

  - *created* the creation date/time of the file containing the tiddler
  - *modified* the modification date/time of the file containing the tiddler
  - ⓘ **New in v5.3.0** *filepath* the path of the file containing the tiddler, relative to the **path** property of the directory (only usable in **directories** declarations)
  - ⓘ **New in v5.3.0** *subdirectories* an array of the subdirectories in the file's path relative, to the **path** property of the directory (only usable in **directories** declarations)
- **prefix** - (optional) a string to be prepended to the value of the field
- **suffix** - (optional) a string to be appended to the value of the field

Tiddlers section

The file specifications in the `tiddlers` array support the following properties:

- **file**: (required) the absolute or relative path to the file containing the tiddler data (relative paths are interpreted relative to the path of the `tiddlywiki.files` file)
- **isTiddlerFile**: (optional) if `true`, the file will be treated as a tiddler file and deserialised to extract the tiddlers. Otherwise, the raw content of the file is assigned to the `text` field without any parsing
- **fields**: (optional) an object containing values that override or customise the fields provided in the tiddler file (see above)
- **prefix** & **suffix**: (optional) strings to be prefixed and suffixed to the tiddler `text` field

> Note that providing a **prefix** here is equivalent to setting the `text` field of the **fields** object to `{"prefix":"<prefixvalue>"}`.

Directories section

Directory specifications in the `directories` array may take the following forms:

- a **string** literal, specifying the absolute or relative path to the directory containing the tiddler files (relative paths are interpreted relative to the path of the `tiddlywiki.files` file). The directory is recursively searched for tiddler files
- ⓘ **Introduced in v5.1.14** an **object** with the following properties:
  - **path** - (required) the absolute or relative path to the directory containing the tiddler files (relative paths are interpreted relative to the path of the `tiddlywiki.files` file). Note that by default the directory is not recursively searched; sub-directories are ignored unless the *searchSubdirectories* flag is set to `true` (see below).
  - **filesRegExp** - (optional) a regular expression that matches the filenames of the files that should be processed within the directory
  - **isTiddlerFile** - (required) if `true`, the file will be treated as a tiddler file and deserialised to extract the tiddlers. Otherwise, the raw content of the file is assigned to the `text` field without any parsing
  - **isEditableFile** - ⓘ **Introduced in v5.1.23** (optional) if `true`, changes to the tiddler be saved back to the original file. The tiddler will be saved back to the original filepath as long as it does not generate a result from the *$:/config/FileSystemPath* filters, which will override the final filepath generated if a result is returned from a filter.
  - **searchSubdirectories** - ⓘ **Introduced in v5.1.23** (optional) if `true`, all subdirectories of the *path* are searched recursively for files that match the (optional) *filesRegExp*. If no *filesRegExp* is provided, all files in all subdirectories of the *path* are loaded. Tiddler titles generated via the *source* attribute *filename* (see above) will only include the filename, not any of the subdirectories of the path. If this results in multiple files with loaded with the same tiddler title, then only the last file loaded under that tiddler title will be in memory. In order to prevent this, you can use the *filepath* attribute instead of *filename*. Alternately, you can include multiple directory objects and customise the title field with a *prefix* or *suffix* alongside the *source* attribute.
  - **fields** - (required) an object containing values that override or customise the fields provided in the tiddler file (see above)

Fields can also be overridden for particular files by creating a file with the same name plus the suffix `.meta` – see TiddlerFiles.

Examples

These example `tiddlywiki.files` must be placed in their own sub-directory of the wiki folder.

There are also several examples of `tiddlywiki.files` files in the main TiddlyWiki 5 GitHub repository.

Importing a folder of PDFs

This example retrieves all the files with the extension `.pdf` from a folder specified by a relative path. This path starts with "../../../" indicating 3 directory levels above the folder holding this config file. Each tiddler is set up for LazyLoading with the following fields:

- **title** - set to the URI decoded base filename of the PDF file. URI decoding allows characters like "/" to be included in titles by URI encoding them as "%2F"
- **created** - set to the creation date/time of the PDF file
- **modified** - set to the modification date/time of the PDF file
- **type** - set to `application/pdf`
- **tags** - set to `$:/tags/AttachedFile`
- **text** - set to an empty string
- **_canonical_uri** - set to the string "pdfs/" concatenated with the filename

```
{
    "directories": [
```

```
        {
            "path": "../../../input/pdfs",
            "filesRegExp": "^.*\\.pdf$",
            "isTiddlerFile": false,
            "fields": {
                "title": {"source": "basename-uri-decoded"},
                "created": {"source": "created"},
                "modified": {"source": "modified"},
                "type": "application/pdf",
                "tags": ["$:/tags/AttachedFile"],
                "text": "",
                "_canonical_uri": {"source": "filename", "prefix": "pdfs/"}
            }
        }
    ]
}
```

Importing a folder of text files

This example retrieves all the files with the extension `.txt` from a folder specified by a relative path. This folder is within the wiki's base directory, and the current config file is in a directory within the wiki's "tiddlers/" directory. So, in this case the path starts with "../../" to traverse upwards two directory levels, and then down into the "externalnotes/" directory. Each tiddler is set up with the following fields:

- **title** - set to the URI decoded base filename of the text file. URI decoding allows characters like "/" to be included in titles by URI encoding them as "%2F"

- **created** - set to the creation date/time of the text file
- **modified** - set to the modification date/time of the text file
- **type** - set to `text/plain`
- **tags** - set to `[[note]] [[externalnote]] [[.txt]]` (using array notation)
- **text** - not set, thus the content of the file is loaded as the text field

```
{
    "directories": [
        {
            "path": "../../externalnotes",
            "filesRegExp": ".+\\.txt",
            "isTiddlerFile": false,
            "isEditableFile": true,
            "fields": {
                "title": {"source": "basename-uri-decoded"},
                "created": {"source": "created"},
                "modified": {"source": "modified"},
                "type": "text/plain",
                "tags": ["note", "externalnote", ".txt"]
            }
        }
  ]
}
```

This will load all text files in the `../../externalnotes/` directory into the wiki as individual tiddlers. These can be a collection of snippets in

various markup-languages. Then, the `type` field of each of these tiddlers can be changed to match their languages For example, "text/vnd.tiddlywiki" for wikitext, or "text/markdown" for markdown files. Then, using *$:/config/FileSystemPaths* and *$:/config/FileSystemExtentions* tiddlers with the following lines we can cause any changes to these tiddlers to be saved back to the directory they started from, and also as "*.txt" files with accompanying "*.txt.meta" files. These meta files will be generated as required, and will then over-ride any fields generated from the config `tiddlywiki.files` file (such as the tiddler's `type` field) when the server is restarted.

From the examples in Customising Tiddler File Naming we see that the final `[!tag[externalnote]addprefix[wiki/]]` filter in the *$:/config/FileSystemPaths* tiddler excludes all tiddlers tagged with `externalnotes` (that have not matched an earlier filter). These tiddlers have their filepath retrieved from the $:/config/OriginalTiddlerPaths generated upon boot startup.

Then, the `[tag[.txt]then[.txt]]` filter in the *$:/config/FileSystemExtensions* tiddler forces all these tiddlers to be saved back to disk as *.txt and accompanying *.txt.meta files (overriding the normal tiddler-type to file-type mapping). In this case, allowing the snippets of Tiddlywiki wikitext or markdown-text to be saved back to "text" *.txt files.

Importing and auto-tagging images

This example imports all the image files in the `files` directory and all its subdirectories as external-image tiddlers, and tags them based on their filepath. Each tiddler is set up with the following fields:

- **title** - set to the URI decoded base filename of the text file
- **created** - set to the creation date/time of the text file
- **modified** - set to the modification date/time of the text file
- **type** - set to `image/jpeg`. There is currently no way to infer the correct ContentType of the image tiddler from the file, but `image/jpeg` tiddlers should render correctly even with png or gif images. As an alternative, you could create separate definitions for jpg, png, and gif files with the `image/jpeg`, `image/png`, and `image/gif` types respectively.

- **tags** - generated based on the path of the image relative to the parent directory (`files` in this case). Eg, images in `files/photos` will be tagged with `photos`, those in `files/photos/family` will be tagged with both `photos` and `family`, and those in the root `files` directory will have no tags.
- **text** - set to an empty string
- **_canonical_uri** - set to the full filepath of the image relative to the wiki root

```
{
    "directories": [
        {
            "path": "../../files/",
            "filesRegExp": "^.*\\.(?:jpg|jpeg|png|gif)$",
            "isTiddlerFile": false,
            "searchSubdirectories": true,
            "fields": {
                "title": {"source": "basename-uri-decoded"},
                "created": {"source": "created"},
                "modified": {"source": "modified"},
                "type": "image/jpeg",
                "tags": { "source": "subdirectories" },
                "text": "",
                "_canonical_uri": { "source": "filepath", "prefix": "files/" }
            }
        }
    ]
}
```

tiddlywiki.info Files

14th November 2021 at 11:12am

TiddlyWiki on Node.js TiddlyWikiFolders

TiddlyWikiFolders are configured with a single `tiddlywiki.info` file in the root of the wiki folder. It should contain a JSON object comprising the following properties:

- **plugins** - an array of plugin names to be included in the wiki
- **themes** - an array of theme names to be included in the wiki
- **languages** - an array of language names to be included in the wiki
- **includeWikis** - an array of references to external wiki folders to be included in the wiki
- **build** - a hashmap of named build targets, each defined by an array of command tokens (see BuildCommand)
- **config** - an optional hashmap of configuration options (see below)

includeWikis

The entries in the **includeWikis** array can be either a string specifying the relative path to the wiki, or an object with the following fields:

- **path** - relative path to wiki folder
- **read-only** - set *true* to prevent the tiddlers in the included wiki from being modified. The modifications will be written to the directory specified in **default-tiddler-location**, described below

build

Note that the build targets of included wikis are merged if a target of that name isn't defined in the current `tiddlywiki.info` file.

config

Configuration options include:

- **default-tiddler-location** - a string path to the default location for the filesystem adaptor to save new tiddlers (resolved relative to the wiki folder)
- **retain-original-tiddler-path** - If true, the server will generate a tiddler $:/config/OriginalTiddlerPaths containing the original file paths of each tiddler in the wiki

Example

For example:

```
{
    "plugins": [
        "tiddlywiki/tiddlyweb",
        "tiddlywiki/filesystem"
    ],
    "includeWikis": [
        {"path": "../tw5.com", "read-only": true}
    ],
    "build": {
        "index": [
            "--rendertiddler","$:/core/save/all","index.html","text/plain"],
        "favicon": [
            "--savetiddler","$:/favicon.ico","favicon.ico",
            "--savetiddler","$:/green_favicon.ico","static/favicon.ico"]
    },
    "config": {
```

```
        "retain-original-tiddler-path": true
    }
}
```

TiddlyWiki.mp3

TiddlyWiki2ReadMe

Building TiddlyWikiClassic

TiddlyWiki5 can be used to build older 2.x.x versions of TiddlyWikiClassic from their constituent components. Doing so involves these features:

- The `tiddlywiki/classictools` plugin, containing a deserializer module which allows tiddlers to be loaded from TiddlyWiki 2.x.x `.recipe` files
- The `stripcomments` format for the ViewWidget, which strips single line JavaScript comments starting `//#`
- The `stripTitlePrefix='yes'` attribute of the FieldsWidget, which removes prefixes wrapped in curly braces from the `title` attribute
    - For example, `{tiddler}HelloThere` would be transformed to `HelloThere`

Usage

TiddlyWikiClassic is built from the command line by running TiddlyWiki on Node.js. A typical usage would be:

```
node ../../tiddlywiki.js \
    --verbose \
    --load <path_to_recipe_file> \
    --rendertiddler $:/core/templates/tiddlywiki2.template.html <path_to_write_index_file> text/plain \
    || exit 1
```

TiddlyWiki5

13th June 2022 at 4:50pm

Concepts

TiddlyWiki5 is a reboot of TiddlyWiki for the next 25 years. It is a complete interactive wiki in JavaScript that can be run in the browser or on the server under Node.js.

The TiddlyWiki repository can be found at GitHub.

TiddlyWiki5 Versioning

12th September 2014 at 4:26pm

Releases

Each release of TiddlyWiki5 is identified by a version number that complies with the Semantic Versioning 2.0.0 standard.

TiddlyWiki Core Version

According to the standard:

```
Given a version number MAJOR.MINOR.PATCH, increment the:

MAJOR version when you make incompatible API changes,
MINOR version when you add functionality in a backwards-compatible manner, and
PATCH version when you make backwards-compatible bug fixes.
Additional labels for pre-release and build metadata are available as extensions to the MAJOR.MINOR.PATCH
format.
```

Alpha and Beta versions

While in alpha TiddlyWiki5 uses the pre-release label "alpha", for example:

```
5.0.1-alpha
```

Each new alpha or beta release will bump the `PATCH` version number. This breaks the strict semantics of versioning because `PATCH` increments are supposed to be reserved for compatible changes.

*Note that prior to **5.0.1-alpha**, TiddlyWiki5 used version numbers formatted as **5.0.0-alpha.19**. The change was made to enable the upgrade mechanism to recognise plugin updates from the version information.*

Interim versions

During development when a new release is being prepared, the pre-release label is set to `prerelease`.

Plugin Versions

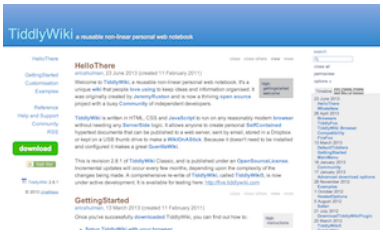
Version numbers

TiddlyWiki5 uses the version information attached to plugins for determining which of two plugins is more recent during an upgrade or import. The pre-release label is ignored when performing these comparisons.

TiddlyWikiClassic

5th January 2014 at 5:32pm

Definitions



"TiddlyWiki Classic" refers to versions prior to 5.0, before TiddlyWiki was completely rewritten. TiddlyWiki Classic is still actively maintained at:

https://classic.tiddlywiki.com/

TiddlyWikiFolders

15th October 2016 at 3:44pm

TiddlyWiki on Node.js

As well as traditional single file wikis, TiddlyWiki on Node.js supports wikis that are stored as a folder of individual tiddler files.

Wiki Folder Structure

Wiki folders can contain the following files and folders:

- **tiddlywiki.info** - JSON file containing metadata for the wiki – see tiddlywiki.info Files
- **tiddlers** - folder containing tiddler files comprising the wiki
- **plugins** - folder containing plugin folders to be included in the wiki
- **languages** - folder containing plugin folders for language plugins to be included in the wiki
- **themes** - folder containing plugin folders for theme plugins to be included in the wiki

Only the **tiddlywiki.info** file is required; everything else is optional. Any files and folders not listed above are ignored.

Plugin processing

To be usable in the browser, plugins just need to be included in the wiki. For wikis that are generated on the server, TiddlyWikiFolders can contain a tiddlywiki.info file that identifies the plugins to be included in this wiki:

```
{
    "plugins": [
        "tiddlywiki/slider",
        "tiddlytools/chooser"
    ]
}
```

Plugins names refer to plugin folders listed in TiddlyWiki5's root `plugins` folder. Plugins can also be included manually by copying them into the `plugins` subfolder of the wiki.

Processing of `tiddlers` folder

All the TiddlerFiles in the `tiddlers` folder are read into the wiki at startup. Sub-folders are scanned recursively for TiddlerFiles. Newly created tiddlers are stored in TiddlerFiles directly beneath the `tiddlers` folder, unless configured otherwise.

The default processing for sub-folders within the `tiddlers` folder can be overridden by providing a JSON file called `tiddlywiki.files` – see tiddlywiki.files Files.

TiddlyWikiLinks

picture



Tidme by oflg

5th October 2023 at 8:02am

Community Editions

Lifelong knowledge, deep in Mind.

https://github.com/oflg/Tidme

Tidme uses advanced Incremental Learning concepts to make it your powerful second brain for acquiring lifelong knowledge.

- Read Articles like *SuperMemo*
- Learn languages like *LingQ*
- Memory Notes like Anki

timeline Macro

23rd February 2017 at 4:36am

Core Macros Macros

The **timeline** macro returns a list of tiddlers in reverse chronological order of modification (or some other date field), grouped by day.

Parameters

limit

The maximum number of tiddlers to include, defaulting to 100. But if *ANY* tiddlers are included for a particular day, *ALL* of the other tiddlers for that day will also be included – even if this exceeds the limit

format

A string specifying the desired format, defaulting to `DDth MMM YYYY`

subfilter

An optional extra filter step, e.g. `tag[MyTag]`

dateField

The name of the date field to use, defaulting to `modified`

The tiddlers are selected by means of a filter expression, into which the **`subfilter`** and **`limit`** parameters are spliced as follows:

> `[!is[system]$subfilter$has[$dateField$]!sort[$dateField$]limit[$limit$]eachday[$dateField$]]`

Examples

timeline Macro (Examples)

21st February 2015 at 11:46pm

Macro Examples timeline Macro

copy to clipboard

```
<<timeline format:"DD/MM/YYYY">>
```

Try it

copy to clipboard

```
<<timeline limit:30 subfilter:"tag[Definitions]" format:"DD/MM/YYYY">>
```

Try it

Timelines by Mohammad

17th November 2020 at 5:17pm

Community Plugins

Timelines plugin is for creating vertical and horizontal timeline and sequences.

https://kookma.github.io/TW-Timelines/

The content or description and data of events are stored in individual tiddlers.

Timelines contain two timeline macros and two sequence macros. A sequence macro shows event contents in sequence not necessarily in chronological order. The order can be determined by tag or list field or other methods.

Timimi: WebExtension and Native Host by Riz

6th January 2021 at 4:10pm

Chrome Edge Firefox Linux Mac Opera Other Resources plugins Saving Windows

Timimi is a web-extension accompanied by a native host that allows you to save and backup your standalone HTML tiddlywiki files **anywhere in your hard-drive**. Once installed, you can save the tiddlywiki files without any extra steps, like the original Tiddlyfox addon.

- The native host requires a component installed on the host computer, outside the browser.

https://ibnishak.github.io/Timimi/

As of version 2.1, Timimi supports the following browsers

- Chrome/Chromium
- Firefox
- Opera
- Microsoft Edge (Chromium)

It is also reported to work seamlessly in chrome based browsers like Brave and Vivaldi.

Timimi also provides users with 4 backup strategies, viz:

- Create a backup Every n^{th} save
- Create a backup every n^{th} minute
- Customised Tower of Hanoi
- First in First Out

Tinka by Andreas Hahn

6th January 2021 at 4:10pm

Community Plugins

The Tinka plugin provides the ability to create and modify plugins in the browser, without requiring the use of Node.js.

See https://github.com/TinkaPlugin/Tinka for the GitHub repo.

https://tinkaplugin.github.io/

> This is a Control Panel extension that aims to simplify the plugin creation and editing process. After installing, you will find a new tab in your control panel that makes creating and modifying plugins a little bit easier.

Title List

19th October 2023 at 5:50pm

Concepts

A **title list** is a line of text that presents one or more tiddler titles, strung together with a space between each one and the next.

If a title *CONTAINS* a space, it needs double square brackets around it:

```
GettingStarted [[Discover TiddlyWiki]] Upgrading
```

Title lists are used in various places, including PermaLinks and the ListField.

They are in fact the simplest case of a filter, and are thus a way of expressing a selection of titles.



The Title List format cannot reliably represent items that contain certain specific character sequences such as `]]` . Thus it should not be used where there is a possibility of such sequences occurring.

See also:

- The format Operator with the 'titlelist' suffix conditionally wraps double square brackets around a string if it contains whitespace

title Operator

3rd February 2015 at 8:17pm

Common Operators Filter Operators Negatable Operators Selection Constructors

purpose	select a single title
input	ignored
! **input**	a selection of titles
parameter	**T** = a tiddler title
output	a selection containing only **T**

`!` **output**	the input, but with tiddler **T** filtered out if it exists in the wiki

Learn more about how to use Filters

`[title[An Example]]` can be shortened to `[[An Example]]`, because **`title`** is the default filter operator. Consequentially, `[!title[An Example]]` can be shortened to `[![An Example]]`.

`title` is a constructor (except in the form `!title`), but **`field:title`** is a modifier.

Examples

title Operator (Examples)

24th January 2015 at 3:03pm

Operator Examples title Operator

```
[title[HelloThere]]
```

Try it

```
[[HelloThere]]
```

Try it

```
HelloThere
```

Try it

```
[title[HelloThere]] [title[Filter Operators]]
```

Try it

```
[[HelloThere]] [[Filter Operators]]
```

Try it

```
HelloThere [[Filter Operators]]
```

Try it

```
[tag[Filters]] +[!title[Filter Operators]]
```

Try it

```
[tag[Filters]] +[![Filter Operators]]
```

Try it

```
[tag[Filters]] -[[Filter Operators]]
```

Try it

Title Selection

24th January 2015 at 9:29pm

Concepts

A **title selection** is an ordered set of tiddler titles (or similar strings), in which no title appears more than once.

Title selections are important in filter processing.

The simplest way to write one down is as a title list.

titlecase Operator

20th June 2019 at 3:50pm

Filter Operators

purpose	returns each item in the list with each word capitalised
input	a selection of titles
output	the input titles with each word capitalised

Learn more about how to use Filters

Introduced in v5.1.20

See also lowercase Operator, uppercase Operator, sentencecase Operator.



This operator does not change anything except the first letter of each word, so `[[aBcD eFgH]]` would become `[[ABcD EFgH]]`. If you want `[[Abcd Efgh]]`, then use a filter like `[lowercase[]titlecase[]]`.

Examples

titlecase Operator (Examples)

20th June 2019 at 4:04pm

Operator Examples titlecase Operator

```
[[abc def ghi jkl]titlecase[]]
```

Try it

```
[[abc deF gHi jKL]titlecase[]]
```

Try it

```
[[abc deF gHi jKL]lowercase[]titlecase[]]
```

Try it

```
[tag[HelloThere]titlecase[]]
```

Try it

Todolist by Mohammad

16th November 2020 at 10:20pm

Community Plugins

Todolist is a small pure wikitext plugin, contain all tools to work with todo list.

https://kookma.github.io/TW-Todolist/

With Todolist, easily organise and prioritise your tasks and projects so you'll always know exactly what to work on next.

Todolist creates custom UI and lets you to add new items, set priority, done/undone items, archive, delete. Using Todolist plugin it is possible to create several todo lists in one Tiddlywiki.

toggle Operator

18th November 2020 at 8:21pm

Filter Operators Listops Operators Order Operators

purpose	toggle the title specified in the parameter in the input
input	a list of items
parameter	the **`toggle`** operator accepts 1 or more parameters, see below for details
output	the input list with the title specified in the parameter toggled

Learn more about how to use Filters

ⓘ Introduced in v5.1.23

The **`toggle`** operator requires at least one parameter and can accept additional optional parameters. With a second optional parameter it can be used to toggle between a pair of titles:

```
[toggle[<title1>],[<title2>]]
```

- **title1** : a title to toggle in the input list. If it is already present, it is removed. Otherwise, it is added.

- **title2**: (optional). When the second parameter is provided, the operator toggles between the two values in the input list. If neither is present, the first parameter takes precedence is added to the list.

With more than two parameters, the **`toggle`** behaves similar to the cycle and can be used to cycle through a list of values. Note that all parameters should be unique.



While the **`cycle`** operator interprets its first parameter as a list of titles to cycle through and offers similar functionality, the **`toggle`** operator accepts an unlimited number of distinct parameters.

Examples

toggle Operator (Examples)

18th November 2020 at 6:47pm

Operator Examples toggle Operator

These examples make use of the Days of the Week tiddler.

```
[list[Days of the Week]] +[toggle[Wednesday]]
```

Try it

To toggle a tag in a tiddler:

```
<$action-listops  $tiddler="target" $tags="+[toggle[done]]" />
```

To toggle a title in a field:

```
<$action-listops $tiddler="target" $field="my-field" $subfilter="+[toggle[done]]" />
```

To toggle between a value pair:

```
<$action-listops $tiddler="target" $field="my-field" $subfilter="+[toggle[todo],[done]]" />
```

Transcluded Attribute Values

15th June 2023 at 6:53am

Widget Attributes WikiText

Transcluded attribute values are indicated with double curly braces around a TextReference. For example:

```
attr={{tiddler}}
attr={{!!field}}
attr={{tiddler!!field}}
```



The value of the attribute value will be the exact text retrieved from the TextReference. Any wiki syntax in that text will be left as-is.

TranscludeWidget

11th May 2023 at 4:26am

Widgets

Introduction

The **$transclude** widget dynamically includes the content from another tiddler or variable, rendering it as if the transclude widget were replaced by the target content.

The **$transclude** widget can be used to render content of any type: wikitext, images, videos, etc.

Transclusion is the underlying mechanism for many higher level wikitext features, such as procedures, custom widgets and macros.

Example

Here is a complete example showing the important features of the **$transclude** widget:

```
\procedure myproc(name,age)
My name is <<name>> and my age is <<age>>.
\end

<$transclude $variable="myproc" name="James" age="19"/>
```

- `\procedure` defines a variable as a procedure with two parameters, **name** and **age**
- The content of the procedure refers to the parameters as variables
- The **$transclude** widget specifies the variable to transclude, and values for the parameters.

Legacy vs. Modern Mode

The `$transclude` widget can be used in two modes:

- ⓘ **New in v5.3.0** **Modern mode** offers the full capabilities of the `$transclude` widget, and incorporates the functionality of the `$macrocall` widget. It is indicated by the presence of at least one attribute starting with a dollar sign `$`
- **Legacy mode** offers a more limited set of capabilities. It is indicated by the absence of any attributes starting with a dollar sign `$`

Modern mode is recommended for use in new applications.

Attributes

Attribute		Description
(modern)	(legacy)	
$variable	-	Name of the variable to transclude
$tiddler	tiddler	The title of the tiddler to transclude (defaults to the current tiddler)
$field	field	The field name of the current tiddler (defaults to "text"; if present takes precedence over the index attribute)
$index	index	The index of a property in a DataTiddler
$subtiddler	subtiddler	Optional *SubTiddler* title when the target tiddler is a plugin (see below)
$mode	mode	Override the default parsing mode for the transcluded text to "block" or "inline"
$type	–	Optional ContentType used when transcluding variables, indexes or fields other than the **text** field

$output	-	ContentType for the output rendering (defaults to `text/html`, can also be `text/plain` or `text/raw`)
$recursionMarker	recursionMarker	Set to **no** to prevent creation of *Legacy Transclusion Recursion Marker* (defaults to **yes**)
$fillignore	-	Set to **yes** to make this transclusion invisible to the **$depth** attribute of the **$slot** widget (defaults to **no**)
{attributes not starting with $}	–	Any other attributes that do not start with a dollar are used as parameters to the transclusion
{other attributes starting with $}	–	Other attributes starting with a single dollar sign are reserved for future use
{attributes starting with $ $}	–	Attributes starting with two dollar signs are used as parameters to the transclusion, but with the name changed to use a single dollar sign

Basic Operation

The basic operation of the **$transclude** widget is as follows:

`<$transclude/>`	Transcludes the text field of the current tiddler
`<$transclude $variable="alpha"/>`	Transcludes the variable "alpha" (note that procedures, custom widgets and macros are all special types of variable)
`<$transclude $tiddler="foo"/>`	Transcludes the text field of the tiddler "foo"
`<$transclude $field="bar"/>`	Transcludes the field "bar" of the current tiddler
`<$transclude $index="beta"/>`	Transcludes the index "beta" of the current tiddler
`<$transclude $tiddler="foo"`	Transcludes the index "beta" of the tiddler "foo"

```
$index="beta"/>
```

Transclusion Parameters

Named string parameters can be passed to the **`$transclude`** widget. They are made available as variables within the transcluded text. Parameters are only supported in modern mode.

When invoking a transclusion, parameters are specified as additional attributes that do not start with a dollar sign `$`:

```
<$transclude $tiddler="MyTiddler" firstParameter="One" secondParameter="Two"/>
```

To pass parameters whose names start with a dollar sign `$`, prefix them with an extra `$`. For example, to pass a parameter called `$tiddler`:

```
<$transclude $tiddler="MyTiddler" $$tiddler="One"/>
```

There are several different ways to declare parameters within a transclusion:

- the **`$parameters`** widget
- the Pragma: \parameters
- the Pragma: \procedure for declaring procedure
- the Pragma: \widget for declaring custom widgets
- the Pragma: \define for declaring macros

An example of declaring parameters with the **`$parameters`** widget:

```
<$parameters firstParameter="default" secondParameter="another default">
    Parameters are available here as the variables <<firstParameter>> and <<secondParameter>>.
</$parameters>
```

The Pragma: \parameters can be used as a shortcut syntax for declaring parameters. For example:

```
\parameters (firstParameter:"default",secondParameter:"another default")
Parameters are available here as the variables <<firstParameter>> and <<secondParameter>>.
```

Transclusion Slots

Transcluded content can define special named locations called slots. At the point of transclusion, blocks of wikitext can be passed to the **`$transclude`** widget to fill those slots.

