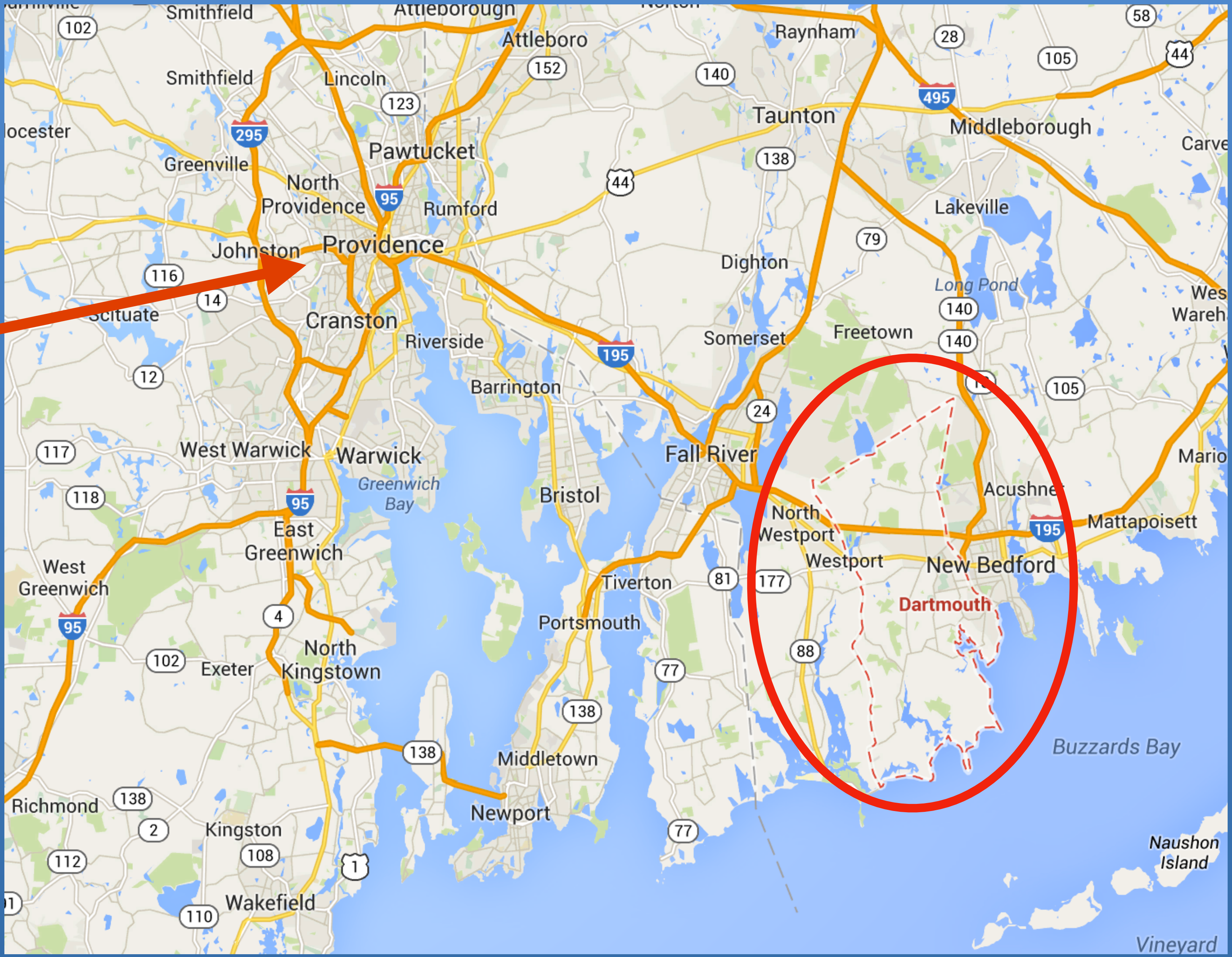
WordPress Plugin Development From Scratch

Jonathan Desrosiers - @Desrosj
jonathandesrosiers.com

About Me

- From: Dartmouth, MA



You’re Here

About Me

- Senior WordPress Engineer @ Linchpin (Pawtucket, RI)

About Me

- Been using WordPress about 8 years
- Theme Developer
- Plugin Developer
- Solution Developer
- Developed 2 years on the WordPress VIP platform

What’s a Plugin?

A WordPress Plugin is a program, or a set of one or more functions that adds a specific set of features, or alters the default behavior of a WordPress site.

Plugins reside in the wp-content/plugins folder

How Do They Work?

- Action Hooks
- Filter Hooks
- WordPress APIs

Actions

Actions are hooks that allow you to execute some code that does something.

WordPress has a ton of these built in for you to use, but you can also create your own.

Filters

Filters are hooks that allow you to CHANGE something.

WordPress also has a ton of these built in for you to use, but you can also create your own.

Goals Of This Workshop

- Learn how to create a plugin from scratch.
- Learn how to follow WordPress best practices
- Learn how to effectively use hooks
- Have a good time!

Resources To Guide You

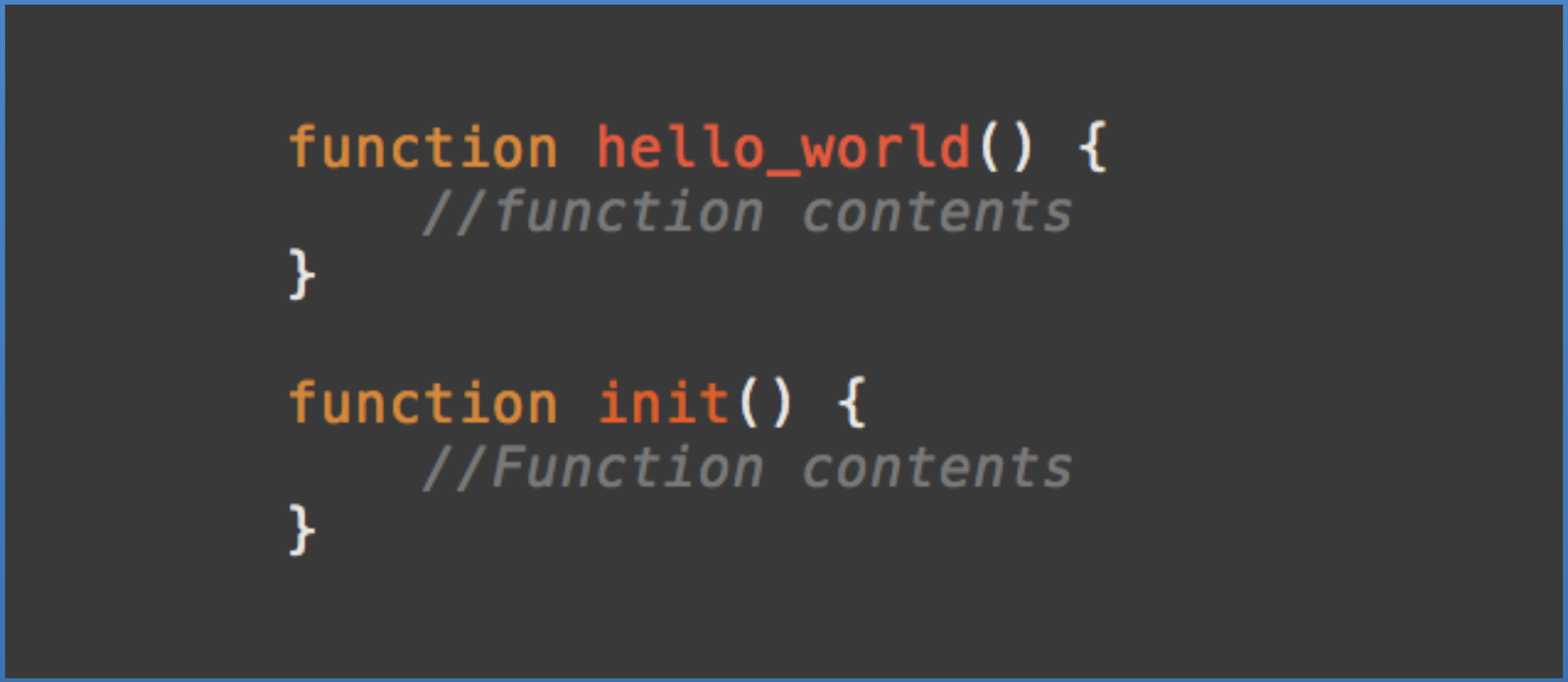
- WordPress Codex: (http://codex.wordpress.org/)
- WordPress PHP Coding Standards: (https://make.wordpress.org/core/handbook/coding-standards/php/)

Important Rules

- Namespacing: Use a unique prefix to all of your plugin’s functions
    - Avoids conflicts with WordPress core functions
    - Avoids conflicts with other plugins

