Wiki 文法

Wiki マークアップは Trac のコアとなる機能です。他のパーツと緊密に統合することによって、 Trac をフレキシブルかつ強力にしています。

Trac は小さく強力な Wiki レンダリングエンジンを内蔵しています。この Wiki エンジンは、他のポピュラーな Wiki からのコマンドの強化サブセットを実装しています。 (特に MoinMoin)

このページは、 WikiFormatting でフォーマットできる文法の文例集です。

フォント書式

Trac の Wiki は以下のフォント書式に対応しています:

```
 * '''■■■■''', '''!''' ■■■■■■■■■■■■■■'''', '''! '''
* ''■■■■■''
* '''''■■■■■■■■■■'''''
* __■■■■■■■■__
* {{{■■■■■■■}}} or `■■■■■■■`
* ~~■■■■■■~~
* ^■■■■■■^
* ,,■■■■■■,,
```

表示内容:

- ボールド, ''' この箇所もボールドになります, !
- イタリック
- ボールドイタリック
- アンダーライン
- ■■■■■■■ or ■■■■■■■
- ~~取り消し線~~
- 上付き文字
- 下付き文字

Notes:

- {{{...}}}と `...`
  コマンドは表示を等角フォントにするだけではなく、内容を整形済みテキストとして扱います。つまりテキストに対して、どのような Wiki 処理も行いません。
- ! は wiki パーサに wiki フォーマットとして解釈しないように伝えます。なので、 !
  の後ろにスペースを置くことに注意して下さい。例えば、太字で行が終わるようなときです。

見出し

1 文字から 5 文字の「イコール '='」を行頭に置き、 1 文字のスペースに続けてタイトルを置くと、
見出しになります。見出しの行末は1文字のスペースに続けて、行頭と同じ数の イコール (=) を 置きます。 見出しに続いて id 属性を明示的に付与することが出来ます。付与しない場合、暗黙の読み取り可能な id 属性が生成されます。 (訳注: 明示的に付与する id 属性では日本語は使用できないようです。)

例:

```
= ■■■ (■1■■■) =
== ■■■ (■2■■■) ==
=== ■■■ (■3■■■) ===
=== ■■■■ id === #using-explicit-id-in-heading
```

表示内容:

見出し (第1レベル)

見出し（第2レベル）

見出し（第3レベル）

明示的な id

改行

2 つのテキストブロックの間に 1 行以上の空行がある場合、新しい段落として改行を行います。

強制的に改行だけを行いたい場合、このように書いてください:

```
1■■[[BR]]2■■
```

以下のように表示されます:

1行目
2行目

リスト

番号付きリストと番号無しリストの両方があります。

例:

```
 * Item 1
  * Item 1.1
     * Item 1.1.1
     * Item 1.1.2
     * Item 1.1.3
  * Item 1.2
* Item 2

1. Item 1
  a. Item 1.a
  a. Item 1.b
     i. Item 1.b.i
     i. Item 1.b.ii
1. Item 2
■■■■■■■■■■■■■■■■■■■■■■■■■■:
3. Item 3
```

表示内容:

- Item 1
  - Item 1.1
    - Item 1.1.1
    - Item 1.1.2
    - Item 1.1.3
  - Item 1.2
- Item 2

1. Item 1
   1. Item 1.a
   2. Item 1.b
      1. Item 1.b.i
      2. Item 1.b.ii
2. Item 2

番号付きリストでは、番号を指定することもできます:

1. Item 3

リストマーカーの前にスペースを一つ以上置かなければなりません。スペースがない場合、通常のパラグラフとして扱われます。

用語定義リスト

Wikiは用語定義リストも作成できます。

例:

```
 llama::
  ■■■■■■■■■■■■■
ppython::
  ■■■■■■■■■■■■
  (typo■■■■■■■■?)
```

表示内容:

llama
: 毛の生えた哺乳類の一種。

ppython
: 毛がない爬虫類の一種。 (typoを見つけたかい?)

定義する用語の前にスペースが一つ必要なことを忘れないで下さい。

整形済みテキスト

整形済みテキストを含むブロックは、ソースコードの一部や、ノート、例示に適しています。引用ブロックとしたいテキストの前後を、 3 つのブレース (curly braces) で括ってください。ブレースは独立した行に置かなければいけません。 例:

```
 {{{
 def HelloWorld():
     print "Hello World"
}}}
```

表示内容:

```
 def HelloWorld():
    print "Hello World"
```

引用ブロック

段落を引用ブロックとしてマークするには、段落を 2 個のスペースでインデントします。

例:

```
  This text is a quote from someone else.
```

表示内容:

  This text is a quote from someone else.

