



# Inhaltsverzeichnis

- Basic Text Formatting** ..... 3
- Links** ..... 4
  - External ..... 4
  - Internal ..... 4
  - Interwiki ..... 5
  - Windows Shares ..... 5
  - Image Links ..... 5
- Footnotes** ..... 6
- Sectioning** ..... 6
  - Headline Level 3 ..... 6
- Images and Other Files** ..... 7
- Lists** ..... 8
- Text Conversions** ..... 9
  - Text to Image Conversions ..... 9
  - Text to HTML Conversions ..... 10
- Quoting** ..... 10
- Tables** ..... 11
- No Formatting** ..... 13
- Code Blocks** ..... 13
  - Syntax Highlighting ..... 14
  - Downloadable Code Blocks ..... 14
- Embedding HTML and PHP** ..... 15
- RSS/ATOM Feed Aggregation** ..... 16
- Control Macros** ..... 17
- Syntax Plugins** ..... 17



# Formatting Syntax

 [DokuWiki](#) supports some simple markup language, which tries to make the datafiles to be as readable as possible. This page contains all possible syntax you may use when editing the pages. Simply have a look at the source of this page by pressing the *Edit this page* button at the top or bottom of the page. If you want to try something, just use the [playground](#) page. The simpler markup is easily accessible via  [quickbuttons](#), too.

## Basic Text Formatting

DokuWiki supports **bold**, *italic*, underlined and monospaced texts. Of course you can **combine** all these.

```
DokuWiki supports **bold**, //italic//, __underlined__ and
'monospaced' texts.
Of course you can **__//''combine''//__** all these.
```

You can use <sub>subscript</sub> and <sup>superscript</sup>, too.

```
You can use <sub>subscript</sub> and <sup>superscript</sup>, too.
```

You can mark something as ~~deleted~~ as well.

```
You can mark something as <del>deleted</del> as well.
```

**Paragraphs** are created from blank lines. If you want to **force a newline** without a paragraph, you can use two backslashes followed by a whitespace or the end of line.

This is some text with some linebreaks

Note that the two backslashes are only recognized at the end of a line  
or followed by  
a whitespace \\this happens without it.

```
This is some text with some linebreaks\\ Note that the
two backslashes are only recognized at the end of a line\\
or followed by\\ a whitespace \\this happens without it.
```

You should use forced newlines only if really needed.

# Links

DokuWiki supports multiple ways of creating links.

## External


External links are recognized automatically: <http://www.google.com> or simply [www.google.com](http://www.google.com) - You can set the link text as well: [This Link points to google](#). Email addresses like this one: [andi@splitbrain.org](mailto:andi@splitbrain.org) are recognized, too.

```
DokuWiki supports multiple ways of creating links. External links are
recognized
automagically: http://www.google.com or simply www.google.com - You
can set
link text as well: [[http://www.google.com|This Link points to
google]]. Email
addresses like this one: <andi@splitbrain.org> are recognized, too.
```

## Internal

Internal links are created by using square brackets. You can either just give a [pagename](#) or use an additional [link text](#).

```
Internal links are created by using square brackets. You can either
just give
a [[pagename]] or use an additional [[pagename|link text]].
```

 [Wiki pagenames](#) are converted to lowercase automatically, special characters are not allowed.

You can use [namespaces](#) by using a colon in the pagename.

```
You can use [[some:namespaces]] by using a colon in the pagename.
```

For details about namespaces see  [namespaces](#).

Linking to a specific section is possible, too. Just add the section name behind a hash character as known from HTML. This links to [this Section](#).

```
This links to [[syntax#internal|this Section]].
```

Notes:

- Links to [existing pages](#) are shown in a different style from [nonexisting](#) ones.
- DokuWiki does not use [CamelCase](#) to automatically create links by default, but this behavior can be enabled in the [config](#) file. Hint: If DokuWiki is a link, then it's enabled.
- When a section's heading is changed, its bookmark changes, too. So don't rely on section linking too much.