Namespacing

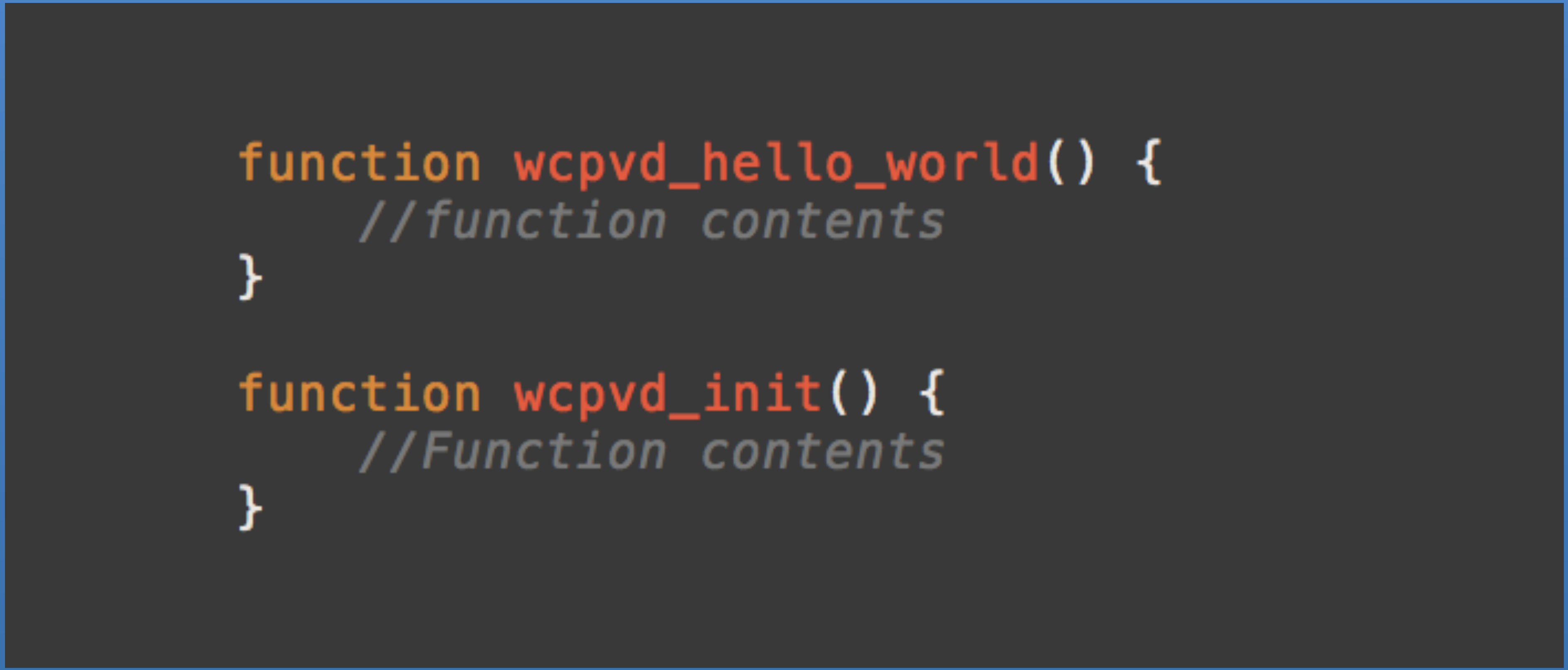
Bad Naming Conventions



```
function hello_world() {
    //function contents
}

function init() {
    //Function contents
}
```

Namespacing

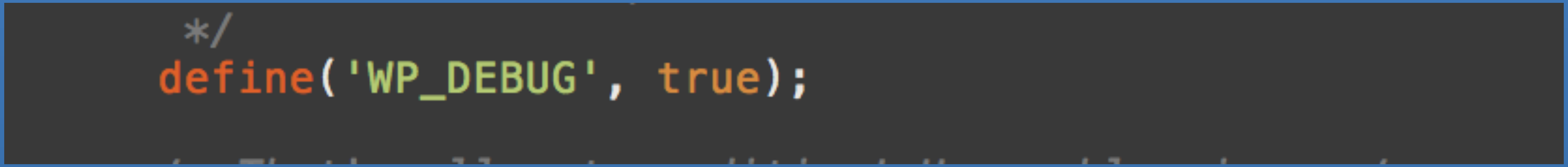
Good Naming Conventions



```
function wcpvd_hello_world() {
    //function contents
}

function wcpvd_init() {
    //Function contents
}
```

WP_Debug Mode

- Turn on WP_Debug mode in your wp-config.php file.
- Will display errors to you on the screen for easier debugging.
- Help you write better code.
- NEVER enable this on production.



```
 */
define('WP_DEBUG', true);
```

Never…

- Develop a new plugin on a production site.

Things Needed

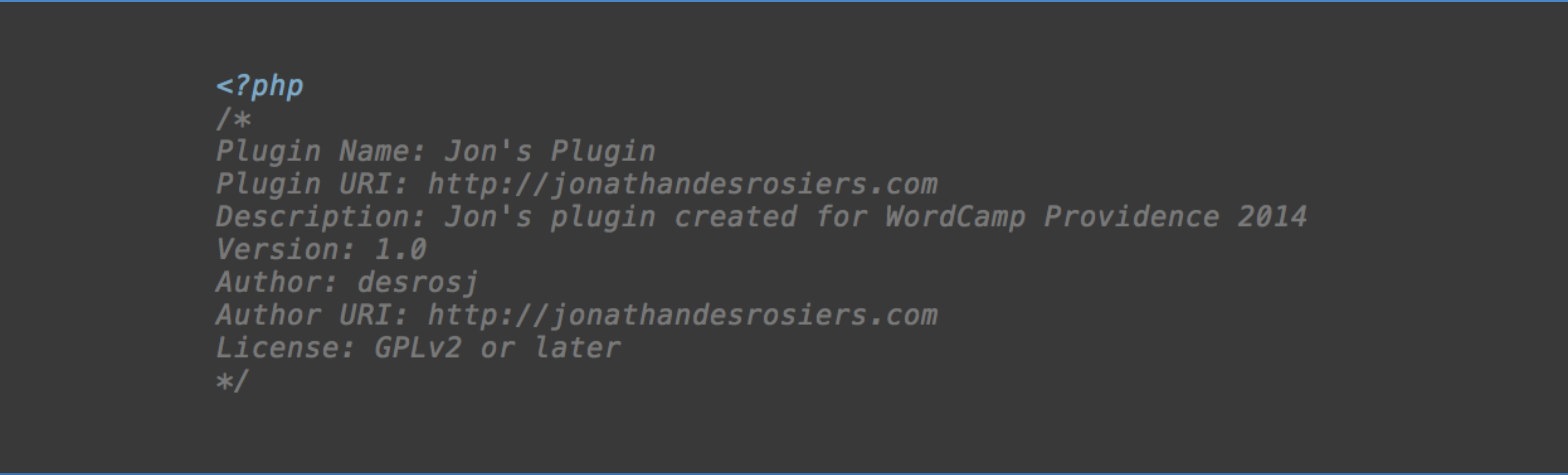
- A playground where you can develop.
  - Local environment
  - WordPress install on your hosting account
  - Anywhere you don’t care about something breaking.
- Code Editor
- Slides & finished plugin file: jonathandesrosiers.com

Questions?

- Do you need help with setting up a local environment?
- Do you need help with choosing a code editor?
- Do you need help with your hair?
- Familiarize yourself with action and filter hook names
  - http://codex.wordpress.org/Plugin_API/Action_Reference
  - http://codex.wordpress.org/Plugin_API/Filter_Reference

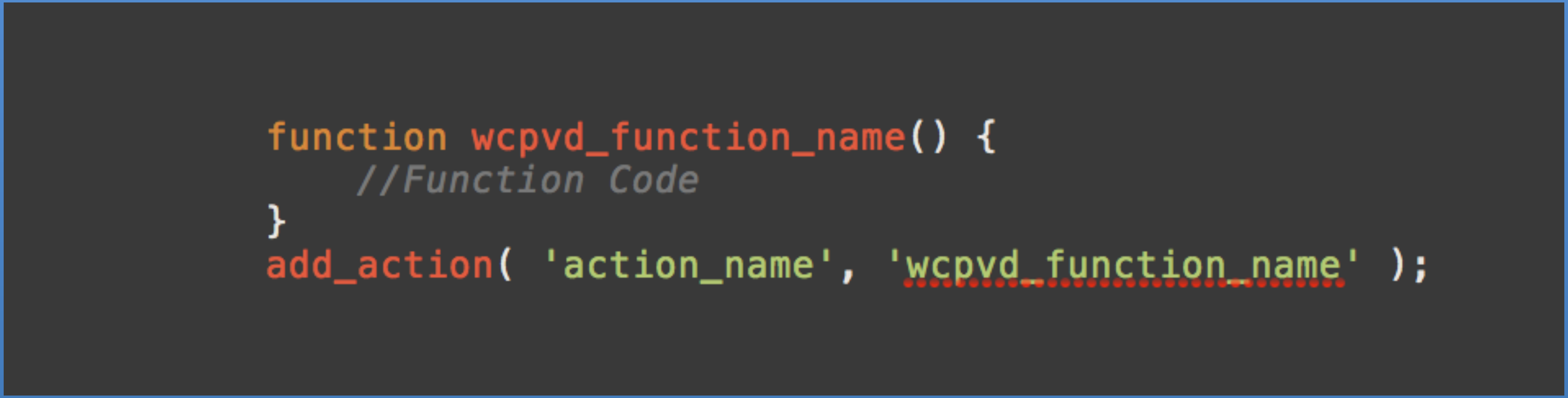
Creating Your Plugin

First part of every plugin is the File Header. This helps WordPress recognize the file as a plugin, and describes itself to admins so they know what they are activating.



```
<?php
/*
Plugin Name: Jon's Plugin
Plugin URI: http://jonathandesrosiers.com
Description: Jon's plugin created for WordCamp Providence 2014
Version: 1.0
Author: desrosj
Author URI: http://jonathandesrosiers.com
License: GPLv2 or later
*/
```

Adding to an Action Hook



```
function wcpvd_function_name() {
    //Function Code
}
add_action( 'action_name', 'wcpvd_function_name' );
```

Create A Custom Post Type

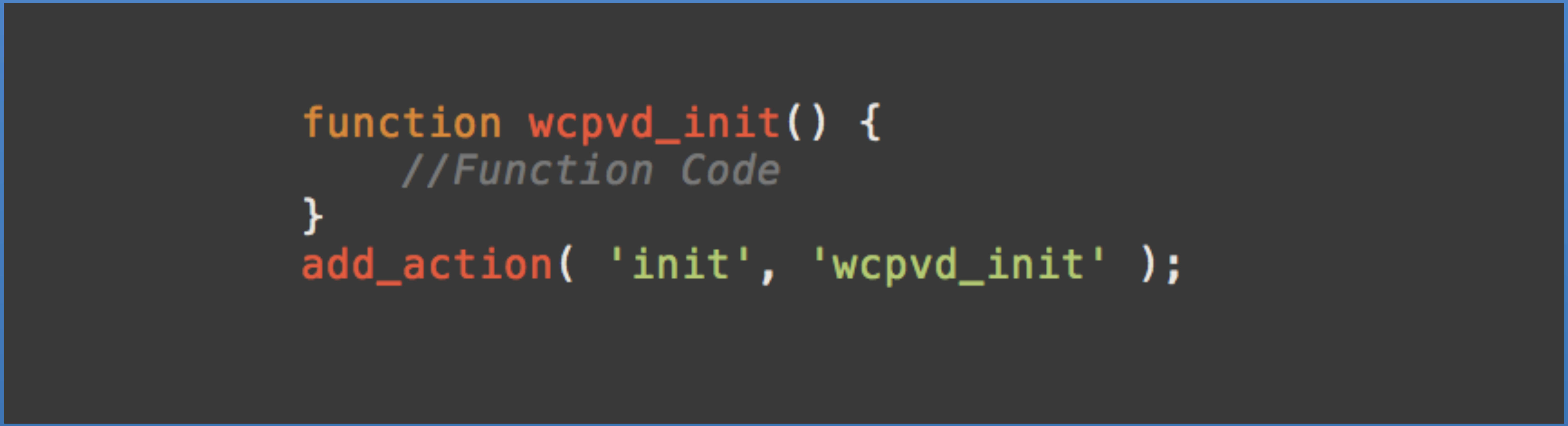
WordPress comes with some core post types (posts, pages, attachments), but sometimes we have other content that should be separate.

