Converting a HTML website to Wordpress

What is wordpress? And why use it?

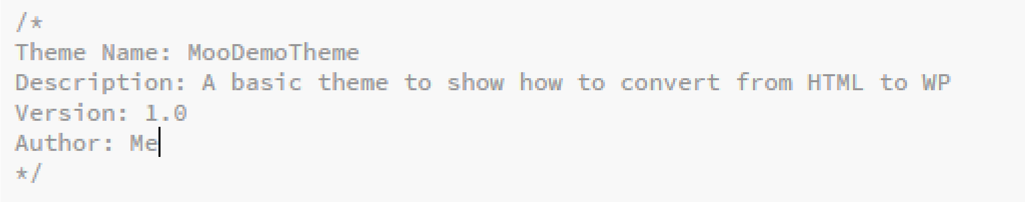
WordPress is a free and open-source content management system based on PHP and MySQL. WordPress is installed on a web server or a network host, there are numerous different options, from hosting yourself to using services like wordpress.com. I think hosting it yourself is easy and most hosts that support php and mysql will support it, with many having the option to install it from within their admin panel. A CMS or content management system allows you to create and organise content easily, in this case the content being web pages or posts. It’s easy to use, powerful, flexible and powers a very very large portion of the web. It’s open source and free to use too!

How wordpress works

Wordpress combines four key technologies, HTML and CSS which you should already know about, PHP which is a server side scripting language used mainly linked with HTML to do things on web sites or applications. This includes bringing information from other sources and displaying it as HTML or changing the structure and content of your pages to respond to users. An example might be pulling text from a database and combining it with some HTML and CSS to display it nicely on the page. Which is exactly what Wordpress does.

First steps

The first step is easy, create a new folder and call it whatever you want your theme to be called. Now, create two new files in that folder, “style.css” and “index.php”. These are your base files for a Wordpress theme, into the style.css file copy the CSS from your original CSS file. I’ve provided a demo CSS file attached to this post. At the top, add this comment into the code;



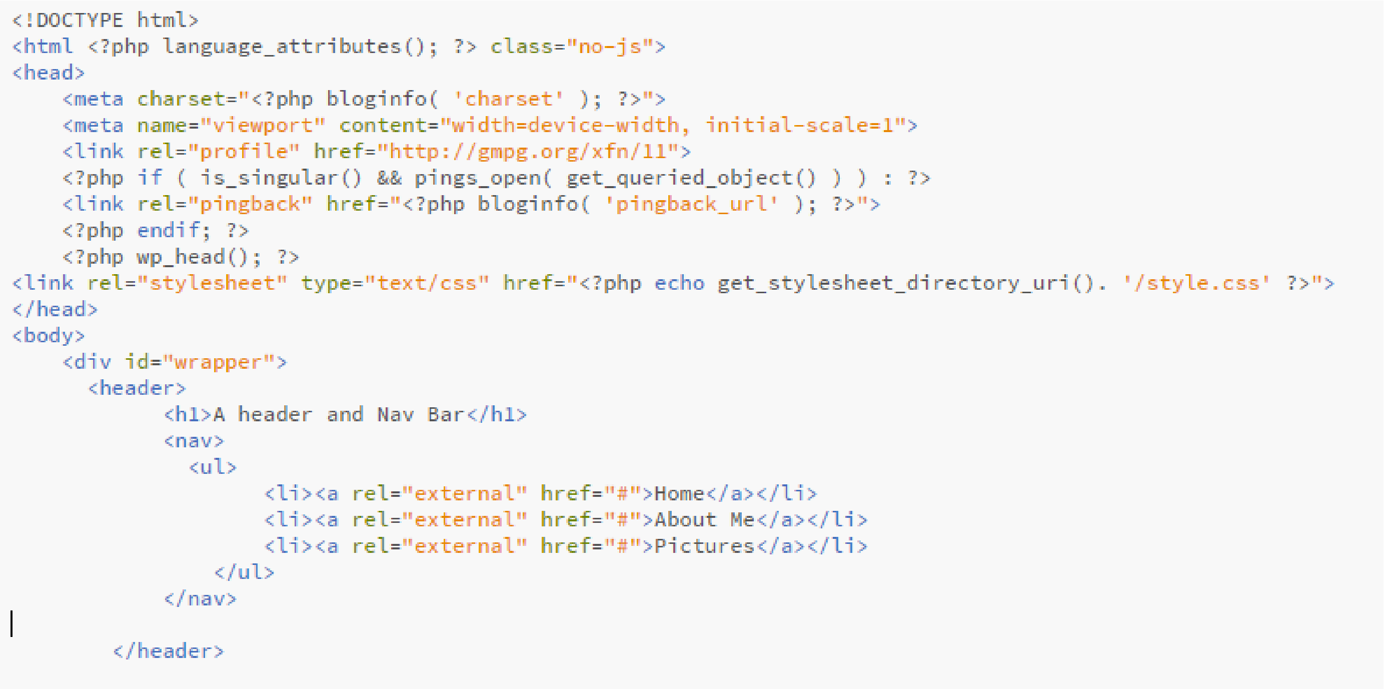
```
/*
Theme Name: MooDemoTheme
Description: A basic theme to show how to convert from HTML to WP
Version: 1.0
Author: Me
*/
```

