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
  <?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:
{{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 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
- 😐 :-|
- 😉 ;-)
- 😄 ^_^
- 🤦 m(
- ❓ :?:
- ❗ :!:
- 🤣 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 (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
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 roff rpmspec rsplus ruby rust sas sass scala scheme scilab scl sdlbasic smalltalk smarty spark sparql sql sshconfig 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 wolfram 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>
```

This is some inline HTML

And this is some block 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>
```

echo 'The PHP version: '; echo phpversion(); echo ' (inline HTML)';

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

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 }}
```

- CT Scans Projected to Result in 100,000 New Cancers in The US von BeauHD (2025/04/15 09:00)
- Indian IT Faces Its Kodak Moment von msmash (2025/04/15 07:00)
- Chinese Robotaxis Have Government Black Boxes, Approach US Quality von BeauHD (2025/04/15 05:30)
- Climate Crisis Has Tripled Length of Deadly Ocean Heatwaves, Study Finds von msmash (2025/04/15 03:50)
- Apple To Analyze User Data on Devices To Bolster AI Technology von msmash (2025/04/15 02:50)

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:

Folded Plugin

Full documentation: https://www.dokuwiki.org/plugin:folded

If you want to make additional information available that is hidden by default, you have two options with this plugin:

Inline:

```
This is example ++text | with some of it only shown when you unfold it++.
And after that
the text just continues to flow in the same paragraph.
```

This is example text with some of it only shown when you unfold it. And after that the text just continues to flow in the same paragraph.

Block:

```
This is example text.

++++ Title |

| This table | is only shown | when you unfold the block |

{{page>some other wiki page&inline}}

++++
```

	Inline	Block
Syntax	++title| formatted text ++	++++title| any content ++++
HTML	<span> tag	<div> tag
Can contain formatting	🙂	🙂
Can contain block elements[3]	—	🙂
Can be used within a paragraph, table, list, etc.	🙂	—

You can see in the image below.

Note: As of version 2005-09-02 the syntax has changed to allow linked titles to unfold and fold the section. The pipe char between title and text is mandatory.

Wrap Plugin

Full documentation: https://www.dokuwiki.org/plugin:wrap

This plugin gives you the ability to wrap wiki text inside containers (divs or spans) and give them

1. a certain class (with loads of useful preset classes)
2. a width
3. a language with its associated text direction

Basic Syntax:

```
<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>
```

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

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

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

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

Since version 2013-06-13 there is also a shorthand syntax (for wraps without content):

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

and

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

Please note, some things **won't work with spans**: **alignments** (including alignments generated by changing the text direction), **multi-columns** and **widths** if the according wrap isn't floated as well.

Examples

The plugin comes with an example page, which should explain a lot and looks like this in the default template, see https://demo.selfthinker.org/plugin:wrap

Classes

The following classes are currently available:

class name	description/notes
columns - similar to columns, side_note, styler, tip	
column	same as `left` in LTR languages and same as `right` in RTL languages
left	same as `column`, will let you float your container on the left
right	will let the container float right
center	will position the container in the horizontal center of the page
col2..**col5**	will show the text in multiple columns determined by their amount (2, 3, 4 or 5), only works in modern browsers (no IE9 and below)
colsmall, **colmedium**, **collarge**	will also show the text in multiple columns but determined by their width (small, medium or large), only works in modern browsers (no IE9 and below)
widths - ⚠ **experimental, might not work as expected, includes mobile support**	
half	fits two columns in a row, should be used in pairs
third	fits three or two columns in a row, should be used in triplets or together with `twothirds`
twothirds	fits two columns in a row when used together with `third`, one 1/3 wide and another 2/3 wide
quarter	fits four columns in a row, should be used in quads
alignments - similar to divalign, columns, styler - ⚠ **don't work with spans!**	
leftalign	aligns text on the left
rightalign	aligns text on the right
centeralign	centers the text
justify	justifies the text
boxes and notes - similar to box, note, tip	
box	creates a box around the container (uses colours from `style.ini`)
info (was `information` in first version)	creates a blue box with an info icon
important	creates an orange box with an important icon
alert (⚠ was `warning` in previous versions)	creates a red box with an alert icon
tip	creates a yellow box with a tip icon
help	creates a violet box with a help icon
todo	creates a cyan box with an todo icon

class name	description/notes
download	creates a green box with a download icon
round	adds rounded corners to any container with a background colour or a border (only works in modern browsers, i.e. no IE)
danger	creates a red danger safety note
warning	creates an orange warning safety note
caution	creates a yellow caution safety note
notice	creates a blue notice safety note
safety	creates a green safety note
marks – similar to **emphasis**, **important_paragraf**, **importanttext**	
hi	marks text as highlighted
lo	marks text as less significant
em	marks text as especially emphasised
miscellaneous	
clear	similar to clearfloat, should preferably be used with divs, i.e. uppercase <WRAP>s
tabs	if wrapped around a list of links, will show those as tabs
hide	hides the text per CSS (the text will still appear in the source code, in non-modern browsers and is searchable)
noprint	displays text on the screen, but not in print, similar to noprint
onlyprint	displays text only in print, but not on the screen
pagebreak	forces a new page in printouts (not visible on the screen), similar to pagebreak
nopagebreak	tries to avoid a pagebreak in printouts (not visible on the screen)
spoiler	shows white text on a white background, only to be revealed by highlighting it; similar to hide
button	when wrapped around a link, styles it like a button
tablewidth	sets widths of tables inside to whichever width the wrap gets, partly replaces tablewidth
indent	indents the text, could be used instead of tab
outdent	"outdents" the text, could partly be used instead of outdent
prewrap	wraps text inside pre-formatted code blocks, similar to wpre

Indexmenu Plugin

Full documentation: https://www.dokuwiki.org/plugin:indexmenu

This plugin allows you to insert a fully customizable index or a list of pages starting from a specified namespace. It should be useful in DokuWiki sites where pages are organized by namespaces. Main features are:

- Fully customizable with a lot of flexible options, but easy to use and configure for standard needs.
- Built-in support of Navigation features like highlighting the current location or dynamically displaying the tree of the current namespace.
- Easily themeable with prebuilt JavaScript themes.
- Assign Namespaces title and link (headpages) to namespaces.
- Sortable by date, title and custom metadata information.
- AJAX support to speed up sites with many pages.

- Customizable [context mouse menu](#) for usual namespace/page actions.
- TOC pages preview.
- [Replace](#) the DokuWiki page index.
- Hide namespaces/pages according to [ACLs](#) and plugin [settings](#).

Minimum syntax:

- **{{indexmenu>.}}**
- **{{indexmenu>:}}**
- **{{indexmenu>|}}**

That means this **{{indexmenu}}** and this **{{indexmenu>}}** do not work.

Basic syntax usage:

Main	Options
{{indexmenu>ns[#n] [ns1[#n] ns2[#n] ...]	*[js[#theme]] [tsort] ...* **}}**

Arguments inside [ ] parenthesis are optional. The # char is always required with related options.

⚠ All the syntax options can be easily accessed with the indexmenu picker in the [edit window toolbar](#).

Full syntax

Settings **before the "|"** separator:

Main	Action	Note
ns	**Main namespace** name. Index starts from here. Syntax complies with DokuWiki [namespaces](#) paths.	"**.**" refers to the namespace of the page containing the indexmenu syntax and **not** to the current DokuWiki namespace context [4] (see the **context** option for this feature). "**..**" or an empty value shows the root site namespace.
#n	**n** is a number that specifies how many namespace levels to display open under the **main namespace**.	If it's not defined then the whole tree, till the deeper node, will be open. If 0 or 1 it'll display only nodes under the main namespace. For example: "#2" will display "root:myns1:myns2" but will keep myns2 closed thus hiding "root:myns1:myns2:myns3". Optional.
