MMP350 Class Notes Week 6

MMP350 Class Notes Week 6	. 1
This Week's goals	. 1
What is WordPress?	. 2
The WordPress Dashboard	. 3
Pages	. 4
Posts and Pages	. 5
Categories	. 5
Tags	. 6
Discussion	. 6
Media Library	. 6
General Settings	. 7
Reading Settings	. 8
Writing Settings	. 8
Php Basics	. 9
What is PHP?	. 9
Helloworld	. 9
Summary	11
Building a WordPress Theme: index.php	11
The WordPress Loop and front-page.php	18
Step 1	23
Step 2	24
Step 3: Displaying the Featured Image as a Thumbnail	24
Step 4: Modifying home.php for blog posts	24
Footer, Sidebars, Single.php and Category.php	25
Footer.php	25
More WordPress – single.php, category.php, category.php, page.php, sidebar.php	3
·	26
Homework / Readings	31
Homework Assignment Error! Bookmark not define	ed.

This Week's goals

- Review the WordPress Dashboard
- PHP overview
- Review of WordPress Architecture
 - o File system
 - Template hierarchy
- Your first WordPress theme
- Review of Fifteen Common WordPress Mistakes, if there is time.

Homework

Read the following article on the WordPress template hierarchy and summarize it in your own words.

http://codex.wordpress.org/Template_Hierarchy

Note that there has been a slight change, relative to the syllabus. The assignment to create your own customized template will be *next week's homework*, not this week's homework.

What is WordPress?

WordPress was created as a blogging platform and has become a powerful backend for many sites that are content based. It is a:

Content Management System – WordPress has an online interface for users with little or not knowledge of HTML or code to post content.

WordPress is open source, it can be downloaded, installed and used for free.

Dashboard – This is the client side of WordPress where content is created and uploaded to a server.

Themes – Visual design of the site is determined by themes, which are applied to content from blog posts, pages and widgets.

WordPress pages are populated by taking code from many sources, which allows them to be dynamic and host many different kinds of content.

WordPress.com – Free, hosted on WordPress server, limited customization. We'll start with a WordPress.com blog today.

WordPress.org – Free, hosted on your own server, completely customizable.



The WordPress Dashboard

Now we're going to review some basics of the WordPress Dashboard.



Figure: The WordPress Dashboard

On the WordPress Dashboard you will find everything you need to post new pages, blog posts and edit the look and structure of your site.

Your portfolio site will include pages for navigation, basic information and organization as well as posts for individual projects.

Pages

Pages are static individual pages that usually contain navigation to other groups of pages. Let's quickly create an About page, and edit the home page using the Page editor. Note that the Page editor is very similar to the Post editor (which we would look at later), but Pages will not be included in the blog feed.

🛞 🔇 MMP 350 _		mium 🛛 New Post 🔽 orobertsbrncc 🔟 🔍
Dashboard	Add New Page	Screen Options * Help *
📄 Store		Publish A
→ Posts	1	Save Draft Preview
୍ୟୀ Media ଡି Links	P1 Add Media	9 Status: Draft <u>Edit</u>
Pages	B I *** 日日 46 一日 主 日日 68 国 学 III *** 日日 46 一日 主 日日 78 国 学 III *******************************	Visibility: Public Edit Publish immediately Edit
All Pages Add New		Move to Trash Publish
Copy a Page Comments		Page Attributes
E Feedback		Parent
Appearance		(no parent) 🗘
🍰 Users チ Tools		Order
Settings		Need help? Use the Help tab in the upper
Collapse menu	p Word count: 0	right of your screen.
	THE COMPLET	Featured Image
	Writing Helper	Set featured image
	Copy a Page Use an existing page as a template. Request Feedback Get feedback on this draft before publishing.	

Figure: Add New Page

There are two ways to create a new Page from the Dashboard:



Figure: Two ways to add a page to WordPress, from the Pages menu item or by pressing the + New option and selecting Page.

Your site can have a number of pages including Home, About, Blog, and links to specific posts.

Posts and Pages

Posts and pages are created in similar ways but serve different functions. *Pages organize content, while posts contain the actual content.* You can add projects you've worked on to posts and organize them with categories and tags. Those posts will then appear in your feed. The interface for creating and editing posts is the same as that for Pages, with a couple of important extra functions.

