

Common Word- Press Mistakes

[and how to fix them]

Editing Core Files

Not even once.



*"With great power comes great
responsibility
to not edit core files."*

-- Voltaire

Why does it matter?

- Updates will wipe out your changes!
- Easy way to get hacked.
- WordPress provides a safer way to do this!

The Fix

- Use hooks to override core functionality.
- Create child theme or small plugin.

Actions

```
add_action ( 'publish_post', 'email_friends' );
```

Filters

```
add_filter ( 'the_title', 'prepend_title' );
```

Some hooks

- init
- plugins_loaded
- pre_get_posts
- publish_post
- wp_footer
- ... one you need that doesn't exist yet?

Resources

- www.elegantthemes.com/blog/tips-tricks/learning-to-use-wordpress-hooks
- themesandco.com/customizr/wordpress-actions-filters-and-hooks-a-guide-for-non-developers/
- codex.wordpress.org/Plugin_API
- www.smashingmagazine.com/2011/10/07/definitive-guide-wordpress-hooks/

Overloading functions.php

Examples

- Adding shortcodes
- Registering widgets
- Defining custom post types
- Anything that changes the way your site **works**, not just how it looks.

Why does it matter?

What happens if you want to change or update your theme?

The Fix

- Use `functions.php` only for theme or display-related functionality.
- Separate functional changes into a standalone plugin.

Resources

- codex.wordpress.org/Functions_File_Explained
- justintadlock.com/archives/2011/02/02/creating-a-custom-functions-plugin-for-end-users
- codex.wordpress.org/Writing_a_Plugin

Haphazard Scripts / Stylesheets

Why does it matter?

- Multiple versions of scripts, e.g. jQuery
- Dependencies

The fix

- Use WP versions of libraries
- `wp_enqueue_script()`
- `wp_enqueue_stylesheet()`
- hook to `wp_enqueue_scripts` action

Example

```
wp_enqueue_script( 'jquery' );
```

Example

```
wp_enqueue_script( 'my-cool-script',  
                  '/path/to/my-cool-script.js',  
                  array( 'jquery' )  
                  );
```

Resources

- codex.wordpress.org/Function_Reference/wp_enqueue_script

No version control

Why does it matter?

- Super easy to lose all your work!
- Mistakes are very costly!

Version control = backups

Version control = checkpoints

Version control = not scary!

Once you have a checkpoint:

- "What exactly changed?" (`git diff`)
- "When did this bug start happening?" (`git bisect`)
- Someone else changed the file I'm working on!
(`git merge`)
- "Everything is ruined, let's go back to how it was!"
(`git reset`)

git quickstart

- `git init`: start a new repository
- `git status`: show the current state of affairs
- `git add <file>`: start tracking a file
- `git commit`: save a checkpoint
- `git diff`: what's different now?

Resources

- GUI clients: git-scm.com/downloads/guis
- Interactive lesson! try.github.io
- classic.scottr.org/presentations/git-in-5-minutes/

Inconsistent code

Why does it matter?

- Hard to maintain
- Easier to introduce bugs
- Harder to find bugs
- Hard to collaborate
- Use your neurons for better things

Use full PHP tags

```
<?php  
do_some_stuff();
```

Descriptive function and variable names

```
$current_post_title = get_the_title();
```

Let code breathe

```
if ( $post->ID == 1 ) {  
    do_something( $post );  
}
```

Indent consistently

```
$r $milk = get_the_milk();  
i: if ( thirsty() {  
    if ( ! is_sour( $milk ) ) {  
        drink_milk();  
    }  
    else {  
e:        throw_the_milk_out();  
    }  
}  
}
```


Yoda conditions

```
if ( 1 == $id )  
    $id = 1 // valid (assignment)  
1 == $id // valid (comparison)  
1 = $id // invalid (syntax error)
```

Comment as necessary

```
/**
 * Should we show the PostPost for the current "Edit Post" context?
 */
function _should_show_postpost( $url, $post_id ) {

    if (
        false === strpos( $url, 'message=6' ) // only when the post was just published
    ||
        ! in_array( get_post_type( $post_id ), array( 'post' ) ) // only for posts
    ||
        ! in_array( get_post_status( $post_id ), array( 'publish' ) ) // only published posts
    ) {
        return false;
    }
}
```

Resources

codex.wordpress.org/WordPress_Coding_Standards

Failing at accessibility

Why does it matter?

- Those who cannot see or use a mouse.
- Deaf users whose first language is sign language.
- Visitors whose primary language is not your language.
- People who use special assistive software or hardware to access the Web.
- People who are colour blind or can't see low colour contrast.

Tips

- Use headings correctly
- Descriptive link text
- Navigational elements should be button, input or a elements
- Keyboard navigation
- Color contrast
- Forms should have labels
- Describe images

Resources

- make.wordpress.org/themes/handbook/guidelines/accessibility/
- <https://getflywheel.com/layout/accessibility-in-wordpress/>
- wptavern.com/wordpress-themes-suck-at-accessibility-its-time-to-fix-it

General Best Practices

- Use the WordPress APIs (filters, actions, options, queries)
- Follow coding style
- Use version control

Thanks!

codebykat.com/ABQWP/
kat@codebykat.com
[@wirehead2501](#)