## Interwiki

DokuWiki supports [Interwiki](#) links. These are quick links to other Wikis. For example this is a link to Wikipedia's page about Wikis: [Wiki](#).

```
DokuWiki supports [[doku>Interwiki]] links. These are quick links to other Wikis.
For example this is a link to Wikipedia's page about Wikis:
[[wp>Wiki]].
```

## Windows Shares

Windows shares like [this](#) are recognized, too. Please note that these only make sense in a homogeneous user group like a corporate [Intranet](#).

```
Windows Shares like [[\\server\share|this]] are recognized, too.
```

Notes:

- For security reasons direct browsing of windows shares only works in Microsoft Internet Explorer per default (and only in the „local zone“).
- For Mozilla and Firefox it can be enabled through different workaround mentioned in the [Mozilla Knowledge Base](#). However, there will still be a JavaScript warning about trying to open a Windows Share. To remove this warning (for all users), put the following line in `conf/local.protected.php`:

```
$lang['js']['nosmblinks'] = '';
```

## Image Links

You can also use an image to link to another internal or external page by combining the syntax for links and [images](#) (see below) like this:

```
[[http://www.php.net|{{wiki:dokuwiki-128.png}}]]
```



Please note: The image formatting is the only formatting syntax accepted in link names.

The whole [image](#) and [link](#) syntax is supported (including image resizing, internal and external images and URLs and interwiki links).

## Footnotes

You can add footnotes <sup>1)</sup> by using double parentheses.

You can add footnotes ((This is a footnote)) by using double parentheses.

## Sectioning

You can use up to five different levels of headlines to structure your content. If you have more than three headlines, a table of contents is generated automatically – this can be disabled by including the string `~~NOTOC~~` in the document.

### Headline Level 3

#### Headline Level 4

##### Headline Level 5

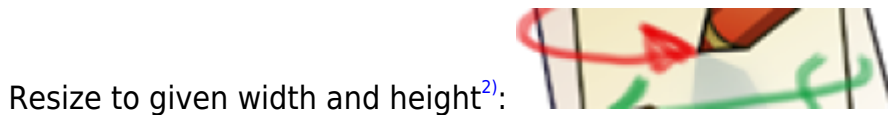
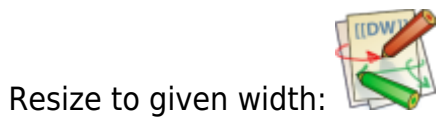
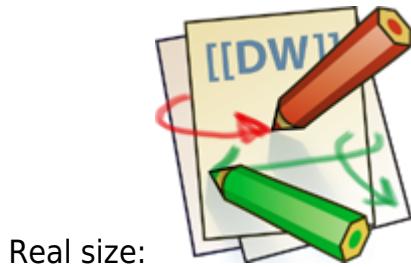
```
==== Headline Level 3 ====  
=== Headline Level 4 ===  
== Headline Level 5 ==
```


By using four or more dashes, you can make a horizontal line:



# Images and Other Files

You can include external and internal  images with curly brackets. Optionally you can specify the size of them.



Resized external image: 

```
Real size:                {{wiki:dokuwiki-128.png}}
Resize to given width:    {{wiki:dokuwiki-128.png?50}}
Resize to given width and height: {{wiki:dokuwiki-128.png?200x50}}
Resized external image:  {{http://de3.php.net/images/php.gif?200x50}}
```

By using left or right whitespaces you can choose the alignment.





```
{{ wiki:dokuwiki-128.png}}
{{wiki:dokuwiki-128.png }}
{{ wiki:dokuwiki-128.png }}
```

Of course, you can add a title (displayed as a tooltip by most browsers), too.



```
{{ wiki:dokuwiki-128.png |This is the caption}}
```

If you specify a filename (external or internal) that is not an image (gif, jpeg, png), then it will be displayed as a link instead.

For linking an image to another page see [Image Links](#) above.

## Lists

Dokuwiki supports ordered and unordered lists. To create a list item, indent your text by two spaces and use a \* for unordered lists or a - for ordered ones.

- This is a list
- The second item
  - You may have different levels
- Another item

1. The same list but ordered
2. Another item
  1. Just use indentation for deeper levels
3. That's it

```
* This is a list
* The second item
  * You may have different levels
```

- \* Another item
- The same list but ordered
- Another item
  - Just use indentation for deeper levels
- That's it

Also take a look at the [FAQ on list items](#).











## Text Conversions











DokuWiki can convert certain pre-defined characters or strings into images or other text or HTML.

The text to image conversion is mainly done for smileys. And the text to HTML conversion is used for typography replacements, but can be configured to use other HTML as well.