Further exploration: The following article has an excellent example that illustrates the relationship between posts and pages:

https://www.webmechanix.com/how-to-add-posts-to-pages-in-wordpresstutorial/#guide

Categories

Categories are a useful way to organize content. You might have categories like: Web Design, Interactive, Fine Art, Game Design, or whatever other kinds of work you want to show on your portfolio.

Categories	
All Categories	Most Used
Assignme Notes Uncatego	nts rized
+ Add New Cate	egory

Figure: Categories

Tags

Tags are another way to organize content. You might add tags like HTML, CSS, JavaScript, PHP, Design, Graphics, Motion, BMCC, or other tags to describe the content in your posts. This will make it easy for readers of your portfolio to see what kind of work you have done.

Tags	
Ad	bb
Separate tags with commas	
Choose from the most used tags	

Figure: Tags

Discussion

What is the difference between categories and tags?

Media Library

The WordPress media library will store all of the images, videos and sound you upload to your blog. You can upload new media either by using the link to the Media Library in the Dashboard, or by simply dragging and dropping images and other media directly onto your blog post.

₩ 🖉 ММР 390Ц		Go Premium New Post 🔽 orobertsbmcc	
Dashboard Media	Library Add New	He	elp 🔻
You are cur	ently using 42.4 kB out of 3.0 GB upload limit (0%).		
🖈 Posts Did you kno	w that you can upload videos and embed them directly on your blog with our VideoPress upgrade?		
91 Media			
Add New	All media items All dates Bulk Select Bulk Select	Search	
 P Linis Pages Comments ● Feedback Appearance Uters Uters Settings O Collapse menu 	Mar		
Thank you fo	r creating with WordPress • Support • Forums • Learn WordPress — Tutorials and Walkthroughs		
Thankyou fe	r crosting with WordPress - Support - Forums - Learn WordPress — Tutorials and Walkthroughs	Ga Pren	alum New Post 🐖 orobertsbrnc
Thank you fe	r crasting with WordPress - Support - Forums - Learn WordPress — Tutorials and Walkthroughs	Go Pver	new Post 🤵 orobertsbraci
Thenkyou Je MMP 350 Insert Media Create Gallery Set Featured Image Insert Tweet Insert YouTube Insert from URL	r createg with WordPress - Support - Forums - Learn WordPress — Tutorials and Walkthroughs	Search	Nev Post 💽 orobertsbris

Figure: Media

General Settings

Take a moment to look at the general settings for you blog. You can change the site title and the formatting for displaying information like the date and time of posts.

🛞 🚯 MMP 350		l	🕼 Præmium 🛛 New Post 🔽 orobertsbrncc 🕅 🔍
Dashboard	General Settings		Help 🔻
) Store	Ū		Plag Dicture / Icon
* Posts	Site Title	MMP 350	Biog Picture / Icon
P1 Media	T !!		Upload a picture (jpeg or png) to be used as
R Links	ragine	In a few words, explain what this site is about.	your blog image across WordPress.com. We will let you crop it after you upload.
Pages			Choose File. No file chosen
Comments (1)	Timezone	UTC+0 \$ UTC time is 2014-09-02 15:38:42	
Feedback		Choose a city in the same timezone as you.	Upload Image
	Data format	Festender 3 3014	
Appearance	Date Format	2014-09-02	
👗 Users		09/02/2014	
∲ Tools		02/09/2014	
Settings		Custom: F J, Y September 2, 2014	
General			
	Time Format	• 3:38 pm	
Reading		3:38 PM	
Discussion		15:38	
Sharing		Custom: g:l a 3:38 pm	
		Documentation on date and time formatting.	
	Week Starts On	Monday	
Email Post Changes	Language	en - English 💠	
OpenID Webhooks		Language this blog is primarily written in. You can also modify the interface language in your profile.	

Figure: General Settings

Reading Settings

The reading settings contain organization that the blog reader or user will see, like the number of posts.

