



Lesson #40

Child Themes

By

Lynette Chandler
Tech Based Training

<http://TechBasedTraining.com/>

**NOTICE: You Do NOT Have the Right
to Reprint or Resell this Report!**

You Also MAY NOT Give Away, Sell or Share the Content Herein

If you obtained this report from anywhere other than **Tech Based Training.com** , you have a pirated copy.

Help stop internet piracy by letting me know. Send email to
customerservice@techbasedsupport.com

© 2010 Copyright Lynette Chandler

Legal Disclaimer: Keeping things simple here, I have to say that I cannot promise you success. I can give you direction and advice based on my experiences and good internet marketing practices. What you do with this information is up to you. As a Tech Based Training member you agree to not hold me responsible for your results.

Introduction

In the recent past, there has been a talk about the legality of premium themes. Automattic, the company who is behind WordPress believes in the open source concept and we're thankful for that. Otherwise, we won't have a business at Blog Evangelists.

It is around that time too that theme developers who want to support the open source concept and still feed themselves began dabble in different business models. Child themes is one result of that.

How Do They Work?

The child themes concept is actually pretty interesting. It works on the piggyback concept. The child theme cannot stand alone on its own and needs a host or parent theme to work.

A child theme is basically just a stylesheet. How would that work? Knowing the amount of customization we do to a theme, how can we achieve all those cool stuff with only CSS?

Well, the secret is, it can expand beyond CSS – with some limitations and we'll learn about that in this lesson.

Why Use A Child Theme?

Great question! I've wondered a lot about that until I tried it. Wow! What a concept. The biggest appeal is to be able to customize a theme without editing the original. This is super useful when you're using premium themes that are being updated over time.

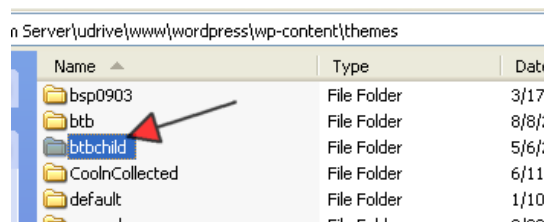
That's one of my first frustrations when it came to building sites with premium themes. I quickly found out that the theme would not work on a newly upgraded WordPress of mine. The theme developer released a newer version but alas! I can't install the newer version without having to re-do all the hours of

customization. Not a good place to be. That's why child themes appealed so much to me.

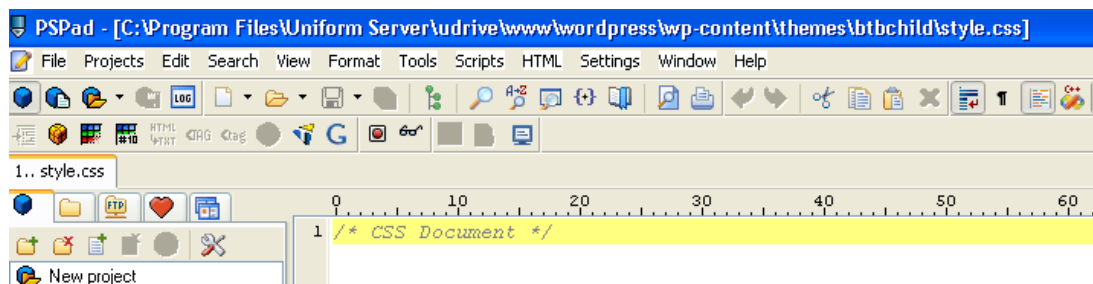
The Makings Of A Child Theme

The best way to demonstrate child themes is to build a child theme for our Blog Theme Bootcamp theme. Your first action is to create a new folder for your theme. To make development easy, I'm going to create the new folder inside my /wp-content/themes folder, yes child themes live in the theme folder too.

I'm going to name it btbchild.



Launch PsPad, create a new stylesheet, name it as style.css and save it inside this btbchild folder.