### Text to Image Conversions

DokuWiki converts commonly used [emoticons](#) to their graphical equivalents. Those [Smileys](#) and other images can be configured and extended. Here is an overview of Smileys included in DokuWiki:

-  8-)
-  8-O
-  :-)
-  :-)
-  =)
-  :-/
-  :-\
-  :-?
-  :-D
-  :-P

-  :-O
-  :-X
-  :-|
-  ;-)
-  ^\_^
-  :?:
-  !:
-  LOL
-  FIXME
-  DELETEME

### Text to HTML Conversions

Typography: [DokuWiki](#) can convert simple text characters to their typographically correct entities. Here is an example of recognized characters.

→ ← ↔ ⇒ ⇐ ⇔ » « - — 640x480 © ™ ® „He thought 'It's a man's world'...”

```
-> <- <-> => <=<=> >> << -- --- 640x480 (c) (tm) (r)
"He thought 'It's a man's world'..."
```

The same can be done to produce any kind of HTML, it just needs to be added to the [pattern file](#).

There are three exceptions which do not come from that pattern file: multiplication entity (640x480), 'single' and „double quotes“. They can be turned off through a [config option](#).

### Quoting

Some times you want to mark some text to show it's a reply or comment. You can use the following syntax:

```
I think we should do it
```

```
> No we shouldn't
>> Well, I say we should
> Really?
>> Yes!
>>> Then lets do it!
```

I think we should do it

```
No we shouldn't
| Well, I say we should
Really?
| Yes!
| | Then lets do it!
```

## Tables

DokuWiki supports a simple syntax to create tables.

Heading 1	Heading 2	Heading 3
Row 1 Col 1	Row 1 Col 2	Row 1 Col 3
Row 2 Col 1	some colspan (note the double pipe)	
Row 3 Col 1	Row 3 Col 2	Row 3 Col 3

Table rows have to start and end with a | for normal rows or a ^ for headers.

```
^ Heading 1      ^ Heading 2      ^ Heading 3      ^
| Row 1 Col 1    | Row 1 Col 2    | Row 1 Col 3    |
| Row 2 Col 1    | some colspan (note the double pipe) ||
| Row 3 Col 1    | Row 3 Col 2    | Row 3 Col 3    |
```

To connect cells horizontally, just make the next cell completely empty as shown above. Be sure to have always the same amount of cell separators!

Vertical tableheaders are possible, too.

	Heading 1	Heading 2
Heading 3	Row 1 Col 2	Row 1 Col 3

	<b>Heading 1</b>	<b>Heading 2</b>
<b>Heading 4</b>	no colspan this time	
<b>Heading 5</b>	Row 2 Col 2	Row 2 Col 3

As you can see, it's the cell separator before a cell which decides about the formatting:

```
|           ^ Heading 1           ^ Heading 2           ^
^ Heading 3 | Row 1 Col 2           | Row 1 Col 3           |
^ Heading 4 | no colspan this time |                       |
^ Heading 5 | Row 2 Col 2           | Row 2 Col 3           |
```

You can have rowspans (vertically connected cells) by adding ::: into the cells below the one to which they should connect.

<b>Heading 1</b>	<b>Heading 2</b>	<b>Heading 3</b>
Row 1 Col 1		Row 1 Col 3
Row 2 Col 1	this cell spans vertically	Row 2 Col 3
Row 3 Col 1		Row 2 Col 3

Apart from the rowspan syntax those cells should not contain anything else.

```
^ Heading 1     ^ Heading 2           ^ Heading 3           ^
| Row 1 Col 1   | this cell spans vertically | Row 1 Col 3           |
| Row 2 Col 1   | :::                       | Row 2 Col 3           |
| Row 3 Col 1   | :::                       | Row 2 Col 3           |
```

You can align the table contents, too. Just add at least two whitespaces at the opposite end of your text: Add two spaces on the left to align right, two spaces on the right to align left and two spaces at least at both ends for centered text.

<b>Table with alignment</b>		
right	center	left
left	right	center
xxxxxxxxxxxxxx	xxxxxxxxxxxxxx	xxxxxxxxxxxxxx

This is how it looks in the source:

```
^           Table with alignment           ^^^
|           right|           center |left           |
|left           |           right|           center |
| xxxxxxxxxxxxxx | xxxxxxxxxxxxxx | xxxxxxxxxxxxxx |
```

Note: Vertical alignment is not supported.

## No Formatting

If you need to display text exactly like it is typed (without any formatting), enclose the area either with `<nowiki>` tags or even simpler, with double percent signs `%%`.

