

M4 - HTML Styling

Every HTML element has a **default style** which is set by the browser, typically the background color is white and text color is black.

Changing the default style of an HTML element, can be done with the **style attribute**. Almost every HTML element can be styled in many appropriate ways

Styling can be added to HTML elements in 3 ways:

- Inline - using a **style attribute** in HTML elements which just styles a **single feature**.
- Internal - using a **<style> element** in the HTML <head> section and can be applied to **all elements on the single web page**.
- External - using one or more **external CSS files**. This method can apply consistent styling to **all web pages on a web site**

The most common way to add styling is to keep the styles in separate CSS files.

Inline HTML Styling

The Style Attribute

The HTML style attribute for in-line styling is placed in the <body> section in the affected markup and has the following **syntax**:

```
style="property:value;"
```

Note the two parts: “Property” and “Value”. Also note the colon (:) after the “Property” element and the semicolon (;) after the “Value”.

Several styles can be strung together as follows:-

```
style="property1:value1; property2:value2; property3:value3;"
```

Strictly speaking, the semicolon after the LAST value can be omitted, but is recommended that it be routinely included.

Now check out **Inline_Text_Styling.html** with both your editor and browser.

Internal Styling (Internal CSS)

An internal style **sheet** can be used to define a common style for all HTML elements on a page. **Internal styling** is defined in the **<head>** section of an HTML page, using a **<style>** element marked- up in the manner shown below:-

```
<style> element { property:value; }</style>
```

Any number of element / property / value markups can be placed between the **<style>** and **</style>** tags. Note the curly brackets surrounding the Property and Value parts and also the colon and semicolon use.

Look at **Internal_Styling.html** with both your editor and browser.

External CSS files

Using an external .CSS file makes it easy to apply consistent styles across all the pages on a web site. This process avoids having to type in the style details multiple times on multiple web pages. Obviously each web page file needs to define **what** css file it needs to link to enable the browser to apply the styles properly. This is achieved by a simple **link** mark-up in the **Head** section of the html file.

```
<link rel="stylesheet" href="css/stylesheetname.css"/>
```

The “href” part identifies the name and location of the stylesheet. It is normal practice to put stylesheets in their own .css subdirectory which makes them easier to find on large website.

The basic format of the entries on the css style sheet file is similar to the inline and internal methods:-

```
element { property1:value1; property2:value2; }
```

Examples:-

```
h1 {font-weight: bold; font-size: 100px; color: #ff0000;}
```

```
h2 {font-weight: normal; font-size: 50px; color: #000000;}
```

More about the color numbers later.

Now check out **External_css_Styling.html** and **externalcss.css** with both your editor and browser. Where is the css file? Why?

CSS Selectors

Sometimes it is desired to change the styling within a particular element. For example, it may be required to change the styling of a complete particular paragraph or paragraphs to emphasize a particular point. This can be achieved in a couple of ways:-

- By allocating a unique **Identity (ID)** to a single individual paragraph along with styling unique to the ID.
- By giving a group of paragraphs a **class name** along with styling for that class.

The html markup for a paragraph with an **identity** would be:-

```
<P id="paratwo">
```

Note the quotes around the name. The corresponding **css** style markup would begin:-

```
p#paratwo {b.....}
```

Note the “#” before the identity name. The part inside the curly brackets is the normal markup for the style required. The “p” could be omitted from the HTML markup as the ID name should be unique to the one item that is to be different.

The html markup for each of the paragraphs with a **class** would be:-

```
<p class="blueback">
```

The **css class** markup is similar except that a period (.) is used instead of the #-

```
p.blueback {bac.....}
```

Note that although the above description uses paragraph markup, the selector process can be applied to pretty much any html element.

Now check out **css_Selector_Styling.html** and **morecss.css** with both your editor and browser.