



- □□□□



1. Hooks

WordPress uses a hook system to allow developers to modify the default behavior of the platform without changing the core files.

- Action Hooks:** Used to execute a function at a specific point in the WordPress lifecycle.
- Filter Hooks:** Used to modify data before it is used or displayed.

2. Action Hooks

WordPress provides several action hooks throughout its execution. Some common ones include:

WordPress Hooks

1. Adding Hooks

- `add_action()` is used to attach a function to an action hook.

```
function my_custom_function() {  
    // Custom code  
    echo "This is my custom function!";  
}  
  
add_action('wp_footer', 'my_custom_function');
```

2. Removing Hooks

- `remove_action()` is used to detach a function from an action hook.

```
remove_action('wp_footer', 'my_custom_function');
```

WordPress Hooks

- init:** Fired after WordPress has finished loading the database and before the `wp_head` hook.

```
add_action('init', 'my_init_function');  
  
function my_init_function() {  
    // Custom code  
}
```

- wp_enqueue_scripts:** Used to enqueue scripts and styles for the front-end of the site.

```
add_action('wp_enqueue_scripts', 'my_enqueue_scripts');

function my_enqueue_scripts() {
    wp_enqueue_style('my-style', get_stylesheet_uri());
    wp_enqueue_script('my-script', get_template_directory_uri() . '/js/my-script.js', array(), false, true);
}
```

- **save_post**: 文章保存時に実行されるフック。

```
add_action('save_post', 'my_save_post_function');

function my_save_post_function($post_id) {
    // 保存時に実行される処理
}
```

3. フィルター(Filter)

WordPressでは、デフォルトの動作をカスタマイズするために、フィルター(Filter)という仕組みが提供されています。

基本的な使い方

1. フィルターを追加

- **add_filter()**: 特定のフィルターにカスタム関数を追加する。

```
function my_custom_title($title) {
    return 'Prefix - ' . $title;
}

add_filter('the_title', 'my_custom_title');
```

2. フィルターを削除

- **remove_filter()**: 特定のフィルターからカスタム関数を削除する。

```
remove_filter('the_title', 'my_custom_title');
```

具体的な例

- **the_content**: 表示されるコンテンツの文字列をカスタマイズする。

```
add_filter('the_content', 'my_custom_content');

function my_custom_content($content) {
    return $content . '<p>Custom content added at the end.</p>';
}
```

- **the_title**: 表示されるタイトルの文字列をカスタマイズする。

```
add_filter('the_title', 'my_custom_title');

function my_custom_title($title) {
    return 'Prefix - ' . $title;
}
```

- **wp_nav_menu_items**: 在 WordPress 中，wp_nav_menu_items 是一个钩子，用于在调用 wp_nav_menu() 函数时，对生成的菜单项进行自定义修改。它允许你在菜单项的 HTML 输出之前，对每个菜单项进行自定义处理，比如添加自定义的类名、修改链接文本、或者添加自定义的 HTML 内容等。这个钩子是在 wp_nav_menu_items 这个钩子上调用的，它接收两个参数：\$items 和 \$args。\$items 是一个数组，包含了所有要显示的菜单项的 HTML 字符串。\$args 是一个数组，包含了与菜单项相关的其他信息，比如当前项是否是选中的项、是否是父项等。通过在这个钩子上添加自定义函数，你可以对菜单项的输出进行自定义控制。

```
add_filter('wp_nav_menu_items', 'my_custom_menu_items', 10, 2);

function my_custom_menu_items($items, $args) {
    return $items . '<li><a href="#">Custom Link</a></li>';
}
```



在 WordPress 中，add_action() 和 add_filter() 是用于添加自定义函数到 WordPress 核心钩子的函数。remove_action() 和 remove_filter() 则是用于移除已经添加的自定义函数。add_action() 用于添加自定义函数到 WordPress 核心钩子上，而 add_filter() 用于添加自定义函数到 WordPress 核心钩子上，用于过滤数据。remove_action() 用于移除已经添加的自定义函数，而 remove_filter() 用于移除已经添加的自定义函数，用于过滤数据。

Q1: 在 WordPress 中，add_action() 和 add_filter() 有什么区别？

Q2: 在 WordPress 中，remove_action() 和 remove_filter() 有什么区别？

Q3: 在 WordPress 中，add_action() 和 add_filter() 的返回值是什么？

You wanna more detailed information?