



パーマリンクをマスターしよう リライトルール・オールライト Lite版

Prime Strategy


- ④ chapter 0 自己紹介
- ④ chapter 1 パーマリンクの基本のキ
- ④ chapter 2 リライトルールをのぞいてみよう
- ④ chapter 3 変換の仕組みを理解しよう
- ④ chapter 4 リライト状態の確認方法
- ④ chapter 5 カスタムパーマリンクの追加方法
- ④ chapter 6 カスタムパーマリンクの実践

Prime Strategy

自己紹介

Prime Strategy

大曲 仁 / jim912

 WordPressのインテグレーションサービスを提供するプライム・ストラテジーで主にシステム開発を担当。

 2010 WordCamp 横浜 実行委員長

WordBench 東京モデレーター

WordPress 日本語フォーラム世話役

Prime Strategy

WordPress Plugins

WordPress Plugin Directory で、プラグインを公開



PS Auto Sitemap



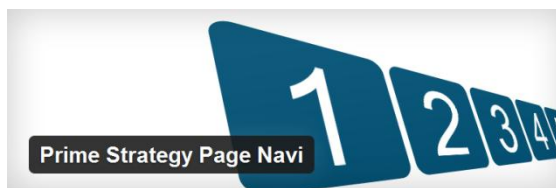
PS Disable Auto Formatting



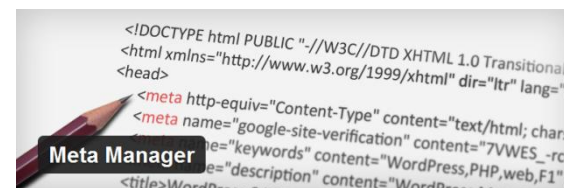
PS Taxonomy Expander



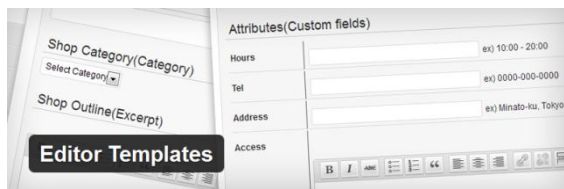
Prime Strategy Bread Crumb



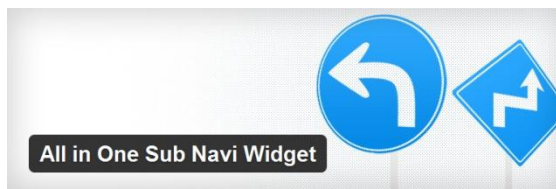
Prime Strategy Page Navi



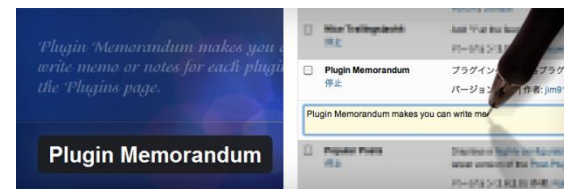
Meta Manager



Editor Templates



All in One Sub Navi Widget

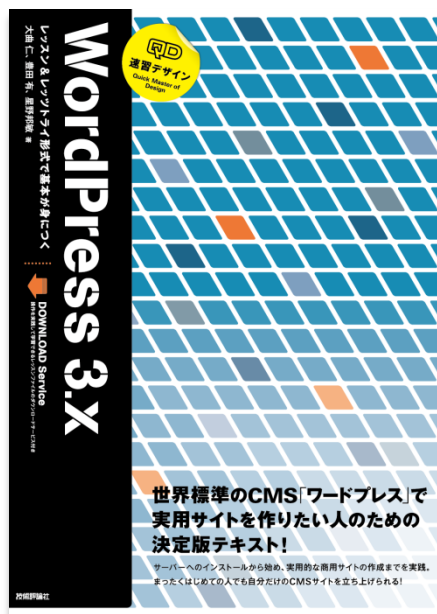


Plugin Memorandum



Books

WordPress の学習書籍を、2冊執筆（共著）



速習デザイン
WordPress 3.x



本格ビジネスサイトを作りながら学ぶ
WordPressの教科書



Simple Colors (<http://www.warna.info/>)



@jim0912



大曲 仁 (hitoshi.omagari)

