



WP JELLY

vol.1 オリジナルのプラグインをつくらう!!



X



X





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

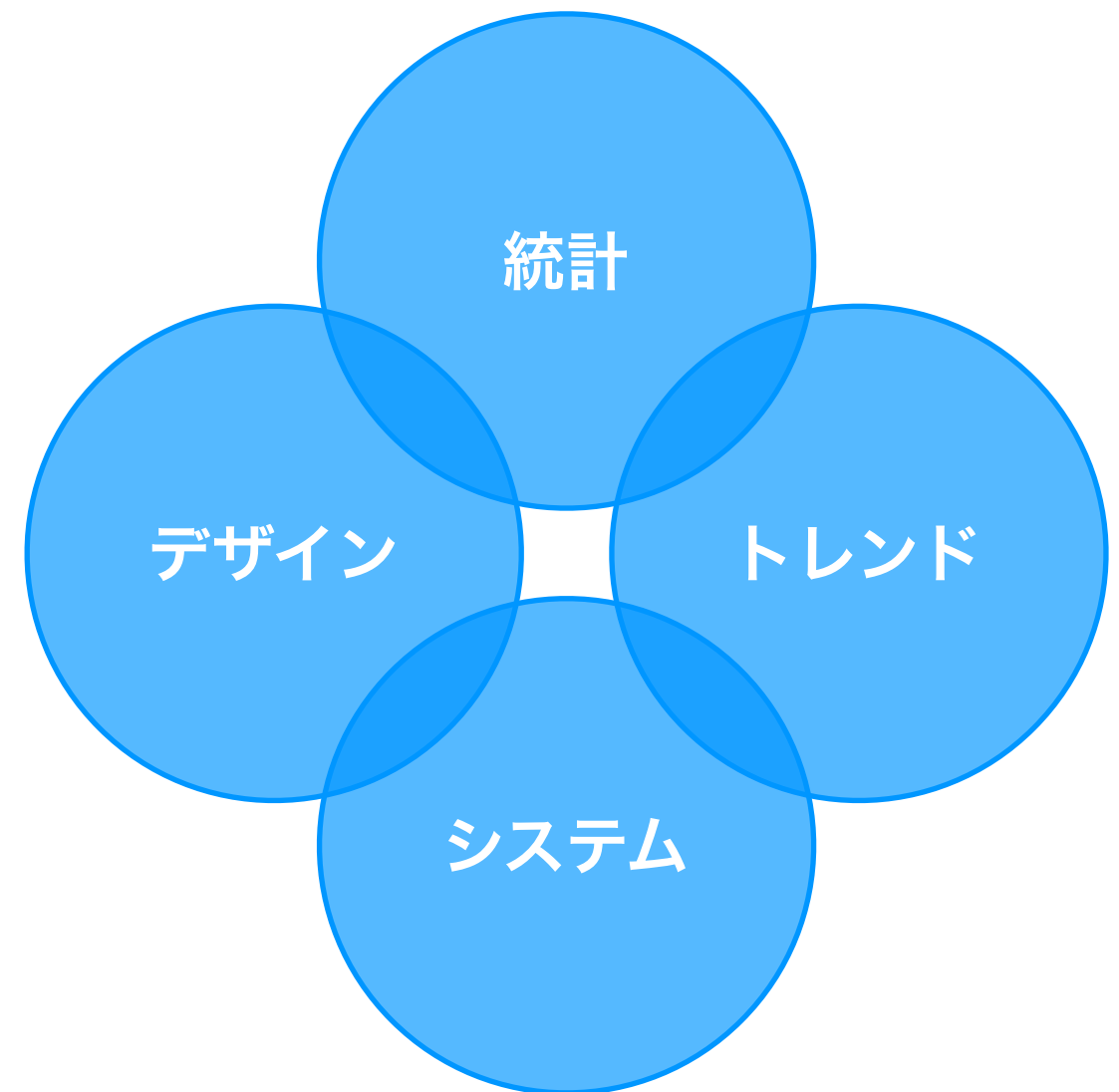
活動

- <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



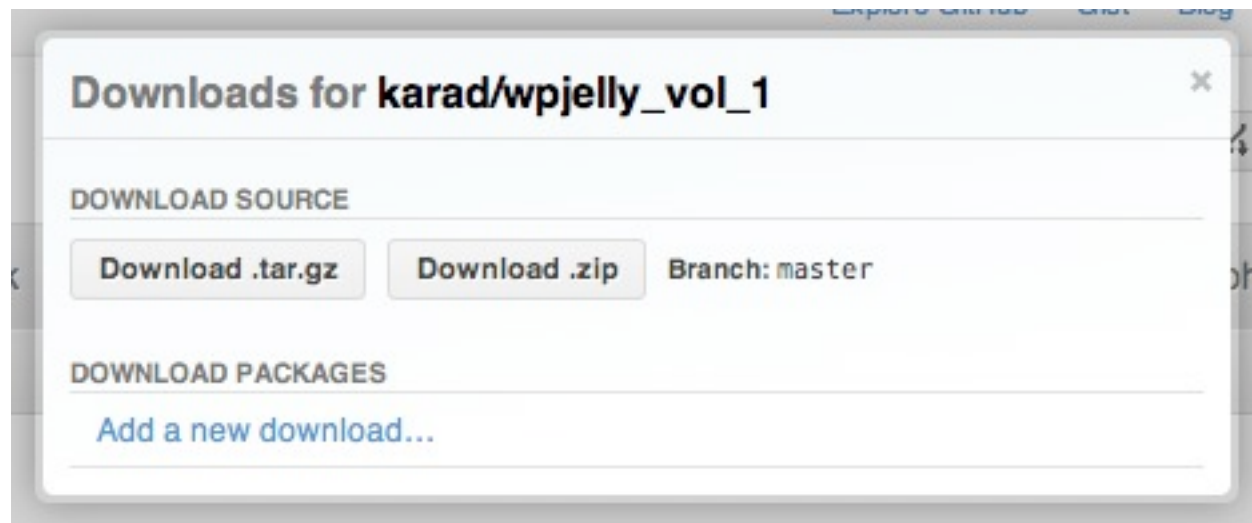


➔ 黒野 明子 (著, 編集), 秋野 琢 (著), 小山 智久 (著), 原 一浩 (著), 柳谷 真志 (著)

下記のURLにアクセスして、[Downloads]ボタンをクリック

https://github.com/karad/wpjelly_vol_1

すると、こんなダイアログが出るので、[Download .zip]をクリック



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

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

- ➔ WordPressの仕組み
- ➔ WordPressのプラグインの仕組み

Session

- ➔ 本日作成するプラグイン「更新用マニュアル」

Develop

- ➔ アイコンを変えてみよう
- ➔ メニューの表示場所を変えてみよう
- ➔ CSSを変えてみよう
- ➔ 応用して独自のプラグインを考えてみよう

Hacking

なかよく
たのしく
りすべくと

→ @ittoi_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
- ・ サンプルアプリ：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
- ・ サンプルアプリ：[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

- ・ サンプルアプリ：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
- ・ サンプルアプリ：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
- ・ サンプルアプリ：未定



Session

WordPressのプラグインについて知ろう!!

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

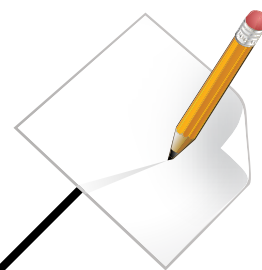
