Best Practices For Building WordPress Plugins

Peter Baylies
(**@pbaylies**)

Rivers Agency

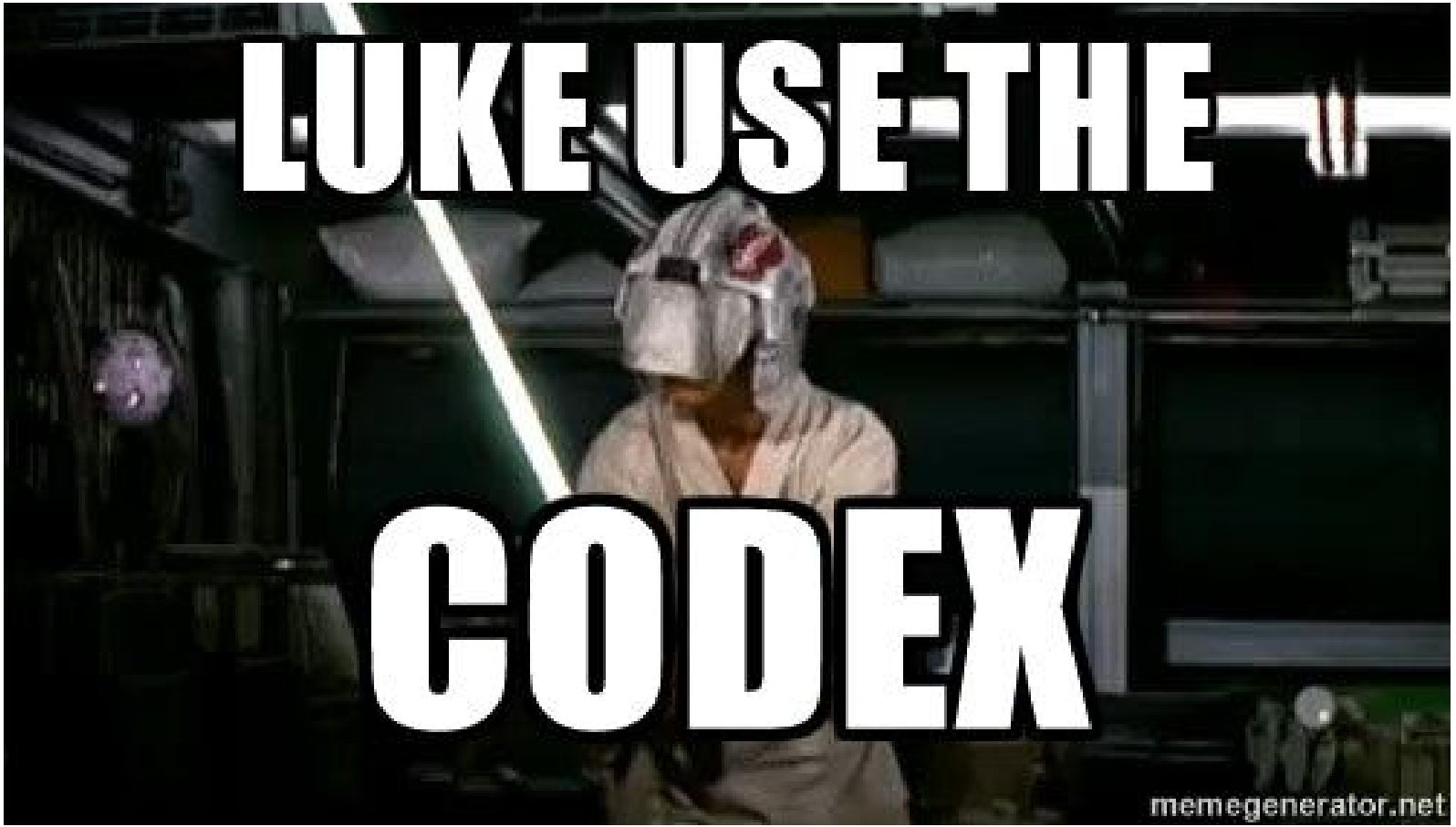
WordCamp Wilmington
2016

RIVERS

Why Bother Doing All This?

- You'll thank yourself for it.
- Other people will thank you for it.
- I'll thank you for it.
- "Always code as if the guy who ends up maintaining your code will be a violent psychopath who knows where you live. Code for readability." - John F. Woods,*comp.lang.c++*,1991

RIVERS



RIVERS

Plugin Handbook

“The Plugin Developer Handbook is a resource for all things WordPress plugins. Whether you’re new to WordPress plugin development, or you’re an experienced plugin developer, you should be able to find the answer to many of your plugin-related questions right here.” - https://developer.wordpress.org/plugins/