Slots work very similarly to parameters except that they can contain structured wikitext, and not just plain text. The primary advantage of slots over parameters is that the contents do not need to be wrapped in quotation symbols, making it much simpler to pass complex structures.

For example, here we transclude the tiddler "Example" while using the **`$fill`** widget to pass wikitext blocks to fill the slots called "positive" and "negative":

```
<$transclude $tiddler="Example">
  <$fill $name="positive">
    <h1>This is positive</h1>
  </$fill>
  <$fill $name="negative">
```

```
    <h3>This is negative</h3>
  </$fill>
</$transclude>
```

The tiddler "Example" uses the **$slot** widget to specify the slots to be filled:

```
<ol>
  <li><$slot $name="positive"/></li>
  <li><$slot $name="negative"/></li>
</ol>
```

The output will be equivalent to:

```
<ol>
  <li>
    <h1>This is positive</h1>
  </li>
  <li>
    <h3>This is negative</h3>
  </li>
</ol>
```

See **$slot** for more examples.

Missing Transclusion Targets

The TranscludeWidget uses the special slot `ts-missing` to specify the content to be rendered if the transclusion target is not defined (i.e. a missing tiddler or a missing field).

For example:

```
<$transclude $tiddler="MissingTiddler">
<$fill $name="ts-missing">
This content is displayed if `MissingTiddler` is missing.
</$fill>
<$fill $name="other">
This content is passed to the transclusion as the slot value `other`
</$fill>
</$transclude>
```

If no slots values are specified within the **`$transclude`** widget then the entire content of the widget is used as the missing content.

For example:

```
<$transclude $tiddler="MissingTiddler">
This content is displayed if `MissingTiddler` is missing.
</$transclude>
```

Parsing modes

TiddlyWiki parses text in two modes:

- inline mode recognises character formatting such as emphasis, links
- block mode recognises all the **inline** formatting, and adds block formatting such as tables, headings and lists

Usually, the mode is determined by whether the transclude widget itself has been parsed in block or inline mode. This can be overridden with the **mode** attribute.

For example, consider tiddler "A" with this content:

```
# Item one
#<$transclude tiddler="B"/>
# Item two
```

And a tiddler "B" with this content:

```
# Item one - a
# Item one - b
```

The result will be something like this:

1. Item one
2. # Item one - a # Item one - b
3. Item two

This can be fixed by amending tiddler "A":

```
# Item one
#<$transclude tiddler="B" mode="block"/>
```

```
# Item two
```

See also these other examples.

SubTiddler Access

The transclude widget allows access to the individual tiddlers stored within a plugin.

The following example will transclude the core version of the tiddler $:/DefaultTiddlers even if it has been overridden:

copy to clipboard

```
<$transclude tiddler="$:/core" subtiddler="$:/DefaultTiddlers"/>
```

That renders as:

GettingStarted

Transclusion

30th November 2014 at 8:54pm

Concepts

Transclusion is the process of referencing one tiddler "A" from another tiddler "B" such that the content of "A" appears to be a part of "B".

Copying and pasting content creates multiple copies of the same content in several different places. With transclusion, there can be a single

copy and a special instruction in "B" which indicates the point at which content should be inserted from tiddler "A".

Note that if the content of "A" is modified then the modification automatically appears in "B". This makes it easier to maintain repetitive content, by allowing every piece to be written in a single place, but viewed from many.

The concept of transclusion plays an important role in the Philosophy of Tiddlers because it is the primary way in which small items of content are combined.

To learn more:

- Transclusion in WikiText
- Transclusion Basic Usage
- Transclusion with Templates
- Transclusion and Substitution
- TextReference
- TemplateTiddlers
- TranscludeWidget

Transclusion and Substitution

30th October 2023 at 1:42pm

WikiText

The power of WikiText comes from the ability to use the content of one tiddler inside another one. This ability takes several different forms that can easily be confused.

The main distinction is between a transclusion and a textual substitution:

- A transclusion is replaced dynamically with the value of either:
  - a tiddler field
  - a variable
- Textual substitutions are performed on the text of macro definitions before they are used

Tiddler Field Transclusion

Transclusion in WikiText describes the basics of transclusion. For example:

```
{{MyTiddler}}
```

As described in HTML in WikiText, you can also transclude tiddler field values as attributes of HTML elements and widgets. For example:

```
<$text text={{MyTiddler}}/>
```

As described in Introduction to filter notation, you can also transclude tiddler field values using the filter syntax. For example:

```
{{{ [tag{TiddlerContainingMyTag}] }}}
```

Variable/Macro Transclusion

Variables that were defined with parameter or variable substitution are referred to as "macros". The value of a variable/macro can be

transcluded with the syntax:

```
<<myMacro param:"Value of parameter">>
```

As described in HTML in WikiText, you can also transclude a variable as the value of an attribute of HTML elements and widgets. For example:

```
<$text text=<<myMacro>>/>
```

As described in Introduction to filter notation, you can also transclude a variable as the value of a filter parameter using the filter syntax. For example:

```
{{{ [tag<myMacro>] }}}
```

Textual Substitution

Textual substitution occurs when the value of a macro/variable is used. It is described in Substituted Attribute Values and substitute Operator

The key difference between substitution and transclusion is that substitution occurs before WikiText parsing. This means that you can use substitution to build WikiText constructions. Transclusions are processed independently, and cannot be combined with adjacent text to define WikiText constructions.

Transclusion Basic Usage

30th November 2014 at 8:48pm

Transclusion in WikiText

Simple Transclusion

To include some content from *TiddlerA* into *TiddlerB*, edit the latter to include the following text:

```
This is the content of TiddlerA: {{TiddlerA}}
```

The result is that the content of the **text** field (i.e. the main content) of *TiddlerA* is rendered within *TiddlerB*.

Usage

The notation `{{TiddlerA}}` is a shortcut for `{{TiddlerA!!text}}`. This is because the default field for transclusion is **text**, but any other field can be used explicitly. For example, you can print the last time *TiddlerA* was modified using:

```
TiddlerA was modified on {{TiddlerA!!modified}}
```

By omitting the tiddler title, the transclusion notation can also be used to display the content of a field from the current tiddler, for example:

```
The current tiddler was modified on {{!!modified}}
```

Recursion Errors

Notice that using `{{!!text}}` or `{{}}` causes an error (*Recursive transclusion error in transclude widget*), because it does not make sense to include the content of the current tiddler into the content of the current tiddler (that is, into itself). Whenever you encouter this error message,

it means that you are trying to include something into itself, directly or indirectly (for example if tiddler A transcludes tiddler B which transcludes tiddler C which, in turn, transcludes tiddler A).

Learning More

In TiddlyWiki, transclusions are not limited to including raw content like the above. To learn about more advanced uses of transclusion, see Transclusion with Templates.

See also:

- Transclusion
- Transclusion with Templates
- Transclusion in WikiText
- TextReference
- TiddlerFields.

Transclusion in WikiText

2nd January 2024 at 8:04am

WikiText

Introduction

You can incorporate the content of one tiddler within another using the Transclusion notation:

- `{{MyTiddler}}` transcludes a single tiddler
- `{{MyTiddler||TemplateTitle}}` displays the tiddler through a specified TemplateTiddler
- `{{||TemplateTitle}}` displays the specified template tiddler without altering the current tiddler
- `{{MyTiddler|Parameter}}` transcludes a single tiddler with a single parameter
- `{{MyTiddler||TemplateTitle|Parameter|SecondParameter}}` transcludes a single tiddler through a specified TemplateTiddler with two parameters

Transcluding Text References

You can also use a TextReference instead of a tiddler title:

- `{{MyTiddler!!field}}` transcludes a specified field of a tiddler
- `{{!!field}}` transcludes a specified field of the current tiddler
- `{{MyTiddler##index}}` transcludes a specified indexed property of a DataTiddler
- `{{##index}}` transcludes a specified indexed property of the current DataTiddler

Filtered Transclusion

A similar syntax can be used to generate or transclude a list of tiddlers matching a specified filter:

```
{{{ [tag[mechanism]] }}}
{{{ [tag[mechanism]]||TemplateTitle }}}
```

In contrast to Filtered Attribute Values, all items matching the filter are transcluded and not only the first.

Generated Widgets

The WikiText transclusion syntax generates a TiddlerWidget wrapped around a TranscludeWidget. For example, `{{MyTiddler||TemplateTitle}}` generates the following pair of widgets:

```
<$tiddler tiddler="MyTiddler">
    <$transclude $tiddler="TemplateTitle"/>
</$tiddler>
```

The filtered transclusion syntax `{{{ [tag[mechanism]] }}}` generates the following widgets

```
<$list filter="[tag[mechanism]]">
    <$link />
</$list>
```

or, when used with a template, `{{{ [tag[mechanism]]||TemplateTitle }}}` expands to

```
<$list filter="[tag[mechanism]]">
    <$transclude tiddler="TemplateTitle" />
</$list>
```



Install the *Internals* plugin to enable the display of the generated widget tree in the preview pane of the editor

---

See also:

- Transclusion Basic Usage
- Transclusion with Templates
- TemplateTiddlers
- TranscludeWidget
- Transclusion and Substitution
- WikiText parser mode: transclusion examples

transclusion Variable

21st February 2015 at 11:56pm

Core Variables Variables

The **transclusion** variable is set by the **`$transclude`** widget to a string that identifies the position of that widget within the widget tree.

TiddlyWiki's core uses it to detect recursive transclusion. It is also used to implement the **`qualify`** macro.

The string has the following syntax:

TypeError: setting getter-only property "parentNode"

a. the title of the current tiddler
b. the title of the tiddler being transcluded
c. the name of the field being transcluded
d. the name of the property name or index being transcluded
e. the name of the subtiddler being transcluded from a plugin

Many of the five items are often blank.

In the sidebar, the value of **transclusion** is:

> `{|$:/core/ui/PageTemplate/sidebar|||}`

When the tiddler `HelloThere` is displayed in the story river, **transclusion** is set to:

> `{HelloThere|HelloThere|||}`

Examples

transclusion Variable (Examples)

28th February 2015 at 11:34am

transclusion Variable Variable Examples

This example shows how to distinguish between the sidebar and other environments:

copy to clipboard

```
We are
<$list
filter="[<transclusion>prefix[{|$:/core/ui/PageTemplate/sidebar|||}]]"
emptyMessage="in the story river.">
in the sidebar.
</$list>
```

Try it

In the sidebar, this would show `We are in the sidebar` instead.

Transclusion with Templates

20th February 2015 at 5:14pm

Transclusion in WikiText

Introduction

In Transclusion Basic Usage we have seen how to include the content of a tiddler A into a tiddler B. Now suppose that tiddler A contains:

```
@@background-color:yellow;
Hello, my title is {{!!title}}
@@
```

This makes tiddler A display its title with a yellow background (see Styles and Classes in WikiText to learn about CSS style). Imagine that you want to display the title in the same way in tiddler B. But you don't want to copy/paste the style instructions, because you might want to change the background colour later and you want to keep it consistent among tiddlers. This looks like a typical case of transclusion, so let's try to transclude tiddler A in tiddler B the usual way with `{{A}}`. You should see the following content in tiddler B:

> Hello, my title is A

The style is applied as expected, but the title is wrong: we want `{{!!title}}` to refer to the target tiddler B, and not the source tiddler A.

The solution is to use a *template*. In this case, the source tiddler A is called the TemplateTiddler, and it is *applied* to tiddler B using the notation `{{||A}}`. The difference is that any TextReference which does not refer explicitly to a specific tiddler is applied to the current tiddler, that is, the target tiddler. As a result, tiddler B now looks as expected:

> Hello, my title is B

Usage

Transcluding via a template is like applying a mask: assuming that the source tiddler contains generic references (like eye holes in a mask), these will be replaced with the target tiddlers values (like the eyes of the person who wears the mask).

A template can be applied to any tiddler, not necessarily the current one. This is achieved using the full notation `{{<target>||<template>}}`. The default `<target>` is the current tiddler (this is what we used in the above example).

Examples

A predefined template to render tags nicely

You can apply the system template `$:/core/ui/TagTemplate` to a tag in order to see it as a tag pill with a drop-down menu:

```
{{Transclusion||$:/core/ui/TagTemplate}}
```

is rendered as Transclusion

See also:

- Transclusion in WikiText
- Transclusion Basic Usage
- TextReference
- TemplateTiddlers
- TranscludeWidget
- Current Tiddler

Translate TiddlyWiki into your language

1st July 2018 at 8:57pm

Community

There is a special edition of TiddlyWiki that simplifies creating and maintaining translations:

- https://tiddlywiki.com/editions/translators/ to translate current version of TiddlyWiki
- https://tiddlywiki.com/prerelease/editions/translators/ to translate latest prerelease version of TiddlyWiki

Note that no knowledge of Node.js or GitHub is required.

You can translate TiddlyWiki on Node.js, type `tiddlywiki editions/translators --listen` and visit http://127.0.0.1:8080/ in your browser.

See https://tiddlywiki.com/dev for technical details of creating and maintaining translations.

TranslationMechanism

11th September 2014 at 8:38pm

Mechanisms

The translation mechanism of TiddlyWiki manages and switches between language plugins that provide translations of the TiddlyWiki user interface. The developer site at https://tiddlywiki.com/dev/ explains how translators can create and submit translations for TiddlyWiki.

The title of the current language plugin is read from the tiddler $:/language. If the selected plugin changes then any displayed translatable text automatically changes.

Translation plugins are bundles of tiddlers that each contain an independent translatable string. The strings are transcluded as needed.

Translatable strings are generally in the namespace `$:/language/`, for example:

- **$:/language/EditTemplate/Shadow/OverriddenWarning**
- **$:/language/Docs/ModuleTypes/isfilteroperator**
- **$:/language/EditTemplate/Fields/Add/Value/Placeholder**

translink Macro

5th May 2023 at 11:03am

Core Macros Macros

The **translink** macro returns a frame with the title and transcluded text of a chosen tiddler. The title links to the transcluded tiddler.

If the chosen tiddler is missing, an appropriate message will be shown instead of the transcluded text.

This is the default macro used when excising text and replacing it with a macro.

Parameters

title

The title of the tiddler to be transcluded

mode

The mode of the transclude widget used inside the macro, defaults to `block`

Examples

translink Macro (Examples)

5th May 2023 at 11:29am

Macro Examples translink Macro

copy to clipboard

```
<<translink "Philosophy of Tiddlers">>
```

Try it

copy to clipboard

```
<<translink "Philosophy of Tiddlers" inline>>
```

Try it

copy to clipboard

```
<<translink Foo>>
```

Try it

Trashbin by Mohammad

17th November 2020 at 4:56pm

Community Plugins

The concept behind Trashbin plugin is to have a simple mechanism to move deleted tiddlers to Trashbin and be able to restore them later if required.

https://kookma.github.io/TW-Trashbin/

The trash (also known as the Recycle Bin in Microsoft Windows) is a temporary storage for tiddlers that have been deleted in a Tiddlywiki by the user, but not yet permanently erased.

Typically, a trash bin is presented as a special storage, allowing the user to browse deleted (removed) tiddlers, undelete those that were deleted

by mistake, or delete them permanently (either one by one, or by the "Empty Trash" function).

tree Macro

28th June 2017 at 6:46pm

Core Macros Macros

The **tree** macro renders an expandable tree view based on a prefix and separator within tiddler titles.

Parameters

prefix

The prefix from which to generate the tree, defaults to `$:/`

separator

The separator between parts of the tiddler titles, defaults to `/`

Examples

tree Macro (Examples)

28th June 2017 at 6:48pm

Macro Examples tree Macro

copy to clipboard

```
<<tree prefix:"$:/">>
```

Try it

copy to clipboard

```
<<tree prefix:"tree-macro-example-" separator:"-">>
```

Try it

tree-macro-example-car

See tree Macro (Examples)

tree-macro-example-car-boot

See tree Macro (Examples)

tree-macro-example-car-boot-handle

See tree Macro (Examples)

tree-macro-example-car-boot-lock

See tree Macro (Examples)

tree-macro-example-car-roof

See tree Macro (Examples)

tree-macro-example-car-roof-aerial

See tree Macro (Examples)

tree-macro-example-car-roof-rails

See tree Macro (Examples)

tree-macro-example-house

See tree Macro (Examples)

tree-macro-example-house-attic

See tree Macro (Examples)

tree-macro-example-house-attic-roof

See tree Macro (Examples)

tree-macro-example-house-attic-window

See tree Macro (Examples)

tree-macro-example-house-garden

See tree Macro (Examples)

tree-macro-example-house-garden-lawn

See tree Macro (Examples)

tree-macro-example-house-garden-shed

See tree Macro (Examples)

tree-macro-example-house-kitchen

See tree Macro (Examples)

tree-macro-example-house-kitchen-sink

See tree Macro (Examples)

tree-macro-example-house-kitchen-table

See tree Macro (Examples)

tree-macro-example-house-kitchen-window

See tree Macro (Examples)

TriggeringWidgets

13th November 2021 at 5:38pm

Widgets

Triggering widgets are a type of widget which can trigger ActionWidgets. Typically these widgets (such as the ButtonWidget) will trigger actions based on interaction from the user.

The following triggering widgets are provided:

- button
- checkbox
- draggable
- droppable
- dropzone
- edit
- EventCatcherWidget
- keyboard
- linkcatcher

- MessageCatcherWidget
- radio
- range
- select

See ActionWidgets for more information and examples of how TriggeringWidgets and ActionWidgets work together.

trim Operator

30th November 2020 at 3:11pm

Filter Operators String Operators

purpose	returns each item in the list with whitespace, or a given character string, trimmed from the start and/or end
input	a selection of titles
suffix	**T** = ⓘ **Introduced in v5.1.23** `prefix` to trim from the start only, `suffix` to trim from the end only. If omitted (default), trim from both start and end
parameter	**S** = ⓘ **Introduced in v5.1.23** a string of characters
output	the input titles with **S**, or whitespace if **S** is not specified, trimmed from the start and/or end

Learn more about how to use Filters

ⓘ **Introduced in v5.1.20**

ⓘ **Introduced in v5.1.23** The **trim** filter allows a parameter specifying a string to trim.

Examples

trim Operator (Examples)

13th June 2019 at 5:39pm

Operator Examples trim Operator

```
[[ a b ]trim[]addprefix[-]addsuffix[-]]
```

Try it

```
[[ abc ]] [[def ]] [[ ghi]] +[trim[]addprefix[-]addsuffix[-]]
```

Try it

```
[[ abc ]] [[def ]] [[ ghi]] +[trim:prefix[]addprefix[-]addsuffix[-]]
```

Try it

```
[[ abc ]] [[def ]] [[ ghi]] +[trim:suffix[]addprefix[-]addsuffix[-]]
```

Try it

```
abacus baobab +[trim[ab]]
```

Try it

```
abacus baobab +[trim:prefix[ab]]
```

Try it

```
abacus baobab +[trim:suffix[ab]]
```

Try it

TriTarget.org by Devin Weaver

24th April 2016 at 1:21pm

Examples

A personal website built with TiddlyWiki.

https://tritarget.org

> *TriTarget*.org is a name I came up with in high school. I was big into computer programming and wanted a company name that was

> kinda catchy. When I made a few programs in BASIC I would brand them with *TriTarget* as a way to show off my company (Even though I didn't have one).

trunc Operator

13th June 2019 at 10:48am

Filter Operators Mathematics Operators Unary Mathematics Operators

purpose	truncates a list of numbers to their integer part, removing any fractional part
input	a selection of titles
output	truncates each of the input numbers to their integer part, removing any fractional part

Learn more about how to use Filters

Introduced in v5.1.20 See Mathematics Operators for an overview.



> The trunc operator rounds towards zero. To round away from zero, use untrunc.

Examples

trunc Operator (Examples)

13th June 2019 at 10:48am

Operator Examples trunc Operator

```
[[1.6]trunc[]]
```

Try it

```
[[-1.6]trunc[]]
```

Try it

```
=-1.2 =-2.4 =3.6 =4.8 =5.1 +[trunc[]]
```

Try it

Tuesday

16th November 2021 at 11:11pm

Operator Examples

This example tiddler is used to illustrate some of the Filter Operators.

Tutorials

2nd June 2016 at 7:27pm

Community

Here are some webpages with tips and tutorials related to TiddlyWiki.

GitHub Saver Tutorial by Mohammad

Posted 7 months ago

GitHub Saver is a step by step tutorial that shows how to integrate Tiddlywiki 5 and GitHub Pages to create websites hosted on https://github.com/.

https://kookma.github.io/TW5-GitHub-Saver/

This instruction is based on Tiddlywiki single html file model, while it can use subfolder for extra materials like images, audios, videos, pdfs,... in separate folders.

Other tutorials

Tiddlywiki, Travis-CI and GitHub Pages
https://kookma.github.io/Tiddlywiki-Travis-CI/
This wiki shows how to set up websites hosted on GitHub Pages using Travis-CI and Tiddlywiki 5 on Node.js.

Tiddlywiki and GitHub Pages
https://kookma.github.io/Tiddlywiki-and-GitHub-Pages/
This instruction is based on local edit, save and push to GitHub. It does NOT use the new GitHub Saver mechanism (requires TW 5.1.20+) which lets edit and save directly from Tiddlywiki!

"TW5 Magick" by Stephan Hradek

Posted 2 years ago

Stephan Hradek's growing catalogue of tips, tutorials, and other resources. Stephan has also developed several plugins and macros that you can find at http://tiddlystuff.tiddlyspot.com/.

http://tw5magick.tiddlyspot.com/

> Some things in TiddlyWiki seem like Magic. As my preferred nick is Skeeve, I felt that TW5 Magick is a proper name for this collection of some of the "magic" tricks one can do with TiddlyWiki.

TW-Scripts by Mohammad

Posted 3 years ago

TW-Scripts is one of the most comprehensive collections of solutions for Tiddlywiki 5.

https://kookma.github.io/TW-Scripts/

TW-Scripts includes:

- Collected solutions to questions in Tiddlywiki Google groups
- Learn through examples
- Templates, stylesheets
- Wikitext, macros, and snippets
- Tips and tricks
- Search tools

TiddlyWiki Video Tutorials by Francis Meetze

Posted 3 years ago

An extensive tutorial for getting started with TiddlyWiki. Recommended

https://www.youtube.com/playlist?list=PLzZCajspPU_UjFn0uy-J9URz0LP4zhxRK

> This tutorial is aimed at giving you a basic foundation so that you can start using TiddlyWiki right away. Here we take a look at the

> technology behind Tiddlywiki, how to install it and create your first Tiddler.

RegExp in Tiddlywiki by Mohammad

Posted 3 years ago

RegExp in Tiddlywiki contains practical use cases of regular expression in Tiddlywiki.

http://tw-regexp.tiddlyspot.com/

The regular expression is a way to describe complex search patterns using sequences of characters .

RegExp in Tiddlywiki has four main parts

- Using regular expression to match a pattern in tiddler title
- Using regular expression to match a pattern in tiddler fields excluding tiddler text (body)
- Using regular expression to match a pattern in tiddler body (text field)
- Using regular expression for validation like username, password, etc.

"TB5 - a pocket full of tips" by Tobias Beer

Posted 3 years ago

An enormous collection of tips beautifully curated by Tobias Beer. It is actually a collection of sites, with sub-sites dedicated to topics such as filtering (http://tobibeer.github.io/tw/filters/).

http://tobibeer.github.io/tb5/

> A collection of references and sandbox for testing concepts around TiddlyWiki 5...

"Heeg.ru" by sini-Kit

Posted 3 years ago

A guide in Russian to creating an online shop with TiddlyWiki. No serverside code is needed, instead orders are handled by Disqus.

http://heeg.ru/

"Filter Examples" by Tobias Beer

Posted 3 years ago

This wiki gives examples for various filters and their use in the list widget

http://tobibeer.github.io/tw/filters/#Filter%20Examples

> This wiki gives examples for various filters and their use in the list widget. A good starting point to understand what filters yield which results also is test-filters.js.

"Microblogging con Tiddlywiki" by Juan

Posted 9 years ago

A Spanish guide to using TiddlyWiki on Android.

http://3palmeras.wordpress.com/2014/11/05/microblogging-con-tiddlywiki/

> Tiddlywiki software is prodigious based on javascript which is capable of holding a notepad that can be edited with Internet Explorer or function as a full blog, contained in a single file.

Translation by Google Translate

"Un wiki sur votre BiblioBox avec TiddlyWiki" from A Penguin in Redmond

Posted 9 years ago

A guide (in French) to using TiddlyWiki on the BiblioBox, a variation of the PirateBox mobile communication and file sharing system.

http://bibliobox.net/blog/post/2014/07/16/un-wiki-sur-votre-bibliobox-avec-tiddlywiki

> I've known TiddlyWiki since its beginning in 2004. For personal or professional needs, I used it regularly for notepad staff in 2007 and 2012. But I admit I ignored the tool during the past two years. And then I said yesterday: but why not put TiddlyWiki on a *BiblioBox*? Because it is a single HTML file with javascript. In addition I knew the existence of a server version for saving online amendments.

Hand optimised from a translation by Google Translate

"TW5 Tribal Knowledge" from Scott Kingery

Posted 9 years ago

Tips and guides for using TiddlyWiki.

http://techlifeweb.com/tw5tribalknowledge.html

> I never used Node.js until a couple weeks ago when I wanted to try a TW5 installation on Node and expand my knowledge a bit. I figured someone else may be in the same boat so I wrote it all down

"Creating a baby journal with TiddlyWiki 5" from A Penguin in Redmond

Posted 10 years ago

A guide to creating a baby journal with TiddlyWiki.

http://a-penguin-in-redmond.blogspot.co.uk/2014/02/creating-baby-journal-with-tiddlywiki-5.html

> Keeping a journal about your baby is fun. It allows you to keep track of those events in your baby's life and development that are most important to you, and to tell them in your own words. [...]
>
> There are plenty of online services that can help to keep such a diary, but you need to have an internet connection to access them, and you have to trust them to keep your data safe.

"デザイン刷新。次の25年を目指すTiddlyWiki" by Moongift

Posted 10 years ago

An introduction to TiddlyWiki in Japanese.

http://www.moongift.jp/2013/11/tiddlywiki5-%E3%83%87%E3%82%B6%E3%82%A4%E3%83%B3%E5%88%B7%E6%96%B0%E3%80%82%E6%AC%A1%E3%81%AE25%E5%B9%B4%E3%82%92%E7%9B%AE%E6%8C%87%E3%81%99tiddlywiki-2/

"Shining Ark Using TiddlyWiki" by Helu

Posted 10 years ago

A gamers guide to using TiddlyWiki.

http://helugame.wordpress.com/2013/09/07/shining-ark-7-using-tiddlywiki/

> I've been using TiddlyWiki for this game mainly to record the monsters' locations and the items that they drop. This is so that when I need a certain item to upgrade/create weapons, I can quickly look it up; which monsters drop it and which quests have those monsters.
>
> As I play the game, as I encounter new information, I just input them into my Shining Ark TiddlyWiki. It's like I'm building a wiki from scratch. The reason why I don't just use online wikis is that 1) it's too easy to get spoiled and 2) sometimes online wikis provide too much

> information (who cares about the non-gameplay related descriptions that can be found in the game, etc).

tv-adjust-heading-level Variable

9th June 2018 at 1:24pm

Configuration Variables Core Variables Variables

The **tv-adjust-heading-level** variable allows the level of headings to be adjusted. It is interpreted as a positive or negative number (e.g. "2" or "-3") that is added to the heading level for display.

Note that if the resulting heading level is less than 1 then level 1 is used instead. This means that specfiying a large negative adjustment factor will reset all headings to display as level 1. Similarly, heading levels larger than 6 are clamped to the maximum HTML heading level of 6.

