



Lesson #18

Cleaning Up

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.

Foreword

Since we started building the theme, there are bits and pieces of things I've left out in order to keep the focus of the lesson at hand. Sometimes they are left out because it was better to do it at the end, after all the template files are in place. Saves us from going back and forth. The more complex the theme, the more of these little bits and pieces you will find. So it is always a good idea to jot down these to do's you should come back to as you work.

I've got this short list in front of me and that's what we will be working on in this lesson. As usual, please make sure you have launched all the necessary tools, start your server or log in to your web host with PsPad.

Fixing Up The Title Tag

In PsPad, open header.php of the theme we've been working on. Take a look at the title tag.



```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
2 <html>
3   <head>
4     <meta http-equiv="content-type" content="text/html; charset=windows-1250"
5     />
6     <title><?php bloginfo('name'); ?></title>
7     <link rel="stylesheet" type="text/css" href="<?php bloginfo('
stylesheet_url'); ?>" />
```

The beautiful thing about WordPress is, you can make it super simple like what we already have, or make it 'dynamic', changing with each area the visitors goes. Right now, the name of the blog is displayed in the title no matter what page we are on. What we want is:

- When viewing a post/Page – display the post/Page title and the blog name

- When viewing archives/categories – display archive/category name and blog name
- When viewing the home page – display only blog name

Good thing for us, WordPress developers had already thought of this and also recently expanded the wp_title tag to be more useful.

In front of the blog name tag (bloginfo) we will add this tag.

```
<?php wp_title('-',true,'right'); ?>
```

So our title tag looks like this now.

A screenshot of a code editor showing HTML code. Line 5 contains the title tag: `<title><?php wp_title('-',true,'right'); ?> <?php bloginfo('name'); ?></title>`. A red box highlights the PHP code inserted into the title tag. Line 6 shows a link tag for the stylesheet: `<link rel="stylesheet" type="text/css" href="<?php bloginfo('stylesheet_url'); ?>" />`.

```
4 <meta http-equiv="content-type" content="text/html; charset=windows-1250"
  />
5 <title><?php wp_title('-',true,'right'); ?> <?php bloginfo('name'); ?></
  title>
6 <link rel="stylesheet" type="text/css" href="<?php bloginfo('
  stylesheet_url'); ?>" />
```

Ok, allow me to explain the tag a little bit. The '-' dash is a text separator. It tells WordPress to add a dash to the title of the post, the archive or category name. 'True' means yes you want to display the post title and 'right' tells WordPress to put the dash at the end of the title. This makes sense because we want the title of the post to appear before the blog name for SEO purposes.

You can put anything to replace the dash. You can type two dashes, a pipe character (|) , slash (/) , even text like 'from' or special characters like »

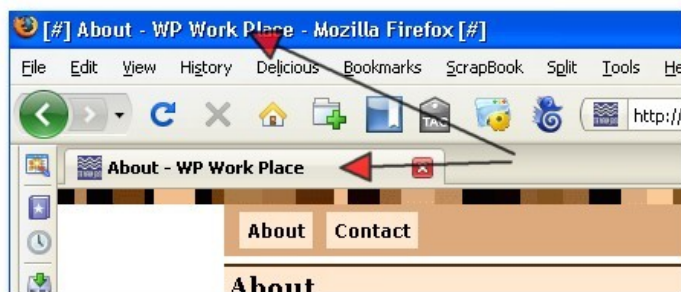
You can also turn it off by changing 'true' to 'false' and move the separator from 'right' to 'left'. The full explanation of the wp_title tag [is here](#) if you want to investigate further.

Now that I have my code in there, let's see how it looks like.

In the home page



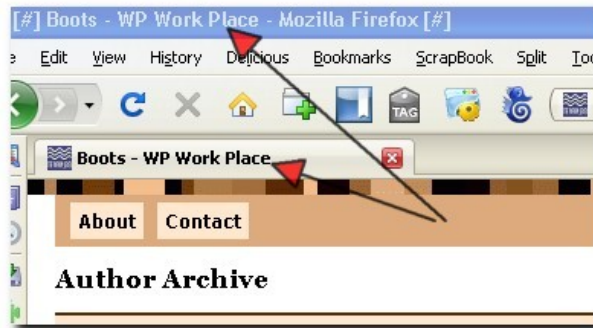
In a Page



In the Archive



In the Author's archive



In a Category



Just what we wanted.

