Welcome to WordCamp Toronto

Network: RU-Secure

Username: trsguest2017

Password: RUguest253\$



HTML in Functions Kills Kittens

MVC In WordPress Plugins and Themes

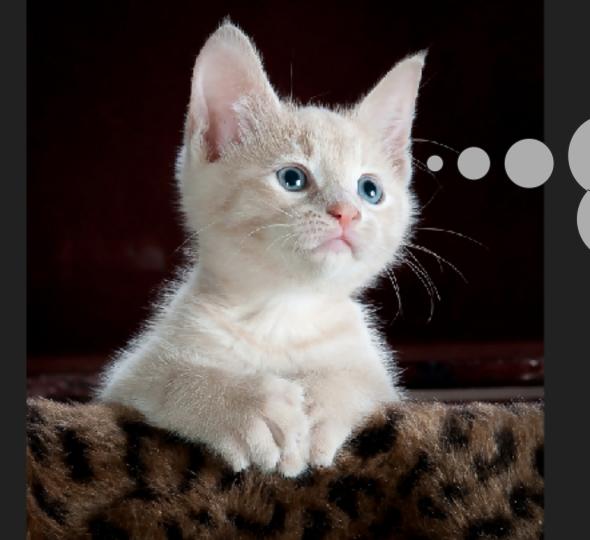
Matt Graham - @themattyg WordCamp Toronto 2017



Resume (Cole's/Cliff's Notes Version)

- Graduated from Durham College Multimedia Design 2009
- •Full Stack PHP/JS Developer for 10 years
- WordPress Developer for 5 years, almost exclusively for 2 years
- Worked with Retail, Ad Agencies, Restaurants, and Education as employee and contractor
- Jack of all digital trades: Audio & Video editor, Live Sound, Photography, Video & Audio Solutions





I DOAN
WANTS 2 DYE

WordPress is not MVC

That's OK. We aren't talking about Core.

```
<form action="<?php echo admin_url( 'admin.php?inport=wordpress&amp;step=2' ); ?>" method="post">
   <?php wp nonce field( |action: 'import-wordpress' ): ?>
   <input type="hidden" name="import_id" value="<?php echo $this->id; ?>" />
<?php if ( ! empty( $this=>authors ) ) : ?>
   <h3><?php _e( text: 'Assign Authors', domain: 'wordpress-importer' ); ?></h3>
   ?php _e( loct 'To make it easier for you to edit and save the imported content, you may want to reassign the author of the imported item to an existing user of
<?php if ( Sthis->allow_create_users() ) : ?>
   <?php printf( _ [ 'If a new user is created by WordPress, a new password will be randomly generated and the new dser$#8217;s role will be set as %s. Manually the</p>
<?php endif; ?>
   sol id="authors">
<?php foreach ( $this⇒authors as $author ) : ?>
       <?php sthis->author_select( $j++, $author ); ?>
<?php endforeach; ?>
   </0l>
<?php endif; ?>
<?php if ( Sthis->allow_fetch_attachments() ) : ?>
   <h3><?php_el look 'Import Attachments', domain 'wordpress-importer' ); ?></h3>
       sinput type="checkbox" value="1" name="fetch_attachments" id="import-attachments" />-
       <label for="import-attachments"><?php _e( text 'Download and import file attachments', domain: 'wordpress-importer' ); ?></label>
   </0>
<?php endif; ?>
```

sp class="submit"><input type="submit" class="button" value="<?php esc_attr_e| text: 'Submit', #domain: 'yopropress-indorter'); ?>" />>/p>

function import_options() {

\$i = 0:

</form>:

```
if (Sposition -- 1680) {
       echo "class=\"repeater_settings field_setting\">
               <label for=\"field repeater start\">Start ";
       gform_tooltip( name: 'form_field_repeater_start');
       echo </label>
              <input type=\"number\" id=\"field_repeater_start\" min=\"i\" value=\"i\" onchange=\"SetFieldProperty['start', this, value];\">
           </11>:
       echo "
              <label for=\"field_repeater_min\">Min ";
       oform tooltip( mamma: 'form field repeater min');
       echo </label>
               singut type=\"number\" id=\"field repeater min\" nin=\"l\" value=\"l\" onchange=\"SetFieldProperty('min', this, value|:\">
           echo "class \"repeater settings field setting\">
              <label for=\"field_repeater_max\">Max ";
       oform tooltip( name: 'form field repeater max');
       echo </label>
              <input type=\"nurber\" id=\"field_repeater_max\" nin=\"1\" onchange=\"SetFieldProperty|'nax', this,value);\">
           public static function gform_appearance_settings($position, $form_id) {
   if (Sposition -- 408) {
       echo "sli class=\"repeater_settings field_setting\">
              <input type=\"checkbox\" id=\"field_repeater_hideLabel\" onchange=\"SetFieldProperty('hideLabel', this.checked);\">
              <label for=\"field_repeater_hideLabel\" class=\"inline\">Hide Label & Description ";
       gform_tooltip( mame: 'form_field_repeater_hideLabel');
       echo </label>
```

