



# WP JELLY

vol.2 ウィジェット型プラグインをつくらう!!



×



×





# WP JELLY

vol.2 ウィジェットプラグインをつくろう!!

JELLY  
JELLY  
CAFE  
TOKYO

X



X

*Greative*

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](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てください



## 原 一浩

- 1976年生まれ
- Greative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

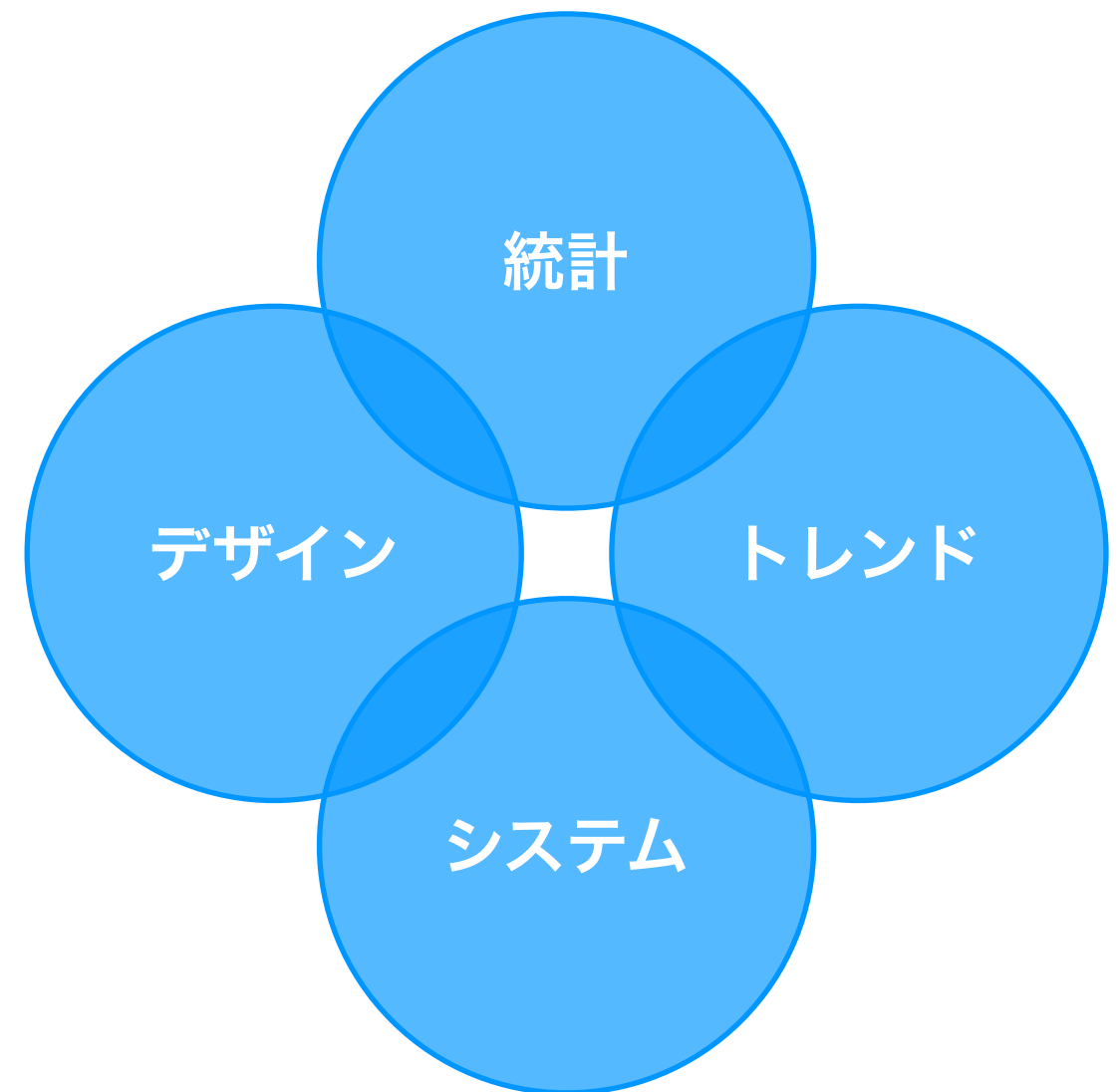
## 活動

- <http://greative.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@greative.jp



