MMP350 Class Notes Week 9

This Week's goals

This week we will examine how to create a contact form and send an email in WordPress. We will then review last week's exercises, including creating Child Themes and Custom Templates and shortcodes.

Exercise: Creating a Contact Form

Step

Note: This is an example taken from https://www.sitepoint.com/build-your-own-wordpress-contact-form-plugin-in-5-minutes/

Before preceding, I recommend that you review

https://codex.wordpress.org/Writing_a_Plugin
https://developer.wordpress.org/plugins/the-basics/header-requirements/
https://developer.wordpress.org/plugins/the-basics/including-a-software-license/

Step

Copy the class file `create_contact_form.php` to the folder `wp-content/plugins`

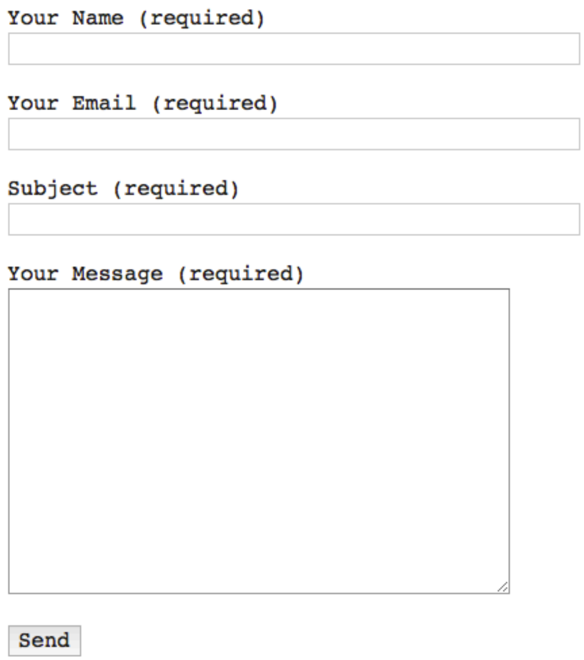
Step

Create a new page and add the following short-code to the content section:

```
[sitepoint_contact_form]
```

Step

View your new page. It should look like the figure below.



Your Name (required)

Your Email (required)

Subject (required)

Your Message (required)

Note that the contact page has been created but will not be fully functional until you add email functionality.

Exercise: Enabling oAuth (open authorization) email functionality using the Gmail SMTP plugin

Although technologically, sending email via php is very straightforward, for example using the PHP_mail script, security issues make it virtually impossible to do so without interacting with a third party. The following links walk you through the steps of authorizing WordPress to use a gmail account to send emails. The example uses the Google developer API to authorize the Gmail SMTP plugin to intercept emails send via PHP_mail and it via a secure process that will not be flagged as spam by email servers. This approach is also encouraged because it does not use logins and passwords, but rather private and public keys.

http://www.wpbeginner.com/plugins/how-to-send-email-in-wordpress-using-the-gmail-smtp-server/

Sidebars and Widgets

There are hundreds of widgets that can be added to a WordPress site to handle common tasks like playing audio, and displaying calendars.

Create a blank document and title it **sidebar.php**. Then add these lines:

```
<div id="sidebar">
```

```
  <?php dynamic_sidebar('Sidebar'); ?>
</div>
```

This will import the sidebar from your Dashboard to wherever you place it. Let's add it to **front-page.php:**

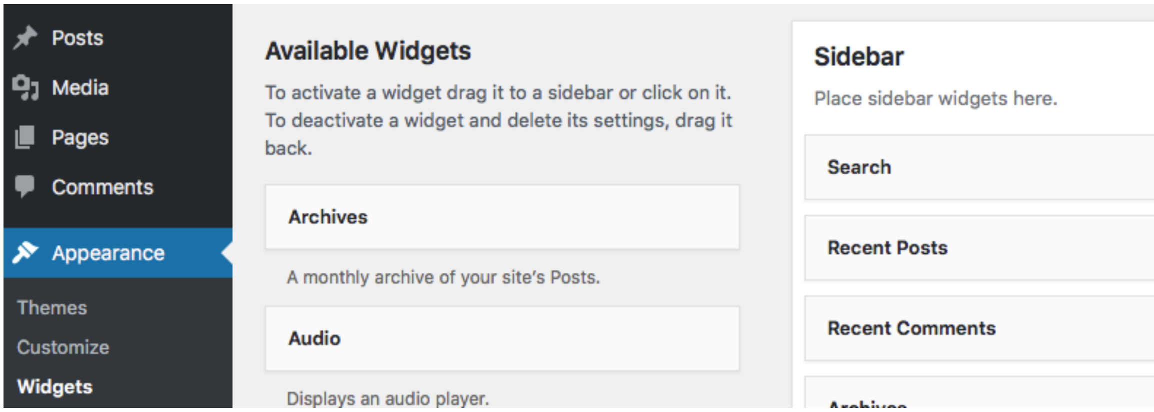
```
<?php get_header(); ?>
<?php get_sidebar(); ?>
<div class="posts">
```

Then we need to add some more lines to **functions.php** to register the sidebar in Dashboard:

```
$sidebar = array(
  'name'        => 'Sidebar',
  'id'          => 'sidebar',
  'description' => 'Place sidebar widgets here.',
);
register_sidebar($sidebar);
```

If we want, later multiple sidebars can be added.

