WP COS Meetup

An Intro to WordPress Hooks

Chris Mospaw – February 13, 2019

An Intro to WordPress Hooks

Who is Chris Mospaw?

- WordPress user/developer since 2004 (WP 1.2!)
- Former *Senior Lead Developer* at Crowd Favorite, Denver CO
- PHP and web developer since 1996
- Currently *Application Developer & Architect* for Wowza Media Systems, Golden CO

An Intro To WordPress Hooks

Pimpin' ain't easy, but hooking is in WordPress…

What is a Hook?

Drunk Octopus wants to fight.
If only he knew about WP hooks...

An Intro To WordPress Hooks

Hooks in a nutshell...

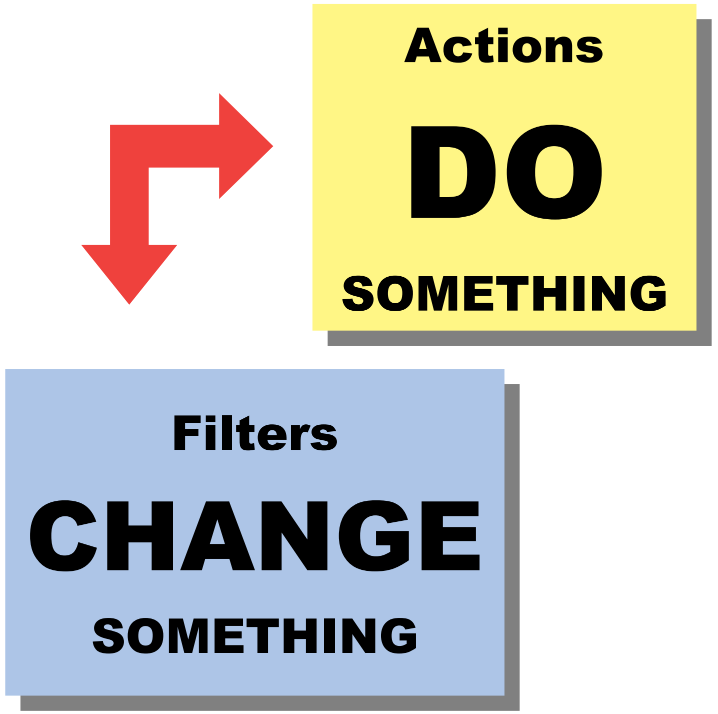
- Hooks are a way to “hook into” WordPress code, be it part of Core, a Theme, or a Plugin
- Allow extending WordPress functionality without changing Core code
- A way of running your code at a specific time in the WordPress life cycle
- Can be added anywhere in your own code, allowing it to be extended as well
- Hooks define part of the Plugin API
  https://codex.wordpress.org/Plugin_API

An Intro To WordPress Hooks

There are two types of hooks in WordPress

- Actions – perform a specific series of things, **with no return value**
  - May or may not pass parameters

- Filters – receive passed data and take action on it, **requiring a returned value**
  - Must pass at least one parameter



An Intro To WordPress Hooks

Much Ado about Actions

Actions

DO

SOMETHING

Actions hooks are defined with `do_action()`. A list of Arguments, *preferably with meaningful names*, is given. There can be zero or more arguments, which are passed via calls made with `add_action()`, shown on the next slide.

```
do_action(
  $hook_name,        // The hook name – used in add_action()
  [$arg1,]           // First argument (if needed)
  [$arg2,] ...       // Second argument, and so on...
);
```

An Intro To WordPress Hooks

Actions in Action

Actions

DO

SOMETHING

Hooking into an action is easy. Simply call `add_action()` with the correct parameters, and the hook is set. Your added function will run any time `do_action` with the same hook name is called.

```
add_action(
  $hook_name,         // Hook names are defined in do_action()
  $function_to_add,   // The function you want called
  $priority,          // Order of execution, lower first
  $accepted_args      // Number of arguments to pass, must
);                    // agree with # of args in do_action()
```

An Intro To WordPress Hooks

Putting them together

Actions

DO

SOMETHING

Somewhere in code...

```
do_action( ‘my_plugin_hook’, $post );
```

Somewhere else in code...

```
add_action( ‘my_plugin_hook’, ‘my_action_function’, 10, 1 );
```

This runs `‘my_action_function()’` every time the `‘my_plugin_hook’` action is triggered, passing the value of `$post` as the only parameter.

