

Unit II

Style Sheet : Introduction, Advantages and applications of style sheet, CSS: Introduction, syntax of CSS with example, Type of style sheet (Internal, External and Inline), Units, Classes and Id attributes, Properties: Text, Font, Color, background, border, display, height, line-height, margin, width. CSS with HTML and XML.

Cascading Style Sheet (CSS)

CSS (Cascading Style Sheet) is a style sheet language that supports the formatting of HTML / XML documents for viewing on the web. CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML / XML document.

CSS is very easy to learn and use and it provides a quick straight forward way to style XML content for browsing. There are three versions of CSS: CSS1, CSS2 and CSS3.

Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and lay out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects. These effects are used to describe the presentation side of content based data, such as the data stored in XML documents.

One of the important ideas in CSS is that you can create a style sheet once for a given class of documents and then we can display any document that belongs to that class using style sheet.

Any XML document output is unformatted. Applying cascading style sheet can format this output.

Style sheet can directly linked with XML Document in the declaration part. The general format for the declaration of style sheet is as follows:

```
<? xml-stylesheet type = "text/css" href = "filename.css:?">
```

```
<style type = "text/css" media = "all">
```

Advantages of CSS

- **CSS saves time** – You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
- **Pages load faster** – If you are using CSS, you do not need to write HTML/XML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.
- **Easy maintenance** – To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
- **Multiple Device Compatibility** – Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions

of a website can be presented for handheld devices such as PDAs and cell phones or for printing.

- **Global web standards** – Now HTML attributes are being deprecated and it is being recommended to use CSS. So its a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.
- **Offline Browsing** – CSS can store web applications locally with the help of an offline cache. Using of this, we can view offline websites. The cache also ensures faster loading and better overall performance of the website.
- **Platform Independence** – The Script offer consistent platform independence and can support latest browsers as well.
- **Superior styles to HTML** – CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.

Syntax of CSS (Structure of CSS)

CSS has a very simple structure. CSS style –sheet consists of a rule that determines how a given set of style is applied to a document. The application of style rules is defined by **SELECTOR**, which is CSS construct that identifies portion of XML document or HTML document.

The general form of **SELECTOR** , style declaration using the CSS style rule is as follows:

SELECTOR

```
{  
property1: value1;  
property2: value2;  
property3: value3;  
.  
.  
.  
}
```

Where,

SELECTOR may be an element type attribute class or attribute id.

Property1, Property2, property3, are the list of properties called as style properties. In XML CSS style properties are display, width, height, border, margin, color, text and font.

Value1, value2, value3 are the corresponding values of the style properties.

Property-value pair linked using the **colon (:)** and separated by using **semi-colon (;)**.

Example :

```
p {  
    font-family: verdana;  
    font-size: 20px;  
}  
  
    email  
    {  
        font-size:12px;  
        font-style:italic;  
        font-weight:bold;  
        display:block;  
        color: blue; }
```

Example :**stud.xml**

```
<?xml version="1.0"?>  
<?xml-stylesheet type="text/css" href="stud2.css"?>  
<!DOCTYPE students SYSTEM "stud1.dtd">  
<students>  
    <firstname> AAA </firstname>  
    <lastname> BBB </lastname>  
    <email>ab123@gmail.com</email>  
    <firstname> CCC </firstname>  
    <lastname> DDD </lastname>  
    <email>cd123@gmail.com</email>  
</students>
```

stud1.dtd

```
<!ELEMENT students (firstname,lastname,email)>  
<!ELEMENT firstname (#PCDATA)>  
<!ELEMENT lastname (#PCDATA)>  
<!ELEMENT email (#PCDATA)>
```

stud2.css

```
students  
{  
    background-color: pink;  
}  
firstname,lastname,email  
{  
    font-size:35pts;  
    font-style:italic;  
    display:block;  
    color: blue;  
    margin-left: 50px; }
```

Type of style sheet (Internal, External and Inline)

Each successive style sheet type adopts all the attributes from the previous style sheet type and overrides any similar attribute. This allows designers to use a linked style sheet to control an entire site, an embedded style sheet to control a single page and an inline style sheet to control a single tag element.