Once **functions.php** is uploaded, widgets will appear in the menu.



Available Widgets

To activate a widget drag it to a sidebar or click on it. To deactivate a widget and delete its settings, drag it back.

A monthly archive of your site's Posts.

Displays an audio player.

Sidebar

Place sidebar widgets here.

Custom Post Formats

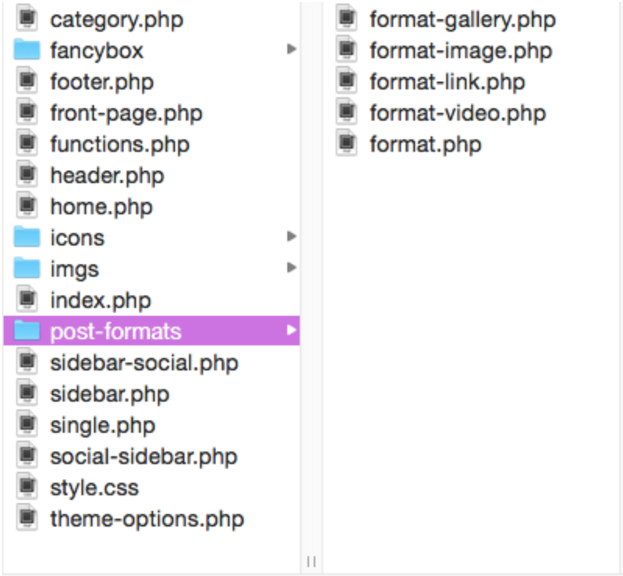
Step: Getting started with post formats

To start creating post formats, add this line to **functions.php**, which will allow us to select post formats within the Dashboard post editor:

```
add_theme_support( 'post-formats', array( 'gallery',
'image', 'video', 'link' ) );
```

Step

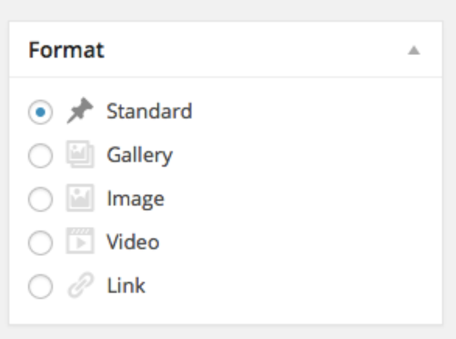
Start by making a new folder in your theme called **post-formats**:



category.php
fancybox
footer.php
front-page.php
functions.php
header.php
home.php
icons
imgs
index.php
post-formats
sidebar-social.php
sidebar.php
single.php
social-sidebar.php
style.css
theme-options.php

format-gallery.php
format-image.php
format-link.php
format-video.php
format.php

In this case I'm only using four of the possible post formats.
Once added, you will see a new widget in your post editor:



Format

● Standard
○ Gallery
○ Image
○ Video
○ Link

Step: Edit post formats and create your own

Copy and edit any of the post formats contained in the folder post-formats, which should be a child of your theme folder.

Review: Creating a Child Theme

Purpose

To demonstrate best practices when customizing an existing theme. See https://codex.wordpress.org/Child_Themes for details.

Overview

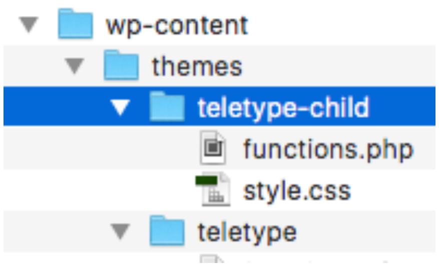
"A child theme is a theme that inherits the functionality and styling of another theme, called the parent theme. Child themes are the recommended way of modifying an existing theme."

To create a child theme you need to create a child theme folder in wp-content/themes, create a style.css and functions.php file in that folder.

Note that this example uses the teletype theme as the parent style. In the example below you will replace *teletype* with your parent theme folder name

Step: Create a Child Theme Folder

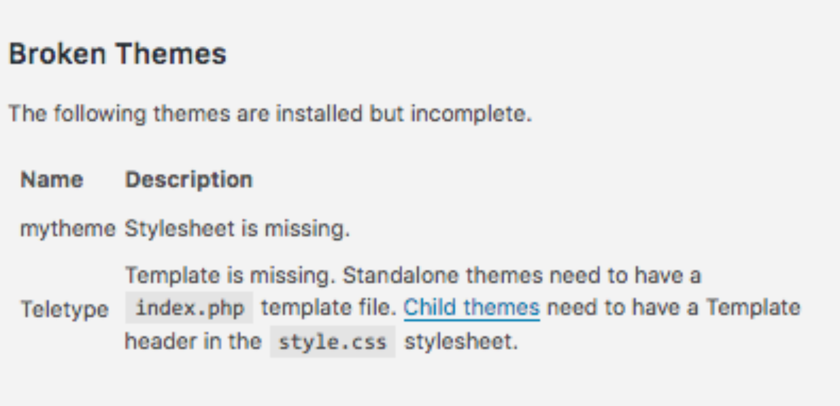
Create a child theme folder in wp-content/themes. Ideally give it the same name as the parent theme, with a –child extension. Do not include spaces in the child theme folder name.



```
▼ wp-content
    ▼ themes
        ▼ teletype-child
            functions.php
            style.css
        ▼ teletype
```

