

Becoming a better WordPress Developer

Joey Kudish

@jkudish

slides.jkudish.com

why am I here?

- developing with WordPress since 2007
- previously: high scale WP client sites + custom plugins
- currently: Code Wrangler for WordPress.com / Automattic
- I want to share my experience and knowledge

what is WordPress development?

themes, plugins, html,
javascript, css, php, html5,
css3, mysql, code, blah,
blah, blah...

all these things come
together

WordPress core



**Every time you hack core
God kills a kitten**

Please, think of the kittens!

contribute to core

- contributing makes you feel good, and helps others
- submit a ticket / bug report
- write a patch
- answer support forum questions
- help document something


Ticket #20000 (closed enhancement: fixed)

←!

We have alot of tickets

Opened [6 months](#) ago

Last modified [6 months](#) ago

Reported by:  [nacin](#)

Owned by:

Priority: [normal](#)

Milestone: [3-4](#)

Component: [General](#)

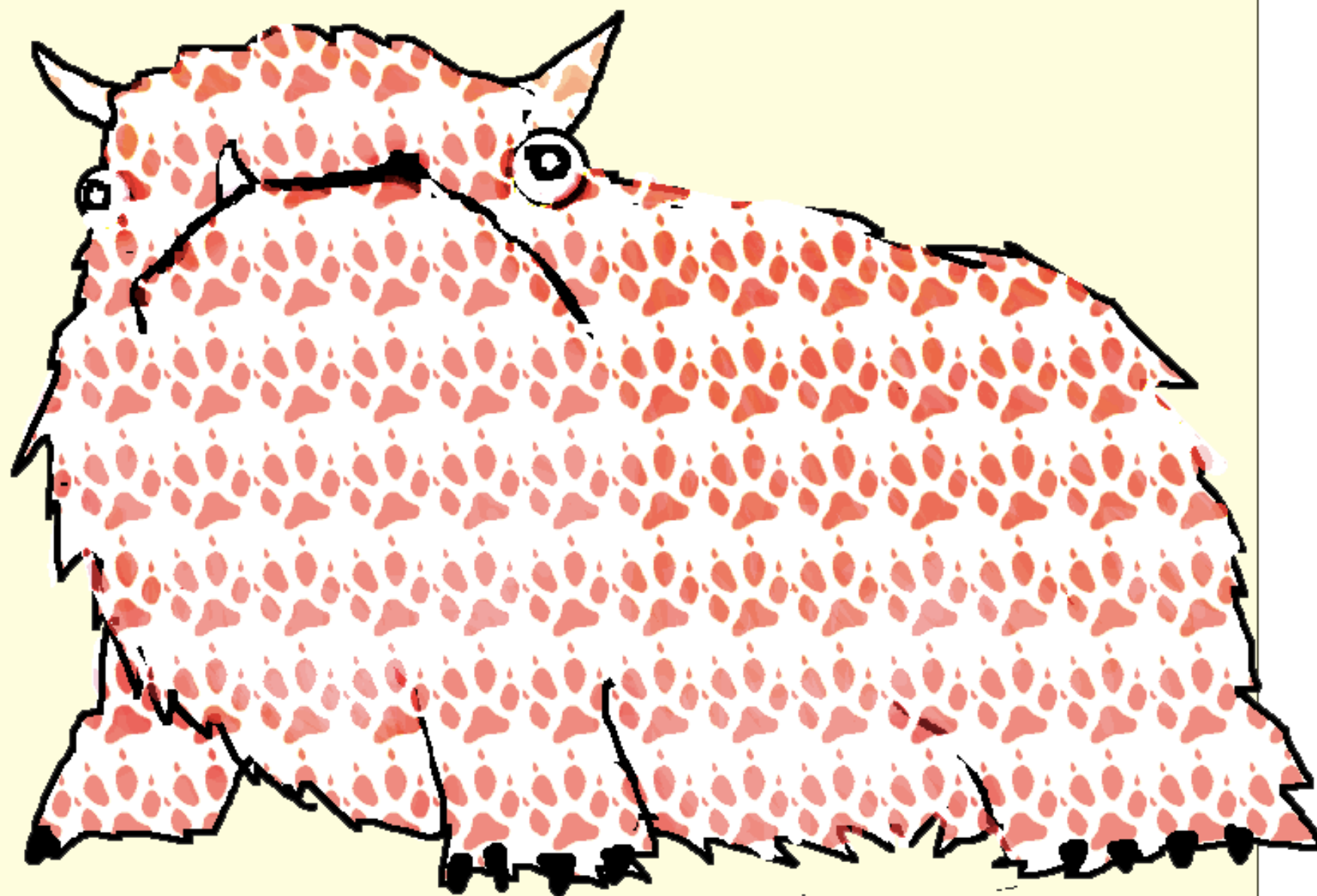
Version:

