

HTML and CSS a further introduction

By now you should be familiar with HTML and CSS and what they are, HTML dictates the structure of a page, CSS dictates how it looks. This tutorial will teach you a few little tips and tricks regarding the HTML5 tags and css3 that can make your life a little easier or make it a little prettier.

```
<!doctype html>
<head>
  <title>Your First HTML5 Page</title>
</head>
<body>
  <div id="wrapper">
    <header>
      <h1>A header and Nav Bar</h1>
    </header>

    <div>
    </div>

    <footer>
      <p>bits of information go here like
      contact</p>
    </footer>
  </div>
</body>
</html>
```

Starting with a base outline as shown below

A nice simple outline, specifying the doctype, a title in the head, a body with a div inside to form the whole body, this lets us better specify changes to the whole body. You'll see I've created a header, div that we can use for our content and footer inside that to give a traditional site.

Nav Bar

Nav Bar first, we need to add a few things to the HTML to define the nav bar first. We'll use a HTML list, pretty standard for building sites, but we'll nestle it into a <Nav> tag. Thus the header now looks like this;

```
<header>
  <h1>A header and Nav Bar</h1>
  <nav>
    <ul>
      <li><a rel="external"
      href="#">Home</a></li>
      <li><a rel="external"
      href="#">About Me</a></li>
      <li><a rel="external"
      href="#">Pictures</a></li>
    </ul>
  </nav>
</header>
```

If you view your site now you'll find that it just gives you a list of links, we can fix that with a bit of CSS.

Create a new CSS document called demoCSS.css copy down the below;

```
nav {  
  display: block;  
  margin-bottom: 10px;  
}  
  
nav ul {  
  list-style: none;  
  font-size: 14px;  
}  
  
nav ul li {  
  display: inline;  
}
```

The first, for everything in between the nav wrapper we tell it to be displayed in a block display (like a <p>) and give a margin around the nav elements.

Now, for the unordered list elements in the nav wrapper, we wipe the list style making it plain and set the font size.

Now, for the list elements (li) in the nav wrapper to be displayed inline.

Underneath, we continue the CSS, the first section here changes the links within the Nav unordered list elements to display in blocks, floating to the left, with some padding around it setting a blue colour for the font and in bold.

```
nav ul li a {  
  display: block;  
  float: left;  
  padding: 3px 6px;  
  color: #575c7d;  
  text-decoration: none;  
  font-weight: bold;  
}
```

For our last one we add a hover effect over the links, when we hover over, we change the background colour and font colour JUST of that list element. The next two links referencing webkit and moz are to deal with browser differences and how they render, adding some padding around the cells, reducing the margin.

```
nav ul li a:hover {  
  background: #deff90;  
  color: #485e0f;  
  -webkit-border-radius: 3px;  
  -moz-border-radius: 3px;  
  border-radius: 3px;  
  padding: 3px 6px;  
  margin: 0;  
  text-decoration: none;  
}
```

There's no better way to discover what works than to experiment and use things like firebug to discover what does and doesn't work.

To get this to work you'll need to add a line in your html code to link it in, something like the below in between the head elements;

```
<link rel="stylesheet" href="demoCSS.css">
```

Sections

Sections help us have different styles for different areas of the page, we have to add a bit more HTML to section up the page and then a bit more CSS to style it.

We'll add all our CSS to the existing document, but first, into our main div we'll add some sections then add a few tweaks.

Let's do that first;

```

<div id="main" class="clearfix">
  <section id="left">
    <p>left hand content here.</p>
  </section>

  <section id="right">
    <p>right hand content here.</p>
  </section>
</div>

```

We change the div between our header and footer elements to add an id of "main" this is going to be our main body after all and a class of clearfix (more on that later).

After that you can clearly see our sections, left and right with their ID's and content between them.

