



WP JELLY

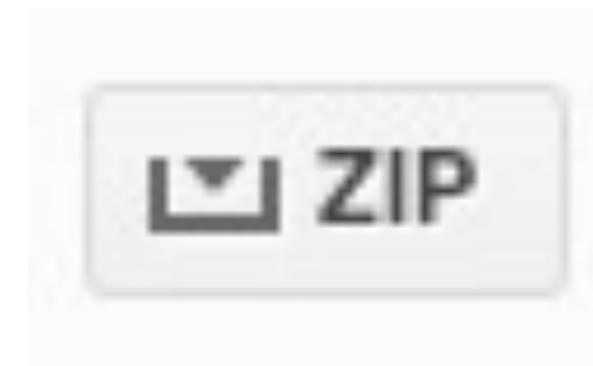
vol.2 ウィ젯型プラグインをつくろう!!



サンプルコードのダウンロード

下記のURLにアクセスして、[ZIP]ボタンをクリック

https://github.com/karad/wpjelly_vol_2



Session編が終わるまでにダウンロードしおわってください



原 一浩

- 1976年生まれ
- Creative代表(Greative is Great Creative)
- DesignWedge編集長
 - 海外デザイン情報収集家
- ECMAScript&CakePHP&Play!Framework好き



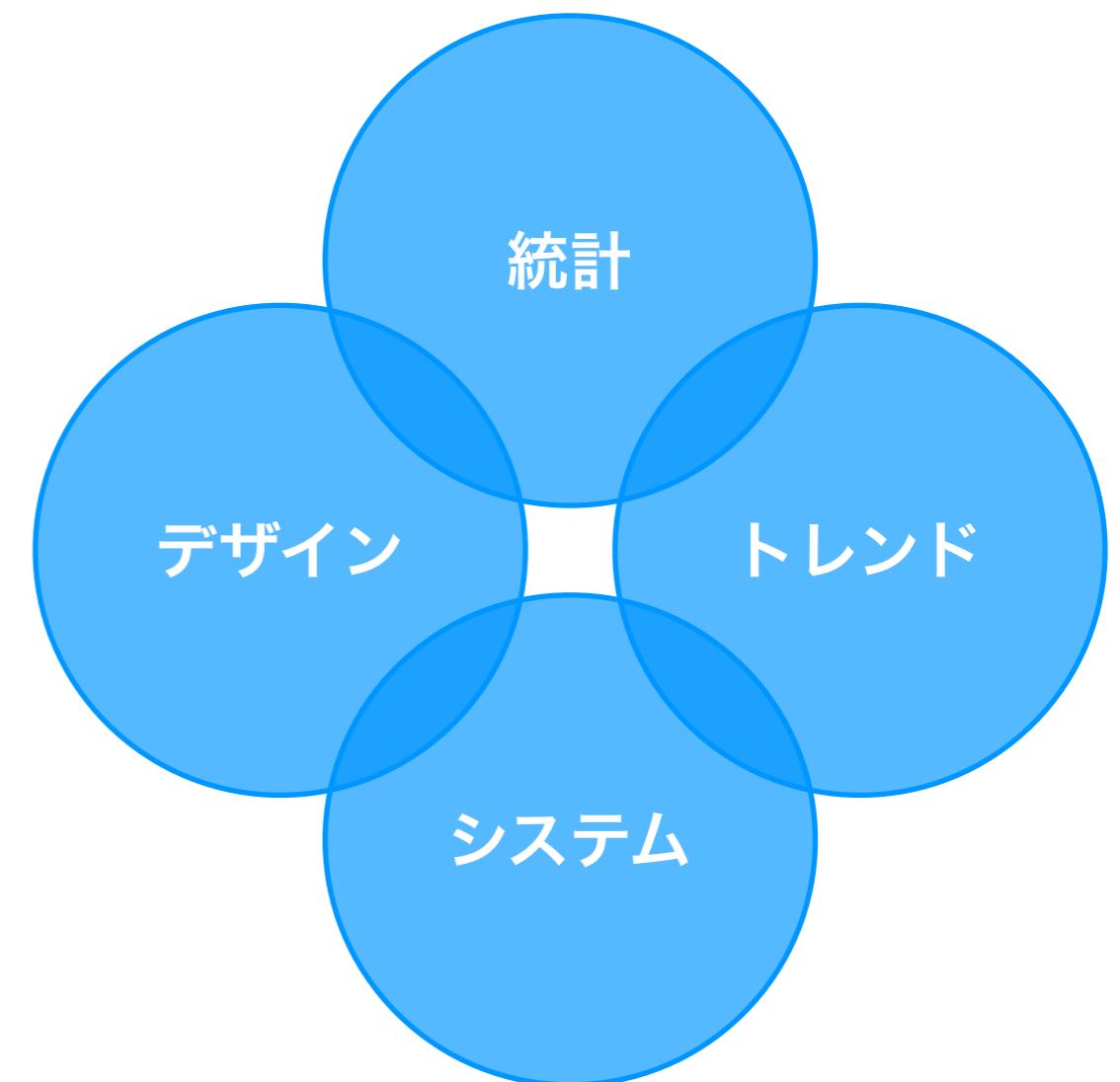
活動

- <http://creative.jp/>
- twitter id : @kara_d
- facebook : kazuhirohara



Creative (グレーティブ)

- クリエイティブルールに沿った成果物の自動化と、トレンドに即した継続的な改良を統計的手法を用いたシステムで解決
アウトプットはデザイン、手法はシステム。
- 〒135-0063
東京都江東区有明3丁目7番26号
有明フロンティアビルB棟9階
- 090-5009-2899
- karad@creative.jp





