

Pourquoi et comment créer un plugin WordPress ?

10/04/18 14 mn de lecture

Ce n'est pas faute de le répéter, les plugins sont l'essence même de WordPress. Il en existe des tonnes, qu'ils soient **premiums**, freemiums ou **gratuits**, si vous utilisez WordPress, vous savez de quoi je parle. Selon la finalité de votre utilisation de ce fabuleux CMS, vous allez un jour vous trouver face à une question : *comment pourrais-je créer ma première extension ?*

Dans cet article, nous allons voir qu'il est plutôt facile de **créer un plugin WordPress** et surtout nous allons voir quels bénéfices vous pourriez en tirer...



Qu'est-ce qu'un plugin ou extension ?

Si vous êtes tout **nouveau sur WordPress**, vous pourriez éventuellement vous poser la question “*mais c'est quoi un plugin ?*”. Vous allez très vite trouver réponse à cette question lorsque vous allez parcourir l'administration de votre site web fraîchement installé.

Sur la gauche de votre écran d'administration se trouve une barre latérale sombre affichant les divers onglets de customisations et paramétrages. Sous l'onglet “**Apparence**” se trouve l'onglet “**Extensions**” avec l'icône d'une prise de courant.

C'est bel et bien ici que vous pourrez **gérer vos plugins** existants et à venir.

Pour répondre donc à la question, les extensions – *ou plugins en anglais* – sont des **fonctionnalités que l'on ajoute à un site WordPress** pour parvenir à l'accomplissement du projet web.

Lorsque j'ai installé WordPress pour la première fois, je me souviens encore avoir posé la question à mon formateur : “*mais pourquoi faut-il ajouter des fonctionnalités, pourquoi ne sont-elles pas embarquées de manière native ?*”. J'ai vite compris que cette question était totalement absurde et heureusement que ce n'est pas le cas ! Étant donné que l'on peut créer **tous les genres de site web avec WordPress**, chaque installation sera différente en fonction des besoins du projet.

Ainsi, grâce aux plugins, vous allez pouvoir façonner votre site ou votre blog WordPress.

Pour ajouter une nouvelle extension, rendez-vous dans l'onglet **Extension > Ajouter** ou bien vous pouvez également le faire en téléchargeant le dossier depuis le **répertoire officiel** et en l'envoyant sur votre serveur via FTP.

Pourquoi créer un plugin WordPress ?

C'est vrai, après tout : pourquoi s'embêter à **créer un plugin WordPress** alors qu'il en existe des tas, certainement bien plus efficaces que celui que vous envisagez de créer ? Je vais vous expliquer et vous donner quelques bonnes raisons de le faire...

Raison n°1 : alléger le *functions.php* du thème enfant

Généralement, la bonne pratique est de **créer un thème enfant** lorsque l'on veut ajouter un peu de code pour modifier certaines actions de WordPress ou du thème et ajouter de nouvelles fonctionnalités (ou en supprimer).

Toutefois, si on travaille en pensant au futur, il faut avoir une certaine rigueur et penser que peut-être, un jour, on aura besoin de revenir sur le code inséré dans le fichier **functions.php**. Si c'est le cas, on aura commenté chaque fonction afin de les retrouver facilement.

Lorsque de nombreuses fonctions se suivent, le fichier **functions.php** peut devenir difficile à maintenir, c'est pourquoi il pourrait être utile de **déplacer certaines fonctions dans un plugin**.

Raison n°2 : faire suivre le code

Pour la même raison évoquée avant, si on pense à l'avenir, il se pourrait qu'un jour, vous ou votre client ait une envie soudaine de **changer de thème**.

Que va-t-il se passer alors ? Le code inscrit dans le fichier **functions.php** du thème enfant disparaîtra comme neige au soleil ! Adieu ces heures passées à tester et à coder, à moins de le faire suivre.

Voici alors une bonne raison de **créer un plugin WordPress**, lorsque vous changerez de thème, votre fonctionnalité sera toujours présente dans votre site.

Raison n°3 : réutiliser le code

Si vous **maintenez plusieurs sites WordPress** ou que vous en créez pour vos clients, vous allez vous rendre compte qu'il y a souvent une ou deux fonctionnalités dont vous avez besoin systématiquement.

Dans ce cas là, rien de tel qu'un plugin pour **réutiliser le code à bon escient à chaque nouveau projet**.