Figure: The theme with the child theme

If you go to Appearance Themes in your dashboard you will see the following error:

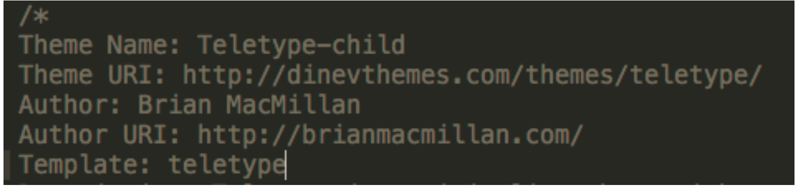


Broken Themes

The following themes are installed but incomplete.

Name	Description
mytheme	Stylesheet is missing.
Teletype	Template is missing. Standalone themes need to have a `index.php` template file. Child themes need to have a Template header in the `style.css` stylesheet.

To correct this, add a template tag to the child theme's style.css with the parent template's name.



```
/*
Theme Name: Teletype-child
Theme URI: http://dinevthemes.com/themes/teletype/
Author: Brian MacMillan
Author URI: http://brianmacmillan.com/
Template: teletype
```

Step: Enqueue the child theme's style sheets

Add the following code to the functions.php file of your child theme:

```
function my_theme_enqueue_styles() {

    $parent_style = 'teletype-style';

    wp_enqueue_style( $parent_style, get_template_directory_uri() . '/style.css' );
    wp_enqueue_style( 'child-style',
        get_stylesheet_directory_uri() . '/style.css',
        array( $parent_style ),
        wp_get_theme()->get('Version')
    );
}
add_action( 'wp_enqueue_scripts', 'my_theme_enqueue_styles' );
```

Note that because parent style has been set as a dependency, the child theme's style sheet will load after the parent theme's style sheet.

Also note that `get_template_directory_uri()` returns the parent styles directory. To get the child directory use `get_stylesheet_directory_uri()`.

Custom Templates

So far we've used standard templates, **index.php**, **front-page.php**, **home.php** and **blog.php**. The following exercise shows you how to create custom templates. This is a WP feature I use all the time.

https://premium.wpmudev.org/blog/creating-custom-page-templates-in-wordpress/

Shortcodes

Shortcodes are instructions that you can insert into a blog or a page that simplify common tasks, like creating a slideshow or linking to a **youtube** video.

https://en.support.wordpress.com/shortcodes/

Creating an Image Gallery in WP

There are a number of ways to create slideshows in WordPress.

The following link provides an excellent example of how to add a gallery.

https://en.support.wordpress.com/gallery/

The following example from **wpbeginner.com** provides two good examples.

http://www.wpbeginner.com/beginners-guide/how-to-create-an-image-gallery-in-wordpress/

Gallery Slideshow

Here is a second example that demonstrates how to create a gallery slideshow using a plugin.

https://wordpress.org/plugins/gallery-slideshow/

Accessing Bootstrap in WordPress

To access bootstrap in a WordPress theme all you need to do is download bootstrap, copy it to a folder in your theme (for example bootstrap) and then enqueue the files with the following code, in functions.php:

```
function register_bootstrap() {
   wp_enqueue_script( 'bootstrap-js', get_template_directory_uri() . '/bootstrap/jbootstrap.min.js', array(
'jquery' ), '3.3.2', true );
   wp_enqueue_style( 'bootstrap.css', get_template_directory_uri() . '/bootstrap/bootstrap.min.css?v= 3.3.7'
);
}
add_action('wp_enqueue_scripts', 'register_bootstrap');
```

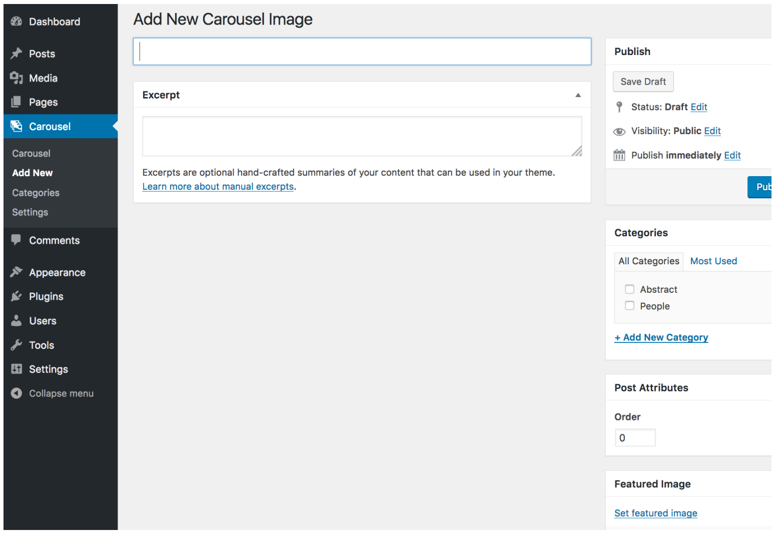
Download bootstrap from:

http://getbootstrap.com/

Exercise: Creating a Bootstrap Carousel

If you are using a theme based on bootstrap then you can use a bootstrap carousel plugin for your slideshows. This example uses the **CPT Bootstrap Carousel**.