## 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

**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

## モデレーター心構え

- ➔ プラグイン作成は、PHPごりごりなデベロッパ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

- ➔ 習うより慣れるで。理論よりまずは手を動かしてみる



## 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:未定

## 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、今回のセッション中に考えてみてください。来週までの宿題です。

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。

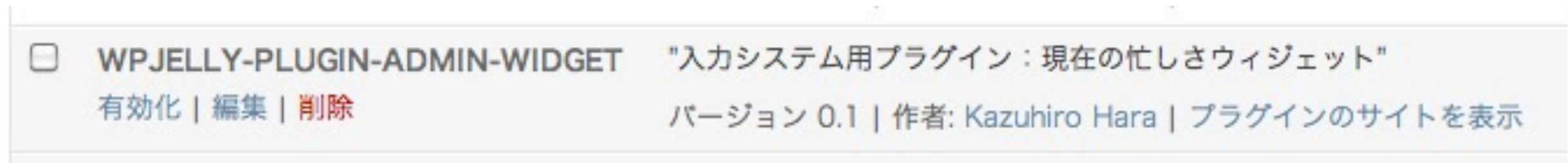
<http://www.facebook.com/WPJelly>

## ダウンロード後

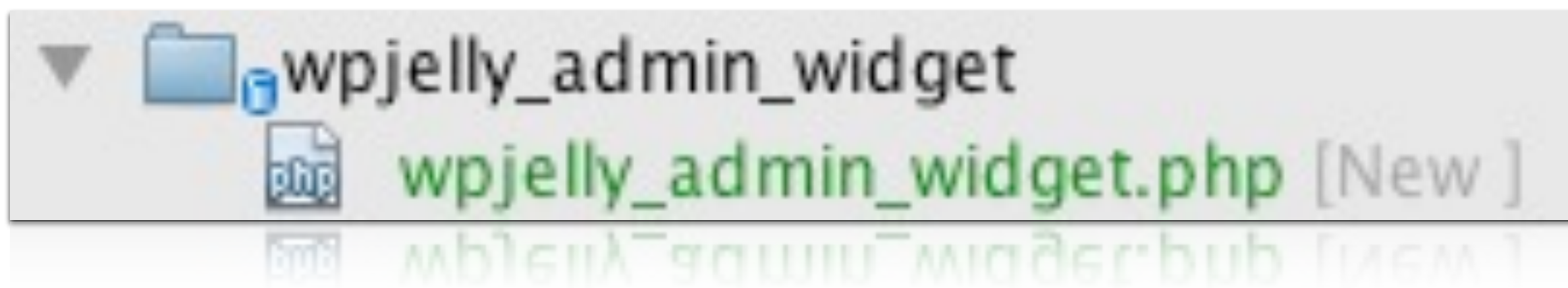
- ➔ ダウンロードしたサンプル内の[wpjelly\_admin\_widgetフォルダ]を、普通のプラグインと同じように  
WP\_INSTALL\_PATH/wp-content/plugins/  
以下に配置
- ➔ 管理メニューのインストール済みプラグインをクリック



- ➔ WPJELLY-PLUGIN-ADMIN-WIDGETを[有効化]



