Introduction to CSS:

CSS: Cascading Style sheet

It is used apply the Look & feel to the HTML page.

It is all about applying the styles to the HTML page.

CSS is also static.

By using CSS, we can apply more look & feel to the HTML page.

By using CSS, we can change the positioning (We can use absolute position).

In case of Absolute position, we can place (locate) the element wherever we want as per the coordinate system(x, y)

Some of the CSS properties:

font-size

font-family

color

margin

padding

position : absolute ,relative

background properties

text properties : text-align, text-transform

table properties: caption,border,cellpading...

URL for CSS Reference:

http://www.w3.org/wiki/CSS

References: Properties/Selectors

How to write CSS?

Syntax of CSS:

```
Selector {Property: value ;}
```

```
p{ background-color:red;}
body{color:red;}
```

Note:

If you want set multiple properties to a selector, use ";"

```
p
{
    font-family:verdana;
    color:blue;
    background-color:red;
}
```

What is selector?

The Selector indicates the element/tag to which CSS style is applied.

We use various selectors for CSS.

Steps to Apply Styles in HTML page

1) Declaring the Styles

2) Embed the Styles in the HTML page

Step 1: Declaring Styles

Syntax: -

```
Selector
{
        property:value;
}


p
{
        background-color:blue;
        color:red;
}
body
{
        font-family:verdana;
```

```
        font-weight:bold;
        color:silver;
        background-color:olive;
}
```

Step 2: Applying Styles in the HTML Page

This is done in different ways:

1) Inline styles

2) Embedded Styles

3) Linked styles

4) Imported styles

Inline Styles

In this case, we can set style to an individual element.

The style is applied to an element, with the help of "style" attribute.

In HTML, for every element you find style attribute.

In the Opening tag of html element, set CSS rule using “style” attribute.

Syntax: -

```
<elementname style="property:value;">content</elementname>
<p style="background-color:blue;color:red;">some paragraph</p>
```

Drawbacks of inline styles

* Duplication occurs, because the same style may be repeated in multiple elements

* It makes the page complex as we setting styles in the html element itself.

To avoid this problem, we use embedded styles.

Embedded Styles (Internal Styles)

In this case, we can apply the common styles to any element with in the same html page.

This is applicable for single page.

This is done with the help of <style> tag of the <head> section.

It separates HTML content and CSS styles.

The content goes in the <body> section.

The CSS style goes in the <head> section.

Syntax: -

```
<head>
        <style type="text/CSS">
                Selector1
                {
                        property:value;
                }
                Selector2
                {
                        Property:value;
                }

        </style>
</head>
```

```
<head>
        <style type="text/CSS">
            h1
            {
            background-color:blue;
            color:red;
            }

            p
            {
            font-family:verdana;
            font-style:italic;
            color:pink;
            }
        </style>
</head>
```

Description:

In the above style, for all <h1> tags, the same style is applied in the current page.

For all <p> tags, the same style is applied.

This is applicable for the current page only.

Here, we are using Element selector.

Note:

You can use Embed styles for applying the same style to multiple instances of the same element.

For all <p> tags, common styles (p is the selector)

For all <h1> tags, common styles (h1 is the selector)

For all <a> tags, common styles. (a is the selector)

Note:

In case of HTML 5, we don't need the "type" attribute in the <style> element.

```
<head>
    <style>
        p
        {
        color:rgb(100,120,230);
        }
    </style>
</head>
```

Note:

The embedded style is applicable only the current page.

To apply the same style for multiple pages, we go for External Styles.

External Styles sheets or Linked Styles

It is used to apply the common styles to multiple pages in the same web site.

Here, we can create external CSS file and then link it to any page within the web site.

We use <link> tag to link External CSS to any web page.

Steps in Creating External Styles

Open any text editor or any designer tool

Define the CSS properties

Save with .CSS extension

Note:

In case of External style sheet file, we don't require <style> element.

We can directly define the CSS properties (CSS rules).

mystyle.css

```
p
{
        font-family:verdana;
}
h1
{
```

```
                color:blue;
        }
        body
        {
                background-color:#b5b5b5;
        }
        a:active
        {
                color:red;
        }
        a:visited
        {
                color:blue;
        }
        img
        {
                height:200px;
                width:200px;
        }

body
{
   background-color:Silver;
   color:Olive;
   font-family:Verdana;
```

```
}

p
{
   font-family:"Monotype Corsiva";
   font-size:x-large;
   font-style:italic;
}
```

Applying External Styles sheet in the HTML Page

[Linking CSS to the HTML Page]

Syntax: -

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

Imported Style sheets

We can also import the External styles sheets into the page.

These imported styles are applied first.

It is used to import one .CSS file into some other .CSS file.

Syntax:

```
<head>
    <style type="text/CSS">
        @import url('site.CSS');
    </style>
</head>
```

Summary:

Inline Styles (for a specific element)

Embedded Styles (for a specific page)

Linked Styles (for multiple pages)

Imported styles (for importing in other .CSS files).

