

Créer une extension (plugin)

En premier lieu nous allons nous rendre dans le dossier "**wordpress/wp-content/plugin/**" et nous allons créer un premier dossier qu'on appellera "**helloworld**" et par convention on fera un fichier php de même nom.

Celui ci contiendra en premier lieu les commentaires suivants :

```
/*
    Plugin Name: Hello World !
    Plugin URI: https://www.google.fr
    Description: Ceci est un plugin affichage des hello world.
    Author: Nolwenn
    Author URI: https://www.marquiset.fr
    Version: 0.1.0
*/
```

Si on va dans nos extensions, on trouvera notre nouvelle extension. Activons là même si elle ne fait rien pour l'instant.

Le principe des principes des extensions Wordpress est de relier des fonctions à des crochets (hook). Ces crochets sont juste des accroches définies par Wordpress qui se déclenchent à différents moments de la vie du site.

Tous ces crochets sont documenter dans la documentation de wordpress. Lorsque l'on veut créer une extension, il nous faut donc trouver les crochets qui conviennent aux actions que l'on doit réaliser, par exemple :

```
add_action("wp_footer","hello");

function hello()
{
    echo "<p>Hello World !</p>";
}
```

Ici nous avons créé une nouvelle fonction nommé "**hello**" et je lui ai dit avec "**add_action**" de se déclencher lorsque wordpress a fini de créer le footer grâce à "**wp_footer**".

Si on regarde, en dessous de notre footer, notre paragraphe apparaît.

Ceci est un crochet de type "**action**" mais il existe aussi des crochets de type "**filtre**". Les principales différences sont :

- L'action :
 - Se déclenche lors de certaines actions du site.
 - Ne retourne aucune valeur.

- Le filtre :
 - Se déclenche lors de manipulation de donnée.
 - Retourne une valeur.

Essayons donc ceci :

```
add_filter("default_content", "world");

function world()
{
    return "<p>Hello World !</p>";
}
```

Le crochet "default_content" va se déclencher lors de la création de page ou d'article pour créer un contenu par défaut. Maintenant, si on tente de créer un article, on verra notre "Hello World !".

Les filtres peuvent aussi recevoir des données que l'on traitera avant de les retourner, par exemple :

```
add_filter("the_content", "goodbye");

function goodbye($content)
{
    return $content . "<hr><p>Goodbye World !</p>";
}
```

On récupère le contenu de nos articles et nos pages, et on leur concatène un petit message d'au revoir.

Mais je trouve que ce message d'au revoir n'a rien à faire sur mes différentes pages, et que l'on peut s'en passer sur les prévisualisations des articles. Pour cela Wordpress intègre des fonctions comme "**is_single**".

Remplaçons le contenu de notre précédente fonction par :

```
if(is_single())
    return $content . "<hr><p>Goodbye World !</p>";
return $content;
```

is_single vérifie que le crochet est pour un post seul, si ils sont plusieurs ou si c'est une page, il répondra "**false**".

Wordpress gère aussi un système de **shortcode**, permettant d'insérer du code là où on en a besoin :

```
add_shortcode("hw", "alertworld");

function alertworld()
{
```

```
    return "<script>alert('Hello World !');</script>";  
}
```

Maintenant, si je tape dans un article, une page ou même le thème de mon site "**[hw]**", une alerte bien casse pied apparaîtra.