

THE BOOK OF TIDDLY WIKI

ADVANCED CUSTOMIZATION

LUIS J. GONZÁLEZ CABALLERO

Your messy thoughts. Organized.



tiddlywiki.com

- your personal wiki
- laptop, mobile, tablet
- own your data 100%
- a single HTML file
- offline or in the cloud
- open source and free

Advanced Customization

Luis Javier González Caballero
Riz - google groups

October 25, 2019

About the writers



Luis Javier González Caballero is an IT Technician with extensive experience in user support. He has done some programming works mainly in Python. He is an Industrial Technician Engineer from the <https://www.unileon.es>, Spain. and a Computer Engineer from the <https://www.uned.es> in Spain too.

He is an active person with a great capacity for learning, which he is involved.

Apart from his professional career he collaborates with numerous non-profit organizations and free projects and is an animal lover. He lives in León, a small town in

northwestern Spain with his family.

@kewapo@gmail.com

Contents

1	Introduction	11
1.1	Key points	11
1.2	What is tiddlywiki	12
1.3	Starting with tiddlywiki	13
1.4	Reasons to use tiddlywiki	14
1.5	Elements of TW	14
1.5.1	The screen	14
1.5.2	Tiddlers	14
1.5.3	The Story River	16
1.5.4	Tags	16
1.5.5	Fields	16
1.5.6	Text format	16
1.5.7	Transclusion	16
1.5.8	Templates	17
1.5.9	Filters	17
1.5.10	Macros and widgets	17
1.5.11	Mechanism	17
1.5.12	Lists	17
2	Loading tiddlywiki in the browser	19
2.1	Key points	19
2.2	Introduction	20
2.3	Architecture	21
2.4	Tiddlers as Basic Coding Elements	21

3	Components of the tiddlywiki screen	23
3.1	Key points	23
3.2	The TW whole screen	24
3.2.1	The story river	25
3.2.2	The right sidebars	25
3.3	Drawing its interface	25
4	Customize the TW screen	27
4.1	Key points	27
4.2	Introduction	28
4.3	PageTemplate	28
4.4	ViewTemplate	29
4.4.1	Example	29
4.5	EditTemplate	30
4.6	Using CSS in TW	31
4.6.1	Example	31
5	Inside a tiddler	33
5.1	Types of tiddlers	33
5.2	Normal tiddler	34
5.2.1	Formatting text	34
5.2.2	Linking content	34
5.2.3	Basic transclusion	34
5.2.4	Showing external content	34
5.3	Adding macros and widgets	34
5.4	CSS tiddlers	34
5.5	Alert tiddlers	35
5.6	Tag tiddlers	35
5.7	Data tiddlers	35
5.8	Macro tiddlers	35
6	The filter notation	37
6.1	The filter basic step	37
6.2	Using filter operators	37
6.3	Combining filter steps	37
6.4	Regular expressions	37
6.5	Filter uses	37
6.6	Examples	37

7	The behavior of tiddlywiki	39
7.1	Creating microcontent	39
7.2	Organizing microcontent with links	39
7.3	Organizing tiddlers with tags	39
7.4	Personalize behavior with macros and widgets	39
8	Macros	41
8.1	Basic macros included	41
8.2	Writing your own macros	41
8.3	Using macros	41
9	Widgets	43
9.1	Basic widgets included	43
9.2	Writting your own widgets	43
9.3	Using widgets	43
10	The scripting business	45
10.1	List scripting	45
10.2	Interface scripting	45
10.3	Data tiddlers scripting	45
10.4	Template scripting	45
10.5	Tiddlers scripting	45
11	Recipe book	47
11.1	Personal todo-list	47
11.2	Writting stories	47
11.3	Image gallery	47
11.4	Lesson planner	47
11.5	Simple game	47
12	Languages	49
12.1	Tiddlywiki languages	49
12.2	Writing international wikis	49
13	Plugins	51
13.1	Extending tiddlywiki functionality	51
13.2	Creating plugins	51
13.3	Where we can find plugins	51
14	The people. The project	53



Mastering Tiddlywiki is a hard a long trip. Why? you can ask yourserf. I answer you. Tiddlywiki is a confluence of many technologies:

1. The first step is understanding Wikitext. It is a markup language like Markdonw: it specify code characters for the format.
2. You need some understanding of HTML. At the bottom, Tiddlywiki is a HTML file with Javascript code.
3. Tiddlywiki uses CSS too. CSS is a technology that goes hand in hand with HTML.
4. If you want to program you own widgets you need Javascript.
5. Although you can use tw as a single file, you can have a self-hosted wiki installing node.js and tiddlywiki from its repository.

But f you want to be a Tiddlywiki master, the first of all is to understand the underlying of Tiddliwiki. This is where this book is going to help you.