Examples:

- Movies
- Locations
- Products
- Cars

Create A Custom Post Type

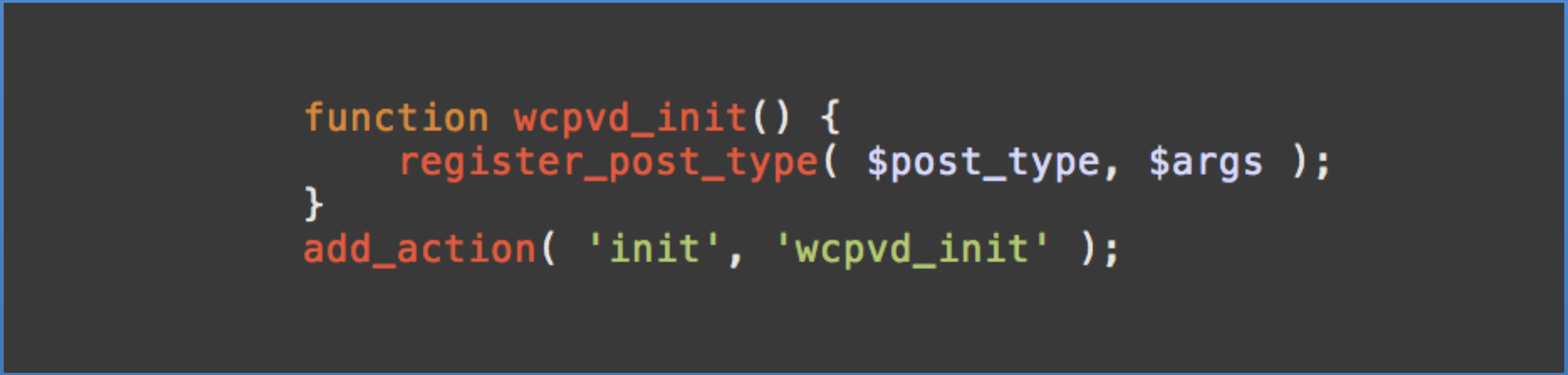
- Should be done within the ‘init’ action (initialization) hook.



```
function wcpvd_init() {
    //Function Code
}
add_action( 'init', 'wcpvd_init' );
```

Create A Custom Post Type

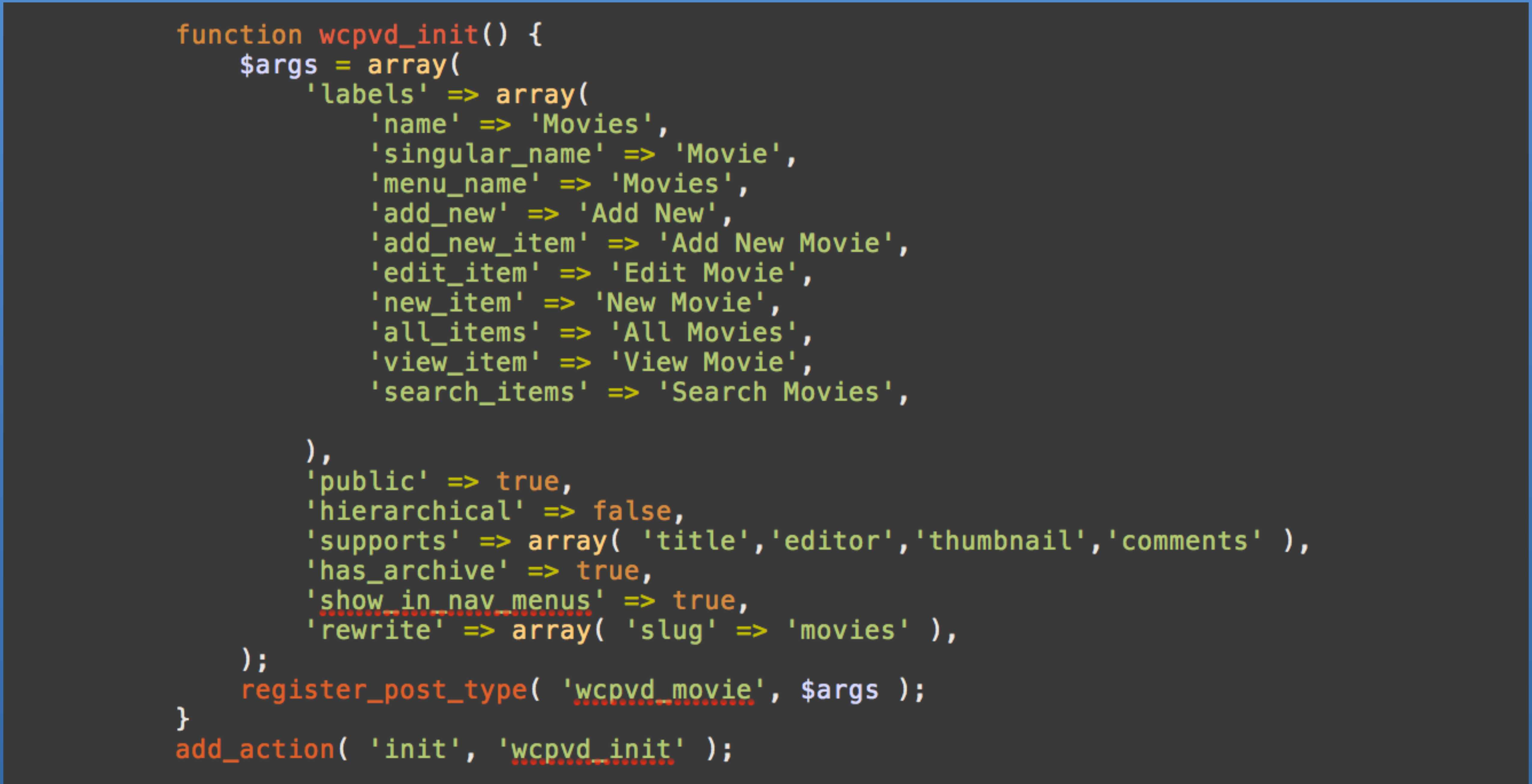
- Use the register_post_type() function



```
function wcpvd_init() {
    register_post_type( $post_type, $args );
}
add_action( 'init', 'wcpvd_init' );
```

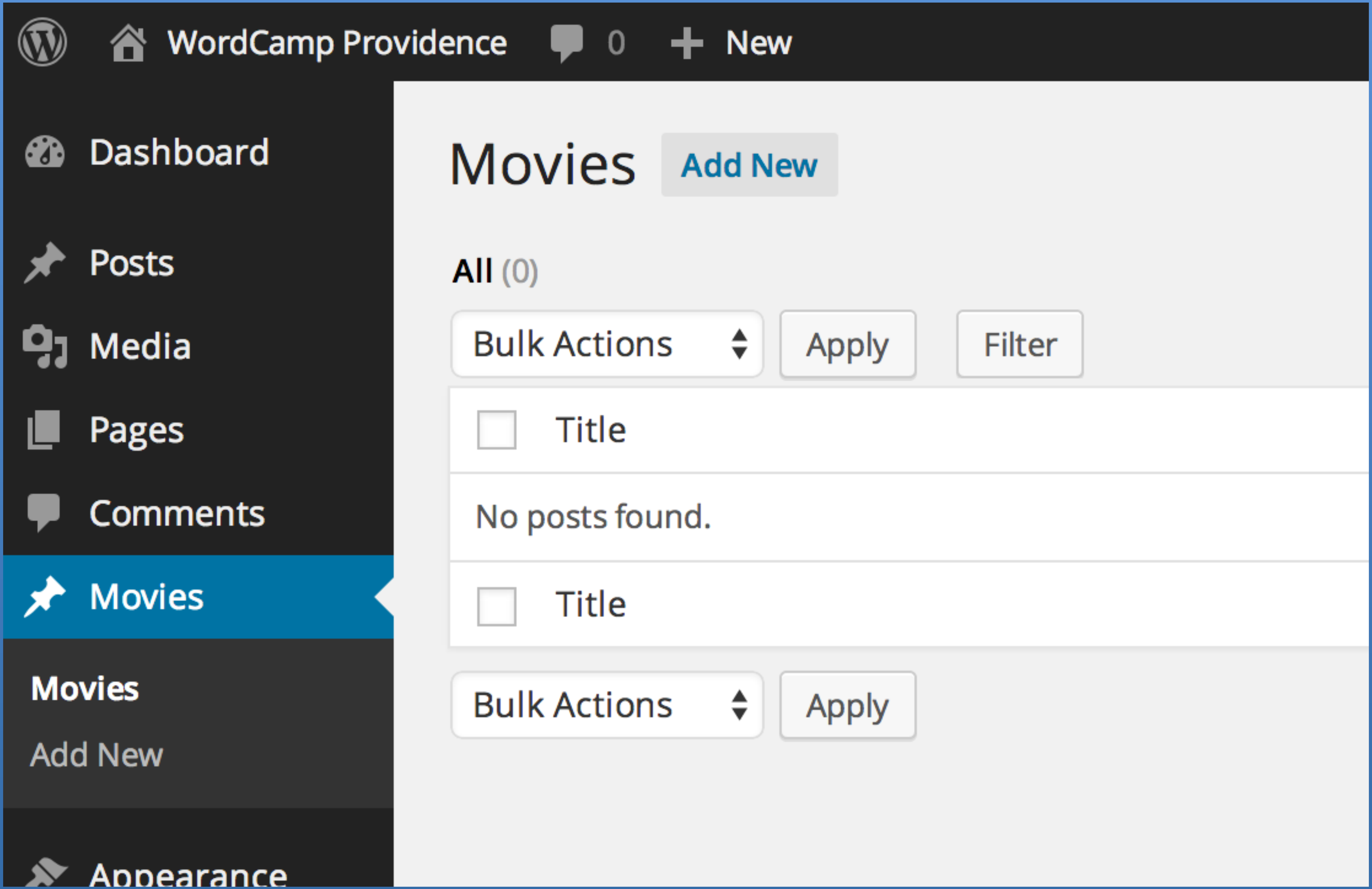
Create A Custom Post Type

- Use the register_post_type() function



```
function wcpvd_init() {
    $args = array(
        'labels' => array(
            'name' => 'Movies',
            'singular_name' => 'Movie',
            'menu_name' => 'Movies',
            'add_new' => 'Add New',
            'add_new_item' => 'Add New Movie',
            'edit_item' => 'Edit Movie',
            'new_item' => 'New Movie',
            'all_items' => 'All Movies',
            'view_item' => 'View Movie',
            'search_items' => 'Search Movies',

        ),
        'public' => true,
        'hierarchical' => false,
        'supports' => array( 'title','editor','thumbnail','comments' ),
        'has_archive' => true,
        'show_in_nav_menus' => true,
        'rewrite' => array( 'slug' => 'movies' ),
    );
    register_post_type( 'wcpvd_movie', $args );
}
add_action( 'init', 'wcpvd_init' );
```

