



WP JELLY

vol.3 フィルターフックを使ったプラグイン作成



×

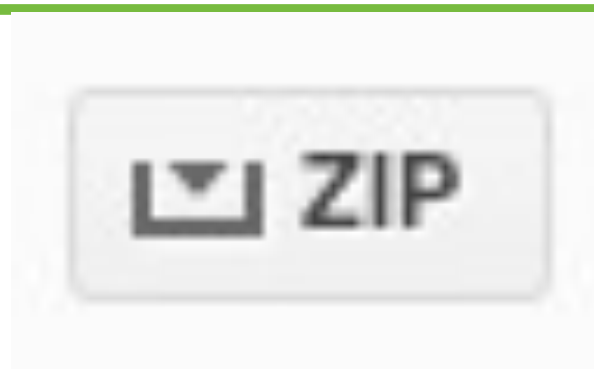


×



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_3



もしくは、WPJelly vol.3のイベントページにリンクがあります



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

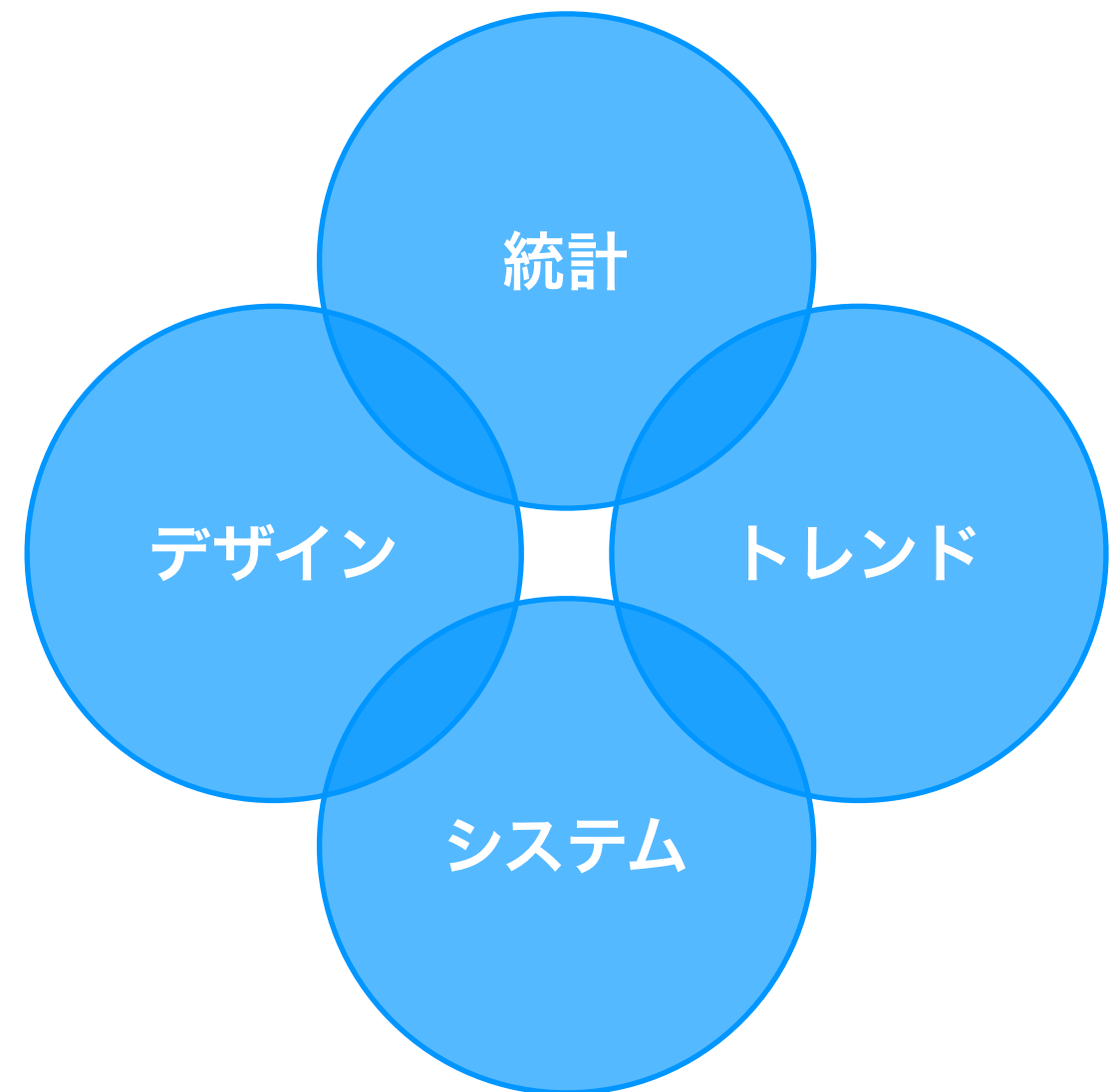
活動

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



Greative (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



WordCamp出演します！



2011年**11月27日** (日)
品川シーサイド 楽天タワー2号館



クリーンな WordPress サイトのための PHP コーディングのお作法

計画的に WordPress を拡張していくためのフレームワーク思考



星野邦敏(ほしの・くにとし)

