

This document is made available to you, on the understanding that it is not to be distributed in any way shape or form, without written permission of the author. Either in part, or in whole. Content is the exclusive work and property of AJ, owner of <http://sbi-help.easy-online-money.net>
Copyright © May 23rd 2008

The Beginners Guide To The CSS Galaxy

Css can seem very daunting. When I first started with it, I thought it would take me forever to learn how to write it! But really, it's not as hard as it might look. This document won't teach you everything there is to know about css, but it WILL give you a good start, and enable you to perform the basics without any trouble. Have fun with it.

The easiest way to learn, is to actually write it, so that's what we are going to do. Im going to give you a very plain html page, and we are going to add css to it step by step. You wont recognize it, once we're finished, and best of all.....you will know why it all works!

We are going to design a simple, but effective single column tableless layout, that would be great for a sales page. So the first thing to do, is go and download the [template page folder](#)

Making a start

Please save this folder to your desktop or where-ever you want, then use Arachnophilia to open the index.html page.

Not exactly impressive, is it? A very plain page, with some headline, text, and a couple of pictures. I have set up the folder structure, so it is exactly as you would use for SBI. So there is the main site folder, named "**css-tutorial**", and that contains the **image-files**, and **support-files** folders, along with the index.html page.

Adding the stylesheet

Our very first job, is to create a stylesheet, and link it to the page. So.....start a new text document, and save it as **style.css** in the support files folder. It will be totally blank, we will add something to it shortly. First job is to actually create it, and link to it.

Now, we want to link to it on our page. So open index.html if it isnt already and add this in the head section.

```
<link rel="stylesheet" type="text/css" href="support-files/style.css">
```

The first css code is added

Firstly, let's understand how css is written. In its most basic form, it is made up of 3 parts.

A selector, property, and a value.

Selector {property:value;} You will also see it written out this way

```
Selector {  
Property:value;  
}
```

So the basic format is **Selector, opening curly brace, property, colon, value, semi colon, closing curly brace**

Now, **selector** could be a confusing term, but it is just a name for any element that we wish to apply properties and values to. It can be body, hr, p, div h1, h2 etc.

A selector can have multiple **properties**. For example, the body selector might have properties of background-color, font-size, max-width and so on.

And the **value** is pretty self explanatory. What values do we want those properties to hold? A value can be a size, a percentage, a position, width etc.

Okay then, time to add our first css statement to the style sheet! Paste this into your blank stylesheet, save it, and then view your index page.

```
Body{  
background-color:#00C;  
font-family: Arial, Verdana, sans-serif;  
font-size: 93%;  
color: #000;  
margin:0px;  
padding:0px;  
text-align:center;  
}
```

Hmmm, not what we want, that's for sure! Here's what we REALLY want...*a nice blue background on either side, with a white page in the center.* So how do we get that?

Before we think about that, let's just take a quick look at what we have added to the body selector.

Examining the first css statement

I said earlier, a selector could have multiple properties, and we have added seven of them.

Background-color is very simple to understand. Maybe later we will change that to an image instead.

Font-size. I like to work with percentages rather than a fixed size, such as 14px, or ems. This is because you get better results with scaling and resizing. 100% is equal to 1 em, which is equal to the default font settings of 16px. Make sense? But 93% is equal to around 14px, just a good readable size.

Margin and padding we don't actually have anything to set margin and padding against yet. (well, actually, we do have the html wrapper, but let's not go there just yet!) So just leave them as they are for now. I will explain more when the time comes. We are basically "wiping the slate" of margins, to avoid problems further on.

Creating our first class

Our next job, is to actually create an area for our content to sit in. At the moment, it sprawls right across the page, so we are going to create a content area, with a white background. We have the choice of using a class, or an id to do this, so let's use a class first.

Copy this into your stylesheet, save, then view the page.

```
div.content{
width:800px;
position:center;
background-color: #fff;
text-align:left;
}
```

Aha, no change, right?
So let's add this to just after the ,body tag.

```
<div class="content">
and this
</div>
just before the </body>
```

Well, that has certainly made some changes, but still not what we want (yet). Our aim is to not only have a totally tableless page, but have it display exactly the same regardless of what browser is used to view it with. That means two things.

