

Before we start it's important that you understand what a child theme is. Read [this guide from the WordPress Codex](#). Our ThemeSquared Themes comes with child theme so you can use them as your starting point.

Getting Started

To get started we need to prepare or upload a child theme which means you'll need **FTP access** to your host, so you can upload the new child theme. If you don't have this, you should talk to your host so they can give you your FTP login details, and download a FTP program to upload your files.

Making the child theme

First have a look at the stylesheet comes with child theme. This is the only file required to make a child theme. This file contains this code in it:

```
/*
Theme Name: Child ThemeSquared Theme
Version: 1.0
Description: Child theme for ThemeSquared Themes.
Author: ThemeSquared
Author URI: http://themesquared.com
Template: themedir
*/

@importurl("../themedir/style.css");
```

[view rawstyle.css](#) hosted with ❤ by [GitHub](#)

Adding custom CSS

The next step is to activate the child theme by going to our WP admin panel. We are ready to customize custom.css in our child theme folder. The reasons for doing this is that we want our styles to be loaded after the themes styles, since custom.css is loaded last of all theme styles. We've now got all we need to start customizing the theme.

Customizing styles and template files

Currently it doesn't hold any customization, so let's look at a couple of examples on how we can customize the child theme without touching the parent theme. Let's do a simple example where we change the color of the site navigation (find CSS with Firebug or Chrome Web Developer).

Add this to your `/theme-child/assets/css/custom.css`

```
navullia:hover {  
    color: #D2B55B;  
}
```

[view rawcustom.css](#) hosted with ❤ by [GitHub](#)

Editing template files

But wait, there is more! You can do the same with the template files (*.php) in the theme folder. For example if we wanted to modify some code in the header, we need to copy `header.php` from our parent theme folder `/wp-content/themes/porto/header.php` to our child theme folder `/wp-content/themes/porto-child/header.php`. Once we have copied it to our child theme, we edit `header.php` and customize any code we want. The `header.php` in the child theme will be used instead of the parent theme's `header.php`. Neat!

Pluggable Functions

NOTE: The `functions.php` in your child theme should – **be empty** – and not include anything from the parent theme's `functions.php`. The `functions.php` in your child theme is loaded **before** the parent theme's `functions.php`. If a function in the parent theme is **pluggable**, it allows you to copy a function from the parent theme into the child theme's `functions.php` and have it replace the one in your parent theme. The only requirement is that the parent theme's function is **pluggable**, which basically means it is wrapped in a conditional if statement e.g:

```
if ( !function_exists( 'function_name' ) ) {  
    function function_name() {  
        ...  
    }  
}
```

[viewrawfunctions.php](#) hosted with ❤ by [GitHub](#)

If the parent theme function is **pluggable**, you can copy it to the child theme functions.php and modify the function to your liking.