- 株式会社コミュニティコム 代表取締役。
WordPress日本語公式サイトイベントカレンダーの更新、WordPressのイベントであるWordCampやWordBenchにスタッフやスピーカーとして関わっている。WordPressを活用した自社サービスの制作運営や、他企業向けのWordPressによるCMSパッケージの制作も手掛けている。

活動

- twitter : @khoshino
- facebook : kunitoshi.hoshino
- <http://profiles.wordpress.org/users/khoshino/>

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
- ➔ フィルター型プラグインとは

Session

- ➔ 本日の作成するプラグイン「IKZO」

Develop

- ➔ 休憩（10min）ドリンクやビールなどどうぞ

- ➔ アイデアグループワーク
 - ・ フィルター型でどんなのが作れるか
- ➔ 実際に構築してみよう

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:IKZOプラグイン

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？

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？

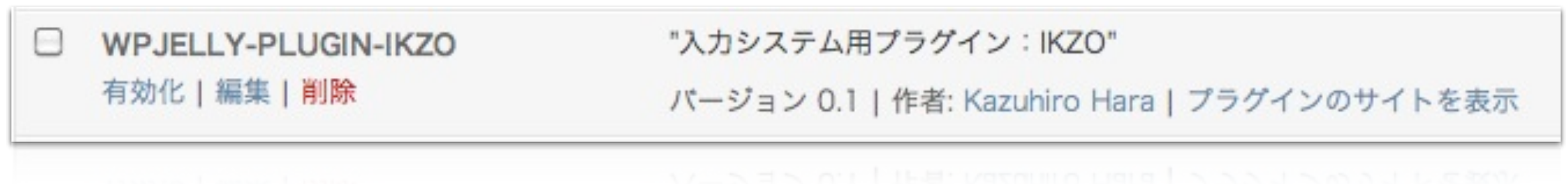
→ 各自のブログに書く

ダウンロード後

- ➔ ダウンロードしたサンプル内の[wpjelly_ikzo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

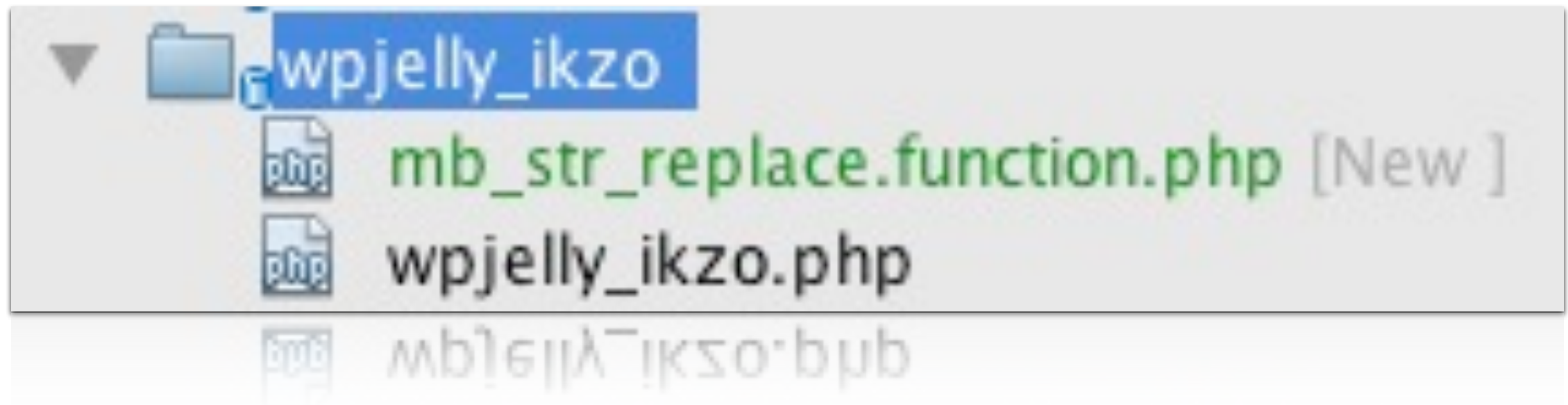


- ➔ WPJELLY-PLUGIN-IKZOを[有効化]



ソース解説

- wpjelly_ikzo.php … 本体
- mb_str_replace.function.php … マルチバイト対応のstr_replace



Hello world。

1

投稿日: [2011年10月12日](#)