🛞 🚷 MMP 350 _		Co 1	Premium	New Post	orobertsbmcc 🔲	۹ I
Dashboard Dashboar	Reading Settings				Help	Y
💓 Store	Redding Settings					
	Front page displays	Your latest posts				
<i>y</i> [™] Posts		A static page (select below)				
9] Media						
& Links		Front page: — Select — \$				
Pages		Posts page: — Select — \$				
투 Comments 🚺						
Feedback						
	Blog pages show at most	10 posts				
A lisers	Sundication feeds show the					
& Tools	most recent	10 items				
P Settinge						
Jettings	For each article in a feed, show	Full text				
General		Summary				
Reading						
Discussion	Site Visibility	Allow search engines to index this site				
Media		Discourage search engines from indexing this site				
Sharing		Note: Weither of these options blocks access to your site — it is up to search engines to honor your request.				
Polls		I would like my site to be private, visible only to myself and users I choose				
Ratings						
Email Post Changes	Related posts	Hide related content after posts				
OpenID		Show related content after posts				
Webhooks		Snow a "keiated" neader to more clearly separate the related section from posts I lise a large and visually striking layout				



Writing Settings

Writing settings will affect the CMS and has some short cuts you can use for your own blog posts.



Php Basics

Before we create our first theme, let us review some php basics.

What is PHP?

PHP is a hypertext preprocessor. In practical terms, that means it is a server-side scripting language that is used to process http requests (typically to a fileserver or database server) and create html code.

Helloworld

Let's examine a very simple php file. To start, go to:

http://php.net/manual/en/tutorial.firstpage.php

... and create the file helloworld.php

```
<html>
<head>
<title>PHP Test</title>
</head>
<body>
<?php echo '<p>Hello World'; ?>
</body>
</html>
```

Try to view it in a browser. It doesn't work.

Now click on this link to my webserver:

http://45.55.178.198/BMCC/hello.php

PHP needs to be served by a webserver like apache. It will not automatically display in a browser.

Let's look at the syntax of a second php file:



What do you notice about this file?

Now let's look at a more complicated php file, getDemographicsJSON.php, which was mailed to you.



Figure: A code fragment from getDemographicsJSON.php

Summary

- 1. PHP requires a server like Apache in order to be executed.
- 2. PHP can be embedded in html. This embedding is called a server-side-include, because it executes code on the server.
- 3. Variables begin with \$
- 4. All lines end with a semi-colon
- 5. PHP files / statements begin with <?php and end with ?>
- 6. = is the assignment operator
- 7. **==** is the string comparison operator
- 8. (period) is the string concatenation operator
- 9. php variables can be concatenated within a strong, for example sFullname = "\$fname . \$lname"

Building a WordPress Theme: index.php

We're going to start building out the basics of our WordPress theme starting with a template with two files: **index.php** and **style.css**. This is the minimum requirement for a WordPress theme to be installed.

To get started unzip **myfirsttheme.zip**.

http://wpmmp.bmcc.cuny.edu/~bmacmill/classes/themestarter.zip

Note that index.php is the base file for the folder, as opposed to index.html. We'll add a few things to **style.css** and then test it on our WP installations.

Let's start with the <head> section of index.php; it is at the beginning of the document.



```
Figure: the head section
```

Most of this will stay the same, but we need to change two lines.

First is the <title>. WordPress themes should be customizable, so we will begin replacing default or hard coded content with php variables that are provided by WordPress.

<title><?php bloginfo('name');?> | <?php bloginfo('description')?></title>

Figure: Inserting WordPress functions into the <title> element.

This is the code you will be adding:

<?php bloginfo('name'); ?> | <?php bloginfo('description')?>

The first php tag uses the **bloginfo()** function to retrieve the blog name. The second line uses the same function to retrieve the blog description, which is a reference to the tagline from the Dashboard settings

By using php variables, we can get those pieces of information, filled in by the user or by us, at any part of our website.

Next change the CSS link tag:

k rel="stylesheet" type="text/css" href="<?php bloginfo('stylesheet_url'); ?>">

Figure: Adding a php function to a <link> element

This is the code you will be adding:

```
<link rel="stylesheet" href="<?php bloginfo('stylesheet_url'); ?>">
```

Again we're using the bloginfo() function, this time to get the URL location of the stylesheet. Because WP uses a more complex file structure than a static website, it is a good idea to use this references instead of a direct URL to the style sheet.

Note that php can be used inside of the quotes of html attributes, like href.

Finally, add this tag inside the head section:

<?php wp_head(); ?>

Figure: Adding a function in the head section.

This is the code you will be adding:

```
<?php wp_head(); ?>
```

This is going to load some WordPress functionality into the head of the website. We're going to skip the <header> and <nav> for now, we'll cover those in depth in upcoming lessons.

Skip ahead to this section:



We're going to add a few lines and make some changes here.

First add this line, called the WP Loop. We'll cover this in depth soon. The important thing to know now is that WP Loop does most of the work of going through all of your posts or pages and grabbing the content from the database.



