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 «Edit this 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 subscript and superscript, 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 automagically: http://www.google.com or simply www.google.com - You can set the link text as well: This Link points to google. Email addresses like this one: 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/lang/en/lang.php` (more details at localization):

  conf/lang/en/lang.php

  ```
  <?php
  /**
   * Customization of the english language file
   * Copy only the strings that needs to be modified
   */
  $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://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 [1] 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:

Media Files

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

Real size: 

Resize to given width: 

Resize to given width and height[2)]: 

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://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}}
```

For linking an image to another page see Image Links above.

Supported Media Formats

DokuWiki can embed the following media formats directly.

Image	`gif`, `jpg`, `png`
Video	`webm`, `ogv`, `mp4`
Audio	`ogg`, `mp3`, `wav`
Flash	`swf`

If you specify a filename that is not a supported media format, then it will be displayed as a link instead.

By adding `?linkonly` you provide a link to the media without displaying it inline

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

dokuwiki-128.png This is just a link to the image.

Fallback Formats

Unfortunately not all browsers understand all video and audio formats. To mitigate the problem, you can upload your file in different formats for maximum browser compatibility.

For example consider this embedded mp4 video:

```
{{video.mp4|A funny video}}
```

When you upload a `video.webm` and `video.ogv` next to the referenced `video.mp4`, DokuWiki will automatically add them as alternatives so that one of the three files is understood by your browser.

Additionally DokuWiki supports a «poster» image which will be shown before the video has started. That image needs to have the same filename as the video and be either a jpg or png file. In the example above a `video.jpg` file would work.

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 indention 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 indention 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
- Fix Me! FIXME
- Delete! DELETEME

Text to HTML Conversions

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

→ ← ↔ ⇒ ⇐ ⇔ » « – — 640×480 © ™ ® «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 (640×480), '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
Heading 4	no colspan this time	
Heading 5	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.

Heading 1	Heading 2	Heading 3
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.

Table with alignment		
right	center	left
left	right	center
xxxxxxxxxxxx	xxxxxxxxxxxx	xxxxxxxxxxxx

This is how it looks in the source:

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

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 uses the same code and file blocks described in the previous section, but this time the name of the language syntax to be highlighted is included inside the tag, e.g. `<code java>` or `<file java>`.

```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 6502acme 6502kickass 6502tasm 68000devpac abap actionscript3 actionscript ada aimms algol68 apache applescript apt_sources arm asm asp asymptote autoconf autohotkey autoit avisynth awk bascomavr bash basic4gl batch bf biblatex bibtex blitzbasic bnf boo caddcl cadlisp ceylon cfdg cfm chaiscript chapel cil c_loadrunner clojure c_mac cmake cobol coffeescript c cpp cpp-qt cpp-winapi csharp css cuesheet c_winapi dart dcl dcpu16 dcs delphi diff div dos dot d ecmascript eiffel email epc e erlang euphoria ezt f1 falcon fo fortran freebasic freeswitch fsharp gambas gdb genero genie gettext glsl gml gnuplot go groovy gwbasic haskell haxe hicest hq9plus html html4strict html5 icon idl ini inno intercal io ispfpanel java5 java javascript jcl j jquery julia kixtart klonec klonecpp kotlin latex lb ldif lisp llvm locobasic logtalk lolcode lotusformulas lotusscript lscript lsl2 lua m68k magiksf make mapbasic mathematica matlab mercury metapost mirc mk-61 mmix modula2 modula3 mpasm mxml mysql nagios netrexx newlisp nginx nimrod nsis oberon2 objc objeck ocaml-brief ocaml octave oobas oorexx oracle11 oracle8 oxygene oz parasail parigp pascal pcre perl6 perl per pf phix php-brief php pic16 pike pixelbender pli plsql postgresql postscript povray powerbuilder powershell proftpd progress prolog properties providex purebasic pycon pys60 python qbasic qml q racket rails rbs rebol reg rexx robots rpmspec rsplus ruby rust sas sass scala scheme scilab scl sdlbasic smalltalk smarty spark sparql sql standardml stonescript swift systemverilog tclegg tcl teraterm texgraph text thinbasic tsql twig typoscript unicon upc urbi uscript vala vbnet vb vbscript vedit verilog vhdl vim visualfoxpro visualprolog whitespace whois winbatch xbasic xml xojo xorg_conf xpp yaml z80 zxbasic*

There are additional advanced options available for syntax highlighting, such as highlighting lines

or adding line numbers.

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
<?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 'The PHP version: ';
echo phpversion();
echo ' (generated inline HTML)';
</php>
<PHP>
echo '<table class="inline"><tr><td>The same, but inside a block level
element:</td>';
echo '<td>'.phpversion().'</td>';
```

```
echo '</tr></table>';
</PHP>
```

<php> echo 'The PHP version: '; echo phpversion(); echo ' (inline HTML)'; </php> <PHP> echo '<table class=«inline»><tr><td>The same, but inside a block level element:</td>'; echo '<td>'.phpversion().'</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 HTML is disabled all tags will be stripped
nosort	do not sort the items in the feed
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.

