



Dreamweaver 8 / CS3

Lesson 3

# Web Design - CSS List Menus & Validation

page01

## Creating CSS List menus

For this exercise, please download the following file -

[www.willgoldstone.com/downloads/LessonVsite.zip](http://www.willgoldstone.com/downloads/LessonVsite.zip)

This will contain a ready made site from Lesson 2, with a text based menu for you to work on. Remember you'll need to add this to your list of sites by defining it in Dreamweaver, see the first few pages of the Lesson 1 PDF for details if you've forgotten how.

Historically, web designers have created menus in a variety of different ways. From simple text only based menu items, to image-based menus, through to javascript based menus and animated menus created in Flash.

As navigational is a pivotal part of any website, accesible functionality is the key to creating an ideal menu. Currently the best way to guarantee great results with your menu system is to use a purely CSS & HTML based solution. This means that any browser will be able to display your menu, and will not need to have any third party scripting (such as javascript) or plugins (such as Flash) in order to a) display correctly and b) work!

Lets take a look at a standard text-based 3 button menu, in this example we have our 3 menu items wrapped inside a division called "menu" -

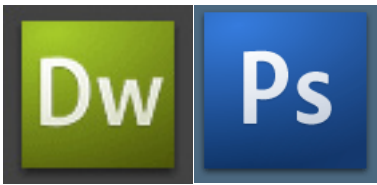
```
<div id="menu">
  <a href="index.html" target="_self">Home</a>
  <a href="portfolio.html" target="_self">Portfolio</a>
  <a href="links.html" target="_self">Links</a>
</div>
```

Next we'll add the code for an Unordered List. Unordered Lists are a staple feature of HTML that with a little CSS tweaking, can be used to create menu systems to great effect.

Here is the same slice of HTML as seen above, but with <UL> (Unordered List) tags wrapping around the 3 <A> tags (the links), and <LI> (List Item) tags wrapping each link -

```
<div id="menu">
  <ul>
    <li><a href="index.html" target="_self">Home</a></li>
    <li><a href="portfolio.html" target="_self">Portfolio</a></li>
    <li><a href="links.html" target="_self">Links</a></li>
  </ul>
</div>
```

Now lets take a look at how this makes our menu display in the Design View of Dreamweaver. You should notice that it now attempts to display our set of links as a List, going down the page as if there were line breaks after each link.



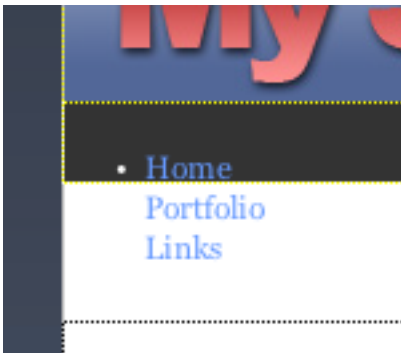
Dreamweaver 8 / CS3

Lesson 3

# Web Design - CSS List Menus & Validation

page02

It should look like this -



As you can see from the first menu / list item “Home”, list items place a bullet point next to them, like a bulleted list you’ve probably seen in software such as Microsoft Word.

Now of course we don’t want our Menu to look or really behave like a List, we simply use the list in order to make sure that search engines such as Google are indexing the links we make on our menu.

In order to make our List look more like a menu, we’ll need to cancel out everything that makes it look list-like by default. We do this using CSS commands written directly for the <UL>, <LI> and <A> tags we are using in the #menu division.

## Styling the List

Open the stylesheet file included in the ready made site (stylesheet.css) and scroll to the bottom of the code.

### 1) The UL Tag

Firstly we’ll address the <UL> tag, that’s the master Unordered List tag that wraps around our 3 links.

We do this by writing into our stylesheet -

```
#menu ul {  
}
```

This sets up a style rule for any <UL> tag in our division called “menu”. Lets take a look at the same rule filled with parameters it needs applied to it (see above right).

```
#menu ul{  
    list-style-type:none;  
    padding:0px;  
    margin:0px;  
}
```

Add this code to your stylesheet now.

As you can see from the example above, we’ve added 3 parameters. Firstly we use “list-style-type”, with the value “none” in order to cancel out the List actually displaying with bullet points as lists ordinarily do.