Inside this style.css, add this:

```
/*  
Theme Name: BlogThemeBootcamp Child Theme  
Theme URI: http://blogthemebootcamp.com
```

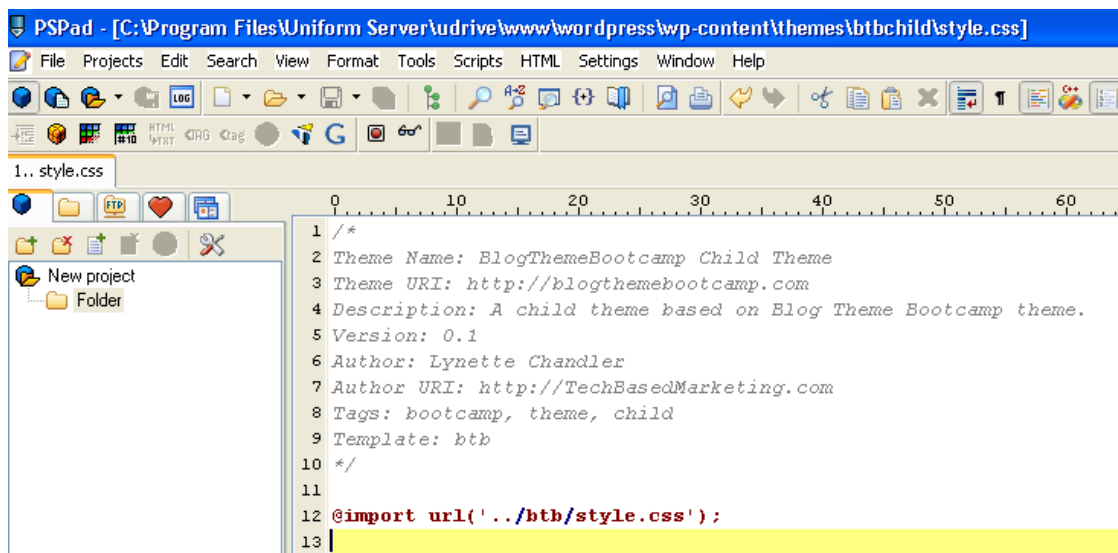
```
Description: A child theme based on Blog Theme
Bootcamp theme.
Version: 0.1
Author: Lynette Chandler
Author URI: http://TechBasedMarketing.com
Tags: bootcamp, theme, child
Template: btb
*/
```

You can change the name, URI etc to yours and your sites or give it a new name whatever you wish. The most important part, I've highlighted in red. Without this line, your child theme won't work and the template must be the same name as the folder name of the parent theme. It is also case sensitive.

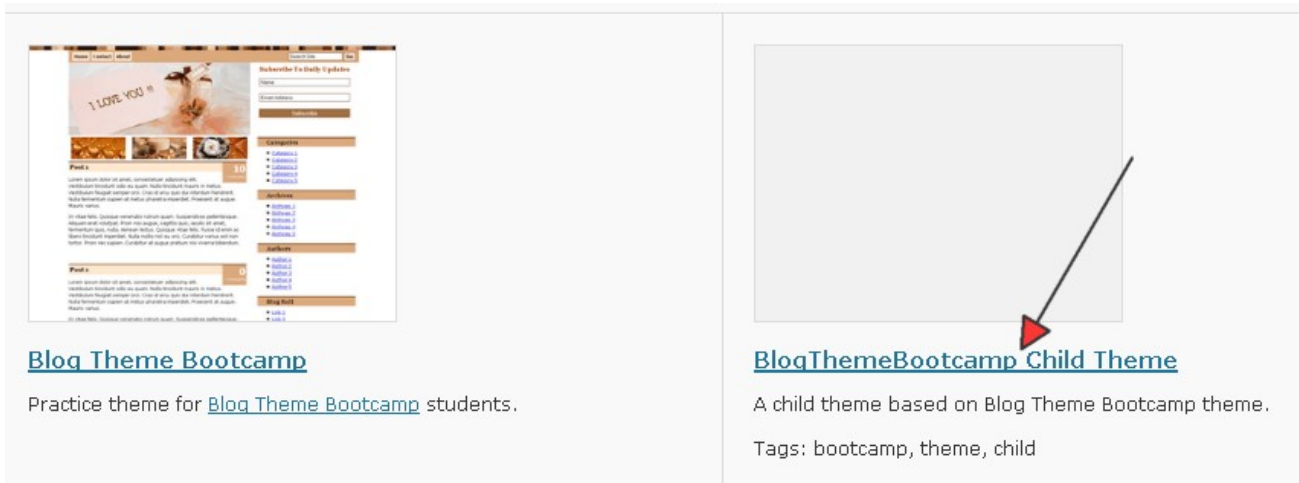
Next, you'll need to import the parent theme's stylesheet in here and it must be the first thing after the code above. When you add or modify any design rules later, put them under this import command, not above. Also, make sure the path to the parent theme is correct or it won't load.

```
@import url('../btb/style.css');
```