Chapter 1

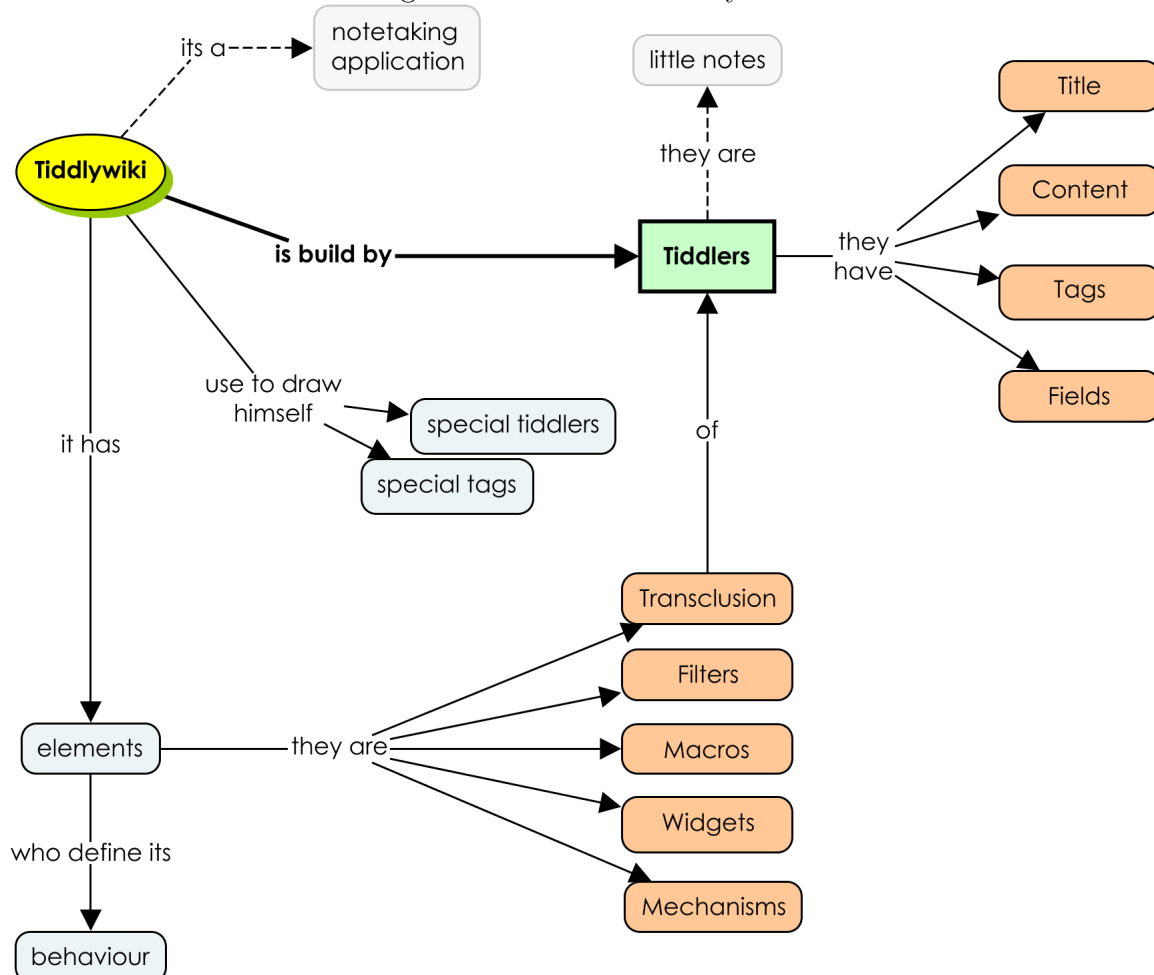
Introduction

1.1 Key points

- Tiddlywiki is more than a note-taking application.
- It is an advanced way of organizing your information.
- All notes you will add to it are called tiddlers.
- You can download an empty wiki from its web site.
- It is portable and multi platform.
- To add format to your text inside the tiddlers you use format characters: `//text//` for italics, `__text__` for underline, `[[Tiddler]]` for links etc. This is called Wikitext.
- It use tags to organize, classify etc the tiddlers.
- It use fields to add extra information to the tiddlers. For example the created date field, the modified date field etc.
- You can include the information of a tiddler inside other tiddler without writing twice. This is called Transclusion.
- You can use templates to personalize the way that tiddlywiki shows the tiddlers.
- Tiddlywiki uses Filters to get a group of tiddlers. Filters are written in a little language inside tiddlywiki. For example, to get all tiddlers tagged with the “Person” tag you write `{{ [tag[Person]] }}`.
- You can add code to your wiki in Macros and Widgets. This is used to personalize the wiki.

1.2 What is tiddlywiki

Figure 1.1: What is Tiddlywiki

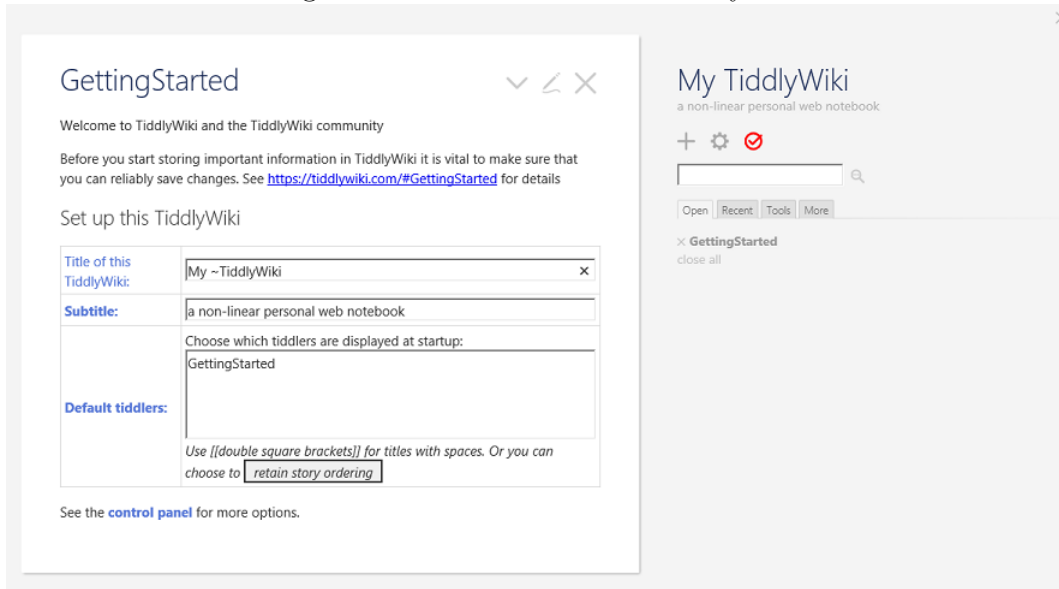


TiddlyWiki is a personal and a non-linear notebook for organizing and sharing complex information. It is an open-source single page application wiki in the form of a single HTML file that includes all javascript code, the CSS format and the content. It is designed to be easy to customize and re-shape depending on application. It facilitates re-use of content by dividing it into small pieces called Tiddlers. It is not an application but a large html page that runs in almost all web browsers so it is very portable: you can use it in a USB stick, in a phone or tablet or as a web page in some internet servers. Tiddlywiki is made of tiddlers: little text areas or notes with a title and a content. You can add all the tiddlers you want and each of them will contain certain information.

It was created by the British software developer Jeremy Ruston in 2004. Tiddlywiki is free and open source software and is distributed under the terms of the BSD license.

Tiddlywiki introduces the concept of microcontent: the smallest structured and addressable piece of information, the smallest semantically meaningful units. This small piece of information is called tiddler. The purpose of this tiddlers is recording and organizing information is so that it can be used as many times as necessary.

Figure 1.2: The first run of Tiddlywiki



You can find the basic information of Tiddlywiki in the web site:<https://tiddlywiki.com/>. I recommend you take a look.