WordPress へようこそ。これは最初の投稿です。編集もしくは削除してブログを始めてください！

カテゴリー: [未分類。](#) | [1件のフィードバック](#)

編集



Hello worldも電気もねえ！

1

投稿日: [2011年10月12日](#)

WordPress へようこそも電気もねえ！ これは最初の投稿ですもテレビもねえ！ 編集もしくは削除してブログを始めてください！

カテゴリー: [未分類。](#) | [1件のフィードバック](#)

編集



Previously

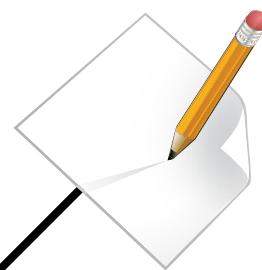
on ... 前回までのあらすじ

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

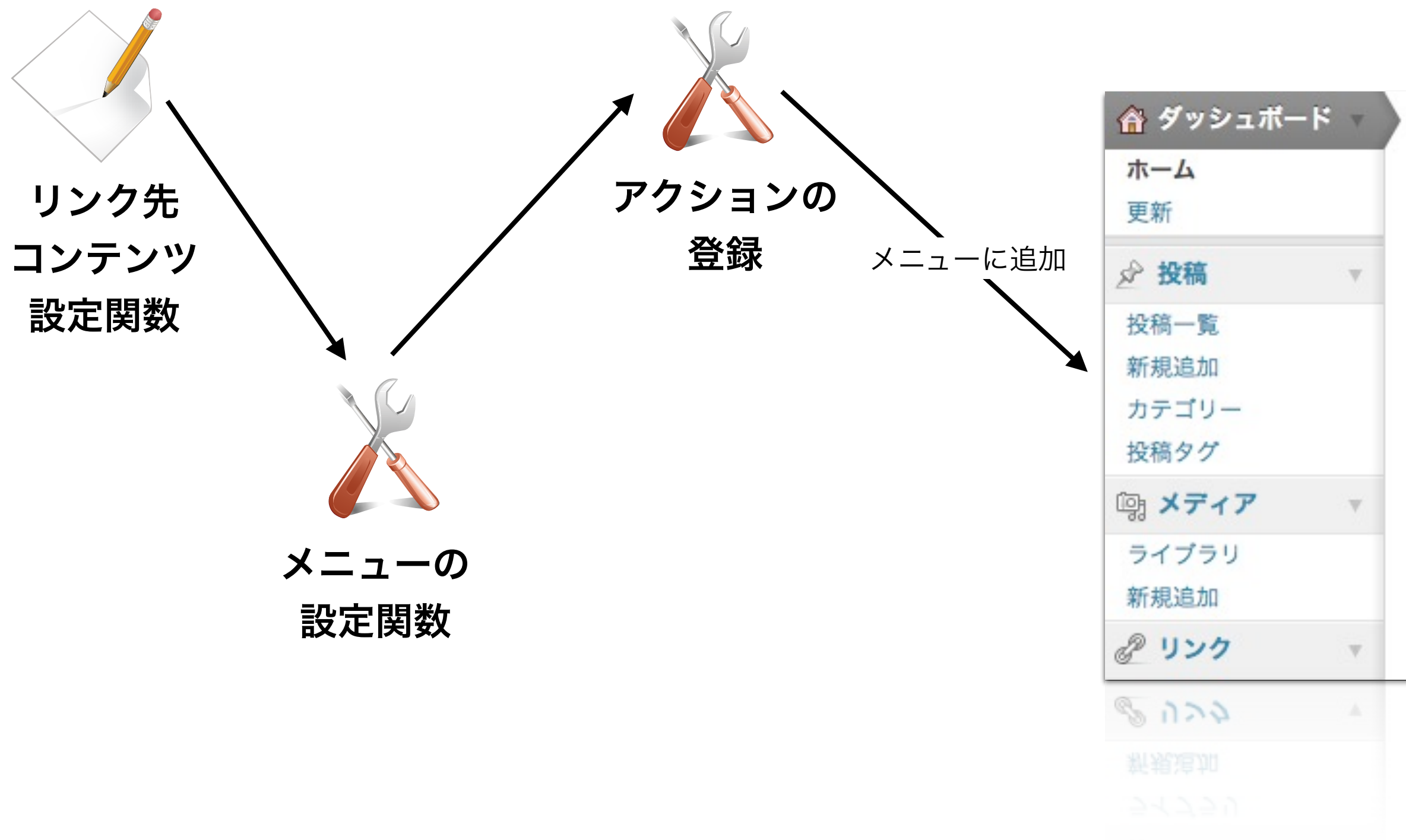
プラグインは、フックを使って、アクションを登録する

- ➔ 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



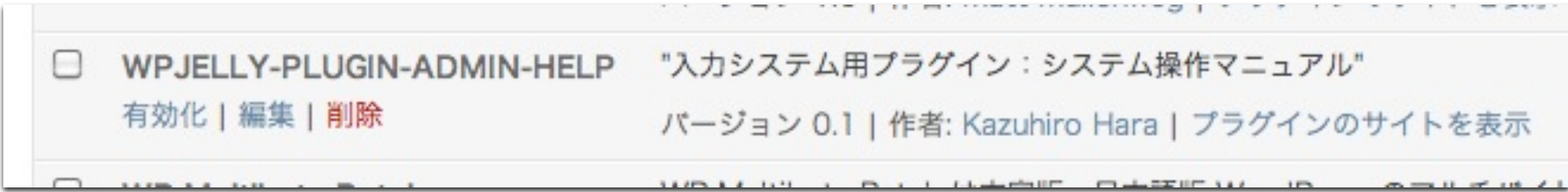
ここに1つメニューを追加したい!!