What you have is this so far. This is your child theme. Once this is done, you're ready to rock and roll with your new child theme.



Log in to your WordPress admin, go to Appearance >> Themes. You should see the child theme in the list of available themes. Click to activate like any normal theme.



When you look at your blog now, it doesn't seem like anything has changed. Of course not. Because we haven't made any changes yet. But we will. Let's get busy customizing.

Change The Header Ribbon

One of the first things I want to change is that ribbon of colors up on top. Inside the child theme's style.css (/btbchild/style.css), we'll add this.

```
body {  
    background-image: none;  
}
```


The result is this

[Home](#) [About](#)


This is a post title

by admin on August 28, 2008

0
Comments

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum tincidunt odio eu quam. Nulla tincidunt mauris in metus. Vestibulum feugiat semper orci. Cras id arcu quis dui interdum hendrerit. Nulla fermentum sapien at metus pharetra imperdiet. Praesent at augue. Mauris varius.

In vitae felis. Quisque venenatis rutrum quam. Suspendisse pellentesque. Aliquam erat volutpat.



Subscribe To Daily Updates

Categories

Next change I'd like to make is a border around the 'page'.

Page Border


To achieve that, I confirm using Firebug that the 'page' is actually a DIV with the ID of container.

[Home](#) [About](#)


This is a post title

by admin on August 28, 2008

0
Comments

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum tincidunt odio eu quam. Nulla tincidunt mauris in metus. Vestibulum feugiat semper orci. Cras id arcu quis dui interdum hendrerit. Nulla fermentum sapien at metus pharetra imperdiet. Praesent at augue. Mauris varius.

In vitae felis. Quisque venenatis rutrum quam. Suspendisse pellentesque. Aliquam erat volutpat.



Subscribe To Daily Updates

Categories

- [Dresses](#)
- [Gifts](#)
- [Toys](#)
- [Uncategorized](#)

Archives

- [February 2009 \(1\)](#)
- [August 2009 \(1\)](#)
- [April 2008 \(2\)](#)

Authors

- [admin](#)

