



Lesson #10

Developing index.php

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.

The Template Hierarchy

Our theme is coming together. In this lesson, we are working on index.php which is the linchpin of any WordPress theme.

The index file displays the posts on a blog but that's not all it does. It is the master template that WordPress will use when it doesn't find any other template file it can use. Allow me to explain further.

When displaying a page, WordPress performs a number of logical decisions. One of them is to decide which template to use. Supposing you click on a category link. In the background, WordPress will check to see if the template has a category.php it can use to display the page. If not, it will use the next best alternative, archive.php. If the theme doesn't have an archive template, it'll use index.php. This is the hierarchy.

At the very bottom, is the index.php template. Higher up are files like archive.php, single.php, home.php, category.php and other less common template files. Index.php is the catch all template. That's why it is important. Without an index.php your theme wouldn't work.

Beyond that, in theme development index.php is also the basis of the archives, category and single templates. Get the index right and you'll have less to do in other templates. When we develop the other template files this will become clearer to you.

Ready to get cracking on this all important file? Good. Gather your common tools – PsPad, Web browser – startup your server or login to your web host. Let's go.

index.php

Our index.php has a division named featurebar. This is the feature spot that is on the top of the page. We will do something really special with that later. This lesson we are going to skip it.

```
3 <div id="featurebar">
4 <div id="lefttube">
5 <div id="featureimg">
6 
7 </div>
8 </div>
9 <div id="righttube">
10 <div id="subscribe">
11 <h1>Subscribe To Daily Updates</h1>
12 <form action="#" method="post">
13 <input type="text" name="" value="Name" /><br />
14 <input type="text" name="" value="Email Address" /><br />
15 <input type="submit" name="" value="Subscribe" class="button" />
16 </form>
17 </div>
18 </div>
19 </div>
```

Fix up the images

Are you annoyed by the three smaller broken images? Good news! We're fixing that now. In one of our earlier lessons, remember we had to fix the stylesheet so that the theme will display properly? This is pretty much the same thing.

When converting an HTML template to WordPress, most of the time your images will break. We need to tell WordPress where the images are, but we do not want to 'hard code' it. For example, we do not want to fix it by entering <http://domain.com/wordpress/wp-content/themese/images/image.jpg> that is not good because we don't know yet where this theme will end up being installed.

The images have to work whether the theme is installed on our sandbox, our clients' blog, our own blog or any other blog. Instead, we will use the bloginfo tag.

Open index.php in PsPad. Around line 23, inside the imgnav unordered list. Just before the images/imagename.jpg, enter this code.

```
<?php bloginfo('template_url') ?>/
```

Note: Don't forget the trailing slash. It is important.

Just to be clear, you are starting from:

```

21 <div id="content">
22 <div id="lefttube">
23 <ul id="imgnav">
24 <li><a href="#"></a></li>
25 <li><a href="#"></a></li>
26 <li><a href="#"></a></li>
27 </ul>

```

Insert bloginfo code here.

And ending up with this.

```

23 <ul id="imgnav">
24 <li><a href="#"></a>
</li>
25 <li><a href="#"></a>
</li>
26 <li><a href="#"></a>
</li>
27 </ul>

```

Refresh your blog in the browser. Your images should come up.



Delete the dummy posts

Next, we will delete the dummy Post 2 – 5 from index.php. This is because we won't be needing them. We will be adding The Loop (remember that one?) to this template. WordPress will generate the list of posts automatically once The Loop is in place. But how do we find out where each post starts and which post ends?

Just look for this pattern. It starts with

```
<div class="post" id="pt-2">
```

and ends with

```
</div>
```

The </div> that is immediately after the last paragraph of the dummy text.

```
42 <div class="post" id="pt-2">
43 <div class="posttitle">
44 <div class="commentnbr">
45 <h4>0</h4>
46 <p>Comments</p>
47 </div>
48 <h1>Post 1</h1>
49 </div>
50 <p>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.</p>
51 <p>In vitae felis. Quisque venenatis rutrum quam. Suspendisse
    pellentesque. Aliquam erat volutpat. Proin nisi augue, sagittis quis,
    iaculis sit amet, fermentum quis, nulla. Aenean lectus. Quisque vitae
    felis. Fusce id enim ac libero tincidunt imperdiet. Nulla mollis nisl
    eu orci. Curabitur varius est non tortor. Proin nec sapien. Curabitur
    at augue pretium nisi viverra bibendum.</p>
52 </div>
```

You'll want to delete that whole block. And repeat for each one from there down. Leave the first dummy post intact because this is one that we will work with.

