



Lesson #20

Widgetizing Your Theme

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.

Why Widgets

I'm assuming most who work with WordPress know what widgets are and how they work at a basic level but here's a quick explanation if you don't. Commonly used in sidebars, widgets make it super easy for anyone to change the menus, links and any type of content in the sidebar. From WordPress admin, you access it by going to Design >> Widgets. From there, it is all point, select to click and type. Simple.

Widgets can also be used for more things than just the sidebar. You can even use widgets in the header, footer or just about any 'spot' on your theme. But it is not automatic and this is where we as the theme developer come in. It is up to us to define the spots that widgets can be used in.

What can be widgetized? Lots of things. Here is a short but not comprehensive list of areas you can widgetize.

1. Sidebar – this is the expected norm.
2. Header
3. Footer
4. Featured content

Why bother widgetizing other areas? Because for some clients it is easier and better that they use widgets to customize these spots than to allow them to edit the theme. In our BlogThemeBootcamp theme, the featured spot on the home page is a perfect example. Featured news or items should not be perpetual. They should be semi long term yet changed up once in a while to make the site look 'fresh'.

Let's get busy.

The Basics of Widgetizing Anything

To make a theme support widgets is super easy. You first need to define which areas you want users to be able to use widgets in which we have already done. And then you need these two simple pieces of code.

This that goes into functions.php (inside your theme)

```
if ( function_exists('register_sidebar') )
register_sidebar(array('name'=>'sidebar1',
'before_widget' => '<li id="%1$s" class="widget
%2$s">',
'after_widget' => '</li>',
'before_title' => '<h2 class="widgettitle">',
'after_title' => '</h2>',
));
```

And this one that goes into the spot you want to apply widgets to

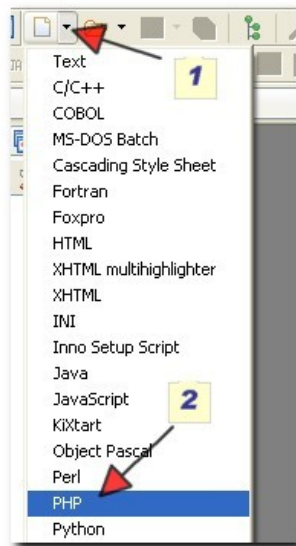
```
<?php if ( !function_exists('dynamic_sidebar') ||
!dynamic_sidebar('sidebar1') ) : ?>

<?php endif; ?>
```

Those are the only things we will be using although we will tweak them slightly for each application. Now is the time to start your servers, open PsPad (or any other text editor you are using) and log in to your web host.

Applying Widgets To The Sidebar

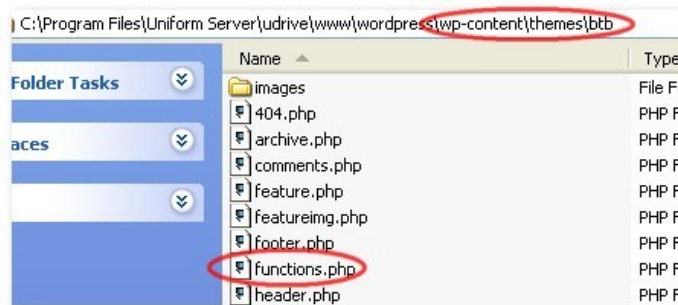
The first thing we are going to do is to create an empty file inside our theme folder and name it functions.php. In PsPad, I am going to open a new php file by clicking on the new file drop down in the tool bar (1) and then selecting PHP (2).



Then, copy and paste this code in it:

```
if ( function_exists('register_sidebar') )
register_sidebar(array('name'=>'sidebar1',
'before_widget' => '<li id="%1$s" class="widget
%2$s">',
'after_widget' => '</li>',
'before_title' => '<h2 class="widgettitle">',
'after_title' => '</h2>',
));
```

Then I will save this file into the btb theme folder, with a file name functions.php.