Create A Custom Post Type

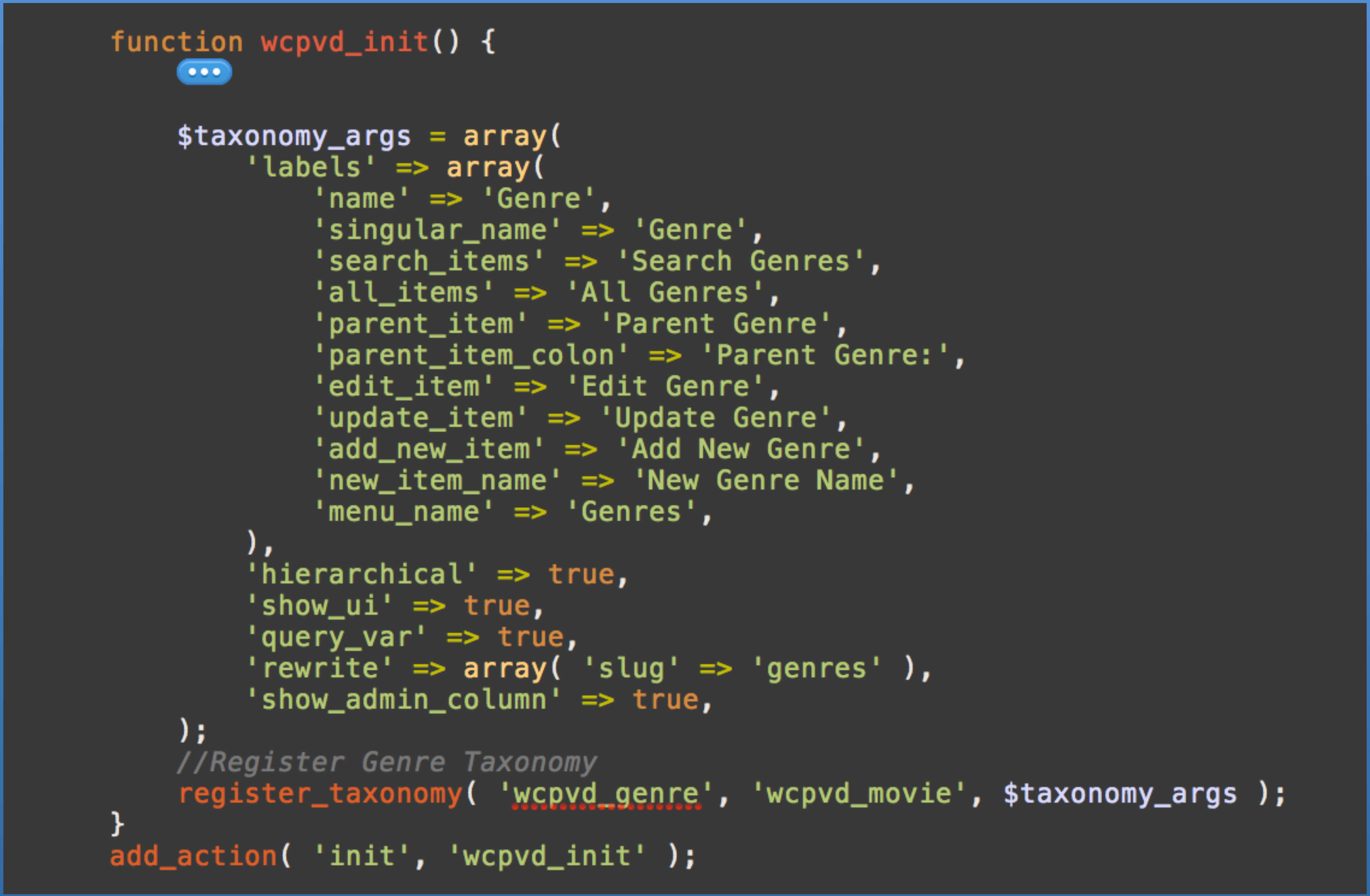


Create A Custom Taxonomy

- Also Should be done within the ‘init’ action hook.

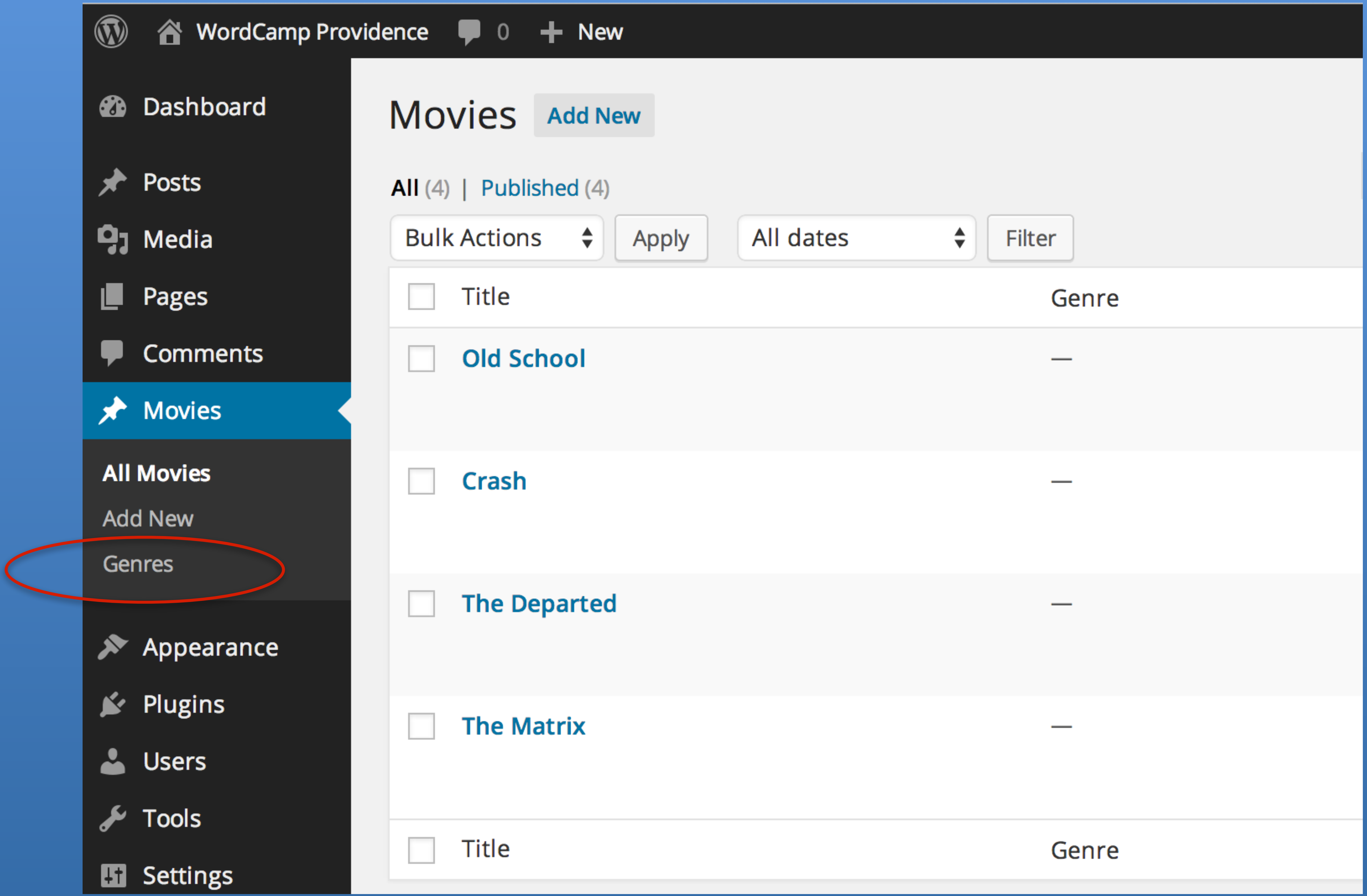
Create A Custom Taxonomy

- Use register_taxonomy() function.



```
function wcpvd_init() {
    •••

    $taxonomy_args = array(
        'labels' => array(
            'name' => 'Genre',
            'singular_name' => 'Genre',
            'search_items' => 'Search Genres',
            'all_items' => 'All Genres',
            'parent_item' => 'Parent Genre',
            'parent_item_colon' => 'Parent Genre:',
            'edit_item' => 'Edit Genre',
            'update_item' => 'Update Genre',
            'add_new_item' => 'Add New Genre',
            'new_item_name' => 'New Genre Name',
            'menu_name' => 'Genres',
        ),
        'hierarchical' => true,
        'show_ui' => true,
        'query_var' => true,
        'rewrite' => array( 'slug' => 'genres' ),
        'show_admin_column' => true,
    );
    //Register Genre Taxonomy
    register_taxonomy( 'wcpvd_genre', 'wcpvd_movie', $taxonomy_args );
}
add_action( 'init', 'wcpvd_init' );
```