Top Navigational Menu

If you notice in the navigational menu on the top of this theme, I have About & Contact listed. This might be different for yours depending what Pages you have created. If you have created a ton of Pages, his menu is probably overflowing and possibly pushed out of whack. We will get to that in a later lesson where I can fully explain it.

We want to focus on adding a 'Home' link here, before 'About' and 'Contact'. This is because, as we pace through our site, there seems to be no easy way to get back to the home page. In 99% of the themes I built, the theme usually has a header graphic and that is what I normally link to the home. This makes good sense. In

my personal observations, header images get a good number of clicks because people get used to the fact that it (usually) takes them back to the home page.

In this theme design however, there is no header graphic. What looked like the header graphic only appears on the home page since we differentiated that from the other pages. Correcting this is easy. Still in header.php, this is the code we have to date.

```
16 <input type="submit" name="submit" value="Submit" />
17 </form>
18 </div>
19
20 <ul>
21 <li><a href="<?php bloginfo('url'); ?>">Home</a></li>
22 </ul>
23 </div>
24
```

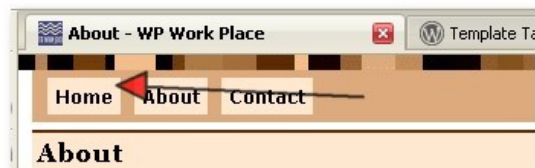
Grab this code, put it before wp_list_pages.

```
<li><a href="<?php bloginfo('url'); ?>">Home</a></li>
```

The result should be like this.

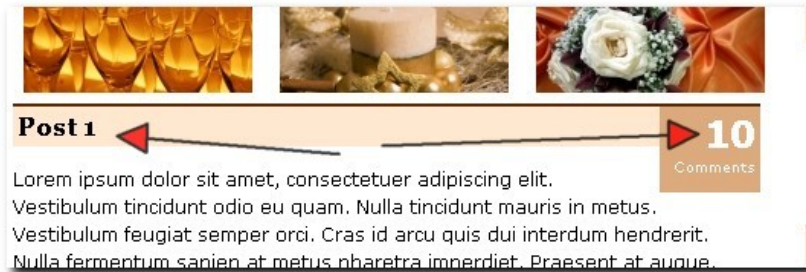
```
18 </div>
19
20 <ul>
21 <li><a href="<?php bloginfo('url'); ?>">Home</a></li>
22 <li><a href="<?php wp_list_pages('title_li='); ?>">Pages</a></li>
23 </ul>
24 </div>
25
```

After saving your work, go check it out. Click through a few pages. 'Home' should appear everywhere.



Post Title & Comment Number Style

If you look back at our original HTML template – the one before we 'WordPressed' it, the post title and comment numbers were stylized to fit into the design.



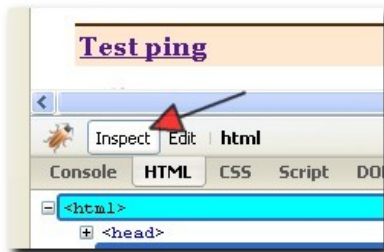
But since we made a theme out of it, they look like normal blue/purple links.



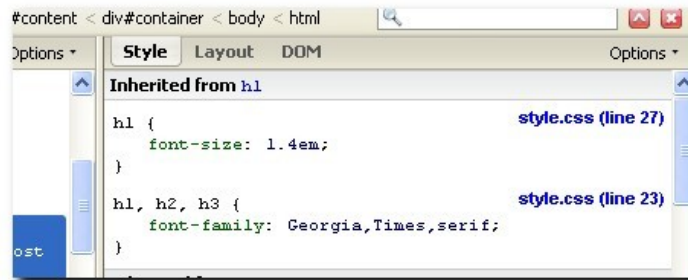
If you've been around Internet marketing circles, you have probably been told that you should leave links blue. I concur but only for certain things. Like links in a post and even sidebar links depending on the design. Most people by now know or are in the habit of clicking on post titles expecting them to be links.

