CSS Basics - Module 1

We've talked about CSS in general terms in previous modules but now we'll start to look into it in more depth.

This module will cover the core syntax of CSS and the use of different types of Style Sheets.

Review of Previous Module

The previous module focussed on the key concepts of HTML.

First we looked at the use of **Doctypes** in **HTML documents** to make sure that the browser interprets the markup correctly.

Then we discussed how the majority of HTML elements are made up of **tags** and **content**, but that some elements don't have content or closing tags. We also took an initial look at the **Document Object Model**, or **DOM**.

We covered the key **structural elements** of HTML and also some of the most common **content elements**, such as **headings**, **paragraphs**, **lists** and **links**.

Finally we took an initial look at HTML element **attributes**.

Review of Lab 1

At this point we'll take 10 minutes to go through the Lab that was set at the end of the HTML-1 module.

CSS-1 Module - Introduction

In this module we'll start to look in more depth at CSS and we'll cover the following topics:

- An introduction to CSS
- Introduction to Style Sheets
- Style Sheets types
- The CSS Cascade
- CSS rules, selectors and declarations
- Styling text, fonts, lists and links
- The id and class selectors
- Grouping and nesting selectors

CSS – Introduction

CSS stands for Cascading Style Sheets

Styles define how to display HTML elements by applying rules and declarations.

Styles were added to HTML 4.0

Prior to that styles were typically applied with special attributes in HTML elements, which made editing and maintaining HTML documents very labor intensive.

Style Sheets save a lot of work and make applying styles more manageable.

Introduction to Style Sheets

A Style Sheet is a collection of CSS Style rules and declarations.

Style Sheets may be saved as separate documents or incorporated directly into an HTML document.

There are pros and cons associated with each method, which we'll discuss as we go through the slides and notes.

Like HTML documents, Style Sheets are saved as plain text files.

The files are normally saved with a '.css' extension

Applying a Style sheet

There are three ways of applying CSS styles to an HTML document:

- By defining an internal style sheet with a <style> element in the document's <head> element.
- By using an external style sheet
- By defining one or more styles as an attribute of an element

Internal Style Sheets

An internal style sheet is defined as a style element inside the head section of a HTML document.

Internal style sheets are typically used when the styles apply to just that one document.

```
<head>
  <title>Example internal style sheet</title>
  <style type="text/css">
    h1{color:#ff0000;}
    body{
      font-family:"Times New Roman", Times, serif;
    }
  </style>
</head>
```

Internal Style Sheets – Example

```
<head>
  <title>Example internal style sheet</title>
  <style type="text/css">

    h1{color:#ff0000;}

    body{
      font-family:"Times New Roman", Times, serif;
    }

  </style>
</head>
```


Style 'type' attribute

When defining a style element it is usual practice to add a **type** attribute to indicate to the browser that the contents of the element are to be treated as CSS text.

```
<head>
  <style type="text/css">
    h1, h2{
      color:#f5b;
    }
    body{
      font-family:"Times New Roman", Times, serif;
    }
  </style>
</head>
```

Note that this is not strictly necessary if the HTML5 doctype is being applied but we would recommend that it is used for the time being.

Exercise

Open the 'html-exercise-4.html' file from the previous module and add the type attribute to the opening <style> tag.

Inline Styles

Inline styles are defined by use of the style attribute in an opening HTML tag. The style attribute can contain any valid CSS rules.

Internal style sheets can be used when the styles apply to just that one element.

```
<div style="color:#f00; font-weight:bold;" ></div>
```

The use of **internal style sheets** and **inline styles** means that the html document in which they are defined has to be edited.

External style sheets provides much greater flexibility and better separation of content and styling.

External Style Sheets

An external style sheet is created as a separate file and then **linked to** from the HTML document.

External style sheets are ideal when the styles are to be applied to multiple pages in a website, as it allows a single point of update.

Each HTML document in which the styles are to be applied then link to the external style sheet using the **<link>** tag.

```
<head>
  <title>Example external link</title>
   <link rel="stylesheet" type="text/css" href="styles.css">
</head>
```


The <link> element

The link element is an empty element and therefore only has an opening tag.

Attributes in the opening tag are used to identify the type and location of the file that is being linked to.

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

link element attributes – rel

The **rel** attribute specifies the relationship between the document that the link is to and the HTML document that the link element is in.

For linking to stylesheets you should used the “stylesheet” value:

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


link element attributes – type

The type attribute is used to indicate to the browser what type of document is being linked to.