Raison n°4 : prévenir les conflits et incompatibilités

Surtout, l'un des avantages de **créer un plugin WordPress** plutôt que **d'ajouter le code directement dans le thème enfant** est que vous allez pouvoir prévenir rapidement les conflits.

Admettons que votre plugin fonctionne parfaitement, puis que vous **mettez à jour WordPress, votre thème** ou vos autres extensions et que vous vous rendez compte qu'il y a un conflit...

Dans ce cas, il sera facile, pour vous, de **vérifier si le conflit** provient de votre **extension faite-maison** : il vous suffira de la désactiver pour en avoir le cœur net.

Cela vous évitera, par la même occasion, de vérifier chacune des fonctions ajoutées dans le thème enfant. Bref, c'est très pratique et surtout, c'est un gain de temps.

Comment créer un plugin WordPress simple ?

On va bien, ici, **distinguer le fait de créer un plugin simple d'un plugin complexe**. Si dans le premier cas, vous n'avez pas besoin d'être un développeur confirmé – *vous allez même trouver cela facile* – dans le second cas il vous faudra un certain bagage et de bonnes connaissances en PHP et en la manière dont fonctionne WordPress.

1 – Les pré-requis pour créer une extension WordPress

Vous n'avez pas besoin de grands choses pour **créer un plugin WordPress**, voici les pré-requis :

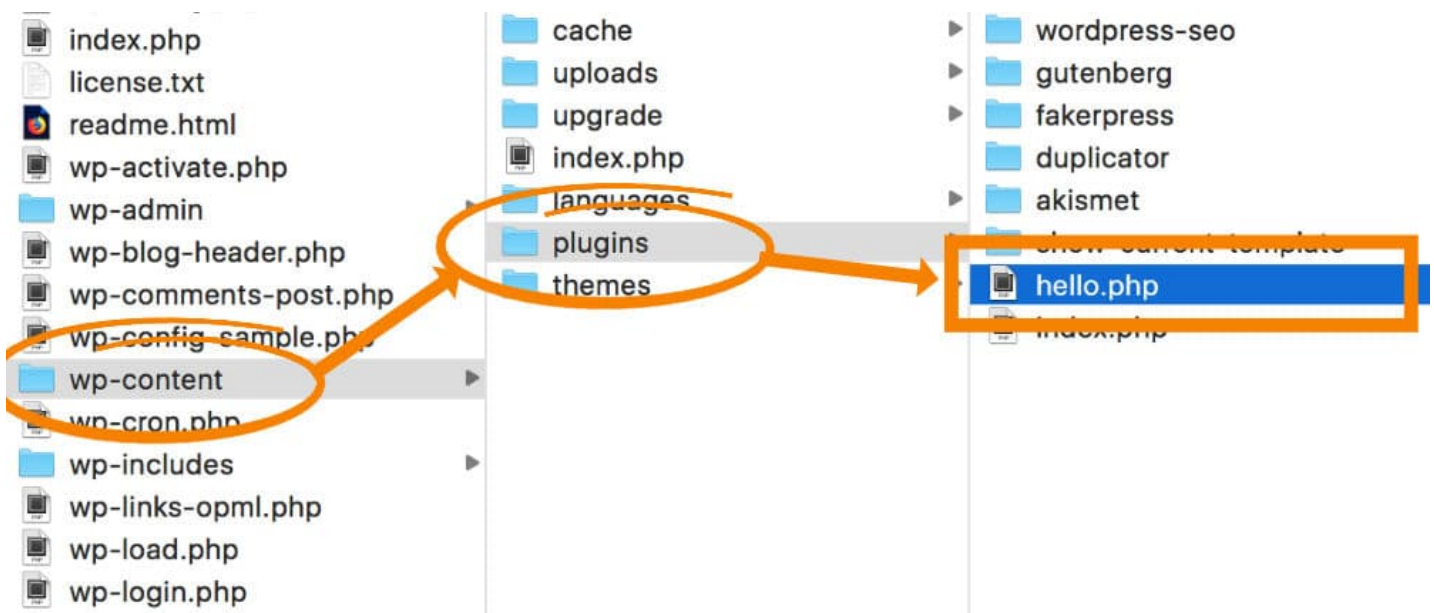
- avoir un environnement de test comme **Local by FlyWheel**, Mamp, Xamp etc.
- avoir une installation WordPress à jour : si possible identique à la version de production pour voir si votre plugin n'émet pas des conflits avec les autres extensions ou thèmes
- être en possession d'un éditeur de texte comme **Brachet** ou **SublimText**

2 – Étude rapide du plugin Hello Dolly

Si vous avez envie de **créer votre premier plugin**, vous pouvez déjà commencer par étudier le fameux plugin *“Hello Dolly”* qui est embarqué avec toute installation WordPress.