1.3 Starting with tiddlywiki

The first step will be download an empty wiki page to our computer:

1. Go to<https://tiddlywiki.com/>.
2. Look for the tiddler“GettingStarted”
3. Click on the red button,“Download Empty”
4. Save the empty tiddlywiki in your computer.

Once you have the file you can open it with the browser (best with Firefox or Chrome). You will see the page of the Figure1.2. Assign a title and a subtitle to your wiki. You will notice that the tick icon above the subtitle change to red.

This shows us an important thing: as an html web page, the TW file can't save by itself. You must save them by hand if you don't want to lose the data. Click in this red icon and save the file in the same location (maybe you have to configure your browser). The next time you open the empty wiki in the browser the title and subtitle will have changed. Other operations you can do is to create an initial tiddler and change the“Default tiddlers” text area to the title of this first tiddler. To create tiddlers you will click on the "plus" icon under the subtitle.

The First Rule of using Tiddlywiki: backup your wiki file regularly retaining some backward copies.

1.4 Reasons to use tiddlywiki

You have powerful reasons to use tiddlywiki:

- You can use them without installing any software.
- It is multiplatform: use your wiki in your tablet, mac, linux, windows or phone systems.
- It is portable: you can put in a USB stick, copy to your computer or upload to many internet servers.
- Its unlinear approach allows you to use the information in new and helpful ways.
- You can organize your information and knowledge your way.
- You can change the appearance and behavior of your wiki and adapt them to your preferences.
- Tiddlywiki promote information sharing.

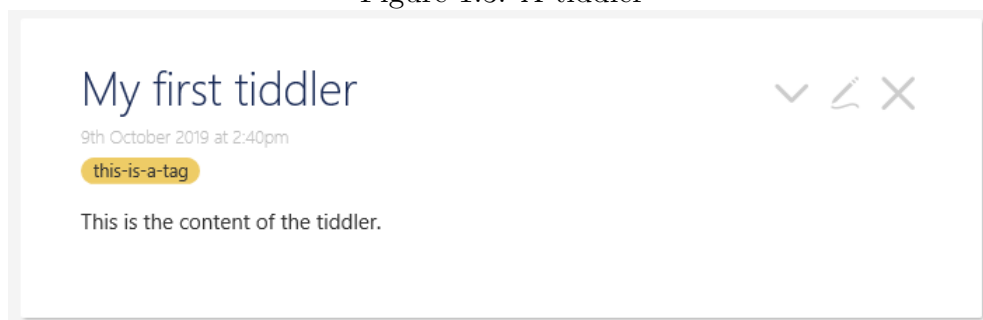
1.5 Elements of TW

1.5.1 The screen

When you open a Tiddlywiki file you can see a left side with all open tiddlers and a right side with the title, buttons, and menus. In chapter 3 we will see the whole screen (look at the [Figure 3.1](#)).

1.5.2 Tiddlers

Figure 1.3: A tiddler



All notes you can add to Tiddlywiki are written in Tiddlers. It consist of a Title on the top, the tags and its content. A Tiddler is the basic element of Tiddliwiki. All things inside TW are made of tiddlers. The GettinStarted page you can see is a tiddler. The right menus too.

The most important thing in a tiddler is its title: this will be unique. In Figure 1.3 you can see the first tiddler:

- Buttons to edit and close the tiddler.
- Title: “My first tiddler”
- The field: created date.
- Tags: only one: “this-is-a-tag”. You can add as many tags as you want.
- Content: “This is the content of the tiddler”.

When you click the “pen” icon at the top you can see a tiddler in edition mode:

Figure 1.4: Creating a tiddler

The screenshot shows the 'Draft of New Tiddler' form. At the top, there's a title field with 'New Tiddler' and a close button (X). Below the title is a tag zone with a 'tag name' input and an 'add' button. A rich text editor toolbar is visible with various icons for bold, italic, strikethrough, underline, superscript, subscript, text color, background color, link, unlink, list, and heading. Below the toolbar is a large text area for the tiddler's content. At the bottom, there's a 'Type' dropdown set to 'content type' and a trash icon. Below that is a 'fields zone' with 'Add a new field:', a 'field name' input, a 'field value' input, and an 'add' button.

- Buttons to delete, discard and save the tiddler
- Title at the top: “New Tiddler”
- Tag zone
- The format toolbar
- The content of the tiddler
- The type of the tiddler
- The fields zone. There are predefined fields but you can add your own fields.

If you close the tiddler it disappears from the screen. You can search it with the search bar, under the save button or choosing in the right menu More / All.

1.5.3 The Story River

The left side of the Tiddlywiki is called the Story River and shows all open tiddlers.

A typical wiki contains hundreds or thousands of tiddlers, some of them open in the left side and others closed stored in the file. You can search the closed tiddlers with the search bar or with the menus.

1.5.4 Tags

Tagging is a way of organizing 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. By tagging your tiddlers, you can view, navigate and organize your information.

1.5.5 Fields

A tiddler has field. There are system field like the created and modified date but you can add your own ones.

1.5.6 Text format

The formatted text inside a tiddler is called "Wikitext". The best way of learning Wikitext is playing with the toolbar above the tiddler in edition mode. For example, if you want some words to be italicized and click in the italic icon of the tool bar while editing a tiddler, Tiddlywiki will add two slashes around the words: `//some words//`. 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.

Other example: you can make a link of a tiddler inside other tiddler writing `[[Tiddler Title Linked]]`.

1.5.7 Transclusion

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". It avoid Avoid having duplicate information.

- To show the information of a tiddler inside other, write `{{Tiddler Title}}`.
- To show the content of a field write `{{Tiddler Tittle !! field name}}`.
- to show the content of a field in the same tiddler write: `{{!! field name}}`

1.5.8 Templates

Is a role a tiddler can have. Is like a shape for other tiddlers. It tells other tiddlers the way they have to display: how to show the title, the tags, its content and the other tiddlers. When you download an empty tiddler the initial template for all tiddlers is:

- `$/core/ui/ViewTemplate` if the tiddler is in view mode (you are not editing it)
- `$/core/ui/EditTemplate` if you are editing the tiddler.

They (View and Edit templates) are tiddlers. You can search them in the advanced search and look for its content. And you can add your own templates.

1.5.9 Filters

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. For example, to show the titles of all tiddlers tagged with the “learn” tag we can write `{{ [tag[learn]] }}`. The `[tag[learn]]` item is the filter and the brackets the way to add the links.

