Session three: Introducing CSS

During our third session we shall learn the basics of CSS (Cascading Style Sheets), how to apply CSS styles to our HTML pages, and how to create a simple CSS page layout.

By the end of the session we shall have created an external style sheet and linked it to the index, links and image pages we've already made. We shall also have created a new page with a two column CSS layout.

Read through the web pages for this session (links below) and follow the practical exercises to begin to add styles to your documents.

Session three

What is CSS?

CSS (Cascading Style Sheets) is a presentation language which we use to add styling to our structural HTML elements. CSS is not the same as HTML, it is written rather differently. CSS is downloaded and the styles applied by the browser at the same time as the HTML page is loaded.

Applying CSS styles gives us an exciting level of control over the way our pages look: including the typography, colour, background images, borders, and element placement on the page.

Using CSS for presentation and HTML for structure means that presentation and structure are completely separate, so we can make changes to the one without affecting the other. For example, we can change background and text colours in CSS to make our website look completely different, without touching the HTML. Similarly, we can change textual content,

headings and links without breaking the page layout. This was not the case in early versions of HTML but, fortunately, you don't have to worry about earlier difficulties!

Using CSS has more great advantages:

- our pages can be accessible to all, as semantic HTML markup, with no presentational attributes getting in the way, is easy for screenreaders to read
- we can create a consistent look across our website by linking the same CSS stylesheet to every page
- we can make changes to the way our website looks very quickly: want a different font, or background colour? It's easy, just change two properties in one stylesheet, and your whole website will reflect the changes at once

CSS is powerful and can be complex, so we can't cover everything on this course. What we do aim for is to understand the basic principles and how to apply them, and, from that sound foundation you can develop your own learning as far as you wish.

An example: CSS Zen Garden

To see how CSS can make the same HTML page look completely different, have a look at:

www.csszengarden.com

CSS Zen Garden was a showcase for cutting-edge CSS for several years, and the designs on display show us a lot about the possibilities of using CSS. While the site remains a testament to what can be achieved, no new designs have been added for some time.

How to write CSS styles

A stylesheet contains one or more rules which describe how page elements should be displayed.

There are two parts to a style rule: the **selector** and the **declaration block**.

A style rule is written like this:

```
  selectordeclaration block
selector {property:value;
          property:value;
          property:value;
         }
```

On the left of the curly bracket:

the **selector**: selects the parts of the document to which the styles will be applied

On the right of the curly bracket:

a **declaration block**: made up of one or more style **declarations**, each declaration is a **property:value** pair, the declarations each separated by a semi-colon (**;**).

The closing curly bracket completes the declaration block.

An example

To set the font style, size, colour and background colour of our web page, we could set up the following style rule:

```
body {font-family: Arial, Helvetica, sans-
serif;
      font-size: 90%;
      color: #009B8F;
      background-color: #FFFFB0;
     }
```

What do these properties and values mean?

Font-family

Font-family specifies the font to be used for display. A computer can only display fonts it has already installed, and these may be limited in number. Also, Windows and Mac computers often have different fonts installed.

To get around these difficulties we specify a family of fonts, starting with our preferred display font, followed by alternatives, in order of preference, down to the generic serif or sans-serif font. If the browser cannot find the first listed font on the user's computer it will look for the second, then the third, and so on.

Font names with spaces should be quoted, such as: "Times New Roman".

A useful website for viewing different fonts and checking their popularity on both Windows and Mac operating systems is:

www.codestyle.org/css/font-family/

Font-size

Font size can be specified in several ways in CSS. Here, to keep things simple, we'll stick to using percentages.

A size of 90% applied to the body selector will display the text content of the main body of the document at 90% of the user's browser basefont size. A user with low vision may have their browser basefont size set at 30pt, for example, rather than the standard 12pt.

We shall also specify sizes for different heading levels as percentages too, giving an h1 a size of 250%, for example.

Using percentages for font size means that our text is scalable for users, but the headings and text will always remain in relative proportion to one another.