➔ wpjelly\_admin\_widget.php ... 本体



The screenshot displays the WP Jelly WordPress dashboard. The left sidebar contains navigation links: ダッシュボード (Dashboard), 投稿 (Posts), 投稿一覧 (Post List), 新規追加 (Add New), カテゴリー (Categories), 投稿タグ (Post Tags), システム操作 (System Operation), システム操作マニュアル (System Operation Manual), サンプルウィジェット (Sample Widgets), メディア (Media), and ユーザー (Users). The main content area shows the '現在の忙しさウィジェット' (Current Busy Widget) with a title '現在の忙しさウィジェット' and a subtitle '現在の忙しさをアピールしましょう！' (Let's show our current busyness!). Below the subtitle is a dropdown menu set to '普通' (Normal). A yellow notification bar states '忙しさを保存しました' (Saved busyness). Below the notification bar is a dropdown menu set to 'ひま' (Free). A blue '保存' (Save) button is visible. A modal window titled 'メインサイドバー' (Main Sidebar) is open, showing the '現在の忙しさ' (Current Busy) section with a 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) link. Below this is a 'カレンダー: カレンダー' (Calendar: Calendar) section. On the right side of the dashboard, there is a '現在の忙しさ' (Current Busy) section with a 'ひま' (Free) status and a 'カレンダー' (Calendar) section showing the month of October 2011. The calendar table is as follows:

2011年10月						
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

## functions.phpとプラグイン

- ➔ 個人的にはテーマに関係する部分はfunctions.phpに入れるのが良いかも
- ➔ 汎用的なものや、大きめの機能、テーマ関係の設定以外の管理画面系はプラグインにするといい

