Using HTML in Wiki Text

Trac supports inserting HTML into any wiki context, accomplished using the `#!html` WikiProcessor.

However a constraint is that this HTML has to be well-formed. In particular you can't insert a start tag in an `#!html` block, resume normal wiki text and insert the corresponding end tag in a second `#!html` block.

Fortunately, for creating styled s, s or even complex tables containing arbitrary Wiki text, there's a powerful alternative: use of dedicated `#!div`, `#!span` and `#!table`, `#!tr`, `#!td` and `#!th` blocks.

Those Wiki processors are built-in, and does not require installing any additional packages.

How to use `#!html`

To inform the wiki engine that a block of text should be treated as HTML, use the *html* processor.

Wiki Markup	Display

`{{{`
`#!html`
`<h1 style="text-align: right; color: blue">HTML Test</h1>`
`}}}`
	HTML Test

Note that Trac sanitizes your HTML code before displaying it. That means that if you try to use potentially dangerous constructs such as Javascript event handlers, those will be removed from the output.

Since 0.11, the filtering is done by Genshi, and as such, the produced output will be a well-formed fragment of HTML. As noted above in the introduction, this mean that you can no longer use two HTML blocks, one for opening a , the second for closing it, in order to wrap arbitrary wiki text. The new way to wrap any wiki content inside a is to use the `#!div` Wiki processor.

How to use `#!div` and `#!span`

Wiki Markup	Display

`{{{`
`#!div class="important"`
`**important** is a predefined class.`
`}}}`
	important is a predefined class.

`{{{`
`#!div style="border: 1pt dotted; margin: 1em"`
`**wikipage** is another predefined class that will`
`be used when no class is specified.`
`}}}`
	wikipage is another predefined class that will be used when no class is specified.

`{{{`
`#!div class="compact" style="border: 1pt dotted; margin: 1em"`
`**compact** is another predefined class reducing`
`the padding within the ` ` to a minimum.`
`}}}`
	compact is another predefined class reducing the padding within the `<div>` to a minimum.

`{{{`
`#!div class="wikipage compact" style="border: 1pt dotted"`
`Classes can be combined (here **wikipage** and **compact**)`
`which results in this case in reduced //vertical//`
`padding but there's still some horizontal space for coping`
`with headings.`
	Classes can be combined (here **wikipage** and **compact**) which results in this case in reduced *vertical* padding but there's still some horizontal space for coping with headings.

	Explicitly specifying no classes is *not* the same as specifying no class attribute, as this will remove the

```
}}}
```

wikipage default class.

```
{{{
#!div class="" style="border: 1pt dotted; margin: 1em"
Explicitly specifying no classes is //not// the same
as specifying no class attribute, as this will remove
the //wikipage// default class.
}}}
```

Note that the contents of a `#!div` block are contained in one or more paragraphs, which have a non-zero top and bottom margin. This leads to the top and bottom padding in the example above. To remove the top and bottom margin of the contents, add the `compact` class to the `#!div`. Another predefined class besides `wikipage` and `compact` is `important`, which can be used to make a paragraph stand out. Extra CSS classes can be defined via the `site/style.css` file for example, see TracInterfaceCustomization#SiteAppearance.

For spans, you should rather use the Macro call syntax:

Wiki Markup

```
Hello
[[span(''WORLD'' (click [#anchor here]), style=color: green; font-size: 120%, id=anchor)]]!
```

Display

Hello *WORLD* (click here)!

How to use `#!td` and other table related processors

`#!td` or `#!th` processors are actually the main ones, for creating table data and header cells, respectively. The other processors `#!table` and `#!tr` are not required for introducing a table structure, as `#!td` and `#!th` will do this automatically. The `|-` row separator can be used to start a new row when needed, but some may prefer to use a `#!tr` block for that, as this introduces a more formal grouping and offers the possibility to use an extra level of indentation. The main purpose of the `#!table` and `#!tr` is to give the possibility to specify HTML attributes, like *style* or *valign* to these elements.

Wiki Markup

```
Simple 2x2 table with rich content:
{{{#!th align=left
 - Left
 - Header
}}}
{{{#!th align=left
 - Right
 - Header
}}}
|----------------------------------
{{{#!td style="background: #ffd"
 - Left
 - Content
}}}
{{{#!td style="vertical-align: top"
!RightContent
}}}
|----------------------------------
|| ... and this can be mixed||\
||with pipe-based cells ||
{{{#!td colspan=2
Pick the style the more appropriate
to your content

See WikiFormatting#Tables for details
on the pipe-based table syntax.
}}}
```

Display

Simple 2x2 table with rich content:

• **Left** • **Header**	• **Right** • **Header**
• Left • Content	RightContent
... and this can be mixed	with pipe-based cells
Pick the style the more appropriate to your content See WikiFormatting#Tables for details on the pipe-based table syntax.	

If one needs to add some attributes to the table itself...

Left header	Right header
1.1	1.2

2.1	2.2

```
If one needs to add some
attributes to the table itself...

{{{
#!table style="border:none;text-align:center;margin:auto"
  {{{#!tr ====================================
    {{{#!th style="border: none"
    Left header
    }}}
    {{{#!th style="border: none"
    Right header
    }}}
  }}}
  {{{#!tr ==== style="border: 1px dotted grey"
    {{{#!td style="border: none"
    1.1
    }}}
    {{{#!td style="border: none"
    1.2
    }}}
  }}}
  {{{#!tr ====================================
    {{{#!td style="border: none"
    2.1
    }}}
    {{{#!td
    2.2
    }}}
  }}}
}}}
```

Note that by default tables are assigned the "wiki" CSS class, which gives a distinctive look to the header cells and a default border to the table and cells (as can be seen for the tables on this page). By removing this class (`#!table class=""`), one regains complete control on the table presentation. In particular, neither the table, the rows nor the cells will have a border, so this is a more effective way to get such an effect than having to specify a `style="border: no"` parameter everywhere.

Wiki Markup	Display

`{{{#!table class=""`
`||  0||  1||  2||`
`|| 10|| 20|| 30||`
`|| 11|| 22|| 33||`
`||||||=  numbers  =||`
`}}}`
	0 1 2
	10 20 30
	11 22 33
	numbers

Other classes can be specified as alternatives (remember that you can define your own in site/style.css).

Wiki Markup	Display

`{{{#!table class="listing"`
`||  0||  1||  2||`
`|| 10|| 20|| 30||`
`|| 11|| 22|| 33||`
`||||||=  numbers  =||`
`}}}`
	0 1 2
	10 20 30
	11 22 33
	numbers

HTML comments

HTML comments are stripped from the output of the `html` processor. To add an HTML comment to a wiki page, use the `htmlcomment` processor (available since 0.12). For example, the following code block:

Wiki Markup

```
{{{
#!htmlcomment
This block is translated to an HTML comment.
It can contain <tags> and &entities; that will not be escaped in the output.
}}}
```

Display

```
<!--
This block is translated to an HTML comment.
It can contain <tags> and &entities; that will not be escaped in the output.
-->
```

Please note that the character sequence "`--`" is not allowed in HTML comments, and will generate a rendering error.

More Information

- ?http://www.w3.org/ -- World Wide Web Consortium
- ?http://www.w3.org/MarkUp/ -- HTML Markup Home Page

See also: WikiProcessors, WikiFormatting, WikiRestructuredText