By default the feed will be sorted by date, newest items first. You can sort it by oldest first using the `reverse` parameter, or display the feed as is with `nosort`.

Example:

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

- AI-Assisted Coders Run Windows on iPhone Duo, Build a Floppy Disk-Based AI Chatbot for MS-DOS — EditorDavid (16.11.2026 12:34)
- Purism Announces 'Librem 5 v2' Linux Phone with RISC-V SoC — EditorDavid (16.11.2026 08:34)
- ULA Vulcan Rocket Failure Destroys 40 Amazon Leo Satellites — EditorDavid (16.11.2026 06:34)
- AI-Generated Books Confront Publishing Industry With Big Questions — EditorDavid (16.11.2026 03:45)
- Should Jensen Huang Return as Nvidia's CEO? Supporters and Critics Disagree — EditorDavid (16.11.2026 02:02)

Control Macros

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

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 rerender 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:

- Add New Page plugin *2026-11-16* — Damien Regad, Benjamin Santalucia, Sam Wilson and other contributors
  Adds a "new page form" to any wiki page.
- comment *2023-08-18* — Gina Häußge, Michael Klier
  Add comments/notes to your wiki source that won't be shown on the page.
- Dw2Pdf plugin *2026-10-29* — Andreas Gohr and Luigi Micco
  DokuWiki to PDF converter
- Gallery Plugin *2026-09-02* — Andreas Gohr
  Creates a gallery of images from a namespace or RSS/ATOM feed
- Graphviz Plugin *2025-08-27* — Andreas Gohr
  Create graph images from Graphviz dot language
- iframe plugin *2023-08-17* — Christopher Smith
  Allows external URLs to be loaded into an iframe in your DokuWiki page.
- Info Plugin *2020-06-04* — Andreas Gohr
  Displays information about various DokuWiki internals
- tablelayout plugin *2026-09-24* — Michael Große
  Set the width of table columns and set which lines to fix
- Video Sharing Site Plugin *2025-12-10* — Andreas Gohr
  Easily embed videos from various Video Sharing sites. Example: {{youtube>XXXXXX}}
- Wrap Plugin *2026-06-02* — 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.

Examples for the iframe Plugin

Simple:

```
{{url>http://www.example.com/somepage.html}}
```

Complete Syntax:

```
{{url>someurl width,height noscroll noborder alignment|alternate-text}}
```

That is an URL to embed, followed by several optional parameters. The following parameters are recognized:

- *width* and *height* are optional. When only one value is given it is assumed to be the height. Otherwise the first value is width and the second value is height. You can give the values in `px`, `em`, `pt` or `%`. When no unit is given `px` ~~is~~ is NOT assumed (you have to give the value). Defaults values are: width - 98%, height - 400px.
- `noscroll` will disable the scrollbars of the iframe
- `noborder` will disable the frame border, note that your CSS might still add borders or may disable them without using this setting
- Optionally an *alignment* can be given as `left` or `right` to have the iframe floating to the left or right. Should be used with a width.
- An *alternate text* is optional. If not specified an empty string will be used.

Examples for the Wrap Plugin

Basic syntax

An uppercase **<WRAP>** (or alternatively **<block>** or ** **) creates a **`div`** and should be used for «**big**» containers, **surrounding** paragraphs, lists, tables, etc.

```
<WRAP classes #id width :language>
"big" content
</WRAP>

or
<block classes #id width :language>
"big" content
</block>

or
<div classes #id width :language>
"big" content
</div>
```

A lowercase **<wrap>** (or alternatively **<inline>** or ****) creates a **span** and should be used for «**small**» containers, **inside** paragraphs, lists, tables, etc.

```
<wrap classes #id width :language>"small" content</wrap>

or
<inline classes #id width :language>"small" content</inline>

or
<span classes #id width :language>"small" content</span>
```

Please note, some things **won't work with lowercase spans**:

- **alignments** (including alignments generated by changing the text direction)
- **multi-columns**
- and **widths**

if the according wrap isn't floated as well.