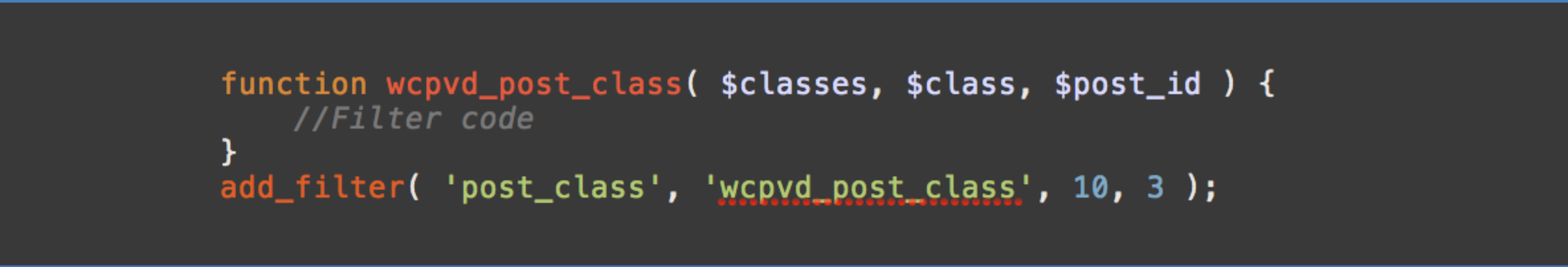
Create A Custom Taxonomy



□	Title	Genre
□	Old School	—
□	Crash	—
□	The Departed	—
□	The Matrix	—
□	Title	Genre

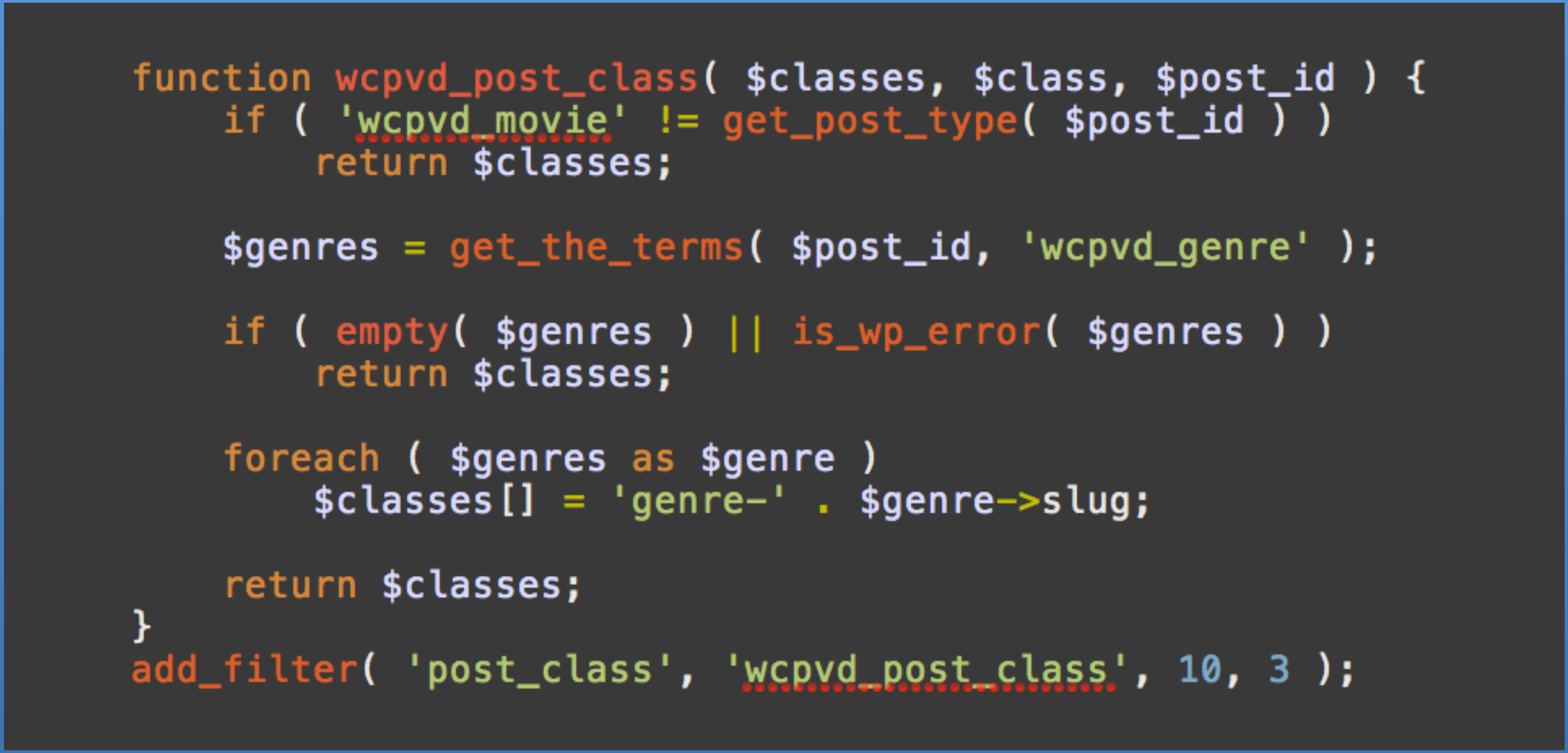
Add CSS Class to Movie CPT

- post_class() function outputs classes for a post.
- Add to it with the ‘post_class’ filter.



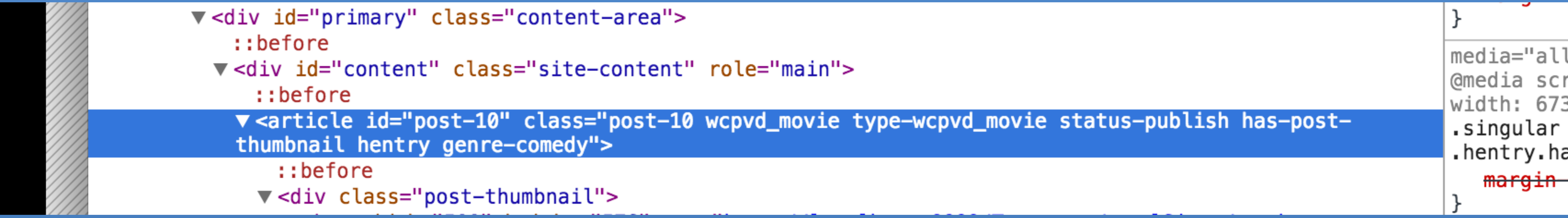
```
function wcpvd_post_class( $classes, $class, $post_id ) {
    //Filter code
}
add_filter( 'post_class', 'wcpvd_post_class', 10, 3 );
```

Add CSS Class to Movie CPT



```php
function wcpvd_post_class( $classes, $class, $post_id ) {
    if ( 'wcpvd_movie' != get_post_type( $post_id ) )
        return $classes;

    $genres = get_the_terms( $post_id, 'wcpvd_genre' );

    if ( empty( $genres ) || is_wp_error( $genres ) )
        return $classes;

    foreach ( $genres as $genre )
        $classes[] = 'genre-' . $genre->slug;

    return $classes;
}
add_filter( 'post_class', 'wcpvd_post_class', 10, 3 );
```

Add CSS Class to Movie CPT



```
▼<div id="primary" class="content-area">
   ::before
  ▼<div id="content" class="site-content" role="main">
     ::before
    ▼<article id="post-10" class="post-10 wcpvd_movie type-wcpvd_movie status-publish has-post-thumbnail hentry genre-comedy">
       ::before
      ▼<div class="post-thumbnail">
```

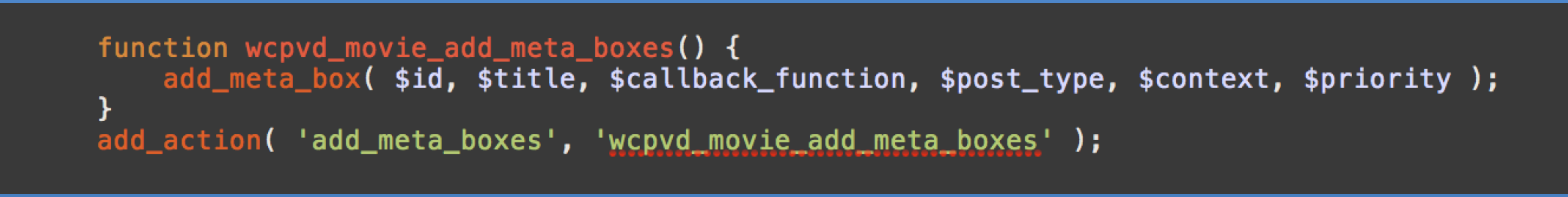
```
}
media="all
@media scr
width: 673
.singular
.hentry.ha
  ~~margin~~
}
```

Custom Post Meta

- Metadata is data that describes other data
- Use the WordPress Metadata API

Custom Post Meta - Step 1

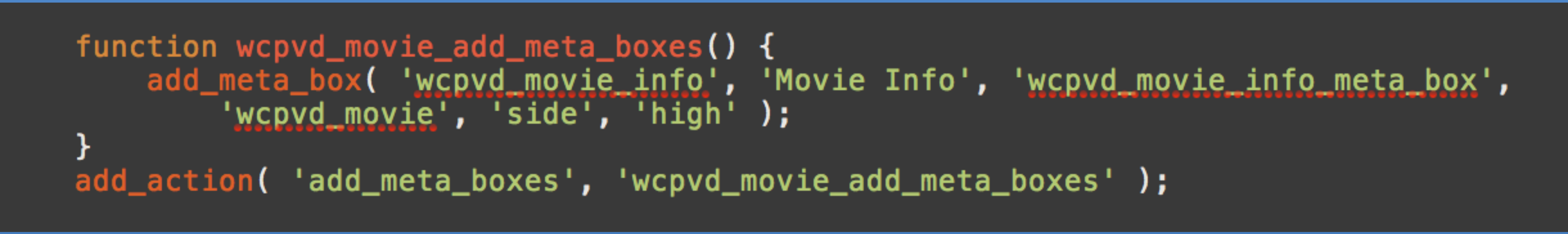
- Register our post meta box



```
function wcpvd_movie_add_meta_boxes() {
    add_meta_box( $id, $title, $callback_function, $post_type, $context, $priority );
}
add_action( 'add_meta_boxes', 'wcpvd_movie_add_meta_boxes' );
```

