



THE SOUTHWEST FLORIDA HACKERSPACE PRESENTS:

# EXTENDING WORDPRESS WITH PLUGIN DEVELOPMENT

# WHO AM I

## RUSSELL BENZING

me@russellbenzing.com

github.com/rbenzing

- Father
- Husband
- Dog Lover
- Entrepreneur
- Engineer
- Artist
- Gamer
- Handyman
- WordPress Developer
- Technology Hacker
- Marvel & Star Wars Nerd
- Breaker of Chains



Co-Developer of

**Inbound Rocket**

InboundRocket.co



Operating President  
swflhackerspace.com

Web & App Developer

www.impulsecreative.com



**IMPULSE  
CREATIVE**

We Are Hiring!





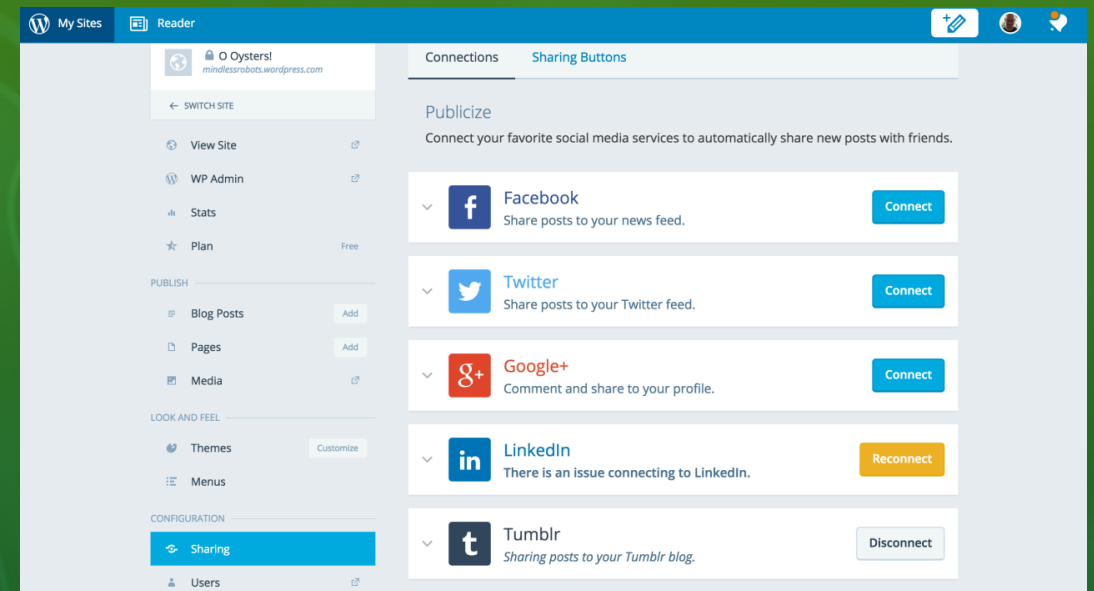
# WORDPRESS IS A FRAMEWORK AND A PLATFORM

Dedicated/Shared  
Hosted Server



- OR -

Paid/Free Account  
WordPress.com



Requirements:

- PHP version 7.2 or greater.
- MySQL 5.6+ or MariaDB 10+.
- HTTPS support.

Requirements:

- Keyboard, Mouse & Monitor
- Ability to remember login info
- An endless drive to write

# SO WHY USE WORDPRESS?

- It's Open Source! Uses GPL (General Public License)
- 28% of the now wide web runs on WordPress.
- 55,000+ WordPress plugins in the official repo.
- 1.25m+ total plugin downloads on WordPress.org.
- WP sites receive 22+ billion monthly page views.
- 2.7m+ global monthly searches for 'WordPress'.
- WP 4.8 has been downloaded 60 million+ times.



# WHAT SHOULD WE KNOW FIRST?

- Don't touch WordPress core directly.
- Know how the WordPress Loop works.
- Understand there are many ways to extend WordPress.
- Always try to prevent conflicts within your plugin.
- Be unique in your class, function and var names.
- Validate, Sanitize and Escape your data.
- Leave a small footprint.

```
<?php

if ( have_posts() ) {
    while ( have_posts() ) {
        the_post();
        //
        // Post Content here
        //
    } // end while
} // end if

?>
```

- function\_exists()
- class\_exists()
- Jquery no conflicts mode
- Use bundled libraries