こんな風を書く

```
/*  
Plugin Name: WPJELLY-PLUGIN-ADMIN-HELP  
Plugin URI: http://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/  
*/
```



<input type="checkbox"/>	WPJELLY-PLUGIN-ADMIN-HELP	"入力システム用プラグイン：システム操作マニュアル"
	有効化 編集 削除	バージョン 0.1 作者: Kazuhiro Hara プラグインのサイトを表示

リンク先のHTMLは関数の中に書く

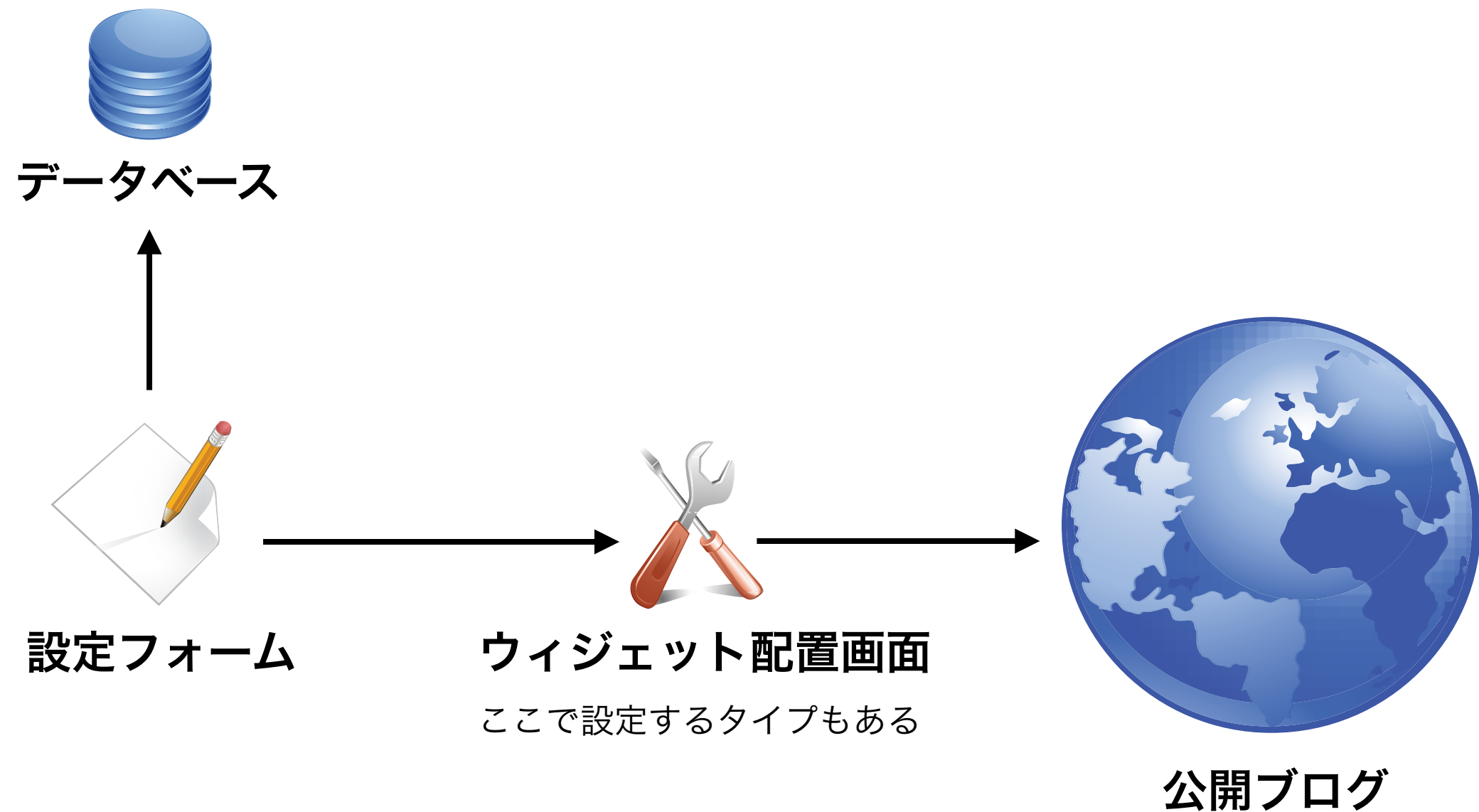
```
function wpjelly_admin_help() {  
    $wpjelly_admin_help_images = WP_PLUGIN_URL . '/  
wpjelly_admin_help/images';  
    $str = <<< EOF  
<div class="wrap">  
    ...  
</div>  
EOF;  
    echo $str;  
}
```