This is the code you will be adding:

```
<?php if(have_posts()) : while(have_posts()) : the_post(); ?>
```

Next, replace all of The Title of the Post with the following:



This is the code you will be adding:

These lines are using functions to get the link to each post and the title of the post. You should start to see a pattern emerging.

Next, replace the entire section <div id="content"> ... </div> with this line:

<?php the_content(); ?>

```
Next replace the line Posted on February 21, 2005 by <span class="author">author</span> with this:
```

```
Posted on <?php the_date(); ?> by
<span class="author"><?php the_author(); ?></span>
```

Figure: adding the author of the website and date of each post.

This is the code you will be adding:

```
Posted on <?php the_date(); ?> by
```

<?php the_author(); ?>

Finally, delete the entire second <article>. With WordPress working, we only have to make a single template for each post, which will be populated with new content.

Then, after the <article> section, before the end of the <div id="content"> section, add the following:



Figure: Exception handling while loading blog posts.

This tells WordPress what to do once it's loaded all of the blog posts or there are no blog posts to load.

Let's change a couple of things in the <footer>:



Add the current the_date() and the_author () functions to the footer:



That's it for **index.php** for now.

Let's edit **style.css** and then test our theme.



Right now this is all you will see in **style.css**. In order for WordPress to recognize the theme, it needs to have this comment with this basic information. Go ahead and replace the info for **Theme Name**, **Author** and **Description** with your own information.

Now let's upload these themes and test them out. If this is the first time uploading the theme, then turn its folder structure into a zip file and upload it. Note that in this example I have created a zip file called myfirsttheme.zip by highlighting its parent folder and compressing it.



Select Upload Theme

	o Downloads	\$		Q Search
Name		Date	Modified	~
myfirsttheme.zip		Today	/, 4:36 PM	
myfirsttheme		Today	/, 4:07 PM	
MMP350Midterm.docx		Today	, 3:02 PM	
themestarter(2).zip		Today	, 2:35 PM	
optical.gif		Today	, 11:49 AM	4
wampserver2.5-Apache-2.4.9-Mys	ql-5.6.17-php5.5.12-32b.exe	Today	, 9:32 AM	39
Report_w_smaller_images.doc		Today	/, 8:41 AM	30
Report (1).doc		Yeste	rday, 8:05 PM	35
Report.doc		Yeste	rday, 8:03 PM	35
 MacMillan Rita for entry.jpg 		Yeste	rday, 6:32 PM	
themestarter(1).zip		Yeste	rday, 6:29 PM	
		Yeste	rday, 11:53 AM	
my_derivative_theme 2.zip		Yeste	rday, 11:52 AM	
Options			Cance	el Open
Installing Theme from up myfirsttheme.zip	bloaded file:			
Unpacking the package				
Installing the theme				
Theme installed successfully.				
Live Preview Activate Return to Then	<u>nes page</u>			

Your theme should just consist of **index.php** and **style.css**. Once it's uploaded, you should the theme in the WordPress Dashboard in **Appearance > Themes**.



From here, click **Activate** to see your theme in action:

Themes 🖪	Add New	Search installed th	emes	
New theme activ	vated. <u>Visit site</u>			
Twenty Fifteen The Wardroo order is draw to 2005. Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marce Marcee Marce Marce Marce Marce Marc	The Myth of the P	ier		Theme Details
men data series of the series	Arrend wher the balance of the Media disabless or ghoted pair of height is then was the bags sear work which is man obtained by made of them. Then appear the disable with the minimum of the theory of the search of the search of the characteristic of the search of the search of the characteristic of the search of the search of the characteristic of the search of the search of the Fifteen	App, Shop Yan Egon Mor Instance David Vargerightet Instance David Vargerightet were studie of Tambaco, and a very were studie of Tambaco, and a very more studie of Tambaco, and a very more studie of Tambaco, and a very were studie of Tambaco, and a very were studie of Tambaco, and a very were studie of Tambaco, and a very Customizee	My Theme	Activate Live Preview

The WordPress Loop and front-page.php

Now we're going to go over the WordPress Loop, which is used to populate most of the content on our pages and posts, and creating a front page for the portfolio sites.

Exercise file: wp_loop.zip

The Loop

The WordPress loop is the process by which WordPress compiles pages from its component parts.

Here's the WordPress explanation of the Wordpress Loop.