Once installed and activated click on the Carousel menu

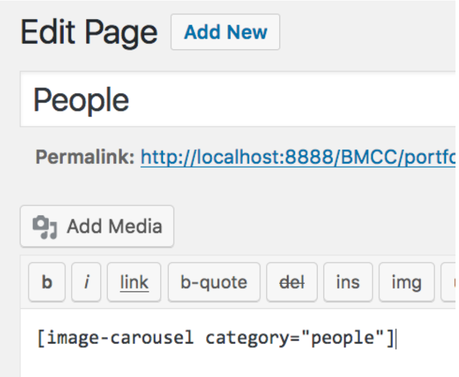


Excerpts are optional hand-crafted summaries of your content that can be used in your theme. Learn more about manual excerpts.

Order: 0

You can now give your image a title and excerpt. You choose an image by clicking Set Featured Image. You should also assign the image a category.

Once this is done you can easily set up a carousel with the following commands:



People

Permalink: http://localhost:8888/BMCC/portfc

```
[image-carousel category="people"]
```

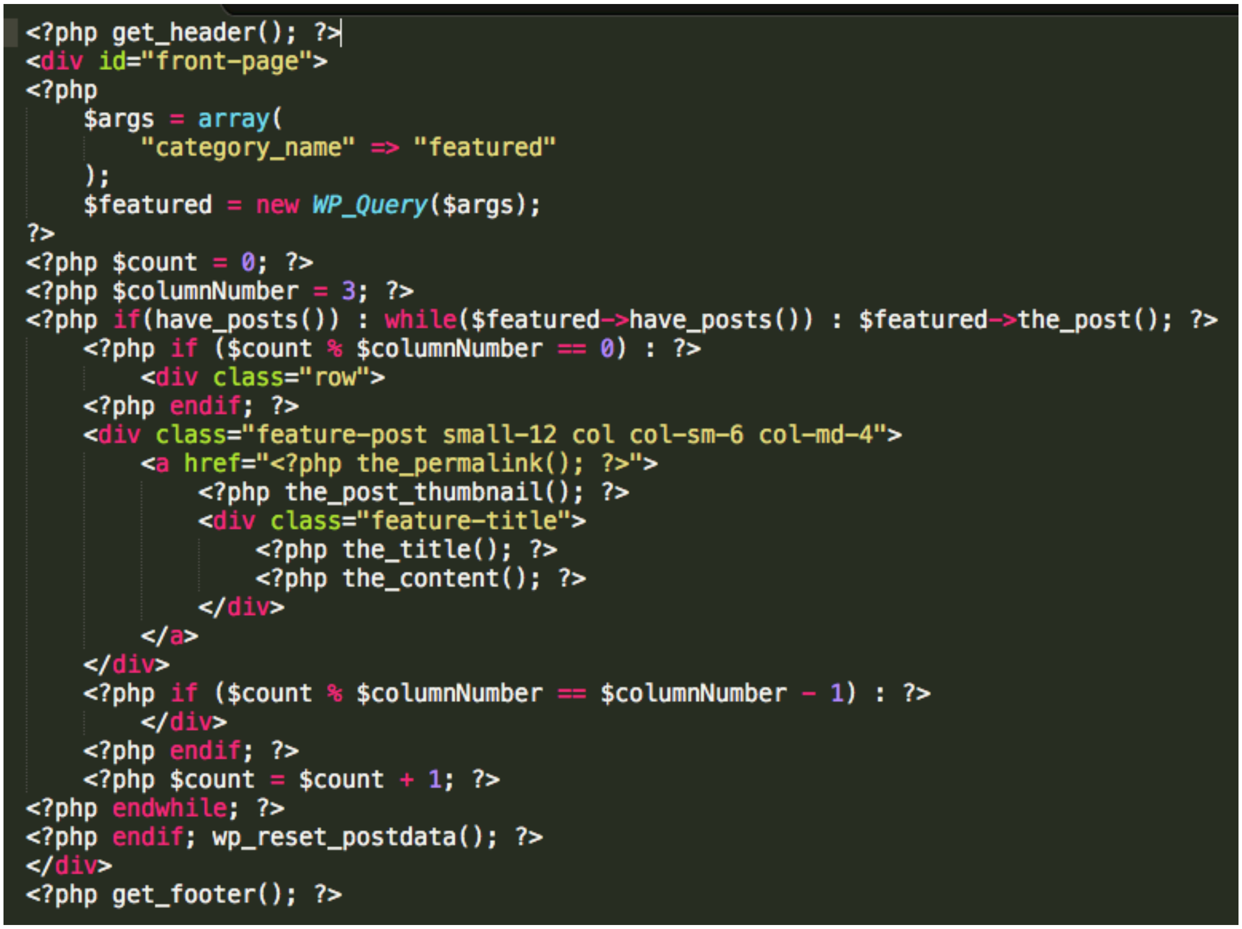
Exercise: Creating a multi-column post template using bootstrap and frontpage.php