Now for our CSS, we style our main body first, then we'll deal with that clearfix. So, with the div's of ID main, we display them in blocks and clear both sides. This means we stop any floating elements either side. Look [here](#) for more information. Finally we add a 20 pixel bottom margin.

```

/* page main */
div#main {
  display: block;
  clear: both;
  margin-bottom: 20px;
}
section#left {
  width: 550px;
  float: left;
  margin: 0 15px;
}
section#right {
  float: left;
  width: 300px;
}

```

Now, we set up our two sections, the left one of 550 pixel width, the right with 300 pixel, both floating left (relative to their position). There is a big debate about fixed width versus relative or liquid layouts, for instance if you used a fixed pixel width then viewed it on a smaller pixel density screen it can be too big. However, liquid layout is generally being replaced with responsive design.

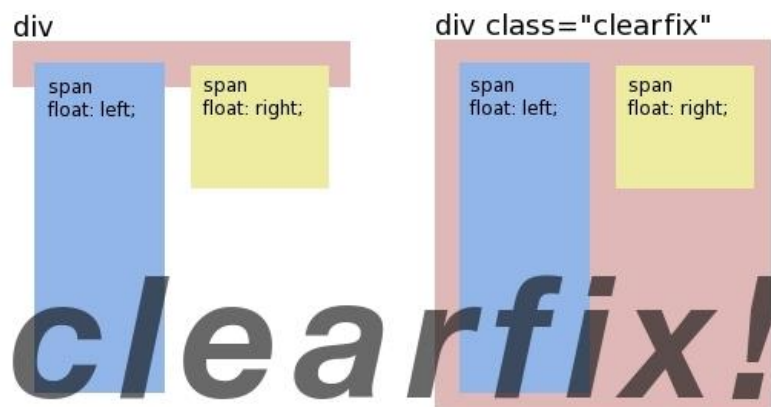
Following on from setting up the two sections, we then add the clearfix code. This code stops things from overlapping other divs and causing problems as bits try to move and overflow. Ordinarily you'd have to use things like clear, empty containers that sit between two floated sections, with this, we don't have to do anything of the sort. Below is an image from stack overflow that may help.

```

/* clearfix */
.clearfix:after {
  content: ".";
  display: block;
  clear: both;
  visibility: hidden;
  line-height: 0;
  height: 0;
}
.clearfix {
  display: inline-block;
}

html[xmlns] .clearfix {
  display: block;
}
* html .clearfix {
  height: 1%;
}

```



So, now you've got a website layout setup, I've used a lorem ipsum generator to flesh it out a bit more. It makes the division a bit clearer. You can see the example below.

A header and Nav Bar

[Home](#) [About Me](#) [Pictures](#)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed vel sodales urna, non eleifend nulla. Suspendisse ac ligula elit. Aliquam cursus vel leo non malesuada. Maecenas pulvinar orci sit amet urna hendrerit, a volutpat orci fringilla. Suspendisse commodo dui ac justo fermentum, vitae lobortis augue dictum. Phasellus at interdum purus. Nulla facilisi. Donec aliquet et massa nec pharetra. Praesent tristique mauris et sapien congue sodales. Nunc maximus pellentesque lacinia. Suspendisse id convallis ante, eget rhoncus metus. Duis at posuere turpis. Sed posuere, justo non congue dapibus, lacus arcu dignissim augue, tincidunt tristique risus massa et magna. Nulla lacinia eu nulla vitae vehicula. Fusce pulvinar luctus diam, sed varius neque vehicula eget.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed vel sodales urna, non eleifend nulla. Suspendisse ac ligula elit. Aliquam cursus vel leo non malesuada. Maecenas pulvinar orci sit amet urna hendrerit, a volutpat orci fringilla. Suspendisse commodo dui ac justo fermentum, vitae lobortis augue dictum. Phasellus at interdum purus. Nulla facilisi. Donec aliquet et massa nec pharetra. Praesent tristique mauris et sapien congue sodales. Nunc maximus pellentesque lacinia. Suspendisse id convallis ante, eget rhoncus metus. Duis at posuere turpis. Sed posuere, justo non congue dapibus, lacus arcu dignissim augue, tincidunt tristique risus massa et magna. Nulla lacinia eu nulla vitae vehicula. Fusce pulvinar luctus diam, sed varius neque vehicula eget.

bits of information go here like contact

Styling it up

This section is about styling it up, we've got a reasonable page layout now and, like the nav bar we can style the boxes. I'll go through a few different ways of styling pages, using just CSS and HTML5.