## 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-content/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**



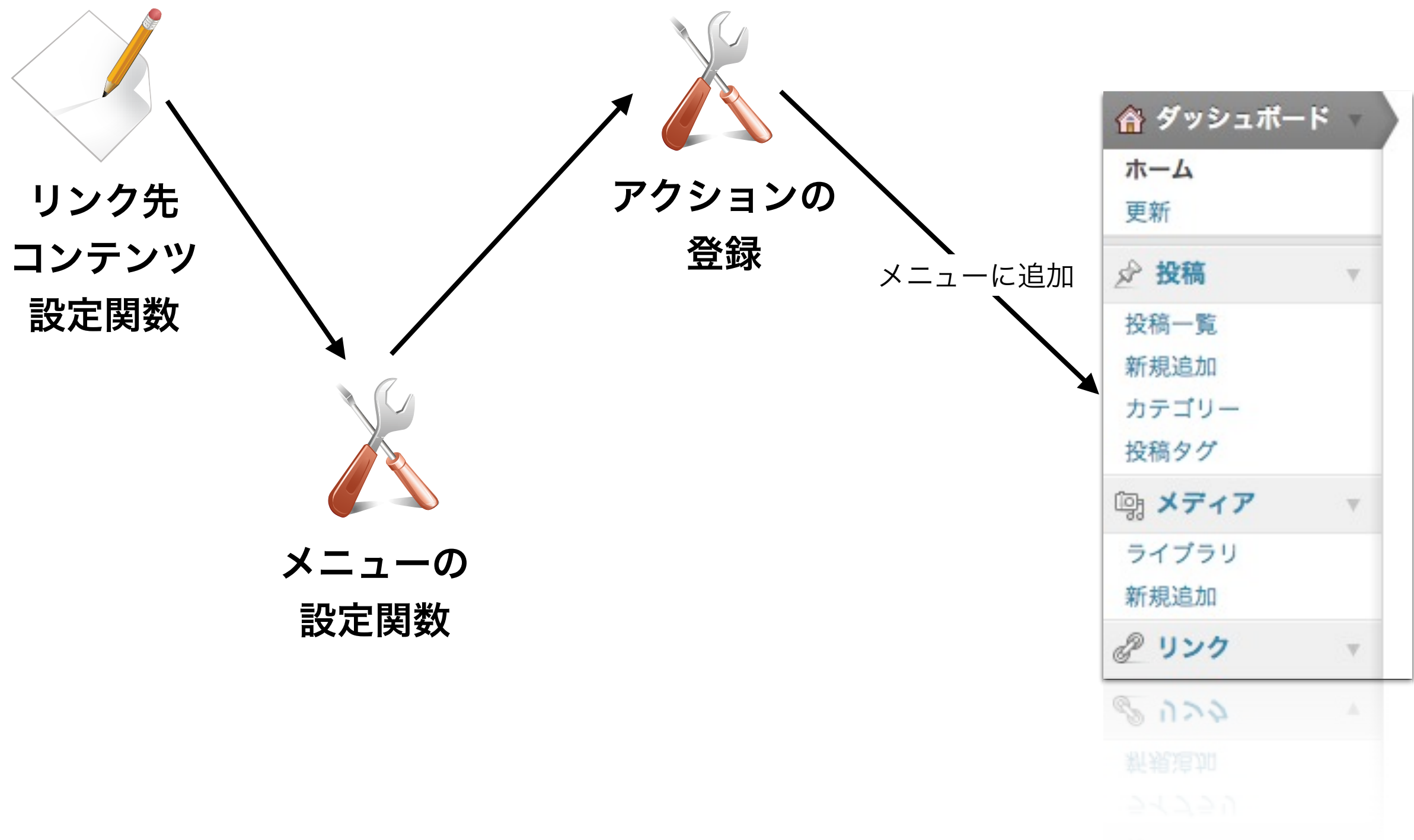
## プラグインは、フックを使って、アクションを登録する

- ➔ 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



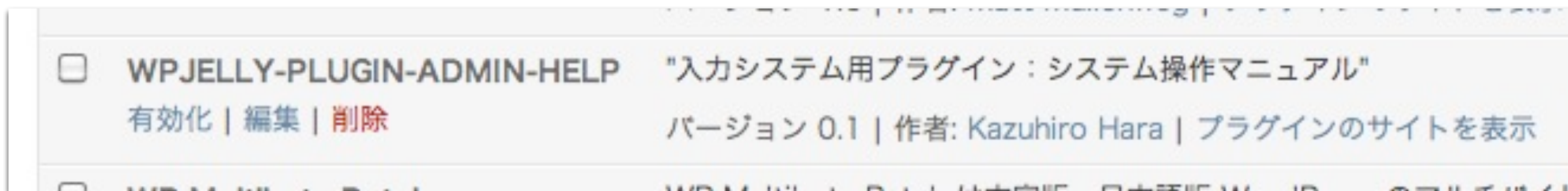
ここに1つメニューを  
追加したい!!





## こんな風を書く

```
/*  
Plugin Name: WPJELLY-PLUGIN-ADMIN-HELP  
Plugin URI: http://greative.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://greative.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');  
}
```

### [edit.php]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');  
}
```

add\_actionを忘れずに

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

## 権限のこと

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_posts						
edit_published_posts						



### [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');  
}
```