The WordPress loop iterates through available content and makes it available to the template code. This makes it easy to customize the content we create in the WordPress CMS to fit our theme and design.

<div class="posts"> 020 Starting posts Deginning <?php if(have_posts()) : while(have_posts()) : the_post(); class="blogPost" Title of the no %href="<?php the_permalink(); ?>"> <?php the_title(); ?> </h2> class="content"> 2 content <?php the_content(); ?> <?php endwhile; else: ?> 🕊 end lovp <?php _e('Sorry, no posts matched your criteria.'); ?> </ > <?php endif; ?> <!--End of posts </div> <!-- end content -

Loop template tags

There are many template tags that can be used only within the WordPress loop. A few tags that may be useful on our blogs:



The excerpt is a short line or description of the post. If there is no excerpt, WP fills in some text from the beginning of your post.

the_content() is the content of the post/page, so any text, images or other media that are part of the body of an entry.



The content will typically go into a div or article.



IDs and classes can be set for the post, which can be used for CSS styling or JavaScript functionality.



The date the post was written is added with **the_date()**. An argument can be added for format.



the_author_posts_link() adds a link to all posts by the author. Your site will most likely have one author, but this will be useful for future projects or as an option on your theme.



Tags are a great way to organize content. WP helps style the tag with arguments to determine how they are separated with HTML and text.



Categories can also take an argument for how they are separated.

```
<?php the_post_thumbnail(); ?>
```

the_thumbnail() will show the featured image tagged in the post/page.

```
<?php previous_post_link('%link', '%title', TRUE ); ?>
<?php next_post_link('%link', '%title', TRUE ); ?>
```

Links to next/last post. Third argument is set TRUE to only let posts with same category, FALSE for another post/page to come up.

Okay, now that we have some new WP toys to play with, let's build a front page. Most of your portfolio designs require a homepage, either with a large image, some information or other design elements. Currently our sites all open up to blog posts. We can use **front-page.php** to create a separate page to appear at the index of the site.

Let's look quickly at the WP <u>template hierarchy</u> to understand how WP decides what page to load at the root of the site.

First, create a page called **Front Page** and a Page called **Blog** from the Dashboard.

Dashboard	Add New Page
📌 Posts	Front Page
9; Media	
Pages	
Dashboard	Add New Page
📌 Posts	Blog
9; Media	
Pages	

Front Page will be the splash page of the portfolio, and blog will be the new page to display posts. We can leave them blank at first.

Pages	Title
All Pages Add New	About
Comments 1	Blog
🔊 Appearance	Front Page
🖌 Plugins	

Next go to Settings > Reading and set the front page and posts page to Front Page and Blog.



Reading Settings			
Front page displays	O Your latest	posts	
	• A <u>static pag</u>	<u>te</u> (select belo	ow)
	Front page:	Front Page	¢
	Posts page:	Blog	\$

Now we need to add Features posts to appear on the front page. Add a new Category called Featured. In All Posts, Quick Edit a couple of posts and add "Featured".

📌 Posts	
All Posts	Add New Category
Add New	Name
Categories	Featured
Tags	The name is how it appears on your site.

Now for some PHP fun.

Go to your local WP files. Notice that there are a number of new files, header.php, functions.php, front-page.php and home.php.

🔻 📄 wp	_loop
	front-page.php
	functions.php
	header.php
	home.php
0	index.html
	index.php
	style.css

Let's talk about header.php first. Header.php is a generic header that can be reused by all of your pages. It is called by the function get_header() which you will see that the top of index.php, front-page.php and home.php.



html
<html lang="en"></html>
<head></head>
<meta charset="utf-8"/>
<pre><meta content="width=device-width, initial-scale=1.0" name="viewport"/></pre>
<pre><title><?php bloginfo('name'); ?> <?php bloginfo('description'); ?></title></pre>
<pre><link href="<?php bloginfo('stylesheet_url'); ?>" rel="stylesheet" type="text/css"/></pre>
php wp_head(); ?
 body>
<pre><div class="wrapper"></div></pre>
<header></header>
<h1><a href="<?php echo home_url('/'); ?>"><?php bloginfo('name'); ?></h1> <h2><?php bloginfo('description'); ?></h2>
<pre><?php \$main_menu_top = array(</td></pre>
?>
<pre><?php wp_nav_menu(\$main_menu_top); ?></pre>

Figure: the contents of header.php

Q: What do you notice about header.php?