Severity: [normal](#)

Keywords:

Cc:

Description (last modified by [nacin](#)) (diff)



[make.wordpress.org/ core/handbook/](https://make.wordpress.org/core/handbook/)

Core Contributor Handbook

Making WordPress

[Handbook Home](#)

[WordPress Development Blog](#)

[Project Organization](#)

[Our Philosophies](#)

CONTENTS

- [Learn About WordPress](#)
 - [Project Organization](#)
 - [Our Philosophies](#)
 - [Codex](#)
 - [WordPress.TV](#)
- [Contributing With Words](#)
 - [Translating WordPress](#)
 - [Files and Tools](#)
 - [GlotPress](#)
 - [Evangelize and Organize](#)
- [Learn About Code](#)
 - [The WordPress Codebase](#)
 - [Following the Development Process](#)
 - [Code Repository \(SVN\)](#)
 - [Researching Code History](#)
 - [The Bug Tracker \(Trac\)](#)
 - [Trac Workflow](#)
 - [Keywords](#)

Core Contributor Handbook

Howdy! This is a rough draft of the handbook for contributing to WordPress.

Work here is still in progress. Indeed, some pages are only outlines, and there are many other topics left to explore. We hope for this guide to come together more fully during the 3.5 development cycle.

If you are a core contributor who wants to help, you can:

- You can leave comments on any of these pages, which can be incorporated into the main text. Particularly helpful when fixing Jorbin's typos.
 - Or, write something on your own blog and suggest we include it here.
 - Or, track down a core developer who can give you access to this site.
-

**use core to your
advantage**

use built-in APIs

- **WP_HTTP:** http://codex.wordpress.org/HTTP_API
- **Filesystem:** http://codex.wordpress.org/Filesystem_API
- **Settings:** http://codex.wordpress.org/Settings_API
- **CPTs/Taxonomies:** http://codex.wordpress.org/Post_Types
- **WPDB:** http://codex.wordpress.org/Class_Reference/wpdb
- **Hooks**
- **And much much more...**

action hooks

Actions allow you to run arbitrary code at a specific point during WordPress code execution

```
do_action( 'wp_head' );
```

```
add_action( 'wp_head', 'jkudish_head' );
```

filter hooks

Filters allow you to modify a variable (before it is output, saved, used, whatever...)

```
apply_filters( 'the_content', $post_content );
```

```
add_filter( 'the_content', 'jkudish_filter' );
```

hooks API FTW

```
do_action( 'the_action',  
$addtl_arg1, $addtl_arg2 [...] );
```

```
$variable =  
apply_filters( 'the_filter',  
$variable, $addtl_arg1,  
$addtl_arg2 [...] );
```

registering hooks

- Actions:

```
add_action( 'the_action',  
  'callback_func', 10, 3 );
```

- Filters:

```
add_filter( 'the_filter',  
  'callback_func', 10, 3 );
```

- Callback function:

```
function callback_func( $value,  
  $addtl_arg1, $addtl_arg2 ... )
```


finding the right hooks

- Search through core
- Plugin API on the Codex: [codex.wordpress.org/
Plugin_API](https://codex.wordpress.org/Plugin_API)
- Adam Brown's Hook Database: [adambrown.info/p/
wp_hooks](https://adambrown.info/p/wp_hooks)
- WP Candy Article on hooks: [wpcandy.com/teaches/
how-to-use-wordpress-hooks](https://wpcandy.com/teaches/how-to-use-wordpress-hooks)

better yet

```
add_action( 'all',  
  'jkudish_all_the_hooks' ) );
```

```
function jkudish_all_the_hooks() {  
    error_log( print_r(  
        current_filter(), true ) );  
}
```

other helpers

`has_action()`;

