

## UNIT –II

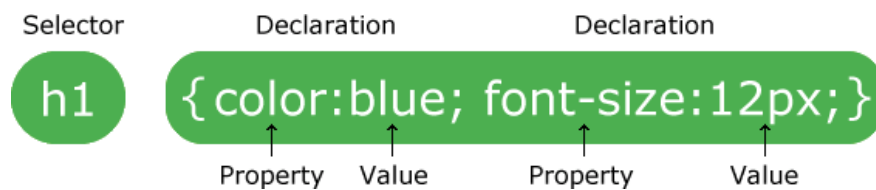
### CASCADING STYLE SHEETS

#### Introduction:

- ❖ CSS stands for Cascading Style Sheets.
- ❖ CSS describes how HTML elements are to be displayed on screen, paper, or in other media.
- ❖ CSS saves a lot of work. It can control the layout of multiple web pages all at once.
- ❖ External style sheets are stored in CSS files.
- ❖ CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

#### SYNTAX:

A CSS rule-set consists of a selector and a declaration block:



- ❖ The selector points to the HTML element you want to style.
- ❖ The declaration block contains one or more declarations separated by semicolons.
- ❖ Each declaration includes a CSS property name and a value, separated by a colon.
- ❖ A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.

#### Example:

In the following example all <p> elements will be center-aligned, with a red text color:

```
<!DOCTYPE html>
<html>
<head>
<style>      p { color: red; text-align: center; }  </style>
</head>
<body>
<p>Hello World!</p>
<p>Aditya Degree College Rajamahendravaram</p>
</body>
</html>
```

#### Advantages and Disadvantages of CSS:

- ❖ CSS allows you to specify how documents are presented to users by using various style properties for a given HTML element.
- ❖ CSS properties are the key to altering the styling of HTML elements in your web documents.
- ❖ **Separation of Style and Structure:** It allows separating content of an html document from the layout & style of that document.
- ❖ **Search engine friendly:** It helps get your pages found in a search engine. Since your CSS website contains less code and has a simpler structure which allows search engine spiders to pass from your code faster means your web pages can be indexed faster.

- ❖ **Save time:** It flexibility to set the properties of an component. You can write CSS code and then the same code can be applied to the HTML components groups, and also can be reused in different HTML pages.
- ❖ **Easy to Maintain:** It gives an easy mode to update document formatting and manage consistency across various documents. By doing one change to the website's CSS file, all the web pages' elements will be updated automatically.
- ❖ **Pages load faster:** It empowers multiple pages to share formatting, and reduce complexity and redundancy in the structural content. It significantly reduces the file transfer size, which helps to lode page faster.
- ❖ **Superior styles to HTML:** It has more extensive presentation capabilities than HTML, so you can give far superior look to your HTML pages in contrast with the HTML elements and attributes.
- ❖ **Multiple Device Compatibility:** It allows the HTML document to be improved for more than kind of device. Using it the same HTML document can be presented in various review styles for various rendering devices.
- ❖ **Provide Platform independence:** It offer consistent platform independence and also support latest browsers as well.
- ❖ **Global web standards:** It's a better to start using CSS in all the HTML pages to make them compatible for future browsers. There are various CSS properties which help to enhance your web experience.

## TYPES OF CSS

There are the following three types of CSS:

1. Inline CSS.
2. Internal / Embedded CSS.
3. External CSS.

### Inline CSS

- ❖ For Inline CSS every style content is in HTML elements.
- ❖ It is used for a limited section.
- ❖ Whenever our requirements are very small we can use inline CSS.
- ❖ It will affect only single elements.
- ❖ In HTML we require that various HTML tag's views are different so then we use inline Cascading Style Sheets.
- ❖ There are disadvantage of inline Cascading Style Sheets.
- ❖ It must be specified on every HTML tag.
- ❖ There is very much time consumed by that and it is not the best practice for a good programmer and the code will be quite large and very complex.