Step 1

Front-page.php and home.php are clones of index.php Both files will use the loop to display different content. They are activated in different contexts.

Let's edit **front-page.php** first.

For the front page we're not going to use the title for the post, just the content.

```
<div class="front-page">
    <?php if(have_posts()) : while(have_posts()) : the_post(); ?>
    <div class="intro">
        <?php the_content(); ?>
        </div>
        <?php endwhile; else: ?>

        <?php __e('Sorry, no posts matched your criteria.'); ?>

        <?php endif; ?> <!--End of posts --->
</div>
```

I want to make sure only the Featured posts appear on the front page, so I'm going to make a custom query using **WP_Query()**. I'm going to create an argument for the query as a variable and save the result in another variable:

Step 2



and feed that variable into my content loop:

<?php if(have_posts()) : while(\$featured->have_posts()) : \$featured->the_post(); ?>

Finally, I'm going to take out the **else** statement. If there are no Featured posts, well, I'm going to have to make some to fulfill my design. I also need to add the function **wp_reset_postdata()** to reset my query for later loops.



Step 3: Displaying the Featured Image as a Thumbnail

Okay, this still looks like a blog. Let's change the template a bit to make it look like a portfolio. We'll need to add some basic styles and change the content a bit.

First, let's set the featured image with a single line:



Figure: add theme support for featured image thumbnails to functions.php

Step 4: Modifying home.php for blog posts

Now modify home.php so that it looks like the following:

```
<?php get_header(); ?>
       <div class="posts">
    <?php if(have_posts()) : while(have_posts()) : the_post(); ?>
                  le class="post">
                   ader>
<h2 class="entry-title"><a href="<?php the_permalink(); ?>">
                <h
                   <?php the_title(); ?></a></h2>
                </
                <div class="content">
                    <?php the_content(); ?>
                </div>
                Posted on <?php the_date(); ?> by <span</pre>
                class="author"><?php the_author(); ?></span>
            </article>
            <?php e
                        le; else: ?>
                No posts are found.
            <?php endif; ?>
        </div> <!-- end content -->
            © <?php the_date(); ?>
website by <?php the_author(); ?>
              :er>
        </
                end wrapper -->
    </
</
</
```

Extra work if there is time

The files for this exercise are in category_and_sidebar.zip

Footer.php

The file **footer.php** is accessed from **front-page.php** and other pages using:

<?php get_footer(); ?>

Inside the **footer.php** folder we added a new menu for the footer links. To do this we had to register two menus inside of **functions.php**:

```
1 register_nav_menus( array (
2 "main-menu" => "Main Menu",
3 "footer-menu" => "Footer Menu"
4 )
5 );
```

And then used the same loop as the Main Menu in **header.php** with some modifications:

```
1 <nav class="small-12 large-3 columns">
2 <?php $footer_menu = array(
3 'theme_location' => 'footer-menu'
4 ?>
5 <?php wp_nav_menu($footer_menu); ?>
6 </nav>
```

7

We also used three different instances of the **WP Loop**, using different queries to populate different sections of the prototype. We introduced new arguments for the query, such as posts_per_page.

```
1
    <?php
2
      $args = array(
3
         "category name" => "featured",
4
         "posts_per_page" => 3
      );
5
      $featured = new WP Query($args);
6
    ?>
7
1
    <?php if(have posts()) : while($featured->have posts()) : $featured->the post();
2
      <div class="footer-link small-12 large-3 columns">
3
        <?php the post thumbnail(); ?>
      </div>
4
    <?php endwhile; ?>
5
    <?php endif; wp reset postdata(); ?>
6
```

More WordPress – single.php, category.php, category.php, page.php,

sidebar.php

We're going to go over single.php, category.php, page.php and sidebar.php.

category.php can be used like **front-page.php**, to create an overview of posts, so we'll start there. This time, duplicate **front-page.php** and change the name to **category.php**.



You can use categories to organize your portfolio into different sections, depending on medium, discipline or other factors.

To show which category is being viewed, use the single_cat_title() function:

```
<?php echo single_cat_title(); ?>
```

This will print the name of the category using the echo function, which outputs a given string. If we want to include more information about the category, we can create conditional statements to determine what category is being called and add contextual content.

```
1
2 <?php if (is_category('Web')) : ?>
3 <h2>Web Design</h2>
3 A sample of previous web design work.
4 <?php elseif (is_category('Photography')) : ?>
5 <h2>Photography</h2>
6 <?php endif; ?>
```

This is the first time we have seen the elseif conditional. This is used to evaluate a series of conditional statements.