Examples

tv-adjust-heading-level Variable (Examples)

9th June 2018 at 1:24pm

tv-adjust-heading-level Variable Variable Examples

Tiddlers

Sample Headings 1,2,3

```
! This is a Level 1 Heading
```

```
This is a paragraph.

<$vars tv-adjust-heading-level="-2">

{{Sample Headings 3,4,5}}

</$vars>

!! This is a Level 2 heading

<$vars tv-adjust-heading-level="-1">

{{Sample Headings 4,5,6}}

</$vars>
```

Sample Headings 3,4,5

```
!!! This is written as a Level 3 heading

!!!! This is written as a Level 4 heading

!!!!! This is written as a Level 5 heading
```

Sample Headings 4,5,6

```
!!!! This is written as a Level 4 heading

!!!!! This is written as a Level 5 heading

!!!!!! This is written as a Level 6 heading
```

Example

copy to clipboard

```
<$transclude tiddler="Sample Headings 1,2,3" mode="block"/>
```

Try it

tv-auto-open-on-import Variable

28th February 2015 at 2:04pm

Configuration Variables Core Variables Variables

The **tv-auto-open-on-import** variable controls whether the `tm-import-tiddlers` message causes the tiddler *$:/Import* (which lists the pending imports) to open in the story river.

By default, the tiddler is opened. But if this variable has the value **no**, it isn't.

An example of setting this variable to **no** can be found in the upgrade plugin within the TiddlyWiki Upgrade Wizard. People can drag their wiki files onto the wizard without triggering the normal import display.

tv-config-static Variable

17th June 2023 at 10:55am

Configuration Variables Core Variables Variables

New in v5.3.0 The **tv-config-static** variable is set to `yes` within static rendering templates, and is unset in other contexts.

It is useful for selectively hiding or showing content depending on whether a rendering is static or interactive.

tv-config-toolbar-class Variable

28th February 2015 at 2:58pm

Configuration Variables Core Variables Variables

The **tv-config-toolbar-class** variable controls the value of the CSS `class` attribute on the HTML element for a toolbar button.

It can be set prior to transcluding such a button.

In most environments, it defaults to **`tc-btn-invisible`**, which removes the button's background colour and border.

Examples

tv-config-toolbar-class Variable (Examples)

28th February 2015 at 2:57pm

tv-config-toolbar-class Variable Variable Examples

copy to clipboard

```
<$set name="tv-config-toolbar-class" value="tc-btn-invisible">
{{$:/core/ui/Buttons/new-tiddler}}
</$set>
```

Try it

copy to clipboard

```
<$set name="tv-config-toolbar-class" value="green-background">
{{$:/core/ui/Buttons/new-tiddler}}
</$set>
```

Try it

The `green-background` CSS class is declared in a stylesheet within this tiddler.

tv-config-toolbar-icons Variable

28th February 2015 at 3:13pm

Configuration Variables Core Variables Variables

The **tv-config-toolbar-icons** variable controls whether toolbar buttons display icons.

It can be set to **yes** or **no** prior to transcluding such a button.

The standard page template sets it to the value found in **$:/config/Toolbar/Icons**, with the result that this becomes the default for the whole page. The user can adjust this default by using a tickbox on the **Settings** tab of the **Control Panel**.

Examples

See also **tv-config-toolbar-text**.

tv-config-toolbar-icons Variable (Examples)

28th February 2015 at 3:14pm

tv-config-toolbar-icons Variable Variable Examples

copy to clipboard

```
<$set name="tv-config-toolbar-icons" value="no">
<$set name="tv-config-toolbar-text" value="yes">
{{$:/core/ui/Buttons/new-tiddler}}
</$set>
</$set>
```

Try it

copy to clipboard

```
<$set name="tv-config-toolbar-icons" value="yes">
{{$:/core/ui/Buttons/new-tiddler}}
</$set>
```

Try it

tv-config-toolbar-text Variable

28th February 2015 at 3:14pm

Configuration Variables Core Variables Variables

The **tv-config-toolbar-text** variable controls whether toolbar buttons display text.

It can be set to **yes** or **no** prior to transcluding such a button.

The standard page template sets it to the value found in **$:/config/Toolbar/Text**, with the result that this becomes the default for the whole page. The user can adjust this default by using a tickbox on the **Settings** tab of the **Control Panel**.

Examples

See also **tv-config-toolbar-icons**.

tv-config-toolbar-text Variable (Examples)

28th February 2015 at 3:05pm

tv-config-toolbar-text Variable Variable Examples

copy to clipboard

```
<$set name="tv-config-toolbar-text" value="no">
{{$:/core/ui/Buttons/new-tiddler}}
</$set>
```

Try it

copy to clipboard

```
<$set name="tv-config-toolbar-text" value="yes">
{{$:/core/ui/Buttons/new-tiddler}}
</$set>
```

Try it

tv-filter-export-link Variable

28th August 2017 at 11:57am

Configuration Variables Core Variables Variables

Introduced in v5.1.15 The **tv-filter-export-link** variable controls the value of the `href` attribute on the HTML `a` element generated by the **`$link`** widget. If defined, it takes precedence over the tv-wikilink-template Variable.

The variable is treated as a filter that is given the target tiddler title as input. The filter is evaluated and the first result is used as the `href` attribute.

For example:

```
\define tv-filter-export-link() [encodeuricomponent[]encodeuricomponent[]addsuffix[.html]]
```

See also the **`tv-get-export-link`** variable, which dominates over this one.



This variable has no useful effect when TiddlyWiki is running in a browser, as the `href` attribute is ignored there – links between tiddlers are performed by JavaScript instead. The variable comes into play when one is using the Node.js configuration to generate a static version of a wiki.

tv-get-export-image-link Variable

28th February 2015 at 2:09pm

Configuration Variables Core Variables Variables

The **tv-get-export-image-link** variable controls the value of the `src` attribute on the HTML `img` element generated by the **`$image`** widget when the value of its `source` attribute is not the title of a tiddler.

The variable should be a macro with the following parameter:

src

The value of the `source` attribute – equivalent to the image name specified in the shorthand syntax `[img[source]]`

The ability to override image URIs in this way can be useful when one is using the Node.js configuration to export a static version of a wiki.

Examples

tv-get-export-image-link Variable (Examples)

tv-get-export-image-link Variable Variable Examples

This example fetches the TiddlyWiki icon:

```
\define tv-get-export-image-link(src) https://www.tiddlywiki.com/$src$
```

copy to clipboard

```
[img[favicon.ico]]
```

Try it

tv-get-export-link Variable

28th February 2015 at 2:09pm

Configuration Variables Core Variables Variables

The **tv-get-export-link** variable controls the value of the `href` attribute on the HTML `a` element generated by the **`$link`** widget.

The variable should be a macro with the following parameter:

to

The title of the target tiddler of the link, with no escaping

See also **`tv-wikilink-template`**. If both that variable and this one exist, this one dominates.



This variable has no useful effect when TiddlyWiki is running in a browser, as the `href` attribute is ignored there – links between tiddlers are performed by JavaScript instead. The variable comes into play when one is using the Node.js configuration to generate a static version of a wiki.

tv-get-export-path Variable

28th February 2015 at 2:17pm

Configuration Variables Core Variables Variables

The **tv-get-export-path** variable specifies the full pathname to which the rendertiddlers Node.js command writes each tiddler.

The variable should be implemented as a JavaScript macro with the following parameter:

title
: The title of the tiddler

If no such macro exists, the tiddlers are written to files whose names are percent-encoded, in the command's output folder.

tv-history-list Variable

29th June 2017 at 9:46am

Core Variables Variables

The **tv-history-list** variable is assigned by the NavigatorWidget to contain the title of the tiddler containing the current history list.

Compare `tv-story-list`.

tv-show-missing-links Variable

Core Variables Variables

The **tv-show-missing-links** variable defines if **missing links** within the **scope** of the variable are rendered as links or not.

Globally, the variable is set within the PageTemplate and can be changed in the **$:/ControlPanel** under Settings - Show missing links to tiddlers

If **tv-show-missing-links** is set to **no**, missing links are rendered as plain text

If **tv-show-missing-links** is set to **yes**, missing links are rendered as wiki-links

tv-story-list Variable

29th June 2017 at 9:45am

Core Variables Variables

The **tv-story-list** variable is assigned by the NavigatorWidget to contain the title of the tiddler containing the current story list.

Compare `tv-history-list`.

tv-tiddler-preview Variable

20th June 2015 at 10:27am

Core Variables Variables

The **tv-tiddler-preview** variable indicates whether content is being rendered in a tiddler preview panel.

The value is **yes** within the preview panel, and undefined elsewhere.

tv-wikilink-template Variable

28th August 2017 at 11:56am

Configuration Variables Core Variables Variables

The **tv-wikilink-template** variable controls the value of the `href` attribute on the HTML `a` element generated by the **`$link`** widget. The **`tv-filter-export-link`**, if defined, it takes precedence over the **`tv-wikilink-template`** variable.

The variable is treated as if it was a macro with the following parameters:

uri_encoded
: The title of the target tiddler of the link, but percent-encoded

uri_doubleencoded
: The value of the **`uri_encoded`** parameter but percent-encoded again, i.e. with its `%` characters further converted to `%25`

```
\define tv-wikilink-template() ../tiddlers/$uri_encoded$.html
```

The variable defaults to `#$uri_encoded$`.

See also the **`tv-get-export-link`** variable, which dominates over this one.



This variable has no useful effect when TiddlyWiki is running in a browser, as the `href` attribute is ignored there – links between tiddlers are performed by JavaScript instead. The variable comes into play when one is using the Node.js configuration to generate a static version of a wiki.

tv-wikilink-tooltip Variable

28th February 2015 at 2:04pm

Configuration Variables Core Variables Variables

The **tv-wikilink-tooltip** variable specifies the default value for the **`tooltip`** attribute of the **`$link`** widget.

It is often defined as a macro, and its value parsed as inline WikiText.

Examples

tv-wikilink-tooltip Variable (Examples)

28th February 2015 at 1:06pm

tv-wikilink-tooltip Variable Variable Examples

This example shows a way of giving links a tooltip derived from the target tiddler's **`caption`** field:

```
\define tv-wikilink-tooltip()
<$transclude field="tooltip">(<$transclude field="caption"/>)</$transclude>
\end
```

copy to clipboard

```
Here is a link to [[backlinks Operator]] with a custom tooltip.
```

Try it

Here is a standard link for comparison.

tv-wikilinks Variable

28th February 2015 at 2:04pm

Configuration Variables Core Variables Variables

The **tv-wikilinks** variable controls the behaviour of the **$link** widget.

The widget normally produces a link to a tiddler. But if this variable has the value **no**, the widget suppresses the link and simply displays the text that would otherwise have served as the link.

You can suppress links for a whole tiddler by placing the following line at the start of the tiddler's text:

```
\define tv-wikilinks() no
```

This variable has no effect on external links, as those do not involve the **$link** widget.

Examples

tv-wikilinks Variable (Examples)

28th February 2015 at 11:44am

tv-wikilinks Variable Variable Examples

copy to clipboard

```
HelloThere,
[[HelloThere]],
<$link to="HelloThere">is //this// a link?</$link>
```

Try it

copy to clipboard

```
<$set name="tv-wikilinks" value="no">
HelloThere,
[[HelloThere]],
<$link to="HelloThere">is //this// a link?</$link>
</$set>
```

Try it

TW-Scripts by Mohammad

6th January 2021 at 4:10pm

Other Resources Tutorials

TW-Scripts is one of the most comprehensive collections of solutions for Tiddlywiki 5.

https://kookma.github.io/TW-Scripts/

TW-Scripts includes:

- Collected solutions to questions in Tiddlywiki Google groups
- Learn through examples
- Templates, stylesheets
- Wikitext, macros, and snippets
- Tips and tricks
- Search tools

TW2Parser Plugin

28th February 2017 at 11:24am

OfficialPlugins Plugin Editions

This experimental plugin adds the ability to display WikiText written for the original Classic version of TiddlyWiki.

See https://tiddlywiki.com/plugins/tiddlywiki/tw2parser

TW5-firebase: TiddlyWiki5 for Google Firebase by Peter Neumark

15th January 2021 at 1:10pm

Chrome Edge Firefox Internet Explorer Linux Mac Opera Other Resources plugins Safari Saving Windows

Google-Firebase hosted version of TiddlyWiki5.

https://github.com/neumark/tw5-firebase

> I've been using TiddlyWiki5 with Google Firebase for over a year now. It's stable enough to use on a daily basis. There's a detailed walkthrough of how to create your own instance with lots of screenshots. The entire process takes less than 30 minutes: https://neumark.github.io/tw5-firebase/
>
> Please consider this version a proof of concept rather than a polished product! My focus was on multi-device or small-team collaboration:
>
> - Basic authorisation (bags have an access policy determining who can read / write them).
> - Multiple wikis can be hosted under a single firebase account.
> - It uses Firebase's built-in social auth to log in users.
> - Tiddlers are written individually with locking, preventing users overwriting each others' updates.

TW5-SingleExecutable by Jed Carty

6th January 2021 at 4:10pm

Other Resources

TiddlyWiki5 Packaged with the multi-user plugin in a single executable file.

https://github.com/OokTech/TW5-SingleExecutable

> Tiddlywiki, node js, multiuser plugin - all packaged into a single file so user can just download the file and run it and not have to install anything else. Versions available for windows, osx, and linux.

TWaddle by Matias Goldman

29th May 2015 at 10:43am

Other Resources

A collection of hints and tips, musings and proposals from long-time TiddlyWiki contributor Mat Goldman.

http://twaddle.tiddlyspot.com/

> This is Mat's, a.k.a <:-) little man-cave in the TiddlyVerse.
>
> Most TW development is, understandably, based on the premise that *"Improve code ➔ Better TW"*. **TWaddle** is also about developing TW but from the perspective that *Bigger community ⇄ Better TW*. Thus, TWaddle looks more to people issues - how to attract them, how to make them stick with TW etc.
>
> I'm not a programmer but I *am* a TW enthusiast so I tiddlefiddle enough to make the occasional discovery of something cool. Given the amount of words I utter, it is also a mere numbers game before I say something that makes sense. TWaddle is intended to capture these eventualities.

TWCommunitySearch

6th January 2021 at 4:10pm

Other Resources

The CommunitySearch wiki aggregates many public wikis from the community. It is updated automatically every day. At the time of writing, more than 5000 tiddlers are indexed, thus giving access to the largest known collection of TW content.

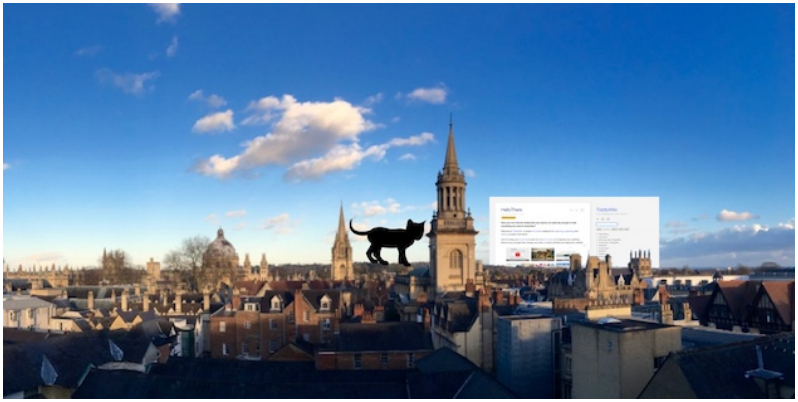
It lets you search for some specific content among a preselected list of wikis. The indexed content is mostly focused on TW usage, so you would typically use this search system to find information about a particular aspect of TiddlyWiki. Say for example that you want to learn how to make a table of content, type "table of content" in the CommunitySearch box (also accessible in the default tiddler GettingStarted). The search results are links to various wikis, and point directly to the specific tiddlers containing your request.

The CommunitySearch wiki also lets you:

- Browse the list of wikis
- Browse the list of author
- Browse the list of plugins
- Access content by tag (see also BookmarkingTags)
- Follow the latest news from the community (and publish your own news if you are a wiki author)

TWEUM Thumbnail.jpg

picture



Twexe: Single File Tiddlywiki5 executable

6th January 2021 at 4:10pm

Edge Firefox Internet Explorer Linux Opera Other Resources Safari Saving Windows

Single File Tiddlywiki5 executable

https://ihm4u.github.io/twexe/

Features

- Automatic backups
- Automatic saving
- No browser add-ons needed!!
- Works on Linux and Windows
- Single file executable, can be moved, copied, etc.
- Wiki is compressed, occupying less space on your disk (sometimes even a third of the space!)
- Converts *any* Tiddlywiki5 file to a single file executable , no need for special tiddlywiki plugins
- Automatically uses the saving tab in the control panel (there you can specify a backup directory if you don't like the default)
- Easily get back the html wiki file from the executable twexe (if you need it for some reason)
- Opens up the possibility to run external executables from your wiki (to draw charts, etc.) - stay tuned!!
- Supports external images (i.e. with _canonical_uri field)

Twitter Plugin

28th March 2017 at 7:39pm

OfficialPlugins

This plugin adds a `<$twitter>` widget that can embed a variety of entities from twitter.com:

- Individual tweets and conversation threads
- Buttons to tweet a hashtag/account, follow/like an account, or share a URL
- Timelines showing tweets from a user, hashtag, list or collection

twproxy by Steve Gattuso

6th January 2021 at 4:10pm

Other Resources

An authentication proxy for your TiddlyWiki.

https://github.com/stevenleeg/twproxy

> One of my favorite aspects of TiddlyWiki has been that it's web-based, making it possible for me to access my wiki from everywhere (especially my phone). That being said, I'm a bit paranoid, so I was left a bit unsatisfied with the tiddlywiki server's HTTP basic auth for protecting my wiki. ~23 commits later, I've created something called twproxy that I'd like to share with you all today, as I'm hoping somebody other than myself will find it useful.
>
> Essentially it is a simple proxy that puts your wiki behind a username, password, and optional 2-factor auth prompt. This gives you added security in addition to the ability to remember your credentials past one browsing session (I was getting sick and tired of typing my username/password in over and over using basic auth).

Typed Blocks in WikiText

20th January 2014 at 6:14pm

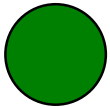
WikiText

WikiText can include blocks of text that are rendered with an explicit ContentType like this:

```
$$$image/svg+xml
<svg xmlns="http://www.w3.org/2000/svg" width="150" height="100">
```

```
  <circle cx="100" cy="50" r="40" stroke="black" stroke-width="2" fill="green" />
</svg>
$$$
```

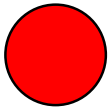
This renders as:



It is also possible to abbreviate the ContentType to a file extension. For example:

```
$$$.svg
<svg xmlns="http://www.w3.org/2000/svg" width="150" height="100">
  <circle cx="100" cy="50" r="40" stroke="black" stroke-width="2" fill="red" />
</svg>
$$$
```

This renders as:



Unknown types render as plain text:

```
$$$text/unknown
Some plain text, which will not be //formatted//.
$$$
```

Which renders as:

```
Some plain text, which will not be //formatted//.
```

A render type can also be specified, causing a particular text rendering to be displayed. For example:

```
$$$text/vnd.tiddlywiki>text/html
This is ''some'' wikitext
$$$

$$$text/vnd.tiddlywiki>text/plain
This is ''some'' wikitext
$$$
```

Renders as:

```
<p>This is <strong>some</strong> wikitext</p>
```

```
This is some wikitext
```

Typography

17th January 2015 at 4:25pm

Use the documentation macros to keep the text maintainable in the face of change.

Be wary of arbitrarily applying raw bold or italic markup in a sentence. If there's a suitable macro, use that instead. If there isn't a suitable macro, write one or request one.

Use simple backticks (`` `...` ``) for fragments of WikiText, but not for the names of things like fields and widgets. These have their own macros.

To keep things clean and simple, quotation marks and apostrophes should be straight `'`, not curly `’`, and the ellipsis should be three separate dots `...` rather than `…`.

Use `"` as the primary quotation mark, reserving `'` for the rare case of a nested quotation.

Avoid using a single hyphen `-` as sentence punctuation. Instead, use a double hyphen – which TiddlyWiki renders as an en-dash – with a space on either side.

Items in lists and tables should only end with a full stop (period in US English) if they are complete sentences. They should have no trailing punctuation otherwise.

It is very rarely necessary to use an exclamation mark in professional documentation.

Uninstalling a plugin

17th June 2022 at 3:29pm

Plugins

If you want to follow the steps *side by side* you can open **this** tiddler in a new window.

Follow these instructions when using TiddlyWiki as a standalone, single file wiki.

1. Create a backup of your current TiddlyWiki HTML file (just in case)
2. Open the ControlPanel and go to the **Plugins** tab
3. Click on the plugin you want to delete to open it as a standalone tiddler
4. Click the **more** button and **delete** the tiddler
5. Save your TiddlyWiki
6. If a yellow warning bar appears at the top of the window, refresh the window so that TiddlyWiki completely removes the plugin
    - **Introduced in v5.1.22** It is no longer necessary to refresh TiddlyWiki when deleting plugins that support dynamic loading. See PluginMechanism for more details
7. The plugin should no longer be active or available

Note

> If you use TiddlyWiki with node.js have a closer look at Uninstalling a plugin with Node.js
> An overview of working with plugins can be found at Plugins

Uninstalling a plugin with Node.js

17th June 2022 at 3:37pm

PluginsCS TiddlyWiki on Node.js

Follow these instructions when using TiddlyWiki with client-server Node.js configuration:

1. Quit the server if it is running
2. Edit the `tiddlywiki.info` file (it is in JSON format) and locate the `plugins` and `themes` section (see below)
3. Remove the entries corresponding to the plugins you wish to remove
    - Take care to retain **commas** to separate items
    - Do **not** terminate the last item in a list with a comma
4. Restart the server

```
{
    "plugins": [
        "tiddlywiki/codemirror"
    ],
    "themes": [
        "tiddlywiki/vanilla",
        "tiddlywiki/snowwhite"
    ]
}
```

Note

> An overview of working with plugins can be found at Plugins

unique Operator

Filter Operators Listops Operators Order Operators

purpose	remove all duplicate items from the current list
input	a list of items
parameter	ignored
output	a list of unique items

Learn more about how to use Filters

Examples

unique Operator (Examples)

17th February 2021 at 1:32am

Operator Examples unique Operator

To remove duplicate entries from a list where deduplication has been prevented using the filter run prefix `=`:

```
=[[E]] =[[A]] =[[B]] =[[C]] =[[C]] =[[D]] =[[C]] +[unique[]]
```

Try it

UnpackPluginCommand

29th October 2014 at 12:43pm

Commands

Extract the payload tiddlers from a plugin, creating them as ordinary tiddlers:

```
--unpackplugin <title>
```

untagged Operator

3rd February 2015 at 8:18pm

Filter Operators Negatable Operators Tag Operators

purpose	discard any input titles that have tags
input	a selection of titles
parameter	none
output	those input tiddlers that have no tags
! **output**	those input tiddlers that have at least one tag

Learn more about how to use Filters

A tiddler is deemed to have no tags if it:

- doesn't exist
- doesn't have a **tags** field
- has an empty **tags** field

Examples

untagged Operator (Examples)

24th January 2015 at 3:27pm

Operator Examples untagged Operator

```
[untagged[]]
```

Try it

```
[all[shadows]untagged[]]
```

Try it

```
[list[HelloThere]!untagged[]]
```

Try it

untrunc Operator

17th June 2019 at 6:33pm

Filter Operators Mathematics Operators Unary Mathematics Operators

purpose	rounds a list of numbers to the next integer with largest absolute value, that is, away from zero
input	a selection of titles
output	rounds each of the input numbers to the next integer with largest absolute value, that is, away from zero

Learn more about how to use Filters

Introduced in v5.1.20 See Mathematics Operators for an overview.



This is the inverse operation of trunc: trunc rounds to zero, but untrunc rounds away from zero.

Examples

untrunc Operator (Examples)

17th June 2019 at 6:29pm

Operator Examples untrunc Operator

```
[[1.6]untrunc[]]
```

Try it

```
[[-1.6]untrunc[]]
```

Try it

```
=-1.2 =-2.4 =3.6 =4.8 =5.1 +[untrunc[]]
```

Try it

unusedtitle Macro

27th April 2021 at 8:40pm

Core Macros Macros

The **unusedtitle** macro returns an unused title. Optionally you can provide a base title to generate the new title.

It uses the same method as the create new tiddler button, a number is appended to the end of the base name.

Parameters

baseName

A string specifying the desired base name, defaulting to `New Tiddler`.
The default setting can be adjusted in the **$:/ControlPanel : Info : Basics - tab.**

separator

ⓘ **Introduced in v5.2.0** An **optional** string specifying the separator between baseName and the unique number. eg: `separator:"-"`.
Defaults to a space: `" "`. If you need an empty separator use the **template**!

template

ⓘ **Introduced in v5.2.0** A **optional** template string can be used to allow you maximum flexibility. If the template string is used, there will always be a counter value.

Template String

`$basename$`
This variable will be replaced by the content of the **baseName** parameter

`$separator$`
This variable will be replaced by the **separator** parameter

`$count$`
This variable will be created automatically and is a counter starting with 0

`$count:4$`
This variable will be created automatically and starts at 0000
`:4` represents the number of digits

Examples

- unusedtitle Macro (Examples)
- unusedtitle Macro (Examples 1)

unusedtitle Macro (Examples 1)

18th November 2021 at 3:55am

Macro Examples

```
<<unusedtitle template:"$count:2$-new">>
```

Create Tiddler

```
<<unusedtitle baseName:"new" template:"$count:2$-$basename$">>
```

New Tiddler

```
<<unusedtitle baseName:"new" separator:"-" template:"$count:2$$separator$$basename$">>
```

Create Tiddler

▸ Show the code

unusedtitle Macro (Examples)

28th February 2021 at 3:12pm

Macro Examples unusedtitle Macro

copy to clipboard

```
<<unusedtitle>>
```

Try it

The following example works best if there is an open tiddler in draft mode, or there is a tiddler named "New Tiddler". So you can see the automatic numbering.

copy to clipboard

```
<<unusedtitle separator:"-">>
```

Try it

copy to clipboard

```
<<unusedtitle baseName:"anotherBase">>
```

Try it

copy to clipboard

```
<<unusedtitle baseName:"About" separator:"-">>
```

Try it

copy to clipboard

```
<<unusedtitle template:"$count:2$-test">>
```

Try it

Working buttons can be found at: unusedtitle Macro (Examples 1). You'll have to examine the code to see, what's going on.

UpgradeMechanism

6th June 2016 at 2:28pm

Mechanisms

1. Open upgrade.html
2. Includes a data tiddler called `$:/UpgradeLibrary` that contains the latest compatible versions of all plugins in the library
3. Drag in old wiki file
4. Place tiddlers into a data tiddler `$:/Import` that is typed as a "pending import"
5. Kick off import processing for each tiddler

1. Give each "upgrader" module a chance to inspect the incoming tiddlers
2. Upgrader modules can trigger actions for each tiddler:
    - Display a warning message
    - Don't import
    - Replace with another tiddler from the upgrade library
    - Disable incompatible plugins

6. Display the newly created pending import tiddler through a new view template segment
    1. Displays the payload tiddlers as a list of titles and checkboxes, with a dropdown showing the full details of the tiddler
    2. Perhaps we also suppress the usual JSON display for data tiddlers behind a reveal widget
7. The user can adjust the selection checkboxes
8. Clicking "Upgrade" unpacks the selected tiddlers from the pending import tiddler
9. The pending import tiddler and the upgrade library tiddler are excluded from the subsequent save operation

Upgrading

17th June 2016 at 12:51pm

Features Working with TiddlyWiki

There are regular releases of TiddlyWiki with bug fixes and improvements. It's a good idea to keep up to date by regularly upgrading to the latest version.

Introduction

The process described here is for upgrading standalone TiddlyWiki files. There is a different procedure for upgrading TiddlyWiki on Node.js.

> When upgrading, please remember the The First Rule of Using TiddlyWiki:
>
> *You are responsible for looking after your own data; take care to make backups, especially when upgrading the TiddlyWiki core*

Online upgrading

This process will work on most desktop browsers. Note that none of your personal data leaves your browser with this process.

1. Locate your TiddlyWiki file in the file system (i.e. using Windows Explorer, the Finder on Mac OS X, or your file manager on Linux)
2. Visit https://tiddlywiki.com/upgrade.html in your browser
3. Drag your old TiddlyWiki HTML file into the browser window
    - If the file is encrypted you will be prompted for the password
4. Review the list of tiddlers that will be upgraded
5. Click **Upgrade**
6. Save changes to save the new version (⊘)

This will download a file called **upgrade.html** to your computer. This file is the upgrade of your old file. You may need to open the location where **upgrade.html** was downloaded, rename **upgrade.html** with the name of the old file you are upgrading, and replace the old file by moving the new file in its place.