script: File | **div#container** | body | html

HTML CSS Script DOM Net

Options

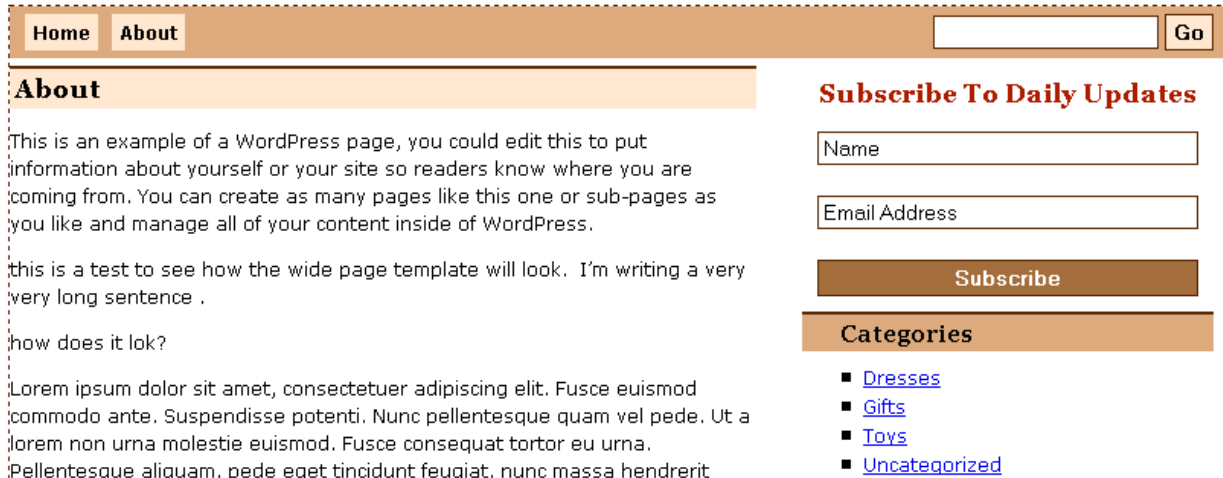
Style Layout DOM

<div id="container">
 <div id="topnav">
 <div id="context">

So in the child theme, I'll add this.

```
#container {  
    border: #541E04 dashed 1px;  
}
```

Now I'll have a nice dashed border around my page.

A screenshot of a WordPress page titled "About" with a dashed border. The page has a navigation bar with "Home" and "About" links, a search bar, and a "Go" button. The main content area is titled "About" and contains several paragraphs of placeholder text. To the right, there is a "Subscribe To Daily Updates" section with input fields for "Name" and "Email Address", a "Subscribe" button, and a "Categories" section listing "Dresses", "Gifts", "Toys", and "Uncategorized".

Home About

About

This is an example of a WordPress page, you could edit this to put information about yourself or your site so readers know where you are coming from. You can create as many pages like this one or sub-pages as you like and manage all of your content inside of WordPress.

this is a test to see how the wide page template will look. I'm writing a very very long sentence .

how does it lok?

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce euismod commodo ante. Suspendisse potenti. Nunc pellentesque quam vel pede. Ut a lorem non urna molestie euismod. Fusce consequat tortor eu urna. Pellentesque aliquam, pede eget tincidunt feugiat, nunc massa hendrerit

Subscribe To Daily Updates

Name

Email Address

Subscribe

Categories

- [Dresses](#)
- [Gifts](#)
- [Toys](#)
- [Uncategorized](#)

Hmmm the text is now all pushed to the side, we don't want that so obviously that needs changed. Using Firebug, I find out that the post is contained inside a DIV with the class named post. Armed with that information, I'm going to write in my child theme's stylesheet.css the following.

```
.post {  
    padding-left: 10px;  
}
```


[Home](#) [About](#)

About

This is an example of a WordPress page, you could edit this to put information about yourself or your site so readers know where you are coming from. You can create as many pages like this one or sub-pages as you like and manage all of your content inside of WordPress.

this is a test to see how the wide page template will look. I'm writing a very very long sentence .

how does it lok?

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce euismod commodo ante. Suspendisse potenti. Nunc pellentesque quam vel pede. Ut a lorem non urna molestie euismod. Fusce consequat tortor eu urna. Pellentesque aliquam, pede eget tincidunt feugiat, nunc massa hendrerit magna, non ultricies neque lectus nec dui. In hac habitasse platea dictumst.

Subscribe To Daily Updates

Categories

- [Dresses](#)
- [Gifts](#)
- [Toys](#)
- [Uncategorized](#)

That's much better!

Even though we've only changed a few things so far, I'm pretty sure you can see how a child theme can be of use to you. Go ahead. Modify the child theme. Add style rules, change the colors, change the width of the sidebars. Test it out.

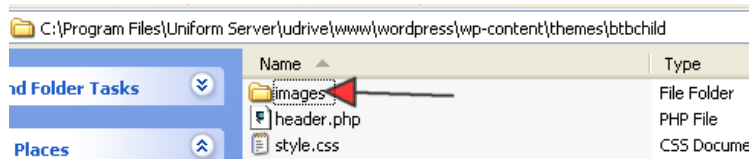
If you're nervous or don't know how to write your own CSS rules, just open the parent's style.css, find the part you want to edit, copy and paste the rule into your child theme's style.css then edit it there. It's allowed.