This is some text which contains addresses like this: `http://www.splitbrain.org` and `**formatting**`, but nothing is done with it. The same is true for `//__this__ text//` with a smiley `;-)`.

```
<nowiki>
This is some text which contains addresses like this:
http://www.splitbrain.org and **formatting**, but nothing is done
with it.
</nowiki>
The same is true for %%//__this__ text// with a smiley ;-)%%.
```

## Code Blocks

You can include code blocks into your documents by either indenting them by at least two spaces (like used for the previous examples) or by using the tags `<code>` or `<file>`.

```
This is text is indented by two spaces.
```

```
This is preformatted code all spaces are preserved: like
<-this
```

```
This is pretty much the same, but you could use it to show that you
quoted a file.
```

Those blocks were created by this source:

```
This is text is indented by two spaces.
```

```
<code>
This is preformatted code all spaces are preserved: like
<-this
</code>
```

```
<file>
This is pretty much the same, but you could use it to show that you
quoted a file.
</file>
```

## Syntax Highlighting

[DokuWiki](#) can highlight sourcecode, which makes it easier to read. It uses the [GeSHi](#) Generic Syntax Highlighter – so any language supported by GeSHi is supported. The syntax is the same like in the code and file blocks in the previous section, but this time the name of the used language is inserted inside the tag. Eg. `<code java>` or `<file java>`.

```
/**
 * The HelloWorldApp class implements an application that
 * simply displays "Hello World!" to the standard output.
 */
class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello World!"); //Display the string.
    }
}
```

The following language strings are currently recognized: *4cs, abap, actionscript-french, actionscript, actionscript3, ada, apache, applescript, asm, asp, autoconf, autohotkey, autoit, avisynth, awk, bash, basic4gl, bf, bibtex, blitzbasic, bnf, boo, c, c\_mac, caddcl, cadlisp, cfdg, cfm, chaiscript, cil, clojure, cmake, cobol, cpp, cpp-qt, csharp, css, cuesheet, d, dcs, delphi, diff, div, dos, dot, ecmascript, eiffel, email, erlang, fo, fortran, freebasic, fsharp, gambas, genero, genie, gdb, glsl, gml, gnuplot, groovy, gettext, gwbasic, haskell, hicest, hq9plus, html, icon, idl, ini, inno, intercal, io, j, java5, java, javascript, jquery, kixtart, klonec, klonecpp, latex, lisp, locobasic, logtalk, lolcode, lotusformulas, lotusscript, lscript, lsl2, lua, m68k, magiksf, make, mapbasic, matlab, mirc, modula2, modula3, mmix, mpasm, mxml, mysql, newlisp, nsis, oberon2, objc, ocaml-brief, ocaml, oobas, oracle8, oracle11, oxygene, oz, pascal, pcre, perl, perl6, per, pf, php-brief, php, pike, pic16, pixelbender, plsql, postgresql, povray, powerbuilder, powershell, progress, prolog, properties, providex, purebasic, python, q, qbasic, rails, rebol, reg, robots, rpmspec, rsplus, ruby, sas, scala, scheme, scilab, sdlbasic, smalltalk, smarty, sql, systemverilog, tcl, teraterm, text, thinbasic, tsql, typoscript, unicon, vala, vbnet, vb, verilog, vhdl, vim, visualfoxpro, visualprolog, whitespace, winbatch, whois, xbasic, xml, xorg\_conf, xpp, z80*

## Downloadable Code Blocks

When you use the `<code>` or `<file>` syntax as above, you might want to make the shown code available for download as well. You can do this by specifying a file name after language code like this:

```
<file php myexample.php>
<?php echo "hello world!"; ?>
</file>
```

myexample.php

```
<?php echo "hello world!"; ?>
```

If you don't want any highlighting but want a downloadable file, specify a dash (-) as the language code: `<code - myfile.foo>`.

## Embedding HTML and PHP

You can embed raw HTML or PHP code into your documents by using the `<html>` or `<php>` tags. (Use uppercase tags if you need to enclose block level elements.)