Offline upgrading

You can also download https://tiddlywiki.com/upgrade.html locally and perform the same drag-and-drop procedure to upgrade your files.

Problems with Upgrades

Firefox Security Restrictions

The following error occurs when trying to perform the online upgrade procedure using Firefox:

> Error while saving:
>
> Error:NS_ERROR_DOM_BAD_URI: Access to restricted URI denied

The upgrade operation falls foul of a security restriction in Firefox. Until this can be resolved, we suggest either using the offline upgrader, or using Chrome to perform the upgrade:

1. Use Chrome to open https://tiddlywiki.com/upgrade.html, then drag the TiddlyWiki HTML file to be upgraded into the upgrader window, as described above in **Online upgrading**
2. After you've saved your upgraded file, you should be able to open it in Firefox and save using TiddlyFox again

Incompatible Customisations

It is possible for a customisation applied in a previous version to break when upgraded to the latest version. There are two techniques you can use to help track down issues:

- Try repeating the upgrade having selectively unchecked any tiddlers that may be applying customisations to TiddlyWiki
- Use SafeMode to disable all customisations of shadow tiddlers

You can see which shadow tiddlers have been overridden in the **Filter** tab of **advanced search**. Select "Overridden shadow tiddlers" from the

dropdown.

Upgrading TiddlyWiki on Node.js

12th September 2014 at 4:18pm

TiddlyWiki on Node.js

If you've installed TiddlyWiki on Node.js on the usual way, when a new version is released you can upgrade it with this command:

```
npm update -g tiddlywiki
```

On Mac or Linux you'll need to add **sudo** like this:

```
sudo npm update -g tiddlywiki
```

uppercase Operator

19th June 2019 at 1:08pm

Filter Operators

purpose	returns each item in the list as uppercase
input	a selection of titles
output	the input titles with each lowercase letter replaced by the equivalent uppercase letter

Learn more about how to use Filters

ⓘ Introduced in v5.1.20

See also lowercase Operator and sentencecase Operator.

Examples

uppercase Operator (Examples)

13th June 2019 at 5:36pm

Operator Examples uppercase Operator

```
[[Abc]uppercase[]]
```

Try it

```
[tag[HelloThere]uppercase[]]
```

Try it

URI

28th February 2015 at 12:36pm

Definitions

A **URI** (also often known as a **URL**) is a string of characters used to specify the location of a resource such as a web page.

Using a custom path prefix with the client-server edition

12th September 2014 at 4:16pm

TiddlyWiki on Node.js

By default, when running TiddlyWiki on Node.js, the server exposes the wiki at the URI formed from the protocol, host and port - for example, `http://127.0.0.1:8080/`.

There are two steps to running the wiki at a custom path like `http://127.0.0.1:8080/path/to/my/wiki/`:

1. Configure the server by passing `/path/to/my/wiki` as the **pathprefix** argument of the ServerCommand
2. Configure the client by creating a tiddler called `$:/config/tiddlyweb/host` that contains `$protocol$//$host$/path/to/my/wiki/`

Using Excise

5th May 2023 at 12:42pm

Editor toolbar

Excise text

From the *EditorToolbar* you can export selected text to a new tiddler and insert a link, Transclusion or macro in its place. Click **Excise text** (), input name of the new tiddler, and choose excise method.

How to excise text

1. Highlight the relevant piece of text
2. Click **Excise text** ('')
3. Give the new tiddler a title.
4. Choose if the new tiddler will be tagged with the title of the current tiddler (see note below).
5. Choose replacing method: link, transclusion, or macro.
6. Click the **Perform excision** button

Note! If you choose the option to `Tag new tiddler with the title of this tiddler`, the new tiddler will be tagged with the name of the current tiddler before it has been edited. If you have changed the title of the current tiddler, save it first and edit it again to perform excision with this option.

Using HTTPS

11th November 2021 at 3:36am

WebServer Guides

By default, TiddlyWiki's WebServer serves resources over the insecure HTTP protocol. The risk is minimal if it is only being used within a private, trusted network but in many situations it is desirable to use a secure HTTPS connection.

HTTPS requires the server to be configured with a certificate via a "cert" file and a "key" file, configured via the tls-cert and tls-key parameters.

ⓘ **Introduced in v5.2.2** The optional tls-passphrase parameter allows the server to use certificate files that have been generated with a passphrase/password.

Certificates can be obtained from a certification authority such as https://letsencrypt.org/, or you can create a self-signed certificate for internal testing.

To create the required certificate files with the popular openssl utility:

```
openssl req -newkey rsa:2048 -new -nodes -keyout mywikifolder/key.pem -out mywikifolder/csr.pem
openssl x509 -req -days 365 -in mywikifolder/csr.pem -signkey mywikifolder/key.pem -out mywikifolder/
server.crt
tiddlywiki mywikifolder --listen username=joe password=bloggs tls-key=key.pem tls-cert=server.crt
```

If using a tls-passphrase to generate the certificate files, the commands would change as below:

- remove the `-nodes` flag, which specifies "no encryption"
- replace `TLS_PASSPHRASE` in the `-passout` and `-passin` parameters in the below commands with your chosen string.

This is the simplest, but least secure method, of passing a passphrase to the certificate utility. See this Stack Exchange anwser on openssl certificates and the openssl and openssl-passphrase-options page in the openssl utility documentation.

```
openssl req -newkey rsa:2048 -passout pass:TLS_PASSPHRASE -new -keyout mywikifolder/key.pem -out
mywikifolder/csr.pem -passout pass:TLS_PASSPHRASE
openssl x509 -req -days 365 -in mywikifolder/csr.pem -signkey mywikifolder/key.pem -out mywikifolder/
server.crt -passin pass:TLS_PASSPHRASE
tiddlywiki mywikifolder --listen username=joe password=bloggs tls-key=key.pem tls-cert=server.crt tls-
passphrase=TLS_PASSPHRASE
```

Using links to navigate between tiddlers

29th November 2020 at 7:30pm

Working with TiddlyWiki

You can use links (normally displayed as blue text) to navigate from one tiddler to another. Clicking on a link to any tiddler will take you to that tiddler. If the tiddler is closed, it will be opened. The wonderful thing about TiddlyWiki is that it makes links to tiddlers as accessible as possible. There are links everywhere! Here are the key places where you can find links to tiddlers in TiddlyWiki:

- You can **create a link** to a tiddler, whether it exists yet or not, in the body of any tiddler. See Linking in WikiText to see the various easy ways to create links between tiddlers.
- Each **tag pill** in your tiddler (such as the **Working with TiddlyWiki** tag pill below the title of this tiddler) contains a link to that tag's tiddler, as well as lists of all the tiddlers that carry that tag. This lets you go to any of those tiddlers.
- The InfoPanel of each tiddler gives you access to four tabs containing additional lists of related tiddlers:
  - The **Backlinks** tab lists all the tiddlers that link *to* the current tiddler.
  - The **Tagging** tab lists all the tiddlers that have been tagged with the current tiddler's title.
  - The **List** tab lists all the tiddlers (or potential tiddlers) mentioned in the list field of the current tiddler.
  - The **Listed** tab lists all the tiddlers that mention the current tiddler in *their* list fields.
- The **sidebar tabs** contain numerous lists of links to tiddlers:
  - The **Open** tab lists all the tiddlers that are currently open, i.e. visible somewhere on the page.
  - The **Recent** tab lists the 100 most recently modified tiddlers, beginning with the most recently modified.
  - The **More** tab offers eight additional lists of tiddlers:
    - **All** lists all tiddlers in alphabetical order.
    - **Tags** lists all the tags. You can click on the pill for any tag to access a list of the tiddlers tagged with that tag.
    - **Missing** lists any tiddlers that don't yet exist, but have been linked to from other tiddlers. This is helpful for finding tiddlers you planned to create, but never got around to.
    - **Drafts** lists any tiddlers that are currently in draft mode. TiddlyWiki considers a tiddler's draft to be a separate tiddler for as long as you are editing it, so while you're editing a tiddler entitled **Australia**, there will be two tiddlers, **Australia** and **Draft of**

'Australia'. When you close the draft by saving your changes to it, those changes will be applied to the **Australia** tiddler. So treat the **Drafts** tab as a way of finding any unfinished drafts you started.

- **Orphans** lists all the tiddlers that are not directly linked from another tiddler. This is a helpful aid to editing your file - it shows you which tiddlers need to be integrated more carefully with the others.
- **Types** lists any tiddlers with special content, such as images or audio.
- **System** lists all of the SystemTiddlers.
- **Shadows** lists all of the ShadowTiddlers.

- **Search results** are a list of the tiddlers that contain the text you type in the search box.
- Finally, you can create your own custom lists of tiddlers by various methods:
  - You can transclude a filter (see Transclusion in WikiText). For example, adding `{{{ [tag[mountain]] }}}` to a tiddler will insert a list of all tiddlers tagged with **mountain**.
  - You can use the ListWidget. This is more complicated than transcluding a filter, but in return it allows you more flexibility in designing and displaying the list exactly as you want it to appear.

Using Stamp

1st January 2023 at 12:49pm

Editor toolbar

Snippets are pre-configured snippets of text which can be inserted into the editor by clicking the **stamp** (🖃) button on the toolbar and then on the required item.

Snippets can either **replace**, or be **added before and/or after**, the selected text in the editor.

Create a snippet

Whilst editing a tiddler

1. Click **stamp** (🖃) on the toolbar
2. Click "*Add your own*" at the bottom of the menu
3. Type the text for the snippet in the editor
4. Enter the caption for the snippet menu item in the `caption` field
5. Click the ✓ *Confirm changes to this tiddler* button

Manually

1. Create a new tiddler by clicking the *Create a new tiddler* + button in the sidebar
2. Change the title for the tiddler (from e.g. `New Tiddler 1`). The tiddler title determines the menu item for the stamp if you do not set a `caption` field (see below).

   Pro tip Set a title like `$:/yourusername/snippets/My new stamp` to 'file' your new tiddler away as a system tiddler which doesn't appear in the normal search results
3. Add the tag $:/tags/TextEditor/Snippet by entering `$:/tags/TextEditor/Snippet` in the `tag name` tag box and clicking *Add* (or pressing the *Enter* key)
4. Enter the snippet content (which you wish to appear in the editor when you click the stamp menu 🖃) in the editor
5. Create a field with the name `caption` and value set to the text for the snippet's stamp menu item:
   1. Enter `caption` in the *Add new field* `field name` box
   2. Enter the stamp menu item caption in the `field value` box

Re-ordering snippets

Your new snippet will appear at the bottom of the stamp menu by default. To adjust the order of snippets:

1. Navigate to a snippet tiddler
2. Click the $:/tags/TextEditor/Snippet tag pill
3. Drag items up and down the menu of snippets

Adding a prefix and/or suffix to a selection ⓘ Introduced in v5.1.20

By default, the stamp button **replaces** text you have selected in the editor (let's say `selectedText`)with your snippet (let's say `$:/my/snippet`).

If instead of *replacing* selected text, you wish the stamp button to *add* wikitext before and/or after the selection:

1. Create a snippet tiddler (tagged $:/tags/TextEditor/Snippet , with `caption` field set to desired stamp menu item text) using either of the two methods above
2. Create a new tiddler with either `/prefix` or `/suffix` appended to the end of the snippet tiddler's title, according to whether you wish the stamp to insert the snippet content *before* (`/prefix`) or *after* (`/suffix`) the text selected in the editor.
    - Do not tag the tiddler $:/tags/TextEditor/Snippet ; the caption field is ignored
    - Set the tiddler text to the text you wish to be inserted before or after your selection

Tiddler	Tiddler content	Tag	`caption` field

Example configuration for `snippet2` with prefix+suffix

`$:/snip/snippet2`		$:/tags/TextEditor/Snippet	`My second stamp menu item`
`$:/snip/snippet2/prefix`	`TextIWantPrependedToSelection`		
`$:/snip/snippet2/suffix`	`TextIWantAppendedToSelection`		

Using Stylesheets

25th November 2021 at 6:57pm

Customise TiddlyWiki

Theme and Colorpalette

The first steps to changing the appearance of TiddlyWiki are to choose and apply:

- One of the available themes: theme
- Modify the colour palette: palette
- Experiment with the ⚙ ControlPanel

Work with Stylesheets

In addition to the control panel, custom styles can be defined by tagging a tiddler `$:/tags/Stylesheet`. Try creating a custom stylesheet now with the following content in order to change the page background colour to red:

```
body.tc-body {
```

```
    background: red;
}
```

You can then use your own styles and classes in WikiText.

Additional Resrouces

- Cascading Style Sheets (CSS) at mozilla
- Cascading Style Sheets (CSS) at w3scools

Overriding Theme Settings

Custom stylesheets are applied independently from theme stylesheets. Therefore, it is often necessary for the css rules in your custom stylesheet to be more specific than those of the theme you want to override. For example, `html body.tc-body` is more specific than `body.tc-body`.



You should always start with the least specific value that works!

Stylesheet Types

Usually it is best to use the type `text/css` for stylesheets. This treats them as plain stylesheets, and ensures that TiddlyWiki doesn't apply any wiki processing to them.

If you wish to use macros and transclusions in your stylesheets you should instead use the default WikiText type `text/vnd.tiddlywiki`. This allows full WikiText processing to be performed. Here is an example:

```
\rules only filteredtranscludeinline transcludeinline macrodef macrocallinline html

body.tc-body pre {
    <<box-shadow "inset 0 1px 0 #fff">>
}
```

The `\rules` pragma at the top of the tiddler restricts the WikiText to just allow macros and transclusion. This avoids mistakenly triggering unwanted WikiText processing.

A stylesheet tiddler is processed such that it is first wikified and then the text portion of the ouput is extracted to apply as the CSS. Any HTML tags you will use in your stylesheet are thus ignored. For example, HTML elements generated by the RevealWidget will not affect the output. As in the following example, you can wrap CSS rules in `<pre>` tags to display them as a codeblock without affecting processing, including handling the inner macro.

```
\rules only filteredtranscludeinline transcludeinline macrodef macrocallinline html

<pre>body.tc-body pre {
    <<box-shadow "inset 0 1px 0 #fff">>
}
</pre>
```

Stylesheet Macros

The TiddlyWiki core provides several global macros that are helpful in constructing stylesheets.

See Also

- ◦ Debugging Stylesheets

Using SVG

18th June 2016 at 10:58am

Features

TiddlyWiki5 allows you to use SVG to display vector graphics in two ways:

- Tiddlers with the type `image/svg+xml` are interpreted as SVG images, and displayed and transcluded as self-contained `<img>` elements with the SVG embedded as a data URI in the `src` attribute.
  - For examples of SVG images see Motovun Jack.svg and Tiddler Fishes.svg
- WikiText can also include inline SVG elements directly. See below for an example.

Embedding SVG tiddlers

You can embed an SVG image tiddler using the ordinary transclusion syntax:

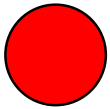
```
{{Motovun Jack.jpg}}
```

You can also use Typed Blocks in WikiText to embed an inline SVG tiddler.

The implications of the image being rendered within an `<img>` element are that it is sandboxed; it can't use CSS styles from the parent document, for example. Neither can the image use WikiText features like transclusion.

Embedding SVG elements

The other way to use SVG is to embed the `<svg>` element directly. For example:



Note that inline SVG elements don't need an `<?xml version="1.0"?>` directive.

Including HTML or WikiText content in SVG images

You can include simple text strings in SVG images using the `<text>` element:

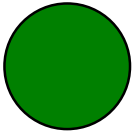


HTML or WikiText content can be included within inline SVG images using the `<foreignObject>` element. For example:



Transcluding SVG elements

When embedding SVG elements you can also use WikiText features like transclusion. For example, here is an SVG circle with the radius set to the value in the tiddler $:/SVGExampleRadius:



You can edit the value of the radius here: 50

Making curved text with SVG

This demo shows how to use SVG to render transcluded text along a path. Enter some text in the textbox below to try it out; view the source to see how it is done.

Everything in federation



Using the external JavaScript template

19th March 2023 at 2:08pm

WebServer Guides

You can use a special template to externalise TiddlyWiki's core code into a separate file. This configuration allows the browser to cache the core for improved efficiency.

Background

TiddlyWiki in the single file configuration ordinarily packs everything into a single file: your data, and the JavaScript, CSS and HTML comprising

TiddlyWiki itself. This lack of dependencies is usually very convenient: it means that it is impossible for the parts of a TiddlyWiki to become separated, and enormously improves the chances of it still functioning in the future.

However, there is some inefficiency in this arrangement because the core code is repeatedly loaded and saved every time the content of the wiki is saved. This inefficiency is partially ameliorated when working in the client server configuration because once the wiki is loaded by the browser the synchronisation process only transmits individual tiddlers back and forth to the server.

The remaining inefficiency when working in the client server configuration is that the single page wiki that is initially loaded will contain a copy of the entire core code of TiddlyWiki, making it impossible for the browser to cache it.

Using the external JavaScript template with the client-server configuration

The mechanism is activated by setting the root-tiddler parameter to `$:/core/save/all-external-js`. This template externalises TiddlyWiki's core JavaScript into a separate file. For example, the following command will start your server with caching enabled. It will transfer the wiki with two GET requests, and the core can be cached by the browser.

```
tiddlywiki YOUR_WIKI_FOLDER --listen 'root-tiddler=$:/core/save/all-external-js' use-browser-cache=yes
```



You can activate your browser's **Developer Tools: Network** tab and reload the page to see how it works.



On Windows, Command Prompt (CMD) users need to replace single quotes `'` with double quotation marks `"`.

Initialising a new wiki

The provided edition `server-external-js` contains all the configuration necessary to use the external JavaScript template. Here is an example:

```
tiddlywiki ./myNewWiki --init server-external-js
tiddlywiki ./myNewWiki --build listen
```

The above commands perform the following:

- Create a new wiki with external JavaScript customisation included.
- Start the server with external JavaScript enabled. The server listens on port 8080. Visit http://localhost:8080 in your browser.

To customise your `--build listen` command, see tiddlywiki.info Files and ListenCommand.

Using the external JavaScript template with the single file configuration

You can use the "external-js" template with your single file wiki, but this requires that you have TiddlyWiki's core JavaScript saved alongside your HTML file. You may prefer this configuration, for example, if you have several wikis on a WebDav server. (See: Saving via WebDAV)



This functionality is intended for advanced users: there are many more ways for things to go wrong than with the standard single file configuration

Saving your snapshots

If you start your TiddlyWiki server in the "external-js" configuration, snapshots you save from the wiki will also have the external-js configuration.

You can click on the "cloud" button and choose **Save snapshot for offline use**. The html wiki you saved will have a reduced file size compared to a regular snapshot because the TiddlyWiki core code has been externalised. However, to be able to use this wiki, you **must** also have a copy of TiddlyWiki's core JavaScript in the same directory; see below for instructions for obtaining it

The "server-external-js" edition lets you save the snapshot from the command line:

```
tiddlywiki YOUR_WIKI_FOLDER --build index
```

The files `external-5-x-x.html` and `tiddlywikicore-5.x.x.js` will be saved in your wiki folder's `output` directory.

Obtaining the TiddlyWiki core in the browser

For advanced users

Export the TiddlyWiki core JavaScript code for running with external JavaScript:

> Download TiddlyWiki core

Further information

Obtaining the TiddlyWiki core with Node.js

You can also run the following command to obtain the core JavaScript. The file `tiddlywikicore-5.x.x.js` will be saved in the `output` directory of your wiki folder:

```
tiddlywiki YOUR_WIKI_FOLDER \
```

```
    --render '$:/core/templates/tiddlywiki5.js' \
    '[[tiddlywikicore-]addsuffix<version>addsuffix[.js]]' \
    'text/plain'
```

The "server-external-js" edition has a shortcut for the BuildCommand to do the same thing:

```
tiddlywiki YOUR_WIKI_FOLDER --build tiddlywikicore
```

Upgrading your standalone wiki



> This procedure is experimental, please take care to backup your data

Before you proceed, backup your wiki first! Follow the steps below to upgrade a single-file wiki with the external JavaScript template:

1. Proceed with the Upgrade Process for Standalone TiddlyWikis. Your wiki will be converted to a full standalone HTML.
2. Open your upgraded wiki in the browser. If you'd like to revert to using the regular template, restore the original shadow tiddler `$:/config/SaveWikiButton/Template` by deleting any custom copy. Save your wiki and you are done.
3. Otherwise, from your wiki, select the **Tools** tab from the sidebar and click on the ★ **export tiddlywiki core** button to obtain the core from your wiki.
4. Verify that `$:/config/SaveWikiButton/Template` still contains `$:/core/save/offline-external-js`. Modify if necessary.
5. Save your wiki again. Your wiki is now converted to using the external core.

Using the integrated static file server

3rd July 2018 at 12:04pm

WebServer Guides

Any files in the subfolder `files` of the wiki folder will be available via the route `\files\<uri-encoded-filename>`. For example: http://127.0.0.1:8080/files/Motovun%20Jack.jpg

This can be useful for publishing large files that you don't want to incorporate into the main wiki (PDFs, videos, large images, etc.).

Static files can be referenced directly:

- `[ext[./files/a-big-document.pdf]]` - to make a link to a PDF
- `[img[./files/a-big-image.png]]` - to embed an image

Alternatively, the **_canonical_uri** field can be used to reference the files as external tiddlers.

Using the read-only single tiddler view

24th August 2018 at 9:32am

WebServer Guides

TiddlyWiki's experimental single tiddler per page, read-only view uses a simplified page layout, and implements links between tiddlers, but there are no other interactive features. Compared to a full TiddlyWiki user interface, it is very lightweight and usable even over very slow connections.

Alongside serving the full interactive wiki at the path `/` (e.g. http://127.0.0.1:8080/), TiddlyWiki serves each tiddler at the path `/<url-encoded-tiddler-title>`. For example:

- http://127.0.0.1:8080/HelloThere
- http://127.0.0.1:8080/Philosophy%20of%20Tiddlers

Ordinary, non-system tiddlers are rendered through a special view template while system tiddlers are rendered through a template that returns the raw text of the rendered output. In this way ordinary tiddlers can be browsed by end users while system tiddlers can be included in their raw form to use them as JS, HTML or CSS templates. Additionally these defaults can be overwritten on a per tiddler basis by specifying the `_render_type` and `_render_template` fields accordingly.

The templates are controlled by these parameters:

- system-tiddler-render-type
- system-tiddler-template
- tiddler-render-type
- tiddler-template

Using TiddlyWiki for GitHub project documentation

12th September 2014 at 4:15pm

TiddlyWiki on Node.js

TiddlyWiki5 can be used to produce documentation for GitHub projects. It lets you maintain a single set of documentation as a TiddlyWikiFolder containing separate tiddler files under source code control, and then use it to produce `readme.md` files for inclusion in project folders, or HTML files for storage in GitHub Pages. Both features are demonstrated by TiddlyWiki5 itself.

Generating `readme.md` files

When displaying the contents of a folder GitHub will look for a `readme.md` file and display it. Note that it will not display full HTML files in this

way, just static *MarkDown* files (this is a security measure). Happily *MarkDown* permits a safe subset of HTML, and thus to generate a `readme.md` file that is suitable for GitHub it is just necessary for TiddlyWiki5 to generate the content of the `<body>` element of an HTML document, and give it the appropriate filename.

This is done with this command:

```
--rendertiddler ReadMe ./readme.md text/html
```

It saves the tiddler ReadMe to the file `./readme.md` in the `text/html` format.

By default, tiddler links will be rendered as `<a>` links to a relative URI consisting of the title of the tiddler. This behaviour can be overridden by defining the macro `tv-wikilink-template`, as is done at the top of the tiddler ReadMe:

```
\define tv-wikilink-template() https://tiddlywiki.com/static/$uri_doubleencoded$.html
```

See the LinkWidget for more details.

In this example, tiddler links are rendered as links to the static rendering of tw5.com.

Using TiddlyWiki on Node.js

13th June 2022 at 7:23pm

TiddlyWiki on Node.js

TiddlyWiki5 includes a set of commands for use on the command line to perform an extensive set of operations based on TiddlyWikiFolders, TiddlerFiles.

For example, the following command loads the tiddlers from a TiddlyWiki HTML file and then saves one of them in static HTML:

```
tiddlywiki --verbose --load mywiki.html --rendertiddler ReadMe ./readme.html
```

Running `tiddlywiki` from the command line boots the TiddlyWiki kernel, loads the core plugins and establishes an empty wiki store. It then sequentially processes the command line arguments from left to right. The arguments are separated with spaces.

Introduced in v5.1.20 First, there can be zero or more plugin references identified by the prefix `+` for plugin names or `++` for a path to a plugin folder. These plugins are loaded in addition to any specified in the TiddlyWikiFolder.

The next argument is the optional path to the TiddlyWikiFolder to be loaded. If not present, then the current directory is used.

The commands and their individual arguments follow, each command being identified by the prefix `--`.

```
tiddlywiki [+<pluginname> | ++<pluginpath>] [<wikipath>] [--<command> [<arg>[,<arg>]]]
```

For example:

```
tiddlywiki --version
tiddlywiki +plugins/tiddlywiki/filesystem +plugins/tiddlywiki/tiddlyweb mywiki --listen
tiddlywiki ++./mygreatplugin mywiki --listen
```

Introduced in v5.1.18 Commands such as the ListenCommand that support large numbers of parameters can use NamedCommandParameters to make things less unwieldy. For example:

```
tiddlywiki wikipath --listen username=jeremy port=8090
```

See Commands for a full listing of the available commands.

Utility by Mohammad

17th November 2020 at 5:02pm

Community Plugins

The utility plugin objective is to provide set of tools for developers and authors.

https://kookma.github.io/TW-Utility/

These tools include simple transclusions, show raw contents of tiddlers, author tools, wikitext macros, show fields in view mode, etc.

Utility Classes

10th October 2022 at 9:42am

WikiText

ⓘ **Introduced in v5.2.4** The following outlines a few predefined CSS classes intended to make it simpler to style HTML block-elements and wikitext tables.

General Utility Classes

`tc-center`	Centres a block-element to the middle of the container
`tc-max-width`	Expands a block-element to use the maximum width of the container
`tc-max-width-80`	Sets the width of a block-element to use 80% of the maximum container width. This setting is useful with the

	`tc-center` class
`tc-edit-max-width`	Expands TextWidgets to use the maximum available width. See **ControlPanel -> Info -> Basics**
`tc-first-link-nowrap`	Ensures that any links in the first table column will never wrap to the next line

Table Utility Classes

`tc-table-no-border`	Removes the borders of a table
`tc-first-col-min-width`	The first column of a table will take up minimal possible width. It adapts to the content

Variable Attribute Values

26th July 2023 at 5:10pm

Widget Attributes WikiText

Variable attribute values are indicated with double angle brackets around a macro invocation. For example:

```
<div title=<<MyMacro "Brian">>>
...
</div>
```

The behaviour of variables invoked via widget attributes is not the same as when they are invoked via normal wikitext. In addition, the behaviour depends on how the variable is declared:

how declared	behaviour

\define	Textual substitution of parameters is performed on the body text. No further processing takes place. The result after textual substitution is used as the attribute's value
$set, **$let**, **$vars**, \procedure, \widget	Body text is retrieved as-is and used as the attribute's value.
\function	When a function is invoked as `<div class=<<macro>>/>`, it is a synonym for `<div class={{{[function[macro]]}}}/>`. As with any filtered transclusion (i.e. triple curly braces), all results except the first are discarded. That first result is used as the attribute's value. Note that functions are recursively processed even when invoked in this form. In other words a filter expression in a function can invoke another function and the processing will continue

Variable Usage

26th July 2023 at 4:59pm

Ways to define variables and parameters

how declared	how parameters are defined	accessing parameter values in the body
\define	`()`	`$param$, <<__param__>>`
	$parameters or `\parameters`	`<<param>>`
$set, **$let**, **$vars**	**$parameters** or `\parameters`	`<<param>>`

\procedure, \widget	`()`, `$parameters` or `\parameters`	`<<param>>`
\function	`()`	`<param>`
javascript macros	`exports.params` javascript property array	passed as normal javascript function parameter and so accessed as a normal javascript variable

Examples

These examples are meant to provide insight into the various ways of defining and using parameters. In many cases they do not illustrate best practices.

