

Unit - 5

ADVANCED DEVELOPMENT

Advanced functions

add_action()

Actions are the hooks that the WordPress core launches at specific points during execution, or when specific events occur.

Parameters

\$tag (string) (Required) The name of the action to which the \$function_to_add is hooked.

\$function_to_add (callable) (Required) The name of the function you wish to be called.

\$priority (int) (Optional) Used to specify the order in which the functions associated with a particular action are executed.

\$accepted_args (int) (Optional) The number of arguments the function accepts. Default value: 1

Example :

```
<?php
function save_my_post() {
echo '<div>Post Content</div>';
}
add_action( 'save_post', 'save_my_post'); ?>
```

add_filter()

WordPress offers filter hooks to allow plugins to modify various types of internal data at runtime.

A plugin can modify data by binding a callback to a filter hook. When the filter is later applied, each bound callback is run in order of priority, and given the opportunity to modify a value by returning a new value.

Example :

```
function add_new_menu ( $menu)
{
    return $menu;
}
add_filter( 'add_menu', 'add_new_menu' );
```

add_shortcode()

There can only be one hook for each shortcode. Which means that if another plugin has a similar shortcode, it will override yours or yours will override theirs depending on which order the plugins are included.

Parameters

\$tag (string) (Required) Shortcode tag to be searched in post content.

\$func (callable) (Required) Hook to run when shortcode is found.

Example :

```
<?php
function button_shortcode()
{
    return '<a href="http://twitter.com/hns" class="twitter-
button">Follow me on Twitter!</a>';
}
add_shortcode('button', 'button_shortcode');
?>
```

Usage : [button]

do_shortcode()

Search content for shortcodes and filter shortcodes through their hooks. If there are no shortcode tags defined, then the content will be returned without any filtering.

Example

Square brackets are required to echo the shortcode, not just the name of the shortcode.

```
<?php
    echo do_shortcode( '[your shortcode]' );
?>
```

register_nav_menu()

Registers a single custom Navigation Menu in the custom menu editor (in WordPress 3.0 and above). This allows administration users to create custom menus for use in a theme.

Example

```
<?php
    register_nav_menu( $location, $description );
?>
```

What is a taxonomy in WordPress?

In the simplest sense, WordPress taxonomies are just a way to group related posts (or other pieces of content) together.

For example, if you have a blog about sports, you could **create taxonomies for the different sports (football, basketball, cricket, etc.)** and assign relevant posts to each taxonomy.

Taxonomies are helpful because they make it easier for readers to find related content.

By default, WordPress includes two types of public-facing taxonomies:

- **Categories**
- **Tags**

Custom Post Types

register_post_type()

With this tag we can registers a post type.

Example

```
function book_setup_post_type() {  
    $args = array(  
        'public'    => true,  
        'label'     => 'Books',  
        'menu_icon' => 'dashicons-book',  
    );  
    register_post_type( 'book', $args );  
}
```

Custom Post Types

register_taxonomy()

Creates or modifies a taxonomy object.

Example

```
<?php
function create_book_tax()
{
    register_taxonomy( 'general', 'book', array( 'label'
=> 'General', 'public' => false, ) );
}
?>
```

Widget Area

register_sidebar()

This function is used to create a sidebar in your theme with writing code in function.php.

register_sidebar() takes an array of arguments (the default value is \$args).

Example

```
register_sidebar(array(  
    "id" => "sidebar",  
    "name" => "Sidebar",  
));
```

Widget Area

dynamic_sidebar()

Dynamic Sidebars can be added to your theme in the functions.php file, using the register_sidebar() function.

After register it you can use in your theme using dynamic_sidebar() function.

Example :

```
<div class="sidebar new-sidebar">  
<?php  
dynamic_sidebar( 'Sidebar' );  
?>  
</div>
```