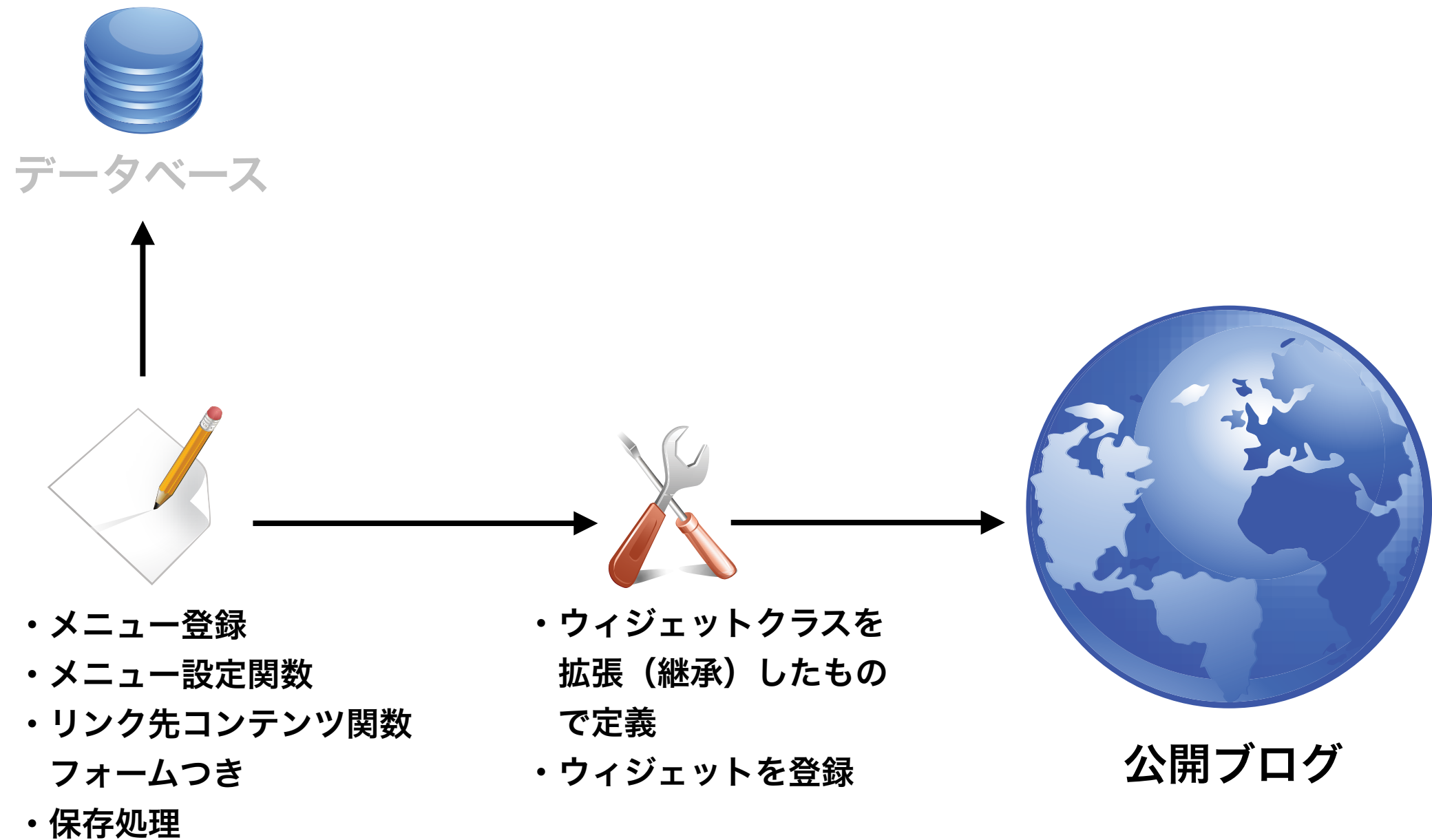
リンク先のコンテンツを記述

```
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'  
);
```





データベースへの保存

→ update_optionを使う。wp_optionsテーブルに入る

```
update_option(  
    '保存名',  
    '保存内容');
```

→ そのまま保存はせず、なんらかのエスケープとかを行う事

ほぼ定型に近い書き方

```
class HogeWidget extends WP_Widget {  
    function HogeWidget() {  
        parent::WP_Widget(false, $name = 'ウィジェット名');  
    }  
    function widget($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;  
    }  
}
```

ウィジェットの登録はこんな感じで

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



Session

WordPressのプラグインについて知ろう!!