\define

copy to clipboard

```
\define mp1(a1)  $a1$ - <<__a1__>>

\define mp2() <$parameters a1><<a1>></$parameters>

\define mp3()
\parameters(a1)
<<a1>>
\end

|<<mp1 foo>>|<<mp2 foo>>|<<mp3 foo>>|
```

$set, $let, $vars

copy to clipboard

```
<$set name="sp1" value="<$parameters a1><<a1>></$parameters>">
<$set name="sp2" value="""
\parameters(a1)
<$parameters a1><<a1>></$parameters>
""">
<$vars vp1="<$parameters a1><<a1>></$parameters>" vp2="""
\parameters(a1)
<$parameters a1><<a1>></$parameters>
""">
<$let lp1="<$parameters a1><<a1>></$parameters>" lp2="""
\parameters(a1)
<$parameters a1><<a1>></$parameters>
""">

|<<sp1 foo>>|<<sp2 foo>>|
|<<vp1 foo>>|<<vp2 foo>>|
|<<lp1 foo>>|<<lp2 foo>>|
</$let>
</$vars>
</$set>
</$set>
```

\procedure, \widget

copy to clipboard

```
\procedure pp1(a1)  <<a1>>
```

```
\procedure pp2() <$parameters a1><<a1>></$parameters>

\procedure pp3()
\parameters(a1)
<<a1>>
\end

\procedure wp1(a1)  <<a1>>

\widget wp2() <$parameters a1><<a1>></$parameters>

\widget wp3()
\parameters(a1)
<<a1>>
\end

|<<pp1 foo>>|<<pp2 foo>>|<<pp3 foo>>|
|<<wp1 foo>>|<<wp2 foo>>|<<wp3 foo>>|
```

\function

copy to clipboard

```
\function fp1(a1)  [<a1>]
|<<fp1 foo>>|
```

Behaviour of invoked variables depends on how the variable was declared

Invoked in normal wikitext context: `<$transclude $variable=macro/>` or `<<macro>>`

how declared	behaviour
\define	All wikitext and variable substitution and textual substitution takes place
$set, **$let**, **$vars**, \procedure, \widget	All wikitext and variable substitution takes place
\function	Invoking a function in this way (`<<macro>>`) is a synonym for `<$text text={{{[function[macro]]}}}/>`. As with any filtered transclusion (i.e. triple curly braces), all results except the first are discarded

Invoked via widget attribute: `<div class=<<macro>>/>`

how declared	behaviour
\define	Textual substitution of parameters is performed on the body text. No further processing takes place. The result after textual substitution is used as the attribute's value
$set, **$let**,	Body text is retrieved as-is and used as the attribute's value.

$vars, \procedure, \widget	
\function	When a function is invoked as `<div class=<<macro>>/>`, it is a synonym for `<div class={{{[function[macro]]}}}/>`. As with any filtered transclusion (i.e. triple curly braces), all results except the first are discarded. That first result is used as the attribute's value. Note that functions are recursively processed even when invoked in this form. In other words a filter expression in a function can invoke another function and the processing will continue

Invoked via filter operator parameter: `[<macro>]`

how declared	behaviour
\define	Textual substitution of parameters is performed on the body text. No further processing takes place. The result after textual substitution is used as the filter operator's parameter.
$set, **$let**, **$vars**, \procedure, \widget	Body text is retrieved as-is and used as the filter operator's parameter.
\function	The body text of the function is treated as a filter expression and evaluated. The first result is passed to the operator as a parameter. The remaining results are discarded.

Invoked via function call in a filter expression: `[function[macro]]`

how	behaviour

declared	
\define, **$set**, **$let**, **$vars**, \procedure, \widget	Every function is a variable, but only variables defined using \function are invokable using the **function** filter operator. Attempts to use a non-function variable is the same as if the function doesn't exist. The behavior in this case is like the identity function. All filter input is passed unchanged to the output.
\function	The body text of the function is treated as a filter expression and evaluated. This filter expression can itself contain a function call. Filter expressions can be factored out into functions arbitrarily deep.

Examples

Below is an example macro, procedure and function definition. All three forms of parameter substitution `$a1$`, `<<__a1__>>`, and `<<a1>>` are included in each definition. The output helps illustrate when each form of substitution will or will not have affect.

```
\define m1(a1) $a1$ - <<__a1__>> - <<a1>>
\procedure p1(a1) $a1$ - <<__a1__>> - <<a1>>
\function f1(a1) $a1$ "-" [<__a1__>] ="-" [<a1>] :and[join[ ]]
```

Variable transclusion	output
`<<m1 foo>>`	foo - foo -
`<<p1 foo>>`	$a1$ - - foo
`<<f1 foo>>`	$a1$ - - foo
Widget attribute	**output**
`<$text text=<<m1 foo>>/>`	foo - <<__a1__>> - <<a1>>

`<$text text=<<p1 foo>>/>`	$a1$ - <<__a1__>> - <<a1>>
`<$text text=<<f1 foo>>/>`	$a1$ - - foo
Filter operator parameter	**output**
`[<m1 foo>]`	foo - <<__a1__>> - <<a1>>
`[<p1 foo>]`	$a1$ - <<__a1__>> - <<a1>>
`[<f1 foo>]`	$a1$ - - foo
Function call in filter expression	**output**
`[function[m1],[foo]]`	"A free, open source wiki revisited" by Mark Gibbs, NetworkWorld
`[function[p1],[foo]]`	"A free, open source wiki revisited" by Mark Gibbs, NetworkWorld
`[function[f1],[foo]]`	$a1$ - - foo

Namespaces

- **tiddler titles** - tiddlers are uniquely identified by their title. The namespace for tiddler titles and variable names are completely separate.
- **variables** - \define, **`$set`**, **`$let`**, **`$vars`**, \procedure, \widget, \function all create variables. If the same name is used, then later define will overwrite earlier defined
- **`function` filter operator parameter** - only variables defined using \function can be called using the **`function`** operator
- **filter operators** - only the javascript defined filter operators and variables defined using \function with name containing a dot can be called
- **widgets** - variables defined using \widget can be invoked using `<$widget/>` syntax ONLY if the name starts a dollar sign. Without the dollar sign prefix, defining variables using \widget is no different than using \procedure.

Variables

22nd April 2023 at 5:04pm

Concepts Reference WikiText

Introduction

A **variable** is a snippet of text that can be accessed by name. The text is referred to as the variable's **value**.

Variables are defined by widgets. Several core widgets define variables, the most common being the `$set`, `$let` and `$list` widgets.

The values of variables are available to descendant widgets, including transcluded content. For example, within each tiddler in the main story river the variable "currentTiddler" is set to the title of the tiddler.

Variables can also be overwritten by descendent widgets defining variables of the same name, thus binding a different snippet to that name for the scope of the children of the widget.

Special Kinds of Variables

There are several special kinds of variable that extend their basic capabilities:

- Procedures are snippets of text that can be passed parameters when wikified
- Functions are snippets of text containing filters with optional named parameters
- Custom Widgets are snippets of text containing definitions of custom widget
- Macros are snippets of text that can contain placeholders that are filled in with parameters whenever the macro is used

Note that these special kinds of variable can only be created with the associated shortcut definition syntax.

For a more detailed comparison of these special kinds of variables, see Variable Usage.

Defining Variables

The following core widgets are commonly used to define variables:

- **$let** widget – the easiest way to define multiple variables
- **$set** widget – the most flexible way to define a single variable
- **$parameters** widget – used to declare parameter variables within procedures and custom widgets
- **$list** widget – defines a loop variable and optional counter variable
- **$setmultiplevariables** widget – allows creation of multiple variables at once where the names and values are not known in advance

Using Variables

Once a variable is defined there are several ways to access it.

Transcluding Variables

Transcluding a variable renders the text contents of the variable as if it replaced the call. It is a shortcut syntax for the **$transclude** widget with the `$variable` attribute.

```
<<varname>>
```

Parameters can be passed to the transclusion as follows:

```
<<varname "This is a parameter">>
<<varname param:"This is a parameter">>
<<varname param:"This is a parameter" second:"Another parameter">>
```

The handling of these parameters depends on the kind of variable:

- Procedures assign the parameters to variables that are available within the procedure
- Macros replace the text of the special markers `$param$` with the values passed to the macro for those parameters (see Macro Parameter Handling for the details)

The parameters are ignored for other kinds of variable.

Macro Variable Substitutions

Before the text of a macro is used, the special markers `$(variable)$` are replaced with the values of the named variable.

Variable Attributes

Variables can be used as the value of attributes of widgets or HTML elements:

```
<div class=<<varname>>>
```

Parameters can be passed:

```
<div class=<<varname "This is a parameter">>>
...
<div class=<<varname param:"This is a parameter">>>
```

```
...
<div class=<<varname param:"This is a parameter" second:"Another parameter">>>
...
```

The handling of these parameters depends on the kind of variable:

- Functions assign the parameters to variables that are available within the function
- Macros replace the text of the special markers `$param$` with the values passed to the macro for those parameters (see Macro Parameter Handling for the details)

The parameters are ignored for other kinds of variable.

Variables in Filters

Variables can be accessed within Filters using angle brackets to quote the name:

```
[<varname>]
```

Parameters can be passed in the usual way:

```
[<varname "This is a parameter">]
[<varname param:"This is a parameter">]
[<varname param:"This is a parameter" second:"Another parameter">]
...
```

See Also

- The **dumpvariables** macro lists all variables that are available at that position in the widget tree
- Complete listing of TiddlyWiki's built-in Core Variables

Examples

Example of Defining a Variable

copy to clipboard

```
<$set name=animal value=zebra>
<<animal>>
</$set>
```

Try it

Example of Defining a Macro

The `\define` pragma below defines a macro called **tags-of-current-tiddler**. The macro returns the value of the tiddler's **tags** field, and can be accessed from anywhere else in the same tiddler (or in any tiddler that imports it).

```
\define tags-of-current-tiddler() {{!!tags}}
```

copy to clipboard

```
The tags are: <<tags-of-current-tiddler>>
```

Try it

Example of Using a Variable as a Filter Parameter

This example uses the **backlinks** operator to list all tiddlers that link to this one.

copy to clipboard

```
<<list-links filter:"[<currentTiddler>backlinks[]]">>
```

Try it

Variables in WikiText

21st February 2015 at 11:18pm

See Variables.

variables Operator

30th March 2019 at 11:01am

Filter Operators Selection Constructors Special Operators

purpose	select the names of all the actively defined variables

input	ignored
parameter	none
output	the names of all the actively defined variables

Learn more about how to use Filters

Introduced in v5.1.20 The primary purpose of the `variables` operator is to implement the dumpvariables Macro:

```
\define dumpvariables()
\whitespace trim
<ul>
<$list filter="[variables[]]" variable="varname">
<li>
<strong><code><$text text=<<varname>>/></code></strong>:<br/>
<$codeblock code={{{ [<varname>getvariable[]] }}}/>
</li>
</$list>
</ul>
\end
```

Examples

variables Operator (Examples)

30th March 2019 at 11:01am

Operator Examples variables Operator

```
[variables[]prefix[colour]]
```

→ returns the names of any variables whose names start with **colour**

Try it

variance Operator

26th April 2021 at 3:15pm

Filter Operators Mathematics Operators Reducing Mathematics Operators

purpose	treating each input title as a number, compute their variance
input	a selection of titles
output	the variance of the input as numbers

Learn more about how to use Filters

Introduced in v5.2.0 See Mathematics Operators for an overview.

The `standard-deviation` operator treats the input as a complete population and not a sample

Examples

variance Operator (Examples)

26th April 2021 at 3:15pm

Operator Examples variance Operator

```
1 3 4 5 +[variance[]]
```

Try it

Note that if there is no input the operator returns `NaN`

```
[title[NotATiddler]is[tiddler]get[price]] +[variance[]]
```

Try it

VarsWidget

26th April 2015 at 1:59pm

Widgets

Introduction

The **$vars** widget allows multiple variables to be set in one operation. In some situations it can result in simpler code than using the more flexible **$set** widget. It differs from the **$let** in that variables cannot interfere with the evaluation of other variables within the same **$vars**.

Content and Attributes

The content of the **$vars** widget is the scope for the value assigned to the variable.

Attribute	Description
{attributes not starting with $}	Each attribute name specifies a variable name. The attribute value is assigned to the variable

Attributes will not interfere with the evaluation of other attributes. So if one attribute sets **currentTiddler**, and another attribute uses **currentTiddler** in its evaluation, it will use the value of **currentTiddler** that exists outside the widget's scope.

Examples

Consider a case where you need to set multiple variables.

Using the **$vars** widget, this situation may be handled in the following way:

```
\define helloworld() Hello world!

<$vars greeting="Hi" me={{!!title}} sentence=<<helloworld>>>
  <<greeting>>! I am <<me>> and I say: <<sentence>>
</$vars>
```

In contrast, here is the same example using the **$set** widget:

```
<$set name="greeting" value="Hi" >
<$set name="me" value={{!!title}} >
<$set name="sentence" value=<<helloworld>> >
  <<greeting>>! I am <<me>> and I say: <<sentence>>
</$set>
</$set>
</$set>
```

Remarks

It should be noted that this widget differs from the **`$set`** widget in the following ways:

- A fallback (also known as "emptyValue") cannot be specified
- Filters cannot be used to produce a conditional variable assignment
- Variable names must be literal strings

VerboseCommand

Commands

Triggers verbose output, useful for debugging

```
--verbose
```

version Macro

21st February 2015 at 10:41pm

Core Macros Macros

The **version** macro returns the current version number of TiddlyWiki.

Parameters

(none)

Examples

version Macro (Examples)

21st February 2015 at 10:41pm

Macro Examples version Macro

copy to clipboard

```
Version number: <<version>>
```

Try it

VersionCommand

Commands

Displays the version number of TiddlyWiki.

```
--version
```

Videos

19th September 2014 at 6:12pm

Learning

The following TiddlyWiki videos are available.

- Getting Started Video
- Introducing TiddlyDesktop Video
- Introduction Video
- OXTWIG
- TiddlyWiki Hangouts
- TiddlyWiki on Firefox for Android Video

The aim is to curate a series of videos to guide people through getting up and running with TiddlyWiki. Contributions are welcome.

View Template Body Cascade

6th December 2021 at 5:16pm

Cascades

The view template body cascade is a cascade used by the default view template to choose the template for displaying the tiddler body.

The default view template body cascade consists of:

1. If the tiddler title starts with any of a list of known system tiddler prefixes, use the template **$:/core/ui/ViewTemplate/body/code** to display the body as preformatted code
2. If the tiddler has the field **plugin-type** set to **import** then use the template **$:/core/ui/ViewTemplate/body/import** which displays the custom import user interface
3. If the tiddler has the field **plugin-type** then use the template **$:/core/ui/ViewTemplate/body/plugin** to display the plugin information badge
4. If the tiddler has the field **hide-body** set to **yes** then use the template **$:/core/ui/ViewTemplate/body/blank** to hide the body
5. Otherwise, use the default template **$:/core/ui/ViewTemplate/body/default**

You can see the current settings for the view template body cascade in **$:/ControlPanel** under the **Info** -> **Advanced** -> **Cascades** -> **View Template Body** tab.

View Template Title Cascade

6th December 2021 at 5:11pm

Cascades

The view template title cascade is a cascade used by the default view template to choose the template for displaying the tiddler title.

The core view template title cascade can be found in **$:/core/ui/ViewTemplate/title**

The default view template title cascade consists of:

1. If the tiddler title starts with `$:/` then use the template **$:/core/ui/ViewTemplate/title/system** which causes the `$:/` prefix to be displayed in pale text
2. Otherwise, use the template **$:/core/ui/ViewTemplate/title/default** which displays the title in plain text

You can see the current settings for the view template title cascade in **$:/ControlPanel** under the **Info** -> **Advanced** -> **Cascades** -> **View Template Title** tab.

ViewWidget

19th December 2021 at 10:34pm

Widgets

Introduction

The view widget displays the contents of a tiddler field in a specified format.

Content and Attributes

The content of the `<$view>` widget is displayed if the field or property is missing or empty.

Attribute	Description
tiddler	The title of the tiddler (defaults to the current tiddler)

field	The name of the field to view (defaults to "text")
index	The name of the index to view
format	The format for displaying the field (see below)
template	Optional template string used when the `format` attribute is set to "date"
subtiddler	Optional *SubTiddler* title when the target tiddler is a plugin (see below)
mode	ⓘ **Introduced in v5.1.15** Optional transclusion parsing mode for wikified formats. May be "inline" or "block" (the default)

Formats

The following formats can be specified in the `format` attribute:

Format	Description
text	Plain text (default) See also the TextWidget. `<$view format=text tiddler=MyTiddler/>` renders the same as `<$text text={{MyTiddler}}/>`
htmlencoded	The field is displayed with HTML encoding. Rendered value is the same as the output from the encodehtml filter operator
htmltextencoded	ⓘ **Introduced in v5.2.0** The field is displayed with HTML encoding, only double quotes (") are not escaped. This creates a more compact htmlencoding appropriate for html text content, but *not* for attributes.
urlencoded	The field is displayed with URL encoding Rendered value is the same as the output from the encodeuri filter operator
doubleurlencoded	The field is displayed with double URL encoding Rendered value is the same as the output from calling the encodeuri filter operator twice
htmlwikified	The field is wikified according to the mode attribute and the resulting HTML returned as plain text (ie HTML

	elements will appear in plain text)
plainwikified	The field is wikified according to the mode attribute and the text content of the resulting HTML returned as plain text (ie HTML elements will be removed)
htmlencodedplainwikified	The field is wikified according to the mode attribute and the text content of the resulting HTML returned as HTML encoded plain text (ie HTML elements will be removed)
date	The field is interpreted as a UTC date and displayed according to the DateFormat specified in the `template` attribute Rendered value is the same as the output from the format:date filter operator
relativedate	The field is interpreted as a UTC date and displayed as the interval from the present instant Rendered value is the same as the output from the format:relativedate filter operator
stripcomments	The field is interpreted as JavaScript source code and any lines beginning `\\#` are stripped
jsencoded	The field is displayed as a JavaScript encoded string Rendered value is the same as the output from the stringify filter operator

"WikiText format"

To display the field content as regular WikiText, for example with working links, use the TranscludeWidget instead.

SubTiddler Access

The view widget allows access to the individual tiddlers stored within a plugin.

The following example will view the core version of the tiddler $:/DefaultTiddlers even if it has been overridden:

```
<$view tiddler="$:/core" subtiddler="$:/DefaultTiddlers"/>
```

That renders as:

GettingStarted

Examples

Given this Saturday example tiddler:

tiddler	raw content	rendered content
Saturday	<<.this-is-operator-example>>	This example tiddler is used to illustrate some of the Filter Operators.

Here is how `<$view tiddler=Saturday format=<<format>>/>` renders for each format:

format	renders as
text	<<.this-is-operator-example>>
htmlencoded	<<.this-is-operator-example>>
htmltextencoded	<<.this-is-operator-example>>
urlencoded	%3C%3C.this-is-operator-example%3E%3E%0A
doubleurlencoded	%253C%253C.this-is-operator-example%253E%253E%250A
htmlwikified	<p>This example tiddler is used to illustrate some of the <a class="tc-tiddlylink tc-tiddlylink-resolves"

	href="#Filter%20Operators">Filter Operators.</p>
plainwikified	This example tiddler is used to illustrate some of the Filter Operators.
htmlencodedplainwikified	This example tiddler is used to illustrate some of the Filter Operators.
stripcomments	<<.this-is-operator-example>>
jsencoded	<<.this-is-operator-example>>\n

vis.js Timeline by emkay, revived by kixam

6th January 2021 at 4:10pm

Community Plugins

An interactive timeline visualisation plugin based on Vis.js. A demo can be found here: http://kixam.github.io/TW5-visjsTimeline/.

> vis.js Timeline is a TiddlyWiki plugin that allows you to link your wiki-topics (tiddlers) in order to create clickable timelines. This project is a direct continuation of emkay's plugin.

Visible Transclusions

19th April 2023 at 12:31pm

Learning

Visible Transclusions

Block transclusions are shown in red, and inline transclusions are shown in green.

Click here to make transclusions visible within story river tiddlers

Web Developer Tools

20th November 2020 at 4:46pm

Definitions

Most browsers provide **Web Developer Tools** that include a JavaScript console. In Chrome and Firefox it can be opened via the **Tools** menu, or by pressing `Ctrl + Shift + J` (`Cmd + Shift + J` on a Mac).

WebServer

16th December 2018 at 7:19pm

Features ListenCommand ServerCommand

When running under Node.js, TiddlyWiki includes a simple HTTP/HTTPS web server that allows you to use it from any browser running on the same machine or over a network.



The web server includes a very simple mechanism allowing multiple users to log in with different credentials. The implementation is designed to be simple and easy to use, and would not generally be considered robust enough for use on the open internet. It is intended for use by individuals or small groups on a trusted network. It is recommended to use an external proxy before exposing it on the Internet.

How It Works

The web server listens for requests coming over the network, and performs the following actions in turn:

- Authentication is the process of identifying the current user. TiddlyWiki supports three types of authentication: Anonymous, Basic and Header
- Authorization is the process of determining which resources may be accessed by a particular user. TiddlyWiki implements a simple scheme whereby read and write access to the wiki can be independently controlled.
- Routing is the process of dispatching the request to the API handler, and returning any required data.

Usage

Anonymous Access

The web server is started with the ListenCommand (which supersedes the older ServerCommand). All the NamedCommandParameters are optional, so the simplest form is:

```
tiddlywiki mywikifolder --listen
```

Visit http://127.0.0.1:8080/ to access the wiki. Access is anonymous, so anyone can read or write to the wiki.

This will typically be available only to users on the local machine. For information on how to open the instance to the local network see the Web Server host parameter entry.

Authenticated Access

Adding username and password parameters enforces basic authentication for both reading and writing:

```
tiddlywiki mywikifolder --listen username=test password=tset
```

Visiting the wiki will prompt for a username and password, and access is denied if they do not match the provided credentials.

Anonymous Read, Authenticated Write

This example adds the authorization parameters readers and writers to grant read access to anonymous users, but require authentication as "joe" in order to gain write access.

> Note that anonymous users can trigger a username/password prompt by visiting the route `\login-basic` (eg http://127.0.0.1:8080/login-basic).

```
tiddlywiki mywikifolder --listen "readers=(anon)" writers=joe username=joe password=bloggs
```

Note the double quotes that are required for parameters containing special characters.

Arguments

The full list of available optional parameters is:

- admin
- anon-username
- authenticated-user-header
- credentials
- csrf-disable

- debug-level
- host
- password
- path-prefix
- port
- readers
- required-plugins
- root-render-type
- root-serve-type
- root-tiddler
- system-tiddler-render-template
- system-tiddler-render-type
- tiddler-render-template
- tiddler-render-type
- tls-cert
- tls-key
- tls-passphrase
- username
- writers

Guides

Further information on usage of the integrated WebServer:

- TiddlyWeb JSON tiddler format
- Using HTTPS
- Using the external JavaScript template
- Using the integrated static file server
- Using the read-only single tiddler view

WebServer Anonymous Access

1st July 2018 at 10:23pm

WebServer Authentication

Anonymous access is only permitted if the special **(anon)** token is present in the readers (for reading) and optionally writers (for writing) authorisation parameters.

WebServer API

2nd October 2018 at 3:18pm

WebServer

The TiddlyWiki WebServer API supports the following routes:

- WebServer API: Delete Tiddler
- WebServer API: Force Basic Authentication Login
- WebServer API: Get All Tiddlers

- WebServer API: Get Favicon
- WebServer API: Get File
- WebServer API: Get Rendered Tiddler
- WebServer API: Get Server Status
- WebServer API: Get Tiddler
- WebServer API: Get Wiki
- WebServer API: Put Tiddler

WebServer API: Delete Tiddler

18th April 2021 at 12:07pm

WebServer API

Delete a tiddler

```
DELETE /bags/default/tiddlers/{title}
```

Parameters:

- **title** - URI encoded title of the tiddler to delete

Headers:

- **x-requested-with** - must be set to `TiddlyWiki` in order for the request to succeed, unless WebServer Parameter: csrf-disable is set to `yes`

Response:

- 204 No Content

WebServer API: Force Basic Authentication Login

16th December 2018 at 7:19pm

WebServer API

Forces the server to request basic authentication login, and then redirects to the root

```
GET /login-basic
```

Requests an basic authentication from the browser, and redirects to the root if successful.

Parameters:

- none

Response:

- 302 Found

  ```
  Location: /
  ```

- 401 Unauthorized

  ```
  WWW-Authenticate: Basic realm="Please provide your username and password to login to
  ```

```
<servername>"
```

WebServer API: Get All Tiddlers

9th September 2022 at 11:43am

WebServer API

Gets an array of all raw non-system tiddlers, excluding the **text** field.

```
GET /recipes/default/tiddlers.json
```

Parameters:

- **filter** - filter identifying tiddlers to be returned (optional, defaults to "[all[tiddlers]!is[system]sort[title]]")
- **exclude** - comma delimited list of fields to excluded from the returned tiddlers (optional, defaults to "text")

In order to avoid denial of service attacks with malformed filters in the default configuration the only filter that is accepted is the default filter "[all[tiddlers]!is[system]sort[title]]"; attempts to use any other filter will result in an HTTP 403 error.

Note

> System tiddlers will not be returned by this API unless the Hidden Setting: Sync System Tiddlers From Server is explicitly switched on by setting **$:/config/SyncSystemTiddlersFromServer** to `yes`

To enable a particular filter, create a tiddler with the title "*$:/config/Server/ExternalFilters/*" concatenated with the filter text, and the text field set to "yes". For example, the TiddlyWeb plugin includes the following shadow tiddler to enable the filter that it requires:

```
title: $:/config/Server/ExternalFilters/[all[tiddlers]] -[[$:/isEncrypted]] -[prefix[$:/temp/]] -[prefix[$:/status/]] -[[$:/boot/boot.js]] -[[$:/boot/bootprefix.js]] -[[$:/library/sjcl.js]] -[[$:/core]]
text: yes
```

It is also possible to configure the server to accept any filter by creating a tiddler titled *$:/config/Server/AllowAllExternalFilters* with the text "yes". This should not be done for public facing servers.

Response:

- 200 OK

  > `Content-Type: application/json`
  >
  > Body: array of all non-system tiddlers in TiddlyWeb JSON tiddler format

- 403 Forbidden

WebServer API: Get Favicon

2nd October 2018 at 2:41pm

WebServer API

Get the favicon for the wiki from the tiddler $:/favicon.ico

```
GET /favicon.ico
```

Parameters:

- none

Response:

- 200 OK

> `Content-Type: image/x-icon`
>
> Body: image data from the tiddler $:/favicon.ico

WebServer API: Get File

2nd October 2018 at 2:43pm

WebServer API

Get the content of a static file. See using the integrated static file server

```
GET /files/<pathname>
```

Parameters:

- **pathname** - URI encoded path to the file

Response:

- 200 OK

> `Content-Type: <content-type>` (determined from file extension)

> Body: data retrieved from file

- 403 Forbidden
- 404 Not Found

WebServer API: Get Rendered Tiddler

2nd October 2018 at 3:54pm

WebServer API

Gets a rendering of the specified tiddler. See using the read-only single tiddler view for more details.

```
GET /{title}
```

Parameters:

- **title** - URI encoded title of the tiddler to render

Response:

- 200 OK

  > `Content-Type: <content-type>`
  >
  > Body: tiddler rendering

- 404 Not Found

WebServer API: Get Server Status

2nd October 2018 at 2:58pm

WebServer API

Get server status information

```
GET /status
```

Parameters:

- none

Response:

- 200 OK

  > `Content-Type: application/json`
  >
  > Body: see below

The JSON data returned comprises the following properties:

- **username** - the username of the currently authenticated user. If undefined, the WebServer Parameter: anon-username is returned instead
- **anonymous** - true if the current user is anonymous
- **read_only** - true if the current user is restricted to read only access to the server
- **logout_is_available** - true if the server supports logging out (optional, defaults to true)

- **space** - always contains the object `{recipe: "default"}`
- **tiddlywiki_version** - the current TiddlyWiki version

For example:

```
{
  "username": "",
  "anonymous": true,
  "read_only": false,
  "space": {
    "recipe": "default"
  },
  "tiddlywiki_version": "5.1.18"
}
```

WebServer API: Get Tiddler

2nd October 2018 at 3:04pm

WebServer API

Gets the raw fields of a tiddler

```
GET /recipes/default/tiddlers/{title}
```

Parameters:

- **title** - URI encoded title of the tiddler to retrieve

Response:

- 200 OK

  > `Content-Type: application/json`
  >
  > Body: tiddler in TiddlyWeb JSON tiddler format

- 404 Not Found

WebServer API: Get Wiki

2nd October 2018 at 2:47pm

WebServer API

Get the main wiki

```
GET /
```

The wiki is composed by rendering the tiddler identified in the root-tiddler parameter with the render type in the root-render-type parameter. This is then served with the content type from the root-serve-type parameter.

