



## **The Handbook**

### **Codex**

[http://codex.wordpress.org/Dynamic Menu Highlighting](http://codex.wordpress.org/Dynamic_Menu_Highlighting)

### **Version Date**

6 August 2005

# Dynamic Menu Highlighting

Many people have asked how they should go about making a navigation menu that dynamically highlights the currently displayed page. If you have asked yourself this question before and have not been able to find the answer, then keep reading, because that's just what we are going to be learning here today!

## *Basic Navigation Lists*

The basic navigation list might look something like this:

```
<div id="navigation">
<ul>
<li><a href="#">Page One</a></li>
<li id="current"><a href="#">Page Two</a></li>
<li><a href="#">Page Three</a></li>
<li><a href="#">Page Four</a></li>
</ul>
</div>
```

And there would likely be some bit of [CSS](#) in the page's stylesheet that would specifically style list items with the "current" ID differently from other list items.

This works fine with static HTML pages, but when dealing with dynamic pages, things become a little more complicated. Perhaps this menu is supposed to be in a site's sidebar and the sidebar is contained in a single file that is called from multiple places. Obviously, as written, only one item would ever be highlighted, no matter what page is actually being viewed. That is not what we want at all!

## *Making the Code Dynamic*

PHP allows us to add the desired highlighting effect that reacts to whatever page is being viewed. Coupled with WordPress' [is\\_\\* functions](#), we can dynamically test what page is being viewed and then rewrite the code based on the results of that test.

The first step is to remove the one `id="current"` from the list and then add a unique `id` attribute to each list item.

```
<div id="navigation">
<ul>
<li id="one"><a href="#">Page One</a></li>
<li id="two"><a href="#">Page Two</a></li>
<li id="three"><a href="#">Page Three</a></li>
<li id="four"><a href="#">Page Four</a></li>
</ul>
</div>
```

The next part is where PHP comes in. We're going to write a conditional statement that will test what page is being viewed and define a variable based on the results of that test.

```
<?php
if ( is_page('Page One') ) { $current = 'one'; }
elseif ( is_page('Page Two') ) { $current = 'two'; }
elseif ( is_page('Page Three') ) { $current = 'three'; }
elseif ( is_page('Page Four') ) { $current = 'four'; }
?>
```

This piece of code uses the `is_page()` function to check the title of the current page. If the title is "Page One," the variable `$current` is assigned the value of "one;" if it is "Page Two," `$current` becomes "two;" etc., etc. On a WordPress template, this would go in the `header.php` file between the `<head></head>` tags.

Now, we need to write some CSS that will respond to this and highlight the appropriate list item based on what `$current` is. We can't do this in a CSS file because we can't put dynamic content in a CSS file. So we need to move the CSS that highlights the list item out of the CSS file and into our page itself where the dynamic content will work.

So, say we just wanted to do a simple highlight by changing the background color of the highlighted item to yellow. Our CSS might look like this:

```
#current {
background-color: yellow;
}
</style>
```

We'll move that out of the CSS file and into an inline CSS block in the header of our page.

```
<style type="text/css">
#current {
background-color: yellow;
}
</style>
```

Using a WordPress template, this would go in the `header.php` file, between the `<head></head>` tags.

Now we need to make it dynamic. So we will replace the `#current` selector with a bit of PHP:

```
<style type="text/css">
#<?php echo $current; ?> {
background-color: yellow;
}
</style>
```

This will print the value of `$current` thus completing the CSS and successfully highlighting the appropriate item.

## ***Examples***

Of course, the method here will only work with Pages created by the new Page feature in WordPress 1.5. It could be easily expanded to test for other conditions by using different `is_` functions. Ryan Boren has a good

[summary](http://boren.nu/archives/2004/10/16/templates-and-the-is-functions/) (<http://boren.nu/archives/2004/10/16/templates-and-the-is-functions/>) of the different `is_` functions and what they test for. You can also check the onsite summary, [Conditional Tags](#).

You can see this method in action at [Perfect Illusion](http://www.perfect-illusion.com) (<http://www.perfect-illusion.com>), in the upper menu bar. This menu is a slightly modified version of the [Simple Bits Tabbed Navbar](http://css.maxdesign.com.au/listamatic/horizontal06.htm) (<http://css.maxdesign.com.au/listamatic/horizontal06.htm>) from [Listamatic](http://css.maxdesign.com.au/listamatic/) (<http://css.maxdesign.com.au/listamatic/>).

You can also try the [WP-pagesnav WordPress Plugin](http://www.adsworth.info/wp-pagesnav) (<http://www.adsworth.info/wp-pagesnav>) which creates a dynamic menu utilizing the normal Page structure. This plugin includes dynamic highlighting and menus of multiple levels with options.

## **Resources**

These links have some information that you might find useful in your customization of this method and in the creation of menus and navigation lists in general.

- [Conditional Tags](#)
- [Creating Horizontal Menus](#)
- [Styling Lists with CSS](#)
- [Listamatic](http://css.maxdesign.com.au/listamatic/) (<http://css.maxdesign.com.au/listamatic/>)
- [Ryan Boren and the `is\_` functions](http://boren.nu/archives/2004/10/16/templates-and-the-is-functions/) (<http://boren.nu/archives/2004/10/16/templates-and-the-is-functions/>)

A collection of some relevant links will go here when I have more time to collect them. Others are invited to contribute some if they have them at hand.