

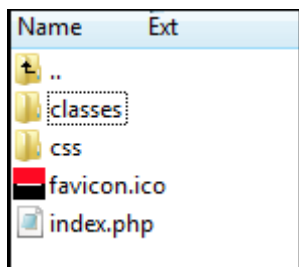
Créer un plugin WordPress avec POO

21 mai 2013 [0 Comments](#)



Nous allons voir ce jour la création d'un plugin sous WordPress, mais on va utiliser le principe de la programmation orientée objet afin d'avoir une meilleure maintenance et gestion de code.

Tout d'abord, je vous donne l'architecture du plugin. Dans le dossier plugins vous pouvez créer un nom de votre plugins et à l'intérieur, vous aurez quelque chose comme suit :



Comme vous pouvez le comprendre vite fait, le dossier classes va contenir notre classe, le dossier css pour les feuilles de style. le fichier index.php, celui qui va contenir l'instanciation de notre classe

```
<?phphttp://localhost/wordpress-3.5-fr_FR/wordpress/wp-admin/users.php
/*
Plugin Name: Chapter 2 00
Description: OOP
Version: 1.0
Author: Amine BETARI
*/

define('PLUGIN_PATH', dirname(__FILE__));
define('PLUGIN_PATH_CSS', plugins_url('css/stylesheet.css', __FILE__));
define('FAVICON', plugins_url('favicon.ico', __FILE__));
require 'PLUGIN_PATH.'/classes/ch2-oo.php';

$exempleOOP = new CH2_00();
```

Ensuite dans le fichier ch2-oo.php, nous avons notre classe, qu'on trouve dans son constructeur '___construct', les « hooks » de WordPress.

Bien sûr, il ne s'agit qu'un simple plugin. Mais pour toute personne qui souhaite se consacrer au développement de plugin sous WordPress, je ne peux que vous encouragez à coder d'une façon orientée objet vos plugins.

Le code source du fichier ch2-oo.php

```

<?php
class CH2_00
{
function __construct()
{
    add_action( 'wp_enqueue_scripts', array($this, 'ch2pitQueueStylesheet') );
    add_action( 'admin_menu', array($this, 'SettingsMenuAdmin') );
}

function ch2pitQueueStylesheet()
{
    wp_enqueue_style( 'privateshortcodestyle', PLUGIN_PATH_CSS);
}

function SettingsMenuAdmin()
{
    add_menu_page('AxaMAROC', 'PageMenu', 'manage_options', 'slug-admin-menu', array($this,
    add_submenu_page('slug-admin-menu', 'AxaOujda', 'SousPage', 'manage_options', 'slug-admin-menu', array($this, 'pageConfigSubMenu') );
}

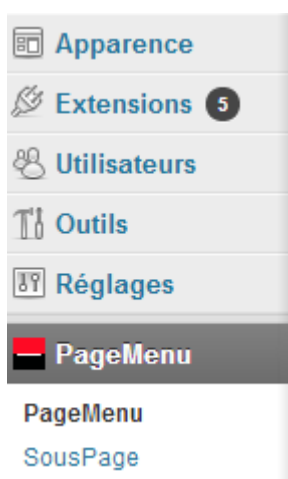
function pageConfigSubMenu()
{
    echo esc_html('<h2>PAGE_SUB_MENU_ADMIN</h2>');
}

function pageConfig()
{
    echo esc_html('<h2>PAGE MENU ADMIN</h2>'); }}

```

Pour le fichier index.php

Dans cet exemple, on ajoute un menu et un sous_menu dans le back_office comme vous pouvez le voir:



Nous avons vu comment peut-on créer un plugin sous WordPress avec la technique orientée objet.