public static function gform standard settings(Sposition, Sform id) {

```
public static function gform standard settings(Sposition, Sform id) {
   if (Sposition -- 1680) {
       echo "«li class»\"repeater_settings field_setting\">
              slabel for=\"field repeater start\">5tart ";
       gform_tooltip( name: 'form_field_repeater_start');
       echo </label>
              <input type=\"number\" id=\"field_repeater_start\" min=)</pre>
                                                                     value=\"1 onchange=\"SetFieldProperty('start', this.value):\">
          </11>:
       echo "
              <label for=\"field_repeater_min\">Min ";
       gform tooltip( mammes 'form field repeater min');
       echo </label>
              sinput type=\"number\" id=\"field repeater min\" nin-
                                                                TV" value
                                                                                     /-\"SetFieldProperty('min', this.value);\">
          echo "class \"repeater settings field setting\">
              <label for=\"field_repeater_max\">Max ";
       oform tooltip( name: 'form field repeater max');
       echo </label>
              <input type=\"number\" id=\"field_repeater_max\" r</pre>
                                                             e\"1\\ nchange=\ etFieldProp\ y|'nax', this.value!;\">
          public static function gform_appearance_settings($position, $form)
   if (Sposition -- 408) {
       echo "sli class=\"repeater_settings_field_setting\">
                                                                 <input type=\"checkbox\" id=\"field_repeater_hideLab\</pre>
              <label for=\"field_repeater_hideLabel\" class=\"inline\"</pre>
                                                                   de Label & Desertion "
       gform_tooltip( mame: 'form_field_repeater_hideLabel');
       echo </label>
```

MVC Model, View & Controller

WHY?



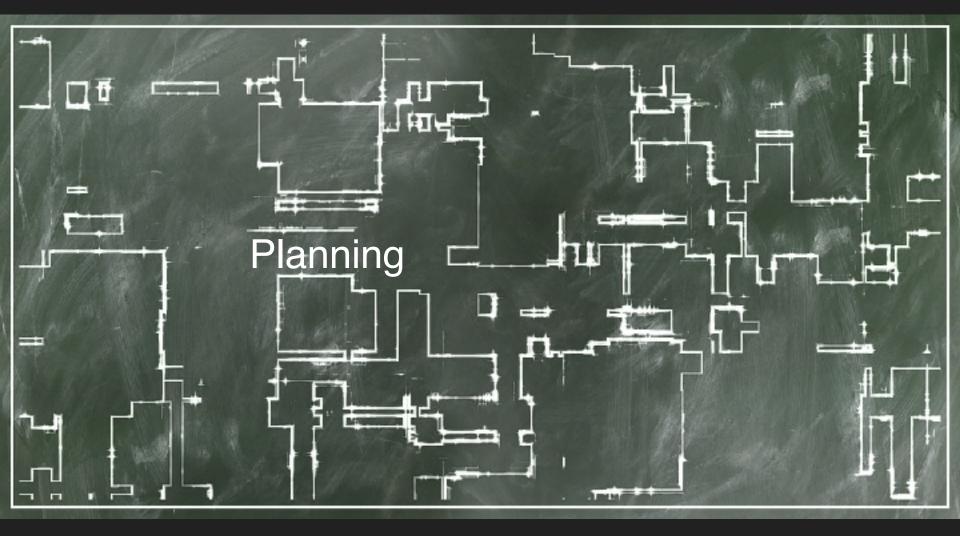


Open Source === Giving





HOW?









MY STRUCTURE for plugins

Controller

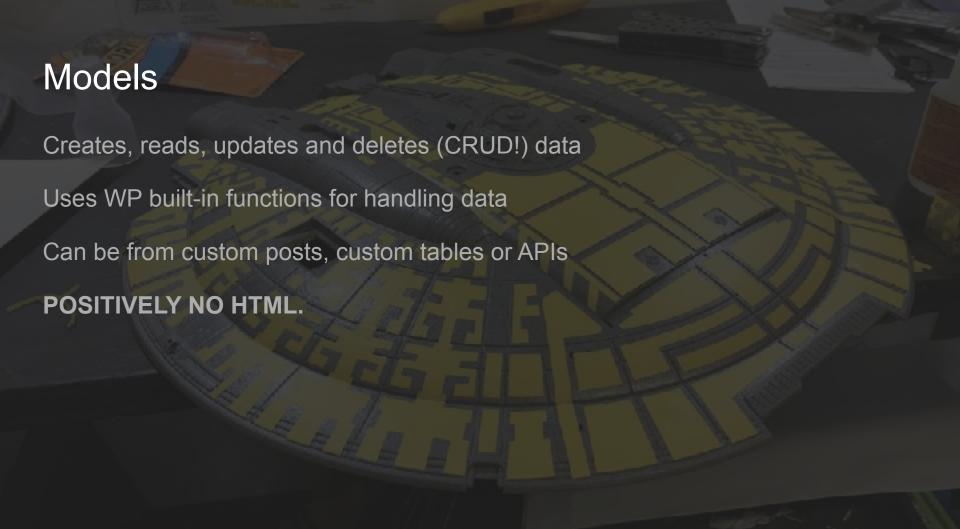
Handles setup and teardown (install, uninstall, startup, finishing tasks)

