

Wikiprint Book

Title: Wiki テキスト内で HTML を使用する

Subject: SilverFrost - WikiHtml

Version: 3

Date: 12/20/25 10:38:59

SilverFrost 目次

Wiki テキスト内で HTML を使用する	3
#!html の使い方	3
#!div and #!span ブロックの使い方	3
#!td とその他の表関連のプロセッサの使い方	4
HTML コメント	6
より詳しい情報	6

Wiki テキスト内で HTML を使用する

`#!html` [Wiki プロセッサ](#) の機能によって、Trac では Wiki コンテキスト内の HTML 利用が可能です。

しかし、HTML は well-formed でなければいけないという制約があります。とりわけ、1つ目の `#!html` ブロックに開始タグを挿入し、その後普通の wiki テキストを続けて、2つ目の `#!html` ブロックに1つ目のブロックに挿入した開始タグに対応する終了タグを挿入することはできません。

幸いなことに、`<div>` や `` やもっと複雑なスタイルを使用して任意の Wiki テキストを含んだ表を作成する場合は、強力な代替手段があります：表専用の `#!div`, `#!span`, `#!table`, `#!tr`, `#!td`, `#!th` ブロックを使用することです。

これらの Wiki プロセッサはビルトインされているので、追加で他のパッケージをインストールする必要はありません。

`#!html` の使い方

テキストブロックが HTML として取り扱われるよう、Wiki エンジンに html プロセッサを使用するという情報を与えます。

Wiki マークアップ	表示
<pre>{{{#!html <h1 style="text-align: right; color: blue">HTML ■■■</h1> }}}</pre>	HTML テスト

Trac が HTML コードを表示する前にサニタイズすることに注意して下さい。つまり Javascript のイベントハンドラのような潜在的に危険なコードを使おうとしても、出力から削除されるということです。

0.11 以降では、フィルタリングは Genshi が行いますので、出力は well-formed な HTML の断片でなければなりません。上記イントロダクションで記述したとおり、`<div>` を開く HTML ブロックと閉じる HTML ブロックの 2 つを使って Wiki テキストを囲むことができなくなります。`<div>` タグ内部に含まれている Wiki テキストをラップする新しい方法として、`#!div` Wiki プロセッサを使用する方法があります。

`#!div` and `#!span` ブロックの使い方

Wiki マークアップ	表示

```
{{{  
#!div class="important"  
**important** ██████████  
}}}
```

```
{{{  
#!div style="border: 1pt dotted; margin: 1em"  
**wikipage** ██████████  
██████████  
}}}{{}}
```

```
{{{  
#!div class="compact" style="border: 1pt dotted; margin: 1em"  
**compact** █████ `<div>` █████  
██████████  
}}}{{}}
```

```
{{{  
#!div class="wikipage compact" style="border: 1pt dotted"  
██████████ ( █████ **wikipage** █ **compact**)  
// █ // █  
██████████  
}}}{{}}
```

```
{{{  
#!div class="" style="border: 1pt dotted; margin: 1em"  
██████████  
██████████ // █ // █  
██████████ //wikipage// ██████████  
}}}{{}}
```

`important` は 定義済みのクラスです。

`wikipage` は クラスが指定されていない場合に
使用される別の定義済みクラスです。

`compact` は 最小限に `<div>` 内のパディングを
軽減するための別の定義済みクラスです。

クラスは組み合わせることができます (ここでは `wikipage` と
`compact`) この場合の結果は、垂直方向の パディングが
軽減されますが、見出しのため水平方向のスペースが
残っています。

明確にクラスを指定しないことは、
属性にクラスを指定しないことと同じではなく、
デフォルトクラスの `wikipage` を削除するようなものです。

Note: `#!div` ブロックの内容に 1
つもしくは複数の段落が含まれる場合、上下にマージンが追加されます。これは、上記の例にある通り、上下のパディングをもたらします。コンテンツの上下のマージン
プロックの `class` 属性に `compact` を追加して下さい。`wikipage` と `compact` の他に予め定義された値は、`important` です。これを指定すると段落を目立たせることができます。例えば、その他の CSS のクラスは、`site/style.css` ファイル経由で定義することができます。 [サイトの外観](#) を参照して下さい。

`span` については、むしろマクロ呼び出しのシンタクスを使用るべきです:

Wiki マークアップ	
Hello	
<code>[[span(''WORLD'') (click [#anchor here]), style=color: green; font-size: 120%, id=anchor)]]!</code>	
表示	
Hello WORLD (click here)!	

