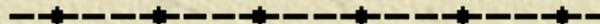




WordPress 2.8用の ウィジェットの開発

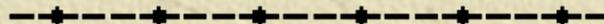
藤本 壱



今日のアジェンダ

- ✧ 自己紹介
- ✧ 新しいウィジェットAPIの概要
- ✧ 作成するウィジェット(ステータス表示)
- ✧ クラスとコンストラクタの定義
- ✧ formメソッド
- ✧ updateメソッド
- ✧ widgetメソッド
- ✧ ウィジェットの登録
- ✧ まとめ

自己紹介



自己紹介

- ✦ 藤本 壱(ふじもと はじめ)
- ✦ フリーライター
 - パソコン関係
 - ここ数年はMovable Type関連の書籍を多数執筆
 - マネー関係
 - 主に株式投資関係
- ✦ Movable TypeやWordPressのプラグインを開発

ブログ

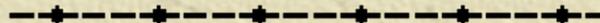
✦ <http://www.h-fj.com/blog/>

The screenshot shows a Mozilla Firefox browser window displaying the blog 'The blog of H.Fujimoto'. The address bar shows the URL <http://www.h-fj.com/blog/archives/2009/06/17-114008.php>. The page features a blue header with the title 'The blog of H.Fujimoto' and the subtitle 'Personal Computer, Financial Planning, etc...'. A navigation menu includes 'Home', 'MTPlugin', 'Guestbook', 'About', and 'Contact'. The main content area is divided into several sections:

- お知らせ** (Notice): A post dated 2009年2月21日 (February 21, 2009) titled '初体験 Excel VBA' (First Experience Excel VBA). The text mentions the release of a new book '初体験Excel VBA' and describes it as an introductory guide to Excel VBA, focusing on practical examples. It also mentions that the author has written many other books and provides a link to view them.
- MT用横型Flashカレンダー (V3.0) をWordPress2.7以降で使う** (Using MT Horizontal Flash Calendar (V3.0) with WordPress 2.7 and later): A post dated 2009年6月17日 11:40 (June 17, 2009, 11:40) with 0 comments and 0 trackbacks. The text discusses the author's experience with using the calendar plugin on WordPress 2.7 and later, mentioning that the plugin was not working as intended and that the author has since updated it.

On the right side of the page, there is a section for 'Movable Type Developer's Guide Volume 1' with a grid of icons representing the book's content. Below this, there is a section for '応援しています' (Supporting) featuring a book cover for 'CSSNITE LP6 CMSリベンジ編' (CSSNITE LP6 CMS Revival Edition) by Belzard Shinshu, with a price of 9,906.27. At the bottom right, there is a search bar and a '購読' (Subscribe) section with links for Atom and RSS 2.0 feeds.

新しい ウィジェットAPIの 概要



WP_Widgetクラス

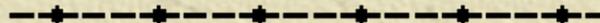
- ✦ WP2.8でWP_Widgetクラスが新設
- ✦ ウィジェットの基本動作を定義したクラス
- ✦ WP標準のウィジェットもWP_Widgetクラスを使って定義しなおされている
- ✦ これまでのAPIよりもウィジェット作成が簡単
- ✦ 同一ウィジェットを複数回追加する機能が標準装備されている

WP2.8用ウィジェットの開発手順

- ✦ WP_Widgetクラスを継承してオリジナルのウィジェットのクラスを定義
- ✦ 4つの関数(メソッド)を定義

関数(メソッド)	内容
コンストラクタ	ウィジェットの名前等の定義
formメソッド	ウィジェットの設定フォームを出力
updateメソッド	ウィジェットの設定を保存
widgetメソッド	ウィジェットの内容を出力

作成するウィジェット (ステータス表示)



ステータス表示ウィジェット

✦ 最終更新日 / 投稿数 / コメント数を表示

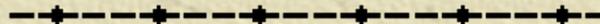


ウィジェットの設定

✦ ウィジェットの管理画面で設定可能

The screenshot shows the WordPress Widgets management interface in a Mozilla Firefox browser window. The browser title is "ウィジェット < My First Weblog - WordPress - Mozilla Firefox". The address bar shows the site URL. The page title is "My First Weblog" with a "サイトを表示" button. The user is logged in as "adminさん" with a "ターボ" mode and a "ログアウト" button. The main content area is titled "ウィジェット" and shows a list of available widgets under the heading "利用できるウィジェット". The widgets listed are: アーカイブ (Blog archive), カテゴリー (Categories), カレンダー (Calendar), ステータス (Status), タグクラウド (Tag cloud), テキスト (Text), ページ (Pages), and メタ情報 (Meta information). On the right side, a sidebar titled "サイドバー 1" is shown, and the "ステータス" widget is highlighted with a red box. The "ステータス" widget configuration shows a "タイトル:" field and "削除 | 閉じる" and "保存" buttons. The bottom of the browser window shows "完了" (Done) and system icons.

