Filters Are For More Than Making Coffee

Who am I?

I write code.
I write music.

Gregory Cornelius
Web Developer III, Interactive Design
Boston University

Twitter: @gcorne
Web: http://gregorycornelius.com
Blog: http://bitswapping.com

- ▸ Powered by WordPress Multisite (WPMU 2.9.2, soon to be 3.1).

- ▸ Hosts over 400 university websites.

- ▸ Over 55 themes.

- ▸ Comprises of over 50 custom and 3rd-party plugins including custom plugins to handle site organization, user management, access control list, taxonomy, and specific post meta data.

BU Showcase

- http://www.bu.edu/abroad/

- http://www.bu.edu/academics/

- http://www.bu.edu/met/

- http://www.bu.edu/com/

- http://www.bu.edu/cghd/

- http://www.bu.edu/students/
  (First BU CMS site with a mobile version.)

Two rules.

①

Don't Fight WordPress.

Read the source.

②

Learn the stack.

You don't want to be just a WordPress guy/gal.

What is an action/filter?

The action/filter **callback** system is the primary API through which WordPress developers do work.

"By making a component callback aware, you give others the power to extend our code in context you don't yet know about."

— Matt Zandstra*, PHP Objects, Patterns, and Practice*

Simple Callback

```php
<?php

main();

function main() {
    call_user_func('callback', "Hello, World.");
}

function callback($text) {
    echo $text . PHP_EOL;
}


?>
```

Filters
add_filter
apply_filters
apply_filters_ref_array
current_filter
has_filter
remove_filter
remove_all_filters

Actions
add_action
do_action
do_action_ref_array
did_action
has_action
remove_action
remove_all_actions

See: `wp-includes/plugin.php`

An action is a callback that is called at a particular point in the life cycle of a request.

```
do_action('handle', [$arg1, $arg2, ...])

add_action('handle', $callback, [int $priority = 10, int $num_args = 1])

remove_action('handle', $callback, [int $priority = 10, int $num_args = 1])
```

Code.

A filter is a callback that processes a piece of data.

```
add_filter('handle', $callback, [int $priority = 10, in $num_of_args = 1])

apply_filters( handle, data1, [$context1, $context2, …])
```

Each filter is one of a possible chain of filters. The output of one filter becomes the input of another.

▸ Use appropriate names for your callbacks.

▸ Register your actions and filters close to their callbacks.

More code.

Core Actions and Filters

? Quick Tip

Use Subversion

Develop against a working copy of WordPress core.

▶ Easily revert debugging code.

▶ Review core changes by browsing history.

▶ Quickly switch between different versions for testing.

▶ Makes contributing/reviewing patches to core easy.

Demo.

Give others a chance to play.

Incorporating actions/filters into plugins and themes give developers downstream the ability to modify behavior.

- And -

Unlike attempting to call an undefined function, when WordPress executes "do_action," if there is no action registered, there are no ill effects.
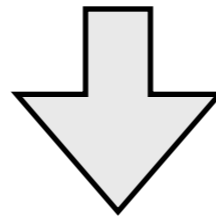
3rd-party code initialization sequence

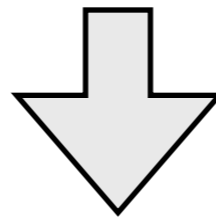**Non-core code is loaded in the following order:**

1. mu-plugins.

2. network-wide activated plugins.

3. site activated plugins.

4. current theme functions.php.

5. parent theme functions.php (if the current theme has a parent).

WordPress Core

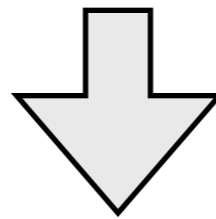⬇

Plugin

⬇

Theme

⬇

Child Theme

▸ Easy to use.

▸ Simple.

▸ Consistent. Once your plugin API has users, you can't change the API at will.

▸ Good for the rest of your code. Thinking APIs means better code.

See: http://lcsd05.cs.tamu.edu/slides/keynote.pdf

Even More Code.

https://github.com/gcorne/modern-posts-widget

Resources

- wp-includes/plugin.php

- wp-includes/default-filters.php

- http://adambrown.info/p/wp_hooks

- http://codex.wordpress.org/Plugin_API/Action_Reference

- "Designing a good API" http://lcsd05.cs.tamu.edu/slides/keynote.pdf

Want to work for BU?

We are looking for a good developer or two.