For the WP loop, we no longer need the query for featured posts that was being used in **front-page.php** (unless you have another set of featured posts you want to use for each category). So we can revert to the original WordPress loop:

```
1 <?php if (have_posts()) : while(have_posts()) : the_post(); ?>
```

Those are the primary changes to make. You may need to add or remove HTML content from the loop if you plan to change the style of the category page.

single.php and page.php.

Let's take a look at the <u>WordPress Template Hierarchy</u> again and talk about the difference between these two templates. Okay, so **single.php** will be used for a single blog post, while **page.php** will be used for a static page. This time I'm going to duplicate **index.php** again, because I want see the full content of the pages and posts. Depending on how your content is organized, you may not need both of these options, or they might be exactly the same, or they might be different.

So these template files will look almost exactly the same as **index.php** except I want to add custom classes so make it easier to change the style.

```
In index.php I had this after <?php get_header(); ?>:
```

```
1 <div class="posts">
2   <?php if(have_posts()) : while(have_posts()) : the_post(); ?>
3   <article class="post">
```

For **single.php** I'll change it to:

```
1
    <div class="single-post">
2
      <?php if(have posts()) : while(have posts()) : the post(); ?>
      <article class="post">
3
```

And for **page.php**:

```
1
    <div class="page">
2
      <?php if(have posts()) : while(have posts()) : the post(); ?>
      <article class="post">
3
```

The only difference is the CSS classes.

Notice that the posts section is the same in **page.php** and **post.php**. This is confusing, but although pages and posts use different formats, the information is still accessed via the WordPress loop, so the word **post** is still used to describe the content in a page.

For **page.php**, I want to add a featured image. Unlike the thumbnail for the category pages and front page, I want this to be a full sized image. To support this, I need to add something to **functions.php**.



This adds the option of multiple featured images. The

first **add** image size() function is for small thumbnails, the second for a large featured image. You can add as many as you like.

Once we've added this, we have to go back and change the image size in earlier versions, including front-page.php and category.php.

```
<div class="featured-item">
1
```

```
<h3><a href="<?php the permalink(); ?>"><?php the title(); ?></a></h3>
2
```

```
<?php the excerpt(); ?>
```

```
3 <?php the_post_thumbnail('thumb'); ?>
4 </div>
5
```

Then we'll update page.php.

```
1
2 <article class="post">
2 <?php the_post_thumbnail('feature'); ?>
3 <h2><a href="<?php the_permalink(); ?>"><?php the_title(); ?></a></h2>
4 <div class="content">
5 <?php the_content(); ?>
6 </div>
6 </article>
7
```

sidebar.php

Create a blank document and title it **sidebar.php**.

Then add these lines:

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

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

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

```
1 $sidebar = array(
2 'name' => 'Sidebar',
3 'id' => 'sidebar',
4 'description' => 'Place widgets here.',
5 );
6 register_sidebar($sidebar);
```

If we want, later multiple sidebars can be added. Once **functions.php** is uploaded, widgets will appear in the menu.



Then you can dynamically change what appears in the sidebar:

Widgets Manage in Custom	izer	
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.		Sidebar Widgets go in this sidebar.
Archives	Calendar	Search
A monthly archive of your site's Posts.	A calendar of your site's Posts.	
Categories	Custom Menu	
A list or dropdown of categories.	Add a custom menu to your sidebar.	
Meta	Pages	
Login, RSS, & WordPress.org links.	A list of your site's Pages.	
Recent Comments	Recent Posts	
Your site's most recent comments.	Your site's most recent Posts.	
RSS	Search	
Entries from any RSS or Atom feed.	A search form for your site.	
Tag Cloud	Text	
A cloud of your most used tags.	Arbitrary text or HTML.	

Homework / Readings

Comments

https://codex.wordpress.org/Function_Reference/comments_template

Date and Time Formats

https://codex.wordpress.org/Formatting_Date_and_Time

Tags

http://codex.wordpress.org/Function_Reference/the_tags

Author posts

http://codex.wordpress.org/Function_Reference/the_author_posts_link

Posts and Pages <u>https://www.webmechanix.com/how-to-add-posts-to-pages-in-wordpress-tutorial/#guide</u>