Color

Color sets the text colour. Note the American spelling of **color**, the English spelling will not work.

Colour in HTML and CSS is specified in Hexadecimal code: three pairs of two digit-numbers or letters (00 to FF), starting with a **#** sign.

For more on HTML colors visit:

www.w3schools.com/html/html_colors.asp

Background-color

Sets the background colour for the page.

CSS reference sources

Only four of the many CSS properties are outlined above. For a complete list of CSS properties view:
www.w3schools.com/css/css_reference.asp

If you prefer a book for reference, the standard pocket version is:
CSS Pocket Reference: Visual Presentation for the Web, by Eric a Meyer, 3rd edition (Sebastopol, CA:O'Reilly, 2007)

Many HTML textbooks also include a CSS as well as an HTML reference section, for example:
HTML, XHTML, and CSS: Visual QuickStart Guide: With XHTML and CSS, by Elizabeth Castro, 6th edition (Berkeley, CA:Peachpit, 2006)

How to link CSS and HTML

We have set up a style rule:

```
body {font-family: Arial, Helvetica, sans-serif;
      font-size: 90%;
      color: #009B8F;
      background-color: #FFFFB0;
     }
```

but, where do we write this rule, and how do we apply it to our HTML page?

There are three ways to add style rules to your document (in order of preference):

1. **External stylesheets**: write a separate stylesheet (in Notepad, as we shall see) and link it to your HTML page
2. **Document level styles**: write style rules within a <style> tag in the <head> section of a document
3. **Inline styles**: include style rules within an individual tag, using the style attribute

External stylesheets

A separate stylesheet file is created, saved with the file extension **.css**.

This is the most useful way of attaching styles to your documents, as one stylesheet can be linked to any number of web pages. Any changes made to the stylesheet will immediately be reflected in all of the pages it is linked to.

To link a stylesheet to a page, use the following HTML markup:

<link href="css/styles.css" rel="stylesheet" type="text/css">

this markup assumes you have saved a stylesheet file called **styles.css** in a sub-folder named **css**.

The link tag should appear within the <head> section of the page, as in this example (link tag is in bold):

<!DOCTYPE HTML public "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/HTML4/loose.dtd">
<html>

<head>

<title> </title>

<meta http-equiv="content-type" content="text/HTML; charset=UTF-8">

<link href="css/styles.css" rel="stylesheet" type="text/css">

</head>

<body>

</body>
</html>

You can add a link to the stylesheet into as many HTML files as you wish, and they will all have the same styling applied consistently.

If a style rule specified in an external stylesheet conflicts with a style rule specified in a document level style, the document level style takes precedence.

Document-level styles

Document-level styles specify presentation rules within the <head> section of a document which will be applied within that individual document only. In this example, the font colour and level one heading colour are specified:

```
<!DOCTYPE HTML public "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/HTML4/loose.dtd">
<html>

<head>

<title> </title>

<meta http-equiv="content-type" content="text/HTML; charset=UTF-8">

<style type="text/css">
<!--

   body {font-family: Arial, Helvetica, sans-serif;
         font-size: 90%;
         color: #009B8F;
         background-color: #FFFFB0;
        }

-->
</style>

</head>

<body>



</body>
</html>
```

The style rules themselves are enclosed by HTML comments <!-- --> so that very old browsers which do not support CSS will simply ignore them; otherwise they would display the CSS code in the browser window as it is not HTML markup.

The drawback of document-level styles is that they are individual to one document, and are time-consuming to change. They can be useful to use when working on a page prototype; when the styles are finalised, they can be moved to an external stylesheet.

If a style rule specified in a document level style conflicts with a style rule specified in an inline style, the inline level style takes precedence.

Inline styles

This is the least useful way to add styles. The style attribute is included in an individual tag, together with a list of properties and their values. The style will only affect that particular instance of the tag.