1.5.10 Macros and widgets

Tiddlywiki is highly customizable. It use macros and widgets to personalize its appearance.

A macro is a named snippet of text. When you use it, Tiddlywiki shows its content.

A widget is a piece of code to perform some actions.

1.5.11 Mechanism

All elements of tiddlywiki fit together through mechanism. For example, the HistoryMechanism keeps track of a list of tiddlers comprising the navigation history. The StartupMechanism runs the installed startup modules at the end of the boot process.

1.5.12 Lists

The manage of list is an important question in Tiddlywiki. The more usual list is a tiddler list. This is a list with a few number of tiddlers. For example:

[[First Tiddler]] SecondTiddler [[Third tiddler]] Finaltiddler

is a list with 4 tiddlers. You can see tiddlers inside `[[-]]` and other tiddlers not. If your tiddler contains spaces you have to surround with `[[-]]`.

You can construct a list with filters. For example if you find this code in a tiddler:

`<$list filter="[tag[$:/tags/PageTemplate]]">`

you can find a list with all tiddlers tagged with the `$:/tags/PageTemplate` tag.

The most important thing here is the order of the tiddlers inside the list. In the first example the order is clear but in the second is not. Tiddlywiki will order the list alphabetically. But, imagine you need other order in this `$list` inside your tiddler. In this case you can include this order in the “list” field of the tiddler. Tiddlywiki will respect that order.

Chapter 2

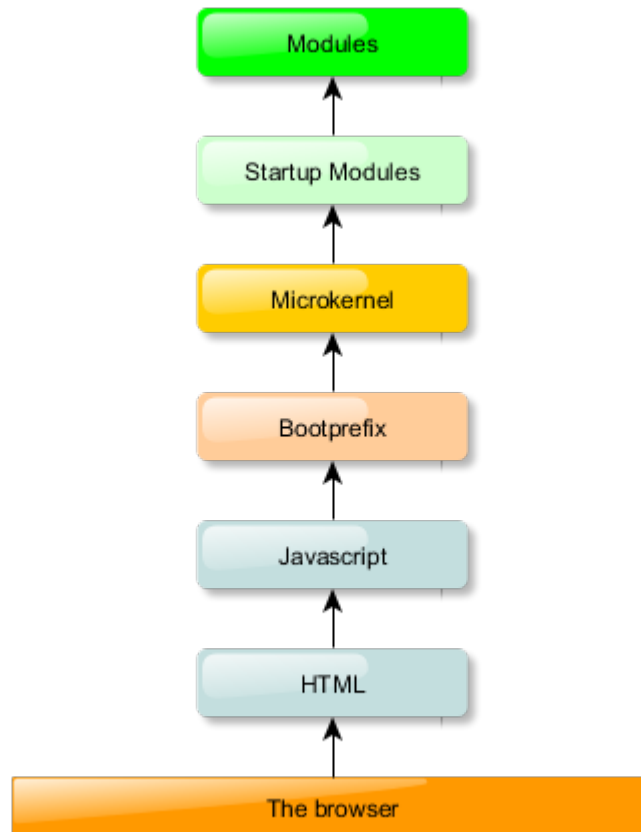
Loading tiddlywiki in the browser

2.1 Key points

- When the browser loads the Tiddlywiki file it runs its javascript code.
- The Bootprefix of the code loads the Microkernel.
- The Microkernel is the the only part of Tiddlywiki that is not managed by tiddlers.
- The rest of the application is managed by modules stored in tiddlers as javascript code.
- A tiddler can contain many kinds of data: text, images, javascript code (modules), JSON data...
- A tiddler can have many roles: a plugin, data, formatting code...

2.2 Introduction

Figure 2.1: Architecture of Tiddlywiki



If you edit the empty.html wiki file you will see something like this:

```
<html>
<head> ... </head>
<body>
<div id="styleArea">css styles</div>
<div id="storeArea">The tiddlers of the wiki</div>

<div id="bootKernelPrefix" ... type="application/javascript">
<script>
  var __bootprefix=(function($tw){
    "use strict";
    $tw=$tw||Object.create(null);
    ...
  }
</script>
</div>
...
</body>
</head>
```

So, at the bottom Tiddlywiki is a html page with a lot of javascript code inside it.

2.3 Architecture

What happens when loading a tiddlywiki file? In Figure2.1 we see its architecture.

1. First of all the operating system loads the browser.
2. The browser loads the HTML page.
3. After some css styles and the store area we can see the javascript code. The browser runs this code.
4. It loads the bootprefix. The bootprefix is responsible for preparing the kernel to boot on different engines e.g. browsers and node.js.
5. The microkernel is the first thing to run, when the application is started and it puts some initial objects and functions into the application tree, which are needed to load and manage tiddlers. After the microkernel built this initial application tree, the remaining parts of the application can be loaded as module tiddlers.
6. The microkernel load the startup modules (startup tiddlers).
7. At the top we have all modules (tiddlers).
8. Loading startup tiddlers

2.4 Tiddlers as Basic Coding Elements

Only a small part of the Tiddlywiki is not managed by tiddlers: the Microkernel. After the microkernel built this initial application tree, the remaining parts of the application can be loaded as module tiddlers.

A tiddler is the smallest unit of the TiddlyWiki system. It can contain any data like plain text, WikiText markup, JavaScript code (module tiddler), JSON structures (JSON structures might even contain additional tiddlers. Plug-ins are implemented this way to pack multiple tiddlers in a single plug-in tiddler), images in SVG format or even binary images encoded with base64. Internally Tiddlers are immutable objects containing a bunch of key:value pairs called fields. The only required field of a tiddler is the title field.