西川 伸一

- 東京都狛江市在住。2歳の娘と-5ヶ月の息子がいます。
- WordPressでサイトを作ったり、ディレクションしたりしています。
- colog.jpという子育て家族専用のブログサービスをWordPressを利用して準備しています。

プラグイン

- cawaii-admin
 - 管理画面をかわいくシンプルにして、誰でも使えるようにするプラグインです！
 - <http://wordpress.org/extend/plugins/cawaii-admin/>

活動

- twitter : @shinichiN
- facebook : shinichi.nishikawa

もくじ

- この勉強会の趣旨、目的
- 勉強会でのゴールについて
- 暫定カリキュラム

- 前回までのあらすじ
- ウィジェット型プラグインとは

Session

- 本日作成するプラグイン「忙しさウィジェット」

Develop

- 西川さん([@shinichiN](#))作のKids Dataのコードリーディング
- アイデアグループワーク
 - ・ ウィジェット型でどんなのが作れるか

Hacking

なかよく
たのしく
りすべくと

→ @itoi_shigesatoのTwitterプロフィールより。

みんなのゴール

- 各自オリジナルプラグインを作って、ブログとかで公開する

モダレーター心構え

- プラグイン作成は、PHPごりごりなデベロップ大会になりがちなので、ちょっとした技術をアイデアで活かすみたいな難易度で進められれば

進めかた

- 習うより慣れろで。理論よりまずは手を動かしてみる

カリキュラム

vol.1

- 管理画面のメニューを拡張しよう
 - 学べる事：プラグインの基本的な仕組み、書き方、管理メニューのつけかた
 - Sample:メニューからいける更新マニュアル

vol.2

- ウィジェット型プラグインを作ろう
 - 学べる事：ウィジェット型プラグインの作成、データベース保存型のプラグインを作成
 - Sample:[西川](#)さん提供の「Kid's Data」

vol.3

- フィルター型プラグインをつくってみよう
 - 学べる事：フィルターを使う系のプラグインを作成できる
 - Sample:**募集中!!**

vol.4

- 独自のショートコードを作る・入力フォームをカスタマイズする
 - 学べる事：ショートコードを作れる、入力をカスタマイズできる
 - Sample:未定

vol.5

- オリジナルプラグインを完成させよう
 - 1回目から4回目までで頑張った内容をもとにオリジナルプラグインを作成
 - 最後にLT風に発表
 - Sample:未定

宿題はやってきましたか？

どんなプラグインが作りたいか、
来週までの宿題ということでした。

作りたいプラグインを
Facebookページの



Kazuhiro Hara

WPJelly vol.1の宿題：みんなの作りたいプラグインをここに書きましょう

Like · Comment · 7 minutes ago ·

に作りたいプラグインを書いてください

URLはこちら。

→ <http://www.facebook.com/WPJelly>

ダウンロード後

- ダウンロードしたサンプル内の[wpjelly_admin_widgetフォルダ]を、普通のプラグインと同じように WP_INSTALL_PATH/wp-contents/plugins/ 以下に配置
- 管理メニューのインストール済みプラグインをクリック

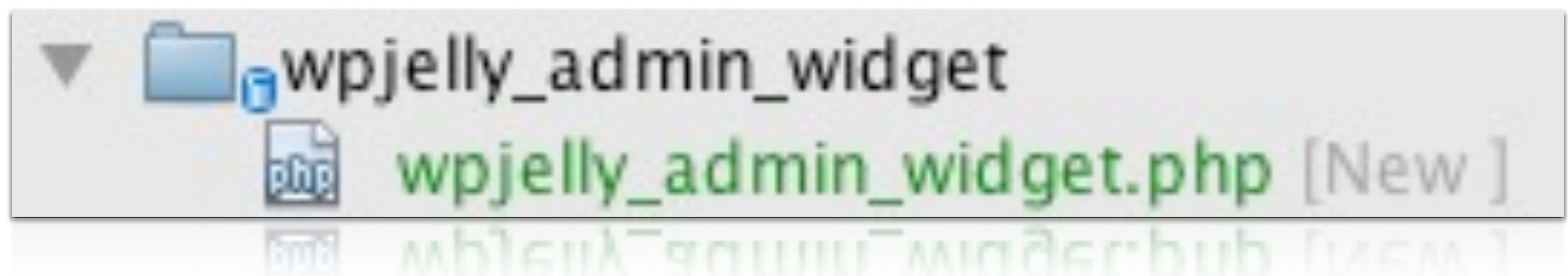


- WPJELLY-PLUGIN-ADMIN-WIDGETを[有効化]

A screenshot of the WordPress plugin settings page for 'WPJELLY-PLUGIN-ADMIN-WIDGET'. The plugin name and description are at the top. Below is a table with one row. The first column has a checkbox labeled 'WPJELLY-PLUGIN-ADMIN-WIDGET'. The second column contains the description: "'入力システム用プラグイン：現在の忙しさウィジェット'". The third column contains three buttons: 'Activate' (blue), 'Edit' (blue), and 'Delete' (red). At the bottom right of the table, there is additional text: 'Version 0.1 | Author: Kazuhiro Hara | Plugin's site'.

ソース解説

→ wpjelly_admin_widget.php … 本体



ウィジェットって？

demo

The screenshot shows the WordPress dashboard with the WP Jelly plugin active. The left sidebar is the standard WordPress admin menu. The main area is titled "WP Jelly" and "ウィジェット". It displays a grid of widget options under the heading "利用できるウィジェット". Each option has a small icon and a brief description. To the right of the main content area are several sidebar panels: "メインサイドバー" (Main Sidebar) containing "現在の忙しさ" (Current Activity), "カレンダー: カレンダー" (Calendar: Calendar), and a "保存" (Save) button; "ショーケースサイドバー" (Showcase Sidebar); "フッターエリア1" (Footer Area 1); "フッターエリア2" (Footer Area 2); and "フッターエリア3" (Footer Area 3). At the bottom of the page, there are footer links for Creative, RSS, and other site navigation.