For linking to stylesheets you should used the “text/css” value:

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

link element attributes - href

The href attribute is used to specify the location of the document being linked to.

This will either be a relative or absolute location similar to those used for the href attribute of anchor elements.

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


Quick Quiz

At this point we'll take 10 minutes to go through the Lab that was set at the end of the HTML-1 module.

Exercise

Edit the ‘hello-world-4.html’ document again.

Copy the CSS rules from the <style> element and paste them into a new text file in the same folder. Save this file as ‘styles.css’

Note: only copy the CSS rules, not the opening and closing <style> tags.

Now delete the complete style element from the HTML document and add a link to the new external style sheet.

Save and reload the page and make sure that the styles are still applied.

The CSS Cascade

The ‘C’ in CSS stands for ‘Cascading’, but what does that actually mean?

Any ideas ?

The CSS Cascade

Here are a couple of definitions for the word ‘cascade’ :

Noun
A small waterfall, typically one of several that fall in stages down a steep rocky slope.

Verb
(of water) Pour downward rapidly and in large quantities: "water was cascading down the stairs".

In both cases we have the concept of something that is flowing in one more stages or steps to an eventual finishing point.

In CSS this represents the stages or steps in processing any Style Sheets or styles.

This is pretty important, so let’s look at some examples.

Confidential

The CSS Cascade

First of all let’s look at a HTML snippet with a <style> element in the head.

```
<head>
  <title>...</title>
  <style type="text/css">
    h1 {color:red;}
    h2 {color: blue;}
  </style>
</head>
```

The browsers reads the HTML document from top to bottom and processes each tag/element is sequence.

After generating the title it then encounters an opening <style> tag and processes any text up to a closing tag As CSS.

The CSS Cascade

```
<head>
  <title>...</title>
  <style type="text/css">
    h1{color:red;}
    h2 {color: blue;}
  </style>
</head>
```

The first CSS rule that it applies is the one that makes the color of h1 heading text red (#ff0000)

Once it has applied that style it moves on to the next one, which makes the text of h2 headings blue.

So far so good...

Confidential

The CSS Cascade

```
<head>
  <title>...</title>
  <style type="text/css">
    h1 {color:red;}
    h2 {color:blue;}
    h1 {color:green;}
  </style>
</head>
```

Now let's consider the code above. What color do you think any h1 text in the document will be?

The CSS Cascade

```
<head>
  <title>...</title>
  <style type="text/css">
    h1 {color: red;}
    h2 {color: blue;}
    h1 {color: green;}
  </style>
</head>
```

This gets overwritten

This one 'wins'

In this example all h1 text in the document will be green. The process of the browser reading the document is like a flow, or cascade, downhill (top to bottom) and any CSS rules that occur further down the flow will overwrite any similar rules that occur nearer the start of the document.

The CSS Cascade

Now let's look at another example

```
<head>
  <style type="text/css">
    h1 {color: red;}
    h2 {color: blue;}
  </style>
</head>
<body>
  <h1>Welcome</h1>

  <h2 style="color:yellow;">Widgets</h2>

  <h2>Wombats</h2>

</body>
```

What color do you think the text will be for each h2 heading?

The CSS Cascade

Here's another example...

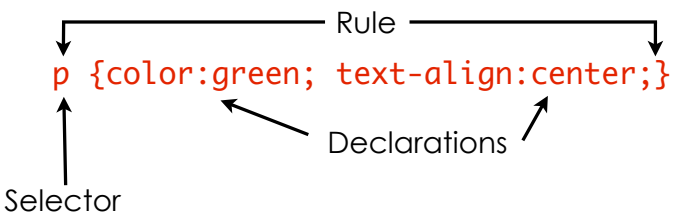
```
<head>
  <style type="text/css">
    h1 {color: red;}
    h2 {color: blue;}
  </style>
  <link rel="stylesheet" type="text/css" href="mystyles.css">
</head>
<body>
  <h1>Welcome</h1>

  <h2 style="color:yellow;">Widgets</h2>

  <h2>Wombats</h2>

</body>
```

What color do you think the text will be for each h2 now?

CSS Rules

A CSS rule has two main parts: a selector, and one or more declarations:

CSS declarations always ends with a semicolon, and declaration groups are surrounded by curly brackets:



CSS Rules

Rules can be split over multiple lines to help with readability:

```
p {
  color:green;
  text-align:center;
}
```

As with HTML, any white space is ignored by the browser

CSS Comments

Style sheets can include comments for documentation, although the syntax for CSS comments is slightly different to that for HTML comments.