Step

Add bootstrap files to your project and register them in `functions.php`

Step

Copy the file `front-page.php` from the file https://wpmmp.bmcc.cuny.edu/~bmacmill/classes/week8.zip to your root folder. If you already have a front-page.php, then consider merging the two files. Please talk with the instructor if you have questions.



```
<?php get_header(); ?>
<div id="front-page">
<?php
    $args = array(
        "category_name" => "featured"
    );
    $featured = new WP_Query($args);
?>
<?php $count = 0; ?>
<?php $columnNumber = 3; ?>
<?php if(have_posts()) : while($featured->have_posts()) : $featured->the_post(); ?>
    <?php if ($count % $columnNumber == 0) : ?>
        <div class="row">
    <?php endif; ?>
    <div class="feature-post small-12 col col-sm-6 col-md-4">
        <a href="<?php the_permalink(); ?>">
            <?php the_post_thumbnail(); ?>
            <div class="feature-title">
                <?php the_title(); ?>
                <?php the_content(); ?>
            </div>
        </a>
    </div>
    <?php if ($count % $columnNumber == $columnNumber - 1) : ?>
        </div>
    <?php endif; ?>
    <?php $count = $count + 1; ?>
<?php endwhile; ?>
<?php endif; wp_reset_postdata(); ?>
</div>
<?php get_footer(); ?>
```

Step: Add the appropriate bootstrap classes

Referring to the following documentation, assign classes to div.feature-post based on the the bootstrap column format you wish to use for your site.

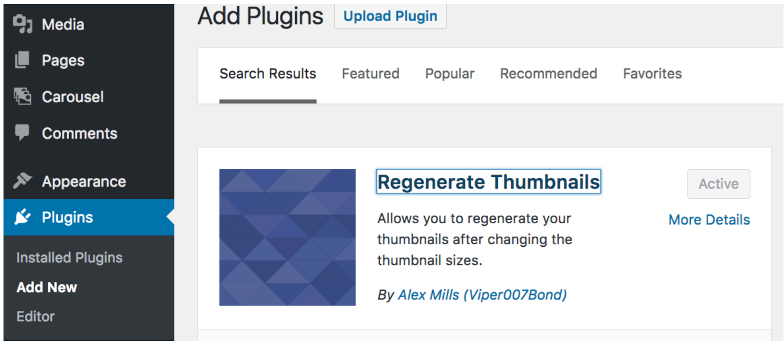
https://v4-alpha.getbootstrap.com/layout/grid/

Note

The code used in this example can be copied to any custom template file.

Exercise: Regenerating Thumbnails

There are a number of utilities that allow you to regenerate thumbnails. This exercise will use the Regenerate Thumbnails plugin.

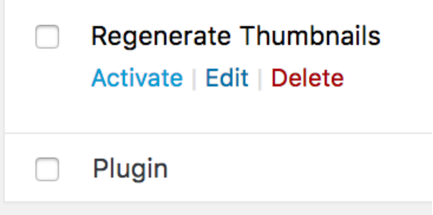


Allows you to regenerate your thumbnails after changing the thumbnail sizes.

By Alex Mills (Viper007Bond)

Figure: Go to Plugins Add New and search on Regenerate Thumbnails.

Next activate the plugin



Select Regenerate Thumbnails from the Settings screen

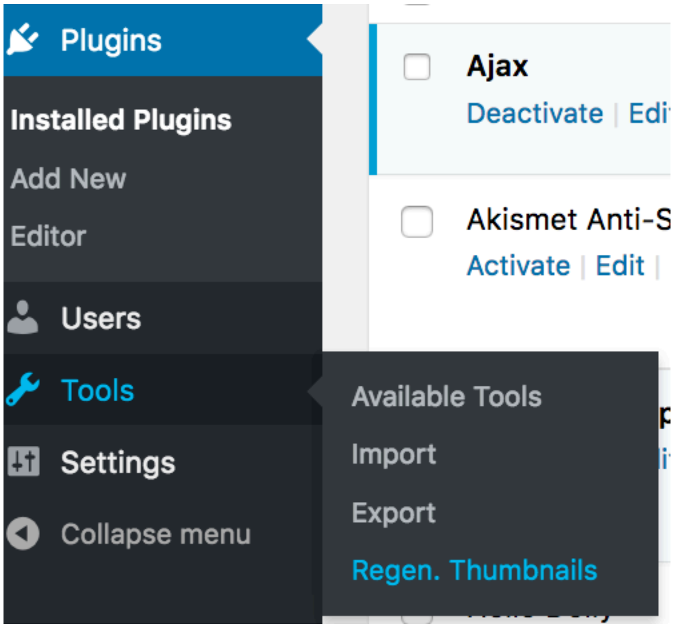
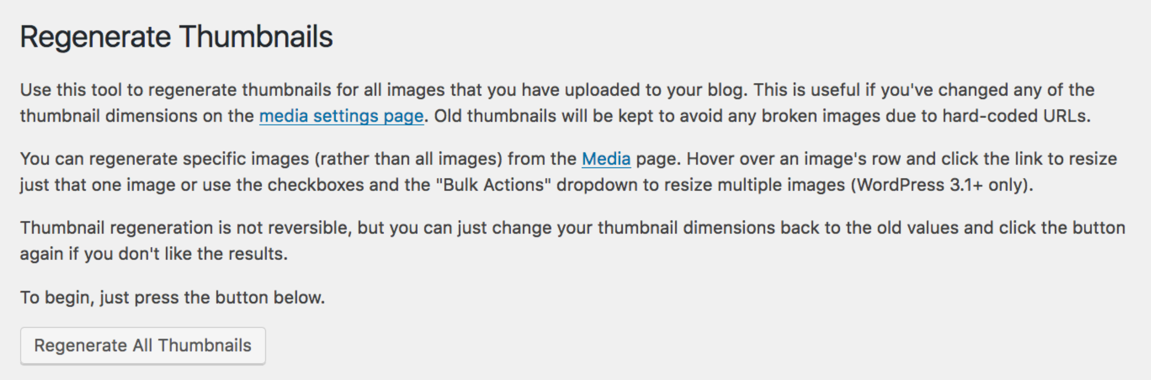