A shorthand of uppercase **<WRAP />** and lowercase **<wrap />** are available if the container is empty or unnecessary.

```
<WRAP classes #id />
or
<block classes #id />
or
<div classes #id />
```

and

```
<wrap classes #id />
or
<inline classes #id />
or
<span classes #id />
```

Classes and Styles

Columns and Floats

You can have columns easily by adding the class `column` and a width, e.g.

```
<WRAP column 30%>...content...</WRAP>
```

The example below uses the following structure:

```
<WRAP group>

<WRAP third column>...content...</WRAP>
<WRAP third column>...content...</WRAP>
<WRAP third column>...content...</WRAP>

</WRAP>
```

Floating Options

Normally you would only need the class `column`, but for more sophisticated uses (not only for columns, but for any other classes, like boxes and notes as well) you can have several kinds of «floats»:

- **`column`** is the same as `left` in LTR languages and the same as `right` in RTL languages

- **`left`** will let you float your wrap on the left
- **`right`** will let the wrap float right
- **`center`** will position the wrap in the horizontal center of the page

Widths

You can set any valid widths (but only on divs): `%, px, em, rem, ex, ch, vw, vh, pt, pc, cm, mm, in`, but most of the time you'd only want either

type	e.g.	note
`%`	`30%`	makes sense in most cases
`px`	`420px`	makes sense if your container contains images with a certain width
`em`	`20em`	makes sense if you like your wrap container to grow and shrink with the font size

Width Keywords

With certain width keywords you can fit your columns automatically to fill the available horizontal space. Those columns will also react to the screen size, so will be responsive and wrap underneath each other on mobile devices.

There are four width keywords. These should not be combined with any other width, only `third` can be combined with `twothirds`.

- **`half`** fits two columns in a row
- **`third`** fits three columns in a row
- **`quarter`** fits four columns in a row
- **`twothirds`** together with `third` fits a 2/3 and a 1/3 column in a row

Attention: In order to work properly, wraps with width keywords need an **additional <WRAP group> around a set** of them. If you mix several types in one group, you might need a <WRAP `clear/`> after a row.

You can use the same options with spans (as each element that floats is automatically a block level element), but it probably doesn't make too much sense. Widths on spans normally do not work (by design), but can make sense, when it is floating.

Attention: What is the difference between widths and width keywords and when is it best to use which? **Widths** can cause problems and will never fully add up, therefore will break the layout under some circumstances. (See box model for a technical explanation.) So, getting widths right will need some fiddling around and testing in various browsers. **Width keywords** on the other hand fit automatically and work better on mobile devices. But the drawback is that they need an extra wrap around them and don't work properly in older browsers (IE8 and under).

All of those options will also work in the boxes and notes wraps (see below).

Old Emulated Headline (deprecated)

When the `emulatedHeadlines` config option is enabled, every `//**__text like this__**//` or `//**like that**//` will create an «emulated headline» within a wrap. This feature is deprecated

and will be removed at some point as standard headlines within wraps are supported for a while now.

If that config options is enabled and you need text that is bold and italic, simply use it the other way around: `**//No Headline//**`.

Multi-columns

Multi-columns work best in modern browsers (no IE9 and below) but should still be considered experimental as some browser behaviour is still inconsistent and buggy.

Just use **`colsmall`** for small width columns, **`colmedium`** for medium width columns and **`collarge`** for large width columns. Or you can use **`col2`** for 2 columns, **`col3`** for 3 columns, **`col4`** for 4 columns and **`col5`** for 5 columns. The former type of column is determined by its width, the latter by its amount. The width-based columns are ideal for different layouts and device widths.

Note: Multi-columns don't make sense for spans.

Don't use this for bigger columns containing more than just text. Use the columns and floats mentioned above instead.

Alignments

You can use these different text alignments:

- `leftalign`
- `rightalign`
- `centeralign`
- `justify`

Center aligned text ...

... and right aligned.

```
<WRAP centeralign>
Center aligned text ...
</WRAP>

<WRAP rightalign>
... and right aligned.
</WRAP>
```

You cannot add alignments to spans.

Boxes and Notes

round box 500px center

- box creates a box around the container and uses the colours from the template's `style.ini` as default colours (`__background_alt__` and `__text__`)
- any of the classes `info`, `tip`, `important`, `alert`, `help`, `download`, `todo` will add a special note container with a corresponding icon
- the classes `danger`, `warning`, `caution`, `notice`, `safety` use safety colours (and no icons)
- `round` can be added to anything with a background colour or a border and will only work in modern browsers (no IE8 and under)

Info

<WRAP info></WRAP>

Tip

<WRAP tip></WRAP>

Important

<WRAP important></WRAP>

Alert

<WRAP alert></WRAP>

Help

<WRAP round help></WRAP>

Download

<WRAP download></WRAP>

Todo

<WRAP todo></WRAP>

Safety Notes:

Best only use simple markup in safety notes.

Danger

<WRAP danger></WRAP>

Warning

<WRAP warning></WRAP>

Caution

<WRAP caution></WRAP>

Notice

<WRAP round
notice></WRAP>

Safety

<WRAP round
safety></WRAP>

You can use notes and boxes also inside text with spans like this: info, help, alert, important, tip, download, todo and round box and danger, warning, caution, notice, safety.

```
<wrap info>info</wrap>, <wrap help>help</wrap>, ...
```

Marks

You can mark text as highlighted, less significant and especially emphasised.

```
You can mark text as <wrap hi>highlighted</wrap>, <wrap lo>less
significant</wrap> and <wrap em>especially emphasised</wrap>.
```

This might look ugly in some templates and should be adjusted accordingly.