Parameters:

- none

Response:

- 200 OK

> `Content-Type: text/html`
>
> Body: data retrieved from file

WebServer API: Put Tiddler

18th April 2021 at 12:07pm

WebServer API

Saves the raw fields of a tiddler

```
PUT /recipes/default/tiddlers/{title}
```

The body should be in TiddlyWeb JSON tiddler format.

Parameters:

- **title** - URI encoded title of the tiddler to save

Headers:

- **x-requested-with** - must be set to `TiddlyWiki` in order for the request to succeed, unless WebServer Parameter: csrf-disable is set to `yes`

Response:

- 204 No Content

```
Content-Type: text/plain

Etag: "default/<title>/<changecount>:"
```

WebServer Authentication

1st July 2018 at 8:45pm

WebServer

Authentication is the process of identifying the current user. TiddlyWiki supports three types of authentication:

- Anonymous Access allows any user to access resources without requiring authentication. Optionally, a username can still be specified for signing edits
- Basic Authentication requires the user to enter a username and password combination which TiddlyWiki validates against an internal database of credentials
- Header Authentication requires an external proxy to place the username of the current user in a trusted header of the request. It is often used as the basis of "single sign-on" features

WebServer Authorization

22nd November 2021 at 1:41am

WebServer

Authorization is the process of determining which resources may be accessed by a particular user. It occurs after authentication has determined the identity of the user. TiddlyWiki's WebServer implements a simple authorization scheme which permits independent control of who has administrator access to the server, and read and write access to a wiki.

The WebServer parameters admin, readers and writers each contain a comma separated list of *principals* (which is to say, either usernames or certain special tokens) which should have read or write access respectively.

The available special tokens are:

- **(anon)** - indicates all anonymous users
- **(authenticated)** - indicates all authenticated users

Admin Functions



> The **(anon)** token is not valid for the admin parameter.

At this time, no server functions are restricted to **admin** authorized users in the unmodified Tiddlywiki server. Third party plugins can leverage this to restrict routes or commands to a subset of authorized users.

Read-only Mode

Read-only mode is engaged when the current user is not authorized to write to the current wiki.

User interface features concerned with creating or editing content are disabled in read-only mode:

- **clone**, **delete**, **new-here** and **new-journal-here** tiddler toolbar buttons

- **import**, **manager**, **new-tiddler new-image** and **new-journal** page control buttons

The tiddler *$:/status/IsReadOnly* is set to `yes` when read-only mode is engaged.

Examples

These example use the credentials parameter to specify the location of a file containing usernames and passwords.

In the first example, read access is permitted for the users "joe" and "mary", with write access restricted to "mary":

```
tiddlywiki mywikifolder --listen credentials=myusers.csv readers=joe,mary writers=mary
```

In the following example, read access is granted to all authenticated users, but only "mary" is granted write access:

```
tiddlywiki mywikifolder --listen credentials=myusers.csv "readers=(authenticated)" writers=mary
```

In the following example, read and write access is granted to all authenticated users, but only "mary" is granted admin access:

```
tiddlywiki mywikifolder --listen credentials=myusers.csv "readers=(authenticated)"
"writers=(authenticated)" admin=mary
```

WebServer Basic Authentication

16th December 2018 at 7:19pm

WebServer Authentication

Basic authentication is a standard mechanism for servers to instruct browsers to prompt the user for credentials. It is recommended to use it in

association with HTTPS due to the way that it passes unencrypted passwords over the network.

Basic authentication is activated if credentials are specified via the username/password or credentials parameters.

If WebServer Authorization is configured to allow access by both anonymous and authenticated users then by default users will not be prompted for credentials, and will be given anonymous access. To force a password prompt visit the route `/login-basic` (for example, http://127.0.0.1:8080/login-basic).

WebServer Guides

3rd July 2018 at 11:47am

WebServer

Further information on usage of the integrated WebServer:

- TiddlyWeb JSON tiddler format
- Using HTTPS
- Using the external JavaScript template
- Using the integrated static file server
- Using the read-only single tiddler view

WebServer Header Authentication

2nd July 2018 at 4:02pm

WebServer Authentication

Header authentication is a web integration technique enabling external entities to securely pass details of the authenticated user to an application.

Header authentication is activated if is configured via the authenticated-user-header

Usage in SSO

Header authentication is commonly used for "single sign on" in corporate environments. When doing header authentication, the user is not prompted for a username and password on TiddlyWiki. Instead, the user is required to login at a SSO proxy server. When the user authenticates themselves to the SSO proxy server, the proxy server redirects the user request to the TiddlyWiki server with this additional request header containing the username. Then TiddlyWiki server is able to use the value of this request header to identify the user.

WebServer Parameter: admin

11th November 2021 at 3:31am

WebServer Parameters

ⓘ **Introduced in v5.2.2** The web server configuration parameter **admin** is used to specify the security principals with administrator access to the WebServer. Does not accept the **(anon)** special token. See WebServer Authorization for more details.

WebServer Parameter: anon-username

2nd July 2018 at 2:48pm

WebServer Parameters

The web server configuration parameter **anon-username** provides an optional username for signing edits from anonymous users.

Without this parameter, anonymous users will be given a blank username.

WebServer Parameter: authenticated-user-header

22nd May 2023 at 8:44pm

WebServer Parameters

The web server configuration parameter **authenticated-user-header** activates header authentication by specifying the name of the HTTP header field that will be used to pass the username to TiddlyWiki.

For example, if the **authenticated-user-header** is set to **X-Authenticated-User**, then the HTTP request must include a header field **X-Authenticated-User** with a value that is the username:

```
X-Authenticated-User: JeremyRuston
```

ⓘ **New in v5.3.0** Value of this header field should be URI-encoded before transit on the client (using `encodeURIComponent` JS function or encodeuricomponent Operator), and will be URI-decoded by the server.

WebServer Parameter: credentials

2nd July 2018 at 5:44pm

WebServer Parameters

The web server configuration parameter **credentials** contains the pathname of a CSV file containing a list of username/password combinations. Using the **credentials** parameter activates WebServer Basic Authentication.

The CSV file must contain a header row and columns labelled **username** and **password**. For example:

```
username,password
jane,do3
andy,sm1th
roger,m00re
```

Notes:

- The optional username/password parameters may be used to provide an additional single set of credentials
- The pathname is taken relative to the wiki folder
- Passwords cannot contain the comma character `,`
- The header row must be present

WebServer Parameter: csrf-disable

19th April 2019 at 7:13pm

WebServer Parameters

The web server configuration parameter **csrf-disable** causes the usual cross-site request forgery checks to be disabled. This might be necessary in unusual or experimental configurations.

Setting **csrf-disable** to `yes` disables the CSRF checks; `no` (or any other value) enables them.

The only currently implemented check is the use of a custom header called `x-requested-with` that must contain the string `TiddlyWiki` in order for write requests to succeed.

WebServer Parameter: debug-level

2nd July 2018 at 4:21pm

WebServer Parameters

The web server configuration parameter **debug-level** determines the level of debugging information printed to the console:

- **full** - maximum logging
- **none** - no logging

WebServer Parameter: host

7th November 2018 at 7:31pm

WebServer Parameters

The web server configuration parameter **host** is the IP address on which the server listens. The most common settings are:

- **127.0.0.1** (default) - only listens for connections from browsers on the same computer
- **0.0.0.0** - listens for connections on all network interfaces, and thus from any browser on a reachable network
- **n.n.n.n** - listens for connections on the network interface with the specified IP address

Note: Using **0.0.0.0** or **n.n.n.n** in a public environment (e.g. coffee shop, library, airport) is inadvisable as it **will** expose your system to possible intrusion.

WebServer Parameter: password

30th June 2018 at 7:11pm

WebServer Parameters

The web server configuration parameter **password**, is used with its companion username as a shortcut for setting user credentials for WebServer Basic Authentication.

WebServer Parameter: path-prefix

27th October 2020 at 1:29pm

WebServer Parameters

The web server configuration parameter **path-prefix** can be used to set an optional prefix for all paths served.

This example causes the server to serve from http://127.0.0.1/MyApp instead of the default http://127.0.0.1/.

```
tiddlywiki mywikifolder --listen "path-prefix=/MyApp"
```

Notes:

- Further steps are required to configure the client-side components to use the prefix. See Using a custom path prefix with the client-server edition
- If no path prefix is required then the `path-prefix` parameter should be omitted or set to the empty string, and not to `\` as might be expected

WebServer Parameter: port

19th December 2019 at 1:37pm

WebServer Parameters

The web server configuration parameter **port** specifies the TCP port on which the server will listen for connections. The default value is `8080`.

The **port** parameter accepts two types of value:

- Numerical values are interpreted as a decimal port number
  - The special value 0 (zero) causes the operating system to assign an available port
- Non-numeric values are interpreted as an environment variable from which the port should be read

This example configures the server to listen on port 8090:

```
tiddlywiki mywikifolder --listen port=8090
```

This example configures the server to listen on the port specified in the environment variable `THE_PORT`:

```
tiddlywiki mywikifolder --listen port=THE_PORT
```

WebServer Parameter: readers

2nd July 2018 at 5:51pm

WebServer Parameters

The web server configuration parameter **readers** is used to specify the security principals with read access to the wiki. See WebServer Authorization for more details.

WebServer Parameter: required-plugins

26th December 2021 at 5:07pm

WebServer Parameters

Introduced in v5.2.2 The web server configuration parameter **required-plugins** is used to specify the plugins required to start the WebServer. It take a comma seperated list of plugin titles. The WebServer will issue a warnign in the console if the required plugins are not loaded. This parameter defaults to `$:/plugins/tiddlywiki/filesystem,$:/plugins/tiddlywiki/tiddlyweb`.

WebServer Parameter: root-render-type

2nd July 2018 at 6:04pm

WebServer Parameters

The web server configuration parameter **root-render-type** determines the way that the root wiki tiddler is rendered:

- `text/plain` (default) – the plain text content of the output is rendered (i.e. HTML elements are ignored)
- `text/html` – the full HTML content of the output is rendered (i.e. including HTML elements)

WebServer Parameter: root-serve-type

2nd July 2018 at 6:05pm

WebServer Parameters

The web server configuration parameter **root-serve-type** determines the content type with which the root wiki tiddler is rendered. The default is `text/html`.

WebServer Parameter: root-tiddler

2nd July 2018 at 6:06pm

WebServer Parameters

The web server configuration parameter **root-tiddler** determines the title of the tiddler that is rendered as the root wiki. The default setting is `$:/core/save/all`.

WebServer Parameter: system-tiddler-render-template

10th August 2018 at 7:31am

WebServer Parameters

The web server configuration parameter **system-tiddler-render-template** is used to specify the template for serving system tiddlers in the read-only single tiddler view. The default value is `$:/core/templates/wikified-tiddler` which renders the tiddler raw, without any special viewing template.



This setting may be overwritten by specifying the `_render_template` field of a tiddler.

WebServer Parameter: system-tiddler-render-type

10th August 2018 at 7:31am

WebServer Parameters

The web server configuration parameter **system-tiddler-render-type** is used to specify the render type for serving system tiddlers in the read-only single tiddler view. The default value is `text/plain`, causing the raw text of rendered system tiddlers to be returned. Alternatively,

`text/html` can be used to cause the full HTML of the rendered tiddlers to be returned.



This setting may be overwritten by specifying the `_render_type` field of a tiddler.

WebServer Parameter: tiddler-render-template

10th August 2018 at 7:32am

WebServer Parameters

The web server configuration parameter **tiddler-render-template** is used to specify the template for serving ordinary, non-system tiddlers in the read-only single tiddler view. The default value is `$:/core/templates/server/static.tiddler.html` which renders tiddlers in a lightweight page with a simple sidebar.



This setting may be overwritten by specifying the `_render_template` field of a tiddler.

WebServer Parameter: tiddler-render-type

10th August 2018 at 7:31am

WebServer Parameters

The web server configuration parameter **tiddler-render-type** is used to specify the render type for serving ordinary, non-system tiddlers in the read-only single tiddler view. The default value is `text/html`, causing the full HTML of the rendered output to be returned. Alternatively, `text/plain` can be used to cause the raw text of rendered system tiddlers to be returned.



This setting may be overwritten by specifying the `_render_type` field of a tiddler.

WebServer Parameter: tls-cert

3rd July 2018 at 12:06pm

WebServer Parameters

The optional web server configuration parameter **tls-cert** contains the pathname to the certificate file required when running the web server under HTTPS. The pathname is taken relative to the wiki folder.

See Using HTTPS for details.

WebServer Parameter: tls-key

3rd July 2018 at 12:06pm

WebServer Parameters

The optional web server configuration parameter **tls-key** contains the pathname to the key file required when running the web server under HTTPS. The pathname is taken relative to the wiki folder.

See Using HTTPS for details.

WebServer Parameter: tls-passphrase

11th November 2021 at 3:21am

WebServer Parameters

ⓘ **Introduced in v5.2.2** The optional web server configuration parameter **tls-passphrase** contains the "certificate passphrase", a string used to decrypt the certificate file used when running the web server under HTTPS.

See Using HTTPS for details.

WebServer Parameter: username

2nd July 2018 at 2:46pm

WebServer Parameters

The web server configuration parameter **username**, in conjunction with its companion password:

- Enables Basic Authentication with the specified username/password combination being added to any credentials specified with the credentials parameter
- The specified username is used as a default value for the readers and writers authorisation parameters if they are not specified

Examples

Serve anonymous users, setting the username to "joe":

```
tiddlywik mywikifolder --listen anon-username=joe
```

Restrict access to the user "joe" with a password of "secret":

```
tiddlywik mywikifolder --listen username=joe password=secret
```

WebServer Parameter: writers

2nd July 2018 at 5:51pm

WebServer Parameters

The web server configuration parameter **writers** is used to specify the security principals with write access to the wiki. See WebServer Authorization for more details.

WebServer Parameters

30th June 2018 at 7:09pm

WebServer

The following configuration parameters are supported by the integrated WebServer:

- admin
- anon-username
- authenticated-user-header
- credentials
- csrf-disable
- debug-level
- host
- password
- path-prefix
- port
- readers
- required-plugins
- root-render-type

- root-serve-type
- root-tiddler
- system-tiddler-render-template
- system-tiddler-render-type
- tiddler-render-template
- tiddler-render-type
- tls-cert
- tls-key
- tls-passphrase
- username
- writers

Wednesday

16th November 2021 at 11:11pm

Operator Examples

This example tiddler is used to illustrate some of the Filter Operators.

What happened to the original TiddlyWiki?

4th September 2014 at 10:51am

HelloThere

The original Classic version of TiddlyWiki is still available at:

https://classic.tiddlywiki.com.



Note that the current version of TiddlyWiki is not fully backwards compatible with TiddlyWikiClassic. Content can be imported but will need massaging to adapt to the new WikiText format. A **tw2parser** plugin is under development that will allow faithful display of most content created for TiddlyWikiClassic:

https://tiddlywiki.com/plugins/tiddlywiki/tw2parser/

Widdly by Opennota

6th January 2021 at 4:10pm

Other Resources

A cross platform server application that can save tiddlers to a local database

https://gitlab.com/opennota/widdly

> a minimal self-hosted app, written in Go, that can serve as a backend for a personal TiddlyWiki.

Widget Attributes

31st July 2023 at 11:06pm

WikiText

Attributes of HTML elements and widgets can be specified in several different ways:

- a literal string

- a transclusion of a textReference
- a transclusion of a macro/variable
- as the result of a filter expression
- **New in v5.3.0** as the result of performing filter and variable substitutions on the given string

attribute type	syntax
literal	single, double or triple quotes or no quotes for values without spaces
transcluded	double curly braces around a text reference
variable	double angle brackets around a macro or variable invocation
filtered	triple curly braces around a filter expression
substituted	single or triple backticks around the text to be processed for substitutions

Literal Attribute Values

Literal attribute values can use several different styles of quoting:

- Single quotes (eg `attr='value'`)
- Double quotes (eg `attr="value"`)
- Tripe double quotes (eg `attr="""value"""`)
- No quoting is necessary for values that do not contain spaces (eg `attr=value`)

Literal attribute values can include line breaks. For example:

```
<div data-address="Mouse House,
```

```
Mouse Lane,
Rodentville,
Ratland."/>
```

By using triple-double quotes you can specify attribute values that contain single double quotes. For example:

```
<div data-address="""Mouse House,
"Mouse" Lane,
Rodentville,
Ratland."""/>
```

Transcluded Attribute Values

Transcluded attribute values are indicated with double curly braces around a TextReference. For example:

```
attr={{tiddler}}
attr={{!!field}}
attr={{tiddler!!field}}
```



The value of the attribute value will be the exact text retrieved from the TextReference. Any wiki syntax in that text will be left as-is.

Variable Attribute Values

Variable attribute values are indicated with double angle brackets around a macro invocation. For example:

```
<div title=<<MyMacro "Brian">>>
...
</div>
```

The behaviour of variables invoked via widget attributes is not the same as when they are invoked via normal wikitext. In addition, the behaviour depends on how the variable is declared:

how declared	behaviour
\define	Textual substitution of parameters is performed on the body text. No further processing takes place. The result after textual substitution is used as the attribute's value
$set, **$let**, **$vars**, \procedure, \widget	Body text is retrieved as-is and used as the attribute's value.
\function	When a function is invoked as `<div class=<<macro>>/>`, it is a synonym for `<div class={{{[function[macro]]}}}/>`. As with any filtered transclusion (i.e. triple curly braces), all results except the first are discarded. That first result is used as the attribute's value. Note that functions are recursively processed even when invoked in this form. In other words a filter expression in a function can invoke another function and the processing will continue

Filtered Attribute Values

Filtered attribute values are indicated with triple curly braces around a Filter Expression. The value will be the first item in the resulting list, or the

empty string if the list is empty.

Introduced in v5.2.2 To improve readability, newlines can be included anywhere that whitespace is allowed within filtered attributes.

This example shows how to add a prefix to a value:

```
<$text text={{{ [<currentTiddler>addprefix[$:/myprefix/]] }}} />
```



The value of the attribute will be the exact text from the first item in the resulting list. Any wiki syntax in that text will be left as-is.

Substituted Attribute Values

New in v5.3.0

Substituted attribute values can use two different styles of quoting:

- Single backticks

  ```
  attr=`value`
  ```

- Triple backticks

  ```
  attr=```value```
  ```

The value of the attribute will be the text denoted by the backticks with any of the placeholders for filter expressions and variables substituted with their corresponding values. Filter expression placeholders are substituted before variable placeholders, allowing for further variable

substitution in their returned value.



Any other wiki syntax in that text will be left as-is.

placeholder syntax	description
`$(varname)$`	Text substitution of a variable. Undefined variables are replaced with an empty string.
`${ filter expression }$`	Text substitution with the first result of evaluating the filter expression.

Examples

Substituting a variable value into a string

copy to clipboard

```
<$text text=`Hello there this is the tiddler "$(currentTiddler)$"`/>
```

That renders as:

Hello there this is the tiddler "Substituted Attribute Values"

Substituting a variable value and the result of evaluating a filter expression into a string

copy to clipboard

```
<$text text=`This tiddler is titled "$(currentTiddler)$" and was last modified on ${[{!!modified}
```

```
format:date[DDth MMM YYYY]]}$`/>
```

That renders as:

This tiddler is titled "Substituted Attribute Values" and was last modified on 15th June 2023

Concatenating strings and variables to create a URL

copy to clipboard

```
<$let hash={{{ [<currentTiddler>encodeuricomponent[]] }}}>
<a href=`http://tiddlywiki.com/#$(hash)$`>this tiddler on tiddlywiki.com</a>
</$let>
```

That renders as:

this tiddler on tiddlywiki.com

Concatenating variables and a text reference to create a URL

copy to clipboard

```
<$let hash={{{ [<currentTiddler>encodeuricomponent[]] }}}>
```

```
<a href=`${ [{!!base-url}] }$#$(hash)$`>this tiddler on tiddlywiki.com</a>
</$let>
```

That renders as:

this tiddler on tiddlywiki.com

Widget Documentation Style Guide

25th March 2023 at 5:25pm

Reference Tiddlers

Widget documentation generally consists of

- an introductory summary of the widget's function
- an overview of attributes and how the widget content is handled
- more detailed attribute explanation when required
- examples

The *Content and Attributes* section should describe how the content of the widget is used, followed by a table of the possible attributes, each with a short description of a single sentence. The attribute names in the table and elsewhere in the tiddler should be rendered with the **`.attr`** macro. The **`.from-version`** macro should be used as first item in the description to designate the version at which a parameter became available.

A subsection should be used for parameters that need an *extended description*, possibly with separate examples. When there are more than 2

such subsection necessary, the subsections should be split into their own tiddlers, and the **.doc-tabs** macro should be used to transclude them into the widget documentation tiddler as tabs.

- The **.doc-tabs** macro is used without parameters
- The **.doc-tabs** macro must not be used more than once within a single documentation tiddler
- The tiddlers to be included in the tab group should be tagged with the respective widget documentation tiddler and have a **description** field set to **tab**
- The **caption** contains the tab button text. It should be as short as possible. For the attribute name, the **.attr** macro should be used in the **caption**
- The tab tiddlers should start with a level-2-heading that links to itself, so that they may be opened from within the tabs
- They can contain examples if these are specific to the topic of the subsection

In the attribute table, the **.widget-attr-link** macro can be used to turn attributes into a button that activates the respective tab. The **.widget-attr-link** macro is used like this

```
<<.widget-attr-link text:<display text> target:<title of corresponding tiddler> >>
```

Elsewhere, the **.doc-tab-link** macro can be used to activate the respective tab

```
<<.doc-tab-link text:<display text> target:<title of corresponding tiddler> tooltip:<tooltip to show when hovering over the button> class:<additional classes> >>
```

The parameters **tooltip** and **class** are optional.

Both link macros scroll to the tab group additionally to setting the selected tab. They can also be used within the tab contents tiddlers themselves. If a tab tiddler is opened outside the tab group, **doc-tab-link** will open the tiddler containing the tab group if it is not yet open,

and scroll to its tab group if it is.

WidgetMessage: tm-add-field

8th September 2014 at 8:51pm

Messages

The `tm-add-field` message is handled by the FieldManglerWidget. It adds the specified field with a blank value if the field doesn't already exist.

Name	Description
param	Name of field to add

The add field message is usually generated with the ButtonWidget, and is handled by the FieldManglerWidget.

WidgetMessage: tm-add-tag

8th September 2014 at 8:51pm

Messages

The `tm-add-tag` message is handled by the FieldManglerWidget. It adds the specified tag.

Name	Description
param	Name of tag to add

The add tag message is usually generated with the ButtonWidget, and is handled by the FieldManglerWidget.

WidgetMessage: tm-auto-save-wiki

30th August 2014 at 1:51pm

Messages

The autosave wiki message causes the current saver module to perform a background save if it is required.

The autosave wiki message should be generated whenever changes are made to the store. For example, the navigator widget generates the autosave wiki message as part of its handling of the WidgetMessage: tm-save-tiddler, WidgetMessage: tm-delete-tiddler and WidgetMessage: tm-perform-import.

The autosave wiki message is handled by the TiddlyWiki core *SyncMechanism* which invokes the current *SaverModule*. Not all *SaverModules* can handle autosaving.

WidgetMessage: tm-browser-refresh

26th August 2014 at 1:05pm

Messages

The `tm-browser-refresh` message refreshes the page, causing the re-initialisation of any plugin tiddlers. It does not require any properties on the `event` object.

The refresh message is usually generated with the ButtonWidget and is handled by the core.

WidgetMessage: tm-cancel-tiddler

26th February 2014 at 8:39pm

Messages navigator-message

The `tm-cancel-tiddler` message abandons the changes in a draft tiddler. It requires the following properties on the `event` object:

Name	Description
param	Title of the tiddler that is being cancelled out of edit mode
tiddlerTitle	Fallback title that is used if **param** isn't specified (automatically set by the ButtonWidget)

The cancel tiddler message is usually generated with the ButtonWidget and is handled by the NavigatorWidget.

WidgetMessage: tm-clear-password

26th February 2014 at 9:49am

Messages

The `tm-clear-password` message clears the current password from the password vault, clearing the $:/isEncrypted tiddler. See *EncryptionMechanism* for details.

This message is typically generated with the ButtonWidget, and is handled by the core itself.

WidgetMessage: tm-close-all-tiddlers

26th February 2014 at 8:43pm

Messages navigator-message

The close all tiddlers message empties the story list.

The close all tiddlers message is usually generated with the ButtonWidget and is handled by the NavigatorWidget.

WidgetMessage: tm-close-all-windows

1st March 2022 at 7:08pm

Messages

ⓘ Introduced in v5.2.2

The `tm-close-all-windows` message closes all additional *browser* window that were opened with tm-open-window.

The `tm-close-window` message is best generated with the ActionSendMessageWidget, which in turn is triggered by a widget such as the ButtonWidget. It is handled by the core itself.

copy to clipboard

```
<$button>Close All Windows
<$action-sendmessage
  $message="tm-close-all-windows"
/>
</$button>
```

That renders as:

Close All Windows

WidgetMessage: tm-close-other-tiddlers

2nd March 2014 at 7:33pm

Messages navigator-message

The `tm-close-other-tiddlers` message removes all but a specified tiddler from the story list. It requires the following properties on the `event` object:

Name	Description
param	Title of the tiddler that is not to be closed
tiddlerTitle	Fallback title that is used if **param** isn't specified (automatically set by the ButtonWidget)

The close other tiddlers message is usually generated with the ButtonWidget and is handled by the NavigatorWidget.

WidgetMessage: tm-close-tiddler

26th February 2014 at 8:42pm

Messages navigator-message

The `tm-close-tiddler` message removes a specified tiddler from the story list. It requires the following properties on the `event` object:

Name	Description
param	Title of the tiddler that is to be closed
tiddlerTitle	Fallback title that is used if **param** isn't specified (automatically set by the ButtonWidget)

The close tiddler message is usually generated with the ButtonWidget and is handled by the NavigatorWidget.

WidgetMessage: tm-close-window

24th July 2023 at 12:05am

Messages

Introduced in v5.2.2

The `tm-close-window` message closes an additional *browser* window that was opened with tm-open-window. Specify which window to close by setting the value of **`param`** to the string used as **`windowID`** when opening the window.

Name	Description
param *{default param}*	String used as **`windowID`** when opening the window

The `tm-close-window` message is best generated with the ActionSendMessageWidget, which in turn is triggered by a widget such as the ButtonWidget. It is handled by the core itself.

When used with the ActionSendMessageWidget, **`param`** becomes **`$param`**

To close all additional browser windows that were opened with tm-open-window use WidgetMessage: tm-close-all-windows

If the **`windowID`** parameter was used with tm-open-window when opening a new window, the value of **`windowID`** is available within that window in the variable **`tv-window-id`**

copy to clipboard

```
<$button>Open Window
<$action-sendmessage
```

```
  $message="tm-open-window"
  $param="$:/temp/openme"
  template="SampleWindowTemplate"
  windowTitle="My Window Title"
  width="400"
  height="500"
  windowID="window1"
  something="I just flew in on a variable, and boy is my Hashmap tired." />
</$button>
<$button>Close Window
<$action-sendmessage
  $message="tm-close-window"
  $param="window1"
/>
</$button>
```

That renders as:

Open Window Close Window

WidgetMessage: tm-copy-to-clipboard

15th December 2017 at 4:06pm

Messages

The "copy to clipboard" message attempts to copy the specified text to the clipboard. If it succeeds, the tiddler **$:/language/Notifications/CopiedToClipboard/Succeeded** is displayed as a notification. If the browser doesn't permit the operation, the tiddler **$:/language/Notifications/CopiedToClipboard/Failed** is displayed instead.

It requires the following properties on the `event` object:

Name	Description
param	Text to be copied to the clipboard

This message is usually generated with the ButtonWidget. It is handled by the TiddlyWiki core.