If you're still not sure, you should be looking to delete the posts/code from around line 42 – line 88.

Now that we got the extra stuff out of the way, we can start filling in with the real content.

Filling in the post id

Around line 30, on this line:

```
<div class="post" id="pt-1">
```

We want to replace the number '1' with the template tag that WordPress will automatically generate for each post. A little note, this one is not a requirement. If you do not do this, you won't break your theme. But we are doing this because having a post id allows us, to use the ID in their style sheet.

In essence, we are putting up little anchor points in the theme so that the designer can use them later on. Go ahead and highlight the number 1 and replace it with this template tag.

```
<?php the_ID(); ?>
```

Once again, we have this:

```
29 <!-- Posts -->
30 <div class="post" id="pt-1">
31 <div class="posttitle">
```

We are changing it to this:

```
29 <!-- Posts -->
30 <div class="post" id="pt-<?php the_ID(); ?>">
31 <div class="posttitle">
```

Adding comments number

Next up is the comments number. Around line 33, look for the h4 tag. You will see a number 10 in it.

```
32 <div class="commentnbr">
33 <h4>10</h4>
34 <p>Comments</p>
35 </div>
```

We will replace that with this template tag.

```
<?php comments_popup_link('0', '1', '% ');
?>
```

```
32 <div class="commentnbr">
33 <h4><?php comments_popup_link('0', '1', '% '); ?></h4>
34 <p>Comments</p>
35 </div>
```

Now refresh your blog. You should see the comments block has changed to the number of comments you actually have in your post. You will also see that the color has changed because it is now a link. We will fix that when we fix our style sheet.



Get the post title

Around line 36, where you see Post 1

```
29 <!-- Posts -->
30 <div class="post" id="pt-<?php the_
31 <div class="posttitle">
32 <div class="commentnbr">
33 <h4><?php comments_popup_link('0'
34 <p>Comments</p>
35 </div>
36 <h1>Post 1</h1>
37 </div>
```

We need to replace that with this template tag

```
<?php the_title(); ?>
```

```
29 <!-- Posts -->
30 <div class="post" id="pt-<?php the_
31 <div class="posttitle">
32 <div class="commentnbr">
33 <h4><?php comments_popup_link('0'
34 <p>Comments</p>
35 </div>
36 <h1><?php the_title(); ?></h1>
37 </div>
```

Go to your browser, refresh your blog. Lets see what you have. Should be the last blog post title.



But something is not quite right yet. This post title is not clickable. Let's go ahead and make that into a link now so people can click on it to go to the full post*. Right after `<h1>`, we will add this:

```
<a href="<?php the_permalink() ?>">
```

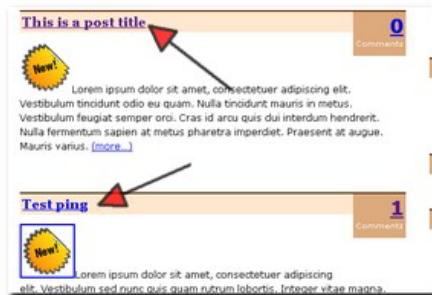
and right before the closing `</h1>`, we add this:

```
</a>
```

```
36     <p>Comments</p>
37   </div>
38   <h1><a href="<?php the_permalink() ?>"><?
  php the_title(); ?></a></h1>
39   </div>
40   <?php the_content(); ?>
```

The beginning part of the first set of the code is simple HTML anchor tag. The second part is the WordPress tag that will generate the URL for that post.

The 2nd code is short and sweet, to close the anchor tag. Now when we refresh the blog in the browser, the post becomes a link.