Modifying Theme Files

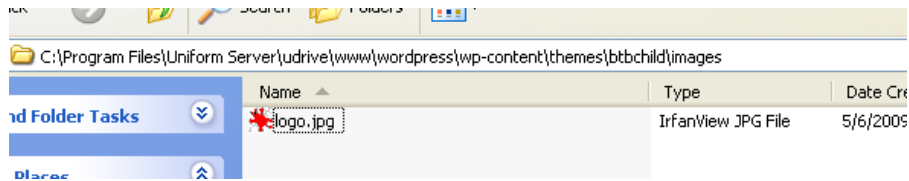
All this is great. But what happens if I want to add HTML or something to the theme files, not the stylesheet? Thankfully since WordPress 2.7, you can do that. Child themes can overwrite the parent. If you're using a version less than 2.7, this won't work. Here's how you do it.

Let's say I want to add a graphic header on top of the navbar. I first need to clone/copy the header.php from the parent theme into my child theme folder.

Then, I create a new folder inside the child theme folder named images. It's the same structure as if I were building a new theme. I'm creating this folder because I want to put my header graphic in it.



Now, copy your header graphic or logo into this images folder.

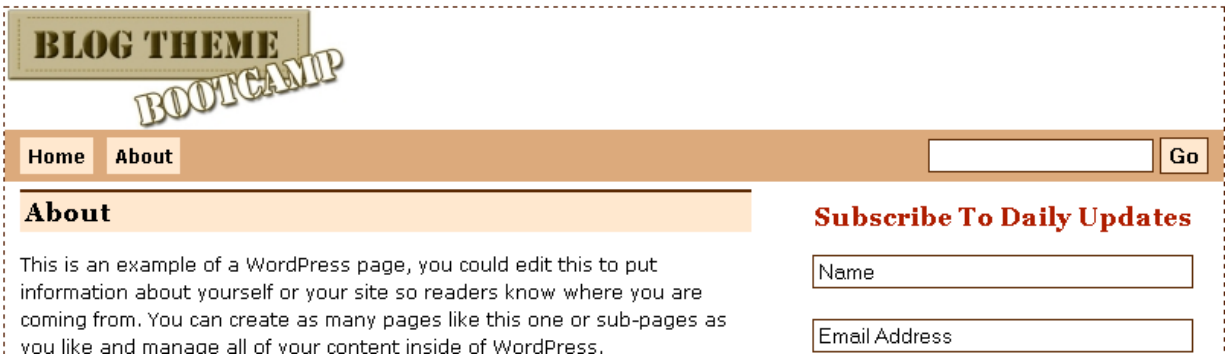


Now that I have all the images in place, I edit the child's header.php to add...

```
<div id="header">
<a href="<?php bloginfo('url'); ?>"></a>
</div>
```

You'll notice one thing in the code above (in red), when calling the image URL, I did not use `bloginfo('template_url');` why? Because WordPress still bases the theme (template) off the parent but the stylesheet is based on the child.

Now when I view my site, I get this:



Pretty awesome huh?

Conclusion

I've demonstrated just enough about child themes for you to have a working understanding how this works and how it can be useful to you. Using child themes, you can copy the templates you want to edit into your theme, any template and modify to your hearts' desire without messing up the original theme.

This is also a good way to work on clients' sites who have existing themes but they only want small changes. It allows you to keep the original theme around as a 'backup'.

There are some limitations to using child themes. Not all themes are suited for child themes although many or most are. But if you use a theme like [WPRemix](#) that has a lot of programming built into the theme, you'll likely run into problems as I did. Otherwise, this is a great tool for a majority of themes out there.

For your benefit, here is the Blog Theme Bootcamp child theme I have created based on this lesson. [Download](#) it, check it out, have fun with it.

Next Lesson...

We're going to dive deep into the database using phpMyAdmin

Lynette Chandler

Lynette Chandler

Tech Based Training