



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 automatically: <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 automatically: `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.

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 height²⁾:



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:
{{https://secure.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 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 [emojicons](#) 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.

→ ← ↔ ⇒ ⇐ ⇔ » « – 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)	

Heading 1	Heading 2	Heading 3
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
xxxxxxxxxxxxx	xxxxxxxxxxxxx	xxxxxxxxxxxxx

This is how it looks in the source:

```

^          Table with alignment          ^^^
|          right|      center      |left      |
|left          |          right|      center      |
| xxxxxxxxxxxxx | xxxxxxxxxxxxx | xxxxxxxxxxxxx |

```

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>`.

```
/**
 * 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 higest hq9plus html html4strict html5 icon idl ini inno intercal io ispfpnl java5 java javascript jcl j jquery julia kixtart klonex klonexcpp 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 mxm1 mysql nagios netrexx newlisp nginx nimrod nsis oberon2 objc object 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 rpspec rsplus ruby rust sas sass scala scheme scilab scl sdlbasic smalltalk smarty spark sparql sql standardml stonescript swift systemverilog tcl tcl 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 xoj 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 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

You can embed raw HTML code into your documents by using the `<html>` 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>
```

This is some inline HTML

And this is some block HTML

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 rerender a page on every call

Examples of Wrapping

Basic syntax

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

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

or

<block classes width :language>
"big" content
</block>

or

<div classes 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 width :language>"small" content</wrap>

or

<inline classes width :language>"small" content</inline>

or

<span classes 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.

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>
```

```
<WRAP column 30%> Emulated Big Headline
```

You can emulate a big headline with italic, bold and underlined text, e.g.

```
/**__Emulated Big Headline__**//
```

Emulated Small Headline

A smaller headline uses no underlining, e.g.

```
/**Emulated Small Headline**//
```

If you need text that is bold and italic, simply use it the other way around:

```
**//No Headline//**
```

```
</WRAP>
```

```
<WRAP column 30%> Different 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

```
</WRAP>
```

```
<WRAP column 30%> Widths
```

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

type	e.g.	note
%	30%	makes sense in a liquid layout
px	420px	makes sense if your layout has a fixed pixel width or 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 or if your layout is em-based

A **table** inside a column or box will always be **100% wide**. This makes positioning and sizing tables possible.

</WRAP>

<wrap em>After using any of the float classes, you might come across something like this, where the following text protrudes into the space where only the floating containers should be ...</wrap>


<WRAP clear></WRAP>


... to prevent that, you should simply add

```
<WRAP clear></WRAP>
```

after your last column.

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: Widths can cause problems and will often look different and break in some browsers. If you're not a web developer, you might not understand any problems regarding the [box model](#). Just try to test your columns in all major browsers and make your widths smaller than you initially think they should be.

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

Multi-columns

<WRAP col3> For modern browsers (Firefox, Chrome and Safari) you can use multi-columns. Just use **col2** for 2 columns, **col3** for 3 columns, **col4** for 4 columns and **col5** for 5 columns.

 Note: Multi-columns don't make sense for spans. </WRAP>

Alignments

You can use these different text alignments:

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

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

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

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

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



You cannot add alignments to spans.

Boxes and Notes

`<WRAP round box 570px center> round box 570px 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 Internet Explorer)

`</WRAP>`

`<WRAP info 220px left> Info`

```
<WRAP info></WRAP>
```

`</WRAP>`

`<WRAP tip 220px left> Tip`

```
<WRAP tip></WRAP>
```

```
</WRAP>
```

```
<WRAP important 220px left> Important
```

```
<WRAP important></WRAP>
```

```
</WRAP>
```

```
<WRAP alert 220px left> Alert
```

```
<WRAP alert></WRAP>
```

```
</WRAP>
```

```
<WRAP round help 220px left> Help
```

```
<WRAP round help></WRAP>
```

```
</WRAP>
```

```
<WRAP download 220px left> Download
```

```
<WRAP download></WRAP>
```

```
</WRAP>
```

```
<WRAP todo 220px left> Todo
```

```
<WRAP todo></WRAP>
```

```
</WRAP>
```

```
<WRAP clear></WRAP>
```

Safety Notes:

```
<WRAP danger 27% left> Danger
```

```
<WRAP danger></WRAP>
```

```
</WRAP>
```

```
<WRAP warning 27% left> Warning
```

```
<WRAP warning></WRAP>
```

</WRAP>

<WRAP caution 27% left> **Caution**

```
<WRAP caution></WRAP>
```

</WRAP>

<WRAP round notice 27% left> **Notice**

```
<WRAP round notice></WRAP>
```

</WRAP>

<WRAP round safety 27% left> **Safety**

```
<WRAP round safety></WRAP>
```

</WRAP>

<WRAP clear></WRAP>

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

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

Marks

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

```
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.

Miscellaneous

Indent

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

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

Outdent

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

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

Prewrap

<WRAP prewrap 250px>

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

</WRAP>

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

```
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.

Hide

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

```
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: `<WRAP pagebreak></WRAP>`

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

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: `<WRAP nopagebreak>much content, belonging together (like a long table)</WRAP>`

```
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

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

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

Onlyprint

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

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

Typography

I advice against using the following typography classes. It's better to create semantic classes that reflect

their meaning instead.

- font family: sansserif, serif, monospace
- font size: bigger, muchbigger, smaller
- font colour: fgred, fggreen, fgblue, fgcyan, fgviolet, fgyellow, fggrey, fgwhite, fgblack
- background colour: bgred, bggreen, bgblue, bgcyan, bgviolet, bgyellow, bggrey, bgwhite, bgblack

Combining and Nesting

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

```
<WRAP box bggreen fgblack 350px right :en> Outer green box floats right
```

```
<WRAP 165px left> Inner nested box floats left and is partly <wrap em hi>emphasized and highlighted
with nested <wrap bigger>bigger text</wrap> inside</wrap>. </WRAP>
```

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

```
<WRAP clear></WRAP>
```

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

```
</WRAP>
```

```
<WRAP box bggreen fgblack 350px right :en>
/**__Outer green box floats right__**//

<WRAP 165px left>
Inner nested box floats left and is partly <wrap em hi>__em__phasized and
__hi__ghlighted with nested <wrap bigger>__bigger__ text</wrap> inside</wrap>.
</WRAP>
```

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

```
<WRAP clear></WRAP>
```

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

```
</WRAP>
```

Draft

Draft allows you to hide a page for everyone that is not signed in.

Usage

- To hide a page, simply add `~~DRAFT~~` anywhere in the text.
 - In case you are using German language, the code is `~~ENTWURF~~`.
 - In case you are using Norwegian language, the code is `~~KLADD~~`.
- The plugin also adds a button to the far right of the editing buttons that looks like a *forbidden* sign. Clicking it will add the correct code.

Effect

If you are an unregistered user, the page will not show up. If you are registered, you can see the page, as well as a red warning on the top of the page, which reminds you that the following page is a draft; hidden from the view of unregistered users.

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

From:

<https://wiki.pangolin.com/> - **Complete Help Docs**

Permanent link:

<https://wiki.pangolin.com/doku.php?id=wiki:syntax&rev=1564872820>

Last update: **2020/06/11 19:21**