Then we set both the Padding and Margin parameters to 0px (pixels) in order to ensure that the default spacing normally seen in a list is not applied to our pieces of text.

### 2) The LI Tag

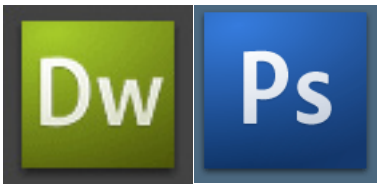
Then we come to styling the <li> (list item) tag, add the following code to your stylesheet then read on for an explanation of why we’re using this parameter.

```
#menu li{  
    float:left;  
}
```

We set the Float command on our list item in the menu in order to make each of our menu items slot in alongside one another, as opposed being listed down the page.

### 3) the A tag

For the anchor tag, we’ll write a number of parameters in order to specify precisely how to display our menu links. See the next page for the code, add it to your stylesheet then read on.



# Web Design - CSS List Menus & Validation

```
#menu a{  
  display:block;  
  width:60px;  
  height:16px;  
  text-decoration:none;  
  padding-top:7px;  
  padding-bottom:7px;  
  padding-right:20px;  
  padding-left:20px;  
  color:#FFFFFF;  
}
```

We begin by using Display:block which sets our <a> tag to display as a squared space on the page, rather than restricting the user to only clicking on the text.

Then we specify the width and height of the box for each link, set text-decoration to "none" to remove the underline, set up padding for each side of the box and finally specify a colour for our font.

Return to the HTML page and check the Design View. You should now see that your page displays like this -



You'll notice that we have a gap to the left of, and above our buttons. This is because we still have Padding applied to the #menu division.

Go back to your stylesheet and look at #menu (should be on line 82). It should look like the image above right.

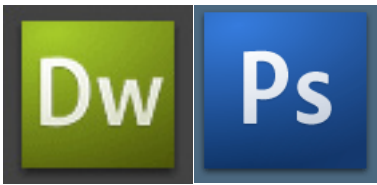
```
#menu{  
  float:left;  
  height:16px;  
  width:760px;  
  font-family:Georgia, "Times  
  font-size:12px;  
  color:#FFFFFF;  
  background-color:#333333;  
  padding-top:7px;  
  padding-right:20px;  
  padding-bottom:7px;  
  padding-left:20px;  
}
```

Now lets remove the four bottom parameters that deal with Padding, and add back the value of padding to the Width and Height of the #menu area, you should be left with the following code -

```
#menu{  
  float:left;  
  height:30px;  
  width:800px;  
  font-family:Georgia, "Time:  
  font-size:12px;  
  color:#FFFFFF;  
  background-color:#333333;  
}
```

Now return to your Design view of your HTML page, your menu should look like this -





Dreamweaver 8 / CS3

Lesson 3

# Web Design - CSS List Menus & Validation

page04

In Dreamweaver, your Design View shows you the edge of your menu items' width/height by showing you that dashed yellow outline. Bear in mind you will not see this in an actual web browser - it is just a visual aid to help you work in Dreamweaver.

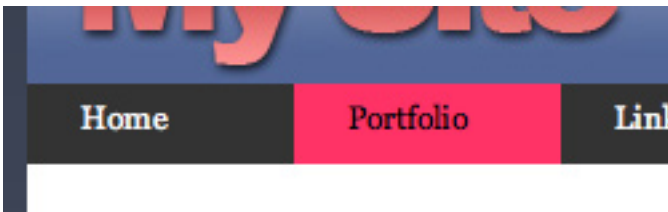
Now we need to set up some CSS code for when the user hovers their cursor over our buttons. Ideally we want some visual representation when the user is about to select a button, for this example, we'll change the background colour, and text colour to suit it, when the user hovers over our button.

Add the following code to the bottom of your stylesheet -

```
#menu a:hover{  
    background-color:#FF3366;  
    color:#000000;  
}
```

Here we are addressing the <a> tag within the #menu division, and stating that when a user hovers over this tag, we want to make the background colour #FF3366, a darkish pink, and change the font colour to #000000 (black). Bear in mind that in CSS, "color" refers to font colour, despite not saying "font-color".