A CSS comment starts with the "/*" characters and ends with the "*/" characters.

The comment can extend over multiple lines.

Single line comments can be added after CSS declarations

```
p {
  color:green;
  text-align:center;
}
```

CSS Comments - Example

```
<style>

  /* Styles for this page
     Created by Nick Roper 07/07/13 */

  /* Set the style of paragraphs */
  p {
    color:green;
    text-align:center; /* Align text centrally */
  }

</style>
```


Quick Quiz

At this point we'll take 10 minutes to go through the Lab that was set at the end of the HTML-1 module.

Exercise

Create a new, empty external style sheet named ‘reset.css’

Then go to the following address and copy and paste the CSS code into your new file.

http://meyerweb.com/eric/tools/css/reset/

Save the file and then edit the ‘hello-world-4.html‘ file again to link this external style sheet as well. Make sure that you add the link to this sheet before the link to styles.css.

Save and reload the page. You may notice some slight differences.

Your instructor will explain what we have just done and why.

Selectors

The first part of a CSS rule is know as the selector and it is used to identify which part of the HTML document should be selected for that particular rule.

The declarations are then applied to the selected parts of the document.

```
p { color:green; text-align:center; }
```

In this example the instructions to the browser are:

- Select all p elements
- Make their text green and align it centrally

Declarations

Declarations are the part of the rule that specify, or 'declare' to the browser what the styles are that should be applied to the selected parts of the document.

Declarations are made up of two parts, a name and a value.

The name identifies the type of style, such as 'color', 'font-family', 'background-color', etc and the value specifies the specific value for that style that should be applied, such as 'red', 'Verdana', 'green', etc.

```
{color:green;}
```

Note that the value is not enclosed in quotes unless it contains a space.

Selectors - HTML elements

A selector is either the name of an HTML element that you wish to style, or can also be an id or class (or a combination of element and id/class)

```
h1 {
  color: #fb5;
  padding: 0.5em;
}
```

Selectors – id and class

In addition to setting a style for a HTML element, CSS allows you to specify your own selectors called "id" and "class".

The id selector is used to specify a style for a single, unique element.

The class selector is used to specify a style for a group of elements.

The id Selector

The id selector is used to specify a style for a single, unique element.

The id selector refers to the id attribute of the HTML element, and is defined with a "#".

```
#para1 {
  text-align:center;
  color:red;
}

...

<p id="para1">this is paragraph 1</p>
```

'#' is used to create an id selector and is matched to a single element with that id attribute

The class Selector

The class selector is used to specify a style for a group of elements. Unlike the id selector, the class can apply to several elements.

The class selector uses the HTML class attribute, and is defined with a "."

```
.testimonial{font-weight:bold; color:green};

...

<p class="testimonial">here is a testimonial</p>
<p class="testimonial">here is another testimonial</p>
```

'.' is used to create a class selector and is matched to all elements with that class attribute

class and id selectors -brain teaser

```
.testimonial {color:green};

p.testimonial {red};

p#intro {color: blue;}
```

What do you think will be selected for each of the style rules above?

CSS Selectors – Grouping

Rules can be applied to multiple selectors at the same time by using a comma-separated list.

For example, the following rule will set the color and font-family for all h1, h2 and p elements:

```
h1, h2, p {
  color: #fb6754;
  font-family:"Times New Roman", Times, serif;
}
```


CSS Color Values

CSS colors values can be expressed in one of 4 ways:

- Color name: red, blue, white, etc.
- Hex values: #ff0000, #0000ff, #ffffff, etc.
- RGB values: rgb(255,0,0), rgb(0,0,255), rgb(255,255,255), etc.

The most commonly used by experienced developers are Hex values.

A list of valid color names and hex values can be found here:

http://www.w3schools.com/tags/ref_colornames.asp

CSS Selectors – Nesting

Selectors can be nested by separating them with a space. This identifies a relationship between the selectors rather than grouping them together.

As an example, the following rule will only apply to any anchor elements that are located inside a p element.

```
p a {
  color: #fb6754;
  font-family:"Times New Roman", Times, serif;
  font-size: 1.25em;
}
```


Exercise

Edit the 'hello-world-4.html' file once again.

Add a class attribute to each of level 2 headings. For the first heading the class name should be 'bands' and for the second it should be 'books'.

The first paragraph should already have a class of 'intro' and the final paragraph should have an attribute of 'thanks'. If not then add them now.