Rounded Borders

This is quite simple, the new spec allows you to implement rounded borders, you can add it to any element. In the example below I've just added it to the left hand box, but could easily add it to the right.

```
section#left {  
  width: 550px;  
  float: left;  
  margin: 0 15px;  
  
  border-radius: 25px;  
  background: #73AD21;  
  padding: 20px;  
}
```

This is the section CSS code for the left, you can see the code on top that we added before, this time we're going to add the bottom half.

The first line sets the border to be a curved one, the second line sets a background to lime green and the third gives some padding around it.

CSS Tooltips

CSS tooltips were a new one on me, they're a very straightforward but effective way of providing more information.

I've started by adding a link to the footer. It's just a `<p>` with some text in, followed by an anchor link of class `tooltip` that has some text and a `span` inside. The space is the bit that will go inside the tooltip with some CSS. At the moment though, it looks terrible, it's just a big link, time to add some CSS.

The first bit is for styling the bit you hover over, it just

```
.tooltip {
  border-bottom: 1px dotted #000000;
  color: #000000; outline: none;
  cursor: help; text-decoration: none;
  position: relative;
}
.tooltip span {
  margin-left: -999em;
  position: absolute;
}
```

makes it stand out a bit more giving it a plain view but ensuring that it has a dotted underline rather than solid. It also uses the `cursor` code to change the cursor.

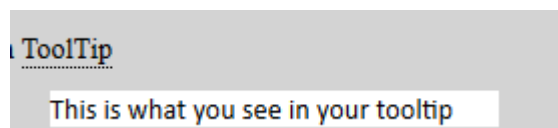
The second bit is for styling the actual tooltip, ensuring that the `span` bit is pushed off the page, hiding the tooltip text.

Now you'll see that the tooltip bit has gone but the text is still there and if you hover over it... nothing will happen. Time to add some more css to make it appear.

```
.tooltip:hover span {
  font-family: Calibri, Tahoma, Geneva, sans-serif;
  position: absolute;
  left: 1em;
  top: 2em;
  z-index: 99;
  margin-left: 0;
  width: 250px;
  background: #ffffff;
}
```

This is for what happens when you hover over the tooltip and applies to the `span`, the text that goes into the tooltip. We set the font family, the positioning is absolute and we make sure it's always on top by changing the `z` index. Finally we update the background colour, for your site you may want to change the colour, just look up a hex code online.

Now, when you hover over the tooltip you can see something appear, we can style it however we want. The basic example is shown below (I've changed the background colour to make it more evident).



Text shadow

A header and Nav Bar

```
h1
{
  text-shadow: 0 3px 0 rgba(137,127,127,.8);
}
```

Text shadow is a simple, easy to use effect to finish off your site. You can see the effect in the picture above, as long as it's subtle and not overused it's a nice way to accentuate.

The CSS is simple, shown above, the only strange thing might be the RGBA bit, the first 0 is the horizontal offset of the shadow, the second (in this case 3px) is the vertical offset, the third 0 is how much it is blurred and the fourth the RGBA is the colour. RGBA is a notation that takes four values, the amount of Red, the amount of Green, the amount of Blue and the alpha (or transparency). You can use a standard RGB generator or colour picker and play around with the alpha value.

Conclusion

This tutorial now gives you a look at creating and adding little HTML5 and CSS3 to improve your website fairly easily. It adds a few little effects but most of the difference is making it easier for you, sometimes that's all it takes. Use the tricks along with some of the websites and tools detailed in order to improve your website looks. It's up to you to add to and improve your work.

<http://css3clickchart.com/#box-sizing>

<https://www.smashingmagazine.com/2009/07/css-3-cheat-sheet-pdf/>

<http://www.color-hex.com/color/897f7f>