- |                         |                           |
|-------------------------|---------------------------|
| ■ Dashboard Widgets API | ■ Rewrite API             |
| ■ Database API          | ■ Settings API            |
| ■ HTTP API              | ■ Shortcode API           |
| ■ File Header API       | ■ Theme modification API  |
| ■ Filesystem API        | ■ Theme customization API |
| ■ Heartbeat API         | ■ Transients API          |
| ■ Metadata API          | ■ Widgets API             |
| ■ Options API           | ■ XML-RPC WordPress API   |
| ■ Plugin API            |                           |
| ■ Quicktags API         |                           |

# WHAT ARE WE GOING TO LEARN?

- How to build a WordPress plugin from scratch.
- Learn built-in WordPress functions useful for rapid plugin development.
- Learn how to get your plugin on the WordPress Directory (SVN Repository)





# HOW DO PLUGINS WORK?

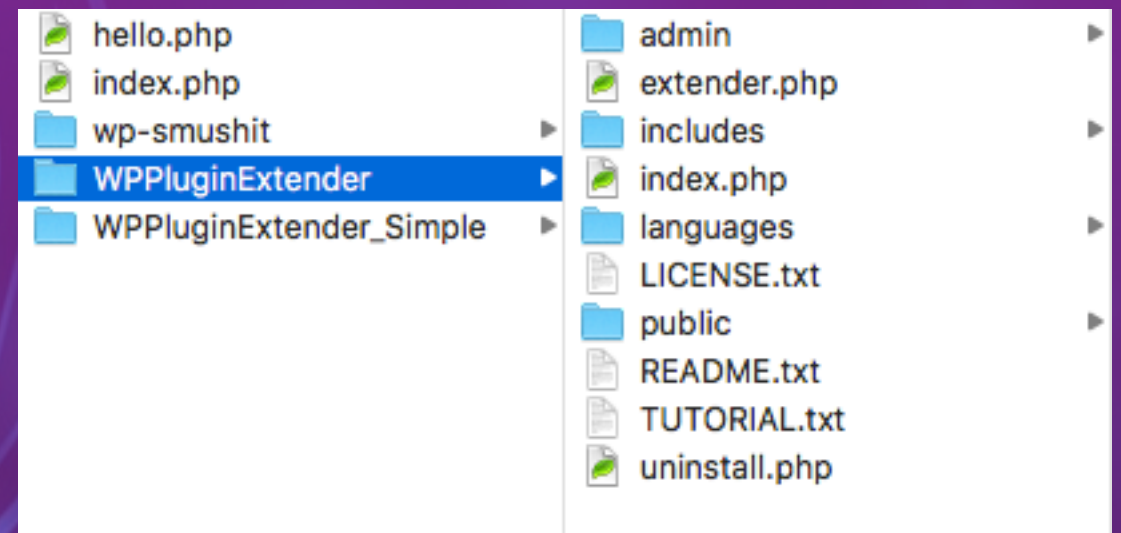
Two Ways to create a WordPress Plugin:

Object-Oriented Programming or Functional Programming

Comment Blocks

Folder Structure

```
3  /**
4   * @link      russellbenzing.com
5   * @since     1.0.0
6   * @package   Extender
7   *
8   * @wordpress-plugin
9   * Plugin Name:   Extender
10  * Plugin URI:    http://github.com/rbenzing/WPPluginExtender
11  * Description:   This is plugin used in a workshop on WordPress plugin development.
12  * Version:      1.0.0
13  * Author:       Russell Benzing
14  * Author URI:    russellbenzing.com
15  * License:      GPL-2.0+
16  * License URI:   http://www.gnu.org/licenses/gpl-2.0.txt
17  * Text Domain:   extender
18  * Domain Path:   /languages
19  */
20
```



DOCUMENT: Functions, Classes, Class members (including properties and constants), Requires and includes, Hooks (actions & filters) and Constants.

```
1  /**
2   * Summary.
3   *
4   * Description.
5   *
6   * @since x.x.x
7   * @deprecated x.x.x Use new_function_name()
8   * @see Function/method/class relied on
9   * @link URL
10  * @global type $varname Description.
11  * @global type $varname Description.
12  *
13  * @param type $var Description.
14  * @param type $var Optional. Description. Default.
15  * @return type Description.
16  */
```

# FUNCTIONS, CONSTANTS AND HOOKS O MY

[developer.wordpress.org/reference/functions](https://developer.wordpress.org/reference/functions/)

[developer.wordpress.org/reference/hooks](https://developer.wordpress.org/reference/hooks/)

