An Introduction to Cascading Style Sheets

- HTML was not intended to be a graphic design tool. It was set up as a simple way to display text in a browser, rather like a word processor displays text on a page. Tags were added over the years in order to add a bit of colour and life into the basic white page .So along came things like images, tables, frames, and forms. These could all be presented on the page using straight HTML code.

- Web designers clamoured for a better way to present their work on a web page. Plain HTML just wasn't enough. After all, with HTML, in order to get text or an image exactly where you want it, you have to resort to complicated tables to force the alignment. And suppose you want colour behind a single paragraph of text, and not have to colour the entire page? Very tricky with straight HTML. And what about hyperlinks? Is there any way, with HTML, that we can turn the underline on and off?
- These questions, and many more, were finally addressed by the introduction of Cascading Style sheets.
- A Style is, basically, just another way to manipulate elements on a page, in order to bring a spark of life into your web design.

What is a Stylesheet?

- If you were using a word processor like Microsoft Word, you could tell the word processor how you want blocks of text to be formatted. For example, all of your page Headings could be in 28 point Times, bold, and coloured red. If you wanted the same Heading again, you can just click a drop down list and select the Heading style you set up.
- Using straight HTML, you can't do that. There's no way to apply the same formatting with a single Tag. Cascading Stylesheets, however, let you do precisely that ' change whole blocks of text with a single tag. This not only makes your code easier to read, it is also very simple to change all the formatted blocks of text to say a different font size or font colour.

- For example, in HTML, if you want to set the first paragraph of every page to bold and italics, you'd have to do this for every single paragraph that needs it:

```
<P>
<B><i>This is the first paragraph on page One. The same font styles are needed for each page on my web site.</i></B>
</P>
```

- With Stylesheets, you can get rid of all that code, and place it in the HEAD section of your page. You would then just apply the Style to any paragraph that needs it. Like this:

  <P Class = "FirstParagraph">
  This is the first paragraph on page one. The same font styles are needed for each page on my web site.
  </P>

- A stylesheet is set up by using the word STYLE in between two angle brackets. An end STYLE tag is needed to tell the browser to stop formatting the style:

  <STYLE>

  </STYLE>

- Your stylesheet code then goes between the two Style tags. Here's a style that can change text blue:

```
<STYLE>
    .Font1 { Color: Blue }
</STYLE>
<P Class =" Font1">
      This is my text.
</P>
```

- You can add other styles to the one above that we have called Font1. We can add a bold style and a size style:

```
<STYLE>
        .Font1 {
            Color: Blue;
            Font-size: 24pt;
            Font-weight: Bold;
        }
</STYLE>
```

Note: The part of the code where we applied the style (P Class = Font1) will have its text updated. We don't have to make any changes at all to the P part of the code.

- So a style is way to change blocks of code (or even individual words) and apply formatting to the block as a whole.
- You don't need individual HTML tags in the BODY of your page; just using the style name once will ensure that your formatting is applied to the whole block.

CSS Rules

- A Cascading Style Sheet rule tells the browser what the HTML looks like, and what it should do.
- A rule can dictate what just one HTML tag should look like, or you can construct your own rule to be applied as and where you want it.
- For example, a rule can be set up that tells the browser to format every <P> tag so that its first line is indented. Or you could construct your own paragraph rule, and just apply the style to certain paragraphs, not all paragraphs.
- There are three parts to a Rule: The Selector, the Property, and the Value.

The Selector

There are three different kinds of CSS Selector: An HTML selector, a Class selector, and an ID selector.

1. An **HTML Selector** is the text part of an HTML tag. The complete paragraph tag is <P>. So its Selector is just P ' in other words, strip the angle brackets off and you get the HTML Selector.
2. A **Class Selector** is one you set up yourself, to be used anywhere on your page. The Font1 from our STYLE example above was a Class Selector. We picked the name ourselves and then applied the style to some text on the page.
3. An **ID Selector** is similar to a Class selector, but you use them to identify a particular element, a text box element on a form, for example.

Examples of what all three selectors look in a STYLE tag.

```
<STYLE>
    H1{ Color: Blue }
    .NewFont1 { Font-Size: 18pt  }
    #NewTextboxColour { Color: Yellow }
</STYLE>
```

- The first one, **H1**, is the HTML Selector. Notice that it has had its angle brackets removed. With an HTML selector, all the HTML tags on the page will be formatted in the style you have set. So for H1 above, all the text between the <H1></H1> tags on the page will now be in Blue.

- The second one, **.NewFont**, is the Class selector. Note that a class selector must start with a full stop. Then you type the name for your selector (anything you want). No space is added between the full stop and the name of your selector.
- The third one, **#NewTextboxColour**, is the ID selector. An ID selector starts with the hash/pound (#) symbol. You then type the name you want to use for your ID selector. Again, no space is added between the symbol and the name of your selector.

Property and Value

Once you have set up your Selector, you then define the Properties and Values for that selector.