`has_filter()`;

`did_action()`;

`remove_action()`;

`remove_all_actions()`;

`remove_all_filters()`;

where to put your code

- themes
 - templates for display on the front-end and related helper functions
- plugins
 - code that can be toggled on/off and modifies the behaviour of WordPress in some way or provides additional functionality
- mu-plugins
 - just like plugins but automatically loaded to run on each page load

let core tell you when you are wrong

- WP_DEBUG
- `_doing_it_wrong();`
- unit tests
- watch Mo's presentation :)

code is poetry

make things legible, classy & elegant;
follow the WP coding standards

http://codex.wordpress.org/WordPress_Coding_Standards

<?php vs <?

single quotes unless you
need to evaluate a variable

```
<?php echo 'a great string'; ?>
```

vs

```
<?php
```

```
$dog_name = 'Winston';
```

```
echo "my dog's name is: $dog_name";
```

```
?>
```

naming is important

- not so much:

```
$myGreatVariable = 2;
```

- even worse:

```
$MygreatVariable = myFunction();
```

- just stupid:

```
$mygReAtvariAbLe = FUNcTi0_n();
```

- correct:

```
$my_great_variable = my_function();
```

spacing for legibility

- not so much:

```
foreach($posts as $post)
```

- even worse:

```
for($i<5;$i=0;$i++)
```

- just stupid:

```
if($codeisineLigible)  
{ $youcantReadthis=true; }
```

- correct:

```
foreach ( $posts as $post )
```

brackets, indentation + empty lines

```
if ( is_page() ) {  
    echo 'this is a page';  
} elseif ( is_single() ) {  
    echo 'this is a post';  
    echo '<br>how wonderful!';  
} else {  
    echo 'not a page or post';  
}
```

Yoda conditions

What happens if you accidentally forget a '=' ?

```
if ( $city == 'Montreal' )
```

vs.

```
if ( 'Montreal' == $city )
```

don't get too clever

Clever, but do you know what it does?

```
isset( $var ) || $var = some_function();
```

Easier to read:

```
if ( ! isset( $var ) )  
    $var = some_function();
```

keep it elegant,
simple and avoid
useless comments

What not to do

```
<?php
/* Get all our options from the database */
global $options;
foreach ($options as $value) {
    if (get_settings( $value['id'] ) ===
FALSE) { $$value['id'] = $value['std'];
    } else { $$value['id'] =
get_settings( $value['id'] ); }
}
```

doing it right

```
<?php
class WP_My_House {
    var $bricks = 0;
    function add_brick() {
        if ( ! $this->is_made_of_cement() )
            return false;

        $this->bricks++;
        return true;
    }
    ...
}
$my_house = new WP_My_House();
```

beyond the
standards

**I DON'T ALWAYS INSERT DATA
INTO THE DATABASE**



```
$wpdb->escape();  
$wpdb->prepare();  
    sanitize_*();
```

escape all the things

- `esc_attr()`: escape an html attribute
- `intval()`: make sure the value is an integer
- `wp_kses()`: strip unwanted html tags
- `esc_html()`: encodes html so it can be output as text
- `esc_js()`: encodes strings to be used in javascript
- `esc_url()`: encodes and validates URLs
- Full list at: codex.wordpress.org/Data_Validation or in `wp-includes/formatting.php`

internationalize

```
__( 'Hello World', 'my-plugin' );  
_e( 'Hello World', 'my-plugin' );  
_n( 'Comment', 'Comments', $comment_count,  
    'my-plugin' );  
  
$mood = 'Cruel';  
  
printf( _x( 'Hello %s World', 'intro to the  
world', 'my-plugin' ), $mood );
```

read more: http://codex.wordpress.org/118n_for_WordPress_Developers

allow for others to hook in

```
do_action( 'jkudish_setup_cpt' );
```

```
$sky_color =  
apply_filters( 'jkudish_sky_color', 'blue' );
```

open source

code review

collaborate

WordPress

Q&A

Merci!

slides.jkudish.com