The different types of style sheets are follow types of CSS

1. Internal CSS (Embedded CSS)

2. External CSS (Linked CSS)

3. Inline CSS

Internal CSS (Embedded CSS)

Internal style sheets are placed within the HTML code of the page they are to be applied to. The style sheet syntax comes between opening and closing <STYLE> Tags. These tags are placed whether in the <HEAD> section or between the </HEAD> and <BODY> tags.

Example:

```
<!DOCTYPE html>
```

```
<html>
<head>
  <style>
    body {background-color: powderblue;}
    h1 {color: blue;}
    p {color: red;}
  </style>
</head>
<body>

  <h1>This is a heading</h1>
  <p>This is a paragraph.</p>

</body>
</html>
```

External CSS (Linked CSS)

Linked style sheets exists as separate files that are linked to a page with <LINK> tag. They have the css extension and are referenced with an URL.

Inside the css file, the style attributes are contained within opening and closing <STYLE> tags. <LINK> tag within the <HEAD> tags. Attributes within the <LINK> tag include: the REL attribute, set to stylesheet; an HREF attribute, whose value is the URL to the css file; and a TYPE attribute, which is set to the MIME (Multipurpose Internet Mail Extensions) type, text/css. These are the only attributes that are needed.

An external style sheet is used to define the style for many HTML pages. With an external style sheet, you can change the look of an entire web site, by changing one file

Syntax of LINK tag used in HTML file:

```
<LINK REL=stylesheet HREF="file.css" TYPE="text/css">
```

Example:

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Here is how the **"styles.css"** looks:

```
body {
  background-color: lightblue;
}

h1 {
  color: blue;
  text-align: center;
}

p {
  font-family: verdana;
  font-size: 20px;
}
```

Inline Style Sheet

You can apply style sheet rules directly to any HTML element using style attribute of the relevant tag. This should be done only when you are interested to make a particular change in any HTML element only.

Rules defined inline with the element overrides the rules defined in an external CSS file as well as the rules defined in <style> element.

Example:-

```
<!DOCTYPE html>
<html>

  <head>
    <title>HTML Inline CSS</title>
  </head>
```

```
<body>
  <p style = "color:red;">This is red</p>
  <p style = "font-size:20px;">This is thick</p>
  <p style = "color:green;">This is green</p>
  <p style = "color:green;font-size:20px;">This is thick and green</p>
</body>

</html>
```

Units, Classes and Id attributes

CSS Units

CSS has several different units for expressing a length. Many CSS properties take "length" values, such as width, margin, padding, font-size, border-width, etc. Length is a number followed by a length unit, such as 10px, 2em, etc.

A whitespace cannot appear between the number and the unit. However, if the value is 0, the unit can be omitted. For some CSS properties, negative lengths are allowed.

There are two types of length units: *relative* and *absolute*.

Relative Lengths

Relative length units specify a length relative to another length property. Relative length units scales better between different rendering mediums.

Unit	Description
1. Em-	Relative to the font-size of the element (2em means 2 times the size of the current font)
2. ex -	Relative to the x-height of the current font (rarely used)
3. ch -	Relative to width of the "0" (zero)
4. rem -	Relative to font-size of the root element
5. vw -	Relative to 1% of the width of the viewport*
6. vh -	Relative to 1% of the height of the viewport*
7. vmin -	Relative to 1% of viewport's* smaller dimension
8. vmax -	Relative to 1% of viewport's* larger dimension
9. %	

Example:

```
p {
  font-size: 16px;
  line-height: 2em;
}
```

Absolute Lengths

The absolute length units are fixed and a length expressed in any of these will appear as exactly that size.

Absolute length units are not recommended for use on screen, because screen sizes vary so much. However, they can be used if the output medium is known, such as for print layout.