Prime Strategy

パーマリンクの基本のキ

Prime Strategy

デフォルトパーマリンク

<http://www.example.com/?cat=7&paged=2>

<http://www.example.com/?m=201210>

<http://www.example.com/?p=48&page=3>

http://www.example.com/?page_id=51

? に続く、複数のパラメーターで構成

Prime Strategy

使用できるパラメーターは、46種類

m, p, posts, w, cat, withcomments, withoutcomments, s, search, exact, sentence, calendar, page, paged, more, tb, pb, author, order, orderby, year, monthnum, day, hour, minute, second, name, category_name, tag, feed, author_name, static, pagename, page_id, error, comments_popup, attachment, attachment_id, subpost, subpost_id, preview, robots, taxonomy, term, cpage, post_type

※カスタム分類を追加すると、その分使用できるパラメーターが増えます。

Prime Strategy

WordPressは、これらパラメーターによって、表示する内容の条件を決定します。

では、デフォルト以外のパーマリンク設定にしている場合は？

Prime Strategy

デフォルト以外の場合は、内部的にパラメーターに変換して、表示する条件を決定しています。

/category/wp/

⇒ category_name=wp

/sample-page/

⇒ pagename=sample-page

/2012/10/

⇒ year=2012&monthnum=10

Prime Strategy

アクセスされたURLをパラメーターに変換する規則を記録してあるのが、リライトルールです。

Prime Strategy

- ① WordPressは、全てパラメーターで表示する条件を決定している
- ① パーマリンク使用時は、パラメーターに変換している
- ① パラメーターへの変換は、リライトルールに基づいて行われる

Prime Strategy

リライトルールをのぞいてみよう

Prime Strategy

リライトルールをのぞいてみよう

パーマリンクをデフォルト以外に設定する。
header.phpの最初に、下記のコードを追加する。

```
<pre>  
<?php var_dump( get_option( 'rewrite_rules' ) ); ?>  
</pre>
```

Prime Strategy

リライトルールをのぞいてみよう

こんな表示になるはず。これがリライトルール

```
array(68) {
  ["category/(.+?)/feed/(feed|rdf|rss|rss2|atom)/?$"]=>
  string(52) "index.php?category_name=$matches[1]&feed=$matches[2]"
  ["category/(.+?)/(feed|rdf|rss|rss2|atom)/?$"]=>
  string(52) "index.php?category_name=$matches[1]&feed=$matches[2]"
  ["category/(.+?)/page/?([0-9]{1,})/?$"]=>
  string(53) "index.php?category_name=$matches[1]&paged=$matches[2]"
  ["category/(.+?)/?$"]=>
  string(35) "index.php?category_name=$matches[1]"
  ["tag/([^/]+)/feed/(feed|rdf|rss|rss2|atom)/?$"]=>
  string(42) "index.php?tag=$matches[1]&feed=$matches[2]"
  ["tag/([^/]+)/feed/(feed|rdf|rss|rss2|atom)/?$"]=>
  string(42) "index.php?tag=$matches[1]&feed=$matches[2]"
  ["tag/([^/]+)/page/?([0-9]{1,})/?$"]=>
  string(43) "index.php?tag=$matches[1]&paged=$matches[2]"
}
```

リライトルールをのぞいてみよう

68個の配列

キーは、正規表現

```
array(68) {  
  ["category/(.+?)/feed/(feed|rdf|rss|rss2|atom)/?$"]=>  
  string(52) "index.php?category=  
  ["category/(.+?)/(feed|rdf|  
  string(52) "index.php?category_name=$matches[1]&feed=$matches[2]"  
  ["category/(.+?)/page/?([0-9]{1,})/?$"]=>  
  string(53) "index.php?category_name=$matches[1]&paged=$matches[2]"  
  ["category/(.+?)/?$"]=>  
  string(35) "index.php?category_name=$matches[1]"  
  ["tag/([^\s]+)/feed/(feed|rdf|rss|rss2|atom)/?$"]=>  
  string(42) "index.php?tag=$matches[1]&feed=$matches[2]"  
  ["tag/([^\s]+)/feed/(feed|rdf|rss|rss2|atom)/?$"]=>  
  string(42) "index.php?tag=$matches[1]&feed=$matches[2]"  
  ["tag/([^\s]+)/page/?([0-9]{1,})/?$"]=>  
  string(43) "index.php?tag=$matches[1]&paged=$matches[2]"
```