Tabs

You can create a row of tabs by simply wrapping a list of links in `<WRAP tabs></WRAP>`.

- Some page
- This page
- Another page

```
<WRAP tabs>
  * [[Some page]]
  * [[example|This page]]
  * [[Another page]]
</WRAP>
```

Please note, the styling of these tabs depend on the template you are using and not on the wrap plugin. If you only see a list of links and no tabs, please make sure to add «tabs» to the `noPrefix` config option and that your template supports at least the 2012-01-25 «Angua» DokuWiki release.

Miscellaneous

Clear float

After using any of the float classes, you might come across following text protruding into the space where only the floating containers should be. To prevent that, you should simply add this after your last column:

```
<WRAP clear />
```

Table width

You can set the width of a table via `tablewidth` as every table inside that wrap will always be 100% wide. This makes it possible to give tables any width by adding an additional width to the wrap (or none for 100%).

Table	is
80%	wide

```
<WRAP tablewidth 80%>
^ Table ^ is ^
| 80% | wide |
</WRAP>
```

Indent

This text will appear indented.

```
<wrap indent>This text will appear indented.</wrap>
```

Outdent

This text will appear «outdented».

```
<wrap outdent>This text will appear "outdented".</wrap>
```

Prewrap

```
Inside this code block the
words will wrap to a new
line although they are all
in one line.
```

```
<WRAP prewrap 250px>
<code>
Inside this code block the words will wrap to a new line although they are
all in one line.
</code>
</WRAP>
```

Spoiler

Here follows a spoiler: Darth Vader is Luke's father.

```
Here follows a spoiler: <wrap spoiler>Darth Vader is Luke's father.</wrap>
```

Just select the text in the spoiler box to be able to read its content.

Button links

A link that looks like a button: Syntax

```
A link that looks like a button: <wrap button>[[wiki:Syntax]]</wrap>
```

Hide

The following text is hidden: John, please revise that sentence.

```
The following text is hidden: <wrap hide>John, please revise that
sentence.</wrap>
```

Warning: The text will still appear in the source code, in non-modern browsers and is searchable. Do not hide any security risky secrets with it!

Pagebreak

The following will add a pagebreak:

```
The following will add a pagebreak: <WRAP pagebreak />
```

This has no effect on the browser screen. A pagebreak will force a new page in printouts.

Nopagebreak

The following will try to avoid a pagebreak:

much content, belonging together (like a long table)

```
The following will try to avoid a pagebreak: <WRAP nopagebreak>much content,
belonging together (like a long table)</WRAP>
```

This also has no effect on the browser screen. It will try to avoid a page break in printouts.

Noprint

This text appears on the screen, but not in print.

```
<wrap noprint>This text appears on the screen, but not in print.</wrap>
```

Onlyprint

This text does not appear on the screen, but only in print.

```
<wrap onlyprint>This text does not appear on the screen, but only in
print.</wrap>
```

Combining and Nesting

You can combine and nest all classes and types of boxes, e.g.

Outer box floats right

Inner nested box floats
left and is partly
emphasized and
highlighted with a
nested notice inside.

Text inside outer right box, but beneath inner left box.

Round tip box underneath, after a `clear`.

```
<WRAP box 350px right :en>
=== Outer box floats right ===

<WRAP 165px left>
Inner nested box floats left and is partly <wrap em hi>__em__phasized and
__hi__ghlighted with a nested <wrap notice>__notice__</wrap> inside</wrap>.
</WRAP>

Text inside outer right box, but beneath inner left box.

<WRAP clear />

<WRAP round tip>
Round tip box underneath, after a ''clear''.
</WRAP>

</WRAP>
```

Language and Text Direction

You can change the language and the reading direction of a wrap container by simply adding a colon followed by the language code, like this:

```
<WRAP :he>
זה עברית. ((<wrap :en>This means "This is Hebrew.", at least according to
[[http://translate.google.com/|Google Translate]].</wrap>))
</WRAP>
```

זה עברית. [3]

The text direction (`rtl`, right to left or `ltr`, left to right) will get inserted automatically and is solely dependent on the language. The list of currently supported languages is taken from: http://meta.wikimedia.org/wiki/Template:List_of_language_names_ordered_by_code (If you specify a language not listed there, it simply won't do anything.)

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
3)
This means «This is Hebrew.», at least according to Google Translate.

From:
https://wiki.direktiva.ai/ - **Connectome.ai Wiki**

Permanent link:
https://wiki.direktiva.ai/wiki/syntax?rev=1555079024

Last update: **19.05.2020 09:47**