Unit Description

1. cm - centimeters
2. mm - millimeters
3. in - inches (1in = 96px = 2.54cm)
4. px * - pixels (1px = 1/96th of 1in)
5. pt - points (1pt = 1/72 of 1in)
6. pc - picas (1pc = 12 pt)

(* Pixels (px) are relative to the viewing device)

Example:-

```
h2 {font-size: 10mm;}
```

```
h2 {font-size: 1cm;}
```

```
h1 {font-size: 0.5in;}
```

```
h2 {font-size: 30px;}
```

```
h2 {font-size: 25pt;}
```

```
h2 {font-size: 3pc;}
```

CSS Classes

It is possible to give an HTML element multiple looks with CSS

For Example, sometimes you want the font to be large and white , while other times you prefer the font to be small and black.

CSS allows you to do the same thing by using of classes.

Format for classes:

To create a class in CSS is simple. You need to add an extension to the typical CSS code and make sure you specify this extension in our HTML.

CSS Coding:

```
p.one{color: blue}  
p.two{color: green}
```

HTML Coding:

```
<HTML>  
<HEAD>  
<LINK REL="stylesheet" HREF="class.css" TYPE="text/css">  
</HEAD>  
<BODY>  
  <P> THIS IS NORMAL PARAGRAPH </P>  
  <P class="one"> THIS IS PARAGRAPH WITH BLUE COLOR </P>  
  <P class="two"> THIS IS PARAGRAPH WITH GREEN COLOR </P>  
</BODY>  
</HTML>
```

In the above example the same <P> tag can display three different styles of text which is only possible because of the class. The are useful for improving the functionality of the HTML tags further.

ID Attributes:

The ID attribute is nothing but the properties which are assigned to any entity. The #id selector allows you to target an element by referencing the id HTML attribute. ID attributes are prefixed with an Hash (#), more commonly known as a "pound sign".

Example:

HTML code:

```
<header id="site-header"> </header>
```

CSS Code:

```
#header { /* this is the ID selector */  
Background: green;}
```

ID attribute values should be unique i.e HTML does not allow two or more identical ids and will produce unpredictable results. If there are two of the same, CSS will still match and style both. Javascript however, when querying for an ID, will find the first and stop.

High specificity and uniqueness mean using #id is a CSS. Avoiding the #id selector in CSS is considered a best practice: it is preferable to use a class in nearly every case.

ID attributes have several uses outside of CSS:

- Providing unique hooks for Javascript.
- Element with id attributes can be targeted by anchor tags, by setting the href attribute to the id value, prefixed by a # symbol. Clicking that anchor link will refocus the current page on the element with the matching id.
- For truly unique elements in your HTML, such as form elements, IDs could be useful for things like linking labels and inputs.

Point to remember

- A valid #ID cannot start with a number.
- A valid # ID must be at least one character long.
- A large part of the Unicode are valid characters in an #ID
- #ID attributes and selectors are case-sensitive.
- #ID attributes must exactly match across HTML, CSS and Javascript.

CSS Stylesheet Properties:

(Text, Font, Color, background, border, display, height, line-height, margin, width)

CSS style properties are used for formatting characteristics of XML contents. Some of the properties are self-explanatory and does not need any explanation regarding their usage. CSS defines multiple style properties that are used to control fonts, color, alignment and margins.

Following are the most commonly used style properties supported in CSS:

***[Display, width, height,
Border, border-width, border-color, border-style, border-left, border-right,
border-top, margin, margin-left, margin-right, margin-top, margin-bottom,
background-color, color, text-align, text-indent, font-family, font-size, font-style]***

Font Properties:

The font properties are used to set the various parameters associated with the font in which name of the font and its family, font-size, font-style and font-weight are used.

Font-family :property is used change the face of a font.

Font-style: property is used to set style of a font ; normal, italic, underline.

Font-size: property specifies the size of the font with one of the unit.

Font-weight: property set the weight of the font; bold, bolder or lighter.