This is just my preference. If you or your client prefer them the default blue then you need not change anything but if you think like me, here's where we edit the style sheet to fix that.

Let's find out which style is used to control the title and the comment number. In Firefox, launch Firebug. Click Inspect in the Firebug Toolbar and then click on the post title.



Look at the right pane on Firebug. You should see something like this.



This tells me that the Post title is styled using styles from h1, h2, h3 and if you scroll down, you'll see #container division styles and the body style. These are 'inherited' styles.

If you are familiar with CSS, web building or WordPress, you are probably wondering where is the 'a' style? The 'a' style is the one that usually tells the browser to style links. But in this case there is no 'a' style. Reason? Because there is none defined.

This is by my design. Many theme developers like to set an 'a' (link) style for the whole theme up front. I don't because I am a marketer first, developer 2nd. Marketers like their links to be blue underlined. Designers like things to 'look' good.

If you set the link style early in the style sheet, this style will carry through all links in your theme which means you will have to write out a new link style for every link you want to change back to the default blue. This is very frustrating for people who customize themes because sometimes they change one area thinking it will change all the links but it may not.

That to me is a lot of work which is why I like to work backwards and don't define link styles in the beginning of the theme and only set them on the elements that I want to apply to. That way, all links by default is blue unless I change it.

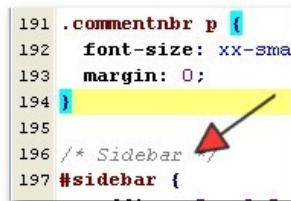
So when you find a theme that does not have a 'a' style, that's OK you can always add to it. That's what we will do now. Open up style.css in PsPad.

If you scroll down and take a good look at the style sheet, you will see I have separated it clearly into 'sections'. To me, this makes sense because it's easier to find the styles later on. There is a section there I've named Content. Viewing it in Pspad, it should be grey like this.



```
136 }
137
138 /* Content */
139 ul#imgnav {
140     margin: 5px 0;
141     padding: 0;
142     list-style-type: none;
143 }
144
145 #imgnav li {
146     margin: 0 7px;
```

In my style sheet it begins on line 138 and goes all the way down to line 195. When you reach the Sidebar section you know that is the 'end' of the styles applied to content. This is not set in stone. When you are building your theme you can organize this any which way you want, you can put the link style at the bottom of the style sheet. It doesn't matter as long as the code is correct.



```
191 .commentnbr p {
192     font-size: xx-small;
193     margin: 0;
194 }
195
196 /* Sidebar */
197 #sidebar {
```

Because I like to keep things organized in compartments like this and the post title does belong within the content area of the theme, I'm going to add it in this area. Looking through the styles I also see a .post h1 style already defined. It styles the background color and borders of the post title. I'm going to add my code right under this style because they 'belong' together.

```
.post h1 a {  
  color: #000000;  
  text-decoration: none;  
}
```

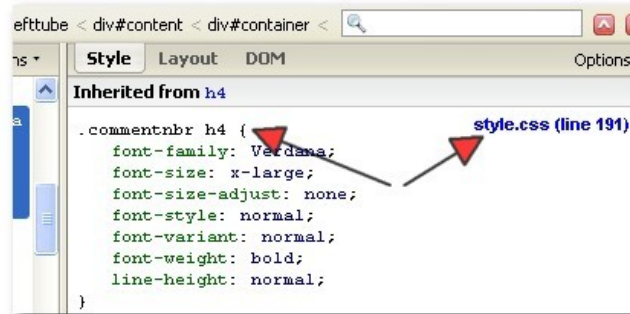
This tells the browser to make **only** the link (a) that is enclosed in h1 tags and found in the post division black, without underlines.

```
161 .post h1 {  
162   padding: 3px;  
163   margin: 0;  
164   background-color: #FFEB3F;  
165 }  
166  
167 .post h1 a {  
168   color: #000000;  
169   text-decoration: none;  
170 }
```

Save your style sheet, refresh your browser. Your post titles should now look like this.



Now let's move on to edit the comments number. Once again, check the style being used using Firebug. In Firebug's right pane you should see something like this but once again no 'a' style.