クラスと コンストラクタの定義



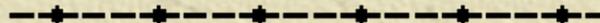
クラスとコンストラクタの定義

- ✦ WP_Widgetクラスを継承してオリジナルのウィジェットのクラスを定義
- ✦ クラス名は「WP_Widget_〇〇〇」
- ✦ コンストラクタでウィジェットの名前等を定義

クラス定義とコンストラクタ

```
class WP_Widget_Status extends WP_Widget {
    function WP_Widget_Status() {
        $widget_ops = array(
            // ウィジェットを囲む要素に割り当てるクラス名
            'classname' => 'widget_status',
            // ウィジェットの説明
            'description' => '最終更新日／投稿数／コメント数の表示'
        );
        // ウィジェットのID／名前／オプションを初期化
        $this->WP_Widget('status', 'ステータス', $widget_ops);
    }
}
```

form メソッドの定義



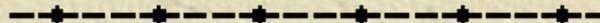
form メソッド

- ✦ ウィジェットの設定フォームを出力
- ✦ input等の入力要素でフォームを組み立てる
- ✦ パラメータの\$instanceで設定値が渡される
- ✦ \$instance ['項目名']が各項目の現在の設定値
例: \$instance['title']はtitle項目の設定値
- ✦ 各入力要素のid属性
→ \$this->get_field_id('項目名')
- ✦ 各入力要素のname属性
→ \$this->get_field_name('項目名')

form メソッドの例

```
function form($instance) {  
    // 「title」の設定からHTMLのタグを除いて$titleに代入  
    $title = strip_tags($instance['title']);  
    ?>  
    <p>  
    <label for="<?php echo $this->get_field_id('title'); ?>">  
    タイトル</label>  
    <input class="widefat" type="text"  
    id="<?php echo $this->get_field_id('title'); ?>"  
    name="<?php echo $this->get_field_name('title'); ?>"  
    value="<?php echo esc_attr($title); ?>" />  
    </p>  
    <?php  
    }
```

*update*メソッドの定義



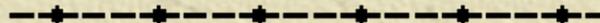
update メソッド

- ✦ 設定を保存
- ✦ パラメータの \$old_instance が元の設定値
- ✦ パラメータの \$new_instance が新しく入力された設定値
- ✦ \$instance に \$old_instance をコピー
- ✦ \$new_instance の値を適宜加工して \$instance に代入
- ✦ 戻り値として \$instance を返す

update メソッドの例

```
function update($new_instance, $old_instance) {  
    // $instanceに元の設定値($old_instance)を代入  
    $instance = $old_instance;  
    // 「title」の設定値を更新  
    $instance['title'] = strip_tags($new_instance['title']);  
    // $instanceを返す  
    return $instance;  
}
```

*widget*メソッドの定義



*widget*メソッド

- ✦ ウィジェットの内容を出力
- ✦ パラメータの\$instanceからウィジェットの設定を得る
- ✦ パラメータの\$argsから以下の情報を得る

情報	内容
\$before_widget	ウィジェットの前に出力する内容
\$after_widget	ウィジェットの後に出力する内容
\$before_title	ウィジェットのタイトルの前に出力する内容
\$after_title	ウィジェットのタイトルの後に出力する内容

- ✦ \$instance / \$argsを組み合わせてウィジェットを出力

widget メソッドの例(1)

```
function widget($args, $instance) {  
    extract($args);  
    $title = apply_filters('widget_title', empty($instance['title'])  
        ? 'ステータス' : $instance['title']);  
    echo $before_widget;  
    if ($title) {  
        echo $before_title . $title . $after_title;  
    }  
}
```

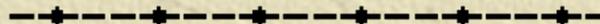
*widget*メソッドの例(2)

```
global $wpdb, $post;
$post = get_posts("numberposts=1&orderby=date&order=DESC");
$post = $posts[0];
setup_postdata($post);
$post_date = get_the_time('Y年m月d日');
$post_count = $wpdb->get_var("SELECT COUNT(*) FROM $wpdb->posts
WHERE post_status = 'publish' AND post_type = 'post'");
$comment_count = $wpdb->get_var("SELECT COUNT(*) FROM $wpdb-
>comments WHERE comment_approved = '1'");
?>
<ul>
<li>最終更新日: <?php echo $post_date; ?></li>
<li>投稿数:<?php echo $post_count; ?></li>
<li>コメント数:<?php echo $comment_count; ?></li>
</ul>
<?php
```

widget メソッドの例(3)

```
echo $after_widget;  
}
```

ウィジェットの登録

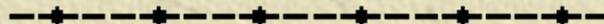


ウィジェットの登録

- ✦ WordPressにウィジェットを登録
- ✦ クラス定義の外に以下を追加

```
add_action('widgets_init', create_function(",  
'return register_widget("WP_Widget_Status");');
```

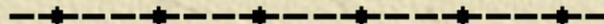
まとめ



まとめ

- ✦ WordPress 2.8でWidget APIが一新
- ✦ WP_Widgetクラスを継承してウィジェットを作成
- ✦ コンストラクタ / formメソッド / updateメソッド / widgetメソッドでウィジェットの動作を定義

おまけ



WordPress 開発系書籍近日発売

- ✦ WordPressの関数を利用してデータに柔軟にアクセス
- ✦ SQLでデータベースに直接アクセス
- ✦ プラグインの基本的な開発手順
- ✦ プラグインの国際対応
- ✦ ウィジェット、ショートコードのプラグイン開発方法
- ✦ プラグインの管理画面の開発方法

ご清聴ありがとうございました