



N'oubliez pas d'activer votre plugin, ou votre thème, pour tester les exemples de cet article !

Les « hooks » sont des crochets, des points d'ancrage, placés par les développeurs dans WordPress. Les hooks vous permettent ainsi d'agir sur des points précis des fichiers de base sans avoir à les modifier.

Il existe deux types de hooks : les **actions** et les **filtres**.

## Les actions

Référence: <https://developer.wordpress.org/plugins/hooks/actions/>

Les actions permettent d'appeler une de vos fonctions au même moment que l'exécution d'une fonctionnalité WordPress.

Pour créer une nouvelle action il faut utiliser la fonction `add_action()` avec au minimum ces deux paramètres :

1. `$tag` le nom de l'action que l'on veut « crocher »
2. `$function_to_add` le nom de la fonction de rappel (callback) qui sera appelée lors de cette action

### Action sur le crochet `wp-head`

Le code ci-après présente la création d'une action qui ajoutera une balise `<meta />` dans l'entête du site en utilisant le crochet `wp-head`

```
//Fonction qui affiche la balise meta
function mon_plugin_meta_keywords() {
    echo '<meta name="keywords" content="HTML,CSS,XML,JavaScript">';
}
```



```
//Ajout d'une action sur 'wp_head' qui appellera mon_plugin_meta_keywords()  
add_action('wp_head', 'mon_plugin_meta_keywords' );
```

Il est recommandé de précéder le nom des fonctions de vos plugins avec le nom du plugin ou le nom du développeur.

Cette pratique limite le risque que vous créiez une fonction avec un nom déjà utilisé par WordPress ou un autre plugin, ce qui provoquerait le crash de WordPress.

C'est pourquoi le nom de la fonction ci-dessus commence par `mon_plugin_`

## Action sur la suppression d'un article

Autre exemple, une action qui envoie un email lors de la suppression d'un article.

```
//Fonction qui envoie par email les infos d'un email supprimé  
function mon_plugin_post_delete_mail($post_id) {  
    //Récupère les informations de l'article supprimé  
    $post = get_post($post_id);  
    //Création du sujet de l'email  
    $sujet = "Artile supprimé :" . $post->post_title;  
    //Création du contenu de l'email  
    $message = "Contenu de l'artilce : " . $post->post_content;  
    //Envoi de l'email à l'administrateur du site  
    wp_mail(get_bloginfo('admin_email'), $sujet, $message);  
}  
//Ajout d'une action sur 'delete_post' qui appellera
```



```
mon_plugin_post_delete_mail()  
add_action('delete_post', 'mon_plugin_post_delete_mail');
```

## Paramètres d'actions

Certaines actions permettent de récupérer un ou plusieurs paramètres. Dans l'exemple précédent, `$post_id` est passé en paramètre pour récupérer l'identifiant de l'article supprimé.

Vous trouverez la liste des différents paramètres disponibles pour chaque actions sur le codex de WordPress: [Action Reference](#)

## Les filtres

Référence: <https://developer.wordpress.org/plugins/hooks/filters/>

Les filtres permettent de changer la valeur d'un contenu durant l'exécution de WordPress.

On peut donc ajouter, supprimer ou modifier une partie du contenu généré par WordPress juste avant l'affichage sans modifier sa valeur d'origine.

Pour utiliser un filtre on utilise la fonction `add_filter()`. Cette fonction prend deux paramètres comme `add_action()` :

1. `$tag` le nom du filtre que l'on veut « crocher »
2. `$function_to_add` le nom de la fonction de rappel (callback) qui sera appelée pour filtrer

## Filtre sur le titre d'un article

Le code ci-après présente la création d'un filtre qui sera appelé sur le titre d'un article (ou d'une page) et remplacera la chaîne « et » par une esperluette « & ».



```
//Fonction qui remplace la chaine 'et' par '&amp;'  
function mon_plugin_the_title( $title ) {  
    //Remplace 'et' dans le titre  
    $title = str_replace( 'et', '&amp;', $title );  
    //Retourne le titre modifié  
    return $title;  
}  
//Ajout d'un filtre sur 'the_title' qui appellera mon_plugin_the_title()  
add_filter( 'the_title', 'mon_plugin_the_title' );
```

## Paramètres des filtres

Chaque filtre permet de travailler sur un ou plusieurs paramètres. Dans l'exemple précédent, `$title` est passé en paramètre pour récupérer le contenu du titre.

Vous trouverez la liste des différents paramètres disponibles pour chaque filtre sur le codex de WordPress: [Filter\\_Reference](#)

*Envie de créer des contenus dynamiques faciles à placer dans vos articles et pages ?*

Alors apprenez comment [créer un shortcode WordPress](#)