Example

This example copies the current time to the clipboard:

copy to clipboard

```
<$button message="tm-copy-to-clipboard" param=<<now>>>
Copy date to clipboard
</$button>
```

That renders as:

Copy date to clipboard

WidgetMessage: tm-delete-tiddler

26th February 2014 at 10:04am

Messages navigator-message

The `tm-delete-tiddler` message deletes the specified tiddler and removes it from the current story. If the tiddler is a draft then it also deletes the tiddler specified in the `draft.of` field. The delete tiddler message requires the following properties on the `event` object:

Name	Description
param	Title of the tiddler that is to be deleted
tiddlerTitle	Fallback title that is used if **param** isn't specified (automatically set by the ButtonWidget)

The delete tiddler message is usually generated with the ButtonWidget and is handled by the NavigatorWidget.

Use the ActionDeleteTiddlerWidget to delete a named tiddler without getting the "Do you wish to delete the tiddler" prompt.

WidgetMessage: tm-download-file

23rd July 2023 at 11:47pm

Messages

The download file message causes the current saver module to prompt the user to download the result of parsing a specified template tiddler as a file. It requires the following properties on the `event` object:

Name	Description
param	Title of a tiddler to use as a template for the new tiddler

{any other params}	Any other parameters are made available as variables within the context of the widget message.

The following variable names have special behaviour:

Name	Description
filename	Filename for the downloaded file (note that this is a hint to the browser, and the actual filename used may be different)

The download file message is usually generated with the ButtonWidget.

The download file message is handled by the TiddlyWiki core *SyncMechanism* which invokes the current *SaverModule*.

WidgetMessage: tm-edit-bitmap-operation

23rd July 2023 at 11:47pm

Messages

A `tm-edit-bitmap-operation` invokes one of the available operations on a surrounding bitmap editor. Therefore the message has to be dispatched within the editor in order for it to catch it. The following properties on the `event` object are required:

Name	Description
param	Name of the operation to be executed, see **below** for a list of possible operations
{any other params}	Any other parameters are made available as variables within the context of the widget message.

The `tm-edit-bitmap-operation` message is usually generated by a ButtonWidget or an *ActionWidget* and is handled by the surrounding bitmap editor.

Bitmap Operations

At this point the following bitmap operations have been implemented:

Name	Description
resize	Resizes the image to the specified width and height. Parameters include: **Name** / **Description** width — Specifies the width the image is resized to height — Specifies the height the image is resized to
clear	Clears the contents of the image and fills it with a solid colour. Parameters include: **Name** / **Description** colour — Colour the image should be filled with, defaults to **`White`** The colour field can take any normal CSS colour value, including the hexadecimal representation or the RGB format.

Example

An example can be seen in **$:/core/ui/EditorToolbar/size-dropdown**:

```
<$button>
<$action-sendmessage
    $message="tm-edit-bitmap-operation"
    $param="resize"
    width={{$config-title$/new-width}}
    height={{$config-title$/new-height}}
/>
...
Resize
</$button>
```

WidgetMessage: tm-edit-text-operation

23rd July 2023 at 11:46pm

Messages

A `tm-edit-text-operation` invokes one of the available operations on a surrounding text editor. Therefore the message has to be dispatched within the editor in order for it to catch it. The following properties on the `event` object are required:

Name	Description
param	Name of the operation to be executed, see **below** for a list of possible operations
{any other params}	Any other parameters are made available as variables within the context of the widget message.

The `tm-edit-text-operation` message is usually generated by a ButtonWidget or an *ActionWidget* and is handled by the surrounding text editor.

Text Operations

At this point the following text operations have been implemented:

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>excise</td><td>Excises the currently selected text into a new tiddler and replaces it with a link, a macro or a transclude of the new tiddler. Parameters include:

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>title</td><td>Title of the new tiddler the selected content is excised to</td></tr>
<tr><td>type</td><td>Type of the replacement to be inserted: Can be one of <code>transclude</code>, <code>link</code> or <code>macro</code></td></tr>
<tr><td>macro</td><td>In case type=<code>macro</code>, specifies the name of the macro to be inserted. The title of the new tiddler is provided as the first parameter to the macro. Defaults to the translink macro</td></tr>
<tr><td>tagnew</td><td>If '<code>yes</code>', will tag the new tiddler with the title of the tiddler currently being edited</td></tr>
</table></td></tr>
<tr><td>replace-all</td><td>Replaces all contents of the editor with the provided text.

<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>text</td><td>Text to be inserted</td></tr>
</table></td></tr>
<tr><td>replace-selection</td><td>Replaces the current selection with the provided text.

<table>
<tr><th>Name</th><th>Description</th></tr>
</table></td></tr>
</table>

	text: Text to be inserted
prefix-lines	Prefixes the currently selected line*(s)* with the provided character. If a line is already prefixed by the provided prefix, the prefix is removed instead. **Name**: **Description** character: Prefix character count: Number of characters that make up the prefix **Example** Setting *character="**!**"* and *count="**3**"* would insert the prefix "**!!!**", which will resolve to a subheading when parsed as WikiText.
wrap-lines	Surrounds the selected *lines* with the provided **prefix** and **suffix**. **Name**: **Description** prefix: String to be prefixed to the selected lines suffix: Suffix to be inserted after the selected lines
wrap-selection	Surrounds the current *selection* with the provided **prefix** and **suffix**. **Name**: **Description** prefix: String to be prefixed to the selection suffix: Suffix to be inserted after the selection

save-selection	Saves the text of the current *selection* into the provided **`tiddler`** and **`field`**. **Name** / **Description** tiddler / Tiddler title to which the selection will be saved field / Field name (defaults to "text")
make-link	Creates a wiki text link to the tiddler specified in **`text`**. If there is a selection, it is used as the text of the link. **Name** / **Description** text / Tiddler title to which the link will be created
insert-text	Inserts the text specified in **`text`** at the caret position. If there is a selection, it is replaced. **Name** / **Description** text / Text to be inserted
focus-editor	ⓘ **Introduced in v5.2.0** Simply focuses the Text Editor

Example

An example can be seen in **$:/core/ui/EditorToolbar/bold**:

```
<$action-sendmessage
```

```
    $message="tm-edit-text-operation"
    $param="wrap-selection"
    prefix="'''"
    suffix="'''"
/>
```

WidgetMessage: tm-edit-tiddler

26th February 2014 at 10:06am

Messages navigator-message

The `tm-edit-tiddler` message replaces the specified tiddler in the current story with a draft version of itself. It requires the following properties on the `event` object:

Name	Description
param	Title of the tiddler that is being switched to edit mode
tiddlerTitle	Fallback title that is used if **param** isn't specified (automatically set by the ButtonWidget)

The edit tiddler message is usually generated with the ButtonWidget or the ActionSendMessageWidget and is handled by the NavigatorWidget.

WidgetMessage: tm-focus-selector

23rd July 2023 at 11:51pm

Messages

The `tm-focus-selector` message sets the focus to the DOM element identified by the selector in the `param` parameter.

Name	Description
param	Selector identifying the DOM element to be focussed
{any other params}	Any other parameters to be passed to the focus() method as variables.



Use preventScroll="true" to prevent the browser from scrolling to the focused element

WidgetMessage: tm-fold-all-tiddlers

28th October 2019 at 12:38pm

Messages

The `tm-fold-all-tiddlers` message folds all tiddlers in the current story list.

Internally, it sets the text of all of the state tiddlers corresponding to tiddlers in the story river to "**hide**". The state tiddler titles are formed by prefixing the tiddler title with a prefix that defaults to `$:/state/folded/`.

Name	Description
foldedStatePrefix	Prefix for the state tiddler in which the fold state is stored. Defaults to `$:/state/folded`

The `tm-fold-all-tiddlers` message is usually generated with the ButtonWidget and is handled by the surrounding NavigatorWidget.

WidgetMessage: tm-fold-other-tiddlers

28th October 2019 at 12:39pm

Messages

The `tm-fold-other-tiddlers` message folds all tiddlers in the current story list, except the one specified in the `param` parameter. It does

so by setting the text of a state tiddler to either "**show**" or "**hide**", according to the fold state.

Name	Description
param	Title of the tiddler that should be ignored by the fold operation.
foldedStatePrefix	Prefix for the state tiddler in which the fold state is stored.



The core uses a foldStatePrefix of `$:/state/folded/` to store the fold states for the default story view.



The state tiddlers title is computed as `foldStatePrefix + TiddlerTitle`. If the foldStatePrefix is not set, it will overwrite the text of the tiddler(s) itself, resulting in data loss.

The `tm-fold-other-tiddlers` message is usually generated with the ButtonWidget and is handled by the surrounding NavigatorWidget.

WidgetMessage: tm-fold-tiddler

28th October 2019 at 12:35pm

Messages

The `tm-fold-tiddler` message toggles the value of a state tiddler. It does so by setting the text of a state tiddler to either "**show**" or "**hide**", according to the fold state.

Name	Description
param	ignored ... foldedState must be used
foldedState	State tiddler in which the fold state is stored



The core uses a foldState prefix of `$:/state/folded/` + `tiddler title` to store the fold states for the default story view.

The `tm-fold-tiddler` message is usually generated with the ButtonWidget and is handled by the surrounding NavigatorWidget.

The core **$:/core/ui/ViewTemplate** defines a global variable `folded-state`, that is used with every tiddler. It's created like this:

```
\define folded-state()
$:/state/folded/$(currentTiddler)$
\end
```

WidgetMessage: tm-full-screen

14th August 2018 at 11:51pm

Messages

The fullscreen message is used to enter, exit or toggle the "fullscreen" mode of the browser, if it supports it. It uses the following properties on the `event` object:

Name	Description
param	`enter` to enter full screen mode, `exit` to exit it, otherwise toggle the full screen status

The fullscreen message is handled by the TiddlyWiki core.

copy to clipboard

```
<$button message="tm-full-screen">
Full screen toggle
</$button>
```

```
<$button message="tm-full-screen" param="enter">
Full screen enter
</$button>

<$button message="tm-full-screen" param="exit">
Full screen exit
</$button>
```

That renders as:

Full screen toggle

Full screen enter

Full screen exit

WidgetMessage: tm-home

19th August 2014 at 1:05pm

Messages

The `tm-home` message closes any open tiddlers and re-opens the default tiddlers set in $:/DefaultTiddlers. It also remove any permalink from the browser address bar. It does not require any properties on the `event` object.

The home message is usually generated with the ButtonWidget and is handled by the core.

WidgetMessage: tm-http-cancel-all-requests

29th April 2023 at 6:14pm

Messages

The **tm-http-cancel-all-requests** message is used to cancel all outstanding HTTP requests initiated with WidgetMessage: tm-http-request.

Note that the state tiddler $:/state/http-requests contains a number representing the number of outstanding HTTP requests in progress.

It does not take any parameters.

WidgetMessage: tm-http-request

23rd July 2023 at 11:53pm

Messages

The **tm-http-request** message is used to make an HTTP request to a server.

It uses the following properties on the `event` object:

Name	Description
param	Not used
{any other params}	Any other parameters are made available as variables within the context of the widget message. See below.

The following parameters are used:

Name	Description

method	HTTP method (eg "GET", "POST")
body	String data to be sent with the request
binary	ⓘ **New in v5.3.1** Set to "yes" to cause the response body to be treated as binary data and returned in base64 format
query-*	Query string parameters with string values
header-*	Headers with string values
password-header-*	Headers with values taken from the password store
password-query-*	Query string parameters with values taken from the password store
var-*	Variables to be passed to the completion and progress handlers (without the "var-" prefix)
bind-status	Title of tiddler to which the status of the request ("pending", "complete", "error") should be bound
bind-progress	Title of tiddler to which the progress of the request (0 to 100) should be bound
oncompletion	Action strings to be executed when the request completes
onprogress	Action strings to be executed when progress is reported

The following variables are passed to the completion handler:

Name	**Description**
status	HTTP result status code (see MDN)
statusText	HTTP result status text
error	Error string
data	Returned data
headers	Response headers as a JSON object

The following variables are passed to the progress handler:

Name	**Description**

lengthComputable	Whether the progress loaded and total figures are valid - "yes" or "no"
loaded	Number of bytes loaded so far
total	Total number bytes to be loaded

Note that the state tiddler $:/state/http-requests contains a number representing the number of outstanding HTTP requests in progress.

Examples

- Zotero's API for retrieving reference items: WidgetMessage: tm-http-request Example - Zotero
- Random Dog's API for retrieving random pictures of dogs showing how to retrieve binary data: WidgetMessage: tm-http-request Example - Random Dog

WidgetMessage: tm-http-request Example - Random Dog

$:/tags/Global

Random Dogs

This demo uses the API of the website https://random.dog/ to import a random dog image or video.

Note that the images and videos can be quite large, so this demo is not recommended to be used over mobile data connections.

Import a random dog image or video

WidgetMessage: tm-http-request Example - Zotero

$:/tags/Global

Specify the Zotero group ID to import 4813312 or com216

Start import from Zotero group

Cancel all HTTP requests Outstanding requests: 0

WidgetMessage: tm-import-tiddlers

28th February 2015 at 3:36pm

Messages navigator-message

The `tm-import-tiddlers` message inserts a list of tiddlers into the pending import tiddler *$:/Import*. It also applies any active **upgrader** modules to each tiddler as it arrives (see the UpgradeMechanism for more details).

Name	Description
param	JSON text of the array of tiddlers to be imported
autoOpenOnImport	ⓘ **Introduced in v5.1.23** Optional value "no" or "yes" that can override tv-auto-open-on-import
importTitle	ⓘ **Introduced in v5.1.23** optional tiddler title to use for import process instead of $:/Import

The import tiddlers message is usually generated with the DropzoneWidget or the BrowseWidget, and is handled by the NavigatorWidget.

Configuration Variables

- **tv-auto-open-on-import**

WidgetMessage: tm-load-plugin-from-library

25th April 2016 at 1:55am

Messages

The `tm-load-plugin-from-library` message load the specified tiddler or plugin from the specified plugin library.

Name	Description
title	Title of the tiddler or plugin to be loaded into the current wiki
url	Url specifying the plugin library from which the tiddler or plugin is to be loaded.

Example

The following snippet will install the *CodeMirror* plugin from the official plugin library:

```
<$button>
<$action-sendmessage $message="tm-load-plugin-from-library" title="$:/plugins/tiddlywiki/codemirror"
url="https://tiddlywiki.com/library/v5.1.11/index.html"/>
Install CodeMirror
</$button>
```

Usually a plugin library has to be loaded (initialized) first before it can be used. If the plugin library specified by 'url' has not been initialized yet, it will be loaded and initialized.

The `tm-load-plugin-from-library` message is usually generated with the ButtonWidget and is handled by the core itself.

WidgetMessage: tm-load-plugin-library

25th April 2016 at 2:04am

Messages

The `tm-load-plugin-library` message loads the specified plugin library and imports information about the contained plugins in temporary tiddlers.

Name	Description
url	Url specifying the plugin library to be loaded.
infoTitlePrefix	Prefix to the temporary tiddlers being created, containing information about the plugins the plugin library offers. Defaults to *$:/temp/RemoteAssetInfo/*

Example

The following snippet will load the official plugin library:

```
<$button>
<$action-sendmessage $message="tm-load-plugin-library" infoTitlePrefix="$:/temp/RemoteAssetInfo/"
url="https://tiddlywiki.com/library/v5.1.11/index.html"/>
```

```
Load official plugin library
</$button>
```

The `tm-load-plugin-library` message is usually generated with the ButtonWidget and is handled by the core itself.

WidgetMessage: tm-login

25th October 2020 at 5:31pm

Messages

The login message prompts the user for a username and password and attempts to login to the current serverside host. The tiddler *$:/status/IsLoggedIn* reflects the current login status with the values "yes" or "no", and *$:/status/UserName* reflects the current username.

The login message is handled by the TiddlyWiki core *SyncMechanism* which invokes the current *SyncAdaptorModule* (typically the **tiddlywiki/tiddlywebadaptor** plugin).

ⓘ **Introduced in v5.1.23** The login message can optionally accept parameters called **username** and **password** that immediately attempts to login with the specified credentials without displaying the prompt

WidgetMessage: tm-logout

11th August 2014 at 1:33pm

Messages

The logout message attempts to log the user out of the current serverside host. The tiddler *$:/status/IsLoggedIn* reflects the current login status with the values "yes" or "no", and *$:/status/UserName* reflects the current username.

The logout message is handled by the TiddlyWiki core *SyncMechanism* which invokes the current *SyncAdaptorModule* (typically the **tiddlywiki/tiddlywebadaptor** plugin).

WidgetMessage: tm-modal

23rd July 2023 at 11:54pm

Messages

Name	Description
param	Title of the tiddler to be displayed
{any other params}	Any other parameters are made available as variables within the context of the widget message.
rootwindow	ⓘ **Introduced in v5.1.18** **yes** or **true** will always display a modal in the wiki-root-window

The modal message is usually generated with the ButtonWidget. The modal message is handled by the TiddlyWiki core.

Fields within the tiddler being displayed in the modal can be used to customise its appearance.

paramObject

The "currentTiddler" variable is set to the title of the modal tiddler, but can be overridden by specifying a different value in `paramObject`.

Example

Here is an example of displaying a modal and passing parameters to it: (Note how all parameters to `action-sendmessage` other than `$param` and `$message` are available as variables within the modal.)

copy to clipboard

```
Your name: <$edit-text tiddler="$:/temp/yourName" tag="input" default="Your name"/>

Your message:
<$edit-text tiddler="$:/temp/yourMessage" default="Your message"/>

<$button>
<$action-sendmessage $message="tm-modal" $param="SampleModal" yourName={{$:/temp/yourName}}
yourMessage={{$:/temp/yourMessage}}/>
Click me!
</$button>
```

That renders as:

Your name: Your name

Your message: Your message

Click me!



Introduced in v5.1.18 if triggered from within a **new window**, the above examples will be displayed within that window. The **rootwindow** attribute can be set to **yes** or **true** to inherit this behavior and to display the Modal within the **root** window



Introduced in v5.1.23 Links to tiddlers within a modal will open in the same story as the widget that sent the `tm-modal` message.

> You can override this by specifying values for the variables `tv-story-list` and `tv-history-list` when sending the `tm-modal` message.

WidgetMessage: tm-navigate

5th July 2021 at 11:49am

Messages

The `tm-navigate` message inserts the specified tiddler into the story and puts it at the top of the history stack. If the tiddler is not already present in the story then it will be positioned immediately after the tiddler specified in `event.navigateFromTitle`.

The navigate message requires the following properties on the `event` object:

Name	Description
navigateTo	Title of the tiddler that is being navigated
navigateFromTitle	Title of the tiddler from which the navigation was initiated
navigateFromClientRect	Bounding rectangle in client page coordinates of the element initiating the navigation
navigateFromClientTop	Y coordinate of top edge of bounding client rectangle
navigateFromClientLeft	X coordinate of left edge of bounding client rectangle
navigateFromClientWidth	Width of bounding client rectangle
navigateFromClientRight	X coordinate of right edge of bounding client rectangle
navigateFromClientBottom	Y coordinate of bottom edge of bounding client rectangle
navigateFromClientHeight	Height of bounding client rectangle
navigateSuppressNavigation	**true** causes the new tiddler to only be added to the story, and not the history stack. This suppresses the scrolling associated with navigating to a tiddler

The navigate message can be generated by the LinkWidget, the ActionNavigateWidget and the ButtonWidget, and is handled by the NavigatorWidget.

WidgetMessage: tm-new-tiddler

23rd July 2023 at 11:58pm

Messages navigator-message

The new tiddler message creates a new draft tiddler and adds it to the current story. It requires the following properties on the `event` object:

Name	Description
param	The optional title of a tiddler to use as a template for the new tiddler
{any other params}	The names and values of additional tiddler fields.
navigateFromTitle	Title of the tiddler from which the navigation to the new tiddler was initiated

The title for the draft tiddler is chosen according to these rules:

- If additional parameters were used and a title field was specified, use that title
- If a template tiddler was used, use the title of the template tiddler, making it unique with a numeric suffix
- Otherwise, generate a new title based on the default new tiddler title with a numeric suffix to make it unique

The new tiddler message is usually generated with the ButtonWidget or ActionSendMessageWidget and is handled by the NavigatorWidget.

Examples

To make a button that creates new tiddlers tagged "task", create a tiddler called "TaskTemplate" with that tag, and then make your button like this:

```
<$button message="tm-new-tiddler" param="TaskTemplate">New Task</$button>
```

To create a new tiddler with explicit parameters rather than by cloning a template tiddler:

```
<$button>
<$action-sendmessage $message="tm-new-tiddler" title="This is newly created tiddler" tags="OneTag [[Another Tag]]" text=<<now "Today is DDth, MMM YYYY">>/>
New Tiddler
</$button>
```

To create a new tiddler from a template with additional parameters:

```
<$button>
<$action-sendmessage $message="tm-new-tiddler" $param=<<currentTiddler>> fieldname="field value"/>
New Tiddler
</$button>
```

WidgetMessage: tm-notify

24th July 2023 at 12:07am

Messages

The notify message briefly displays a specified tiddler as a small alert in the upper right corner of the page. It requires the following properties on the `event` object:

Name	Description
param	Title of the tiddler to be displayed
{any other params}	Any other parameters are made available as variables to the notify message.

The notify message is handled by the TiddlyWiki core.

WidgetMessage: tm-open-external-window

24th July 2023 at 12:08am

Messages

ⓘ Introduced in v5.1.14

The `tm-open-external-window` message opens an external link eg: "https://tiddlywiki.com" in a new *browser* window. If no parameters are specified, it opens the help tiddler. Any additional parameters passed via the **`paramObject`** are being provided as variables to the new window.

Name	Description
param	URL of the tiddler to be opened in a new browser window, defaults to the TiddlyWiki help
{any other params}	Any other parameters are made available as variables to the window. See below.

parmObject

Name	Description	Important
windowName	If a parameter is provided it can be used to open different links in the same	The behaviour is influenced by user settings

	window eg: `_tiddlywiki`. Default is empty, so every link opens a new window.	in the browser and the browsers default behavior!
windowFeatures	This parameter needs to be provided as a single string. eg: `"height=400, width=600"`. For detailed description about possible parameters see: Mozilla Help	

The `tm-open-external-window` message is usually generated with the ButtonWidget or ActionSendMessageWidget and is handled by the core itself.

Examples

copy to clipboard

```
<$button>
<$action-sendmessage $message="tm-open-external-window" $param="https://tiddlywiki.com"
windowName="_tiddlywiki" windowFeatures="height=500, width=900"/>
Open ~TiddlyWiki - Action
</$button>

<$button>
<$action-sendmessage $message="tm-open-external-window" $param="https://developer.mozilla.org/en-US/docs/
Web/API/Window/open" windowName="_tiddlywiki" windowFeatures="height=400, width=600"/>
Open Mozilla Help - Action
</$button>

<$button message="tm-open-external-window" param="https://tiddlywiki.com" >
```

```
Open ~TiddlyWiki - Button
</$button>
```

That renders as:

Open TiddlyWiki - Action

Open Mozilla Help - Action

Open TiddlyWiki - Button

WidgetMessage: tm-open-window

31st August 2023 at 10:15pm

Messages

The `tm-open-window` message opens a tiddler in a new *browser* window. If no parameters are specified, the current tiddler is opened in a new window. Similiar to `tm-modal` any additional parameters passed via the **`paramObject`** are provided as variables to the new window.

Name	Description
param *{default param}*	Title of the tiddler to be opened in a new browser window, defaults to **`currentTiddler`** if empty
template	Template in which the tiddler will be rendered
windowTitle	Title string for the opened window
width	Width of the new browser window

height	Height of the new browser window
left	ⓘ **Introduced in v5.2.2** Optional, left position of new browser window
top	ⓘ **Introduced in v5.2.2** Optional, top position of new browser window
windowID	ⓘ **Introduced in v5.2.2** Optional, unique string used to identify the widow. Can be used with WidgetMessage: tm-close-window to close the window. Defaults to the value of **`param`**
{any other params}	Any other parameters are made available as variables within the new window

The `tm-open-window` message is best generated with the ActionSendMessageWidget, which in turn is triggered by a widget such as the ButtonWidget. The message is handled by the core itself.



When used with the ActionSendMessageWidget, **`param`** becomes **`$param`**.
Parameters **`template`**, **`windowTitle`**, **`width`**, **`height`**, **`left`** and **`top`** require the ActionSendMessageWidget.



ⓘ **Introduced in v5.2.2** To close a window opened with tm-open-window use WidgetMessage: tm-close-window
To open a tiddler in more than one new window, use a unique value for **`windowID`**



ⓘ **New in v5.3.2** If the new window is hidden by other windows, clicking the "open" button again will bring it to the foreground and set focus to the new window. This behaviour should be consistent for all browsers now

copy to clipboard

```
<$button>Open Window
<$action-sendmessage
  $message="tm-open-window"
  $param="$:/temp/openme"
```

```
  template="SampleWindowTemplate"
  windowTitle="My Window Title"
  width="640"
  height="480"
  something="This is my new window. There are many like it, but this one is mine." />
</$button>
```

That renders as:

Open Window

WidgetMessage: tm-perform-import

16th July 2014 at 10:43am

Messages navigator-message

The perform import message copies tiddlers from a specified plugin into the main store. See the UpgradeMechanism for an overview of how it is used by the core.

Name	Description
param	Title of the pending import tiddler. Defaults to *$:/Import*

To select which tiddlers are to be imported, fields with names formed from `selection-` plus the title of the tiddler are used. The value **unchecked** causes the tiddler to be skipped from the import.

The perform import message is usually generated with the ButtonWidget and is handled by the NavigatorWidget.

WidgetMessage: tm-permalink

23rd July 2014 at 12:37pm

Messages

The `tm-permalink` message changes the browser address bar to form a permalink to a specified tiddler, defaulting to the current tiddler. The resulting link will be copied to the clipboard.

The permalink message supports the following properties on the `event` object:

Name	Description
param	Title of the tiddler to be permalinked
tiddlerTitle	The current tiddler (used by default if the tiddler title isn't specified in the `param`)

The permalink message can be generated by the ButtonWidget, and is handled by the story mechanism.

WidgetMessage: tm-permaview

23rd July 2014 at 12:37pm

Messages

The `tm-permaview` message changes the browser address bar to form a permaview that specifies all the open tiddlers in the main story river, and the tiddler to be navigated, defaulting to the current tiddler.

The permaview message supports the following properties on the `event` object:

Name	Description
param	Title of the tiddler to be navigated within the permaview
tiddlerTitle	The current tiddler (used by default if the tiddler title isn't specified in the `param`)

The permaview message can be generated by the ButtonWidget, and is handled by the story mechanism.

WidgetMessage: tm-print

8th October 2016 at 10:56am

Messages

Introduced in v5.1.14 The `tm-print` message causes the browser to display the print dialog for the current page. It does not require any properties on the `event` object.

The print message is usually generated with the ButtonWidget and is handled by the core.

WidgetMessage: tm-relink-tiddler

19th February 2022 at 10:37am

Messages navigator-message

Introduced in v5.2.2 The `tm-relink-tiddler` message relinks a tiddler by deleting it and recreating it with a new title. Unlike the WidgetMessage: tm-rename-tiddler message, this message does not rename the tiddler, it just relinks the references to it.

The relink tiddler message requires the following properties on the `event` object:

Name	Description

from	Current title of tiddler
to	New title of tiddler
renameInTags	Optional value "no" to disable renaming in tags fields of other tiddlers (defaults to "yes")
renameInLists	Optional value "no" to disable renaming in list fields of other tiddlers (defaults to "yes")

The relink tiddler message is usually generated with the ButtonWidget and is handled by the core.

Examples

To relink Tiddler1 to Tiddler2 in tags and list fields of other tiddlers:

```
<$action-sendmessage $message="tm-relink-tiddler" from="Tiddler1" to="Tiddler2" />
```

WidgetMessage: tm-remove-field

8th September 2014 at 8:51pm

Messages

The `tm-remove-field` message is handled by the FieldManglerWidget. It removes the specified field.

Name	Description
param	Name of field to remove

The remove field message is usually generated with the ButtonWidget, and is handled by the FieldManglerWidget.

WidgetMessage: tm-remove-tag

8th September 2014 at 8:51pm

Messages

The `tm-remove-tag` message is handled by the FieldManglerWidget. It removes the specified tag.

Name	Description
param	Name of tag to remove

The remove tag message is usually generated with the ButtonWidget, and is handled by the FieldManglerWidget.

WidgetMessage: tm-rename-tiddler

19th February 2022 at 10:37am

Messages navigator-message

The `tm-rename-tiddler` message renames a tiddler by deleting it and recreating it with a new title. The rename tiddler message requires the following properties on the `event` object:

Name	Description
from	Current title of tiddler
to	New title of tiddler
renameInTags	ⓘ **Introduced in v5.1.23** Optional value "no" to disable renaming in tags fields of other tiddlers (defaults to "yes")
renameInLists	ⓘ **Introduced in v5.1.23** Optional value "no" to disable renaming in list fields of other tiddlers (defaults to "yes")

The very similar WidgetMessage: tm-relink-tiddler message does not rename the tiddler, it just relinks the references to it.