引用

進行中の議題のスレッド、チケットのコメントエリアなどで引用を表すために、メールライクな引用マーク (">", ">>" など) が使用できます。

例:

```
>> Someone's original text
> Someone else's reply text
My reply text
```

表示内容:

> > Someone's original text
>
> Someone else's reply text

My reply text

Note: いくつかの WikiFormatting の要素、
表やあらかじめ整形されたテキストは引用エリアでは整形が崩れてしまいます。きれいな引用を作るには再整形が必要になるでしょう。

表

簡単なテーブルはこのように記述します:

```
||Cell 1||Cell 2||Cell 3||
||Cell 4||Cell 5||Cell 6||
```

表示内容:

Cell 1	Cell 2	Cell 3
Cell 4	Cell 5	Cell 6

より複雑なテーブルは reStructuredText を使用して書くことができます。

リンク

WikiPageNames や URL は自動的にハイパーリンクされます。 WikiPageLinks を無効化したい場合、 `!WikiPageLink`
のように、エクスクラメーションマーク (!) を語頭に置きます。

例:

```
 TitleIndex, http://www.edgewall.com/, !NotAlink
```

表示内容:

TitleIndex, http://www.edgewall.com/, NotAlink

リンク先そのものでなくリンク先の内容説明やタイトルを表示させたい場合、角括弧 (square bracket) で括り、リンクに続いて 1
文字の空白を開けて記述します。もし説明のタイトルが省略された場合、そして明確なプレフィックスが記述されなかったとき、外部へのリンクにはなりません。こ
WikiPageNames の仕様に準拠できない Wiki ページを使いたい場合に役に立ちます。

例:

```
 * [http://www.edgewall.com/ Edgewall Software]
* [wiki:TitleIndex Title Index]
* [wiki:ISO9000]
```

表示内容:

- Edgewall Software
- Title Index
- ISO9000?

Trac リンク

Wiki ページでは Trac の他のコンポーネントに直接リンクできます。チケット、レポート、チェンジセット、マイルストーン、ソースファイル、他の Wiki ページへは以下のようにリンクを記述します:

```
 * ■■■■: #1 ■■■■ ticket:1
* ■■■■: {1} ■■■■ report:1
* ■■■■■■■■: r1, [1] ■■■■ changeset:1
* ...
```

表示内容:

- チケット: #1 もしくは ticket:1
- レポート: {1} もしくは report:1
- チェンジセット: r1, [1] もしくは changeset:1
- ...

Trac は他にも多くの種類のリンクを使用することができます。より詳しい情報は TracLinks を参照してください。

リンクの無効化

エクスクラメーションマーク (!) を語頭に付与することで、 TracLinks が出力するハイパーリンクを無効化できます。

例:

```
 !NoHyperLink
!#42 ■■■■■■■■■■■■■
```

表示内容:

NoHyperLink #42 はリンクではありません。

画像

URL が .png, .gif or .jpg で終わっている場合、自動的にイメージへのリンクとはならずに <img> タグに変換されます。

現在では [[Image]]
マクロを使用しなければなりません。画像を表示する最も簡単な方法は、現在のページへの添付ファイルとして画像をアップロードし、[[Image(picture.gif)]] のようにファイル名を指定してマクロを呼び出す方法です。

現在のページに加えて、他のリソースを参照することも可能です。

- [[Image(wiki:WikiFormatting:picture.gif)]] (referring to attachment on another page)
- [[Image(ticket:1:picture.gif)]] (file attached to a ticket)
- [[Image(htdocs:picture.gif)]] (referring to a file inside project htdocs)
- [[Image(source:/trunk/trac/htdocs/trac_logo_mini.png)]] (a file in repository)

表示例: trac POWERED

[[Image()]] マクロに関するより詳しい情報は WikiMacros に記載されています。

マクロ

マクロは、動的コンテンツを Wiki ページに挿入するための カスタム関数 です。

例:

```
 [[RecentChanges(Trac,3)]]
```

表示内容:

2012/10/20

- TracIni (diff)
- TracRepositoryAdmin (diff)
- TracSearch (diff)

使用方法とインストールされているマクロの一覧は WikiMacros に記述されています。

プロセッサ

Trac は WikiProcessors の機能によって、 Wiki 以外のマークアップにも対応しています。例えば、プロセッサとしてページの描画に reStructuredText や HTML を使用することができます。

例1:

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

表示内容:

HTMLテスト

例:

```
{{{
#!python
class Test:

   def __init__(self):
       print "Hello World"
if __name__ == '__main__':
  Test()
}}}
```

表示内容:

```
class Test:
   def __init__(self):
       print "Hello World"
if __name__ == '__main__':
  Test()
```

Perl:

```
my ($test) = 0;
if ($test > 0) {
   print "hello";
}
```

より詳細な情報は WikiProcessors に記載されています。

コメント

コメントをプレーンテキストに追加することができます。コメントはレンダリングされず、プレーンテキスト形式でのみ表示されます。

```
{{{
#!comment
```

```
Your comment here
}}}
```

その他

4 文字以上のハイフン（-）は水平ライン（<hr>）に置き換えられます。

例:

```
 ----
```

表示内容:

See also: TracLinks, TracGuide, WikiHtml, WikiMacros, WikiProcessors, TracSyntaxColoring.