Figure: Select to regenerate thumbnails.



Regenerate Thumbnails

Use this tool to regenerate thumbnails for all images that you have uploaded to your blog. This is useful if you've changed any of the thumbnail dimensions on the media settings page. Old thumbnails will be kept to avoid any broken images due to hard-coded URLs.

You can regenerate specific images (rather than all images) from the Media page. Hover over an image's row and click the link to resize just that one image or use the checkboxes and the "Bulk Actions" dropdown to resize multiple images (WordPress 3.1+ only).

Thumbnail regeneration is not reversible, but you can just change your thumbnail dimensions back to the old values and click the button again if you don't like the results.

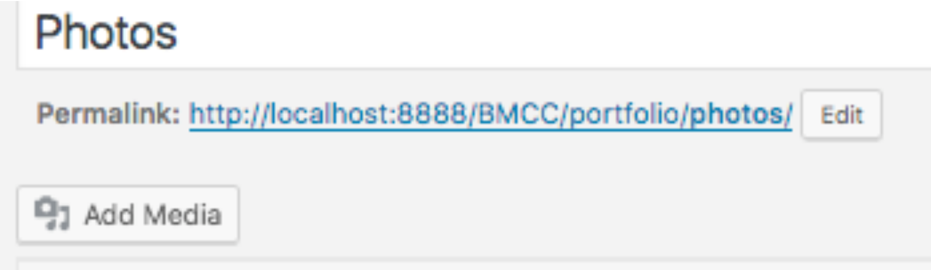
To begin, just press the button below.

Figure: Press Regenerate All Thumbnails to regenerate. The current media settings will be used.

Exercise: Creating and Formatting a Gallery Using Fancybox

Step: Create a gallery

Edit your photos page and click the Add Media button



Photos

Permalink: http://localhost:8888/BMCC/portfolio/photos/

Figure 1: Click the "Add Media" button in a new post or page. Use the photos page for this example.

Select *Create Gallery*

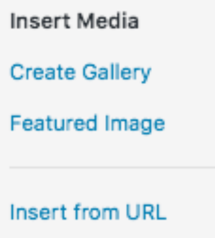
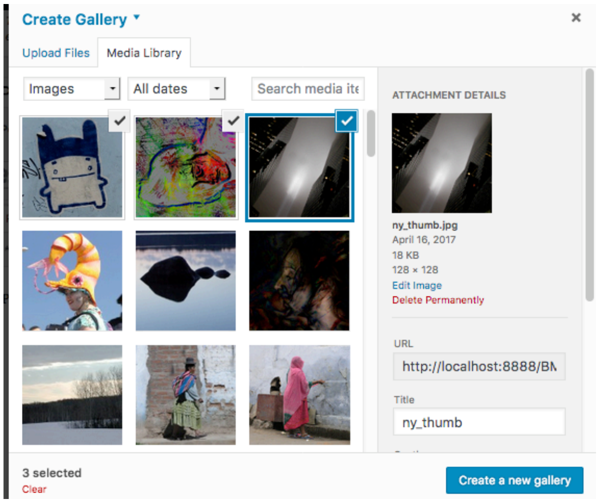


Figure 2: Choose "Create Gallery", select the images you want to use.



ATTACHMENT DETAILS

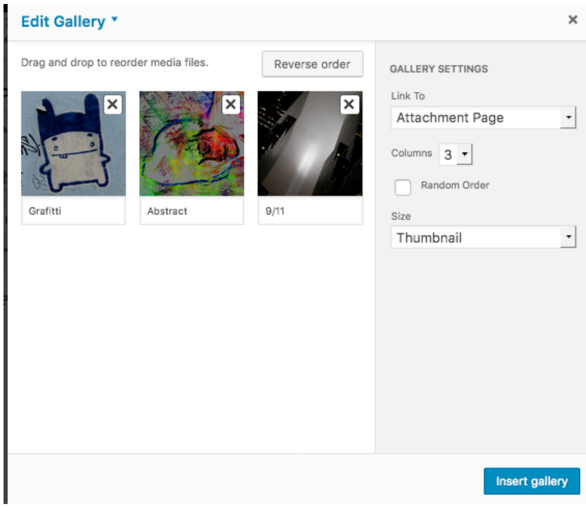
ny_thumb.jpg
April 16, 2017
18 KB
128 × 128

URL: http://localhost:8888/BN

Title: ny_thumb

3 selected

Figure: once you have selected the images you want, press “Create a New Gallery”



Drag and drop to reorder media files.