An Intro To WordPress Hooks

Filters are "applied"

Filters are defined with `apply_filters()` in much the same way that they are via `add_action()` but must assign at least one value, since **filters always return a value.**

```
$value = apply_filters(
  $hook_name,       // The hook name to be used in add_action()
  $value,           // The value being filtered, will be returned
  [$arg1,]          // First argument (if needed)
  [$arg2,] ...      // Second argument, and so on...
);
```

Filters

CHANGE

SOMETHING

An Intro To WordPress Hooks

Adding filters for application

Filters

CHANGE

SOMETHING

The hook to filter a value is added via `add_filter()` in much the same way that `add_action()` works for actions. The parameters are the same.

```
add_filter(
  $hook_name,           // Hook names are defined in apply_filters()
  $function_to_add,     // The function you want called
  $priority,            // Order of execution, lower first
  $accepted_args        // Argument count. Must be 1 or more.
);                      // Must match # of args in apply_filters()
```

An Intro To WordPress Hooks

Filters are applied

Filters

CHANGE

SOMETHING

Somewhere in code...

```
$title = apply_filters( ‘my_filter’, $title );
```

Somewhere else in code...

```
add_filter( ‘my_filter’, ‘my_filter_function’, 10, 1 );
```

This runs `‘my_filter_function’` every time the `‘my_theme_hook’` action is triggered, passing the value of `$title` as the only parameter. `‘my_filter_function’` **must** return a new value for `$title` that gets assigned when `apply_filters()` runs.

An Intro To WordPress Hooks

What about shortcodes?

Filters

CHANGE

SOMETHING

From Otto (aka Samuel Wood):

"Shortcodes are a type of filter. They take in content from the shortcode, they return replacement content of some sort. They are filters, by definition. Always, always keep that in mind."

http://ottopress.com/2011/actions-and-filters-are-not-the-same-thing/

An Intro To WordPress Hooks

A quick note about keeping your priorities straight...

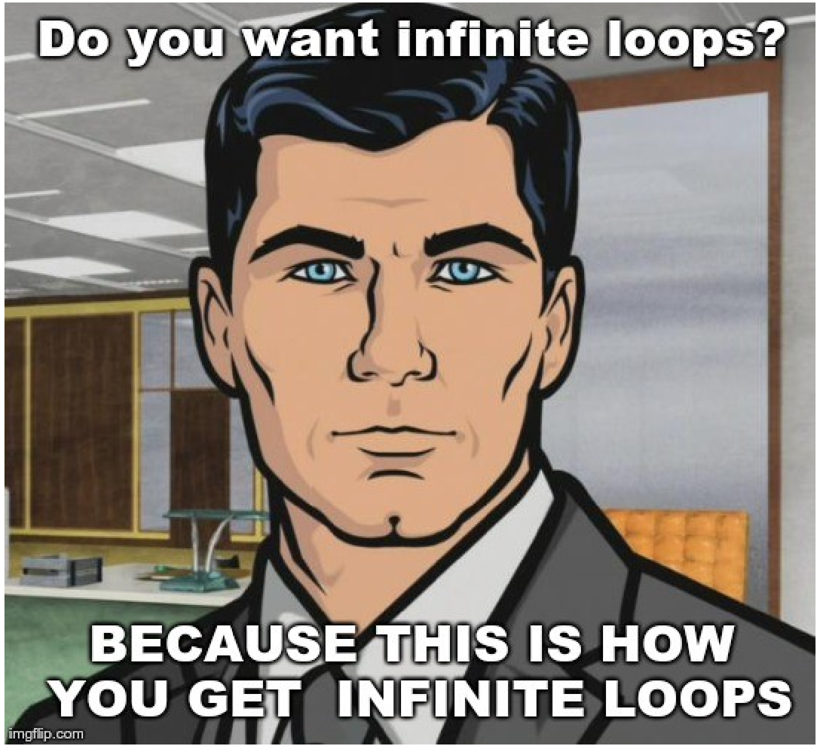
The priority number sent via `add_action()` and `add_filter()` is...

