

# Formatting Syntax

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

## Basic Text Formatting

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

DokuWiki supports **`**bold**`**, *`//italic//`*, `__underlined__` and `'monospaced'` texts.

Of course you can **`**__//''combine''//__**`** all these.

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

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

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

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

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

This is some text with some linebreaks

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

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

You should use forced newlines only if really needed.

## Links

DokuWiki supports multiple ways of creating links.

## External



## 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](#).

## Image Links

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

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



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

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

## Footnotes

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

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

## Sectioning

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







-  **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.

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

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

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

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

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

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





<file>

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

&lt;/file&gt;

## Syntax Highlighting

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

```
/**
 * The HelloWorldApp class implements an application that
 * simply displays "Hello World!" to the standard output.
 */
class HelloWorldApp {
    public static void main(String[] args) {
        http://www.google.com/search?hl=en&q=allinurl%3Adocs.oracle.com+javase+docs+api+stringString s = "Hello World!";
        http://www.google.com/search?hl=en&q=allinurl%3Adocs.oracle.com+javase+docs+api+systemSystem.out.println(s);
    }
}
```

The following language strings are currently recognized: *abap, actionscript-french, actionscript, actionscript3, ada, apache, applescript, asm, asp, autoit, avisynth, bash, basic4gl, bf, bibtex, blitzbasic, bnf, boo, c, c\_mac, caddcl, cadlisp, cfdg, cfm, cil, cmake, cobol, cpp, cpp-qt, csharp, css, d, dcs, delphi, diff, div, dos, dot, eiffel, email, erlang, fo, fortran, freebasic, genero, glsl, gml, gnuplot, groovy, gettext, haskell, hq9plus, html, idl, ini, inno, intercal, io, java5, java, javascript, kixtart, klonex, klonexcpp, latex, lisp, locobasic, lolcode, lotusformulas, lotusscript, lscript, lsl2, lua, m68k, make, matlab, mirc, modula3, mpasm, mxml, mysql, nsis, oberon2, objc, ocaml-brief, ocaml, oobas, oracle8, oracle11, pascal, perl, per, php-brief, php, pic16, pixelbender, plsql, povray, powershell, progress, prolog, properties, providex, python, qbasic, rails, rebol, reg, robots, ruby, sas, scala, scheme, scilab, sdlbasic, smalltalk, smarty, sql, tcl, teraterm, text, thinbasic, tsql, typoscript, vbnet, vb, verilog, vhdl, vim, visualfoxpro, visualprolog, whitespace, winbatch, whois, xml, xorg\_conf, xpp, z80*

## Downloadable Code Blocks

When you use the `code` or `code_block` 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>
```

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

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

```
<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>
```

---

Printed on 2025/07/03 18:20