Chapter 3

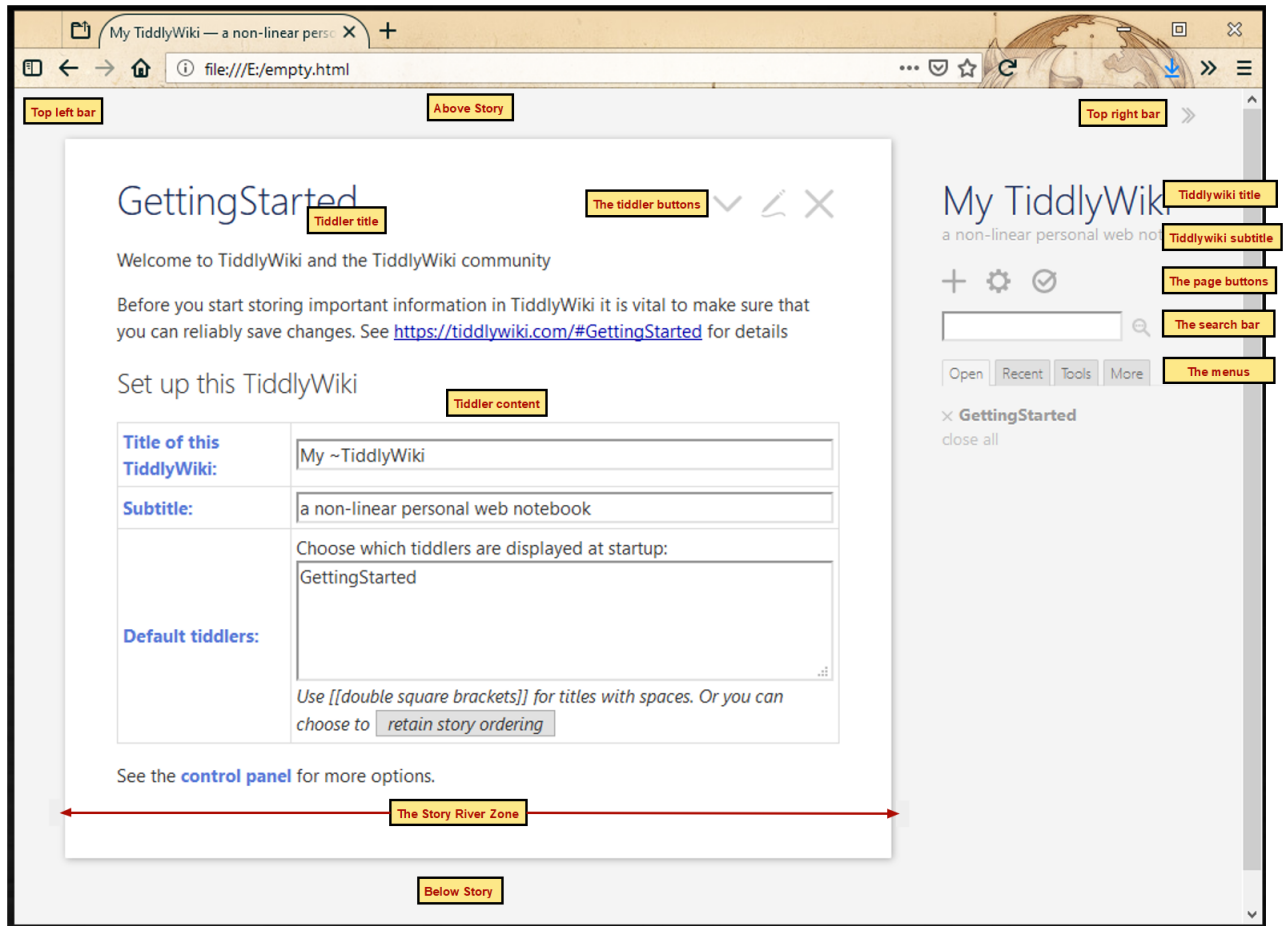
Components of the tiddlywiki screen

3.1 Key points

- Tiddlywiki page is divided in two main sides. The left is the Story River. The right, the sidebars
- In the left side appears the tiddlers you open.
- The right side shows the title, subtitle, buttons and the menus.
- All inside Tiddlywiki is made of tiddlers. A tiddler can have many roles: template, macro, widget, css formating code, etc. depending of its tag or content-type field.
- Tiddlywiki uses special tiddlers and special tags to configure its appearance.

3.2 The TW whole screen

Figure 3.1: The Tiddlywiki page



In Figure 3.1 we can see the Tiddlywiki elements. The most important things are the Story River on the left and the right elements: title, menus etc. The tiddlers that we open will appear in the navigation story on the left. With the right menus we can find all the tiddlers of the file.

These are the elements that we can see in the Tiddlywiki page. But in order to personalize our wiki we must go deeper. You have to know that all these elements are tiddlers: the title and subtitle of your Tiddlywiki are tiddlers. The right menus too. And the search text box is a tiddler too. Even the buttons are tiddlers. The top right and left bars, the Above story, the Below story, all are tiddlers, **system** and **shadow** tiddlers.

Inside Tiddlywiki, all is a tiddler. And you can search and open them. For example if you look for the Subtitle in the search bar in advanced mode (click in the Advanced search icon on the right of the search bar) you will find a tiddler called `$/SiteSubtitle`. It's a shadow tiddler. If you change its content, the Tiddlywiki subtitle will change. And if you look for the save button you will find the tiddler called `$/core/ui/Buttons/save` with the code of the button.

3.2.1 The story river

This place of the Tiddlywiki page is where all tiddlers you will open appear. You can configure the way Tiddlywiki will open them in the configuration tiddler (click the gearwheel icon on the right). At first all tiddlers opens each under the other but you can configure it so that only the active tiddler appears.

3.2.2 The right sidebars

The right place is where you can see the wiki title, subtitle, the page buttons, the search bar.... and under them the sidebars (the menus).

You can see 4 tabs: Open for the open tiddlers, Recent for the tiddlers you have opened, Tools to customize and configure your wiki and More where you can find some utilities. Of course they are four tiddlers: `$/core/ui/SideBar/Open`, `$/core/ui/SideBar/Recent`, `$/core/ui/SideBar/Tools` and `$/core/ui/SideBar/More`. All of them has the tag: `$/tags/SideBar`. So if you want to include your own tab create a tiddler and add to them this tag.

This is an important question about Tiddlywiki: The tiddlers that make up the screen are chosen with its tags. For example, above we have seen that if you put the tag `$/tags/SideBar` to a tiddler it appears on the left side. There are many tags for all places of the screen. You can see then in [Figure 3.2](#)

3.3 Drawing its interface

We know all are tiddlers but how Tiddlywiki draws its interface? This is where templates come in action.

To tell tiddlywiki where to place all elements we have a template: `$/core/ui/PageTemplate`. Try this: Edit this tiddler and add at the bottom of all code a new line with the code: `<hr><hr><hr>`. Then click the save tiddler button and close it. You can see three lines below the story river.