1. We need to use good code.
2. we also need to take away the chance that a browser will use its default settings. They vary too much, so we want to tell the browser exactly what to display, not leave it up to the browser to decide for itself!

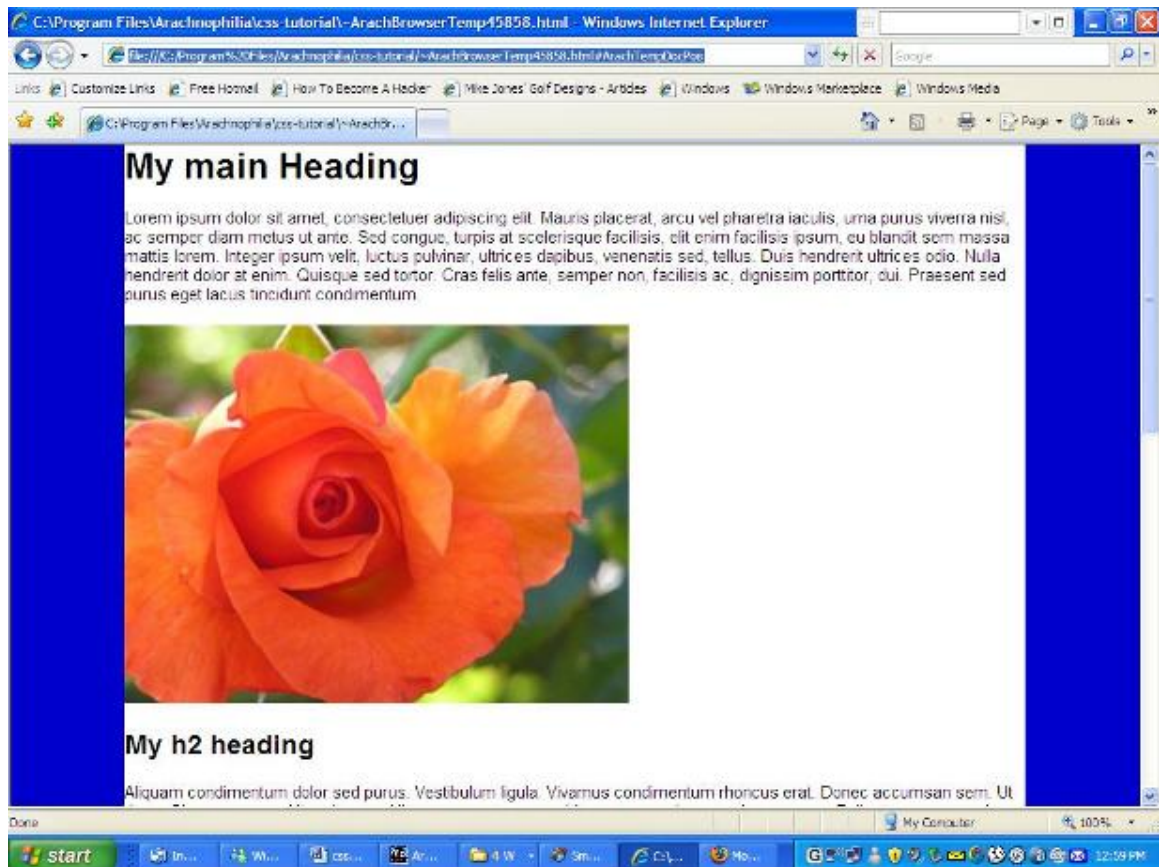
So always get into the habit of viewing your page in as many browsers as you can, when you're designing a page. I have four main ones that I use. The latest version of Firefox, internet explorer 7, opera 9.26 and safari 3.0. However, just FF and IE will do at a pinch.

View of the page with firefox



Woah, that's pretty ugly! At least we have our white content area, and it appears to be 800px wide, but it's left aligned. And there is a gap at the top of the page (also at the bottom, not shown in the screenshot). Still it's a start, what does it look like with IE?

View of the page with internet Explorer



Now, that's much better. The page is centred, and the top margin is gone. But we want the page to be perfect in ANY browser, not just some, so a few things to do yet!

So what have we done wrong? Obviously it has to be the position property, doesn't it? One browser does more or less what we want, another doesn't. So how do we get ALL browsers to do what we want?

Well, lets tell the browsers to display the way WE want them to, not how they "think" they should.