RIVERS

Coding Standards

```
function register()
{
    if (!empty($_POST)) {
        $msg = '';
        if ($_POST['user_name']) {
            if ($_POST['user_password_new']) {
                if ($_POST['user_password_new'] === $_POST['user_password_repeat']) {
                    if (strlen($_POST['user_password_new']) > 5) {
                        if (strlen($_POST['user_name']) < 65 && strlen($_POST['user_name']) > 1) {
                            if (preg_match('/^[a-z\d]{2,64}$/i', $_POST['user_name'])) {
                                $user = read_user($_POST['user_name']);
                                if (!isset($user['user_name'])) {
                                    if ($_POST['user_email']) {
                                        if (strlen($_POST['user_email']) < 65) {
                                            if (filter_var($_POST['user_email'], FILTER_VALIDATE_EMAIL)) {
                                                create_user();
                                                $_SESSION['msg'] = 'You are now registered so please login';
                                                header('Location: ' . $_SERVER['PHP_SELF']);
                                                exit();
                                            } else $msg = 'You must provide a valid email address';
                                        } else $msg = 'Email must be less than 64 characters';
                                    } else $msg = 'Email cannot be empty';
                                } else $msg = 'Username already exists';
                            } else $msg = 'Username must be only a-z, A-Z, 0-9';
                        } else $msg = 'Username must be between 2 and 64 characters';
                    } else $msg = 'Password must be at least 6 characters';
                } else $msg = 'Passwords do not match';
            } else $msg = 'Empty Password';
        } else $msg = 'Empty Username';
        $_SESSION['msg'] = $msg;
    }
    return register_form();
}
```

RIVERS

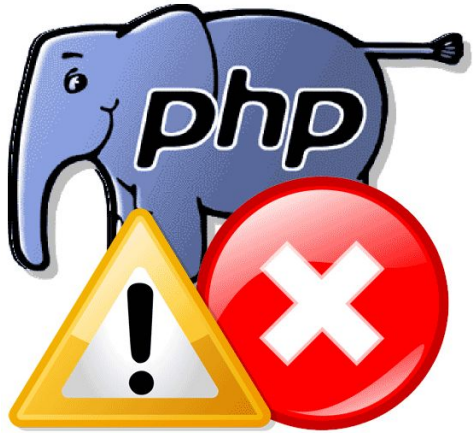
Coding Standards

- WordPress has coding standards, read them and try to use them: https://codex.wordpress.org/WordPress_Coding_Standards
- Space your code out for readability.
- Use underscores in functions, not camel case.
- Don’t use PHP short tags.
- Don’t end a file with a closing php tag, please.
- Yoda conditionals -- use them, you can.

RIVERS

Naming Conventions and Sanity Checks

- Prefix all of your functions and classes, to prevent collisions in the global namespace; don’t use common or short names.
- Check first if things exist:
- Variables - **isset**() or !**empty**()
- Functions - **function_exists**()
- Classes - **class_exists**()
- Constants - **defined**()
- Initialize your variables first.
- Be aware of which features your plugin can use in its environment.
- consider using classes (or namespaces) as organizational tools.



RIVERS

File Organization

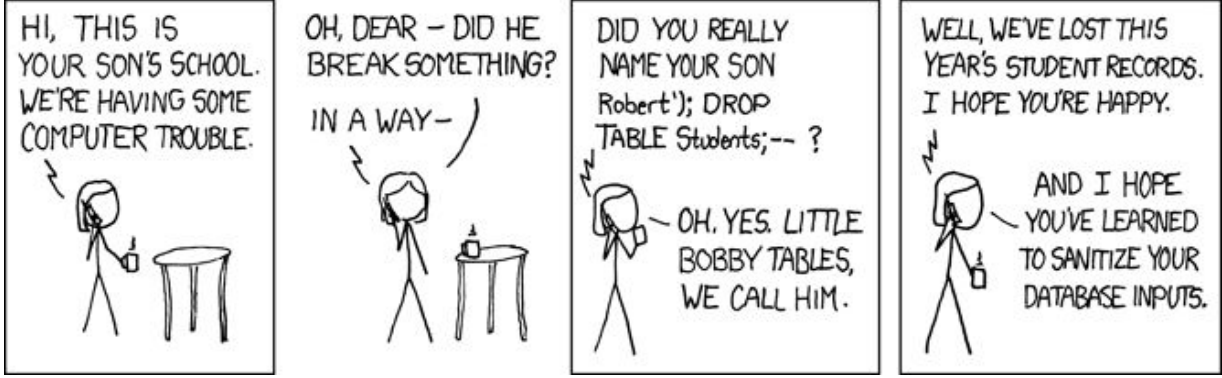
- The convention is to name your plugin the same as your plugin slug; so,

pb-example-plugin/pb-example-plugin.php

- It is recommended to use folders for organization, as needed, so:
- Included files - includes/
- Images - images/
- Translations - languages/
- CSS - css/
- Javascript - js/
- etc.

