



Lesson #6

Acquainting Ourselves With WordPress Tags

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.

Hey there, and welcome back. This week, we are going to set our PHP files aside for a while. Because you are building this for the first time, it is necessary to understand tags and their usage before pressing on. Once you know this, and started building more themes, you can probably skip this in the grand scheme of things.

What Are WordPress Tags?

Let's make this clear. Tags mean a lot of things especially in the Web 2.0 world. There are keyword tags, HTML tags, tags as in blog tags, tags when you categorize your bookmarks and so on. When we talk about tags in Blog Theme Bootcamp we'll be referring to tags in the programming sense.

Let me recall Segment #3 where we learned how to declare and re-use variables like \$copyright and \$sitename. Well... those are tags. I know the changing terminology is kinda confusing. Fact is, they are sometimes used interchangeably. For practical purposes, we'll be calling it tags to be consistent with WordPress. At least now you know what people mean they say variable.

In WordPress, tags are the little pieces of thread that bind your theme (design) to the program and database. As you've seen in segment #4 when we activated the theme in WordPress, it doesn't display any posts or any data from the blog itself. Until you add tags, the theme and the data remain separate.

But tags are not limited to data. That's the nice thing about WordPress. The developers have built in a number of functional tags (this is my term not theirs). Functional tags do not just retrieve data from the database. They actually do something with that data.

These are tags like wp_list_categories, wp_list_pages, wp_list_authors and get_the_category_parents. These tags tell WordPress to go into the database, find the data, then write out and format the results in HTML.

How Do You Use These Tags?

That's easy. Just a matter of copying and pasting the code.

For example, if you want to display the author's name, you simply copy this code

```
<?php the_author; ?>
```

Into your theme. Some tags have additional features so you can customize the output a little bit more but for the most part, it's that simple.

Where Do You Use Tags?

The answer is wherever you want the data to appear. So let's take for example the author tag above. If you want it to appear under the title of post, this is what you would do.

```
<h1>Post 1</h1>
<?php the_author; ?>
```

Many tags can appear anywhere in your theme. But some must be installed within what WordPress calls 'The Loop'.

What's The Loop?

A loop is what programmers use to describe a set of actions that will repeat itself until all data is exhausted. Huh? Say that again?

Alright. Let's just say you instruct your database to retrieve the names of all citrus fruits in your database. The results are, Grapefruit, Lemons and Oranges. But the results it gives you is not formatted. Its only a chunk of text. That's not quite what you want. You would much rather have the result, formatted nicely for display on a web page in a list like this.

```
<ul>
  <li>Grapefruit</li>
  <li>Lemons</li>
```

```
<li>Oranges</li>
</ul>
```

To get the data from simple text of Grapefruit, Lemons and Oranges into nicely formatted HTML, you must instruct the software to enclose each of those results within ``. For this, you'll use a loop.

So what happens when you add more citrus fruit in your database? Then, the software (obeying the loop) will continue writing out `name of fruit` until there are no more left for it to write.

Apply this knowledge to WordPress. In any given blog, there are hundreds if not thousands of posts. The Loop tells WordPress to pull up data for all posts, then format every single one to be displayed on a web page. This is how every post on a blog's home page carries the same HTML code, like a cookie cutter.

I know this is getting kinda technical so won't go digging deeper. When we work on theme again and hit the part about The Loop you'll see exactly how The Loop looks like. You'll have a better understanding how it works then so don't worry too much about it now.

Which Tags Should Be In The Loop and Which Outside?

Thankfully, the WordPress Codex can give us pointers there. The Codex is a living breathing manual. It is never done. Even this state, the Codex is already a pretty comprehensive manual. 95% of what you'll be doing in WordPress especially within the scope of building basic themes can be found in the Codex.

For a theme developer, the [Template Tags page](#) is your bible. We will be pulling tags and referring to this page quite a bit. I also suggest putting that page in your bookmarks so you can refer to it regularly when working on sites for your clients.

Homework!

1. Since this lesson only serves to introduce terminology and concepts, there is no home work. If you are adventurous, check out the template tags page in the Codex. Challenge yourself. See if you can figure out where and how to use some of those tags. This is not a requirement as we will be doing this together shortly.

Coming Up Next ...

Over the next few weeks, we'll be working on one theme file at a time, filling in each one with the relevant template tags. First up, header.php.

Lynette Chandler

Lynette Chandler

Tech Based Training