The good news is, you see that .commentnbr h4 is in style.css line 191 so we can go straight to that line and add the following code under it.

```
.commentnbr h4 a {  
color: #ffffff;  
text-decoration: none;  
}
```

```
191 .commentnbr h4 {  
192     font: bold x-large Verdana;  
193     margin: 0;  
194 }  
195  
196 .commentnbr h4 a {  
197     color: #ffffff;  
198     text-decoration: none;  
199 }
```

Save style.css, refresh your browser. Should look like this now.



Newsletter Subscription Box

When we first started building the theme, the newsletter subscription box appeared through out all the pages on the blog. But once we started adding template files and customizing it, the box went away because it was located in the

feature area. Now we want to add that back but we are going to add it into the sidebar.

To do this, we will use conditional coding. Why? Because if we add the box to sidebar without conditions, you will end up with two of them on the home page. That's not what we want.

Open feature.php in PsPad. We need to grab the subscription box code. Copy the code and set it aside.

```
7 <div id="righttube">
8   <div id="subscribe">
9     <h1>Subscribe To Daily Updates</h1>
10    <form action="#" method="post">
11      <input type="text" name="" value="Name" /><br />
12      <input type="text" name="" value="Email Address" /><br />
13      <input type="submit" name="" value="Subscribe" class="button" />
14    </form>
15  </div>
16 </div>
```

Copy This

Open sidebar.php. Before the for Categories, make a space there.

```
1 <div id="righttube">
2   <div id="sidebar">
3     <ul>
4       <li>
5         <h2>Categories</h2>
6         <ul>
7           <?php wp_list_categories('title_li=&hide_empty=0'); ?>
8         </ul>
9       </li>
10      <li>
11        <h2>Archives</h2>
```

Insert line space here

Type into the space like this.

```
2 <div id="sidebar">
3   <ul>
4     <li>
5     </li>
6     <li>
7       <h2>Categories</h2>
```

Type this in the space

Now paste the subscription box code between those two.

```
2 <div id="sidebar">
3   <ul>
4     <li>
5       <div id="subscribe">
6         <h1>Subscribe To Daily Updates</h1>
7         <form action="#" method="post">
8           <input type="text" name="" value="Name" /><br />
9           <input type="type" name="" value="Email Address" /><br />
10          <input type="submit" name="" value="Subscribe" class="button" />
11        </form>
12      </div>
13    </li>
14    <li>
15      <h2>Categories</h2>
```

We're not done yet. We still need to add the conditional statement. We will add this just before the opening .

```
<?php if ( !is_home() ) { ?>
```

And this just after the closing .

```
<?php } ?>
```

This encloses the box inside the condition. It tells WordPress to only display the box when this condition is met. If (!is_home()) means if this is not the home page. The command we have just given is, if this is not the home page (condition to meet) then display the box.

```
2 <div id="sidebar">
3   <ul>
4     <?php if ( !is_home() ) { ?>
5     <li>
6       <div id="subscribe">
7         <h1>Subscribe To Daily Updates</h1>
8         <form action="#" method="post">
9           <input type="text" name="" value="Name" /><br />
10          <input type="type" name="" value="Email Address" /><br />
11          <input type="submit" name="" value="Subscribe" class="button" />
12        </form>
13      </div>
14    </li>
15    <?php } ?>
16    <li>
17      <h2>Categories</h2>
```

Save sidebar.php, visit your Sandbox blog, click on any page that is not your home page. The subscription box should now appear on all pages.

Conclusion

Wonderful! Our theme now has a good solid base. [Here](#) is the download for my version of this theme up to this lesson. If you run into any problems or get confused, download and compare it with your own work.

One last thing to do before we can package and distribute it – that's adding widget functionality. You can actually begin to use this on most blogs now the only problem is most people have come to expect their sidebars to be widget ready. To keep up with the norm that is what we will do too. But before that...

Next Lesson...

How about a short break from all this dry code and take a look at other technical aspects of running your Blog Evangelist business.

Kelly has talked about using help desks to manage support and communication. We will take a look at what types of help desks are available to you, why and when you should use them and how it can help you stay on top of things.

Lynette Chandler

Lynette Chandler

Tech Based Training