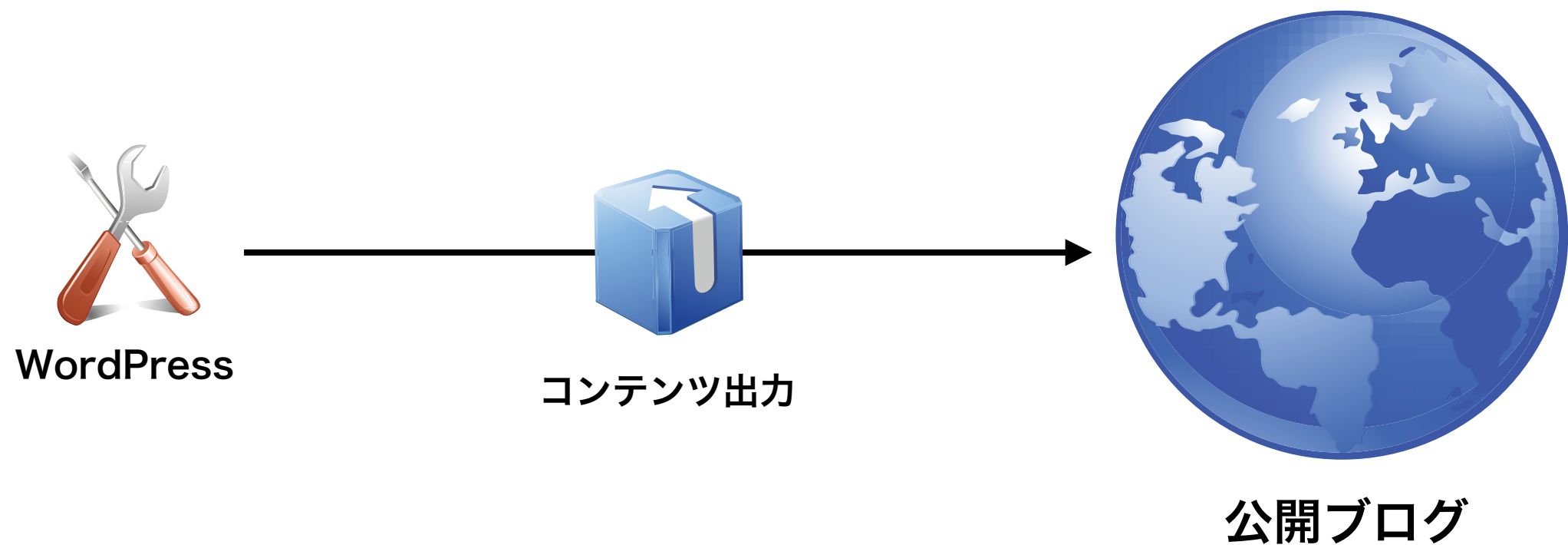
アクションフック

- ➔ 処理の途中のポイントで追加の処理を行う

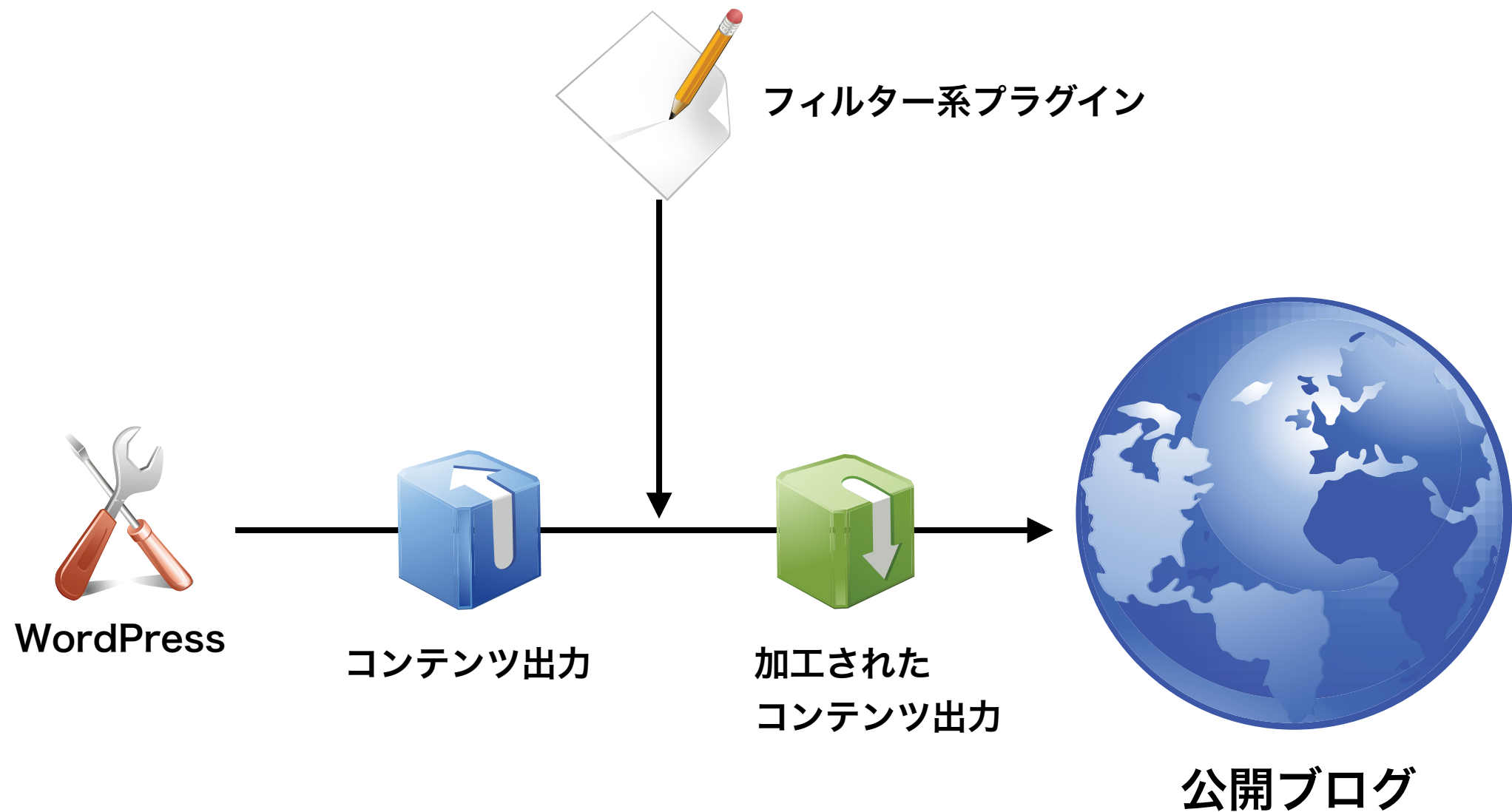
フィルターフック

- ➔ データの流れの途中で追加の加工を加える

通常の実出力



フィルターフックを使った出力



フィルターの例

- ➔ the_content
 - ・ 本文
- ➔ the_title
 - ・ 記事タイトル
- ➔ wp_list_pages
 - ・ wp_list_pages関数が生成したページのリストHTMLに適用される
- ➔ comment_text
 - ・ comment_text関数で取得したコメント本文を画面に出力する前に適用される。
- ➔ list_cats
 - ・ wp_list_categories関数でカテゴリー名に対して適用する
- ➔ the_author
 - ・ get_the_author関数で返されるユーザーの表示名に適用される。
- ➔ the_author_email
 - ・ the_author_email関数で返されるユーザーのメールアドレスに適用される。

フィルターの一覧は下記ページ