WP Jelly

ウィジェット

利用できるウィジェット

ここから右側のサイドバーにウィジェットをドラッグすると、有効化できます。ウィジェットをここにドラッグして戻すと使用を停止し、設定を削除します。

| | | |
|-----------------------------|--|--|
| アーカイブ サイトの投稿の月別アーカイブ | カスタムメニュー カスタムメニューをウィジェットとして追加するために使います。 | カテゴリー カテゴリーのリストまたはドロップダウン表示 |
| カレンダー サイトの投稿のカレンダー | タグクラウド 最もよく使われるタグをタグクラウド形式で表示 | テキスト 任意のテキストと HTML |
| リンク リンクを表示 | 固定ページ サイト内の固定ページ | メタ情報 ログイン/ログアウト、管理、フィードと WordPress のリンク |
| 最近の投稿 サイトの最新の投稿 | 検索 サイトの検索フォーム | 最近のコメント 最近のコメント |
| RSS RSS/Atom フィードから投稿を表示 | Twenty Eleven 短冊 このウィジェットで最近のアサイド、ステータス、引用、リンクを並べることができます | 現在の忙しさ 現在の忙しさ |

こんにちは、adminさん

表示オプション ヘルプ

メインサイドバー

現在の忙しさ

カレンダー: カレンダー

タイトル: カレンダー

削除 | 閉じる 保存

ショーケースサイドバー

フッターエリア1

フッターエリア2

フッターエリア3

Creative Copyright(C) 2011 Creative - Sustainable Automation, for Creative - (<http://creative.jp/>) #wpjelly

2011年10月20日木曜日

サンプルデモ

→ Demo

The screenshot shows the WordPress dashboard with the WP Jelly plugin installed. The left sidebar has a dark theme with the following menu items:

- ダッシュボード
- 投稿 (selected)
- 投稿一覧
- 新規追加
- カテゴリー
- 投稿タグ
- システム操作
- ル
- サンプルウィ
- メディア
- メニュー
- カスタムフィールド
- アーカイブ
- アドミンメニュー

The main content area displays the WP Jelly plugin settings for the 'Now's Busy Widget'. It includes a title '現在の忙しさウィジェット' (Now's Busy Widget), a subtitle '現在の忙しさをアピールしましょう！' (Let's promote our availability!), a dropdown menu for '忙しさ' (Availability) set to '普通' (Normal), and a large yellow button labeled '忙しさを保存しました' (Availability saved). Below this is another subtitle and a dropdown menu set to 'ひま' (Free).

A modal window titled 'メインサイドバー' (Main Sidebar) is open, showing the configuration for the widget. It contains a section for '現在の忙しさ' (Availability) with the message 'このウィジェットにはオプションがありません。' (There are no options for this widget.) and a '削除 | 閉じる' (Delete | Close) button. Another section for 'カレンダー: カレンダー' (Calendar: Calendar) is also visible.

The right sidebar shows a '現在の忙しさ' (Availability) status of 'ひま' (Free) and a calendar for October 2011. The calendar table is as follows:

| 月 | 火 | 水 | 木 | 金 | 土 | 日 |
|----|----|----|----|----|----|----|
| | | | | 1 | 2 | |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 31 | | | | | | |



Previously

on WPJelly vol.01...

前回までのあらすじ

前回やったこと

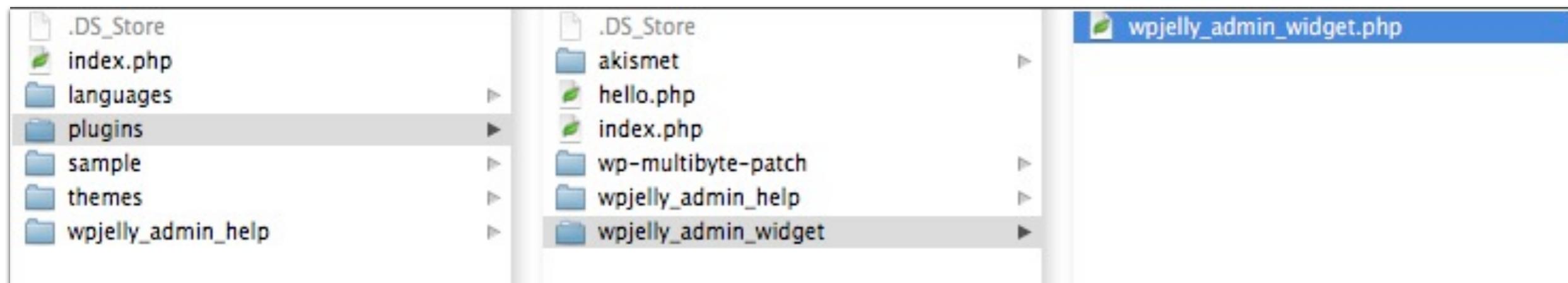
- プラグインのヘッダーの記述方法
- 管理画面のメニューのリンク先の設定
- 管理画面のメニューに追加するためのアクションの記述
- メニューの表示場所の設定
- 管理画面のメニューへのアクションの追加
- アイコンの変更
- 管理画面のスタイルの設定

プラグインディレクトリの構成

- プラグインのphpファイル
- プラグインで使う関数などが書かれたライブラリなどphpファイル
- CSSファイル
- JavaScriptファイル
- 画像ファイル