Grafitti

Abstract

9/11

GALLERY SETTINGS

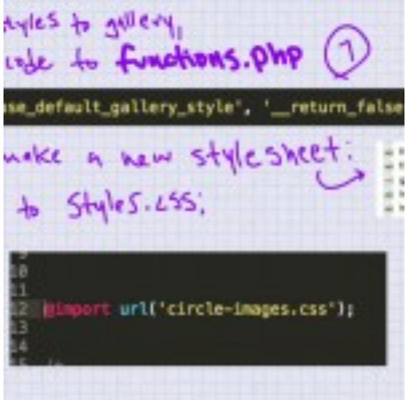
Link To: Attachment Page

Columns: 3

□ Random Order

Size: Thumbnail

Figure: Once you have created a gallery you can edit it, to add captions and reorder images.



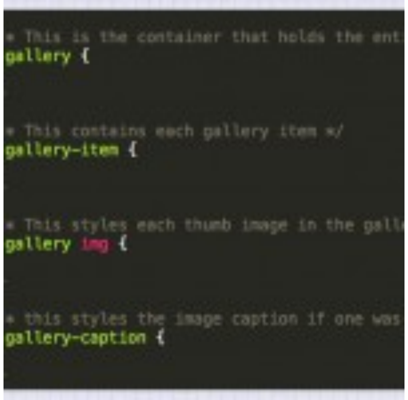
tyles to gallery,
code to functions.php ⑦

```
se_default_gallery_style', '__return_false
```

make a new stylesheet:
to styles.css;

```
@import url('circle-images.css');
```

Figure: You can add your own styles to the gallery using the CSS rules WP automatically applies to galleries. In this example the file ‘circle-images.css’ is being imported to the project.



```
* This is the container that holds the ent
gallery {


* This contains each gallery item */
gallery-item {


* This styles each thumb image in the gall
gallery img {


* this styles the image caption if one was
gallery-caption {

```

Figure: These are the basic style rules. You may want to add your own.

If you want pop up images, use a javascript library like Fancybox.

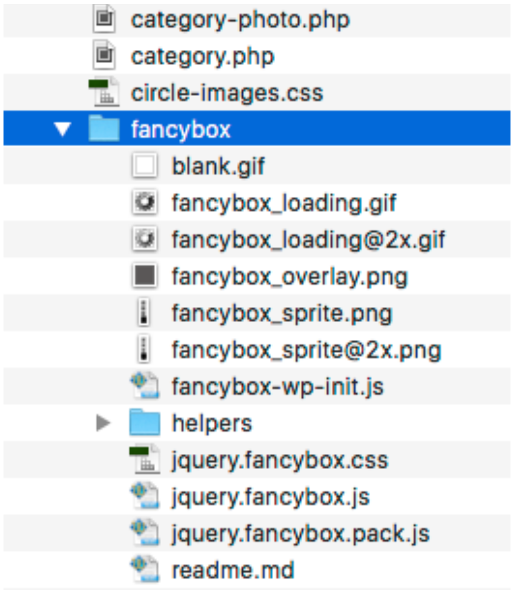
Step: Create a Lightbox using the jQuery plugin Fancybox and a Gallery

What follows is based on the tutorial at:

http://code.tutsplus.com/tutorials/add-a-responsive-lightbox-to-your-wordpress-theme--wp-28100

Fancybox is included in the file week8.zip:

https://wpmmp.bmcc.cuny.edu/~bmacmill/classes/week8.zip



category-photo.php
category.php
circle-images.css
fancybox
blank.gif
fancybox_loading.gif
fancybox_loading@2x.gif
fancybox_overlay.png
fancybox_sprite.png
fancybox_sprite@2x.png
fancybox-wp-init.js
helpers
jquery.fancybox.css
jquery.fancybox.js
jquery.fancybox.pack.js
readme.md

Figure: Week 8 Files

Step

To make the jQuery plugin work you need to enqueue it. To do so, add the following code to functions.php.

```
// Enqueue Scripts/Styles for our Lightbox
function add_fancybox() {
    wp_enqueue_script( 'fancybox-js', get_template_directory_uri() .
'/fancybox/jquery.fancybox.pack.js', array( 'jquery' ), false, true );
    wp_enqueue_style( 'fancybox-style', get_template_directory_uri() .
'/fancybox/jquery.fancybox.css' );

    //this is our code that sets up fancybox to use in WP
    wp_enqueue_script( 'fancybox-wp-init', get_template_directory_uri()
. '/fancybox/fancybox-wp-init.js', array( 'fancybox-js' ), false, true
);

    //this lets you see other images as thumbs
    wp_enqueue_script( 'fancybox-thumbs', get_template_directory_uri()
. '/fancybox/helpers/jquery.fancybox-thumbs.js', array( 'fancybox-js'
), false, true );
    wp_enqueue_style( 'fancybox-style-thumbs',
get_template_directory_uri() . '/fancybox/helpers/jquery.fancybox-
thumbs.css', array('fancybox-style') );
}
add_action( 'wp_enqueue_scripts', 'add_fancybox' );
```

Step

Add the following code to `fancybox-wp-init.js`. This is the file that will initialize the jQuery plugin.