Custom Post Meta - Step 1

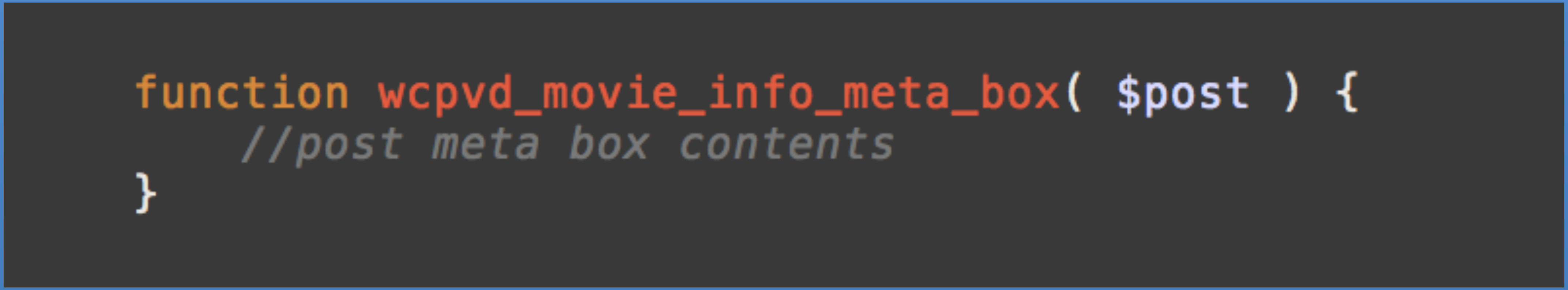
- Register our post meta box



```
function wcpvd_movie_add_meta_boxes() {
    add_meta_box( 'wcpvd_movie_info', 'Movie Info', 'wcpvd_movie_info_meta_box',
        'wcpvd_movie', 'side', 'high' );
}
add_action( 'add_meta_boxes', 'wcpvd_movie_add_meta_boxes' );
```

Custom Post Meta - Step 2

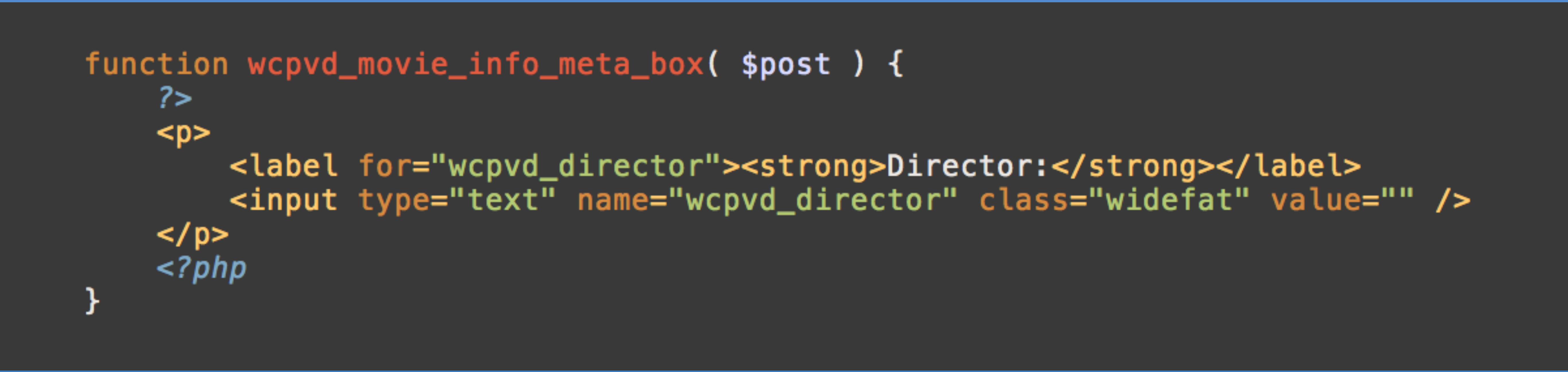
- Display our post meta box.



```
function wcpvd_movie_info_meta_box( $post ) {
    //post meta box contents
}
```

Custom Post Meta - Step 2

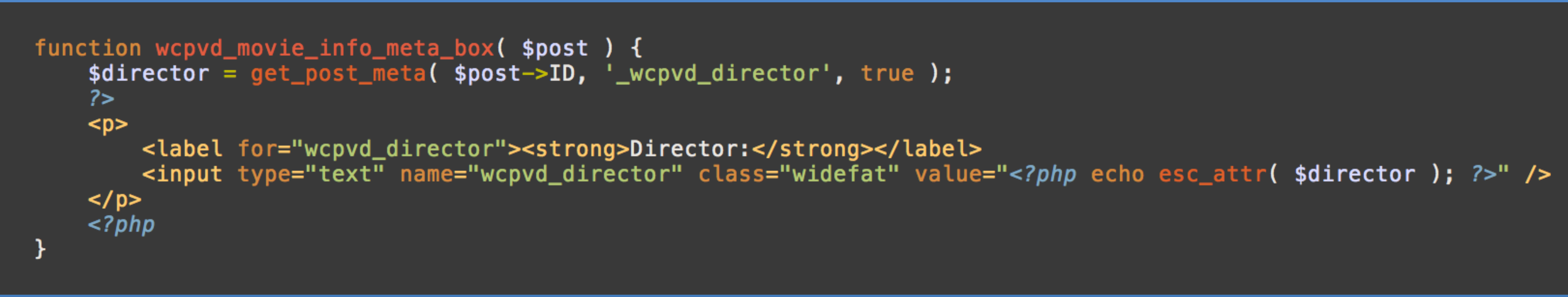
- Display our post meta box.



```
function wcpvd_movie_info_meta_box( $post ) {
    ?>
    <p>
        <label for="wcpvd_director"><strong>Director:</strong></label>
        <input type="text" name="wcpvd_director" class="widefat" value="" />
    </p>
    <?php
}
```

Custom Post Meta - Step 2

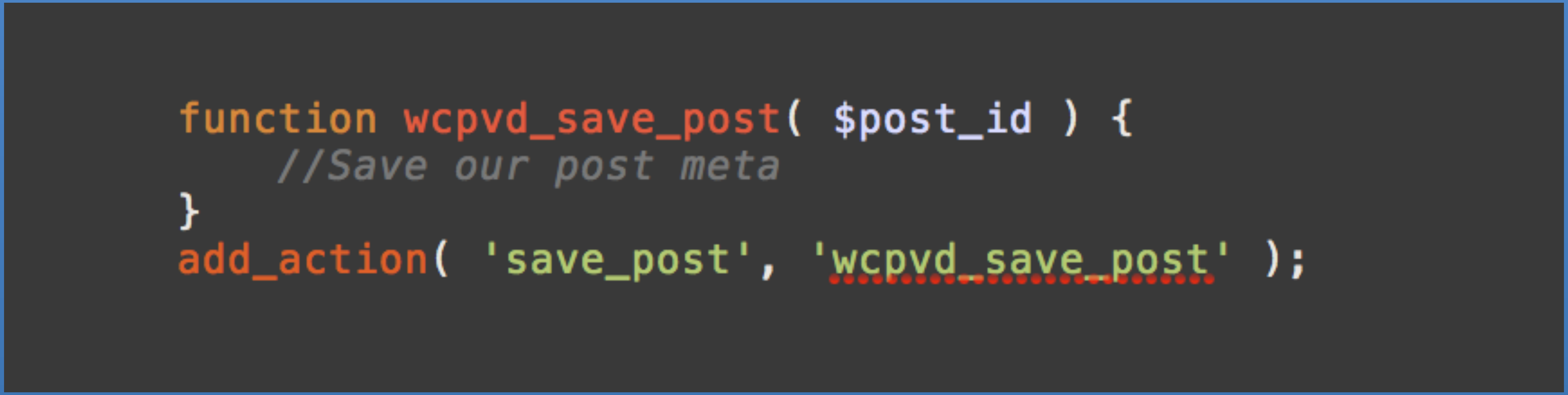
- Display our post meta box.



```
function wcpvd_movie_info_meta_box( $post ) {
    $director = get_post_meta( $post->ID, '_wcpvd_director', true );
    ?>
    <p>
        <label for="wcpvd_director"><strong>Director:</strong></label>
        <input type="text" name="wcpvd_director" class="widefat" value="<?php echo esc_attr( $director ); ?>" />
    </p>
    <?php
}
```