Save the stylesheet and html file by going to **File > Save All**. Now open your index.html file in Safari, Firefox, or a web browser of your choosing, and you should see the following result when you roll over your buttons -



Now that we have our CSS function working, we'll need to copy the HTML menu to our other pages, because at the moment, they still have the same HTML for the menu that our index.html file did when you started this tutorial.

Rather than add the code manually to each page (links.html and portfolio.html) we'll select the code we amended in index.html with your cursor as shown below

```
<div id="menu">  
  <ul>  
    <li><a href="index.html" target="_self">Home</a></li>  
    <li><a href="portfolio.html" target="_self">Portfolio</a></li>  
    <li><a href="links.html" target="_self">Links</a></li>  
  </ul>  
</div>
```

Now go to **Edit > Copy**, or use CTRL & C (Apple-C on mac) to copy the code.

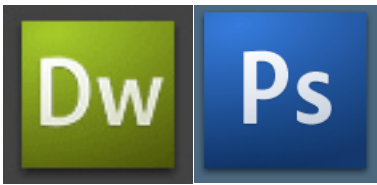
Open your links and portfolio pages from the File Browser or using **File > Open**, and do the following to both -

1) Switch to code view, and select everything on line 13 - that's the line of code containing the complete original menu before we added the list -

```
<div id="menu">  
<a href="index.html" target="_self">Home</a>  
<a href="portfolio.html" target="_self">Portfolio</a>  
<a href="links.html" target="_self">Links</a>  
</div>
```

Make sure you've selected everything from <div id="menu.. to the closing </div> tag, then go to **Edit > Paste**, or use CTRL & V (Apple & V on mac), to replace this with the code we copied from **index.html**.

Now go to **File > Save All** to save changes in all open pages. Then try your pages in a web browser and try clicking through the different pages.



# Web Design - CSS List Menus & Validation

## Validating your Code

When creating websites, ensuring you're writing accesible code is very important. The way to check this (aside from following guidelines outlined in these tutorials) is to check your markup (coding) with the validator on the W3C website.

The W3C stands for the World Wide Web Consortium. They are a group of institutions and governing bodies that work together to establish standards of practice for web development.

In its most mild form, their standards of practice may tell you for example that you must fill in the "alt" (alternate text) parameter for a picture, thereby not exluding disabled access users.

To find out exactly how well our code is written, we can use the validator at -

### validator.w3.org

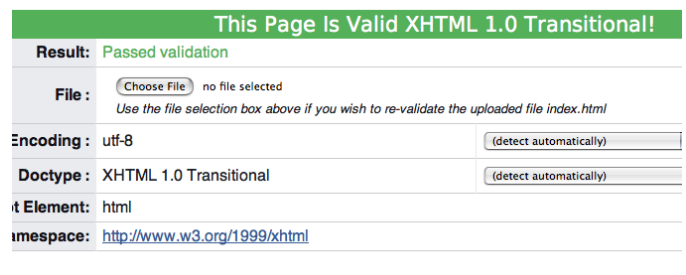
to check our files.

When you load the validator page, you'll see the first tab is "Validate by URL" - try this out by typing in any site you can think of, and the validator will show you any errors that it contains in its coding. Try ebay.co.uk as an example, at the time of writing this contains 242 errors. You will see these errors listed down the page with code line numbers to help you check your code.

However! The way we've written our sites should be XHTML 1.0 valid. To make sure, click on the "Validate by file upload". Then click on the Choose button, and navigate to / selct your index.html file. Select your file then click on "Check".

Your page should be valid XHTML 1.0 transitional, which means that the document type you specify at the start of your HTML (that dreamweaver added for you when you started a new page), matches the rules they have for writing valid code of that type.

Your page should look like this -



This shows that your page is valid, and if you scroll down the page you should be able to see two buttons with accompanying HTML you can Copy & Paste into your page to prove that your page is validated code.

Select the HTML the page gives you, which looks like this -

#### "valid" Icon(s) on your Web page

To show your readers that you have taken the care to create an interoperable W use to add this icon to your Web page:



Go to Edit > Copy or use CTRL & C on the keyboard (Apple & C on mac). Now return to Dreamweaver and in Code View, paste the button code in by going to Edit > Paste or using CTRL & V on the keyboard (Apple & V on mac).

Once your page is uploaded to a web server (covered in a seperate tutorial) users that click on this button in your site will be able to see that "Valid Xhtml 1.0" page.