→ http://wpdocs.sourceforge.jp/%E3%83%97%E3%83%A9%E3%82%B0%E3%82%A4%E3%83%B3_API/%E3%83%95%E3%82%A3%E3%83%AB%E3%82%BF%E3%83%BC%E3%83%95%E3%83%83%E3%82%AF%E4%B8%80%E8%A6%A7

このサイトについて
目次
Codexとは
コミュニティ・ポータル
ヘルプ
最近の更新
索引

特集
WordPress 3.1
WordPress 3.2

サーバ別情報
レンタルサーバ情報
自動アップグレード対応
マルチサイト機能対応

リファレンス
バージョン一覧
テンプレート階層
テーマのインストール
プラグインのインストール
WordPressのインストール
WordPressのアップグレード
WordPressのセキュリティ
WordPressのバックアップ
WordPressの移行
WordPressの移行
WordPressの移行

プラグイン API/フィルターフック一覧

このページ「プラグイン API/フィルターフック一覧」は 情報が古くなっている可能性があります。最新版 (英語) も合わせてご覧ください。最新に更新 してください。協力者を求めています。

この記事は (おそらくほぼ完全な) アクションフックの一覧です。アクションフックはバージョン2.1 以降から、プラグインおよびテーマ開発時に利用できるようになっています。詳しい情報については下記もご覧ください。