You can change the name to whatever you want and the Author to yourself.

Dividing it up

Wordpress uses PHP to pull data from a database, so we need to divide up our HTML code to use this to pull in data where we want it. We need to put the code into a new file for each section, the header in it's own file, the content, etc.

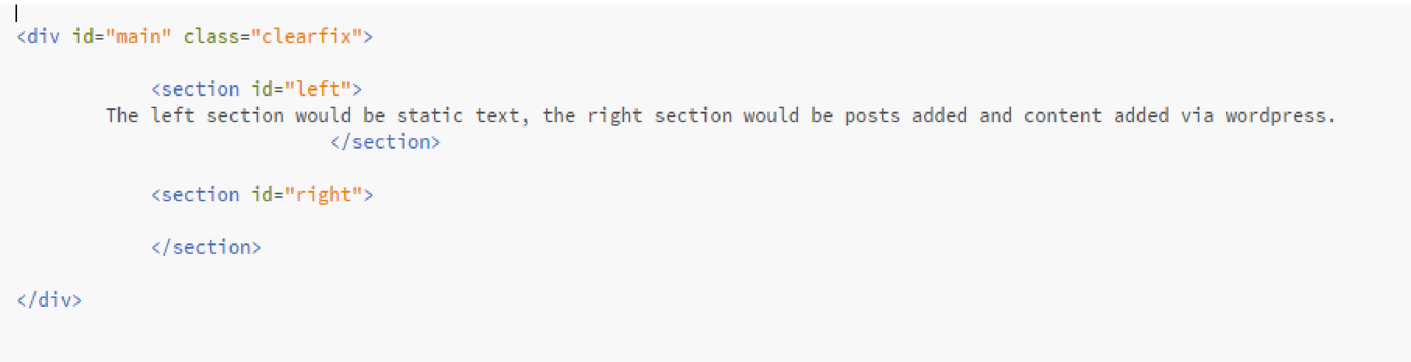
Let's get started, create a new file in the folder you created earlier called "header.php", into it we're going to put our header code, we can literally just chop up the code, so, from the provided HTML you can copy in the following;



```
<!DOCTYPE html>
<html <?php language_attributes(); ?> class="no-js">
<head>
    <meta charset="<?php bloginfo( 'charset' ); ?>">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="profile" href="http://gmpg.org/xfn/11">
    <?php if ( is_singular() && pings_open( get_queried_object() ) ) : ?>
    <link rel="pingback" href="<?php bloginfo( 'pingback_url' ); ?>">
    <?php endif; ?>
    <?php wp_head(); ?>
<link rel="stylesheet" type="text/css" href="<?php echo get_stylesheet_directory_uri(). '/style.css' ?>">
</head>
<body>
    <div id="wrapper">
      <header>
            <h1>A header and Nav Bar</h1>
            <nav>
              <ul>
                    <li><a rel="external" href="#">Home</a></li>
                    <li><a rel="external" href="#">About Me</a></li>
                    <li><a rel="external" href="#">Pictures</a></li>
                </ul>
            </nav>

        </header>
```

That's just the header information, we literally cut off the code after the header.