Since we are creating multiple widget capable areas, we should name each area appropriately so the user won't get confused when they are trying to add widgets. Back to PsPad. In the code we have just pasted, edit 'sidebar1' to 'Sidebar'.

```
1 <?php
2 if ( function_exists('register_sidebar') )
3 register_sidebar(array('name'=>'sidebar1',
4 'before_widget' => '<li id="%1$s" class="widget %2$s">',
5 'after_widget' => '</li>',
```

```
1 <?php
2 if ( function_exists('register_sidebar') )
3 register_sidebar(array('name'=>'Sidebar',
4 'before_widget' => '<li id="%1$s" class="widget %2$s">',
5 'after_widget' => '</li>',
```

Next, open sidebar.php. Around line 15 is where the conditional code for the newsletter sign up box which we put there in an earlier lesson. After that, I will press enter to create a new line, then paste this code.

```
<?php if ( !function_exists('dynamic_sidebar') ||
!dynamic_sidebar('sidebar1') ) : ?>
```

Next, make sure to change 'sidebar1' to 'Sidebar' it **must** match what you just entered in functions.php.

```
13 </div>
14 </li>
15 <?php } ?>
16 <?php if ( !function_exists('dynamic_sidebar') || !dynamic_sidebar('
17 Sidebar') : ?>
18 <li>
19 <h2>Categories</h2>
20 <ul>
```

Still in sidebar.php, scroll all the way down to the bottom. Just before the last , press enter to create a new line, then paste the following code.

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


Save sidebar.php and we're ready to test our sidebar widgets out. Before that you are probably wondering how do you know where to start or stop pasting the codes when working on your own theme design?

This is where knowing HTML is going to be your saving grace. If your HTML knowledge is weak it won't be impossible but it is going to be very difficult for

you. Simpler themes like these are more apparent. More complex designs will be a big challenge.

In this theme, this is how I determined where to start pasting the first set of code in sidebar.php. The WordPress logic is all sidebar and widget items are enclosed in lists. When a widget is added, WordPress automatically inserts `` at the beginning of each widget item and `` at the end. Basic HTML rules says if there is an `` and `` they must be enclosed inside `` OR `` and ``

So, I just went hunting for the first `` and the last ``. But wait, why did I not do this?



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

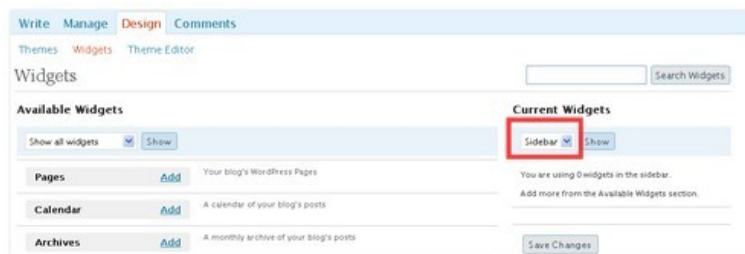
Why did I put the code after the newsletter sign up box?

```
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="text" name="" value="Email Address" /><br />
11 <input type="submit" name="" value="Subscribe" class="button" />
12 </form>
13 </div>
14 </li>
15 <?php } ?>
16 <?php if ( !function_exists('dynamic_sidebar') || !dynamic_sidebar('
17 Sidebar') ) : ?>
18 <li>
19 <h2>Categories</h2>
```

Well, that's a choice. If you look at the code, the newsletter box is enclosed inside conditional tags. Remember we did that in an earlier lesson so it won't be displayed on the home page but on other pages. If you put the code before that, then the moment widgets are added to your sidebar, you will lose the sign up box and the conditions. Every widget you put in the sidebar will appear on all pages including the home page.

There is no rule to say you can't put it at the top. It will depend on how the theme is used/customized later on. Do experiment with both options on your own to fully grasp what happens in each scenario.

Ok, let's go test out our new widgetized sidebar. You probably won't see any changes when you view your Sandbox. That's because you have not applied any widgets. So log in to your Sandbox, go to Design >> Widgets. The first thing you should notice is Sidebar in the drop down on the right side of this screen.



That is good news. WordPress picked up the widget. This also tells you that you are currently working with sidebar widgets. To your left, in the Available widgets

column, Click on Add next to any of the widgets you wish to apply. It'll be moved to the right column. Click Save Changes.



Now view your Sandbox. You should see that all the sidebar items originally in the theme have disappeared and replaced with whichever widget you applied. Congratulations! You have just widgetized your theme. Pretty easy huh?

The hardest part is only to determine where you want to start the widgetized area which we have already discussed. Other than that, this is super simple.

So what if you want to allow people to apply widgets in other areas of your theme? Just rinse and repeat.

Widgetizing Featured Content Area

Earlier in the lesson, we talked about how it would make sense to be able to apply widgets in the featured content area and so we shall make it happen.

In PsPad, close sidebar.php and open feature.php. You may leave functions.php open. You will need it.