`#!td` とその他の表関連のプロセッサの使い方

実際に、それぞれのテーブル行やヘッダのセルを作成するためには、`#!td` や `#!th` がメインのプロセッサになります。その他、`#!table` や `#!tr` プロセッサは `#!td` や `#!th` プロセッサがこれらのプロセッサの役割を自動的に行なってしまうため、テーブル構造を紹介するためには必要ではありません。行セパレータ `|-` を新しい行を始める際に必要に応じて使うことができます。しかし場合によっては、`#!tr` ブロックを使用する方が好ましいかもしれません。`#!tr` ブロックの方がより正式なグルーピングを提供することができ、より多くのインデントレベルを提供できる可能性があります。`#!table` や `#!tr` プロセッサを使用する主な目的は、`style` や `valign` などで HTML の属性を指定できることです。

Wiki マークアップ	表示

```
██████████ 2x2 ██████:
{{#!th align=left
- █
- ███
}}}
{{#!th align=left
- █
- ███
}}}
|-----
{{#!td style="background: #ffd"
- █
- ███
}}}
{{#!td style="vertical-align: top"
████
}}}
|-----
|| ... ██████████ || \
|| ██████████ || |
{{#!td colspan=2
████████████████
████████
████████████████
[WikiFormatting#Tables WikiFormatting] ██████████
}}}

████████████████
████████...  

{{{
#!table style="border:none;text-align:center;margin:auto"
{{#!tr =====
{{#!th style="border: none"
█ ███
}}
{{#!th style="border: none"
█ ███
}}
}}
{{#!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
}}
}}
}}
```

リッチコンテンツを含んだ 2x2 の簡単な表:

• 左	• 右
• ヘッダ	• ヘッダ
• 左	右内容

... パイプでつなげたセルもくっつけることができます

あなたのコンテンツにより適切なスタイルを 選択して下さい。

パイプで表の列をつなげるシンタックスについては、 [WikiFormatting](#) を参照して下さい。

もし、表そのものに属性を追加する 必要がある場合...

左 ヘッダ	右 ヘッダ
1.1	1.2
2.1	2.2

Note: デフォルトの表設定では、 "wiki" の CSS クラスが適用されます。そして、この CSS クラスはヘッダのセルに対して典型的な見た目と表とセルのデフォルトの罫線設定を提供します（このページの表にあるとおりです）このクラスを削除することによって (#!table class="")

表の表示方法を自由に設定することができるようになります。特に、表、行、セルのいずれにも罫線を設定しない場合、いたるところで style="border: no" を指定するよりこの方法を使用したほうが、同じ効果を得るのにより効果的な方法となります。

Wiki マークアップ	表示
```{{#!table class=""   0   1   2     10   20   30     11   22   33       = ■■ =  }}```	0 1 2 10 20 30 11 22 33 数字
```{{#!table class="listing"   0   1   2     10   20   30     11   22   33       = ■■ =  }}```	0 1 2 10 20 30 11 22 33 数字

他のクラスを代替手段として指定することもできます ([site/style.css](#) であなたの自身のスタイルシートを定義できることを忘れないで下さい)

Wiki マークアップ	表示
```{{#!table class="listing"   0   1   2     10   20   30     11   22   33       = ■■ =  }}```	0 1 2 10 20 30 11 22 33 数字
```{{#!table class="listing"   0   1   2     10   20   30     11   22   33       = ■■ =  }}```	0 1 2 10 20 30 11 22 33 数字

HTML コメント

HTML コメントは `html` プロセッサの出力から除外されます。 HTML コメントを `wiki` ページに追加するために、 `htmlcomment` プロセッサを使用して下さい (0.12 以降)。例えば、下記のようなコードブロックになります:

Wiki マークアップ
```{{{#!htmlcomment ----- HTML ----- <tags> ■ &entities; ----- }}}```
表示
```<!-- ----- HTML ----- <tags> ■ &entities; ----- -->

Note: "--" という文字の並びは、 HTML コメントでは使用できません。レンダリング時にエラーが発生します。

より詳しい情報

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

See also: [WikiProcessors](#), [WikiFormatting](#), [WikiRestructuredText](#)