RIVERS

Data Sanitization:



- You need to do this.
- Never trust data that any user can give you.
- Also don't trust data from the database.
- Examples:
- **sanitize_email**($email) -- sanitizes email addresses.
- **sanitize_title_with_dashes**($title) -- how slugs are born!
- **esc_url**($url) -- escapes URLs for viewing.
- **esc_attr**($attr) -- escapes attributes in HTML.
- **wp_kses_post**($html) -- KSES Strips Evil Scripts!
- **$wpdb->prepare**($format, $values...) -- sanitize data for MySQL

RIVERS

Nonces, Security

- Nonce stands for ‘number used once’.
- (In WordPress they are used more than once, but do change every 12 hours.)
- This is used to validate where a user came from when submitting requests.
- **wp_create_nonce**($action) - generates and returns a nonce.
- **wp_verify_nonce**($nonce, $action) - validates a nonce.
- Trust, but verify.

RIVERS

Conditional Loading

- Use **is_admin**() to separate backend Dashboard code from frontend code.

- Remember, is_admin() tells you when code is running in the WordPress Dashboard, not if a user is an administrator. Because, WordPress.

- use **DOING_AJAX** to manage your AJAX backend code -- this constant is defined only when AJAX is being called.

RIVERS

Hooks, aka the WordPress Plugin API

- https://developer.wordpress.org/plugins/hooks/
- Use Hooks -- your own and other people's.
- keep your hooks accessible - make sure people can **unregister** your hooks as well! - https://developer.wordpress.org/plugins/hooks/advanced-topics/
- This is not helpful to others:

```
add_action( 'the_content', function(){ echo "Hello!" } );
```

- Actions and filters, not quite the same in practice:
- Filters should filter data, a filter accepts and returns a value.
- Actions execute code, actions don't return values.

RIVERS

Included Libraries

- Whatever you do, don't include your own version of jQuery.
- https://developer.wordpress.org/themes/basics/including-css-javascript/#default-scripts-included-and-registered-by-wordpress
- Check this list first to see which Javascript libraries WordPress already has available!
- This includes all of jQuery UI; underscores; backbone; and a number of others.

RIVERS

WordPress APIs

- Use them! Use them all?
- https://codex.wordpress.org/WordPress_APIs
- **Examples**:
- Use the **HTTP API** instead of using Curl directly.
- Cache data in a **transient** instead of in a cookie or a session.
- Use $**wpdb->query**() instead of using PHP's MySQL extensions directly.
- Use **WP_Filesystem** instead of using PHP's file access functions directly.
- Use **rewrite rules** in WordPress to handle custom pages or archives.
- Use **metadata** to store extra information for posts, users, taxonomies, etc.
- Use **shortcodes** and **widgets** to make your plugins more client-friendly.

RIVERS

Plugin Frameworks

- This is stylistic, you can use MVC or any other paradigm within your plugins; this is not required.

- There are quite a few boilerplate / starter frameworks for plugins.

- They are fine to look at or use, but you might not need them, especially not for a simple plugin.

- Still, it doesn’t hurt to see how other people implement their plugins.

- Example: https://github.com/DevinVinson/WordPress-Plugin-Boilerplate

RIVERS

Custom Post Types

- Be careful in registering your post types.
  Prefix your post type and taxonomy names.
  Decide first if you are going to use an archive page.

- Register your post types first, and then your associated taxonomies.
  Or associate them later, with **`register_taxonomy_for_object_type`**()

- Also consider using a plugin for this, such as SuperCPT or Types.

RIVERS

Ajax

- Ajax uses Javascript to get data from PHP without reloading the current page.
- All of your Ajax needs to go through **wp-admin/admin-ajax.php** - https://codex.wordpress.org/AJAX_in_Plugins
- Don't run PHP files directly in your plugin, please. Disallow this if possible.
- Passing PHP data into Javascript - use **wp_localize_script**() for this - https://developer.wordpress.org/plugins/javascript/enqueuing/

RIVERS

Internationalization and Localization

- Use translation functions, so that people can translate your plugins.
- `echo __( 'Translate Me!', 'pb-example-plugin' );`
- Remember to use only a string for your text domain, variables will not work.
- Add placeholders for variables using **sprintf**()
- **sprintf**(__('Hello, %s!', 'pb-example-plugin'), $name);

RIVERS

WP.org plugin guidelines

- https://developer.wordpress.org/plugins/wordpress-org/plugin-developer-faq/
- Test your plugin with **WP_DEBUG** on, please.
- Plugins should be self-contained, should not track users without consent.
- Plugins should be useful, helpful, and in general should just work.
- Suggest using Debug Bar and Debug Bar Extender and/or Query Monitor
- https://developer.wordpress.org/plugins/developer-tools/debug-bar-and-add-ons/
- https://developer.wordpress.org/plugins/developer-tools/helper-plugins/

RIVERS

Questions?

- Ask away!

RIVERS

Thank you!

Slides are up at goo.gl/LrYdTF

Search Twitter for **#wc_wilmington**

You can follow me on Twitter - @pbaylies

RIVERS