As before, we go to functions.php, paste the code to define a new widget, right after the first set of code.

```
register_sidebar(array('name'=>'Sidebar',  
'before_widget' => '<li id="%1$s" class="widget  
%2$s">',
```

```
'after_widget' => '</li>',
'before_title' => '<h2 class="widgettitle">',
'after_title' => '</h2>',
));
```

```
1 <?php
2 if ( function_exists('register_sidebar') )
3 register_sidebar(array('name'=>'Sidebar',
4 'before_widget' => '<li id="%1$s" class="widget %2$s">',
5 'after_widget' => '</li>',
6 'before_title' => '<h2 class="widgettitle">',
7 'after_title' => '</h2>',
8 ));
9
10 ?>
```

```
1 <?php
2 if ( function_exists('register_sidebar') )
3 register_sidebar(array('name'=>'Sidebar',
4 'before_widget' => '<li id="%1$s" class="widget %2$s">',
5 'after_widget' => '</li>',
6 'before_title' => '<h2 class="widgettitle">',
7 'after_title' => '</h2>',
8 ));
9 register_sidebar(array('name'=>'Sidebar',
10 'before_widget' => '<li id="%1$s" class="widget %2$s">',
11 'after_widget' => '</li>',
12 'before_title' => '<h2 class="widgettitle">',
13 'after_title' => '</h2>',
14 ));
15 ?>
```

Notice, this time it has slight changes. Note that the 'if function exists' isn't there anymore that's because we have already defined it earlier.

Each widgetized area must be uniquely named. So we cannot name this Sidebar anymore. Besides, that would not make sense to the user. We will name it 'Featured Content'. Change 'Sidebar' to 'Featured Content'.

```
6 'before_title' => '<h2 class="widgettitle">',
7 'after_title' => '</h2>',
8 ));
9 register_sidebar(array('name'=>'Sidebar',
10 'before_widget' => '<li id="%1$s" class="widget %2$s">',
11 'after_widget' => '</li>',
```

```
1 <?php
2 if ( function_exists('register_sidebar') )
3 register_sidebar(array('name'=>'Sidebar',
4 'before_widget' => '<li id="%1$s" class="widget %2$s">',
5 'after_widget' => '</li>',
6 'before_title' => '<h2 class="widgettitle">',
7 'after_title' => '</h2>',
8 ));
9 register_sidebar(array('name'=>'Featured Content',
10 'before_widget' => '<li id="%1$s" class="widget %2$s">',
```

Your functions.php file should now contain two sets of widget codes.

```
1 <?php
2 if ( function_exists('register_sidebar') )
3 register_sidebar(array('name'=>'Sidebar',
4 'before_widget' => '<li id="%1$s" class="widget %2$s">',
5 'after_widget' => '</li>',
6 'before_title' => '<h2 class="widgettitle">',
7 'after_title' => '</h2>',
8 ));
9 register_sidebar(array('name'=>'Featured Content',
10 'before_widget' => '<li id="%1$s" class="widget %2$s">',
11 'after_widget' => '</li>',
12 'before_title' => '<h2 class="widgettitle">',
13 'after_title' => '</h2>',
14 ));
15 >>
```

Set 1

Set 2

First set is for Sidebar, and second set is for Featured content. You can add as many as you want. But you need to be very clear on the names of each and they must match up in the template files.

Back to PsPad, open feature.php if you haven't already. If you look at the HTML in feature.php, you'll see that the featured content block is really a big image and it is inside the featureimg division. Since I want to keep all the <div>'s we will define the starting of the widget area as just before the image and end right after the image.

This code goes right after <div id="featureimg">

```
<?php if ( !function_exists('dynamic_sidebar') ||
!dynamic_sidebar('sidebar1') ) : ?>
```

Make sure you change 'sidebar1' to 'Featured Content' to match up with the name in functions.php.

```
2 <div id="lefttube">
3 <div id="featureimg">
4 <?php if ( !function_exists('dynamic_sidebar') || !dynamic_sidebar('
5 Featured Content') ) : ?>
6 
```

This code, goes right after the image.

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

```
4 <?php if ( !function_exists('dynamic_sidebar') || !dynamic_sidebar('
Featured Content') ) : ?>
5 
6 <?php endif; ?>
7 </div>
```

Before you check it out, we need to add open and closing ``'s – remember WordPress adds ``'s to each widget so you must do this. There are ways around this but that is more advanced and at this point not necessary to go in to. Go ahead and type `` before the widget code and `` right after it. See what I have done here.