<h1 style= "font-style:italic; color:#D9000;">An inline style</h1>

CSS selectors

The CSS selector selects which parts of the document the style rule will be applied to.

There are four types of CSS selector which you will probably use at this stage:

- tag
- class
- id
- compound

We shall also introduce the and tags which we use with CSS selectors.

Tag selector

Any HTML tag can be redefined using a tag selector.

A tag selector will apply the styles you specify to every instance of that particular tag (apart from instances of tags which have styles applied by class, id or compound selectors).

Once the tag selector and style rules are specified in the stylesheet, you don't need to do anything to the HTML file in order for the styles to be applied.

Class selector

A class selector can be applied to any HTML element, and as many times per page as you wish.

For example, if I want to set up three different style rules for colour and alignment, I would write the following style rules:

```
.blue {color:#2D73B9;
       text-align: right;
      }
.green {color: #00A54E;
        text-align: center;
       }
.red {color: #E3372E;
      text-align: left;
     }
```

Note that the class selector has a full stop (**.**) preceding the class name.

You can name the class anything you like, but a meaning associated with what the class does is useful.

Once the style rules are added to the stylesheet, I can apply them to any HTML element I choose, like so:

```
<!DOCTYPE HTML public "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/HTML4/loose.dtd">
<html>

<head>

<title>class selectors</title>

<meta http-equiv="content-type" content="text/HTML; charset=UTF-8">

<link href="css/styles.css" rel="stylesheet" type="text/css">

</head>

<body>

<h1 class="green">Applying classes to HTML</h1>
```

```
<p class="blue">This is a paragraph of text coloured blue and aligned right. This is a paragraph of text coloured blue and aligned right. This is a paragraph of text coloured blue and aligned right. </p>

<h2 class="red">A red heading!</h2>

<p class="green">Some green text which is aligned centre.</p>

<h2 class="blue">A blue heading</h2>

<p class="red"> A paragraph of red text which is aligned left. A paragraph of red text which is aligned left. A paragraph of red text which is aligned left. A paragraph of red text which is aligned left.</p>

<hr class="green">

</body>
</html>
```

View the page which has these styles applied (with some additional Lorem ipsum text).

id selector

An id selector can be applied to any HTML element, but it can only be applied **once** per page (unlike a class selector).

id selectors are often applied to the tag, particularly when building CSS page layouts (see below for an explanation of the tag).

The following id selectors are set up in preparation for creating a CSS page layout. Note that the id name is preceded by a **#** sign.

```
        body {text-align: center;
              }
#wrapper {width: 80%;
              margin: 0 auto;
              text-align: left;
             }
 #header {color: #FFF;
              background-color:#2D73B9;
             }
      #nav {color: #FFF;
              background-color:#00A54E;
```

```
                width: 30%;
                float:left;
               }
    #content {color: #FFF;
                background-color:#E3372E;
                margin-left:32%;
               }
```

View the page which has these styles applied.

In the HTML code for this page we can see how the id selectors are applied using tags:

```
<!DOCTYPE HTML public "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/HTML4/loose.dtd">
<html>
<head>
<title>id selectors </title>
<meta http-equiv="content-type" content="text/HTML; charset=UTF-8">
<link href="css/styles.css" rel="stylesheet" type="text/css">
</head>
<body>

<div id="wrapper">

<div id="header">
<h1>Using id selectors to create a CSS page layout </h1>
</div><!--closes header div-->

<div id="nav">This is the div #nav</div><!--closes nav div-->

<div id="content">
<p>This is the div #content</p>
<p>We have now done most of the work we need to create a CSS page layout!</p>
</div><!--closes content div-->

<hr class="green">
Back to <a href="../selectors.html#back2">coursenotes</a>

</div><!--closes wrapper div-->
</body>
</html>
```

Div and span tags

The (for division) tag enables us to divide the content in a page into logical divisions (for example, banner, sidebar, main content, footer) which can enclose any number of other HTML elements, such as headings, paragraphs, lists, tables. We can then apply style rules to the tag which will style all of the elements it encloses.