For background image

Note:

You can apply image as background of body (complete page) or you can apply for a specific element (block).

```
body                {
background-image:url(plain.jpg);
    background-position:center;
    background-repeat:no-repeat;
  }
```

This can be applied for any element. (body, div, p, span or any HTML element).

```
<!doctype html>
<html>
	<head>
		<title>CSS demo</title>
		<style>
			body
			{
		background-image:url(bak.jpg);
		background-repeat:no-repeat;
		background-size:700px 700px;
			}
		</style>
	</head>
	<body>
		<h1>Welcome Marlabs</h1>
	</body>
</html>
```

Working with Selectors in CSS

What is CSS selector?

The Selector indicates the tag(element) to which the CSS style is applied.

It is used to locate the html element to which CSS properties are applied.

We have various selectors

- HTML element as a Selector
- CLASS as a Selector
- ID as Selector
- Grouped Selector
- Contextual Selector
- CSS 3 Selectors
    - attribute selector
    - parent child selector

HTML element as a selector

You can take any html element/tag as a selector in CSS

(b,p,a,img,table,ol,ul,li,em,body,h1,h2,h3,h4,h5,h6,hr ..)

In this case, for all instances of the one element, the same style is applied.

If we take p as a selector, for all <p> tags, the same style is applied.

If we take div as a selector, for all tags the same style is applied.

Note:

The selector is applicable for only embedded styles and linked styles (imported).

The selector is not required for inline styles as we are setting the style within the element itself.

```
p
{
        color:red;
        background-color:blue;
}

div
{
        height:100px;
        width:200px;
        position:absolute;
}
```

Here, for all the paragraphs in the current page, the same CSS style is applied (color is red, background color is blue)

For the div in the current page, the same height, width and position is applied.

CLASS as a selector

It is used to apply various styles to multiple instances of the same element.

It is also used to share the same style for multiple instances of different elements.

This is done with help of "class" attribute of any html tag.

To define CSS class, we use "." as prefix.

Syntax: -

(In CSS)

```
.classname
{
    property:value;
    property:value;
}

.header
{
    background-color:blue;
    color:red;
}
```

Syntax: - set CSS class to html element.

```
<element class="className"> conent </element>
<p class="header"> some content </p>
<span class="header"> some content</span>
<div class="header"> some content</div>
```

In the above, all are different elements. But sharing the same style.

ID as a Selector

HTML includes ID attribute for every element.

We use ID for uniquely identifying HTML elements.

In CSS, we use ID as a selector for setting unique styles to each element.

To define CSS selector for ID, we use "#" as prefix.

Syntax: -

(In CSS)

```
#idname
{
        property:value;
        property:value;
}

#p1
{
        color:red;
}
#p2
{
        color:blue;
}
```

In HTML element,

Syntax:

<tag ID="someid">some content</tag>

 paragraph1 : red

 paragraph2 : blue

```
<!docotype html>
<html>
    <head>
        <title>CSS demo</title>
        <style>
            #header
            {
                height:200px;
                color:olive;
                background-color:silver;
            }

            #content
            {
        color:purpble;
        background-color:#a1a1a1;;
```

```
            height:500px;
                }

                #footer
                {
                text-align:center;
                font-weight:bold;
                font-family:verdana;
                color:pink;
                background-color:#b2b2b2;
                }
            </style>
        </head>
        <body>
            <div id="header">
            <h1>Marlabs Inc</h1>
            </div>
            <div id="content">
            <p>Marlabs Employee Communication portal
            </div>
            <div id="footer">
        &copy;All Rights Reserved. &reg; Marlabs Inc-2013
            </div>
        </body>
</html>
```

Note:-

HTML Comments

<! -- comment text -->

CSS comments

/* CSS comments */

Grouped Selectors

It is used to share the common styles to multiple elements of different type.

Syntax: -

```
selector1, selector2, …
{
    property:value;
    property:value;
}
```

```
p,h1,b
{
    font-family:verdana;
}
```

Here, all three tags are sharing the same font-family.

Here, you can choose any selector(HTML element or ID , class)

Contextual Selector

It is used to apply styles to a selector inside another selector.

It is for applying the styles to the child's of a parent element.

Here, we use whitespace between the selectors (first is parent, then child)

Syntax: -

```
parent child
{
        property:value;
}
```

Note:

Here, the parent selector might be any selector (ID, CLASS, ELEMENT or any).

It can be a simple element.

It can be ID selector

It can be class selector

It can be again contextual selector.

```
b
{
        color:blue;
}
/* this is for any bold element */
```

```
p b
{
        color:red;
}
        /* this for only bold in paragraph */

<b>blue text with bold</b>   : blue as color for <b> tag
<p> paragraph <b>text</b></p>  : red as color for <b> tag
```

Note:

We have two types of child elements in any parent element.

