

<pre> {{{#!div style="background: #ffd; border: 3px ridge" ■■■ ■■■■■■ "■■■" ■■■■■■■■: {{{ #!python def hello(): return "world" }}} }}}</pre>	<div style="border: 2px solid black; background-color: #ffffcc; padding: 5px; margin-bottom: 5px;"> <p>これは 埋め込まれた "コード" ブロックの例です:</p> </div> <div style="border: 1px solid black; padding: 5px;"> <pre> def hello(): return "world"</pre> </div>
--	--

例 3: Wiki ページでチケットのキーワード検索を行なう

<pre> {{{ #!html <form action="/query" method="get"><div> <input type="text" name="keywords" value="~" size="30"/> <input type="submit" value="■■■■■■■■■■"/> <!-- To control what fields show up use hidden fields <input type="hidden" name="col" value="id"/> <input type="hidden" name="col" value="summary"/> <input type="hidden" name="col" value="status"/> <input type="hidden" name="col" value="milestone"/> <input type="hidden" name="col" value="version"/> <input type="hidden" name="col" value="owner"/> <input type="hidden" name="col" value="priority"/> <input type="hidden" name="col" value="component"/> --> </div></form> }}}</pre>	
---	--

プロセッサ一覧

以下のプロセッサが Trac の配布に含まれています:

#!default	フォーマット済みのテキストブロック内でテキストを逐語的に表示する。これはプロセッサ名を指定しない のと同じです (さらに #! を指定しないのと同じです)。
#!comment	このセクション内のテキストを処理しません (つまり、このセクションの内容はプレインテキスト内にのみ存在します - レンダリングしたページには表示されません)。
HTML 関連	
#!html	Wiki ページにカスタムした HTML を挿入する。
#!htmlcomment	Wiki ページに HTML のコメントを挿入する (0.12 以降)。
	<small>#!html ブロックは 自己閉結型 でなければなりません。つまり、1つ目のブロックである HTML 要素を開始し、後に対応する2つ目のブロックでその要素を閉じるということはできません。これに類似した効果を得るためには、下記に述べるプロセッサを使用して下さい。</small>
#!div	<div> 要素内の任意の Wiki の内容をラップする (0.11 以降)。
#!span	 要素内の任意の Wiki の内容をラップする (0.11 以降)。
#!td	<td> 要素内の任意の Wiki の内容をラップする (0.12 以降)。
#!th	<th> 要素内の任意の Wiki の内容をラップする (0.12 以降)。
#!tr	<small>#!td や #!th ブロックをラッピングしたり、グループ化した列を見やすくするために属性を指定する際、オプションとして使用することができます (0.12 以降)。</small>
	これらのプロセッサについての詳細と使用例については、 WikiHtml を参照して下さい。
その他のプロセッサ	

<pre>#!rst</pre>	<p>Trac における Restructured テキストのサポート。 WikiRestructuredText 参照。</p>
<pre>#!textile</pre>	<p>Textile がインストールされている場合のサポート。 a Textile リファレンス 参照。</p>
<p>コードのハイライトのサポート</p>	
<pre>#!c #!c++ (C++) #!python #!perl #!ruby #!php #!asp #!java #!js (Javascript) #!sql #!xml (XML or HTML) #!sh (Bourne/Bash shell) etc.</pre>	<p>Trac は、さまざまなプログラミング言語のソースコードに対応するインラインのシンタックスハイライトを提供するプロセッサを同梱しています。Trac では、シンタックスのカラーリングについて外部のソフトウェア (例えば Pygments) に依存しています。サポートしている言語の情報と他の言語のサポートを有効にする方法については TracSyntaxColoring を参照してください。</p>
<p>MIME タイププロセッサ</p>	
<p>プロセッサに MIME タイプを使用することによって、ソースコードをブラウザするとき、サポートしている言語に対してシンタックスハイライトを適用することが可能です。</p>	
<p>例:</p> <pre>{{{ #!text/html <h1>■■■■■</h1> }}}</pre>	<p>この結果はシンタックスハイライトされた HTML コードになります:</p> <pre><h1>■■■■■</h1></pre> <p>これは、すべての 対応している MIME タイプ に対して有効です。</p>
<pre>{{{ #!diff --- Version 55 +++ Version 56 @@ -115,8 +115,9 @@ name='TracHelloWorld', version='1.0', packages=find_packages(exclude=['*.tests*']), - entry_points = "" - [trac.plugins] - helloworld = myplugins.helloworld - "", + entry_points = { + 'trac.plugins': [+ 'helloworld = myplugins.helloworld', +], + }, +), }}}</pre>	<p><code>#!diff</code> は細部までレンダリングします:</p> <pre>--- Version 55 +++ Version 56 @@ -115,8 +115,9 @@ name='TracHelloWorld', version='1.0', packages=find_packages(exclude=['*.tests*']), - entry_points = "" - [trac.plugins] - helloworld = myplugins.helloworld - "", + entry_points = { + 'trac.plugins': [+ 'helloworld = myplugins.helloworld', +], + }, +)</pre>

ユーザによるプロセッサマクロの開発やコントリビュート:

- [ProcessorBazaar](#)
- [MacroBazaar](#)
- [Trac Hacks](#) コミュニティサイト

プロセッサは Wiki マacroを開発するのと変わりはありません。 実際、開発方法は同じであり、シンタックスの使用方法のみが異なります。 詳細情報については、 [カスタムマクロを開発する](#) を参照して下さい。

See also: [WikiMacros](#), [WikiHtml](#), [WikiRestructuredText](#), [TracSyntaxColoring](#), [WikiFormatting](#), [TracGuide](#)