配置場所

- WP_INSTALL_PATH/wp-contents/plugins/以下



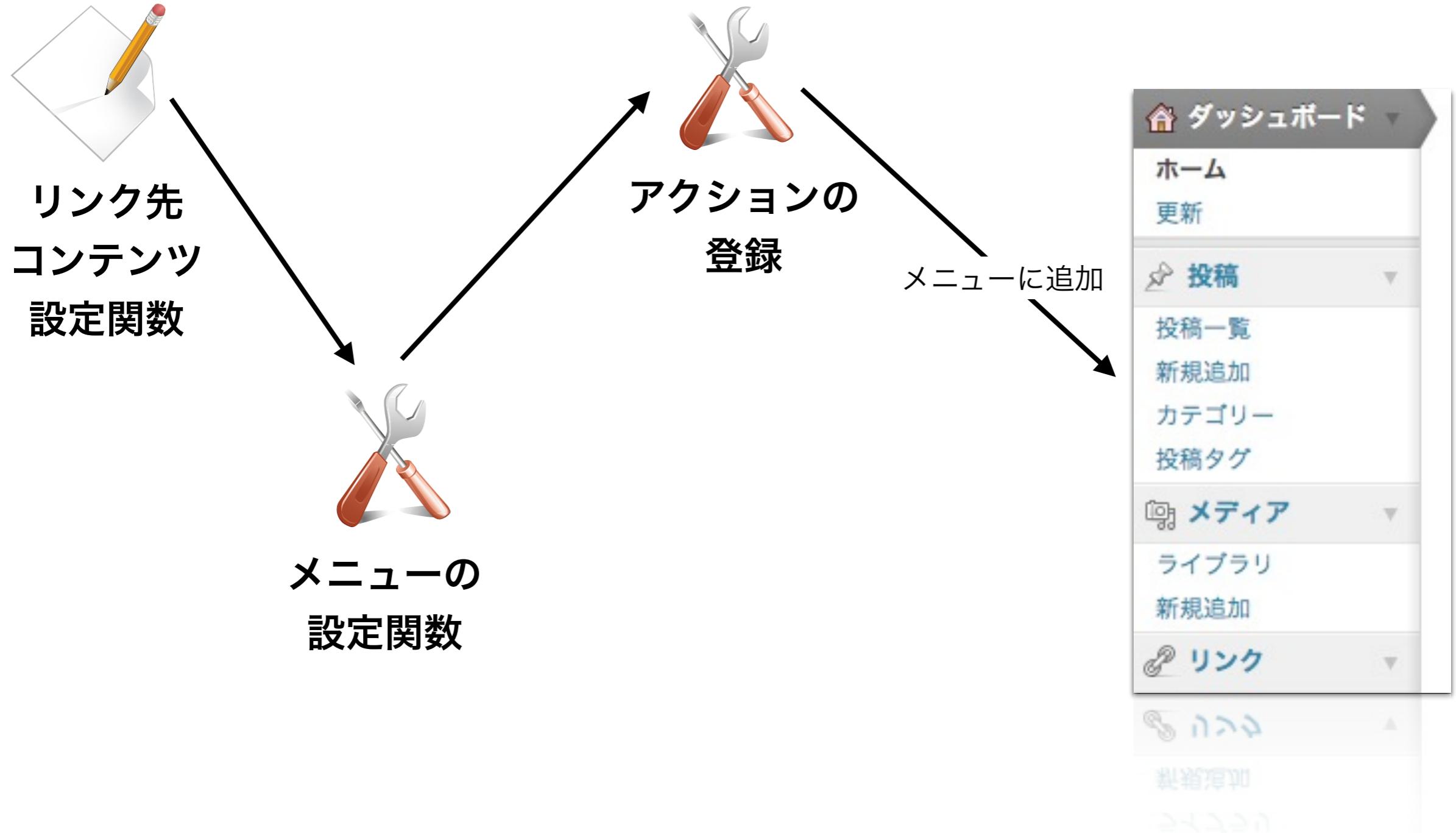
PHPファイルとそれに関係したファイルをフォルダに入れたもの

プラグインは、フックを使って、アクションを登録する

- WordPressを実行していく途中で、「これも追加ね!」っていうのを別ファイルで指定できるようにする方法



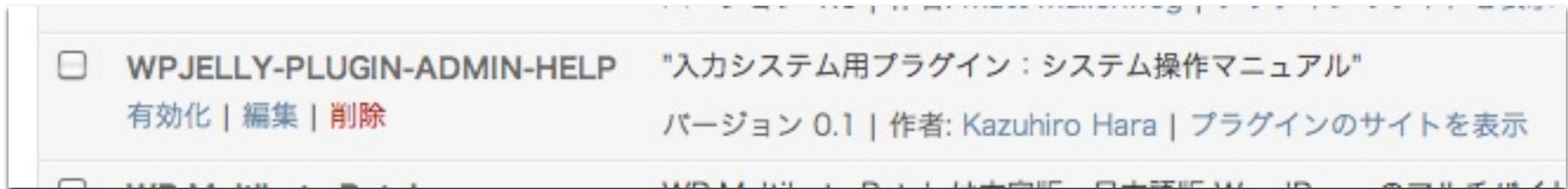
管理画面のメニューとリンクの図



プラグインのヘッダーの記述方法

こんな風に書く

```
/*
Plugin Name: WPJELLY-PLUGIN-ADMIN-HELP
Plugin URI: http://creative.jp/
Description: "入力システム用プラグイン：システム操作マニュアル"
Author: Kazuhiro Hara
Version: 0.1
Author URI: http://creative.jp/
*/
```



リンク先のHTMLは関数の中に書く

```
function wpjelly_admin_help() {  
    $wpjelly_admin_help_images = WP_PLUGIN_URL . '/  
wpjelly_admin_help/images';  
  
    $str = <<< EOF  
<div class="wrap">  
  
...  
</div>  
EOF;  
  
    echo $str;  
}
```

