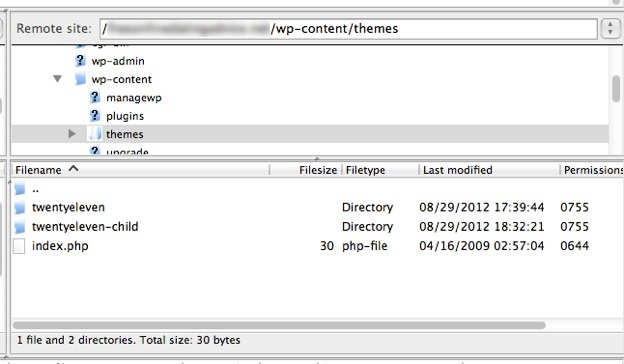
**1. Creating Your Child Theme’s Folder**

Once you have logged into your website via your chosen FTP client, navigate to “*your\_site\_root*/wp-content/themes/”. This directory will contain one or more folders, each of which represents a theme installed on your site.

In this directory, create a new folder. You can name it whatever you like, but I would recommend that you use a descriptive name such as “parenttheme-child” (where “parenttheme” is the name of your active theme).

Here’s an example:



As you can see, my active theme is “twentyeleven”, so I created a “twentyeleven-child” folder.

**2. Creating Your Child Theme’s style.css File**

Once you have created your folder, you need to add the one and only file that is required to create a valid child theme – **style.css**. As you have no doubt guessed, this file will work as an addendum of sorts to your parent theme’s style.css file.

You’ll need to place some vital information inside of this file, so open up your favorite text editor and paste the following into a blank file:

1. */\**
2. *Theme Name: My Child Theme*
3. *Theme URI: http: //mysite.com/*
4. *Description: This is a custom child theme I have created.*
5. *Author: My Name*
6. *Author URI: http: //mysite.com/*
7. *Template: parenttheme*
8. *Version: 0.1*
9. *\*/*

Obviously a lot of the information above is generic, and you can change it to suit your needs. **There is only one variable that you *must* alter in order for your child theme to be valid – “Template”.** This variable is required in order to tell your child theme which parent theme it is related to, and must be changed to match the folder name of your active theme.

There is one other thing of vital importance that you must do. If you were to upload this file in its current state and activate your child theme, WordPress would look to your child theme’s style.css file as the default style file, and see nothing. As a result, your theme would render with absolutely no CSS styling. Therefore, **you need to load the parent theme’s CSS file within your child theme’s style.css file**. You can do this by creating a functions.php file containing code that will load your CSS:

<?php

function enqueue\_my\_child\_theme\_styles() {

wp\_enqueue\_style( 'parent-theme-css', get\_template\_directory\_uri() . '/style.css' ); }

add\_action( 'wp\_enqueue\_scripts', 'enqueue\_my\_child\_theme\_styles' );

?>

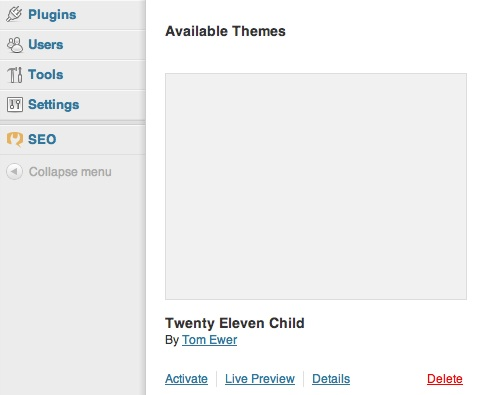
Now when your child theme is activated, WordPress will know to import all of the CSS from your parent theme’s style file before executing any CSS stored in the child theme’s style.css file. Because the CSS contained in your new file will be the last thing WordPress executes, it will take precedence over any preceding CSS.

Once you have added in the above two snippets of necessary information, **save your new style.css file to your child theme’s folder**. Finally, check out our guide to styling your website with CSS.

**3. Activating Your Child Theme**

All that is left to do now is activate your child theme.

Navigate to “Appearance > Themes” in your WordPress admin area. At the top of the screen you will see that your parent theme is active, but you should also see your new child theme listed below:



The information you included in your child theme’s style.css file (such as “Theme Name” and “Author”) will be displayed on this screen.

**Just hit “Activate”**, and your child theme will go live! Your child theme will then be shown as the active theme:

At this stage, **you won’t notice any changes to your theme’s design, because you haven’t actually made any**. However, any future additions to your child theme’s style.css will have an effect upon your site. And as intended, any future update to your parent theme will not affect your child theme’s style.css file.

**4. Making Further Alterations and Additions**

At this point, you may be thinking, “Changing my theme’s CSS is all well and good, but what if I want to make changes to my parent theme’s PHP files?”

Fair question, and the answer is remarkably simple. If you want to make a change to a specific PHP file in your parent theme,**just make a copy of it, and paste that copy into your child theme’s directory**. Any changes you make will go live. This is perhaps most applicable for adding additional functionality via functions.php, but the same process applies to all available PHP files.

**The same process also applies if you want *create* new PHP files**, such as page templates. Just create the new file in your child theme’s directory, include the relevant information, and you’re good to go.

One thing you must bear in mind is that the directory structure in your child theme should match that of the parent theme’s. So if you decide to edit a PHP file in “/parenttheme/includes/external/php/example.php”, you should place it in ”/childtheme/includes/external/php/example.php”.