* Direct child

* In-direct (child of child-grandchild)

Syntax: for all child's, we use whitespace

```
parent child
{
        /* CSS properties */
}
```

syntax : only direct childs ,we use >

```
parent > child
{
        /* CSS properties  */
}
```

```
<!doctype html>
<html>
	<head>
		<title>contextual selector</title>

		<style>
			#d1 b
			{
				/* all childs */
				color:blue;
			}

			#d1 > i
			{
			/* only direct childs*/
				color:red;
			}
		</style>
	</head>
	<body>
		<div id="d1">
			<b>bold in d1</b>
			<i>italic in d1</i>
			<div id="d2">
			<b>bold in d2</b>
```

```
                    <i>italic in d2</i>
                        <div id="d3">
                    <b>bold in d3</b>
                    <i>italic in d3</i>

                        </div>
                    </div>
                </div>

        </body>

</html>
```

Working with and tag

These two are used as containers in CSS.

If you apply styles to div or span, the same style is applied to all the elements under div or span

These two are for creating block of contents.

What is the difference?

 (sub document)

It is used as a container for CSS.

It consists of simple text as well as any html element.

It creates box (block), separates the surrounding content by line breaks.

Each will take full width of the current row (full line) by default.

Using CSS, you can change this format.

height

width

max-height

min-height

Syntax:

any content

any html tag

Note:

You can also nest tags. You can put one in another

SPAN:

It is similar to tag.

But it uses inline flow. It does not break the line.

It is for creating inline blocks.

We use for creating a small dialogs.

<!--embed any markup-->

Note:

you can nest inside the tag.

We can't nest inside tag.

Note:

You can also and as selector in CSS

```
div
{
    /* style for all div */
    background-color: blue;
    color:red;
}
span
{
    /* style for all span */
    font-family:arial;
    font-size:10;
}
```

**Example using **

```
<!doctype html>
<html >
<head>
    <title>CSS demo</title>
                        <style type="text/CSS">
                            body
                            {
                               background-color:#b5b5b5;
                            }
                            #main
                            {
                               width:100%;
                            }

                            #header
                            {
                               width:100%;
                               height:100px;
                               background-color:Aqua;
                               text-align:center;
                            }
                            #menu
                            {
```

```
    width:30%;
    min-height:500px;
    float:left;
    background-color:Olive;
}

#content
{
    width:60%;
    min-height:500px;
    float:left;
    background-color:White;
    color:Black;
}

.clear
{
    clear:both;
}
#footer
{
    background-color:Teal;
    color:Maroon;
    text-align:center;
    font-weight:bold;
```

```
                    }
                </style>

        </head>
<body>
<div id="main">

    <div id="header">
        <h1>Marlabs Software</h1>
    </div>

    <div id="menu">
    <a href="products.html">Products</a><br /><br /><br />
    <a href="services.html">Services</a><br /><br /><br />
    <a href="softwares.html">Softwares</a><br /><br /><br />
    </div>

    <div id="content">
        <h3>main content goes here</h3>
    </div>
```

```
    <div class="clear"></div>

    <div id="footer">
      Copyright &copy; Marlabs Software . All Rights Reserved.
    </div>
</div>
</body>
</html>
```

Positioning of HTML elements

It determines, where to place html element on the web page.

Types of positioning

* Absolute positioning
* Relative positioning

Till now, we used relative positioning.

That means, the elements will locate one after other.

We can't overlap the elements in Relative positioning.

To place (locate) the elements as per coordinate system, we use Absolute positioning.

Absolute positioning

In this case, we can place the element wherever you want.

We use X (LEFT) and Y(TOP) coordinates to place the elements .

We can also overlap the elements using Absolute positions.

Overlaps means, we can place one element over another element.

We can also change the position dynamically using absolute position.

Relative positioning

In this case, we can place the element based on the default flow of the documents.

In relative positioning, the elements are placed next to the previous elements.

Here, we can't overlap the elements.

Syntax: -

```
Selector
{
    position:absolute;
    left:50px;
    top:100px;
}
```

Working with z-index

If at all, we set the same coordinates to multiple elements using absolute position, all elements are overlapped.

Only one element will be visible.

To set, which element to be visible, we use z-index property

Using x-index, we can set the stack order.

Whichever element has more z-index value that will be visible.

Example

```
<html>
<head>
   <title></title>
   <style type="text/CSS">
      #image1
      {
         position:absolute;
         left:50px;
         top:50px;
         z-index:2;
      }

      #image2
      {
         position:absolute;
          left:50px;
         top:50px;
         z-index:1;

      }
   </style>
</head>
<body>
```

```
<img src="cam1.png" id="image1" />

<img src="plain.jpg" id="image2" />

</body>

</html>
```

Using this z-index, we can dynamically change multiple elements at the same location.

Assignment:

Create a Web site for User Management

It should have following pages

Index.html

It is a start page. Provides links to all other pages

Login.html

Provides the UI to enter username and password and login button

Register.html

Proivdes the UI to enter username,password,confirmpassword,email, gender,mobile and submit button

Contacts.html

provides the contact information

AboutUs.html

provides the web site information

Create the menu in the header section (horizontal menu).

This header, footer, sidebar (if there) must be visible in all the pages.

Apply Some CSS styles. (Good look & feel).

transition

text-shadow

border-radius

box-shadow

Note :

Create a common layout for this web site using

e.g.

header, menu, content area, footer

In header section, specify common header in all the pages

In menu section provide the links. It is also common in all the pages.

Content Area, provides content as per the corresponding page

footer section, provides the common footer in all the pages.