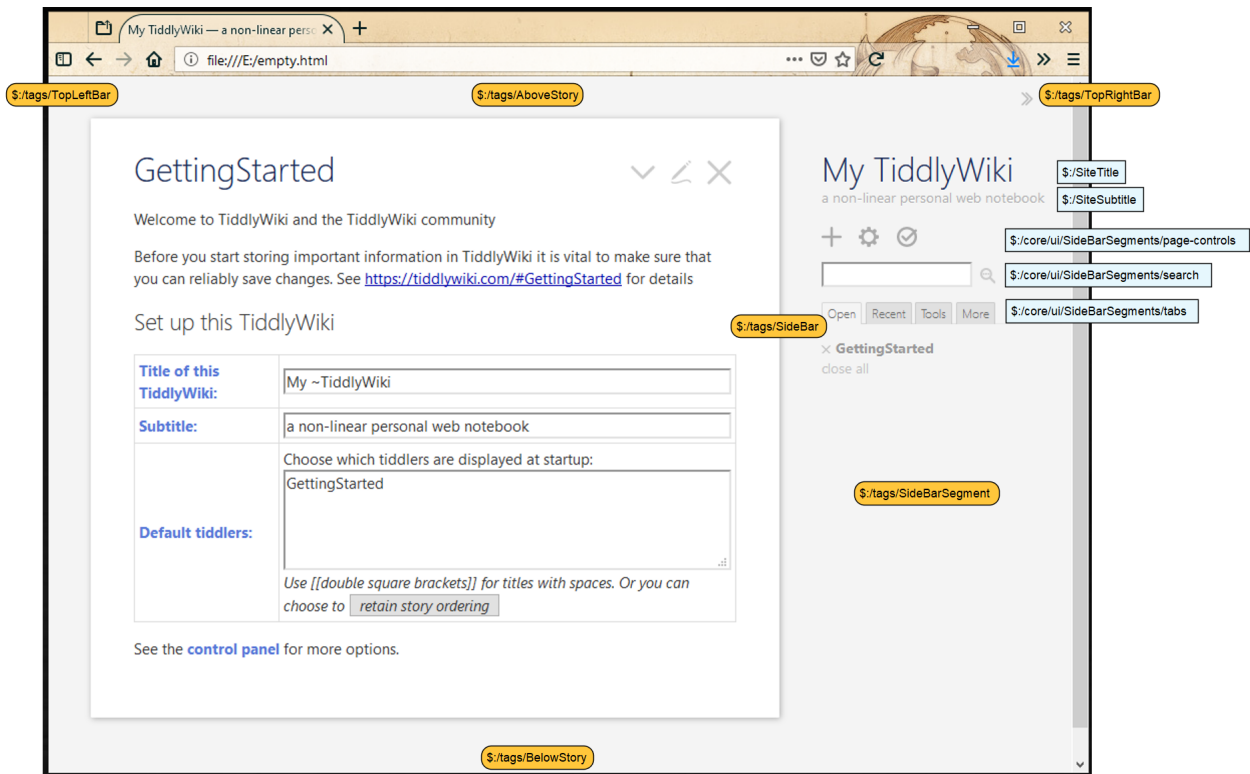
Looking this template we can see that it draws these elements:

1. `$/core/ui/PageTemplate/topleftbar`.- The top left bar
2. `$/core/ui/PageTemplate/toprightbar`.- The top right bar
3. `$/core/ui/PageTemplate/sidebar`.- The right side: title, subtitle, menus...
4. `$/core/ui/PageTemplate/story`.- All open tiddlers: the Story River
5. `$/core/ui/PageTemplate/alerts`.- Special tiddlers called alerts (tiddlers with the `$/tags/Alert` tag)

All elements are tagged with `$/tags/PageTemplate`.

The code of all these elements allows us to add additional elements very easily. We only have to create a new tiddler and tag it with the correct tag. For example, if we want to show some information above the story river we create a tiddler with this information and add the tag `$/tags/AboveStory`. The most important tags are in [Figure 3.2](#).

Figure 3.2: The tags of Tiddlywiki places



What's this? The light blue little squares shows the tiddlers containing the title, subtitle, buttons, search bars and tabs. And the orange bubbles show the tags. So, if you want to include:

- A top left bar: create a tiddler with the buttons and tagged it with `$:/tags/TopLeftBar`.
- A top right bar: There is a tiddler with this tag: `$:/tags/TopRightBar`: the tiddler `$:/core/ui/TopBar/menu`. It is used to hide the right zone.
- Content above the story river: Add this content in a tiddler tagged `$:/tags/AboveStory`.
- Content below the story river Add this content in a tiddler tagged `$:/tags/BelowStory`.
- A new tab in the right menus: Add this content in a tiddler tagged `$:/tags/SideBar`.
- New content below the right menu: Add this content in a tiddler tagged `$:/tags/SideBarSegment`.

Chapter 4

Customize the TW screen

4.1 Key points

- The PageTemplate is used to display the Tiddlywiki screen.
- It shows all tiddlers tagged with `$/tags/PageTemplate`. Its order is in the list field of the tiddler with the same title, `$/tags/PageTemplate`.
- The ViewTemplate is used to display a tiddler in view mode.
- It shows all tiddlers tagged with `$/tags/ViewTemplate`. Its order is in the list field of the tiddler with the same title, `$/tags/ViewTemplate`.
- The EditTemplate is used to display a tiddler when you are editing or creating.
- It shows all tiddlers tagged with `$/tags/EditTemplate`. Its order is in the list field of the tiddler with the same title, `$/tags/EditTemplate`.
- The text/css tiddlers tagged with `$/tags/Stylesheet` are used to format the text of all Tiddlywiki elements.

4.2 Introduction

One major feature of TiddlyWiki that many new users are unaware of is the degree to which TiddlyWiki can be customized, just by adding or removing SystemTags in key shadow tiddlers or in your own custom tiddlers.

- You can add and remove default features in tiddlers in either viewing or editing mode (let's say you find the tiddler subtitle distracting, or you want to add yourself a reminder that you will see when you edit tiddlers)
- You can also add and remove default features from the general page layout (maybe you want to add a clock to the sidebar, or replace one of the page control buttons with your own)
- You can also rearrange the order in which these features are displayed (perhaps you would like tags above tiddler titles, or the subtitle of your TiddlyWiki below the page control buttons)

Once you know what you are doing, all of these things are actually pretty easy to do.