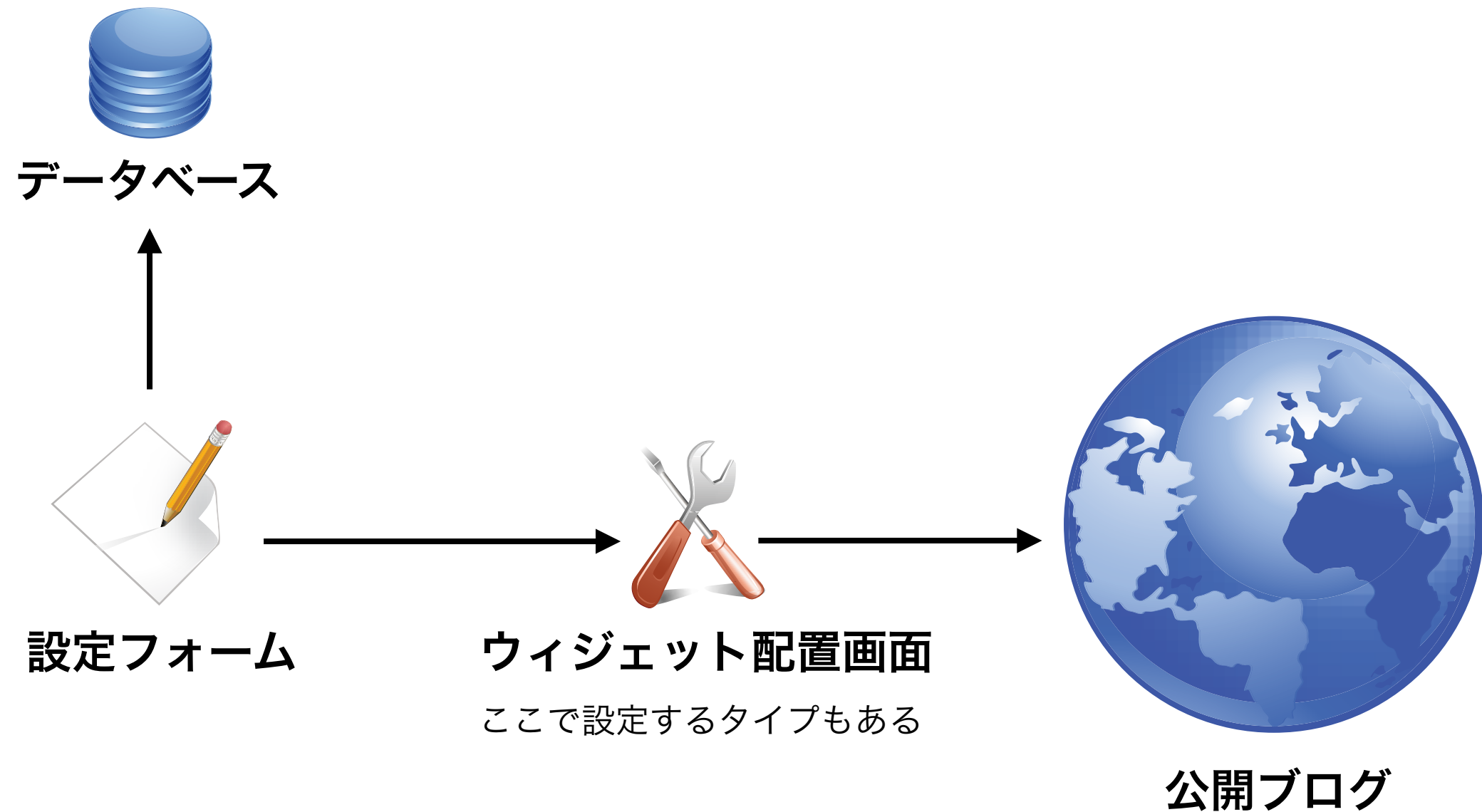
### メニュー登録の関数に1行追記しよう

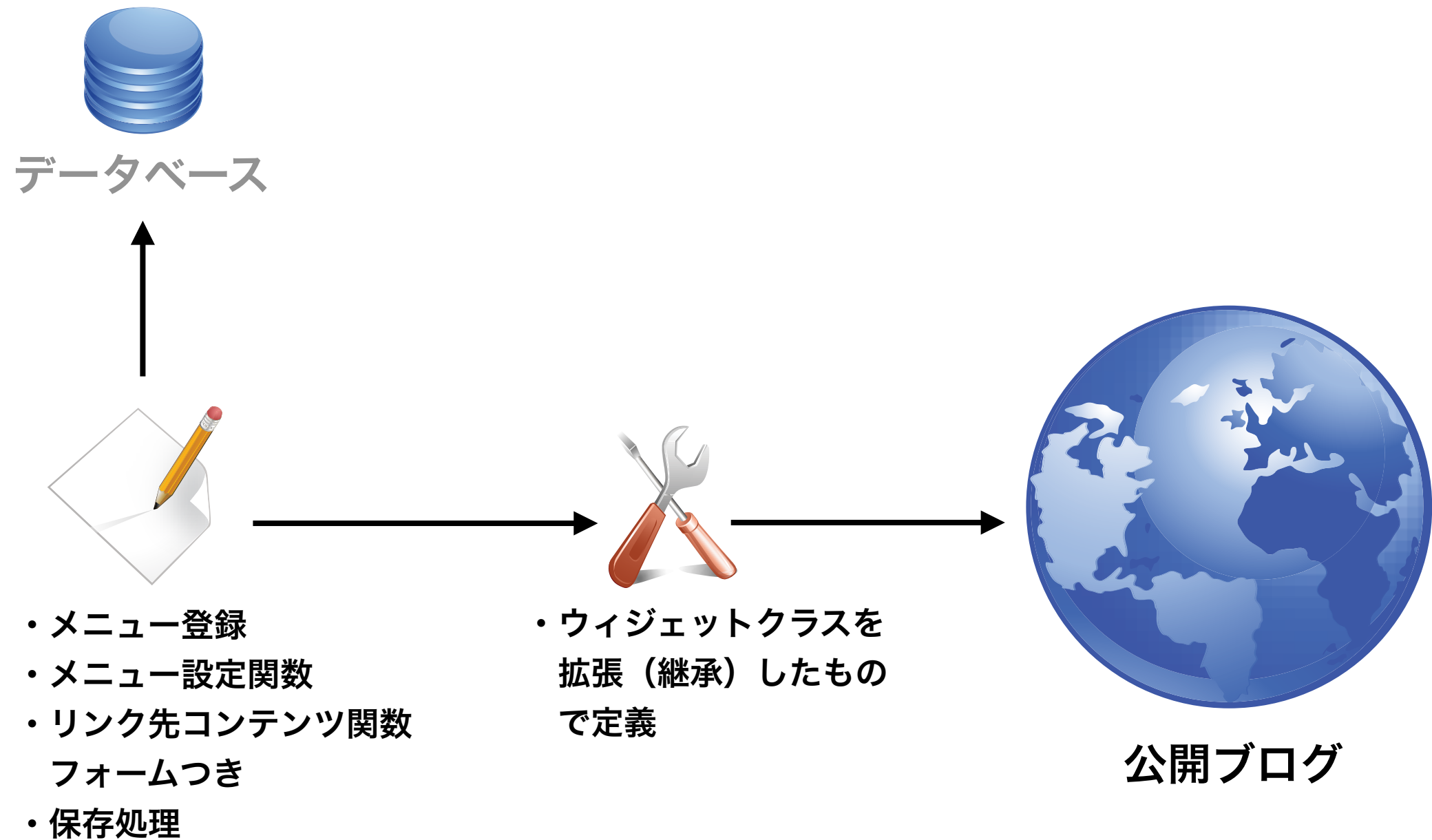
```
function wpjelly_plugin_admin_help_add_menu() {  
    $hookname = add_submenu_page(  
        'edit.php',  
        'システム操作マニュアル',  
        'システム操作マニュアル',  
        'manage_options',  
        __FILE__,  
        'wpjelly_admin_help');  
    add_action('admin_print_styles', 'wpjelly_add_css_all');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



# Session

WordPressのプラグインについて知ろう!!







# 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

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



```
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');  
}
```

## 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) {  
    処理  
}
```

\$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' ) {  
    処理  
}
```

## 挿入1

```
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>';  
}
```

## 挿入2-1

```
$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'),  
);
```

## 挿入2-2

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

## 挿入2-3

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

## 挿入3

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



## 挿入4

```
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;  
    }  
}
```

## 挿入5

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



# Hacking

サンプル・ハッキング!!  
触って、変えて、楽しむ!

## 12行目

```
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

```
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

```