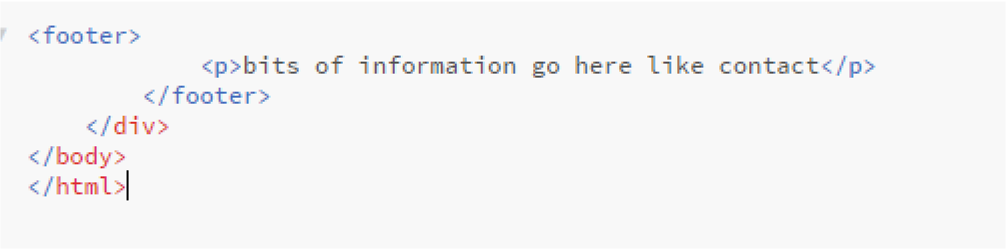
Now, create a file called index.php and add in the following from the HTML;



```
<div id="main" class="clearfix">

           <section id="left">
       The left section would be static text, the right section would be posts added and content added via wordpress.
                         </section>

           <section id="right">

           </section>

</div>
```

Again, this is just the (rather empty) HTML that is used for the main content bit. If you're interested in understanding the HTML and CSS we're using here, check out the HTML5 and CSS3 tutorial. Here we've two sections, the left one will be static text, the same for every post/ page, the right will be posts. Once more we cut off code exactly once the main section is complete, we'll stitch them all together later.

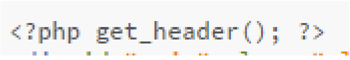
Add finally, the footer info'



```
<footer>
            <p>bits of information go here like contact</p>
        </footer>
    </div>
</body>
</html>
```

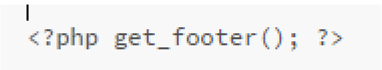
Stitch it together

You should now have 3 different files, header, index, footer, what you need to do now is add some code to stitch them back together. We use some php to do this, open up index.php and add the following;



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

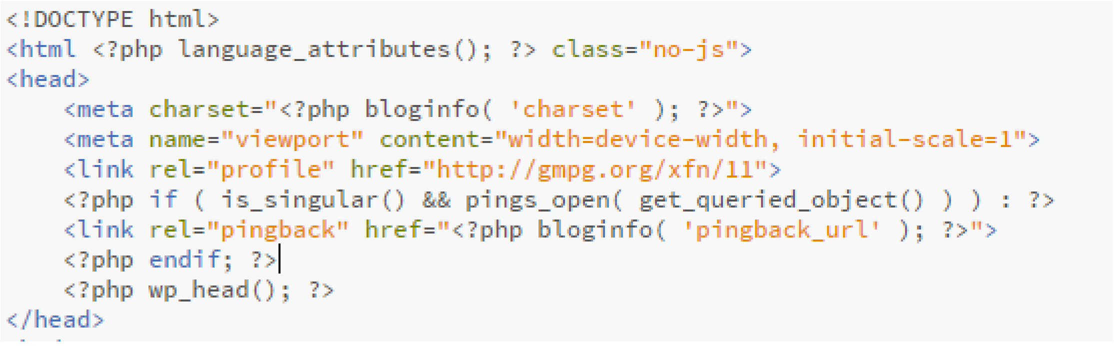
And at the bottom add;



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

These simple commands are PHP that tells Wordpress to fetch the relevant bits and pieces and thus displays them all at once.

In order for them to work, we need to add some code to the head section, open up header.php and add the following.



```
<!DOCTYPE html>
<html <?php language_attributes(); ?> class="no-js">
<head>
    <meta charset="<?php bloginfo( 'charset' ); ?>">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="profile" href="http://gmpg.org/xfn/11">
    <?php if ( is_singular() && pings_open( get_queried_object() ) ) : ?>
    <link rel="pingback" href="<?php bloginfo( 'pingback_url' ); ?>">
    <?php endif; ?>
    <?php wp_head(); ?>
</head>
```

The second line allows you to bring in, from wordpress the language attributes for the country and language you're using.

After the <head> tag, comes the wordpress code, this code establishes the character set, viewing settings and other information for wordpress as well as bringing in and contacting the wordpress server for more information. This especially comes in wp_head() which brings in your style.css.

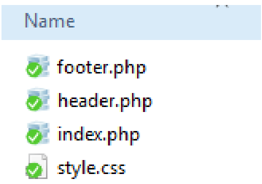
The wordpress loop

The final step is to insert the code that pulls in content from wordpress databases, thankfully wordpress uses PHP to do this too, [“The Loop”](). Copy this into your new index.php file exactly as shown (it needs to be inside the div you’re using to hold your content). I placed mine in <section id=”right”> because that’s where I want all the content pulled in, if in doubt, check the demo files included.