There are three main templates in tiddlywiki:

- PageTemplate
- ViewTemplate
- EditTemplate

4.3 PageTemplate

The `$/core/ui/PageTemplate` tiddler is the template that draws the whole screen of tiddlywiki. It has the tag. All tiddlers **inside** this tiddler one are tagged with the `$/tags/-PageTemplate` tag.

What means“inside” this tiddler? If you look at this tiddler you can see this code:

```
<$list filter="[all [shadows+tiddlers]tag[$:/tags/PageTemplate]!has[draft.of]] " variable="listItem"><$transclude tiddler=<<listItem>>/></$list>
```

This is a list who transcludes (shows) all tiddlers tagged whit the `$/tags/PageTemplate` tag. If you tag a new tiddler with that tag it will be included in the main wiki page. You can find the sort of all the elements tagged in the list field of the tiddler `$/tags/-PageTemplate` (the same title as the tag). You can show your tiddler at the top of the screen if you include it as the first element of the list field in that tiddler.

What about the appearance of all this elements? Each element in the page has one or more CSS classes styling it. The CSS classes associated with the major page template elements are:

- Top left bar: `.tc-topbar .tc-topbar-left`

- Top right bar: `.tc-topbar .tc-topbar-right`
- Sidebar: `.tc-sidebar-scrollable`
- Story river: `.tc-story-river` Alerts: `.tc-alerts`

If user wants to hide a particular element from being displayed, they can create a stylesheet tiddler and add the style `display:none;` to the corresponding class.

CSS offers far more styling options than changing the visibility of an element. A complete description of CSS and its application to the each element is out of scope of this reference material. User is directed to familiarize themselves with various CSS properties from third party sources. All major browsers provide the options to inspect a displayed element on html page and view the CSS properties attached to that element. You may find the way to do it on your browser's website or forums. `display:none` property merely hides the display of the html element. It will not stop the element from loading to the DOM structure. Widespread use of the same will be a inefficient usage of resources.

4.4 ViewTemplate

This is the template Tiddlywiki uses to display a single tiddler when you are reading it. Its title is `$/core/ui/ViewTemplate`. As in the `PageTemplate`, you can edit it and look at its code:

```
<$list filter="[all[shadows+tiddlers]tag[$:/tags/ViewTemplate]!
  has[draft.of]]" variable="listItem">
```

```
<$transclude tiddler=<<listItem>>/>
```

```
</$list>
```

It shows a list with all tiddlers tagged with the `$/tags/ViewTemplate` tag sort with the list field of the tiddler `$/tags/ViewTemplate`.

4.4.1 Example

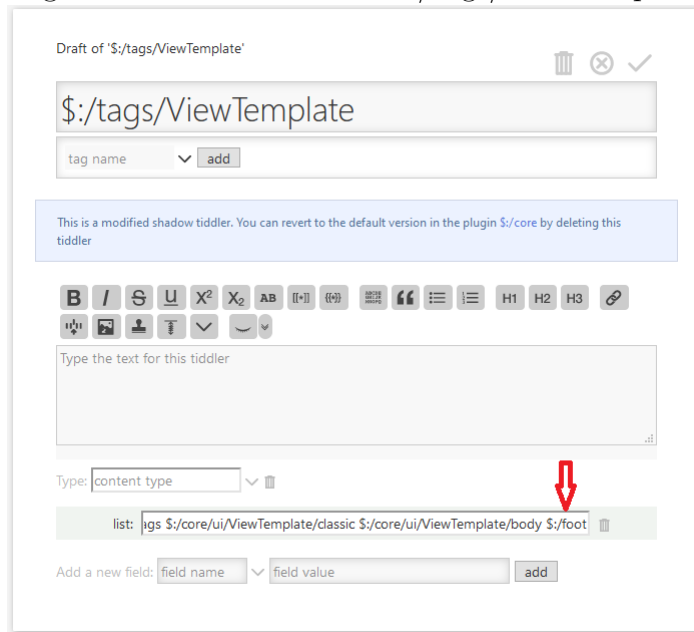
We want to illustrate the power of this templates. In this section we will add a foot at all tiddlers with the date in which the tiddler was added to the wiki.

The steps are:

- Add a new tiddler with the title `$/foot`
- Add this code to the tiddler:

```
<small>
  //(Added to the wiki: <$view field="created" format="date"
    template="DDth_MMM_YYYY"/>)//
</small>
```

Figure 4.1: The list field of \$:/tags/ViewTemplate



- Add the tag \$:/tags/ViewTemplate to the template
- Open the tiddler \$:/tags/ViewTemplate
- Add our tiddler, \$:/foot at the end of the list field of that tiddler, \$:/tags/ViewTemplate (look at Figure4.1).
- Save the tiddler.

You will see a foot in all tiddlers with the created date. In Figure4.2 you will see the new tiddler appearance.

4.5 EditTemplate

The \$:/core/ui/EditTemplate tiddler is the tiddler Tiddlywiki uses to display a tiddler when you are editing or creating it. And inside this tiddler you find:

```
<$list filter="[all[shadows+tiddlers]tag[$:/tags/EditTemplate]!
  has[draft.of]]" variable="listItem">
<$set name="tv-config-toolbar-class" filter="[<tv-config-toolbar
  -class> [<listItem>encodeuricomponent[] addprefix[tc-btn-]]">

<$transclude tiddler=<<listItem>>/>

</$set> </$list>
```

So it shows all tiddlers tagged with \$:/tags/EditTemplate sorting with the list field of the tiddler with the same name as this tag.

4.6 Using CSS in TW

You know HTML uses CSS to format the text. With CSS you can add colors, change font size, add borders and many other things.

If you open the control panel and show the Appearance tab you will see two themes: Snow White and Vanilla. And in Pallette tab you can choose many color combinations. Tiddlywiki uses CSS to change this settings. For example, the Vanilla base configuration is in the tiddler `$/themes/tiddlywiki/vanilla/base`. You can see that it is tagged with `$/tags/Stylesheet`.