リンク先のコンテンツを記述

```
function wpjelly_plugin_admin_help_add_menu() {  
    $hookname = add_submenu_page(  
        'edit.php',  
        'システム操作マニュアル',  
        'システム操作マニュアル',  
        'manage_options',  
        __FILE__,  
        'wpjelly_admin_help');  
}
```

add_actionを忘れずに

```
add_action(  
    'admin_menu',  
    'wpjelly_plugin_admin_help_add_menu'  
);
```

Capabilityについて

権限のこと

Capability vs. Role Table

| Capability | Super Admin | Administrator | Editor | Author | Contributor | Subscriber |
|------------------------|-------------|---------------|--------|--------|-------------|------------|
| manage_network | | | | | | |
| manage_sites | | | | | | |
| manage_network_users | | | | | | |
| manage_network_themes | | | | | | |
| manage_network_options | | | | | | |
| activate_plugins | | | | | | |
| add_users | | | | | | |
| create_users | | | | | | |
| delete_plugins | | | | | | |
| delete_themes | | | | | | |
| delete_users | | | | | | |
| edit_files | | | | | | |
| edit_plugins | | | | | | |
| edit_themes | | | | | | |
| edit_users | | | | | | |

[icon-edit]の[edit]部分を書き換える

→ メインメニューのリンク先.phpを参照しよう

```
<div class="wrap">  
  <div id="icon-edit" class="icon32"><br></div>  
<h2>システム操作マニュアル</h2>  
</div>
```

用意されているアイコン



CSSを読み込んで登録する関数を記述

```
function wpjelly_add_css_all() {  
    $wpjelly_admin_help_css_path =  
        WP_PLUGIN_URL . '/wpjelly_admin_help/css';  
    wp_register_style('wpjelly_admin_help_css',  
        $wpjelly_admin_help_css_path . '/wpjelly.css');  
    wp_enqueue_style('wpjelly_admin_help_css');  
}
```

add_actionでスタイルを追加

```
function wpjelly_plugin_admin_help_add_menu() {  
    $hookname = add_submenu_page(  
        'edit.php',  
        'システム操作マニュアル',  
        'システム操作マニュアル',  
        'manage_options',  
        __FILE__,  
        'wpjelly_admin_help');  
  
    add_action('admin_print_styles', 'wpjelly_add_css_all');  
}
```

第1引数を、他のメインメニューのリンク先に変更

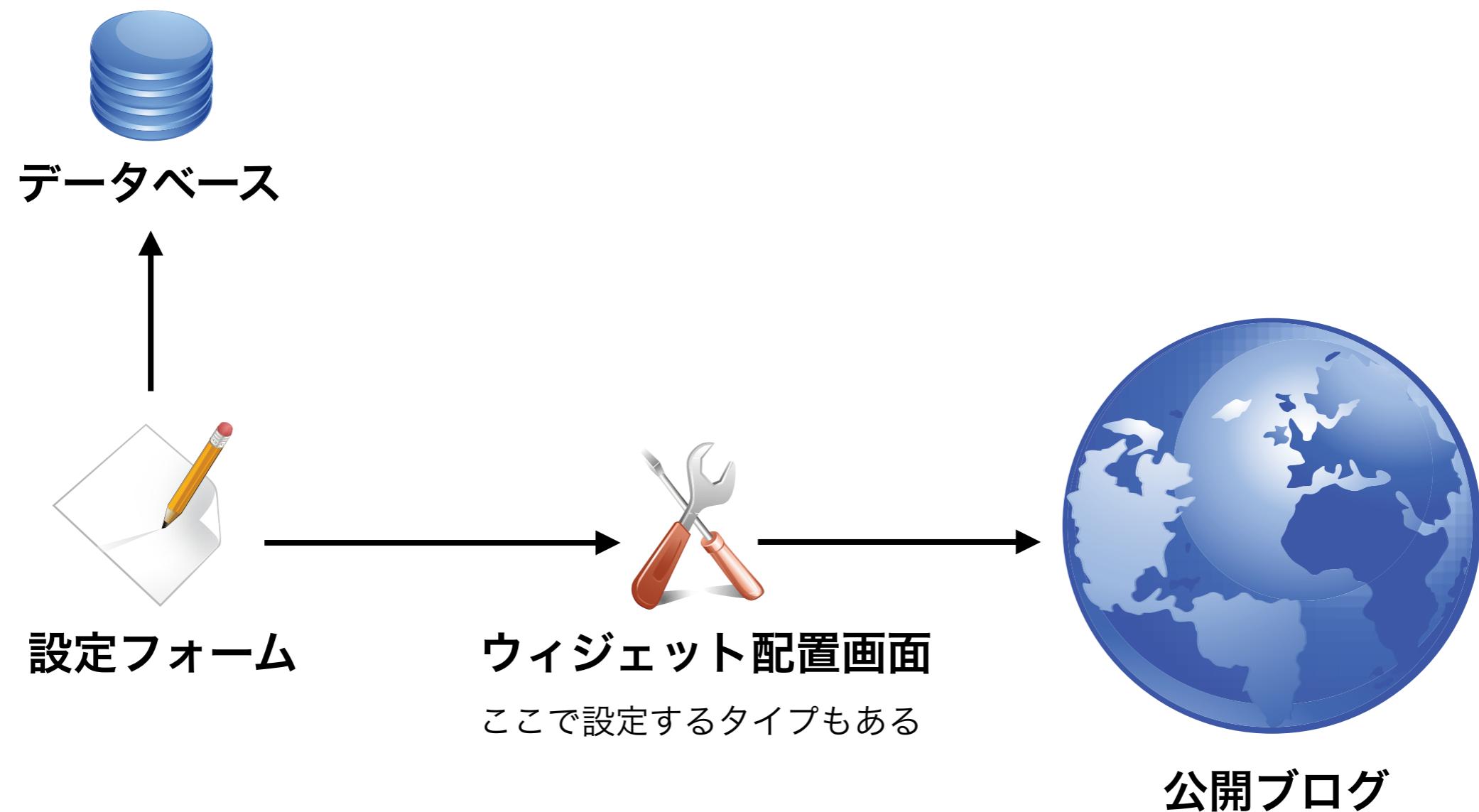
```
function wpjelly_plugin_admin_help_add_menu() {  
    $hookname = add_submenu_page(  
        'edit.php',  
        'システム操作マニュアル',  
        'システム操作マニュアル',  
        'manage_options',  
        __FILE__,  
        'wpjelly_admin_help');  
}
```



