CASCADING STYLE SHEETS (CSS)

Objectives:

- Learn and understand about the basic HTML tags
- Understand and create ordered and unordered lists
- Understand and relate hyperlinks
- Layout your webpage using tables
- Create a simple webpage

LESSON 1 – Introduction to CSS

The acronym CSS stands for Cascading Style Sheets. Style sheets define how to display HTML elements in your web browsers which allows consistency in the look and feel of your pages. It also allows the developer more control over the layout and design than straight HTML ever did.

I. Benefits of CSS
1. CSS separates the content of HTML from its layout.
2. Style sheets can save time and effort.
3. Multiple styles will cascade into one.
II. Cascading Order

In cases where there are more than one style specified for an HTML element, the following will cascade by following these rules with 1-with the highest priority.
1. Inline Style
2. Internal Style
3. External Style
4. Browser default
III. Syntax

The CSS syntax is made up of three parts: a selector, a property and a value.
selector {property: value}

The selector is normally the HTML element/tag you wish to define, the property is the attribute you wish to change, and each property can take a value. The property and value are separated by a colon, and surrounded by curly braces.
body { color: black}

NOTE: If the value is multiple words, put quotes around the value.
p {font-family: "san serif"}

If your wish to specify more than one property, you must separate each property with a semicolon. The example below shows how to define a center aligned paragraph, with a red text color.
p {text-align: center; color: red;}

To make the style definitions more readable, you can describe one property on each line.

p {

text-align: center;

color: black;

font-family: arial;

}

IV. Grouping

You can group selectors. Separate each selector with a comma. In the example below we have grouped all the header elements. All header elements will be displayed green text color.

h1,h2,h3,h4,h5,h6 {

color: green;

}

V. class selector

With the class selector you can define different styles for the same type of HTML element. Say that you would like to have two types of paragraphs in your document: one right-aligned paragraph, and one center-aligned paragraph. Here is how you can do it with styles:

p.right{ text-align: right}
p.center{ text-align: center}

You have to use the class attribute in your HTML document.

<p class = "right">
This paragraph will be right-aligned.
</p>
<p class = "center">
This paragraph will be right-aligned.
</p>

NOTE: To apply more than one class per given element, the syntax is:

<p class = "center bold">
This is a paragraph
</p>

The paragraph above will be styled by the class "center" and "bold".
You can also omit the tag name in the selector to define a style that will be used by all HTML elements that have a certain class. In the example below, all HTML elements with class = "center" will be center-aligned.

.center {text-align: center;}

In the code below both the h1 element and the p element have class = "center". This means that both elements will follow the rules in the ".center" selector.
<h1 class = "center">
This heading will be center-aligned.
</h1>
<p class = "center">
This paragraph will also be center-aligned.
</p>
NOTE: Do NOT start a class name with a number, it will not work in Mozilla/Firefox.

VI. Add Styles to Elements with Particular Attributes

You can also apply styles to HTML elements with particular attributes. The style rule below will match all input elements that have a type attribute with a value of "text".

```
input[type = "text"] {
        background-color: blue;
}
```

VII. The id Selector

You can also define styles for HTLM elements with the id selector. The id selector is defined as a #. The style rule below will match the element that has an id attribute with a value of "green".

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

The style rule below will match the <p> element that has an id with a value of "para1".

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

NOTE: do NOT start an ID name with a number, it will not work in Mozilla/Firefox.

VIII. CSS Comments

Comments are used to explain your code, and may help you when you edit the source code at a later date. A comment will be ignored by browsers. A CSS comment begins with "/*" and ends with "*/".

```
/* This is a comment */
p{
    text-align:center;
}
/* This is another comment */
```

LESSON 2 – Inline, Internal, External and Multiple Style Sheets

I. How to Insert a Style Sheet

When a browser reads a style sheet, it will format the document according to it. There are three ways of inserting a style sheet.

    a. Inline Styles

An inline style loses many of the advantages of style sheets by mixing content with presentation. Use this method sparingly, such as when a style is to be applied to a single occurrence of an element.

To use inline styles you use the style attribute in the relevant tag. The style attribute can contain any CSS property. The example shows how to change the color and the left margin of a paragraph.

```
<p style ="color:sienna; margin-left:20px;>This is paragraph</p
```

b. Internal Style Sheet

An internal style sheet should be used when a single document has a unique style. You define internal styles in the head section by using the <style> tag.

<head>
<style type = "text/css">
  hr{color:sienna;}
  p{margin-left:20px;}
  body{background-image: url("images/back40.gif");
</style>
</head>

The browser will now read the style definitions, and format the document according to it.

NOTE: A browser normally ignores unknown tags. This means that an old browser that does not support styles will ignore the <style> tag, but the content of the <style> tag will be displayed on the page. It is possible to prevent an old browser from displaying the content by hiding it in the HTML comment element.

c. External Style Sheet

An external style sheet is ideal when the style is applied to many pages. With an external style sheet, you can change the look of an entire website by changing one file. Each page must link to the style sheet using the <link> tag. The <link> tag goes inside the head section.

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

The browser will read the style definitions from the file myStyle.css and format the document according to it.

An external style sheet can be written in any text editor. The file should not contain any html tags. Your style sheet should be saved with a .css extension.

d. Multiple Style Sheets

If some properties have been set for the same selector in different style sheets, the values will be inherited from the more specific style sheet.

For example, an external style sheet has these properties for the <h3> selector

  h3{
    color:red;
    text-align: center;
    font-size: 12pt;
  }

And an internal style sheet has these properties for the <h3> selector

  h3{
    text-align: right;
    font-size: 20pt;
  }

If the page with the internal style sheet also links to the external style sheet the properties for h3 will be:

color: red;

text-align: right;

font-size: 20pt;

The color is inherited from the external style sheet and the text-alignment and the font-size is replaced by the internal style sheet.