CSS Tutorial Part 4

Creating the footer.

First we need to make the text look right. The design shows very light grey (#c9c9c9) Tahoma 10px text. I'm sure you can figure this one out yourselves, but for the sake of copy/paste:

```
#footer {
clear: both;
height: 66px;
font-family: Tahoma, Arial, Helvetica, Sans-serif;
font-size: 10px;
color: #c9c9c9;
}
```

And to change the link colour (and remove the link underline) we add this:

```
#footer a {
color: #c9c9c9;
text-decoration: none;
}
```

But the links need some way to stand out when you mouse over them, so we'll make them turn orange on hover:

```
#footer a:hover {
color: #db6d16;
}
```

We've also got to add a 1 pixel top border on the footer div, set some padding, and make the line-height 18px (increasing the leading). Because we are setting padding on the footer div, we will remove the height property to stop the padding/width/height bug I mentioned earlier. We don't really need height on this div anyway:

```
#footer {
clear: both;
font-family: Tahoma, Arial, Helvetica, Sans-serif;
font-size: 10px;
color: #c9c9c9;
border-top: 1px solid #efefef;
padding: 13px 25px;
line-height: 18px;
}
```

The last thing left to do is float the alternate navigation to the right. Note that floated elements must have a width specified to work properly, so set the width to slightly larger than the nav actually needs, and set the text alignment to right so the text sits where it should.

```
#footer #altnav {
width: 350px;
float: right;
text-align: right;
}
```

Tada! We have a footer.

Creating rollovers

There's a lot of funky CSS in this chapter, it's not imperative that you understand exactly what each bit of CSS does, just that you are able to modify this css to do what you want for other websites, which is basically changing heights widths and images. However I'll do my best to explain the code.

Let's start out easy. Remove the red background in the css, and show the navigation by removing the "hidden" class on the definition list.

The method of image rollovers we are going to use for this menu is a 100% CSS solution. The basic premise behind it is to position the items in a definition list next to each other (side by side) hide the text on them, and use CSS to change the background image depending on what state the button is in (rollover, normal, or selected).

ABOUT

ABOUT

ABOUT

For each of the 4 nav items we need to create an image like the one above. The first 3rd of the image is the normal state, the second is the mouseover state, and the third is the selected state. The animated gif on Moodle shows how this will work:

Hopefully you have some kind of understanding of how this works from the diagram above. Let's make our 4 nav images. These are available on moodle

Save them in /images/nav/

Now I'm going to explain the CSS behind this block by block

Replace your #main-nav rule with the code below:

```
/* Main Navigation */
#main-nav { height: 50px; }
#main-nav dl { margin: 0; padding: 0; }
```

This sets the main-nav div height to 50px, and strips all margins from the datalist.

```
/* IE5 Mac Hack \*/
#main-nav { padding-left: 11px; }
/*/
#main-nav { padding-left: 11px; overflow: hidden; }
/* End Hack */
```

This is a hack that does 2 things, sets the left padding of the main-nav to 11px (so its bumped in slightly like the design shows), and fixes a bug on IE5/mac.

```
#main-nav dt { float: left; }
```

This sets the definition titles (our individual nav item containers) to float left, which stacks them left to right, instead of one under the other.

```
#main-nav dt a {
display: block;
height: 0px !important;
height /**/:50px; /* IE 5/Win hack */
padding: 50px 0 0;
overflow: hidden;
background-repeat: no-repeat;
}
```

Sets the link to the same dimensions as its surrounding container, and hides the text using the overflow property.

```
#main-nav dt a:hover {
background-position: 0 -50px;
}
```

Sets the background position to move up 50px when a link is hovered.

```
#main-nav dt#about,
#main-nav dt#about a { width: 71px; background-image:
url(../images/nav/about.gif); }

#main-nav dt#services,
#main-nav dt#services a { width: 84px; background-image:
url(../images/nav/services.gif); }

#main-nav dt#portfolio,
#main-nav dt#portfolio a { width: 95px; background-image:
url(../images/nav/portfolio.gif); }


#main-nav dt#contact,
#main-nav dt#contact a { width: 106px; background-image:
url(../images/nav/contact.gif); }
```

Sets the individual widths of each nav item, and the paths to each image.