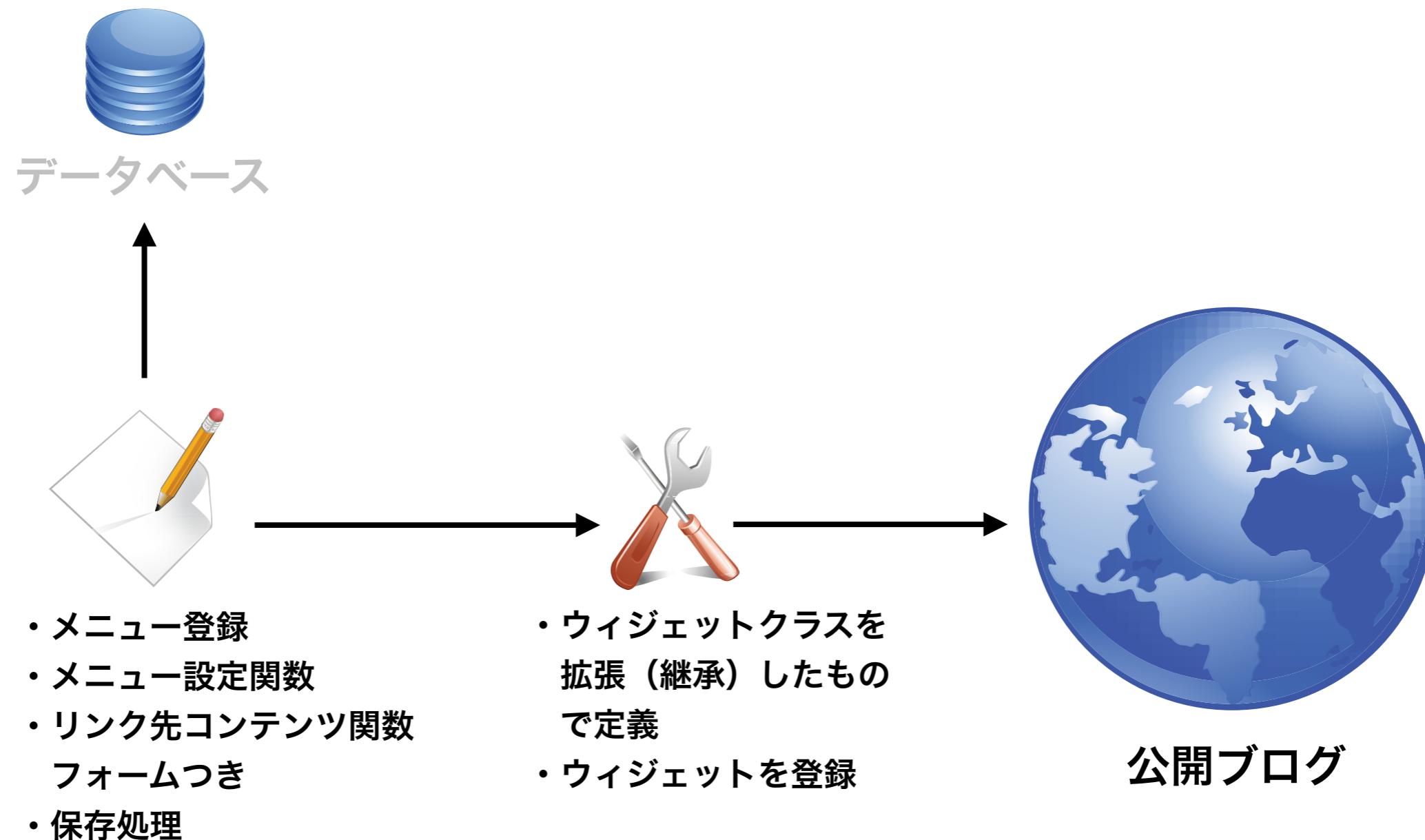
Session

WordPressのプラグインについて知ろう!!

ウィジェットのしくみ



最低限作らないといけないもの





Develop

サンプルを動かしてみよう!!
面倒でも最初は入力(写経)してみよう

続けて連結

```
$var .= "hoge";
```

\$works[0]["value"]と\$works[0]["text"]に入る

```
$works = array(  
    array('value' => '忙しい', 'text' => '忙しい'),  
);
```

\$works内の要素を1つずつ\$workに入れる

```
foreach ($works as $work) {  
    处理  
}
```

\$fooと\$hooが同じだったら「同じです」と返し、
違う場合は「違います」と返して\$hogeに入れる

```
$hoge = ($foo == $hoo) ? '同じです' : '違います';
```

\$hogeが空でなくて、なおかつ中身がyesの場合

```
if ( !empty($hoge) && $hoge === 'yes' ) {  
    处理  
}
```

サンプルデモ

→ サンプルを構築してみよう

The screenshot shows the WordPress dashboard with the sidebar menu open. The '投稿' (Posts) option is selected. On the right, the WP Jelly plugin settings page is displayed. The title is '現在の忙しさウィジェット' (Current Busy Widget). Below it, the text says '現在の忙しさをアピールしましょう！' (Let's promote our current busy status!). A dropdown menu is set to '普通' (Normal). A blue '保存' (Save) button is visible at the bottom left.