Custom Post Meta - Step 2

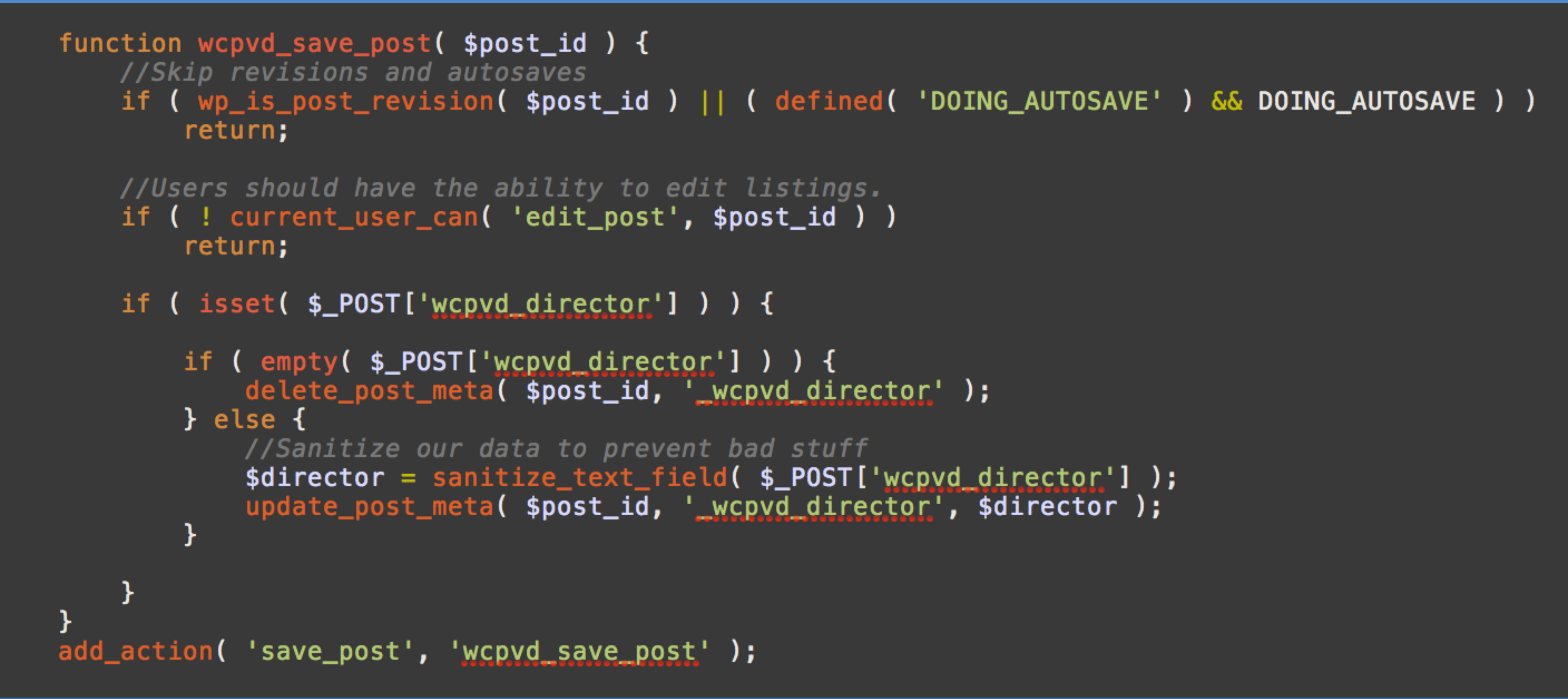
- Save our post meta.
- Use save_post action hook



```
function wcpvd_save_post( $post_id ) {
    //Save our post meta
}
add_action( 'save_post', 'wcpvd_save_post' );
```

Custom Post Meta - Step 2

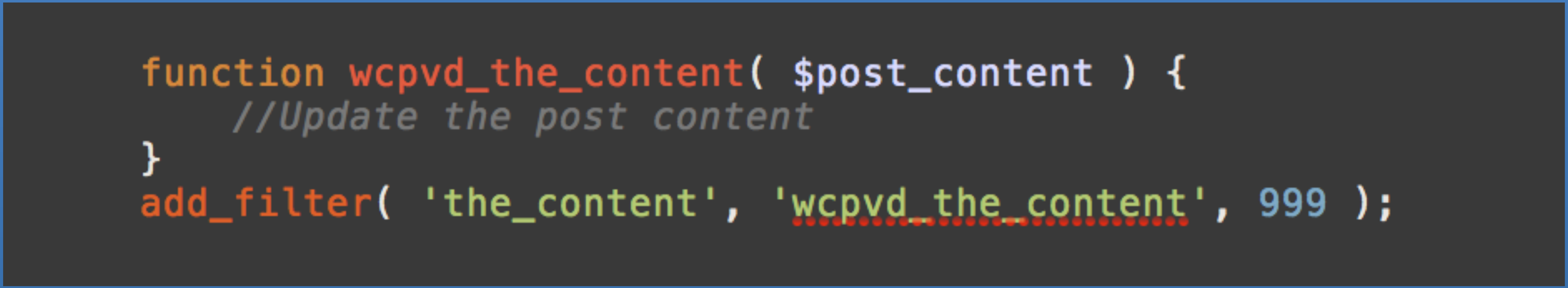
- Save our post meta.



```
function wcpvd_save_post( $post_id ) {
    //Skip revisions and autosaves
    if ( wp_is_post_revision( $post_id ) || ( defined( 'DOING_AUTOSAVE' ) && DOING_AUTOSAVE ) )
        return;

    //Users should have the ability to edit listings.
    if ( ! current_user_can( 'edit_post', $post_id ) )
        return;

    if ( isset( $_POST['wcpvd_director'] ) ) {

        if ( empty( $_POST['wcpvd_director'] ) ) {
            delete_post_meta( $post_id, '_wcpvd_director' );
        } else {
            //Sanitize our data to prevent bad stuff
            $director = sanitize_text_field( $_POST['wcpvd_director'] );
            update_post_meta( $post_id, '_wcpvd_director', $director );
        }

    }
}
add_action( 'save_post', 'wcpvd_save_post' );
```

Custom Post Meta - Step 3

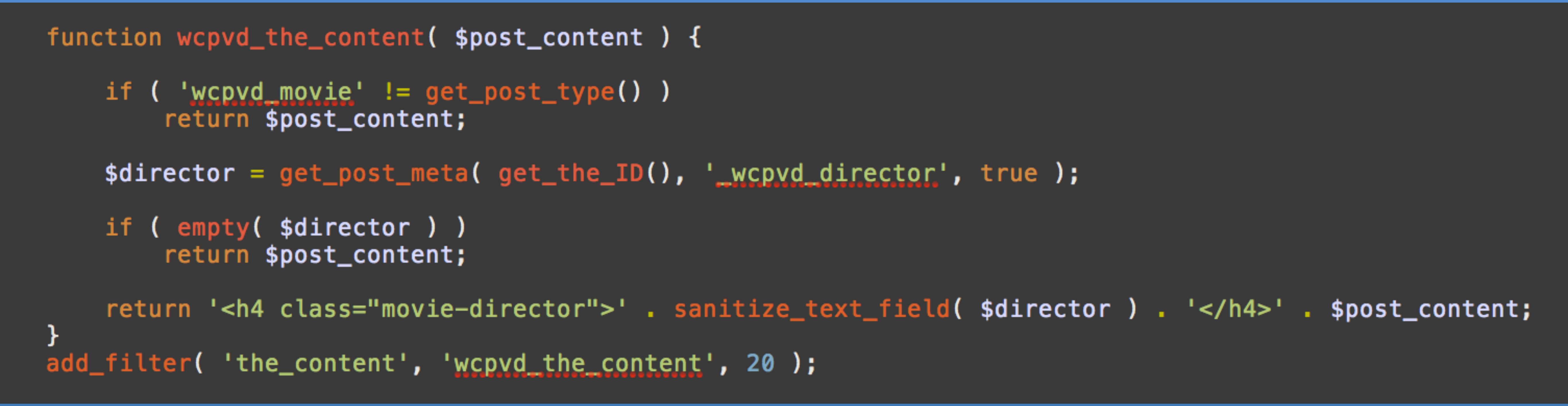
- Do something with the post meta.
- Prepend it to our post content.
- Append it to the title
- Give admins the choice of where it should go



```
function wcpvd_the_content( $post_content ) {
    //Update the post content
}
add_filter( 'the_content', 'wcpvd_the_content', 999 );
```

Custom Post Meta - Step 3

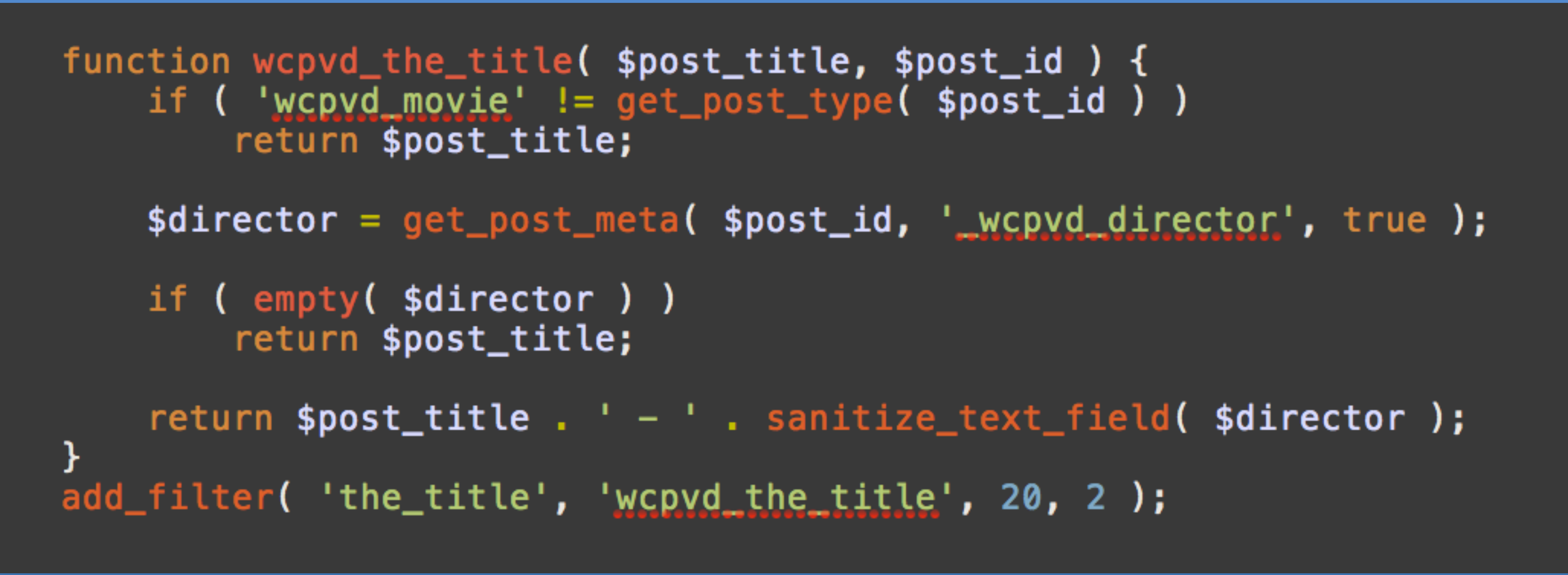
- Do something with the post meta - post content.



```
function wcpvd_the_content( $post_content ) {

    if ( 'wcpvd_movie' != get_post_type() )
        return $post_content;

    $director = get_post_meta( get_the_ID(), '_wcpvd_director', true );

    if ( empty( $director ) )
        return $post_content;

    return '<h4 class="movie-director">' . sanitize_text_field( $director ) . '</h4>' . $post_content;
}
add_filter( 'the_content', 'wcpvd_the_content', 20 );
```