We use the CSS properties of width and float applied to the tag to begin to create CSS page layouts (as you can see in the example above).

The tag is used to apply styles to inline elements, such as snippets of text. For example, these sentences appear thus in the HTML code:

The <span> tag is used to apply styles to inline elements, such as snippets of text. For example, these sentences appear thus in the HTML code:

Compound selectors

Compound selectors enable us to apply different styles to the same elements, applied depending upon each element's position on the page.

For example, in our page layout above, we used an <h2> tag in both the division with the id nav and in the division with the id content , but they look different as they are styled differently. To set this up, we **add** the following style rules to our **existing** stylesheet:

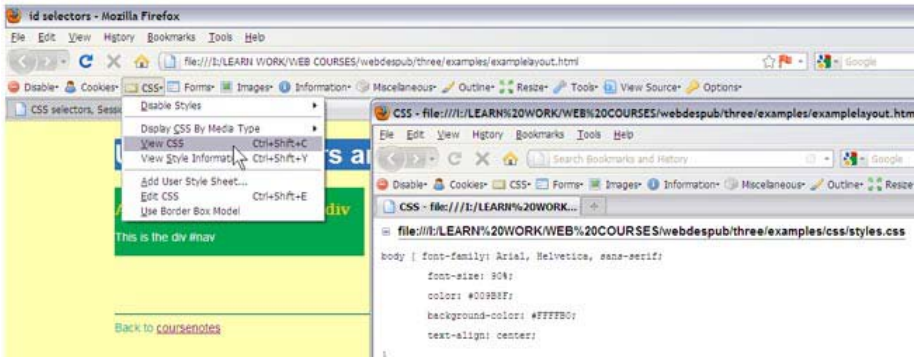
```
      #nav h2 {"Lucida Bright", "Palatino Linotype", Georgia, serif;
                font-size: 125%;
                color: #FC0;
               }
  #content h2 }font-size: 200%;
               }
```

Compound selectors are often used to style <a> tags differently in navigation bars to the way they are styled in the main body of the text.

Finally ...

We've covered a lot of ground on this page. You may need to read through it a couple of times, and try out the examples for yourself, before it drops into place.

Remember, you can view any .css file applied to a web page using the Firefox Web Developer toolbar:



If you don't yet have the Firefox Web Developer Toolbar, simply click on the link here to download and install the Web Developer toolbar by chrispederick.

Session three summary

We've packed a lot into session three! CSS is a large topic, but I hope this introduction to the basics has made you want to know more. We'll keep adding to our CSS knowledge as the course progresses.

At the end of Session three you should know that:

- CSS is a presentation language which we use with HTML to add visual presentation to our web pages, adding visual styling including different fonts, colour and background colour
- CSS helps to make our pages accessible to all (including screen-reading software and search engine robots) by separating presentation from the semantic meaning of our HTML
- CSS is written differently from HTML, with a strict syntax (way of writing the elements)
- selectors apply the style rules to the selected areas of the document
- tag, class, id and compound are all types of selector
- style rules are composed of a selector and a declaration block made up of one or more style declarations, each declaration is a property:value pair, separated by a semi-colon (;)
- the most effective way to apply styles is using external stylesheets
- we can use CSS for page layout, using the tag

You will have set up an external style sheet of your own, attached it to your existing pages with the <link> tag, and experimented with several style properties and values.

Quite an achievement!

Self-study tasks

It is always a good idea to practise new skills: to reinforce what you've already learnt, and give you confidence to tackle the next new topic.

Here are my suggestions:

- spend some time looking at the websites referred to in the How to write CSS styles page, to find out more about font types, colours and CSS properties (we'll be adding more properties next session)
- try changing colours and background colours to find combinations that you like
- try setting up more style rules for classes and ids to add to your pages (you can add them all to the same stylesheet, or you can make a new stylesheet and some new pages for your own experiments)

Next session

We'll create a page layout using CSS.