値は、変換後のパラメーター形式

リライトルールをのぞいてみよう

- ① リライトルールは連想配列
- ② 配列のキーがURLパターンの正規表現、
値が変換ルール

Prime Strategy

変換の仕組みを理解しよう

Prime Strategy

リライトルールをループ処理し、アクセスされたURLと配列のキーの正規表現が合致するものを探す

ループ処理

```
array(68) {  
  ["category/(.+?)/feed/(feed|rdf|rss|rss2|atom)/?$"]=>  
  string(52) "index.php?category_name=$matches[1]&feed=$matches[2]"  
  ["category/(.+?)/(feed|rdf|rss|rss2|atom)/?$"]=>  
  string(52) "index.php?category_name=$matches[1]&feed=$matches[2]"  
  ["category/(.+?)/page/?([0-9]{1,})/?$"]=>  
  string(53) "index.php?category_name=$matches[1]&paged=$matches[2]"  
  ["category/(.+?)/?$"]=>  
  string(35) "index.php?category_name=$matches[1]"  
  ["tag/([^\s]+)/feed/(feed|rdf|rss|rss2|atom)/?$"]=>  
  string(42) "index.php?tag=$matches[1]&feed=$matches[2]"  
  ["tag/([^\s]+)/ (feed|rdf|rss|rss2|atom) /?$"]=>  
  string(42) "index.php?tag=$matches[1]&feed=$matches[2]"  
  ["tag/([^\s]+)/page/?([0-9]{1,})/?$"]=>  
  string(43) "index.php?tag=$matches[1]&paged=$matches[2]"
```

正規表現に合致したら、ループを終了。変換
ルールとして採用する

```
array(68) {  
  ["category/(.+?)/feed/(feed|rdf|rss|rss2|atom)/?$"]=>  
  string(52) "index.php?category_name=$matches[1]&feed=$matches[2]"  
  ["category/(.+?)/(feed|rdf|rss|rss2|atom)/?$"]=>  
  string(52) "index.php?category_name=$matches[1]&feed=$matches[2]"  
  ["category/(.+?)/page/?([0-9]{1,})/?$"]=>  
  string(53) "index.php?category_name=$matches[1]&paged=$matches[2]"  
  ["category/(.+?)/?$"]=>  
  string(35) "index.php?category_name=$matches[1]"  
  ["tag/([*?/]+)/feed/(feed|rdf|rss|rss2|atom)/?$"]=>  
  string(42) "index.php?tag=$matches[1]&feed=$matches[2]"  
  ["tag  
rin  
tag  
strin
```

ループを終了し
変換ルールとして採用

つまり、先にあるものほど
優先度が高い。

アクセスされたURLのうち、正規表現の丸括弧の部分と合致した部分が、変換ルール
の \$matches に
充てられる

```
array(68) {  
  ["category/(.+?)/feed/(feed|rdf|rss|rss2|atom)/?$"]=>  
  string(52) "index.php?category_name=$matches[1]&feed=$matches[2]"  
  ["category/(.+?)/(feed|rdf|rss|rss2|atom)/?$"]=>  
  string(52) "index.php?category_name=$matches[1]&feed=$matches[2]"  
  ["category/(.+?)/page/?([0-9]{1,})/?$"]=>  
  string(53) "index.php?category_name=$matches[1]&paged=$matches[2]"  
  ["category/(.+?)/?$"]=>  
  string(35) "index.php?category_name=$matches[1]"  
  ["tag/([^\s]+)/feed/(feed|rdf|rss|rss2|atom)/?$"]=>  
  string(42) "index.php?tag=$matches[1]&feed=$matches[2]"  
  ["tag/([^\s]+)/ (feed|rdf|rss|rss2|atom) /?$"]=>  
  string(42) "index.php?tag=$matches[1]&feed=$matches[2]"  
  ["tag/([^\s]+)/page/?([0-9]{1,})/?$"]=>  
  string(43) "index.php?tag=$matches[1]&paged=$matches[2]"
```

/category/wp/ にアクセスがあった場合

```
["category/(.+?)/?$"]=>
```

```
string(35) "index.php?category_name=$matches[1] "
```