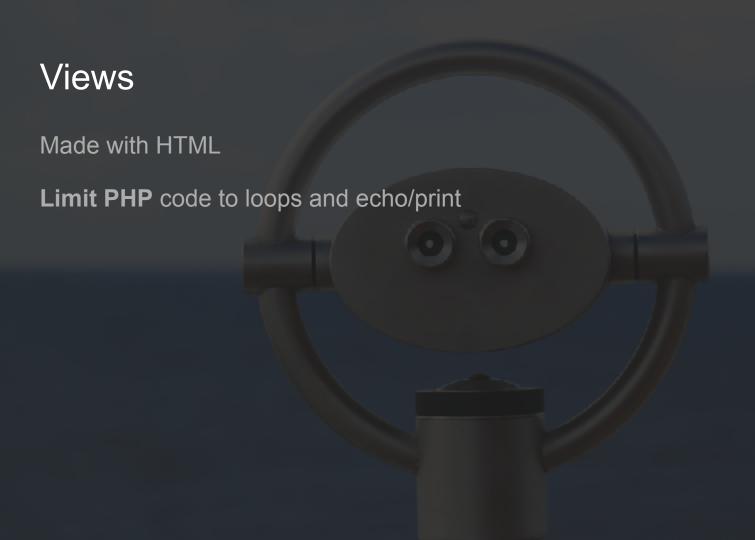
Handles hooks, actions and short codes as required

Gets data from **models** and creates **views** from that data (separate public/admin controllers in separate files)

Passes pertinent information to the views)

NO HTML.





Static files (/static/js, /static/css, /static/img)

Store all the non-processed files

Except JS can be processed, but use localization

Images, CSS, Less/Sass/that other one

ABSOLUTELY NO PHP.

File & folder structure example

- my-awesome-plugin.php
- /classes/
 - class-controller-public.php
 - class-controller-admin.php
 - class-data-model.php
- /views/
 - view-admin-page.php
 - view-public-widget.php
- /static/
 - •js/
 - public.js
 - admin.js
 - css/
 - public.css
 - admin.css
 - img/
 - my-image.jpg

MY STRUCTURE for themes

Controller (functions.php → /classes/class-theme.php)

Handles setup and teardown (install, uninstall, startup, finishing tasks)

Handles hooks and actions custom to your theme

Gets data from models

Let WordPress handle views (standard index.php, page.php, single.php, etc.)

Unless it's admin views, then you'll have to handle it

Easy access for complex data retrieval (via models) for view(s)

NO HTML.



Creates, reads, updates and deletes (CRUD!) data

Uses WP built-in functions for handling data

Can be from custom posts, custom tables or APIs

POSITIVELY NO HTML.

Views

Stick to WordPress' way of handling views

Less confusion for developers new to MVC development

Consistency - be consistent with other theme developers

But don't be afraid to use /views/ for repeat use "partials" that don't fit in the WordPress structure

Static files (/static/js, /static/css, /static/img)

Store all the non-processed files

Except JS can be processed, but use localization

Images, CSS, Less/Sass/that other one

ABSOLUTELY NO PHP.

File & folder structure example

- functions.php
- index.php
- page.php
- archive.php

- /classes/
 - class-theme.php
 - class-controller-admin.php
 - class-data-model.php
- /views/
 - view-admin-page.php
 - view-reusable.php
- /static/
 - js/
 - public.js
 - admin.js
 - •css/
 - public.css
 - admin.css
 - img/
 - my-image.jpg

Theoretical Scenario

Flight & Guest Cross Reference Tracker Plugin

Controllers

Main Controller (/classes/class-controller.php) - instantiates classes and handles plugin startup & pages

Callbacks (/classes/class-callbacks.php) - handle WordPress actions and filters



Flight (/classes/class-flight.php) - extend the WP-HTTP class, call Flight Status API, references and adds methods to the **flight custom post type**

Guest (classes/class-guest.php) - handles data and extra methods for the guest custom post type

Views

single-flight.php - single flight page partial

single-guest.php - single guest page partial

guest-reference.php - repeatable guest display for flight page

flight-reference.php - repeatable flight display for guest page

MVC Plugin Boilerplate

https://github.com/tommcfarlin/WordPress-Widget-Boilerplate

Don't put HTML in a function.

Keep your code clean.

Comment your code.

Be a good open source citizen.



Now go paint a masterpiece.

https://mattgraham.ca/html-in-functions/