Inside we find the css configuration for many html tags:

p: Paragraphs

h1: Title 1

h2: Title 2

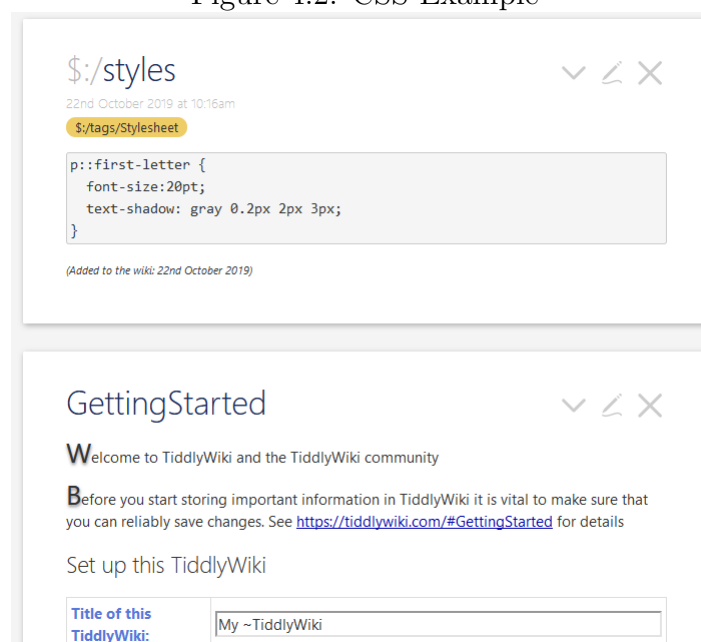
hr: Lines

table, td, tr: Tables

If you read this tiddler you will find all css clases for this theme.

4.6.1 Example

Figure 4.2: CSS Example



We can add a custom stylesheet tiddler for our own wikis. Imagine you want the first word of all paragraphs bigger. These are the steps:

- Create a tiddler, `$/styles`
- Add this content:

```
p::first-letter {  
  font-size:20pt;  
  text-shadow: gray 0.2px 2px 3px;  
}
```
- Add the tag `$/tags/Stylesheet` to our tiddler.
- If you want, you can add the Type, under the content: `text/css` (only for readability)

In [Figure 4.2](#) you see the style tiddler and the new appearance of the paragraphs.

Chapter 5

Inside a tiddler

5.1 Types of tiddlers

Tiddlers are the heart of Tiddlywiki. Once the Microkernel loads the tiddler functions (addTiddler(tdlr), deleteTiddler(tdlr), etc) the rest of the code is stored in tiddlers. This means that a tiddler can hold several roles, not only for our notes.

All tiddlers have a field. It is below the text in edit mode: Type. It is used to tell Tiddlywiki the type of the tiddler. But Tiddlywiki not only look this field to determine the role of the wiki. TW also look the tags assigned to the tiddler

How many of roles does a tiddler have?

Normal tiddler: It stores your notes and thinkings.

Tag tiddler: Is a tiddler with the same title as a tag.

Image tiddler: If you drag a picture into a Tiddlywiki it stores the image in a tiddler.

JSON tiddler: is a tiddler for storing data.

Dictionary tiddler: Is other way of storing data.

CSS tiddler: It stores the css rules for text formatting.

Template tiddler: It stores the template for other tiddler or even the whole Tiddlywiki screen.

Macro tiddler: a tiddler that stores macros.

Javascript tiddler: a tiddler with javascript code. It stores the tiddlywiki code and your widgets.

Role	Type field	Tag
Normal tiddler		
Tag tiddler		
GIF image tiddler	image/gif	
JPG image tiddler	image/jpeg	
PNG image tiddler	image/png	
ICO image tiddler	image/x-icon	
SVG image tiddler	image/svg+xml	
PDF tiddler	application/pdf	
JSON tiddler	application/json	
Dictionary tiddler	application/x-tiddler-dictionary	
CSS tiddler	text/css	\$/tags/Stylesheet
Template tiddler		
Macro tiddler		\$/tags/Macro
Widget	application/javascript	\$/tags/Macro
javascript code tiddler	application/javascript	

5.2 Normal tiddler

5.2.1 Formatting text

5.2.2 Linking content

5.2.3 Basic transclusion

5.2.4 Showing external content

as mime types (videos, images, pdfs), as html elements (video controls...,)

5.3 Adding macros and widgets

Here you can find basic information. Advanced information in later chapters.

5.4 CSS tiddlers

Customizing the appearance of your wiki with CSS tiddlers.

5.5 Alert tiddlers

5.6 Tag tiddlers

A way of ranking tags. A possible use: sidebar index. The color and icon of a tag.

5.7 Data tiddlers

- JSON and Dictionary tiddlers.
- Refering the data.

5.8 Macro tiddlers

Wikitext macros and javascript macros.

Chapter 6

The filter notation

6.1 The filter basic step

6.2 Using filter operators

6.3 Combining filter steps

- combining by OR
- combining by AND
- combining by NOT

6.4 Regular expressions

6.5 Filter uses

6.6 Examples

Chapter 7

The behavior of tiddlywiki

7.1 Creating microcontent

7.2 Organizing microcontent with links

7.3 Organizing tiddlers with tags

7.4 Personalize behavior with macros and widgets

Chapter 8

Macros

Customizing your tiddler functionality with macros.

8.1 Basic macros included

8.2 Writing your own macros

8.3 Using macros

Chapter 9

Widgets

Customizing your tiddler functionality with widgets.

9.1 Basic widgets included

9.2 Writting your own widgets

This question is out of the goal of this book. We show how to add javascript code to your tiddler.

9.3 Using widgets

Chapter 10

The scripting business

Pieces of code.

10.1 List scripting

Lists, nested lists, actions and buttons inside lists...

10.2 Interface scripting

How to use buttons, edittext, radiobuttons, checkboxes...

10.3 Data tiddlers scripting

How to use data tiddlers.

10.4 Template scripting

Revealing sections of tiddlywiki, state tiddlers...

10.5 Tiddlers scripting

Creating, removing and editing tiddlers with code, navigating with code...

Chapter 11

Recipe book

In this section we will show master code pieces to various situations. Personal wikis with many uses.

Maybe the sections will be:

11.1 Personal todo-list

11.2 Writting stories

11.3 Image gallery

11.4 Lesson planner

11.5 Simple game

Chapter 12

Languages

12.1 Tiddlywiki languages

12.2 Writing international wikis

Chapter 13

Plugins

13.1 Extending tiddlywiki functionality

13.2 Creating plugins

13.3 Where we can find plugins

Chapter 14

The people. The project

I want to add a chapter with all the people under this project. From Jeremy though the core developers to the wiki writers.

And the project, its philosophy and the future of tiddlywiki.