Custom Post Meta - Step 3

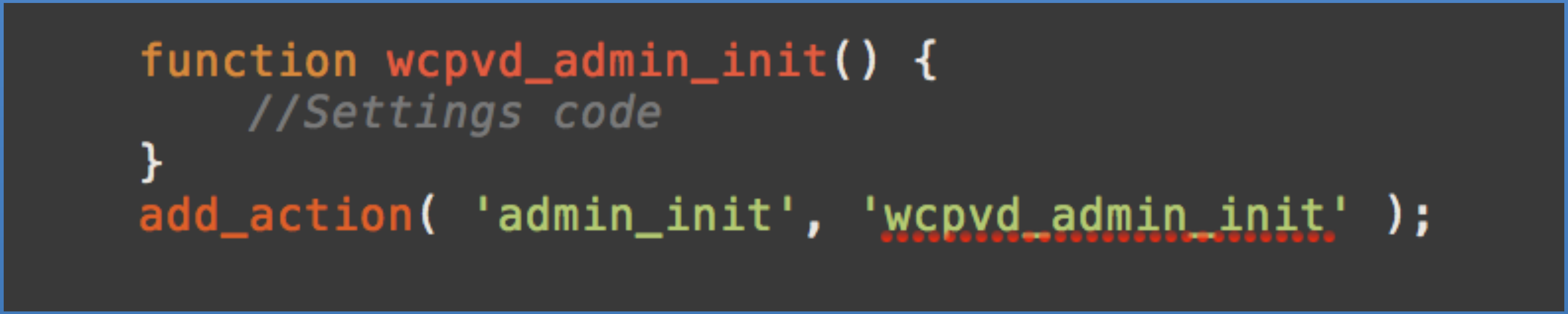
- Do something with the post meta - post title.



```
function wcpvd_the_title( $post_title, $post_id ) {
    if ( 'wcpvd_movie' != get_post_type( $post_id ) )
        return $post_title;

    $director = get_post_meta( $post_id, '_wcpvd_director', true );

    if ( empty( $director ) )
        return $post_title;

    return $post_title . ' - ' . sanitize_text_field( $director );
}
add_filter( 'the_title', 'wcpvd_the_title', 20, 2 );
```

Add Custom Setting

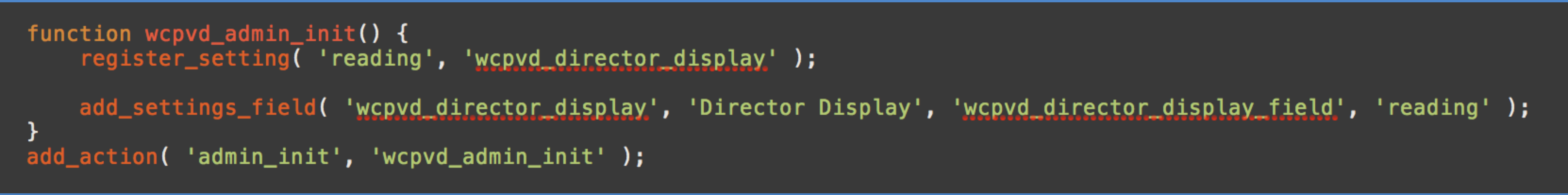
- Let’s add our custom setting
- Use the ‘admin_init’ action hook.



```
function wcpvd_admin_init() {
    //Settings code
}
add_action( 'admin_init', 'wcpvd_admin_init' );
```

Add Custom Setting

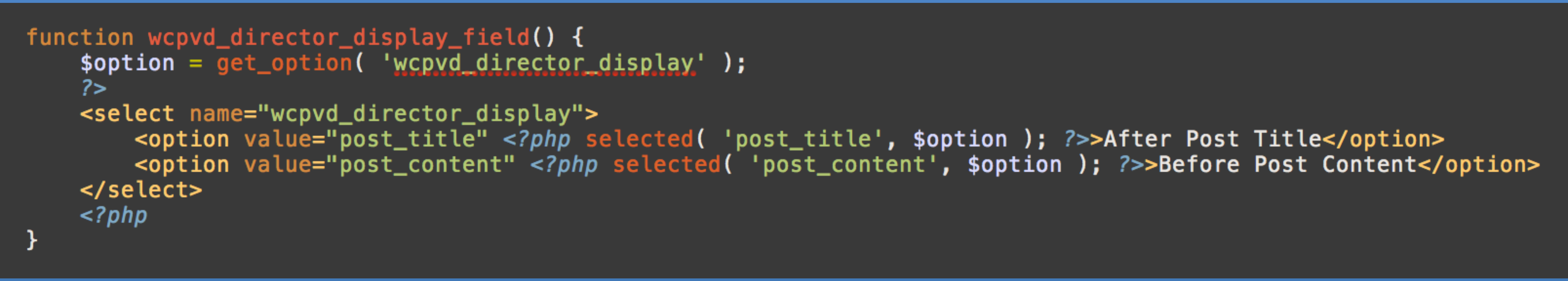
- Let’s add our custom setting



```
function wcpvd_admin_init() {
    register_setting( 'reading', 'wcpvd_director_display' );

    add_settings_field( 'wcpvd_director_display', 'Director Display', 'wcpvd_director_display_field', 'reading' );
}
add_action( 'admin_init', 'wcpvd_admin_init' );
```

Add Custom Setting

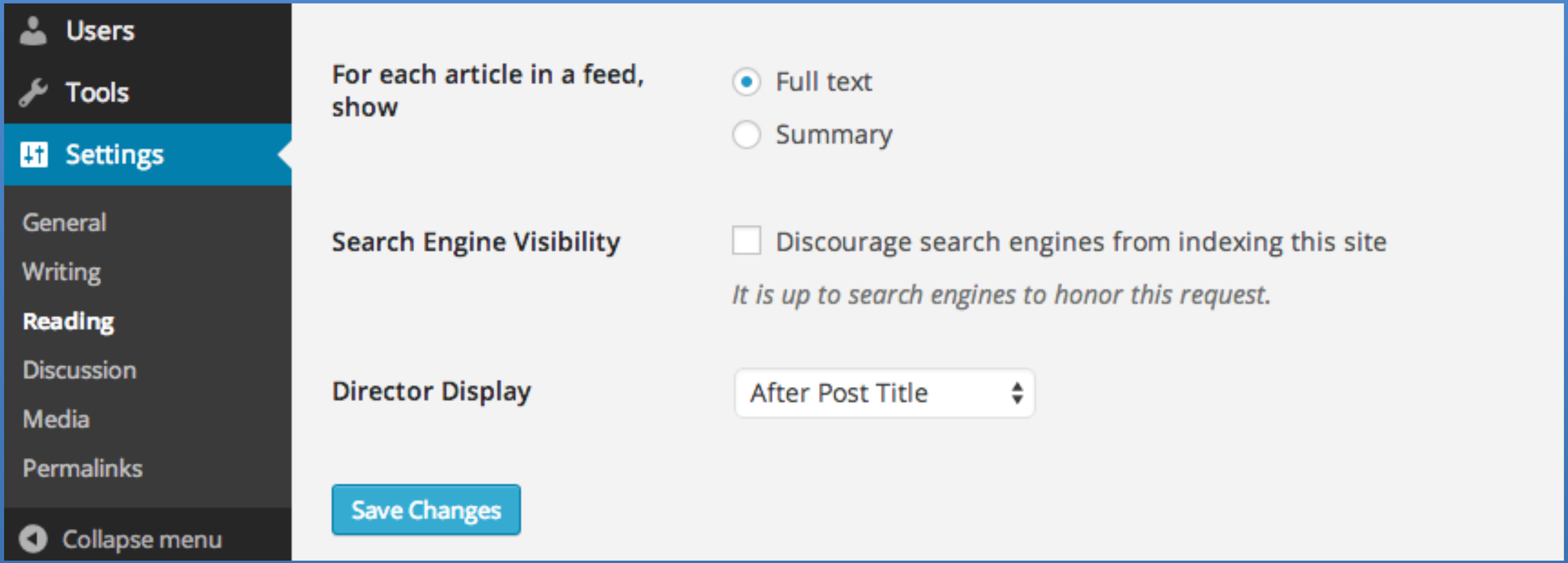
- Let’s add our custom setting



```
function wcpvd_director_display_field() {
    $option = get_option( 'wcpvd_director_display' );
    ?>
    <select name="wcpvd_director_display">
        <option value="post_title" <?php selected( 'post_title', $option ); ?>>After Post Title</option>
        <option value="post_content" <?php selected( 'post_content', $option ); ?>>Before Post Content</option>
    </select>
    <?php
}
```

Add Custom Setting

- Let’s add our custom setting



For each article in a feed, show
● Full text
○ Summary

Search Engine Visibility
□ Discourage search engines from indexing this site

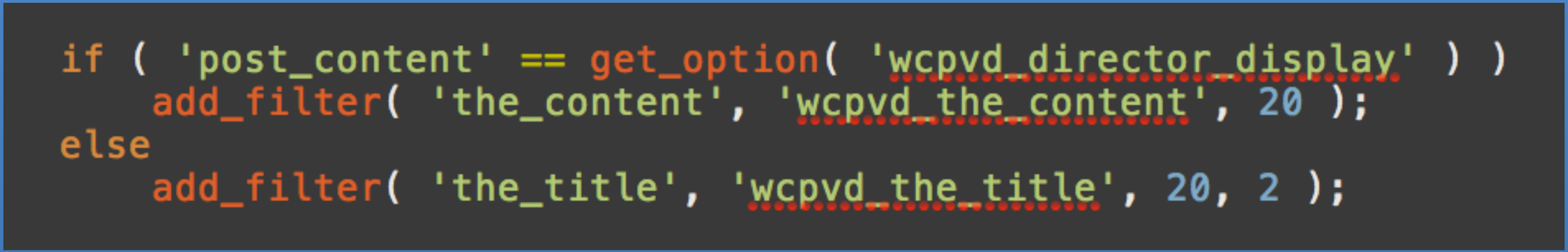
It is up to search engines to honor this request.

Director Display
After Post Title

Add Custom Setting

- Use our new setting

- Update our filters to only add one based on setting



```
if ( 'post_content' == get_option( 'wcpvd_director_display' ) )
    add_filter( 'the_content', 'wcpvd_the_content', 20 );
else
    add_filter( 'the_title', 'wcpvd_the_title', 20, 2 );
```

Good Job!

- Questions?
- Twitter: @Desrosj
- Slides & Finished Plugin: jonathandesrosiers.com