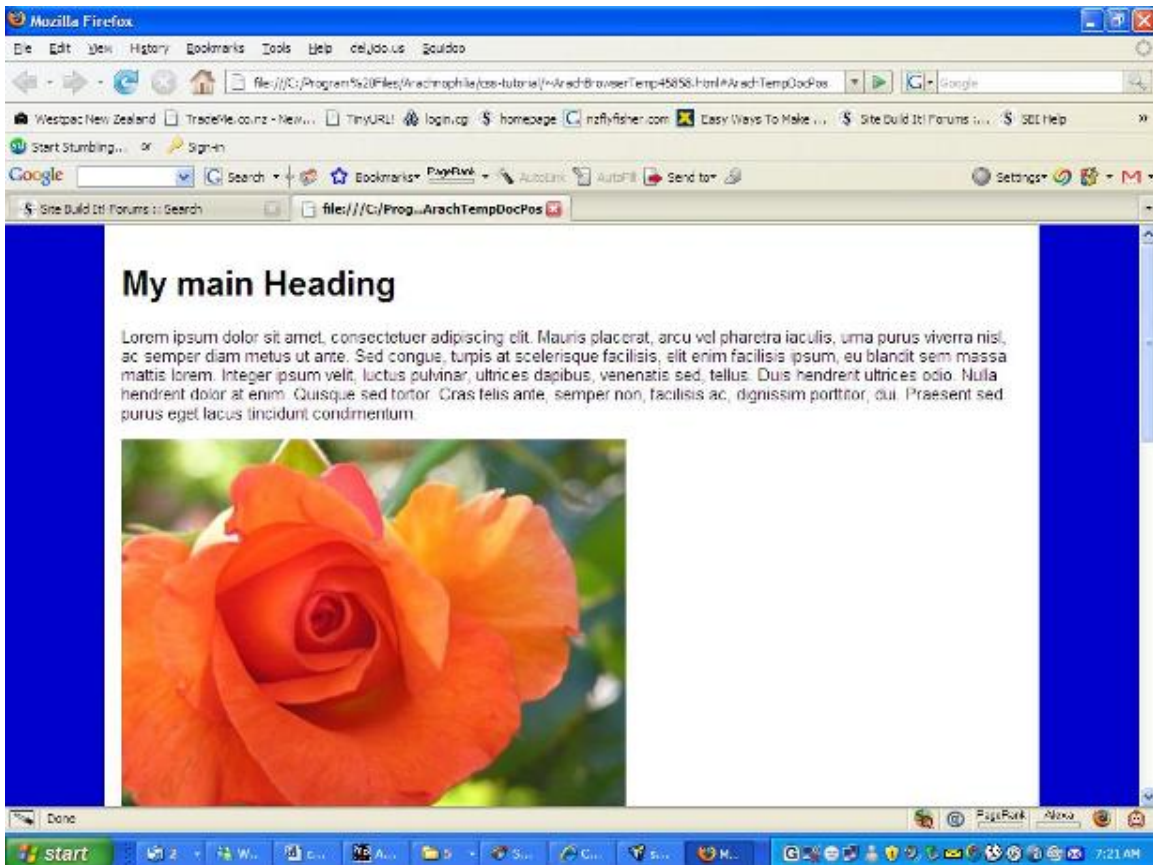
The **margin** property should do the job. If we set margins to auto, then it will balance the page from both sides. We can't set a fixed value of course, as monitors will vary in width. We want the browser to "automatically" center the page, no matter how wide or narrow, or what resolution the viewers monitor is. So delete the div code, and replace it with this.

```

div.content{
    width: 800px;
    background-color: #FFF;
    text-align: left;
    margin-left: auto;
    margin-right: auto;
    margin-top: auto;
    margin-bottom: auto;
    padding: 15px;
}

```

View in Firefox with the new code



That's much better! The 15px padding on each side of the page makes things look much better, as well. And IE looks the same too, so we have got it right. It should also appear similar in opera, safari, or any other browser you care to use.

This wraps up part 1. In part 2, we look at classes in a bit more depth....what they are, and how to construct your own.

But even better, I show you some cool software Debs (from SBI) introduced me to, that will make it SOOOO much easier to edit and troubleshoot your css!