参考リンク



- ➔ WordPressの実行フローを視覚化してみる | Simple Colors
 - <http://www.warna.info/archives/279/>
- ➔ WordPress Internals: How WordPress Boots Up
 - <http://theme.fm/2011/09/wordpress-internals-how-wordpress-boots-up-2315/>
- ➔ WordPress Internals: How WordPress Boots Up Part 2
 - <http://theme.fm/2011/09/wordpress-internals-how-wordpress-boots-up-part-2-2437/>

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!!



→ Demo

The screenshot shows the WP Jelly WordPress dashboard. The sidebar on the left contains navigation links: ダッシュボード, 投稿 (Posts), メディア (Media), リンク (Links), 固定ページ (Static Pages), コメント (Comments), 外観 (Appearance), and プラグイン (Plugins). The main content area displays the 'システム操作マニュアル' (System Operation Manual) page. The page has a header with 'WP Jelly' and a sidebar with '現在の状況' (Current Status) showing 1 posts, 1 static page, 1 category, and 0 tags. The main content area is titled 'システム操作マニュアル' and includes sections for '凡例' (Notes), '目次' (Table of Contents), and '設定' (Settings). The '設定' section is titled 'Cimy Swift SMTP 管理者のみ' and provides instructions for configuring email settings. A red box highlights the 'SMTPサーバーのURL' (SMTP server URL) field in the configuration form.