En effet, ce plugin n'est pas là par hasard – *même si on a tendance à le supprimer automatiquement après l'installation de WordPress* – car il est l'exemple même qui nous montre le chemin pour **réaliser facilement sa propre extension**.

La première des chose qui saute aux yeux, lorsqu'on regarde le dossier **“PLUGINS”** situé dans le répertoire **“WP-CONTENT”**, c'est que **Hello Dolly** n'est pas placé dans un dossier, contrairement aux autres extensions.



Cela nous montre bien qu'un simple plugin peut être composé d'un seul **fichier.php** et n'a pas besoin d'autre chose comme une feuille de style.css ou des fichiers JavaScript pour fonctionner.

WordPress permet donc d'ajouter un simple fichier contenant une fonctionnalité directement dans le dossier **WP-CONTENT > PLUGINS**.

Mais attention, ce "simple fichier" doit suivre les guidelines de WordPress afin d'être exploitable, ouvrons le fichier **hello.php** pour comprendre et voir ce qu'il contient...

```

1  <?php
2  /**
3   * @package Hello_Dolly
4   * @version 1.6
5   */
6
7  /*
8   Plugin Name: Hello Dolly
9   Plugin URI: http://wordpress.org/plugins/hello-dolly/
10  Description: This is not just a plugin, it symbolizes the hope and enthusiasm of an entire generation summed up in two words
11  Author: Matt Mullenweg
12  Version: 1.6
13  Author URI: http://ma.tt/
14  */
15
16  function hello_dolly_get_lyric() {
17      /** These are the lyrics to Hello Dolly
18       $lyrics = "Hello, Dolly
19       Well, hello, Dolly
20       It's so nice to have you back where you belong
21       You're lookin' swell, Dolly
22       I can tell, Dolly
23       You're still glowin', you're still crowin'
24       You're still goin' strong
25       We feel the room swayin'

```

Comme cette capture d'écran ci-dessus nous le montre, le plugin **Hello Dolly**, comme tous les autres plugins, est composé de :

1. Un fichier PHP : **nom.php**, dans ce cas il s'agit de **hello.php**
2. Une ouverture de balise PHP (**<?php**). Pas besoin d'ajouter une balise de fermeture car ce n'est pas obligatoire, contrairement au HTML.
3. Un entête placé entre **/*** et ***/** qui paraît être un commentaire anodin mais vous allez voir que ce n'est pas le cas.
4. Le code PHP qui permet d'ajouter une fonctionnalité à votre site.

C'est tout ! Et vous voyez que ce plugin est fonctionnel. Maintenant, penchons nous sur cette fameuse entête qui est un élément obligatoire et primordial...



Si on regarde attentivement la capture d'écran ci-dessus, vous allez voir à quoi correspondent chaque ligne de l'entête d'un plugin.

Les **pastilles rouges** numérotent les lignes présentes dans le **fichier.php** et les **pastilles bleues** sont leur correspondance dans le back-office.

1. **la ligne *Plugin Name***: => elle correspond au nom du plugin tel qu'il sera affiché dans la partie administrateur. On voit bien que celui-ci n'est pas obligatoirement le même que le nom du fichier, qui lui est nommé **hello.php**.
2. **la ligne *Plugin URI***: => elle correspond à l'URL du plugin sur le repository (répertoire officiel), si ce n'est pas le cas, vous pouvez insérer ici l'URL de votre choix.
3. **la ligne *Description***: => elle permet de décrire la fonctionnalité du plugin.
4. **la ligne *Author***: => elle donne l'identité du créateur du plugin. C'est l'ancre qui apparaîtra sur le lien du site de l'auteur (pastille 6).
5. **la ligne *Version***: => elle permet d'afficher le numéro de version du plugin. À chaque nouvelle version, ce numéro évoluera.
6. **la ligne *Author URI***: => elle fait le lien avec le site de l'extension.

À noter : si vous regardez bien, il n'y a pas d'espace avant les "deux points", veuillez à respecter cela.

3 – Passons à l'action

À présent que nous en savons un peu plus sur la structure d'un plugin, voici la méthode pour créer le vôtre :

- dans un éditeur de texte, créez un nouveau fichier que vous nommerez **nom-du-plugin.php**
- copiez et collez l'entête en début de fichier, comme suit :