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

```
<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>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
    <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

```
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  
        どもを表示します。'  
    );  
    parent::WP_Widget(false, $name = '子供情報', $widget_ops);  
}
```

## 158行目

```
//年齢を求める
```

```
$now = date('Ymd');//今
```

```
$birty_year_1 = get_option('birth_year_1');
```

```
$birth_month_1 = get_option('birth_month_1');
```

```
$birth_day_1 = get_option('birth_day_1');
```

```
$birthday = date('Ymd', mktime(0, 0, 0, $birth_month_1,  
$birth_day_1, $birty_year_1) );//誕生日
```

```
$age = ($now - $birthday);
```

```
$age = floor($age/10000);//年齢
```

## 167行目

```
//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

```
//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); //日付前をUNIXタイム化
```

```
    $u_date_today = strtotime($today_life); //日付後をUNIXタイム化
```

```
    $days_baby_lived = ($u_date_today-$u_date_of_birth)/86400 + 1; //差を24（時間）×60（分）×60（秒）で割る
```

```
    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
```

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

## 220行目

```
<ul>
```

```
<li><?php echo (int)$birty_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

## 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 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



# WP JELLY

vol.2 ウィジェットプラグインをつくろう!!

JELLY  
JELLY  
CAFE  
TOKYO

X



X

*Greative*