[github.com/WordPress/WordPress/blob/master/wp-includes/default-constants.php](https://github.com/WordPress/WordPress/blob/master/wp-includes/default-constants.php)

## Register Plugin Hook Example:

```
1 register_activation_hook( __FILE__, 'pluginprefix_callback_function_to_run' );
2
3 function pluginprefix_callback_function_to_run() {
4     // run code
5 }
```

## Function & Class Checks

```
1 if(!function_exists('my_custom_function_name')){
2     // run code
3 }
4 if(!class_exists('ClassName')){
5     // run code
6 }
7 if(!method_exists($classvar, 'MethodName')) {
8     // run code
9 }
```

## Common Constants

if( !ABSPATH ) - can be used to add to the top of every plugin file to check for proper instantiation.

If( DOING\_AJAX ) - run some code only when a WordPress ajax request is running.

If( DOING\_CRON ) - run some code only when a WordPress cron job is running.



# DON'T FORGET SECURITY!

WordPress has built-in function for you to keep your plugin secure and safe from web threats.

```
sanitize_email()
sanitize_file_name()
sanitize_html_class()
sanitize_key()
sanitize_meta()
sanitize_mime_type()
sanitize_option()
sanitize_sql_orderby()
sanitize_text_field()
sanitize_textarea_field()
sanitize_title()
sanitize_title_for_query()
sanitize_title_with_dashes()
sanitize_user()
```

**Check User Capabilities** - `if ( current_user_can('manage_options') ) { }`

**Validation** - `if ( strlen( intval( $unsafe_zipcode ) ) > 5 ) { }`

**Sanitize input** - Always sanitize input and out.  
← see functions to the left

**Sanitize output** - `esc_attr()`, `esc_html()`, `esc_url()`, `esc_js()`, `esc_textarea()`

**Nonces** - `wp_create_nonce()`, `wp_verify_nonce()` and more.

A nonce is a "multi digit numbers used once" to help protect URLs and forms from certain types of misuse, malicious or otherwise.

[codex.wordpress.org/Validating\\_Sanitizing\\_and\\_Escaping\\_User\\_Data](https://codex.wordpress.org/Validating_Sanitizing_and_Escaping_User_Data)

# PUBLISHING YOUR PLUGIN

WordPress won't accept plugins that do 'nothing,' are illegal, or encourage bad behavior. Don't be spammy!

## 5 STEPS TO SUBMISSION:

1. Sign up for an account on WordPress.org.
2. Package up your plugin and Submit your plugin for review.
3. After your plugin is manually reviewed, it will either be approved or you will be informed to make corrections.
4. Once approved, you'll be given access to a Subversion Repository where you'll store your plugin.
5. Shortly after you upload your plugin (with README.txt file) to your svn, it will be automatically displayed in the plugins directory.



# SVN CHEATS & THINGS TO LOOK OUT FOR

SVN login is your WordPress username (not email) and password

## SVN Checkout & Add Code:

```
mkdir my-plugin-dir
svn co https://plugins.svn.wordpress.org/my-plugin my-plugin-dir
cd my-plugin-dir/
svn add trunk/*
svn ci -m "first commit" --username user --password pass123
```

## SVN Update Existing Code:

```
cd my-plugin-dir/
svn up
svn stat    ** see whats changed
svn diff.   ** see whats changed in code
svn ci -m "update" --username user --password pass123
```

## Plugin denials do and will happen, look out for:

1. Not including a readme.txt file when acting as a service
2. Not testing the plugin with WP\_DEBUG
3. Including custom versions of packaged JavaScript libraries
4. Calling external files unnecessarily
5. "Powered By" links
6. Phoning home
7. Non-Semantic Versioning

LET'S GET STARTED



**WORDPRESS  
DEVELOPMENT**

**IS EASY. IT'S LIKE**

**RIDING A BIKE**

**EXCEPT THE BIKE IS ON FIRE,  
YOU ARE ON FIRE,  
EVERYTHING IS ON FIRE  
AND YOU ARE IN HELL.**





# RECAP ON WHAT WE LEARNED

- Different Plugin Types - OOP or Functional
- PHP Scripting Conventions and Compliance - PSR-1 & PSR-2
- WordPress Plugin Requirements - Header Comments
- Useful WordPress Functions, Hooks and Constants
- Submitting your plugin to the Plugin Directory using SVN
- Helpful Links and Resources



# THANK YOU



[github.com/rbenzing/WPPluginExtender\\_Simple](https://github.com/rbenzing/WPPluginExtender_Simple)