Then edit the 'styles.css' document so that different colors are applied to each of the level 2 headings and the text for the first paragraph is blue and the text for the final paragraph is a light gray.

The font used for each list should also be set to Verdana.

Styling Text, Fonts, Lists and Links

Now that we've covered the core syntax and structure of CSS rules, selectors and declaration let's look at some more of the options that we have for styling our content.

Over the next few slides we'll look at styling the following:

- Text
- Fonts
- Lists
- Links

Styling Text

There are a number of text properties that we can set a value for. In this module we'll look at the following:

- Color
- Alignment
- Decoration

Styling Text – Color

CSS colors values can be expressed in one of 4 ways:

- Color name: red, blue, white, etc.
- Hex values: #ff0000, #0000ff, #ffffff, etc.
- RGB values: rgb(255,0,0), rgb(0,0,255), rgb(255,255,255), etc.

The most commonly used by experienced developers are Hex values.

A list of valid color names and hex values can be found here:

http://www.w3schools.com/tags/ref_colornames.asp

Styling Fonts

The most common properties of fonts that CSS is applied to are:

- Font family
- Font size
- Font style

We'll show some examples of each on the next slides

Styling Fonts – Font family

The term 'font family' is roughly equivalent to the 'name' of a particular font, such as Helvetica, Courier or Times New Roman. In addition there are some generic font family names such as Sans Serif – which applies to all font families that do not use serifs.

The font-family property is used to specify which font to use:

```
p {font-family:  Verdana;}
```

If the font-family name contains spaces, such as Times New Roman, then it must be enclosed in quote:

```
p {font-family:  "Time New Roman";}
```


Confidential

Styling Fonts – Font family

It is recommended practice to specify alternate fonts in case the browser is unable to apply the first choice. This is done by using a list of font families in the declaration

```
p {font-family:  Verdana, Arial, "sans-serif";}
```

In the example above the browse will first try and apply the Verdana font.

If it can't locate that font it will try Arial.

If neither font is available then it will chose a font without serifs.

Styling Fonts – font size

The size of the characters are set with the font-size property. There are a variety of units that can be used to specify the font size but for the moment we'll use pixels.

```
p.intro {font-size:  18px;}
```

The example above will set the size of the font used for paragraphs with a class of 'intro' to 18 pixels.

Styling Fonts – font style

The font-style property can be set to one of three values:

- normal
- italic
- oblique

```
p.intro {font-size:  18px; font-style: italic;}
```

The example above will set the size of the font used for paragraphs with a class of 'intro' to 18 pixels and its style to italic.

Styling Lists

List bullet points or markers used in lists can be styled with the **list-style-type** property. This allows different types of bullet point or annotation to be used for the list items:

```
ul {list-style-type: square;}

ol {list-style-type: lower-alpha;}
```

The examples above will set apply square bullet points to unordered lists and lower case letters for ordered lists.

Styling Links

By default most browsers will style link text in blue and underlined. Once the link has been used (visited) the color will be changed slightly.

You can change the initial color of any item of link text by applying a color property value.

In addition, it is possible to apply different styles to links depending on their state. There are four possible states:

- Link (normal link before it has been clicked)
- Visited (a link that has been visited)
- Hover (a link that the mouse is hovering over)
- Selected (the active link as it is clicked)

Styling Links

A special syntax is used to specify selectors for the four different links states.

First of all the 'a' element selector is used and then that is followed by a colon and then the name of the state.

```
a:link {color: white;}

a:visited {color:red;}

a:hover {color: black;}

a:active { color: green;}
```


Exercise

Edit the 'hello-world-4.html' file once again.

Style the links used in the lists to change the colors for the various states as you wish.

Also apply the following rules:

```
a {text-decoration: none;}
a: hover {text-decoration: underline;}
```

Save and reload the page and try the links.

Module Summary

In this module we took an initial look CSS and CSS Style Sheets

We CSS rules, selectors, declarations, properties and values.

We also looked at basic techniques for styling text, links and lists.

In the next module we'll come return to HTML and start to look at some more advanced topics.

CSS - Lab 2

Now that we've completed the first CSS module we'll use the new skills to build on the lab from the HTML-1 module.

A Lab Specification Document will be available on Basecamp for you to work from after this module.

This will also include details of how to submit your completed lab work.

GitHub Gist

Questions

Do you have any questions before we move on to the next module

?

Don't worry, if you think of something later then just ask!