に合致。

(.+?) にあたる wp が \$matches[1]となり、
index.php?category_name=wp へのアクセス
と同じとして処理される。

Prime Strategy

変換の仕組みを理解しよう

- ① リライトルールは、複数の連想配列
- ② 配列のキーが正規表現、値が変換ルール
- ③ 先に存在するルールほど優先度が高い

Prime Strategy

リライト状態の確認方法

Prime Strategy

URLがリライトルールのどれに合致し、どのようにパラメーターに変換されているか簡単に確認する方法があります。

Prime Strategy

Debug Bar

The screenshot displays the WordPress Debug Bar interface. At the top, it shows system information: 'koop.local #1', 'PHP 5.3.6', 'MySQL 5.5.9', and 'Memory Usage 3,708,912 bytes'. Below this, a sidebar on the left contains a menu with 'Queries', 'WP Query', 'Request', and 'Debug Bar' (which is highlighted). The main content area shows 'TOTAL QUERIES: 12' and 'TOTAL QUERY TIME: 14.3 ms'. A SQL query is displayed: 'SELECT option_name, option_value FROM wpdev_options WHERE autoload = wp_not_installed, is_blog_installed, wp_load_alloptions'. At the bottom, there is a description: 'Adds a debug menu to the admin bar that shows query, cache, and other helpful debugging information.' and a 'Download Version 0.8' button. A 'Favorited' status is also shown.

<http://wordpress.org/extend/plugins/debug-bar/>

Debug Barをインストールすると、ツールバーの右側に「Debug」が追加されます。



Prime Strategy

リライト状態の確認方法

「Debug」をクリックして、パネルを展開。
「Request」タブで、
リライトの状況を確認することができます。

アクセスされたURL

合致したリライトルール

変換されたパラメーター

WordPress3.4 3 2 + 新規

FERNANDO #1	PHP 5.3.8	MySQL 5.5.16	Memory Usage 17,438,896 bytes	Please Enable WP_DEBUG
-------------	-----------	--------------	-------------------------------	------------------------

Request

Object Cache

Request:
2012/10

Query String:
year=2012&monthnum=10

Matched Rewrite Rule:
$([0-9]{4})/([0-9]{1,2})/?$

Matched Rewrite Query:
year=2012&monthnum=10

リライト状態の確認方法

- ④ パーマリンクがおかしいと思ったら、リライトの状況を確認
- ④ Debug Barプラグインを使うと、一目瞭然

Prime Strategy

カスタムパーマリンクの追加方法

Prime Strategy

リライトルールの配列に独自のルールを追加すれば、デフォルトでは存在しない条件での表示を行うことが可能。

例)

- ・ カテゴリーの月別アーカイブ
- ・ 作成者のカテゴリー絞り込み
- ・ カスタム投稿タイプの年別アーカイブ

Prime Strategy

`add_rewrite_rule` という関数を使うと、独自のリライトルールを追加することができます。

```
add_rewrite_rule( $rule, $rewrite, $position );
```

`$rule` : URLパターンの正規表現

`$rewrite` : 変換するパラメーター形式

`$position` : リライトルールへの追加位置 (top or bottom)

Rewrite API/add rewrite rule (英語)

http://codex.wordpress.org/Rewrite_API/add_rewrite_rule

position パラメーターについて

bottom、または省略すると配列の最後に追加。

それ以外は、最初に追加される。

追加するルールが、既存のルールと競合したり

り、上書きしたい場合は、優先度を高くした

いため、top などと指定する必要がある。

Prime Strategy

add_rewrite_ruleのハマリポイント

- ④ パーマリンク設定の変更を保存で、リライトルールを再生成を行う必要がある
- ④ 条件の多いルールが先に追加されるよう、順番に注意

Prime Strategy

- ④ add_rewrite_rule関数を使えば、独自のルールを追加可能
- ④ ルールが適用される優先度を考え、positionパラメーターを指定しよう
- ④ ハマリポイントに注意