- フィルターフック、アクションフックについてもっと詳しく知るには、プラグイン API をご覧ください。
- プラグインの書き方については、プラグインの作成をご覧ください。
- アクションフックの一覧リファレンスは、プラグイン API/アクションフック一覧をご覧ください。
- 他のバージョンのプラグインフックについては、Plugin API/Hooks 2.0.x(en) をご覧ください。

注: もしこの文書に情報を追加したり、内容をより分かりやすく書き換えたい場合は、現在のスタイルに従って編集を行ってください。

目次 [非表示]

- 投稿、ページ、添付ファイル (アップロード) 関連のフィルタ
 - データベース読み出し
 - データベース書き込み
- コメント・トラックスバック・ピンバック関連のフィルタ
 - データベース読み出し
 - データベース書き込み
- カテゴリー関連のフィルタ
- タグ関連のフィルタ
 - データベース読み出し
 - データベース書き込み
- コメント・トラックバック・ピンバック関連のフィルタ
 - データベース読み出し
 - データベース書き込み
- 投稿、ページ、添付ファイル (アップロード) 関連のフィルタ
 - データベース読み出し
 - データベース書き込み

関数の設定

- ➔ 引数（フィルターによって異なる）で入ってくるなんらかのデータを加工（してもしなくてもよい）して、結果を返す関数を用意

```
function FUNCTION($content) {  
    $content = なんらかの処理;  
    return $content;  
}
```

フィルターに追加する

→ add_filter関数を使う

```
add_filter(  
    'the_content',  
    'wpjelly_plugin_filter_ikzo'  
);
```

これだけ。



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

サンプルデモ

➔ サンプルを構築してみよう

Hello world。

投稿日: [2011年10月12日](#)

WordPress へようこそ。これは最初の投稿です。編集もしくは削除してブログを始めてください！

カテゴリー: [未分類。](#) | [1件のフィードバック](#)

[編集](#)



Hello worldも電気もねえ！

投稿日: [2011年10月12日](#)

WordPress へようこそも電気もねえ！ これは最初の投稿ですもテレビもねえ！ 編集もしくは削除してブログを始めてください！

カテゴリー: [未分類。](#) | [1件のフィードバック](#)

[編集](#)

挿入その1に挿入

```
function wpjelly_plugin_filter_ikzo($content) {  
    //挿入その1  
    require_once(  
        dirname(__FILE__) .  
        '/mb_str_replace.function.php');  
    $content = mb_str_replace(  
        '。',  
        'もねえ！',  
        $content);  
    return $content;  
}
```

挿入その2へ挿入

```
add_filter(  
    'the_content',  
    'wpjelly_plugin_filter_ikzo'  
);
```

「。」を、
「テレビもねえ！ 」や、
「ラジオもねえ！ 」や、
「電気もねえ！ 」にする

手順

- ➔ wpjelly_plugin_filter_ikzoを変更
- ➔ wpjelly_plugin_filter_ikzo_cbを追加

関数内の2行をコメントアウト、下記コードを追加

```
function wpjelly_plugin_filter_ikzo($content) {  
    //require_once(  
    //    dirname(__FILE__) .  
    //    '/mb_str_replace.function.php');  
    //$content = mb_str_replace(  
    //    '。',  
    //    'もねえ！',  
    //    $content);  
    $content = preg_replace_callback(  
        '/(。)/',  
        'wpjelly_plugin_filter_ikzo_cb',  
        $content);  
    return $content;  
}
```

挿入その3へ挿入

```
function wpjelly_plugin_filter_ikzo_cb($matches) {  
    $ikzo = array('テレビ','ラジオ','電気');  
    $ikzo_key = array_rand($ikzo,1);  
    $result =  
        'も' . $ikzo[$ikzo_key] . 'もねえ！';  
    return $result;  
}
```

タイトルにフィルタ（挿入その2の後に挿入）

```
add_filter(  
    'the_title',  
    'wpjelly_plugin_filter_ikzo'  
);
```

カテゴリーにフィルタ（挿入その2の後に挿入）

```
add_filter(  
    'list_cats',  
    'wpjelly_plugin_filter_ikzo'  
);
```

管理画面にフィルタ（挿入その2の後に挿入）

```
add_filter(  
    'title_edit_pre',  
    'wpjelly_plugin_filter_ikzo'  
);
```



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

フィルター型プラグインでどんなプラグインが作れるか？

- ➔ テーブル単位でグループワーク
- ➔ 出来そうなアイデアを考えて、実際に作ってみる
- ➔ こういうのどうやるんだろ？は気軽に聞いてください

今日の宿題

➔ 勉強会の感想をブログに書く

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 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



WP JELLY

vol.3 フィルターフックを使ったプラグイン作成

JELLY
JELLY
CAFE
TOKYO

X



X

Greative