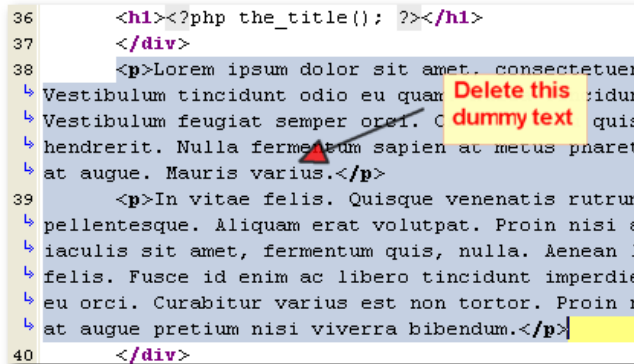
* When you click on it, you should see the post appearing on its own in a new page. You won't see any comment forms nor comments yet because WordPress did not find a single.php template file. As such, it is using index.php to display

the page. Since we won't be adding code to display comments and the comment form inside index.php you will not see those elements in this lesson.

We will add the comments and comment form when we develop single.php

Add the post content

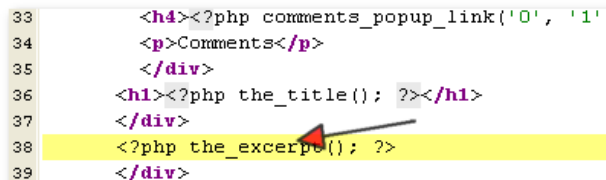
Around line 38 to 39, you'll see the two paragraphs of dummy text from the HTML template. We need to delete this.



```
36     <h1><?php the_title(); ?></h1>
37     </div>
38     <p>Lorem ipsum dolor sit amet. consectetur
    Vestibulum tincidunt odio eu quam
    Vestibulum feugiat semper orci.
    hendrerit. Nulla fermentum sapien at metus pnaet
    at augue. Mauris varius.</p>
39     <p>In vitae felis. Quisque venenatis rutrum
    pellentesque. Aliquam erat volutpat. Proin nisi a
    iaculis sit amet, fermentum quis, nulla. Aenean l
    felis. Fusce id enim ac libero tincidunt imperdie
    eu orci. Curabitur varius est non tortor. Proin n
    at augue pretium nisi viverra bibendum.</p>
40     </div>
```

Replace that with this template tag

```
<?php the_excerpt(); ?>
```



```
33     <h4><?php comments_popup_link('0', '1',
34     <p>Comments</p>
35     </div>
36     <h1><?php the_title(); ?></h1>
37     </div>
38     <?php the_excerpt(); ?>
39     </div>
```

This will pull an excerpt of the blog post. There are actually two template tags you can use to display a post. This one or the_content.

```
<?php the_content(); ?>
```

The difference is, one pulls an excerpt, another one displays the full post. Which should you choose? It's your call, and also your future clients. I have been told that using an excerpt is better for SEO purposes so that your post is not being duplicated in the landing page, the archives and the post page itself.

Please be aware that the excerpts are not generated automatically. WordPress will only display the post as an excerpt when the writer manually breaks the post into two by adding the more tag. If the writer does not do this, WordPress will display the full post. To automatically display an excerpt, you'll need to use a plugin and use a different template tag the plugin supplies.

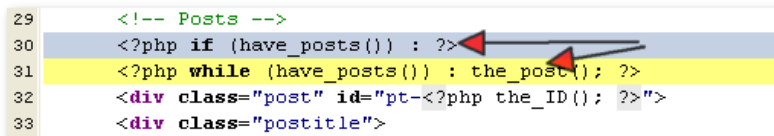
If you go to your blog and refresh your browser now, you probably won't see the post. That is because we haven't added The Loop yet. So let us do that now.

Adding The Loop

We talked about The Loop a few lessons back. This is a piece of code that will grab the posts, write out the code for each post and display them.