The rename tiddler message is usually generated with the ButtonWidget and is handled by the core.

Examples

To rename a tiddler called Tiddler1 to Tiddler2 and also renaming Tiddler1 in tags and list fields of other tiddlers:

```
<$action-sendmessage $message="tm-rename-tiddler" from="Tiddler1" to="Tiddler2" />
```

To rename a tiddler called Tiddler1 to Tiddler2 and not rename Tiddler1 in tags and list fields of other tiddlers:

```
<$action-sendmessage $message="tm-rename-tiddler" from="Tiddler1" to="Tiddler2" renameInTags="no"
renameInLists="no"/>
```

To rename a tiddler called Tiddler1 to Tiddler2 and respect the setting in the tiddler *$:/config/RelinkOnRename* for whether to rename Tiddler1 in tags and list fields of other tiddlers:

```
<$action-sendmessage $message="tm-rename-tiddler" from="Tiddler1" to="Tiddler2" renameInTags={{$:/config/
RelinkOnRename}} renameInLists={{$:/config/RelinkOnRename}}/>
```

WidgetMessage: tm-save-tiddler

26th February 2014 at 10:07am

Messages navigator-message

The `tm-save-tiddler` message is applied to draft tiddlers. It saves the draft over the tiddler identified in the `draft.of` field and then deletes the draft. The save tiddler message requires the following properties on the `event` object:

Name	Description
param	Title of the tiddler that is being switched out of edit mode
tiddlerTitle	Fallback title that is used if **param** isn't specified (automatically set by the ButtonWidget)

The save tiddler message is usually generated with the ButtonWidget and is handled by the NavigatorWidget.

WidgetMessage: tm-save-wiki

24th July 2023 at 12:09am

Messages

The save wiki message causes the current saver module to perform a full save operation. The save operation can involve user interaction. It requires the following properties on the `event` object:

Name	Description
param	Title of a tiddler to use as a template for rendering the wiki (defaults to `$:/core/save/all`)
{any other params}	Any other parameters are made available as variables to use for the rendering.

The save wiki message is usually generated by the ButtonWidget.

The save wiki message is handled by the TiddlyWiki core *SyncMechanism* which invokes the current *SaverModule*.

WidgetMessage: tm-scroll

14th October 2020 at 5:24pm

Messages

The `tm-scroll` message causes the surrounding scrollable container to scroll to the specified DOM node into view. The `tm-scroll` is handled in various places in the core itself, but can also be handled by a ScrollableWidget.

Name	Description
target	Target DOM node the scrollable container should scroll to (note that this parameter can only be set via JavaScript code)
selector	ⓘ **Introduced in v5.1.23** Optional string CSS selector as an alternate means of identifying the target DOM node
animationDuration	ⓘ **Introduced in v5.2.2** Optional number specifying the animation duration in milliseconds for the scrolling. Defaults to the **global animation duration**.



Set `animationDuration` to `0` to scroll without animation

WidgetMessage: tm-server-refresh

11th August 2014 at 1:34pm

Messages

The server refresh message attempts to synchronise the latest changes to the current serverside host.

The server refresh message is handled by the TiddlyWiki core *SyncMechanism* which invokes the current *SyncAdaptorModule* (typically the **tiddlywiki/tiddlywebadaptor** plugin).

WidgetMessage: tm-set-password

26th February 2014 at 9:52am

Messages

The `tm-set-password` message prompts the user for a new password and stores it in the password vault, replacing any existing password. It also sets the $:/isEncrypted tiddler. See *EncryptionMechanism* for details.

This message is typically generated with the ButtonWidget, and is handled by the core itself.

WidgetMessage: tm-unfold-all-tiddlers

28th October 2019 at 12:38pm

Messages

The `tm-unfold-all-tiddlers` message unfolds all tiddlers in the current story list. It does so by setting the text of all state tiddlers to: "**show**".

Name	Description
foldedStatePrefix	Prefix for the state tiddler in which the fold state is stored.



The core uses a foldStatePrefix of `$:/state/folded/` to store the fold states for the default story view.



The state tiddlers title is computed as `foldStatePrefix + TiddlerTitle`. If the foldStatePrefix is not set, it will overwrite the text of the tiddler(s) itself, resulting in data loss.

The `tm-unfold-all-tiddlers` message is usually generated with the ButtonWidget and is handled by the surrounding NavigatorWidget.

WidgetMessage: tm-unload-plugin-library

4th October 2019 at 1:36pm

Messages

The `tm-unload-plugin-library` message unloads the specified plugin library and deletes associated temporary tiddlers.

Name	Description
url	Url specifying the plugin library to be unloaded

Unloading a plugin library deletes the following tiddlers:

- The tiddler titled `$:/temp/ServerConnection/` + `<url>`
- All tiddlers with the prefix `$:/temp/RemoteAssetInfo/` + `<url>` + `/`

The `tm-unload-plugin-library` message is usually generated with the ButtonWidget and is handled by the core itself.

Widgets

19th February 2015 at 7:27pm

Concepts Reference

TiddlyWiki's display is driven by an underlying collection of **widgets**. These are organised into a tree structure: each widget has a parent widget and zero or more child widgets.

TiddlyWiki generates this **widget tree** by parsing the WikiText of tiddlers. Each component of the WikiText syntax, including even the trivial case of ordinary text, generates a corresponding widget. The widget tree is an intermediate representation that is subsequently rendered into the actual display.

Widgets are analogous to elements in an HTML document. Indeed, HTML tags in WikiText generate dedicated **element widgets**.

Each class of widget contributes a specific ability to the overall functionality, such as the ability to display an image or a button, to call a macro or transclude text from elsewhere, or to mark a piece of text as a heading.

The more specialised widgets use a general-purpose widget syntax as their only possible WikiText representation.

The following classes of widget are built into the core:

- action-confirm
- action-createtiddler
- action-deletefield
- action-deletetiddler
- action-listops
- action-log
- action-navigate
- action-popup
- action-sendmessage
- action-setfield
- action-setmultiplefields
- ActionWidget Execution Modes
- ActionWidgets
- browse
- button

- checkbox
- codeblock
- count
- diff-text
- draggable
- droppable
- dropzone
- edit-bitmap
- edit-text
- edit
- encrypt
- entity
- error
- EventCatcherWidget
- fieldmangler
- fields
- fill
- genesis
- image
- importvariables
- jsontiddler
- keyboard

- let
- linkcatcher
- link
- list
- LogWidget
- macrocall
- MessageCatcherWidget
- MessageHandlerWidgets
- navigator
- parameters
- password
- vars
- radio
- range
- reveal
- scrollable
- select
- setmultiplevariables
- setvariable
- set
- slot
- text

- tiddler
- transclude
- TriggeringWidgets
- vars
- view
- wikify

Widgets in WikiText

13th May 2022 at 1:33pm

WikiText

Widgets provide rich functionality within WikiText. They have the same syntax as HTML elements, but the tag name always starts with `$`. For example:

copy to clipboard

```
<$button message="tm-navigate" to="HelloThere">Open ~HelloThere</$button>
```

Try it

Note that widgets inherit all the features of HTML in WikiText:

- Widget attributes can be specified as:
  - Unquoted strings (that cannot contain spaces)

    - Strings quoted with single or double quotes
    - Strings quoted with triple-double quotes
    - Macro invocations (eg `attr=<<myMacro>>`)
    - Transclusions (eg, `attr={{MyTiddler!!field}}`)
    - Filtered transclusions (eg, `attr={{{ [filter[op]] }}}`)
    - ⓘ **New in v5.3.0** Substituted Attribute Values
- The content of a widget is parsed in inline mode unless the opening tag is followed by two linebreaks, which forces block mode
    - inline mode means that block mode parse rules like headings, tables and lists are not recognised

See HTML in WikiText for more details.

The available widgets include:

- action-confirm
- action-createtiddler
- action-deletefield
- action-deletetiddler
- action-listops
- action-log
- action-navigate
- action-popup
- action-sendmessage
- action-setfield
- action-setmultiplefields

- ActionWidget Execution Modes
- ActionWidgets
- browse
- button
- checkbox
- codeblock
- count
- diff-text
- draggable
- droppable
- dropzone
- edit-bitmap
- edit-text
- edit
- encrypt
- entity
- error
- EventCatcherWidget
- fieldmangler
- fields
- fill
- genesis

- image
- importvariables
- jsontiddler
- keyboard
- let
- linkcatcher
- link
- list
- LogWidget
- macrocall
- MessageCatcherWidget
- MessageHandlerWidgets
- navigator
- parameters
- password
- vars
- radio
- range
- reveal
- scrollable
- select
- setmultiplevariables

- setvariable
- set
- slot
- text
- tiddler
- transclude
- TriggeringWidgets
- vars
- view
- wikify

Wiki

Concepts

Groups of uniquely titled tiddlers are contained in *WikiStore* objects.

The *WikiStore* also manages the plugin modules used for widgets, and operations like serialising, deserialising, parsing and rendering tiddlers.

WikificationMechanism

13th October 2019 at 11:40am

Mechanisms

"Wikification" is a general term for the dynamic process of converting tiddlers containing WikiText into the HTML DOM representation needed by the browser, and updating that representation if the underlying tiddlers change.

It is composed of several distinct steps:

- *ParserMechanism*: reading the text of tiddlers and scanning for wikitext constructions, outputting a tree representation of the resulting structure. It is an expensive process so parse trees are cached, and only need to be updated if the corresponding tiddler is changed
- *WidgetMechanism*: starting with a specified root tiddler, recursively instantiate a widget for each parse tree node making a rendering tree. Widgets can optionally also create DOM nodes
- RefreshMechanism: handling changes to the tiddler store by selectively and efficiently updating a rendering tree

This mechanism is used in the browser to build TiddlyWiki's main interactive page. At startup, the tiddler **$:/core/ui/PageTemplate** is parsed and rendered to the DOM, recursively pulling in other tiddlers to build the entire user interface. Any user interactions – following a link, clicking a button, or typing in a text box – trigger a change in the tiddler store which then automatically propagates through the widget tree. For example, if the user clicks a link to navigate to a new tiddler, the following steps take place:

1. Clicking the link triggers the action of the LinkWidget which by default is to add the target tiddler to the list field of the tiddler $:/StoryList
2. The modification to the tiddler store asynchronously triggers the refresh cycle. The asynchronous triggering ensures that the refresh cycle is only run once even if multiple tiddlers were modified in succession
3. The refresh cycle recursively visits each node of the render tree giving them the chance to update themselves in the light of the accumulated changes to the tiddler store. In this case, the ListWidget of the main story river notices that a single tiddler needs to be added to the river, and renders that newly displayed tiddler without disturbing the other tiddlers

The performance of the entire wikification process is critical. If the refresh cycle takes more than about 400ms then the user will notice a delay between their actions and the effects. See Performance for some discussion of how to optimise performance.

The rendering process is also aggressively reused in other parts of TiddlyWiki, both in the browser and on the server:

- Generating TiddlyWiki's standalone HTML representation
- Creating static HTML renderings of tiddlers
- Dynamically rendering CSS stylesheet tiddlers

WikifyWidget

22nd January 2022 at 8:15pm

Widgets

Introduction

The wikify widget parses and renders a string of text and assigns the result to a specified variable. The new value of the variable is available to the content within the wikify widget.

Content and Attributes

The content of the `<$wikify>` widget is the scope for the value assigned to the variable.

Attribute	Description
name	The name of the variable to assign
text	The text to parse and render

type	The ContentType of the text (defaults to `text/vnd.tiddlywiki`)
mode	The parse mode: block (the default) or inline
output	Keyword indicating the desired output type, defaulting to `text` (see below)

The available output types are:

Keyword	Description
text	Return the plain text of the rendered output (ie HTML tags are omitted)
formattedtext	Return the plain text with simple text formatting of the rendered output (ie HTML tags are omitted)
html	Return the rendered HTML of the output
parsetree	Return a formatted JSON representation of the parse tree
widgettree	Return a formatted JSON representation of the widget tree

WikiText

3rd March 2023 at 10:47pm

Concepts Reference

WikiText is a concise, expressive way of typing a wide range of text formatting, hypertext and interactive features. It allows you to focus on writing without a complex user interface getting in the way. It is designed to be familiar for users of Markdown but with more of a focus on linking and the interactive features.

WikiText can also be inserted to the text field using the Editor toolbar buttons.

See Formatting in WikiText and Formatting text in TiddlyWiki for an introduction to WikiText.

The following elements of WikiText syntax are built into the core:

- Block Quotes
- Code Blocks
- Conditional Shortcut Syntax
- Dashes
- Definitions
- Filtered Attribute Values
- Formatting
- Hard Linebreaks
- Headings
- Horizontal Rules
- HTML Entities
- HTML
- Images
- Linking
- Lists
- Literal Attribute Values
- Macro Calls
- Macro Definitions
- Macro Parameter Handling
- Paragraphs
- Procedure Calls
- Procedure Definitions
- Procedure Parameter Handling
- Styles and Classes
- Substituted Attribute Values
- Tables
- Transcluded Attribute Values
- Transclusion and Substitution
- Transclusion
- Typed Blocks
- Utility Classes
- Variable Attribute Values
- Variables
- Widget Attributes
- Widgets
- WikiText Parser Modes

WikiText parser mode transitions

22nd January 2022 at 7:42pm

WikiText Parser Modes

This is a *ROUGH* diagram whose lines mostly correspond to the parser mode transitions described below:

TypeError: setting getter-only property "parentNode"

Start mode

By default the parser starts in block mode. However, a tiddler can instead be transcluded with inline mode in which case block mode WikiText will not be recognised.

At the start of text only, the parser will also recognise any pragma mode WikiText.

Transitions from pragma mode

At the start of text, the parser will recognise any pragma. If none are found then it will move to inline or block mode depending on the transclusion mode. If any pragma are found then it will continue looking for pragma until it finds one or more blank lines not followed by the start of a new pragma.

Transitions from block mode

When the parser encounters most block mode start punctuation it transitions to inline mode. This means he text enclosed by most block mode constructs will be parsed using inline mode.



The start "punctuation" for a paragraph is "invisible". Even for paragraphs the parser moves to inline mode

However, there are a few constructs whose enclosed text is parsed using block mode:

- Multi-line block quotes
- Multi-line style blocks
- Typed blocks

When the start punctuation for these are encountered, the enclosed text will continue to be parsed in block mode.

Horizontal rules are another special case. They do not enclose any text, so there is no opportunity for the parser to transition to inline mode.

Opening widget or HTML tags provide another way the parser can transition. When such a tag is *NOT* followed by a blank line, then the contents enclosed by the tag will be parsed in inline mode. See the HTML examples.

Transitions from inline mode

- The parser will move back to block mode after the end of a line which terminates block mode WikiText. In other words, when the block mode close punctuation is encountered.
- When the opening widget or HTML tag is followed by a blank line, then the contents enclosed by the tag will be parsed in block mode. See the HTML examples.
- Transcluded text and text pulled in via a macro call can transition the parser away from inline mode. See transclusion examples and macro examples for more details.

WikiText parser mode: HTML examples

22nd January 2022 at 7:05pm

WikiText parser mode transitions

copy to clipboard

```
<div>

* blank line after open tag allows
* block mode punctuation (i.e. this list) to
* be recognised
</div>
```

That renders as:

- blank line after open tag allows
- block mode punctuation (i.e. this list) to
- be recognised

The open tag blank line "trick" also works inside of punctuation only recognised in inline mode (such as Formatting in WikiText):

copy to clipboard

```
''<div>

* list item one
* list item two
</div>''
```

That renders as:

- **list item one**
- **list item two**

---

copy to clipboard

```
<div>
* Open tag without a blank line means
* block mode punctuation (i.e. this list) will
* NOT be recognised.

Paragraphs are only recognised in block mode. Even with blank lines, new paragraphs are not recognised.

This also is not a new paragraph.
</div>
```

That renders as:

* Open tag without a blank line means * block mode punctuation (i.e. this list) will * NOT be recognised. Paragraphs are only recognised in block mode. Even with blank lines, new paragraphs are not recognised. This also is not a new paragraph.

Tables in WikiText is one construct for which the open tag blank line "trick" cannot work. In order for the parser to recognise a table row, the entire row must be on a single line.

Adding a blank link after an open tag within a table cell causes the row to span mulitple lines and the parser ceases to be able to recognise it as a table row.

copy to clipboard

```
|table row|all on|one line|is recognised|
```

That renders as:

table row	all on	one line	is recognised

copy to clipboard

```
|For a blank line after open tag|<div>

* list is recognised
* but the surrounding table row is not
</div>|
```

That renders as:

|For a blank line after open tag|

- list is recognised
- but the surrounding table row is not

|

The only way to get block mode punctuation recognised in wikitext tables is to use **`$transclude`** with **`mode`** = **`block`**.

copy to clipboard

```
|Nested table:|<$transclude tiddler="table-example" mode=block/>|
```

That renders as:

Nested table:	cell one	cell two

cell three	cell four

Note that **$macrocall** does not have a similar **mode** attribute and the content of macro definitions can only be parsed in inline mode

---

The less convenient `<table>`, `<tr>`, `<td>` html tags can be used as a foolproof way to get block mode punctuation recognised inside of table cells.

copy to clipboard

```
<table>
<tr><td>

* list item one
* list item two
</td><td>

|nested|table|
</td></tr>
</table>
```

That renders as:

- list item one
- list item two

nested	table

WikiText parser mode: macro examples

22nd January 2022 at 7:09pm

WikiText parser mode transitions

With the macro `boringtable` defined as:

```
|cell one|cell two|
|cell three|cell four|
```

then

	wiki text	renders as
1	`* <<boringtable>>`	• |cell one|cell two| |cell three|cell four|

2	`* <div>`
	
	`<<boringtable>>`
	`</div>`
		• <table><tr><td>cell one</td><td>cell two</td></tr><tr><td>cell three</td><td>cell four</td></tr></table>

3	`* <$macrocall $name=boringtable/>`	• |cell one|cell two| |cell three|cell four|

	wiki text	renders as
4	`* <$macrocall $name=boringtable></$macrocall>`	• |cell one|cell two| |cell three|cell four|

5	`* <$macrocall $name=boringtable>`
	
	`</$macrocall>`
		• <table><tr><td>cell one</td><td>cell two</td></tr><tr><td>cell three</td><td>cell four</td></tr></table>

The list syntax is recognised in block mode and the enclosed contents are parsed using inline mode. When the parser encounters a wikitext macro call it will use the current parse mode to parse the contents of the macro. The contents of the macro contains table syntax which is only recognised in block mode.

Therefore, in #1 above the table syntax is not recognised. In #2 above, the blank line after the open `div` tag moves the parser back into block mode, the macro call inherits it and the table is recognised.

When calling macros using the widget, the parse mode will be inline (#3 and #4 above) unless it is written as an open tag with a following blank line (#5 above).



See also WikiText parser mode: transclusion examples for similar examples

---

These examples have slightly different behavior. In the previous section, the macro calls were enclosed within list items. In these examples, the macro calls are at the top level:

	wiki text	renders as

	wiki text	renders as
1	`<<boringtable>>`	cell one | cell two cell three | cell four

2	`<div>`
	
	`<<boringtable>>`
	`</div>`
		cell one | cell two
		cell three | cell four

3	`<$macrocall $name=boringtable/>`	|cell one|cell two| |cell three|cell four|
4	`<$macrocall $name=boringtable></$macrocall>`	|cell one|cell two| |cell three|cell four|

5	`<$macrocall $name=boringtable>`
	
	`</$macrocall>`
		cell one | cell two
		cell three | cell four

Of these examples, only the two **`$macrocall`** tags which are not followed by a blank line are parsed using inline mode.



See also WikiText parser mode: transclusion examples for similar examples

WikiText parser mode: transclusion examples

22nd January 2022 at 7:08pm

WikiText parser mode transitions

Given the tiddler table-example defined with these contents:

```
|cell one|cell two|
|cell three|cell four|
```

then

	wiki text	renders as
1	`* {{table-example}}`	• |cell one|cell two| |cell three|cell four|

2	`* <div>`
	
	`{{table-example}}`
	`</div>`
		• <table><tr><td>cell one</td><td>cell two</td></tr><tr><td>cell three</td><td>cell four</td></tr></table>

3	`* <$transclude tiddler=table-example/>`	• |cell one|cell two| |cell three|cell four|

	wiki text	renders as
4	`* <$transclude tiddler=table-example></$transclude>`	• |cell one|cell two| |cell three|cell four|

5	`* <$transclude tiddler=table-example>`
	
	`</$transclude>`
		• <table><tr><td>cell one</td><td>cell two</td></tr><tr><td>cell three</td><td>cell four</td></tr></table>

6	`* <$transclude tiddler=table-example mode=block/>`	• <table><tr><td>cell one</td><td>cell two</td></tr><tr><td>cell three</td><td>cell four</td></tr></table>

The list syntax is recognised in block mode and the enclosed contents are parsed using inline mode. When the parser encounters a wikitext transclusion it will use the current parse mode to parse the contents of the transcluded tiddler. The contents of the example tiddler contains table syntax which is only recognised in block mode.

Therefore, in #1 above the table syntax is not recognised. In #2 above, the blank line after the open `div` tag moves the parser back into block mode, the transcluded text inherits it and the table is recognised.

When transcluding tiddlers using the widget, the parse mode will be inline (#3 and #4 above) unless it is written as an open tag with a following blank line (#5 above).



> See also WikiText parser mode: macro examples for similar examples. The only difference here is **`$transclude`** has a **`mode`** attribute (extra example #6) which allows the parse mode to be explicitly overridden rather than come implicitly based on the they way the tag is written

---

In these examples, the transclusions are at the top level instead of enclosed in list items:

	wiki text	renders as
1	`{{table-example}}`	cell one | cell two cell three | cell four

2	`<div>`
	
	`{{table-example}}`
	`</div>`
		cell one | cell two
		cell three | cell four

3	`<$transclude tiddler=table-example/>`	|cell one|cell two| |cell three|cell four|
4	`<$transclude tiddler=table-example></$transclude>`	|cell one|cell two| |cell three|cell four|

5	`<$transclude tiddler=table-example>`
	
	`</$transclude>`
		cell one | cell two
		cell three | cell four

6	`<$transclude tiddler=table-example mode=inline/>`	|cell one|cell two| |cell three|cell four|

Of these examples, only the two **$transclude** tags which are not followed by a blank line and the widget call using **mode=inline** are parsed using inline mode.



See also WikiText parser mode: macro examples for similar examples

WikiText Parser Modes

13th May 2022 at 1:26pm

WikiText

In order to display Tiddlers (usually the text field), the WikiText parser reads and interprets the content and applies WikiText rules. The parser has three modes:

- **pragma mode** - the parser will recognise only pragma mode WikiText punctuation
- **block mode** - the parser will recognise only block mode WikiText punctuation
- **inline mode** - the parser will recognise only inline mode WikiText

The parser transitions between these modes based on the text it encounters. In addition, there are places where the parser ignores WikiText punctuation. The parser mode can also be set directly with the Pragma: \parsermode.



The concept of inline vs block also exists for standard HTML elements. For HTML, these two layout modes determine if the output flows together on the same line or not.

Most block mode WikiText corresponds to block level HTML elements and most inline mode WikiText corresponds to inline level HTML elements. However, for Wikitext the two modes are just as important for determining which syntax will be recognised by the parser as they are for determining how the output will flow.

Windows HTA Hack

7th May 2020 at 1:03pm

Saving Windows

Under Windows it is possible to convert TiddlyWiki into a true local application by renaming the HTML file to have the extension `*.hta`. The **fsosaver** module can then use the ActiveX FileSystemObject to save changes.

Note that one disadvantage of this approach is that the TiddlyWiki file is saved in UTF-16 format, making it up to twice as large as it would be with the usual UTF-8 encoding. However, opening and saving the file via another saving method will re-encode the file to UTF-8.

See Wikipedia for more details: https://en.wikipedia.org/wiki/HTML_Application

Working with the TiddlyWiki5 repository

20th September 2014 at 3:44pm

TiddlyWiki on Node.js

Introduction

If you want to contribute to TiddlyWiki's development, rather than installing TiddlyWiki in the usual way, you can work directly with the GitHub repository.

Mario Pietsch has created a short video introduction.

Setting Up

1. Create an account on GitHub if you don't already have one
2. Fork the TiddlyWiki5 GitHub repository from https://github.com/Jermolene/TiddlyWiki5
3. Clone a local copy of your fork
4. Open a command line terminal and change the current working directory to the root of the repo
5. Type `npm link` (Windows) or `sudo npm link` (Mac/Linux) to tell *npm* to use this copy of the repo as the globally installed one

After this procedure you can work with TiddlyWiki5 via *npm* as though it were installed in the usual way with `npm install -g tiddlywiki`.

See also Scripts for TiddlyWiki on Node.js.

Working with the TiddlyWiki5 repository video

20th September 2014 at 3:40pm

Working with the TiddlyWiki5 repository

Mario Pietsch has created a short video tutorial on working with the TiddlyWiki5 GitHub repository.

Working with TiddlyWiki

19th September 2014 at 9:11pm

TableOfContents

Information to help you get started using TiddlyWiki:

- The First Rule of Using TiddlyWiki
- GettingStarted
- Getting Started Video

- Upgrading
- Navigating between open tiddlers
- Using links to navigate between tiddlers
- Searching in TiddlyWiki
- Creating and editing tiddlers
- Creating journal tiddlers
- Saving
- Formatting text in TiddlyWiki
- Structuring TiddlyWiki
- Tagging
- Images
- KeyboardShortcuts
- Encryption
- Audio
- BrowserCompatibility
- How to update TiddlyWiki to the latest version
- Introduction Video
- Performance

Xememex

10th April 2023 at 12:50pm

Definitions

Xememex is a multiuser TiddlyWiki from Federatial. It allows large groups of people to work together on intertwingled wikis that can share content. It is implemented as a serverless application on Amazon Web Services.

XEMEMEX

The largest customer implementation has hundreds of online wikis with thousands of users. See https://manuals.annafreud.org/

Xememex is currently only available under commercial terms from Federatial. Contact Jeremy Ruston at Federatial for more details.

Xememex Logo

XEMEMEX

XLSX Utilities Edition

23rd October 2016 at 10:23pm

Editions

The **XLSX Utilities** edition of TiddlyWiki contains tools to work with `.XLSX` spreadsheets generated by applications like Microsoft Excel and Google Sheets. It can be used in the browser or under Node.js.

https://tiddlywiki.com/editions/xlsx-utils/

zth Operator

29th October 2021 at 4:37am

Filter Operators Order Operators

purpose	select the **Z** th input title
input	a selection of titles
parameter	**Z** = an integer, defaulting to 0
output	the **Z** th input title

Learn more about how to use Filters

Introduced in v5.2.1

Z is zero-based. In other words, `zth[0]` has the same effect as `nth[1]` and the **first** operator.

See **nth** for an equivalent operator with a 1 based parameter

Examples

zth Operator (Examples)

29th October 2021 at 4:35am

Operator Examples zth Operator

These examples make use of the Days of the Week tiddler.

```
[list[Days of the Week]zth[]]
```

Try it

```
[list[Days of the Week]zth[5]]
```

Try it

中文社区 - Chinese Community

Community

1. A Chinese community tutorial program that people can edit together:
   - Main site: https://tw-cn.netlify.app/
   - Accelerated access: https://tw-cn.cpolar.top/

    - Alternate: https://tiddly-wiki-chinese-tutorial.vercel.app
2. Tiddlywiki Chinese Chat Forum: https://talk.tidgi.fun/topic/6
3. Chinese translation of Tiddlywiki official website https://bramchen.github.io/tw5-docs/zh-Hans/
4. The best Chinese introductory tutorial for newbies https://keatonlao.github.io/tiddlywiki-xp/

---

1. 大家可以一起编辑的中文社区教程项目：
    - 主站：https://tw-cn.netlify.app/
    - 加速访问：https://tw-cn.cpolar.top/
    - 备用：https://tiddly-wiki-chinese-tutorial.vercel.app
2. 太微中文交流论坛：https://talk.tidgi.fun/topic/6
3. 太微官网汉化版：https://bramchen.github.io/tw5-docs/zh-Hans/
4. 最适合新手的中文入门教程：https://keatonlao.github.io/tiddlywiki-xp/