



February 16, 2010

When CSS = Cursing at this Stupid Stuff

For all the wonderful talk of how much you can do with Thesis without knowing any code, I'm willing to wager that anyone who has paid money for a WordPress theme isn't going to take long before deciding they want to do even more customizing of their site's design than what's available in the Design Options panel. You know, crazy stuff. Like bolding text. And then you're going to need to delve into your custom.css file. This issue of the newsletter is going to provide some tips on working with CSS, with a goal of minimizing the resulting profanity. :)

WHAT IS THIS CSS THING OF WHICH YOU SPEAK?

CSS, which stands for Cascading Style Sheets, was created to help people define different styles, much like those in layout and word processing programs like Adobe InDesign and Microsoft Word. The styles then let you a) apply design attributes to specific content elements with greater precision than basic HTML and b) allow you to update those style definitions once and have them be applied automatically to all the elements that use them.

So, for example, you could define one style that says what border color your post photos should have.

Then, six months down the road, when you've written dozens of posts already, you don't have to go into each and every one to change the color. Instead, you can make a single change in your custom.css file and voila! All your photo's borders get changed at once.

Please note that this newsletter is not intended to be a tutorial on how to write CSS from scratch. If you have never seen or used CSS and want a really good primer, check out w3schools.com. It not only explains the concepts really well, it has tons of "Try It!" examples where you can make minor changes to their code and immediately see what the

```
20 .custom #container {
21   width: 950px;
22 }
23
24 .custom .menu {
25   width: 950px;
26   margin-bottom: 0;
27   padding-bottom: 2px;
28   background-color: #f6f6f6;
29   color: #418119;
30 }
31
32 .custom .menu li.tab a,
33 .custom .menu li.cat-item a {
34   background-color: #f6f6f6;
35   color: #418119;
36   font-size: 14px;
37   text-transform: none;
38   letter-spacing: 0;
39 }
```

result is. I go back to them all the time to remind myself how a specific attribute, like position: absolute, is supposed to work (which, in turn, helps me figure out why mine isn't!).

And if you REALLY want to get to know your CSS, check out [60 Excellent CSS3 Tutorials and Techniques You Should Know](#) on Themeflash.

FINDING THE RIGHT SELECTOR

There's very little as frustrating as making changes to your CSS file only to have nothing happen to your actual page. There are lots of reasons this can happen, but one of the more common ones is that you're targeting the wrong *selector*. **Selectors** can be standard HTML tags, like body, h2, and p, or they can be user-defined **IDs and classes**. Many of these are predefined in Thesis, like sidebar_1 and teaser, that you can overwrite in custom.css. You can also create your own selectors when building a custom function.

It's easy to pick the wrong selector, especially for elements that have a lot of styled aspects to it. A good example is the Thesis dropdown menu system, which is built using a list structure. Do you need to modify .menu or ul.menu or .menu li.tab...? Here are some suggestions on how to figure out which selector applies to which element.

Look at the source code: Right-click on your page and select View Source or View Page Source from the menu. Here you'll see the code for the rendered HTML and you can look for the element you're trying to style and see what classes are affecting it.

```
<li class="widget widget_text" id="text-5"><h3>Receive Updates</h3>
```

Add a background color: Think of each element within the page as a box, even if it's text. By giving it a background color, you can see if you've got the right box. I like to use a light gray color that usually doesn't clash with text or images within the box. For example, the code below shows me I've correctly targeted the post headline.

```
.custom .headline_area h1 { background-color: #eee; }
```

Main Menu Item

If you're already using a background color for that element and don't want to replace it for testing purposes, you can also use a 1-pixel border. However, be aware that those pixels are added to the size of the element, which can cause layout problems. For example, if you put a 1-pixel border around a sidebar that's 300 pixels wide, it's now going to act like it's 302 pixels wide. If your main content area and your sidebars were exactly butted up beside each other, this could cause your entire sidebar area to be pushed down the page. On the plus side, you'd know you've got the right element! :)

Go big or go home: When you're trying to adjust the dimensions of something using properties like width, margin, or padding, if you just add 1 or 2 pixels to the value, it can be hard to tell whether you're making an impact on it or not. Instead, try using 40 pixels just to see what the effect is. If you don't see an obvious change, you're probably not targeting the element correctly.

TROUBLESHOOTING CSS

Even when you've got the right selector, there are things that can prevent your CSS code from working correctly. Here's a small sampling. You can find more things that can go wrong with CSS on the [T4L Thesis Troubleshooting page](#).

Syntax, syntax, syntax: Did I mention syntax? Miss a semi-colon or use a round bracket instead of a curly one and it's not going to work. If something's not working, there's a good chance it has to do with syntax.

Mixing up classes and ids: I do this all the time. If the element you're targeting is assigned a class, then the definition uses a period in front of it. For example, where the HTML in the section above shows `<li class="widget">`, then the CSS needs to be:

```
.custom li.widget { width: 100px; }
```

Where it shows `<li id="text-5">`, it uses a hash mark instead:

```
.custom li#text-5 { width: 100px; }
```

Note that the "li" part of the selector doesn't have a period or a hash mark in front of it. That's because it's a standard HTML tag, not a user-created class or id.

Forgetting the "a": When text is a hyperlink, the HTML code usually looks something like this:

```
<p class="subheading"><a href="http://www.techforluddites.com">Tech for  
Luddites</a></p>
```

If you try to change one of the text attributes of Tech for Luddites, like font-family, font-weight, color, line-spacing, etc., using just the subheading class, it's not going to take. You need to further specify that it applies to that class with an anchor tag, like this:

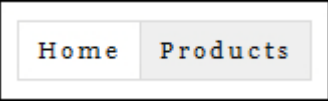
```
.custom .subheading a { font-size: 13px; }
```

Here's a great article that was fortuitously posted a couple of days ago on Smashing Magazine: [The Definitive Guide to Styling Web Links](#)

JUST THE CODE, PLEASE

Removing All Caps from Menu Tabs

I'm not quite sure why they made this the default, since you can easily create the effect by retyping the menu items in all caps in the Thesis Options > Navigation Menu panel in WordPress. I also see lots of people asking how to "turn it off" on the forums. Fortunately, it's an easy fix.



Home Products

In custom.css

```
.custom .menu a {  
text-transform: none;  
}
```

HOT SITE SPOTLIGHT



A Life in Translation

This visual candy bowl of random musings was designed by the site's owner, San Francisco dating columnist and web designer Jamie Varon. (Her company is **Shatterboxx Media**.) Nothing in this look is standard—even the Thesis affiliate ad made me smile. And I couldn't fit it in the screenshot, but check out the footer over there—wild and widget-free!

You can follow Jamie on Twitter at [@jamievaron](https://twitter.com/jamievaron).

RECENT POSTS

[Format Previous Entries Pages Like Your Home Page](#)

[QuickTip: Adding Thumbnails to Your Teasers](#)

[Applying Functions Only to Specific Pages](#)

GET THE STARTER PACK



This **15-page eGuide** offers explanations of the basic Thesis concepts, step-by-step instructions, and code samples for several popular customizations, including adding a clickable header to your site.

The Starter Pack is free to anyone who purchases Thesis through the Tech for Luddites site. The offer applies whether

you're purchasing a Personal or Developer option, or upgrading from Personal to Developer.

"I LOVE Thesis Theme Starter Pack. It answers some simple but important questions that I couldn't find the answer to anywhere else."

– Yael K. Miller, Co-Founder of [Miller Mosaic, LLC](#)

If you've already purchased Thesis elsewhere, you can also purchase the guide for \$9.95.

Get the Starter Pack today!