メニューの登録は前回と同様にこのコード

→ こちらのコードは、既に入力済み

```
add_action(  
    'admin_menu',  
    'wpjelly_plugin_admin_widget_add_menu'  
);
```

メニューの設定も前回と同様

→ こちらも入力済み

```
function wpjelly_plugin_admin_widget_add_menu() {  
    $hookname = add_submenu_page(  
        'edit.php',  
        'サンプルウィジェット',  
        'サンプルウィジェット',  
        'manage_options',  
        __FILE__,  
        'wpjelly_admin_widget');  
}
```

メニューからいけるリンク先コンテンツのポイント

下記コードを、「// ここに挿入」の下に入れます

- ここでは、フォーム入力があった場合に、保存処理を行っています

```
if ( !empty($_POST['posted']) && $_POST['posted'] === 'yes' ) {  
    check_admin_referer('wpjelly_admin_widget', 'creative');  
    update_option(  
        'admin_widget',  
        strip_tags($_POST['admin_widget']));  
    echo '<div class="updated"><p><strong>忙しさを保存しました</strong></p></div>';  
}
```

下記コードを、「// ここに挿入2-1」の下に入れます

- フォームの選択項目を設定しています

```
$works = array(  
    array('value' => '忙しい', 'text' => '忙しい'),  
    array('value' => '普通', 'text' => '普通'),  
    array('value' => 'ひま', 'text' => 'ひま'),  
);
```

メニューからいけるリンク先コンテンツのポイント

下記コードを、「// ここに挿入2-2」の下に入れます

- 保存されているオプションをデータベースより取得

```
$selected_value = get_option('admin_widget');
```

メニューからいけるリンク先コンテンツのポイント

下記コードを、「// ここに挿入2-3」の下に入れます

- 先ほどの選択項目の配列からoptionタグを生成します

```
foreach ($works as $work) {  
    $selected = ($selected_value == $work['value']) ?  
        ' selected="selected"' : '';  
  
    $str .= '<option value="' .  
        esc_attr($work['value']) . '" ' .  
        $selected . '>' .  
        esc_attr($work['text']) .  
        '</option>;  
}  
}
```

メニューからいけるリンク先コンテンツのポイント

下記コードを、「// ここに挿入3」の下に入れます

→ リファラーチェック用のhiddenタグを取得

```
$str .= wp_nonce_field(  
    'wpjelly_admin_widget',  
    'creative',  
    true,  
    false  
);
```

下記コードを、「// ここに挿入4」の下に入れます

```
class AdminWidget extends WP_Widget {  
    function AdminWidget() {  
        parent::WP_Widget(false, $name = '現在の忙しさ');  
    }  
    function widget($args, $instance) {  
        extract( $args );  
        echo $before_widget;  
        echo $before_title . $widget_name . $after_title;  
        $admin_widget = get_option('admin_widget');  
        echo '<p>' . $admin_widget . '</p>';  
        echo $after_widget;  
    }  
}
```

ウィジェットクラスのデフォルトの設定について

- \$before_widgetや、\$after_widget、\$before_title、\$after_title

```
'before_widget' => '<li id="%1$s" class="widget %2$s">',
'after_widget' => "</li>n",
'before_title' => '<h2 class="widgettitle">',
'after_title' => "</h2>n"
```

ウィジェットの登録

下記コードを、「// ここに挿入5」の下に入れます

→ ウィジェットをここで登録しています

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



Hacking

サンプル・ハッキング!!
今回はコードリーディングとディスカッション

12行目

- 設定画面へのリンクを出し、設定画面表示の関数を指定

```
function kd_child_info_admin() {  
    add_submenu_page(  
        'profile.php',  
        '子どものプロフィール',  
        '子どものプロフィール',  
        'add_users',  
        __FILE__,  
        'kd_child_info_admin_output');  
  
}  
  
add_action('admin_menu', 'kd_child_info_admin');
```

19行目

→ 入力チェック後、保存

```
if ( !empty($_POST['posted']) && $_POST['posted'] == 'yes' ) {  
    check_admin_referer('kids_my_love');  
    // 入力値を保存する処理  
    update_option('child_name_1',  
        strip_tags(stripslashes($_POST['child_name_1'])));  
    update_option('birth_year_1', intval($_POST['birth_year_1']));  
    update_option('birth_month_1',  
        intval($_POST['birth_month_1']));  
    update_option('birth_day_1', intval($_POST['birth_day_1']));  
    if ( $_POST['sex_1'] == 'boy' || $_POST['sex_1'] == 'girl' ) {  
        update_option('sex_1', $_POST['sex_1']);  
    }  
}
```

31行目

- 保存完了時はメッセージ

```
if(!empty($_POST['posted']) && $_POST['posted'] == 'yes') : ?>
<div class="updated">
    <p>
        <strong>お子さんのプロフィールを保存しました</strong>
    </p>
</div>
<?php endif; ?>
```

43行目

```
<select name="birth_year_1" id="birth_year_1">  
<?php  
$years = array(  
    array('value' => '2001', 'text'=>'2001'),  
    ....  
    array('value' => '2012', 'text'=>'2012'),  
);  
foreach ($years as $year) : ?>  
    <option value="<?php echo esc_attr($year['value']); ?>"  
    <?php if(get_option('birth_year_1') == $year['value'])){  
        echo ' selected="selected"';  
    } ?>><?php echo esc_attr($year['text']); ?></option>  
<?php endforeach; ?>  
</select>年
```