Example:

```
<html>
  <head>
    <link rel="stylesheet" href=file1.css" type="text/css">
  </head>
  <body>
    <p style="font-family:georgia;"> This is 1 paragraph </p>
  </body>
</html>
```

CSS Coding:

```
P{
font-family : georgia;
font-style : italic
font-weight : bold
font-size :20px }
```

Text Properties:

You can set following text properties of an element –

- The **color** property is used to set the color of a text.
- The **direction** property is used to set the text direction. Possible values are *ltr* or *rtl*.
- The **letter-spacing** property is used to add or subtract space between the letters that make up a word.
- The **word-spacing** property is used to add or subtract space between the words of a sentence.
- The **text-indent** property is used to indent the text of a paragraph.
- The **text-align** property is used to align the text of a document.
- The **text-decoration** property is used to underline, overline, and strikethrough text.
- The **text-transform** property is used to capitalize text or convert text to uppercase or lowercase letters.
- The **white-space** property is used to control the flow and formatting of text. Possible values are *normal*, *pre*, *nowrap*
- The **text-shadow** property is used to set the text shadow around a text.

Example:

```
color:red
direction:rtl
letter-spacing:5px
word-spacing:5px
text-indent:1cm
text-align:right
text-decoration:underline
text-transform:capitalize
text-transform:uppercase
white-space:pre
text-shadow:4px 4px 8px blue
```

Color Property

The color property use for to change the color of text. It can accept the prefixed color or RGB values.

Example:

```
P{ color: blue}
h1{color : rgb(255,0,255)}
```

Background Properties

to set backgrounds of various HTML elements. You can set the following background properties of an element –

- The **background-color** property is used to set the background color of an element.
- The **background-image** property is used to set the background image of an element.
- The **background-repeat** property is used to control the repetition of an image in the background.
- The **background-position** property is used to control the position of an image in the background.
- The **background-attachment** property is used to control the scrolling of an image in the background.
- The **background property** is used as a shorthand to specify a number of other background properties.

Example:

```
background-color:yellow
background-color: #cccccc;
```

```
body {
  background-image: "css.jpg";
  background-repeat: repeat;
  background-position:100px;
  background-position:100px 200px;
  background-attachment: fixed;
}
```

Display Property

The display property specifies the display behavior (the type of rendering box) of an element. Common values for display ;inline, block, flex, grid etc

Example:

```
body {  
    display: inline;}  
display: block;
```

Line Height Property:

This property essentially defines the line spacing for multiple lines of text. The value for this property can be standard lengths or percentages

Example:

```
P{ line-height :1.5em}
```

Height & Width Properties

The height & width properties are used to set the size of the element box. They accept length or percentage values or the auto value that tells the browser to compute an acceptable size.

Example:

```
P{ width:400px;  
height:100px}
```

```
<html>  
<head>  
</head>  
<body>  
  <p style="width:400px; height:100px; border:1px solid red; padding:5px; margin:10px;">  
    This paragraph is 400pixels wide and 100 pixels high  
  </p>  
</body>  
</html>
```

Margin Property

The margin property defines the space around an HTML element

We have the following properties to set an element margin.

- The margin specifies a shorthand property for setting the margin properties in one declaration.
- The margin-bottom specifies the bottom margin of an element.
- The margin-top specifies the top margin of an element.
- The margin-left specifies the left margin of an element.
- The margin-right specifies the right margin of an element.

Example:

```
P{ margin-top:3.0em;  
margin- right:3.0em;  
margin- bottom:5.0em;  
margin- left:5.0em}
```

```
p{ margin:3.0em 3.0em 5.0em 5.0em}
```

Border Property

The border properties allow you to specify how the border of the box representing an element should look. There are three properties of a border you can change:

- The **border-color** specifies the color of a border.
- The **border-style** specifies whether a border should be solid, dashed line, double line, or one of the other possible values.
- The **border-width** specifies the width of a border.

Example:

- 1) `P{ border-width:4px;
border-style:solid;
border-color:blue}`
- 2) `p {border:4px solid red;}`