Now if all your images are named as they are above, and are saved in the correct place, your navigation should work.

Last thing we need to do to make the navigation work, is to get the selected button states to show up when you are on the corresponding page.

We need to add some new css, and modify some existing css to achieve this. Add this CSS below the rest of your navigation CSS:

```
body.about dt#about,
body.about dt#about a,
body.services dt#services,
body.services dt#services a,
body.portfolio dt#portfolio,
body.portfolio dt#portfolio a,
body.contact dt#contact,
body.contact dt#contact a {
background-position: 0 -100px;
}
```

This large CSS selector checks to see what class the body tag has, and then sets the background position of the correct navbar. So if you wanted the about navbar to be selected, you would set a class="about" on the body tag. Let's do that now:

Now the problem we have, is that we also want the header image to change based on what section we are viewing. So we need to modify the #header rule like so:

```
body.about #header {
height: 150px;
background: #db6d16
url(../images/headers/about.jpg);
}
```

Now when you create pages for your other sections, you'd just change the class on the body from about, to say, contact, set up a css rule pointing to the correct header image, and you're done.

Getting it right in I.E.

Let's start the hacks at the current problem child, IE5.

Load up your IE5 browser. There's 2 things that I notice instantly that are wrong. The first is that the page isn't centred in the browser like it should be, and the second is that the footer has a weird alignment issue.

The alignment issue is well known, so we'll tackle that one first.

IE 5 and 5.5 do not recognise the margin: auto; css property like they should. To get around this we need to use the text-align: center; property on the body which will centre the container div.

```
body {
font-family: Arial, Helvetica, Verdana, Sans-serif;
font-size: 12px;
color: #666666;
text-align: center;
}
```

This will centre your container div, but will also centre all the text inside that div. We don't want that, so we need to override the text-alignment inside the container div.

```
#page-container {
width: 760px;
margin: auto;
text-align: left;
}
```

That solves the centring issue. Now the weird footer.

I couldn't actually find any references to this particular bug with a brief google search, so I just set out to figure out how to fix it. I guessed that the bug had something to do with the floated "altnav" div, but couldn't work out what the problem was exactly. So eventually I tried putting a div around the copyright info, and that made the bug disappear.

```
<div id="copyright">
Copyright © Enlighten Designs<br />
Powered by <a href="http://www.enlightenhosting.com/">Enlighten
Hosting</a> and
<a href="http://www.vadmin.co.nz/">Vadmin 3.0 CMS</a>
</div>
```

This triggered some padding-top issues on the footer, so I removed the padding-top: 13px; attribute from the #footer rule, and added it to both the #copyright div and the #altnav div.

```css
#footer #altnav {
clear: both;
width: 350px;
float: right;
text-align: right;
padding-top: 13px;
}
#footer #copyright {
padding-top: 13px;
}
```

There's one last IE bug that I can see, and that is that when you mouseover the selected nav item, it reverts to the white background mouseover as if it wasn't selected. We don't want the selected item to change on mouseover.

If we add a few hover rules to our big rule that sets the nav selection, that will fix our problem.

```css
body.about dt#about,
body.about dt#about a,
body.about dt#about a:hover,
body.services dt#services,
body.services dt#services a,
body.services dt#services a:hover,
body.portfolio dt#portfolio,
body.portfolio dt#portfolio a,
body.portfolio dt#portfolio a:hover,
body.contact dt#contact,
body.contact dt#contact a,
body.contact dt#contact a:hover {
background-position: 0 -100px;
}
```

And there you have it. Hopefully you've learned something about CSS layouts.