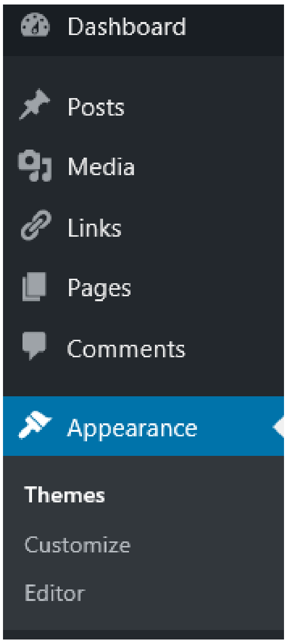
```
<?php if ( have_posts() ) : ?>
<?php while ( have_posts() ) : the_post(); ?>
 <div id="post-<?php the_ID(); ?>" <?php post_class(); ?>>
   <div class="post-header">
      <div class="date"><?php the_time( 'M j y' ); ?></div>
      <h2><a href="<?php the_permalink(); ?>" rel="bookmark" title="Permanent Link to <?php the_title_attribute(); ?>"><?php the_title(); ?></a></h2>
      <div class="author"><?php the_author(); ?></div>
   </div><!--end post header-->
   <div class="entry clear">
      <?php if ( function_exists( 'add_theme_support' ) ) the_post_thumbnail(); ?>
      <?php the_content(); ?>
      <?php edit_post_link(); ?>
      <?php wp_link_pages(); ?>
   </div><!--end entry-->
   <div class="post-footer">
      <div class="comments"><?php comments_popup_link( 'Leave a Comment', '1 Comment', '% Comments' ); ?></div>
   </div><!--end post footer-->
 </div><!--end post-->
<?php endwhile; /* rewind or continue if all posts have been fetched */ ?>
 <div class="navigation index">
   <div class="alignleft"><?php next_posts_link( 'Older Entries' ); ?></div>
   <div class="alignright"><?php previous_posts_link( 'Newer Entries' ); ?></div>
 </div><!--end navigation-->
<?php else : ?>
<?php endif; ?>
```

Setting this up in Wordpress

First you need to make sure wordpress is installed on your web server, this isn’t covered by this tutorial there are TONS of tutorials out there on wordpress setup. Once you’ve got it setup you need to prepare your wordpress style. You should have a folder and inside that there should be four files as shown to the right.

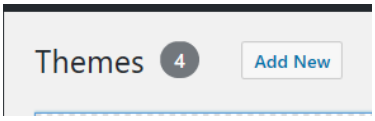


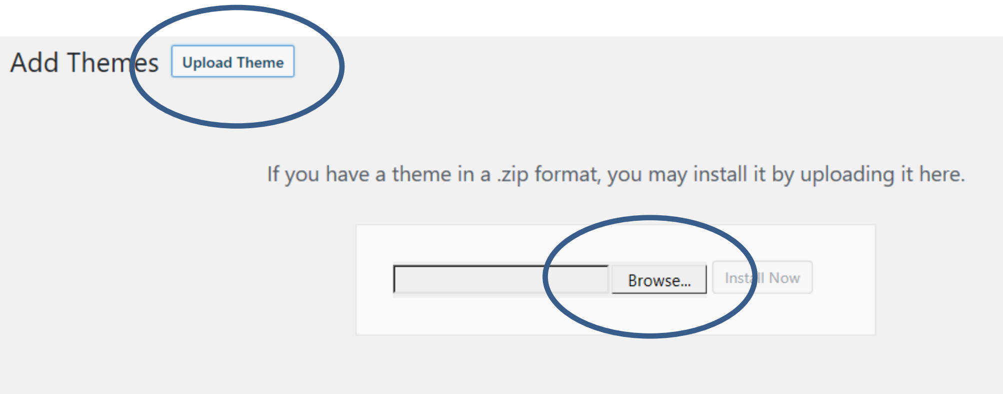
Name
footer.php
header.php
index.php
style.css



These files make up any theme, however we can’t upload them easily individually, instead, we zip the folder they’re in. Once you’re logged into your wordpress site go to Appearance then themes along the left.

Once in themes we’re going to add a new theme, and upload a theme.





If you have a theme in a .zip format, you may install it by uploading it here.

Find where the zip file you just created is saved and double click it, then click install now. All going well you should see a message saying the theme has been installed successfully. Activate it and you should be able to see your changes straight away, you can see what it looks like by going to posts and view.

What now?

This is just the beginning, a very basic introduction, there are plenty of ways to add functionality with Wordpress tags, check out the wordpress codex and it will start a long journey into Wordpress success.