We will need to back track up the file a bit, to around line 29, create a new line just under <!-- Posts --> Add this code here.

```
<?php if (have_posts()) : ?>
<?php while (have_posts()) : the_post(); ?
>
```



```
29 <!-- Posts -->
30 <?php if (have_posts()) : ?>
31 <?php while (have_posts()) : the_post(); ?>
32 <div class="post" id="pt-<?php the_ID(); ?>">
33 <div class="posttitle">
```

These two lines you just added tell WordPress to start The Loop. Interpreted into English, it simply means “If there are posts, then start processing the posts”. Like with any coding, once you open a command, you have to close it. You have to tell WordPress where it should stop processing this loop.

Now skip down to around line 42 (note your line number may vary slightly because of you or I may have added a line or two here and there. Use the line numbers as a guiding point.)

Add a line above <!-- End Posts --> if you haven't already. Add this code here.

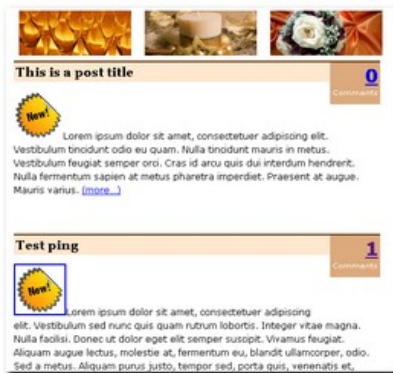
```
<?php endwhile; ?>
```

```
<?php else : ?>
```

```
<?php endif; ?>
```

```
40 <?php the_content(); ?>
41 </div>
42 <?php endwhile;
43 <?php else :
44 <?php endif;
45 <!-- End Posts -->
```

These lines of code tell WordPress to end the while command that was opened, otherwise, execute this code (else:) and then close the loop (endif).



Now, go refresh your blog in the browser. You should see several posts displayed. My sandbox is only set to show 2 posts at once. To the right, is a screen shot of what I get on my sandbox.

Hopefully, you get the same thing. If not, download this lesson's template files. You will find it at the end of the document. Check the file contents against your own.

Failing gracefully

I love this phrase. Failing gracefully is a term many developers use. It simply means developing code that will execute graciously in case something goes wrong. Sometimes also referred to as a fall through.

In general web development, a good and simple example of something 'failing gracefully', is the custom error page. Sometimes called custom 404. Having a custom error page ensures that visitors won't receive rude shocks or messages that only a geek can understand.

One of the things we want to do inside index.php is to have a fail safe. So if there are no posts, we need to tell WordPress to display a nice message.

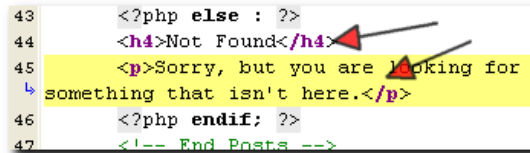
Go to PsPad, around line 44, where we just added the code to close The Loop, in between the else and endif is where we will insert our message. Create a new line.



```
41     </div>
42     <?php endwhile; ?>
43     <?php else : ?>
44
45     <?php endif; ?>
46     <!-- End Posts -->
```

Paste this code

```
<h4>Not Found</h4>
<p>Sorry, but you are looking for
something that isn't here.</p>
```



```
43     <?php else : ?>
44     <h4>Not Found</h4>
45     <p>Sorry, but you are looking for
46     something that isn't here.</p>
47     <?php endif; ?>
48     <!-- End Posts -->
```

You can add any HTML and message in between the else and endif. This is just the default text of most WordPress blogs. You can definitely change this to a friendlier message if you find it necessary.

Conclusion

Now our theme is really starting to look like a real WordPress theme. We still have quite a bit of work ahead of us, but if you have stuck through with the lessons so far, give yourself a pat on the back. You've come further than most people have.

By now you should know what your home work is. So go ahead. Build up your index.php. Here is the [zip to the this lesson's template files](http://TechBasedTraining.com/) so you can download and compare against your work. No cheating! Try adding the template tags yourself and only use this download as a guide when you find errors.

Next Lesson...

Now that we have the bare bones for a theme up and running, we are going to start beefing up this template. The next lesson we are going to work on the featured content block that we found inside index.php. Because we only want the feature strip to appear on the home page, we need to do something with that area. Till next week... good luck.

Lynette Chandler

Lynette Chandler

Tech Based Training