Wrap Plugin

One plugin to rule them all

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

Original plugin page can be found at https://www.dokuwiki.org/plugin:wrap.

Syntax

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 ** **) 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 ****) 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 below).

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 (2, 3, 4 or 5), only works in modern browsers (Firefox, Chrome and Safari)
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 columns in a row, should be used in triplets
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

class name	description/notes
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
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
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

All tables inside a column or box will always be 100% wide. This makes positioning and sizing tables possible and partly replaces tablewidth.

Known restrictions

- WRAPs won't export in ODT format.
- Round corners only work in modern browsers (no IE8 and below).
- Multiple columns only work in modern browsers (no IE9 and below).
- Width classes are experimental and only work in modern browsers (no IE8 and below).
- Normal DokuWiki Headlines used to not work and a work-around was added. Now that headlines

do work, the work-around is not needed anymore but kept for backwards-compatibility. The following syntax would produce two different kinds of emulated headlines inside any of the columns or boxes/notes:
  - //**__Big Underlined Headline__**// (They will look a bit different in safety notes.)
  - //**Small Headline**//

You might need to adjust a few of the classes to your template's needs, especially `hi`, `lo` and `em`. If you have a dark or otherwise heavily coloured theme, please use the `darkTpl` config option.

The classes are easily adjustable and extensible. Any wishes are welcome.

Widths

You can set any valid widths on any uppercase <WRAP> container: `%, px, em, ex, pt, pc, cm, mm, in`. Just set the width before or after or with the classes, e.g.

```
<WRAP someclass 50% anotherclass>...
```

All except percentages will be reduced to have the maximum width available on smaller screens.

You can also use the width keywords `half`, `third` and `quarter`. To work correctly they need another wrap around them. E.g.

```
<WRAP group>
  <WRAP half column>...</WRAP>
  <WRAP half column>...</WRAP>
</WRAP>
```

will result in two columns next to each other, which will wrap underneath each other on smaller screens and mobile devices.

Languages and Text Directions

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

```
<wrap :en>This text is explicitly marked as English.</wrap>
```

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 like to mark a text with a different text direction than the default one, you should use divs, i.e. uppercase <WRAP>s. Otherwise the text alignment won't change as well.

This makes it a better replacement of ltr (and lang).

Demo

You can see a demo of the plugin on demo.selfthinker.org.

“Examples” (demo) in Russian (for v2011-05-15). Source.

Configuration options

Option	Description	Default value
noPrefix	Which (comma separated) class names should be excluded from being prefixed with “wrap_”	tabs, group
restrictedClasses	restrict usage of plugin to these (comma separated) classes	*(empty)*
restrictionType	restriction type, specifies if classes above shall be included or excluded	0
syntaxDiv	Which syntax should be used in the toolbar picker for block wraps?	WRAP (other choices: div, block)
syntaxSpan	Which syntax should be used in the toolbar picker for inline wraps?	wrap (other choices: span, inline)
darkTpl	Optimise colours for dark templates?	0

ODT Support

Since version 2015-06-13 the Wrap plugin supports exporting most of its functionality/styling to ODT when using at least version 2015-06-29 of the ODT Plugin. By default, Wrap syntax will be exported to ODT using 'print' CSS styles. This means the exported Wrap elements will look the same when printing a wiki page. If you want to have the ODT exported Wrap elements look like displayed in the browser (i.e. with 'screen' CSS styles), then use the following ODT plugin configuration settings:

- add wrap to the 'usestyles' config setting
- set the 'media_sel' setting to 'screen'

If you prefer a user defined CSS style for the Wrap ODT export, then simply place a file 'odt.css' into the Wrap plugin folder with your own CSS code (and set config setting 'media_sel' to 'print').

