

WordPress Plugins 101

– How to build a
minimum viable
WordPress plugin

- A WordPress plugin is a file or collection of files written in PHP and are used to extend core functionality.
- Every time a page is loaded in WordPress all active plugins are also loaded.
- More plugins = slower page loads, so only use the plugins you need.

What is a WordPress plugin anyway?

- Code snippets
 - Change theme re-copy the code snippet.
 - Re-use the code snippet across many themes.
- High level code quality judgement.

I'm not a developer, why should I care how a plugin is built?

- Where do we start?
 - Open up a text file, and put a file header in it.

```
1  <?php
2  /*
3  Plugin Name: WordPress.org Plugin
4  Plugin URI:  https://developer.wordpress.org/plugins/the-basics/
5  Description: Basic WordPress Plugin Header Comment
6  Version:     20160911
7  Author:      WordPress.org
8  Author URI:  https://developer.wordpress.org/
9  License:     GPL2
10 License URI: https://www.gnu.org/licenses/gpl-2.0.html
11 Text Domain: wporg
12 Domain Path: /languages
13 */
```

Anatomy of a simple plugin – part 1.

- Next we include our code

```
// This just echoes the chosen line, we'll position it later
function hello_dolly() {
    $chosen = hello_dolly_get_lyric();
    echo "<p id='dolly'>$chosen</p>";
}

// Now we set that function up to execute when the admin_notices action is called
add_action( 'admin_notices', 'hello_dolly' );

// We need some CSS to position the paragraph
function dolly_css() {
    // This makes sure that the positioning is also good for right-to-left languages
    $x = is_rtl() ? 'left' : 'right';

    echo "
    <style type='text/css'>
    #dolly {
        float: $x;
        padding-$x: 15px;
        padding-top: 5px;
        margin: 0;
        font-size: 11px;
    }
    </style>
    ";
}

add_action( 'admin_head', 'dolly_css' );

?>
```















- Minimum requirement in the file header is to have the plugin name.
- More elements are recommended such as :-
 - Description, Version, Author.
 - Full list can be found at <https://developer.wordpress.org/plugins/the-basics/header-requirements/>
- The filename of the file must end in .php
- The filename of the file must be unique, WP slug.

Anatomy of a simple plugin – part 3.

- Could put all the code for your plugin into one file.
- I would recommend not doing this.
 - Change less break less.
 - If you have functionality spread across multiple files, only need to change what is necessary.
- Recommend taking an Object Oriented approach to writing your plugin.
- Steeper learning curve but easier to maintain.
- Can lead to a plugin which is designed better.

- What does an Object Oriented design look like?
- Functionality for a plugin in an object oriented design are held in classes.
- To access the functionality in a class you have to create an object from the class.
- Class = template, object = thing to work with.

- Where you files are is important so you can find them again.

- ▶  css
- ▶  focs-list-table
- ▶  includes
 -  class-focs-ddl-builder.php
 -  class-focs-fm-data-access.php
 -  class-focs-fm-installer.php
 -  class-focs-fm-utilities.php
 -  class-focs-form-monitor.php
 -  class-focs-view-manager.php
 -  class-interactions-detail-table.php
 -  class-interactions-summary-table.php
- ▶  js
- ▶  views
 -  focs-form-monitor.php

A note about file structure.

```
<?php
```

```
class My_Lovely_Plugin {  
    public function do_something() {  
    }  
    public function do_something_else() {  
    }  
}
```

- Remember your class name needs to be unique.
- We can achieve this by using namespaces

Your first class and a note about namespaces.

- A namespace is simply a way of collecting your classes in one place, but allowing your classes to sit in different files.
- It gives you the ability to create an object from a class using a path which uniquely identifies that class.

class-my-lovely-plugin.php

```
<?php
namespace Focs_My_Lovely_Plugin {
    class My_Lovely_Plugin {
        public function do_something() {
        }
        public function do_something_else() {
        }
    }
}
```

Your first class and a note about namespaces – part 1.

class-my-lovely-plugin-utils.php

```
<?php
```

```
namespace Focs_My_Lovely_Plugin {  
    class My_Lovely_Plugin_Utils {  
        public function do_some_utility_routine() {  
        }  
        public function do_another_utility_routine() {  
        }  
    }  
}
```

Your first class and a note about namespaces – part 2.

```
<?php
```

```
namespace Focs_My_Lovely_Plugin {  
  
    class My_Lovely_Plugin {  
  
        public function do_something() {  
  
            //create an object of the class(an instance)  
            $my_lovely_object = new My_Lovely_Plugin_Utils();  
  
            //Now call the function on the instance  
            $my_lovely_object->do_some_utility_routine();  
  
        }  
  
        public function do_something_else() {  
  
        }  
  
    }  
}
```

Calling the function in a class.

```
<?php
```

```
≡ /*  
Plugin Name: My lovely plugin  
Description: My first lovely plugin  
Version: 1.0.0  
Author: Chris O'Dell  
License: GPLv2 or later  
*/  
  
include_once plugin_dir_path( __FILE__ ). 'class-my-lovely-plugin.php';  
  
≡ function my_unique_function_name() {  
    $my_plugin_instance = new My_Lovely_Plugin();  
    $my_plugin_instance->do_something();  
}  
  
≡ function my_unique_function_name_2() {  
    $my_plugin_instance = new My_Lovely_Plugin();  
    $my_plugin_instance->do_something_else();  
}
```

Building the header file in an OO design.

- Take a look at the WordPress codex
https://codex.wordpress.org/Writing_a_Plugin
- Read up on WordPress actions – hooks & filters.
- Google search for building Object Oriented WordPress plugins.
- Pippins plugins : <https://pippinsplugins.com/>
- Tom McFarlin : <https://tommmcfarlin.com/>

What next?

Questions?

Questions?