プラグインは下記決まりを守って書きましょう

Now!

- ➔ 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
-
- ➔ 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 Framework PHP 標準コーディング規約
 - ・ 後の回でやります
 - ➔ セキュリティに気をつけましょう
 - ・ これも後の回とかでやります
 - 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

プラグインの名前は既存にないものが望ましい

特に、プログラム内でつかう名前は、独特なものにする

➔ 名前空間（ネームスペース）という考え方

- xmlns
- CSS3 名前空間モジュール
- とか。

➔ 関数や外に出ている変数は特に気をつける

➔ 作る人のドメインを逆からつける手法がよく使われる

- greative.jpなら、jp.greative.hogehogeとか。関数名、変数名でつけるならjp_greative_admin_hoge()とかみたいにつけると、まずかぶらない。

➔ もしくは確実なユニークなプラグイン名を使う



Develop

サンプルを動かしてみよう!!

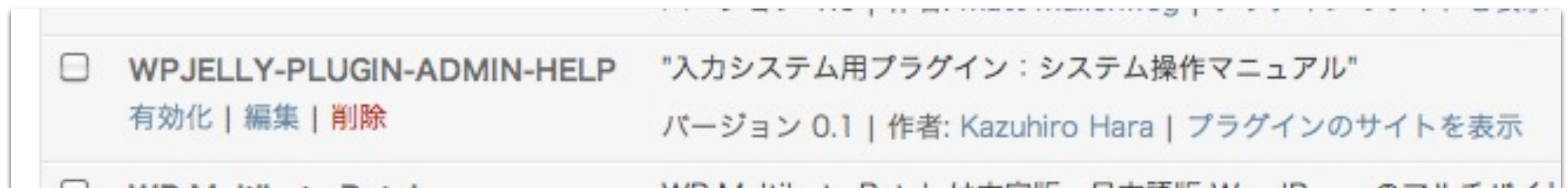
面倒でも最初は入力(写経)してみよう

ダウンロード後

- ➔ ダウンロードしたサンプル内の[wpjelly_admin_help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