- The Property for the selector is the thing you're trying to change. Examples are: Font, Color, Background, Margin, Text.
- The Value for the selector is the new setting for the property. For example, for our COLOR property, we can set it to a value of an actual colour (red, blue, yellow), or a colour code (#FFFF00, #000000).
- The property and the value are enclosed in curly brackets { }. The syntax for the whole thing would then be:

  Selector {Property: Value}

  e.g. H1 {Color: Blue}

- H1 is the selector, Color is the property, and Blue is the value of the property.
- Note the colon (:) after the Property. This is used to separate a Property from a Value, so that the browser knows which one is which.
- If you want to add more than one property and value, there are two way to do it: all on one line, with each pair of properties and values separated by a semi-colon (;). Or you can put each pair of properties and values on multiple lines separated by a semi-colon (;). Like this:

  H1 {Color: Red; Font-weight: Bold; Font-Size: 16pt;}

The multiple lines version is this:

```
<STYLE>
    H1
    {
        Color : Red;
         Font-weight: Bold;
            Font-Size: 16pt;
        }
</STYLE>
```

Where to put your styles

STYLES can be inserted into three locations: Inline, Embedded, and External.

1. **Inline Style Sheets:** You can place a style tag directly in a HTML Tag. This is called Inline. Inline styles will override ones placed elsewhere. Here's an example of an Inline style:

   <H1 STYLE = 'Color: Blue'>My Heading**</H1>**

<H1 STYLE = 'Color: Blue'>My Heading**</H1>**

To place a style in a HTML tag, do the following:

- Type the Tag you want to change
- Next, type a space and then the word STYLE
- Type an equals sign (=) after the word STYLE
- Type a double quote mark
- Type the Property followed by a colon
- Type the Value
- Type the another double quote mark
- Type the right angle bracket (>) of the HTML tag

Embedded Style Sheets

- Embedded styles go in the HEAD section of your HTML page. When you embed a style in the HEAD section, you use the two tags to tell the browser where the style starts and ends. You can add a TYPE attribute, if you want. But modern browsers don't need it.
- Then in between the two STYLE tags, you would type your CSS Rules.

```
<html>
<head>
<title> css </title>
    <style type = “text/css”>

        H1 {Color : Blue}
    </style>
</head>
<body>
```

External Style Sheets

- Instead of typing the <STYLE> tags and the code for all your CSS rules in the HEAD section, you can type it all in a separate text file. You then 'tell' the browser where the text file is. The text file (along with its code) is then treated as though it were in the HEAD section. You set up an External stylesheet like this:

```
<HEAD>
   <TITLE>CSS</TITLE>
        <LINK REL =  Stylesheet TYPE = “text/css” HREF =
        “style1.css” >
</HEAD>
<BODY>
```

- To embed a stylesheet the LINK tag is used. The REL attribute tells the browser that you want to link to a stylesheet; the TYPE tells the browser what sort of file is being used; the HREF attribute tells the browser the name of the file, and where the file is.

How to use CSS Class and ID Selectors

You have seen how to set up a stylesheet. And then all the HTML tags will have their values reset with the new values you specifiedyou know that if you set up a HTML selector,. For example if you had this:

```
<html>
<head>
<title> css </title>
    <style type = “text/css”>

        H1 {Color : Blue}
    </style>
</head>
<body>
```

- Then all the H1 headings you used between the two BODY tags would have the text between the two <H1> tags coloured blue.
- In other words, to use your new HTML selector, you don't have to add anything else to your HTML code. Just use the tag in the normal way.
- However, Class and ID selectors are slightly different.

Using Class Selector

- To set up a Class Selector, the code was this:

.ClassSelector {Property: Value}

- First, you type a full stop (period). Then you type the name of your Class Selector (which can be anything you want). The Property and Value are then as normal. An example is this:

```
<html>
<head>
<title> css </title>
    <style type = “text/css”>

        .Paragraph1 {Color : Blue}
    </style>
</head>
<body>
```

- The **.Paragraph1** is the Class Selector, a name we made up ourselves. In between the curly brackets, we're then saying 'Every time this Rule is applied, the colour of the text will be blue".
- You can use this new Rule inside a normal HTML tag. Like this:
- **<H1 Class = “Paragraph1">**My Heading**</H1>**

contd

- <body>
- <h1> cascading style sheets</h1>
- <ed. For P ID= “BlockIndent”> If you were using a word processor like Microsoft Word, you could tell the word processor how you want blocks of text to be formattexample, all of your page Headings could be in 28 point Times, bold, and colour.
- …….
- </body>

- When applying your new Class Selector Rule, the word 'Class' is used. You then type an equals sign (=), followed by the name you gave your Class Selector. In our case, this was **Paragraph1**.
- But note that the full stop (period) is now missing from the start of the selector name. If you put the full stop in, your Rule won't work.
- Note that although we've used quote marks around the Class name, this is not strictly necessary. It is recommended, though.

Using ID Selectors

- You use an ID selector in exactly the same way that you use the Class selectors. The only difference is in the word used when applying the Rule. Instead of Class you use ID:

.

.

```
<title> css </title>
      <style type = “text/css”>

            #BlockIndent {
            margin-left: 15pt;
            font-weight: Bold;
            color : Blue}
      </style>
</head>
```

But that's enough of the theory for now. Let's get some practical work done. We'll go through the various ways you can add a STYLE to your web pages using your text editor. The first code we're going to write will centre a heading on a page. You can then use the same code to centre anything on your page. Click below to get started.