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 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>
```

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//**
```

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

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.

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 …

… 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

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.

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 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)



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:

Danger

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

Warning

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

Caution

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

Notice

Safety

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

```
<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.

Miscellaneous

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:

```
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:

```
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></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:

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

```
<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>
```

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>__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>
```

Outer green box floats right

Inner nested box floats left and is partly **emphasized and highlighted with nested bigger text inside**.

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



Round tip box underneath, after a `clear`.

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.[1]

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 means „This is Hebrew.“, at least according to Google Translate.

From:
https://services.tubit.tu-berlin.de/wiki/ - **TUB-Wikifarm**

Permanent link:
https://services.tubit.tu-berlin.de/wiki/doku.php?id=wiki:plugins:wrap

Last update: **2021/12/10 12:05**