“Used to specify the order in which the functions associated with a particular action are executed. Lower numbers correspond with earlier execution, and functions with the same priority are executed in the order in which they were added to the action.”

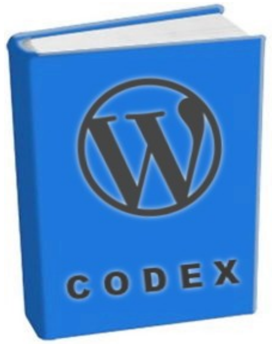
An Intro To WordPress Hooks

Hook removal and why

- Hooks that call hooks that refer to themselves should remove themselves to prevent infinite loops
  - They can be re-added at the end of the function
- Sometimes hooks can interfere with other hooks and should be removed and re-added


Do you want infinite loops?
BECAUSE THIS IS HOW YOU GET INFINITE LOOPS
imgflip.com

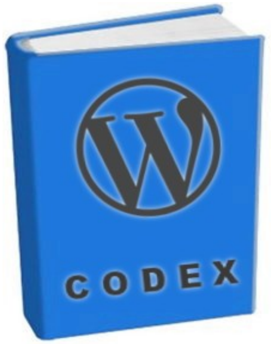
Action and filter removal reference



- **remove_filter()**
  https://codex.wordpress.org/Function_Reference/remove_filter
- **remove_all_filters()**
  https://codex.wordpress.org/Function_Reference/remove_all_filters
- **remove_action()**
  https://codex.wordpress.org/Function_Reference/remove_action
- **remove_all_actions()**
  https://codex.wordpress.org/Function_Reference/remove_all_actions

An Intro To WordPress Hooks

Beyond Actions and Filters: Special Hooks

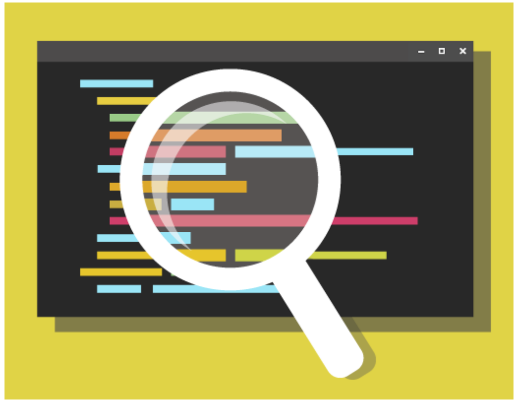


When a plugin is activated, deactivate, or uninstalled, there are special hooks that get called.

- **register_activation_hook**
  https://codex.wordpress.org/Function_Reference/register_activation_hook
- **register_deactivation_hook**
  https://codex.wordpress.org/Function_Reference/register_deactivation_hook
- **register_uninstall_hook**
  https://developer.wordpress.org/reference/functions/register_uninstall_hook/

An Intro To WordPress Hooks

Finding hooks shouldn’t FASE you



- Searching through code for all the various calls to action and filter definitions and calls is tedious.
- Tools like phpDocumentor *cannot* help.
- **WP-FASE** can help. It’s a command-line tool that finds and documents hook definitions and calls.
  - Source: https://github.com/mospaw/wp-fase
  - Presentation for “hookers”: https://mospaw.com/wp-fase-reveal/

WordPress Meetup

Questions or Comments?