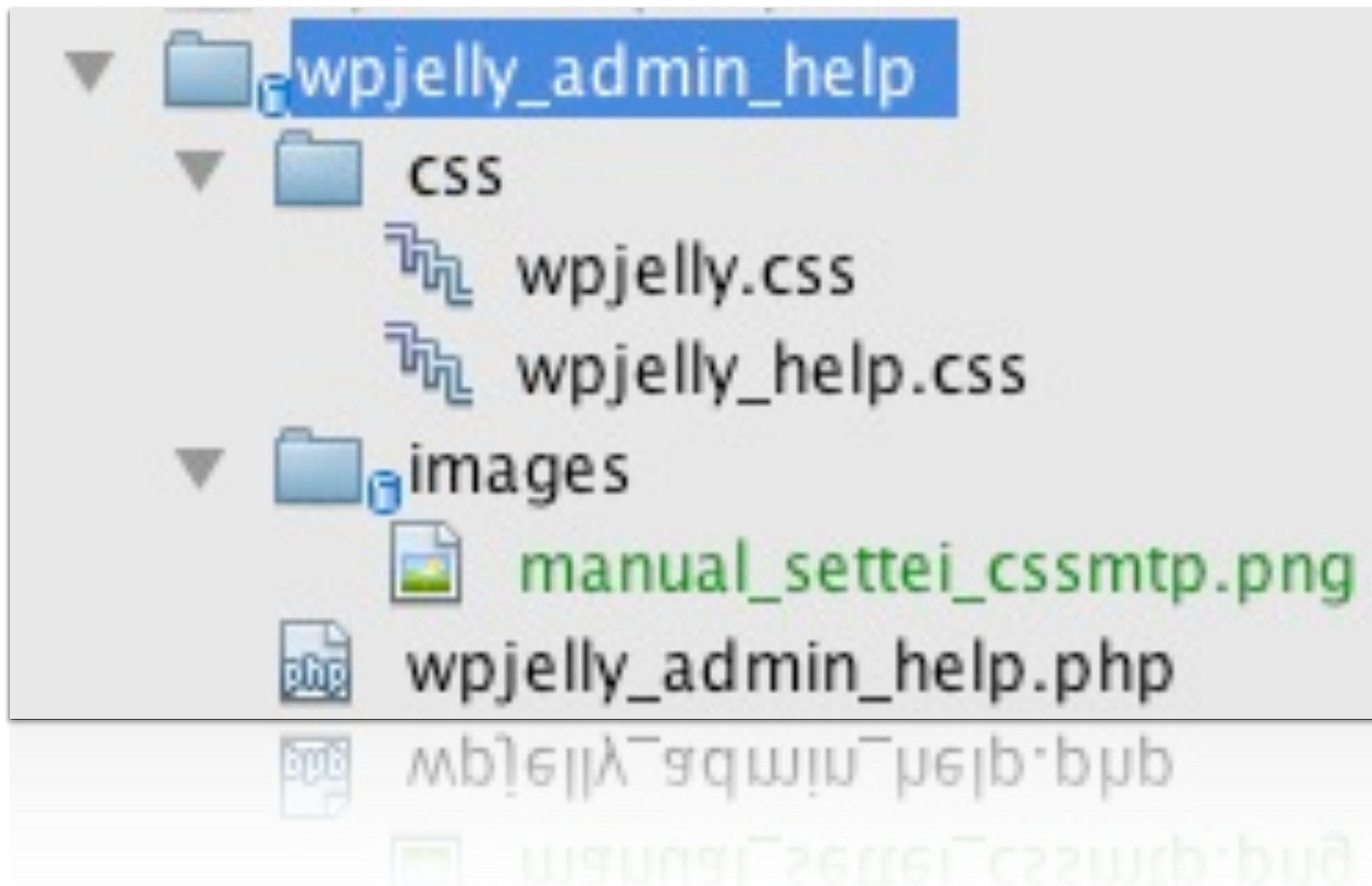


- ➔ WPJELLY-PLUGIN-ADMIN-HELPを[有効化]



ソース解説

- wpjelly_admin_help.php … 本体
- cssフォルダ … 後で使用
- imagesフォルダ … マニュアルで使用している画像



こんな風を書く事

```
/*  
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

ポイントは、画像のパス登録

この関数はコーディングが面倒なので、あらかじめ書いてあります

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

プラグインにメニューとリンク先、タイトルなどを登録する関数を実行

先ほどのリンク先コンテンツを登録している点に注目

```
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');  
}
```

アクション登録はこんな具合に登録する関数を記述

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


Roles and Capabilities « WordPress Codex



→ http://codex.wordpress.org/Roles_and_Capabilities

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_posts						



Hacking

サンプル・ハッキング!!
触って、変えて、楽しむ!

[icon-edit]の[edit]部分を書き換えると、アイコンが変わるよ

→ 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まずは記述しよう

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

wpjelly_plugin_admin_help_add_menu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');  
}
```

[CSSを追加してみよう:1]で入力した関数をコピーして、変更しよう

```
function wpjelly_add_css_help() {  
    $wpjelly_admin_help_images =  
        WP_PLUGIN_URL . '/wpjelly_admin_help/css';  
    wp_register_style('wpjelly_admin_help_css_help',  
        $wpjelly_admin_help_images . '/wpjelly_help.css');  
    wp_enqueue_style('wpjelly_admin_help_css_help');  
}
```

wpjelly_plugin_admin_help_add_menu関数にもう1行追記する

```
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');  
    add_action('admin_print_styles' . '-' . $hookname,  
        'wpjelly_add_css_help');  
}
```

[edit.php]を他のメインメニューのリンク先に変更してみよう

```
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');  
}
```

たとえば、こんな拡張

担当者などの連絡先一覧

jQueryを組み込んでマニュアルを見やすく

TwitterやFacebookを一覧

Backlogのタスクを表示してみる

Googleカレンダーで共有予定を表示

Google Mapで営業所一覧表示

よく使うタグなどを一覧

etc...

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](#)



WordPress JELLY

vol.1 オリジナルのデザインをつくろう!!

JELLY
JELLY
CAFE
TOKYO

X



X