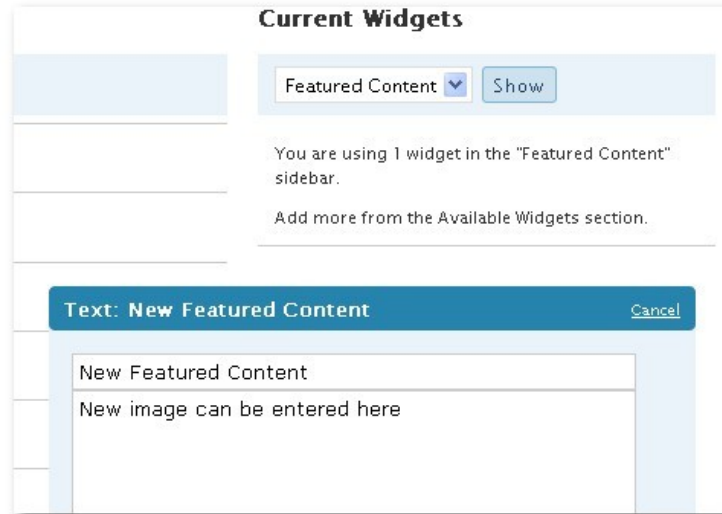
```
2 <div id="lefttube">
3 <div id="featureimg">
4 <ul>
5 <?php if ( !function_exists('dynamic_sidebar') || !dynamic_sidebar('
Featured Content') ) : ?>
6 
7 <?php endif; ?>
8 </ul>
9 </div>
```

Time to test it out. Go to Design >> Widgets in your Sandbox admin. In the right column, select Featured Content in the drop down (1). Click Show (2).

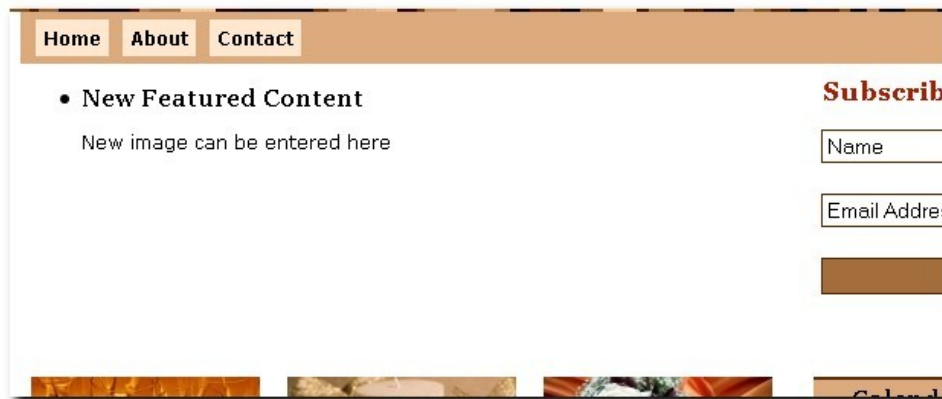


If you don't see Featured Content in the drop down, try refreshing your browser. If you still don't see it, then something has gone wrong. Make sure the code is correct or download and compare your work with [my version](http://TechBasedTraining.com/) of this theme for this lesson to find out the problem.

Add a widget any widget to Featured Content. Save. I have chosen to add the Text widget and type in a title and text since that's is probably what featured content would be. Either text or image.



When I view my Sandbox blog and refresh the page, this is what I see.



Obviously, that may not be what you have in mind. That's because the stylesheet will need to be edited. Our original theme used an image to fill that space. If you used an image as well you won't have to edit the stylesheet but if you decide to add both text and images the style sheet will need to be edited. In this case, I added the following to stylesheet.css.

```
#featureimg ul {  
    margin: 0;  
    padding: 0;
```

```
}  
#featureimg li {  
    list-style-type: none;  
}  
#featureimg h2 {  
    font-size: 1.4em;  
    color: #AF1D00;  
}
```

All that tells the browser to remove the indents, remove bullet lists and finally change the size and color of the heading. Now when I refresh my home page, I will see this.



Much better.

If you want to widgetize other areas of this theme, go for it. Just repeat the steps and make sure to give your widget areas different names. I highly encourage you to experiment. There is no better teacher than that. [This](#) is the ZIP for my version of this theme up to this lesson.

Next Lesson...

We are ready to package our theme for distribution. If you plan to release your own themes, the next lesson will outline the things you

have to do to make it acceptable for WordPress theme directory.

Lynette Chandler

Lynette Chandler

Tech Based Training