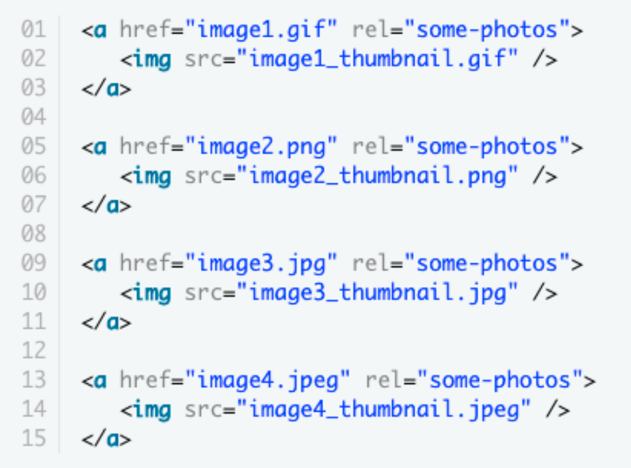
```
(function($) {

    //Make modifications to this file as needed to customize fancybox
to your needs.

    // Initialize the Lightbox for any links with the 'fancybox' class
    $(".fancybox").fancybox();

    // Initialize the Lightbox automatically for any links to images
with extensions .jpg, .jpeg, .png or .gif
    $("a[href$='.jpg'], a[href$='.png'], a[href$='.jpeg'],
a[href$='.gif']").fancybox();

    // Initialize the Lightbox and add rel="gallery" to all gallery
images when the gallery is set up using [gallery link="file"] so that a
Lightbox Gallery exists
    $(".gallery a[href$='.jpg'], .gallery a[href$='.png'], .gallery
a[href$='.jpeg'], .gallery
a[href$='.gif']").attr('rel','gallery').fancybox();

    //If you want your gallery to have thumbnails at the bottom use
this instead of line 10 above
    /*
    $(".gallery a[href$='.jpg'], .gallery a[href$='.png'], .gallery
a[href$='.jpeg'], .gallery
a[href$='.gif']").attr('rel','gallery').fancybox({
         helpers : {
             thumbs: {
                 width  : 50,
                 height : 50
             }
         }
     });
    */



    // Initalize the Lightbox for any links with the 'video' class and
provide improved video embed support
    $(".video").fancybox({
        maxWidth        : 800,
        maxHeight       : 600,
        fitToView       : false,
        width           : '70%',
        height          : '70%',
        autoSize        : false,
        closeClick      : false,
        openEffect      : 'none',
        closeEffect     : 'none'
    });

})(jQuery);
```

Note: The above code contains examples for video and for images. Only include the code you actually use in your final project.

The sample so far is set up to automatically add a lightbox for all links to files with the .jpeg, jpg, .png and gif extensions. Simply referencing a jpg will activate the lightbox:

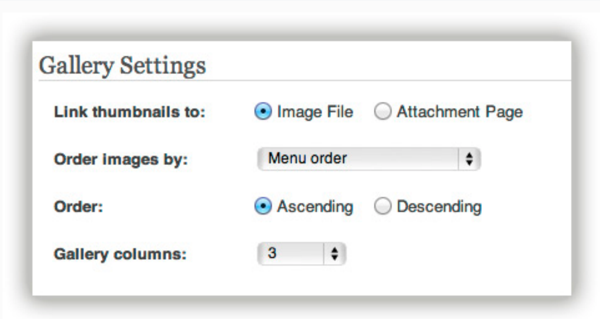
```
<a href="image.jpg"><img src="image_thumbnail.jpg" /></a>
```

You can create a 'lightbox gallery' by adding the same 'rel' attribute to several links. Here is an example from the tutsplus.com tutorial:



```
<a href="image1.gif" rel="some-photos">
   <img src="image1_thumbnail.gif" />
</a>

<a href="image2.png" rel="some-photos">
   <img src="image2_thumbnail.png" />
</a>

<a href="image3.jpg" rel="some-photos">
   <img src="image3_thumbnail.jpg" />
</a>

<a href="image4.jpeg" rel="some-photos">
   <img src="image4_thumbnail.jpeg" />
</a>
```

Or you can insert any WordPress gallery into your post or page. If you do this, you must link the thumbnails to the images files:



Gallery Settings

Link thumbnails to: ● Image File ○ Attachment Page

Order images by: Menu order

Order: ● Ascending ○ Descending

Gallery columns: 3

Figure: Linking thumbnails to image files. Note that this view may vary depending on which version of WordPress you are using.

Video support is added by adding the `video` and `fancybox.iframe` classes to the element linking to your video, for example:

```
<a href="http://player.vimeo.com/video/50006726?badge=0"
class="video fancybox.iframe">
    Click this to open up a Video from Vimeo!
</a>
```

Additional Reading

How to Add Posts to Pages

The Wordpress Loop.

Template hierarchy

Comments template.

Date and time format.

Tags

Author posts

Enqueueing