ns1[#n] ... nsn[#n]	A list **of optional namespaces** inside the **main namespace**. Every namespace will be opened or closed at the specified **n** level. Syntax complies with DokuWiki [namespaces](#).	If **n** is not defined then all namespaces are open, if 0 they are closed. "**.**" refers to the namespace of the page containing the indexmenu syntax and **not** to the current DokuWiki namespace context [5] (see the **context** option for this feature). "**..**" or an empty value shows the root site namespace. Optional.

Optional settings **after the "|"** separator (separated by spaces):

Option	Action	Note
js **undo: nojs**	JavaScript render method: the index is an expandable tree menu. Without the `js` option DokuWikis index renderer is used	Without **n**, all nodes are open, with it, nodes are open till **n** level.
#theme **undo: #default**	Theme name for indexmenu icons	A theme is a set of icons inside *images* directory as described in Theme tutorial. Admins can download and share themes in admin panel. It works only in **js** e.g. **js#tango**
Next options are available with or without **js** option.		
navbar **undo: nonavbar**	The tree opens itself automatically at the current page namespace. Useful in a navigation sidebar.	Without **js** option, the indexmenu page is never cached (just like the default DokuWiki index page) and the DokuWiki loading could be slower depending on the amount of child nodes displayed.
context **undo: nocontext**	Relative **main namespace** and **optional namespaces** will refer to the current DokuWiki namespace context [6] instead of to the namespace of the page containing the indexmenu syntax. Useful in a navigation sidebar.	The indexmenu page is now never cached so the DokuWiki loading could be slower depending on the amount of child nodes displayed (In **js** mode, when a lot of nodes are usually displayed, the *max* option is recommended). It automatically enable the ***nocookie*** option.
tsort **undo: notsort**	Sort (only) pages by title.	Useful when useheading is on. By default namespaces are **not** sorted, you need the ***nsort*** option for this.
dsort **undo: nodsort**	Sort (only) pages by date creation (first the oldest).	By default namespaces are **not** sorted, you need the ***nsort*** option for this.
msort[#meta] **undo: nomsort**	Sort (only) pages by a custom metadata information. Without the #meta parameter, i.e. `msort`, it looks for the custom sorting number specified with the `{{indexmenu_n>N}}` syntax (see the below metadata tag syntax). With #meta parameter you can refer to the meta data structure (Array values are managed through the ":" separator, for example: `msort#date:modified`).	By default, pages without metadata tag are sorted by page name (the default DokuWiki way), but you can override this behaviour adding also the tsort or dsort option in the indexmenu syntax. By default namespaces are **not** sorted, you need the ***nsort*** option for this.
hsort **undo: nohsort**	Sort the headpages as defined by config setting startpage to the top	msort overrules hsort
rsort **undo: norsort**	Reverse the sorting of pages (change between ascending and descending).	By default namespaces are **not** sorted, you need the ***nsort*** option for this.

Option	Action	Note
nsort **undo: nonsort**	Also sort namespaces according to page sort options but grouped separately.	To use in *addition* to the above sort options. tsort, dsort, msort, hsort apply only for namespaces when using [headpages](). rsort is applicable always together with nsort.
nons **undo: ns**	Exclude namespaces nodes from index. It shows only the pages.	Without **js**, the closing **n** namespace option prevents to display nodes below the **n** namespace level.
nopg **undo: pg**	Exclude pages nodes from index. It shows only the namespaces.	All namespace nodes will link to the start pages (as defined by [startpage]() setting)
skipfile[+|=]/regexp/	Skip files matching the regexp.	`skipfile+/../` skips files defined with this regexp additional to [global skip]() config. `skipfile=/../` replace the global skip config with regexp from this syntax. See the global config explanation for some regexp [examples]().
skipns[+|=]/regexp/	Skip namespaces matching the regexp.	Just like skipfile, but [namespaces]().
Next options are *only* available with **js** option.		
max#n*[#m]*** undo: nomax**	If initially closed, the node at **n** level will retrieve all its child nodes through the AJAX mechanism when opened for the first time. Optionally, the nodes after the **n** level can be retrieved with AJAX every **m** sublevels instead of in one go.	It affects the server loading and speeds up the loading of pages in DokuWiki with an high amount of pages. It works only in **js**. Cookie are automatically disabled, just like with **nocookie**.
maxjs#*n* **undo: nomaxjs**	It sets how many js tree levels to render when page loads. Remaining nodes are rendered (slightly slower) only when they are open by users, by **optional namespaces** option, by cookies or by **navbar** option.	Default **n** is 1 so that it will speed up the page loading, above all with an high amount of pages. It affects only the user-client CPU speed, not the webserver load. It works only in **js**

Option	Action	Note
id#[random|ns|*number*] **undo: id#random**	🗩 Cookie Identifier for a js indexmenu where the previously opened/closed nodes by a user are stored.	Useful when a page is uncached and you like the tree state is stored in cookie. (See **nocookie** for disabling cookies.) Default the option *id#random* is active, also when the option is not specified in the syntax. You can apply self a number as unique identifier for your indexmenu (e.g. *id#20*) or let generate a number unique for requested namespace with *id#ns*. Read the Js does not remember its previous state section. **ATTENTION:** ID must be unique for every indexmenu in your DokuWiki site or you'll get strange js behaviors. Tree state storage with cookie only in *js*
nocookie **undo: cookie**	Disable 🗩 cookies. By default js indexmenu remember selected, open and closed nodes by user during navigation. With this option it doesn't remember them and the tree is blocked to its start status.	Tree state storage with cookie only in *js*
noscroll **undo: scroll**	Disable the JavaScript scrolling feature when it doesn't fit the container width. It could solve visualization problems.	It works only in *js*
notoc **undo: toc**	Disable the TOC-preview feature.	ToC preview only available in *js*
nomenu **undo: menu**	Disable the contextmenu feature.	Context menu only available in *js*

Examples

A sample of an indexmenu JS index that could be used inside a navigation sidebar. Its initial status is blocked by the nocookie option, so, when the page is reloaded, it doesn't remember the open and closed nodes by the user.:

```
{{indexmenu>..#1|js navbar nocookie}}
```

JS navigation index with "thread" theme where nodes after the third level are retrieved with Ajax every 2 sublevels. Pages are sorted by title and custom sort number:

```
{{indexmenu>..#1|js#thread navbar max#3#2 tsort msort}}
```

Standard DokuWiki index showing only pages inside wiki:plugins and lower namespaces (max two levels):

```
{{indexmenu>:wiki:plugins#2|nons}}
```

Js tree showing pages and namespaces both sorted by reverse title. For example,if "archive" contains stuff ("news","oldnews",etc) that you need to quickly organize by time, you could create numbered headpages for every namespace (i.e renaming "oldnews" in "news 2006", "news" in "news 2010" and so on) and sort them from new to older:

```
{{indexmenu>:archive#1|js tsort nsort rsort}}
```

Standard index showing the tree of the current context [7] opened at the second level .

```
{{indexmenu>playground#2|context}}
```

Show all current namespace pages .

```
{{indexmenu>.:#1|context}}
```

JS tree showing all (and only) the namespaces of the "private" namespace sorted by date creation. "private" is relative and refers to the private namespace under the page containing the indexmenu syntax.

```
{{indexmenu>private|js nopg dsort}}
```

Metadata tag syntax

By default nodes on the same tree level are sorted by name (or by title/date if you use the tsort/dsort syntax), but you can also specify a custom sort number for every page inserting a metadata tag in the pages with this syntax:

```
{{indexmenu_n>N}}
```

Where N is a number. Then you need to use the "msort" option in your indexmenu tree syntax. If you have the show_sort option enabled in the Configuration Manager, a notice is displayed to admins (only) on every page with this metadata tag (the text defaults to "Indexmenu sort number: N").

Examples:

You can change the order of this tree containing a mix of standard and useheading pages:

```
-Root
  |_don
  |_Mirror sessions        (headline title of the ":mirror" page)
  |_pachuco
  |_At the radar station   (headline title of the ":radar" page)
  |_van
  |_vliet
```

in this way:

```
{{indexmenu>..#1|msort}}
```

```
-Root
  |_vliet                  {{indexmenu_n>1}}
  |_van                    {{indexmenu_n>2}}
  |_don                    {{indexmenu_n>3}}
  |_Mirror sessions        (headline title of the ":mirror" page)
  |_pachuco
  |_At the radar station   (headline title of the ":radar" page)
```

Pages without sort number, like the last three pages, are sorted by page name as default, but you can force a different sort:

```
{{indexmenu>..#1|tsort msort}}
```

```
-Root
  |_vliet                  {{indexmenu_n>1}}
  |_van                    {{indexmenu_n>2}}
  |_don                    {{indexmenu_n>3}}
  |_At the radar station   (headline title of the ":radar" page)
  |_Mirror sessions        (headline title of the ":mirror" page)
  |_pachuco
```

Video Share Plugin

Full documentation: https://www.dokuwiki.org/plugin:vshare

This plugin allows you to embed video players from various video sharing sites. New services can be added by just editing a config file. This is **not** for displaying local video files.

Usage/Syntax

The basic syntax looks like this: {{*videosite>videoid?parameter1&parameter2*|title}}

- Where *videosite* is one of the identifiers listed in Supported Services chapter
- and *videoid* is the identifier of the video at the respective site
- The *parameters* are optional. You start these with a ? and separate more of them by a &. Look in Parameters chapter
- The title is optional as well. Look in Examples chapter
- The video can be aligned by adding spaces on the left or right inside the curly brackets (like in the image syntax). Look in Examples chapter

A toolbar button pops up a prompt where you can simply paste the full URL to the page of the video you want to embed. The plugin will then try to figure out the video ID by itself. For some services you may need to paste a special URL. See the table below.

Parameters

When embedding a video you should add a size parameter.

You can either give it in the form:

- *width*x*height* like 500x300
- or use the keywords `small`, `medium` or `large`
- you can also use the keywords `full` or `half` to have the video adjust to the available screen width (either 100% or 50% width)

All additional parameters will be passed on as-is to the video service. Refer to their documentation for what is available. There are also hints in the table below.

Examples

Display a YouTube Video:

```
{{youtube>L-WM8YxwqEU}}
```

Show a larger player:

```
{{youtube>L-WM8YxwqEU?large}}
```

Right-align the player:

```
{{ youtube>L-WM8YxwqEU}}
```

Show a small, centered player:

```
{{ youtube>L-WM8YxwqEU?small }}
```

Show a small, centered player with a title (look for right space!):

```
{{ youtube>L-WM8YxwqEU?small |Some funny video}}
```

Some other additional parameters are supported (depending on video service) as well:

```
{{youtube>L-WM8YxwqEU?small&start=30&end=45|A random segment of 15 seconds}}
```

Supported Services

Copy paste the video url in the toolbar pop-up prompt to generate the syntax

Identifier	Website	Comments	Supported parameters
youtube	YouTube		start, end, rel, autoplay
vimeo	Vimeo		autoplay
slideshare	Slideshare	paste the Wordpress shortcode	startSlide
dailymotion	Daily Motion		start
twitchtv	Twitch.tv		chapter_id, initial_time
archiveorg	Archive.org		

Identifier	Website	Comments	Supported parameters
soundcloud	SoundCloud		
niconico	NicoNico		
break	Break		
bitchute	BitChute		
coub	Coub		
odysee	Odysee LBRY	paste the embed or download URL	
youku	Youku	unclear if working	
bilibili	bili bili		
msoffice		unclear if working	
msstream		unclear if working	

[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)]

like tables, lists, new paragraphs, included files, etc.

[4)] , [5)] , [6)] , [7)]

the namespace of the page displayed by a user who is navigating your site

From:
https://wiki.loxberry.de/ - **LoxBerry Wiki - BEYOND THE LIMITS**

Permanent link:
https://wiki.loxberry.de/en/wiki/syntax?rev=1663268410

Last update: **2022/09/15 21:00**