HTML example:

```
<html>
This is some <span style="color:red;font-size:150%;">inline
HTML</span>
</html>
<HTML>
<p style="border:2px dashed red;">And this is some block HTML</p>
</HTML>
```

```
<html> This is some inline HTML </html> <HTML> <p style=„border:2px dashed red;“>And
this is some block HTML</p> </HTML>
```

PHP example:

```
<php>
echo 'A logo generated by PHP: ';
echo '<img src="" . $_SERVER['PHP_SELF'] . '?=' . php_logo_guid() .
'" alt="PHP Logo !" />';
echo '(generated inline HTML)';
</php>
<PHP>
echo '<table class="inline"><tr><td>The same, but inside a block
level element:</td>';
echo '<td><img src="" . $_SERVER['PHP_SELF'] . '?=' . php_logo_guid()
. '" alt="PHP Logo !" /></td>';
echo '</tr></table>';
</PHP>
```

```
<php> echo 'A logo generated by PHP: '; echo '<img src=„' . $_SERVER['PHP_SELF'] . '?=' .
php_logo_guid() . '" alt=„PHP Logo !" />'; echo '(inline HTML)'; </php> <PHP> echo '<table
class=„inline“><tr><td>The same, but inside a block level element:</td>'; echo
```

```
'<td><img src=„' . $_SERVER['PHP_SELF'] . '?=' . php_logo_guid() . '" alt=„PHP Logo !“
/></td>'; echo '</tr></table>'; </PHP>
```

**Please Note:** HTML and PHP embedding is disabled by default in the configuration. If disabled, the code is displayed instead of executed.

## RSS/ATOM Feed Aggregation

[DokuWiki](#) can integrate data from external XML feeds. For parsing the XML feeds, [SimplePie](#) is used. All formats understood by SimplePie can be used in DokuWiki as well. You can influence the rendering by multiple additional space separated parameters:

Parameter	Description
any number	will be used as maximum number items to show, defaults to 8
reverse	display the last items in the feed first
author	show item authors names
date	show item dates
description	show the item description. If <a href="#">HTML</a> is disabled all tags will be stripped
n[dhm]	refresh period, where d=days, h=hours, m=minutes. (e.g. 12h = 12 hours).

The refresh period defaults to 4 hours. Any value below 10 minutes will be treated as 10 minutes. [DokuWiki](#) will generally try to supply a cached version of a page, obviously this is inappropriate when the page contains dynamic external content. The parameter tells [DokuWiki](#) to re-render the page if it is more than *refresh period* since the page was last rendered.

### Example:

```
{ {rss>http://slashdot.org/index.rss 5 author date 1h } }
```


- [Anthropic 'Suspends' All Mythos and Fable Access After US Order Limiting Foreign Access](#) von EditorDavid (13.06.2026 07:00)
- [Data Center Opponents Have Blocked Or Delayed Projects Worth Nearly \\$130 Billion In 2026](#) von BeauHD (13.06.2026 03:30)
- [Jeff Bezos' AI Startup Aims To Build an 'Artificial General Engineer'](#) von BeauHD (12.06.2026 23:00)
- [Justice Department Approves Paramount's \\$111 Billion Acquisition of Warner Bros.](#) von BeauHD (12.06.2026 22:00)
- [ShinyHunters Hacked 100+ Organizations By Exploiting an Oracle PeopleSoft 0-Day](#) von BeauHD (12.06.2026 21:20)

## Control Macros

Some syntax influences how DokuWiki renders a page without creating any output it self. The following control macros are available:

Macro	Description
~~NOTOC~~	If this macro is found on the page, no table of contents will be created
~~NOCACHE~~	DokuWiki caches all output by default. Sometimes this might not be wanted (eg. when the <php> syntax above is used), adding this macro will force DokuWiki to re-render a page on every call

## Syntax Plugins

DokuWiki's syntax can be extended by  **Plugins**. How the installed plugins are used is described on their appropriate description pages. The following syntax plugins are available in this particular DokuWiki installation:

- [Bookcreator Plugin](#) 2024-02-07 von [Gerrit Uitslag \(previous: Luigi Micco\)](#)  
Allow to make a book (PDF, ODT, HTML or text) from selected pages
- [Bureaucracy Plugin](#) 2025-03-06 von [Andreas Gohr](#)  
Create forms and generate pages or emails from them
- [comment](#) 2023-08-18 von [Gina Häußge, Michael Klier](#)  
Add comments/notes to your wiki source that won't be shown on the page.
- [const plugin](#) 2023-04-04 von [lisps, peterfromearth](#)  
work on wiki pages with constants, self defined variables and make calculations
- [Structured Data Plugin](#) 2024-01-30 von [Andreas Gohr](#)  
Add and query structured data in your wiki
- [dbquery plugin](#) 2025-07-24 von [Andreas Gohr](#)  
Display query results from a database
- [Definition List plugin](#) 2017-08-17 von [Chris Smith, LarsDW223](#)  
Add syntax for definition lists.
- [discussion plugin](#) 2025-07-04 von [Michael Hamann, Gerrit Uitslag, Gina Häussge, Christopher Smith, Michael Klier, Esther Brunner, Matthias Schulte](#)  
Provides discussion functionality for wiki pages
- [Dw2Pdf plugin](#) 2026-01-08 von [Andreas Gohr and Luigi Micco](#)  
DokuWiki to PDF converter
- [Filelist Plugin](#) 2024-07-03 von [Gina Häußge, Dokufreaks](#)  
Lists files matching a given glob pattern.
- [folded plugin](#) 2023-07-22 von [Michael Hamann, Fabian van-de-I\\_Isle, Christopher Smith, Esther Brunner](#)  
Foldable page sections
- [Gallery Plugin](#) 2024-04-30 von [Andreas Gohr](#)  
Creates a gallery of images from a namespace or RSS/ATOM feed
- [hidden Plugin](#) 2023-11-07 von [Guillaume Turri](#)  
Enable to hide/show details

- [Color Highlight Plugin](#) 2016-07-22 von [Joseph Nahmias](#)  
Highlight text with a specific color - Syntax: `<hi color>highlighted content</hi>`
- [include plugin](#) 2025-07-22 von [Michael Hamann](#), [Gina Häussge](#), [Christopher Smith](#), [Michael Klier](#), [Esther Brunner](#)  
Functions to include another page in a wiki page
- [Indexmenu Plugin](#) 2024-01-05 von [Samuele Tognini](#), [Gerrit Uitslag](#)  
Show a customizable and sortable index for a namespace
- [Info Plugin](#) 2020-06-04 von [Andreas Gohr](#)  
Displays information about various DokuWiki internals
- [Math Plugin](#) 2024-05-23 von [Christopher Smith](#)  
Add mathematical formulae to dokuwiki. Syntax: `<m size>math formulae</m>`
- [Open Document Plugin](#) 2023-03-03 von [Andreas Gohr](#), [Aurelien Bompard](#), [Florian Lamml](#), [LarsDW223](#)  
Export the current Wiki page to a OpenOffice ODT file
- [Pagelist Plugin](#) 2023-08-27 von [Matthias Schulte](#), [Michael Hamann](#), [Michael Klier](#), [Gina Häussge](#)  
Lists pages in a nice formatted way
- [Patch Panel Plugin](#) 2015-02-03 von [Grant Emsley](#)  
Draw patch panel from text description
- [PlantUML Parser](#) 2023-11-02 von [Kyle Carter](#)  
This plugin takes PlantUML markup from a wiki page and has it displayed in a user's browser.
- [simplenavi plugin](#) 2026-02-02 von [Andreas Gohr](#)  
Create a simple navigation tree based on namespaces
- [Snippets Plugin](#) 2022-10-07 von [Myron Turner](#)  
Provides toolbar button+popup for template insertion; can track and insert snippet updates
- [Struct Sections](#) 2025-11-04 von [Michael Große](#)  
New type to show struct wiki fields as actual sections of the page
- [structstatus plugin](#) 2023-06-21 von [Andreas Gohr](#)  
Display a document status using struct data
- [struct plugin](#) 2025-09-24 von [Andreas Gohr](#), [Michael Große](#), [Anna Dabrowska](#)  
Add and query additional structured page data
- [Tag Plugin](#) 2023-10-17 von [Michael Hamann](#), [Gina Häussge](#), [Christopher Smith](#), [Michael Klier](#), [Esther Brunner](#)  
tag wiki pages
- [Typography plugin](#) 2020-07-31 von [Satoshi Sahara](#)  
Enable typesetting capabilities with CSS font properties such as font face, size, weight, and color of text
- [Wrap Plugin](#) 2023-08-13 von [Anika Henke](#)  
Universal plugin which combines functionalities of many other plugins. Wrap wiki text inside containers (divs or spans) and give them a class (choose from a variety of preset classes), a width and/or a language with its associated text direction.

1)

This is a footnote

2)

when the aspect ratio of the given width and height doesn't match that of the image, it will be cropped to the new ratio before resizing