126行目

- 性別のラジオボタンの出力

```
<th scope="row">性別</th>
<td>
    <label for="boy">
        <input type="radio" name="sex_1" value="boy" id="boy"
        <?php if(get_option('sex_1') == 'boy' ){
            echo ' checked="checked"';
        } ?> />男の子</label>&ampnbsp&ampnbsp&ampnbsp
    <label for="girl">
        <input type="radio" name="sex_1" value="girl" id="girl"
        <?php if(get_option('sex_1') == 'girl' ){
            echo ' checked="checked"';
        } ?> />女の子</label>
</td>
```

143行目

→ ウィジェットのオプションや名前を設定

```
function ChildInfo() {  
    $widget_ops = array(  
        'description' => 'お子さんの誕生日のカウントダウンやイベントな  
        どを表示します。'  
    );  
    parent::WP_Widget(false, $name = '子供情報', $widget_ops);  
}
```

158行目

→ 年齢を求める各計算

```
//年齢を求める  
$now = date('Ymd');//今  
$birth_year_1 = (int)get_option('birth_year_1');  
$birth_month_1 = (int)get_option('birth_month_1');  
$birth_day_1 = (int)get_option('birth_day_1');  
$birthday = date('Ymd', mktime(0, 0, 0, $birth_month_1,  
    $birth_day_1, $birth_year_1));//誕生日  
$age = ($now - $birthday);  
$age = floor($age/10000);//年齢
```

167行目

→ 月齢の計算

```
//月齢（年齢は無視して何ヶ月か）を求める  
$m = date('m') - $birth_month_1;  
if ( $birth_day_1 > date('j') ) {  
    $m--;  
}  
if ( $m < 0 ) {  
    $m = $m+12;  
}
```

176行目

→ 敬称関係の判定

```
//性別で敬称を変える  
if ( get_option('sex_1') == 'boy' ) {  
    $sex = 'くん';  
} else {  
    $sex = 'ちゃん';  
}
```

183行目

→ 生年月日を求める

```
// 生後日数を求める関数  
function get_days_baby_lived($birthday_life, $today_life) {  
    $u_date_of_birth = strtotime($birthday_life);  
    $u_date_today = strtotime($today_life);  
    $days_baby_lived =  
        ($u_date_today-$u_date_of_birth)/86400 + 1;  
    return $days_baby_lived;  
}  
  
$days_baby_lived = get_days_baby_lived($birthday,$now);
```

192行目

```
// 次の誕生日までの日数をカウントダウンする関数

function kids_birthday_countdown($birth_month_1,$birth_day_1) {
    $coming_birth_month = $birth_month_1;
    $coming_birth_day = $birth_day_1;
    $one_day_length = 60 * 60 * 24;
    $this_year = date('Y');
    $now = time();
    $the_day = mktime(0, 0, 0, $coming_birth_month, $coming_birth_day,
$this_year);
    if ($now < $the_day) {
        $remain_day = ceil (($the_day - $now) / $one_day_length);
        return $remain_day;
    } elseif ( date ('Ymd', $now) == date ('Ymd', $the_day) ) {
        $remain_day = 0;
        return $remain_day;
    } else {
        $next_year = ++$this_year ;
        $the_day = mktime(0, 0, 0, $coming_birth_month, $coming_birth_day,
$next_year);
        $remain_day = ceil (($the_day - $now) / $one_day_length);
        return $remain_day;
    }
}
```

215行目

→ タイトルの設定

```
//タイトル  
$title = get_option('child_name_1').$sex.' ('.$age.'才'.$m.'ヶ月) ';
```

220行目

→ ウィジェット表示部分

```
<ul>
  <li><?php echo (int)$birth_year_1; ?>年<?php echo (int)
$birth_month_1; ?>月<?php echo (int)$birth_day_1; ?>日生まれ</li>
  <li>生後<?php echo $days_baby_lived; ?>日目</li>
  <li>次のお誕生日まであと<?php echo kids_birthday_countdown
($birth_month_1,$birth_day_1); ?>日</li>
</ul>
```

237行目

```
function kidsdatastyle() {
    $kids_bg_img_path = plugins_url() . '/kids-data/kids-bg-img.png';
    echo "
<style type=\"text/css\">
.kids-data {
    background: url('{\$kids_bg_img_path}') no-repeat scroll
center center transparent;
    width: 42px;
}
</style>
";
}
add_action('admin_head', 'kidsdatastyle');
```

今日のプラグインでどんなウィジェットが作れるか？

- テーブル単位でグループワーク

今日の宿題

- ウィジェット型プラグインでどんなプラグインが出来そうか？
- 各自のブログに書く

リソース

Wordpress プラグイン作成前の基礎知識

→ [Link](#)

プラグインの作成

→ [Link](#)

プラグイン API/アクションフック一覧

→ [Link](#)

プラグイン API/フィルターフック一覧

→ [Link](#)

WordPress ウィジェット API

→ [Link](#)

WordPress コーディング基準

→ [Link](#)

Zend Framework PHP 標準コーディング規約

→ [Link](#)

プラグインは下記決まりを守って書きましょう

- 文字コードは、UTF-8
- 改行コードは、なるべくLF(UNIX)
- 拡張子は.php

Now!

- コーディングガイドラインに沿いましょう
 - WordPressコーディング基準
 - Zend コーディング規約
- セキュリティに気をつけましょう
 - 守らないと、「これだからPHPer(ぺちぱー)は... とDisられます」
- パフォーマンスのことを考えましょう
 - これも後の回とかでやります



WR JELLY

vol.2 ウィジェット+プラグインをつくろう!!

JELLY
JELLY
CAFE
TOKYO



WRできました