Here is what is currently **not** supported:

- columns: left/right/center/column is partly supported; they are positioned correctly, but content is not floating around them
- widths are not supported except % and half/third/quarter
- boxes and notes: hardly any formatting inside them is supported, therefore emulated headings also don't work
- tabs will just show a list of links
- noprint
- nopagebreak
- onlyprint only works on boxes
- languages are set correctly but do not seem to affect text alignment

- shorthand syntax
- Not supported because not relevant in ODT: clear, prewrap

Toolbar picker



The wrap picker in the editing toolbar adds the most common wrap syntaxes.

- “columns” creates a set of half columns
- “simple centered box” creates a standard box (60% wide, centered)
- “info, tip, important, alert, help, download, todo box” creates specifically themed boxes (also 60% wide, centered)
- “clear floats” creates a `<WRAP clear></WRAP>`
- “especially emphasised, highlighted, less significant” creates the respective marks

Extend with custom styles

If you like to add your own classes and styles to the plugin, you can simply add the styles for your class preceded by “`wrap_`” to your user styles. Please note, any classes need to be **lower case**.

E.g. if you need a `<WRAP myclass>`, you edit (or create if it doesn't exist) your `conf/userstyle.css` and add your `.wrap_myclass{}` with its style definitions to it. (If necessary, edit `conf/userprint.css`[1] for the print view, `conf/userrtl.css`[2] for RTL languages and `conf/userall.css`[3] for all styles as well.)

User permissions for every file used must be similar to original Dokuwiki files.

Since version 2010-03-14 you have the possibility to exclude certain class names from being prefixed with “wrap_”. Just add a comma separated list of class names into the config option “noPrefix” in the configuration manager.

Examples

in style.css

```
.dokuwiki div.wrap_note{ /* added */
    background-color: #eee;
    color: #000;
    padding: .5em .5em .5em .5em;
    margin-bottom: 1em;
    overflow: hidden;
}
```

call in DW-page:

```
<WRAP note>...</WRAP>
```

Add former typography classes

The old typography classes were removed in version 2011-05-15. If you need something similar, use the block plugin instead. Or you can use your own and copy them to your own user styles (see above).

How to use the helper

From version 2011-05-15 on the plugin includes a helper plugin which you can use to add classes, width and lang/dir to any other plugin.

Example how to get just one kind of attribute

```
// get lang from wrap helper plugin
$lang = '';
if(!plugin_isdisabled('wrap')) {
    $wrap = plugin_load('helper', 'wrap');
    $attr = $wrap->getAttributes($data);
    if($attr['dir']) $lang = ' lang="'.$attr['lang'].'"
xml:lang="'.$attr['lang'].'" dir="'.$attr['dir'].'"';
}

// add lang to your plugin's output
$renderer->doc .= '<span '.$lang.' class="foo">';
```

getAttributes() expects the string with “classes #id width :language”. It returns an array with

- $attr['class']: CSS class(es)
- $attr['id']: CSS ID
- $attr['width']: width
- $attr['lang'] and $attr['dir']: language and text direction

Example how to get all attributes

```
// get attributes from wrap helper plugin
$attr = '';
if(!plugin_isdisabled('wrap')) {
    $wrap = plugin_load('helper', 'wrap');
    $attr = $wrap->buildAttributes($data, 'additionalClass');
}

// add those attributes to your plugin's output
$renderer->doc .= '<div '.$attr.'">';
```

buildAttributes() expects the same string as above (“classes #id width :language”) and an optional string for additional classes, in case your plugin has CSS classes of its own which it needs to reuse. It

returns a string with all the attributes prepared for HTML.

Localization

You can help me with translations and update the language files. There are two files to translate:

- lang/en/lang.php is for the explaining titles on the picker images.
- lang/en/settings.php is for the configuration options.

[1] `conf/printstyle.css` in Anteater

[2] `conf/rtlstyle.css` in Anteater

[3] `conf/allstyle.css` in Anteater

From:
https://www.jutis.net/wiki/ - **DaWiki**

Permanent link:
https://www.jutis.net/wiki/doku.php?id=wiki:wrap_plugin

Last update: **2019/07/26 14:31**