Prime Strategy

カスタムパーマリンクの実践

Prime Strategy

作成者の月別アーカイブを表示できるように
してみる。アクセスするURLは、

author/(作成者名)/(年)/(月)/

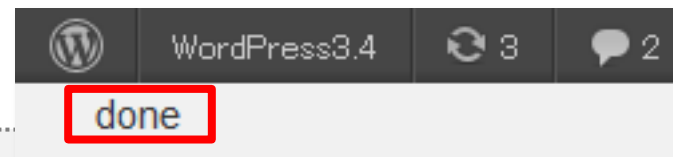
Prime Strategy

まずは、404になることを確認しておく。



Prime Strategy

functions.phpにルールを追加する関数を追記する。関数が実行され、画面上に done の文字が表示されることを確認

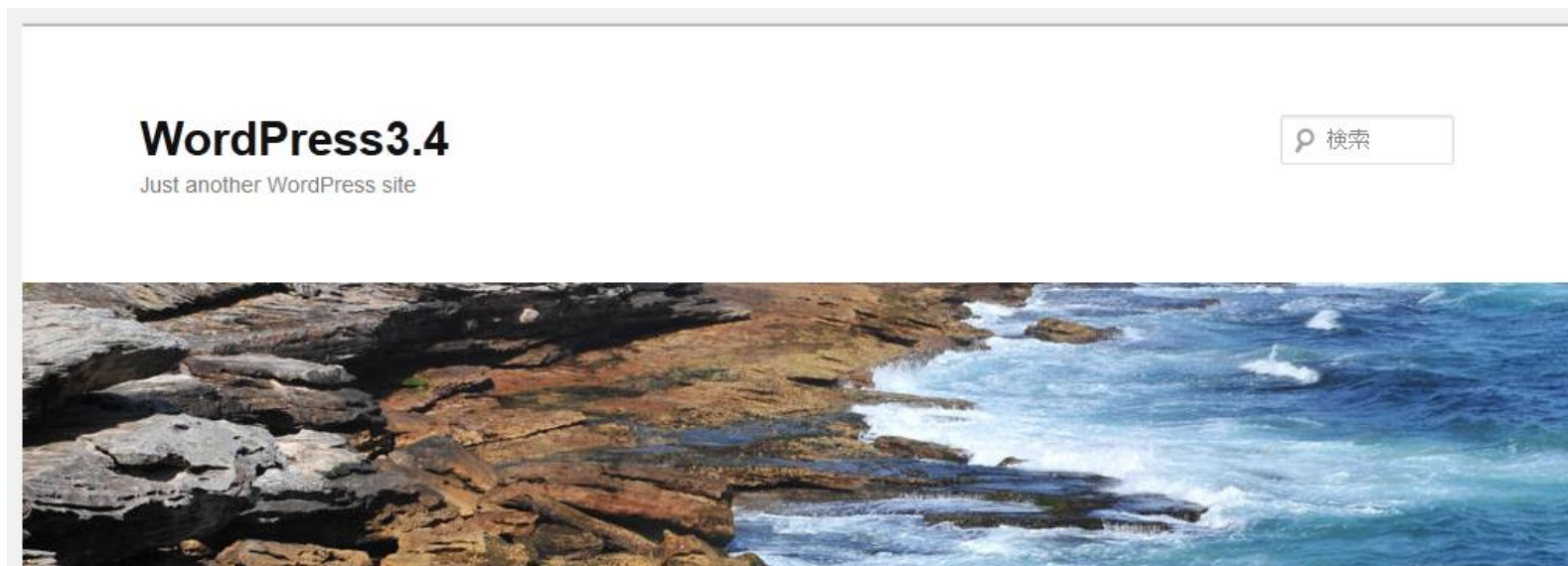


```
function add_author_monthly_archive_rule() {  
    echo 'done';  
}  
  
add_action( 'init', 'add_author_monthly_archive_rule' );
```

echo 'done!'; の一文を消して、関数内に、
add_rewrite_rule 関数を記述。


```
add_rewrite_rule(  
    'author/([^\s/]+)/([0-9]{4})/([0-9]{2})/?',  
    'index.php?author_name=$matches[1]&year=$ma  
tches[2]&monthnum=$matches[3]'  
);
```

パーマリンク設定の保存を行ってから、想定したアドレスにアクセスして、表示ができて
いるか確認。Debug Barでも確認してみよう。



実は、ページ送り用のルールも必要。

先のadd_rewrite_ruleの上に追加。

```
add_rewrite_rule(  
    'author/([^\s/]+)/([0-9]{4})/([0-9]{2})/page/([0-9]+)/?',  
    'index.php?author_name=$matches[1]&year=$ma   
tches[2]&monthnum=$matches[3]&paged=$matches[4]'  
);
```



ご清聴ありがとうございました。

Prime Strategy