Examples

```
<!DOCTYPE html>
<html>
<body>
<h1 style="color:blue;">This is a Blue Heading</h1>
</body>
</html>
```

### Internal CSS

- ❖ In internal CSS the style of CSS is specified in the <head> section.
- ❖ This is internal CSS, it affects all the elements in the body section.
- ❖ Internal CSS is used in the condition when we want a style to be used in the complete HTML body.
- ❖ For that we can use style in the head tag.
- ❖ This style performs an action in the entire HTML body.

```
<!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

- ❖ In External CSS we create a .css file and use it in our HTML page as per our requirements.
- ❖ Generally external Cascading Style Sheets are used whenever we have many of HTML attributes and we can use them as required.
- ❖ There is no need to rewrite the CSS style again and again in a complete body of HTML that inherits the property of the CSS file.

Example:

styles.css

```

body { background-color: powderblue; }
h1 { color: blue; }
p { color: red; }

```

HTML Program

```

<!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>

```

Defining your own styles

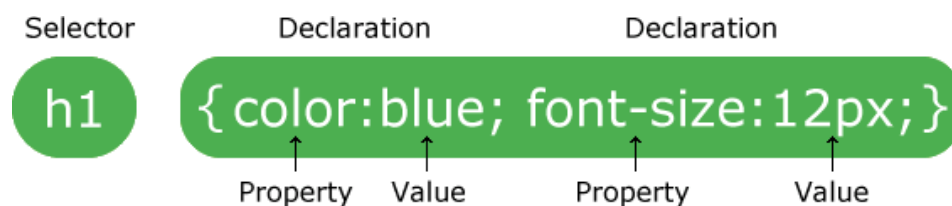
Styles are defined by simple rules. A style can contain as many rules as you want and with processing

### HTML. 1. Cascading styles:

- ❖ Conventionally styles are cascaded (overridden).
- ❖ This means that you do not have to use just a single set of rules inside a document, you can import as many styles as you like.
- ❖ The only difficulty with importing multiple style sheets is that they cascade.
- ❖ This means that the first is overridden by second, the second by third and so on.

### 2. Style Rule:

A style rule has two parts: The selector and a set of declarations. The selector is used to create a link between the rule and the HTML tag.



The declaration part has two parts: a property and a value.

Selectors can be placed into classes so that a tag can be formatted in variety of ways. Declarations must be separated using colons and terminated using semicolon.

### 3. Classes

- A class is defined a set of styles which can be applied as user choice.
- Choices can be applied to a single type of element or may be anonymous and hence applicable to any element.
- If you want to apply a style to a particular element use the following syntax  
Selector.classname{property: value; property: value}  
<selector class="classname">

```
<!DOCTYPE html>
<html>
<head>
<style> p.center { text-align: center; color: red; } </style>
</head>
<body>
<h1 class="center">This heading will not be affected</h1>
<p class="center">This paragraph will be red and center-aligned.</p>
</body>
</html>
```

### 4. Anonymous classes

- Sometimes you want to apply a piece of formatting too many different elements within a page but not necessarily to the entire page.
- Instead of defining styles to individual elements we are creating a style class that can be applicable any number of tags in your HTML document.
- Such classes are called anonymous classes. It has the following syntax  
.classname { Property: value; Property: value; }  
<selector1 class="classname"> ..... <selector1>  
<selector2 class="classname"> ..... <selector2>

Example:

```
<!DOCTYPE html>
<html>
<head>
<style> .center { text-align: center; color: red; } </style>
</head>
<body>
<h1 class="center">Red and center-aligned heading</h1>
<p class="center">Red and center-aligned paragraph.</p>
</body>
</html>
```

### Including style sheets

We can include style sheets in our HTML page in two way

- by linking
- by importing

#### By linking

We can link our .css file in the head section of a HTML page using the following syntax

```
<link rel="style sheet" href="URL" type="text/css" media="screen">
```

- The href is hyperlink to your style sheet
- Rel tells to the browser what type of link you're using
- Type tells to the browser what type of document you're including.
- the 'type' statement gives the relevant MIME type.
- HTML Specifies a Variety of ways of using a document a document, including screen viewing, printing and as a presentations.
- Use the 'media' attribute to describe the type of use.

```

<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" type="text/css" href="mystyle.css" media="screen">
  <style> h1 { color: orange; } </style>
</head>
<body>
  <h1>This is a heading</h1>
  <p>The style of this document is a combination of an external stylesheet, and internal style</p>
</body>
</html>

```

### By importing

```
[<style type="text/css">] <!--@import url(url); --> </style>
```

These lines are both needed if you intended to use more than one style sheet.

The first sheet is included as if it were the only one; any further style sheets have to be imported.

Example:

```

<link rel="stylesheet" type="text/css" href="mystyle.css" media="screen">
<style type="text/css">
  <!--@import url("http://www.smiggins.co./style.css"); --> </style>

```

## Properties and Values

There are number of style properties, including properties that control lists and positing floating elements like images.

Font properties:

- font-family: family name;  
Fonts are identified by giving the name of a specific font.
- font-style: normal | italic | oblique;  
This property provides the style to the font.
- font-weight: normal | bold | bolder | lighter;  
The font-weight property selects the weight or darkness of the font. Values of the property range from 100 to 900 in increments of 100.
- font-size: small | medium | large | smaller | larger;  
This property is used to set the relative or physical size of the font used.

Example:

```

<html>
<head>
<title> Font properties </title>
<style type="text/css">
  h1{ font-family:arial; font-style: italic; }
  P{ Font-weight: bold Font-size:100px; }
</style>
</head>
<body>
  <h1> demo from css properties </h1>
  <p> this paragraph represents font size and font weight</p>
</body>
</html>

```

## Backgrounds and color properties

Color:<value>

Background-color:<value> | transparent

Background-image: URL | none

- The color of any attribute can be changed. Value should be given as hexadecimal values.

- Backgrounds for the whole page or individual element can have their color set from the style sheet.
- Elements can also have transparent backgrounds.
- Instead of a color an image can be used, identified by its URL.

Example:

```
<html>
<head>
<title> Font properties </title>
<style type="text/css">
    h1{ background-color: white; color: black; }
    P{ background-color: black; color: white; }
</style>
</head>
<body>
    <h1> demo from css background and color properties </h1>
    <p> this paragraph represents font size and font weight</p>
</body>
</html>
```

## Text Properties

Text properties are used to effect the presentation, spacing and layout of text. The basic properties includes such as decoration, indentation, word spacing, letter spacing, spacing between lines, horizontal and vertical text alignment.

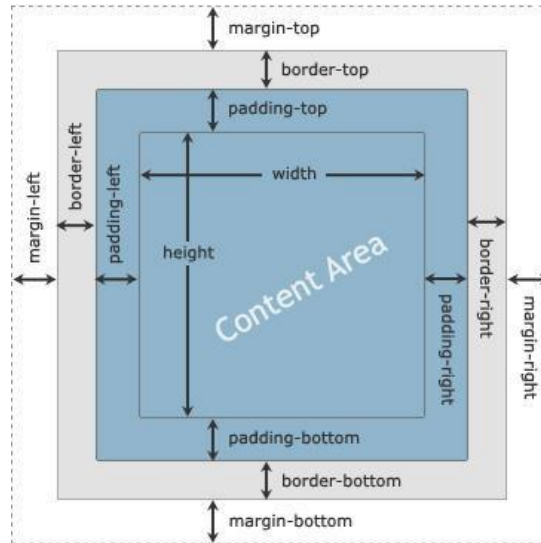
- text-decoration: none | underline | overline| line-through| blink;
- text-transformation: none | uppercase | lowercase| capitalize;
- text-align: left | right | center | justify;
- text-indentation: length | percentage;
- line-height : length | percentage;
- letter-spacing: length | percentage;
- word-spacing: length | percentage;

Example

```
<html>
    <head>
        <title> text properties </title>
        <style type="text / css">
            a{ text-decoration: overline;
                text-transformation: uppercase;
                text-align: left;
                text-indentation: 12px; }
        </style>
    </head>
    <body>
        <a href="D:/CSS Demo/My css.html"> This hypertext has text properties </a>
    </body>
</html>
```

## Border and Margin Properties:

The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content. The image below illustrates the box model:



- Margin: length | percentage | auto { 1,4 }
- Border-width: thin | thick | medium | length { 1,4 }
- Padding: length | percentage { 1,4 }

Padding - Clears an area around the content. The padding is transparent

Border - A border that goes around the padding and content

Margin - Clears an area outside the border. The margin is transparent

Border-color: value { 1,4 }

Border-style: none | dotted | solid | double | groove Ridge { 1,4 }

This sets the color of the border around the element.

Up to four different colors can be specified.

They are applied to the borders in the same orders as margins.

Each edge of the border can have a different style.

Width: length | percentage | auto Height: length | auto

Example:

```
<!DOCTYPE html>
<html>
<head>
<style> div { width: 320px; padding: 10px; border: 5px solid gray; margin: 0; } </style>
</head>
<body>
  <h2>Calculate the total width:</h2>
  
  <div>The picture above is 350px wide. The total width of this element is also 350px.</div> </body>
</html>
```

## Formatting Blocks of Information

- 🚦 A webpage may consist of more than one block.
- 🚦 We can format all the blocks with single style or separate style for each and every block.
- 🚦 This can be achieved in three ways
  1. Using style class
  2. Using <div> tag
  3. Using <span> tag

## Using class

A class is defined a set of styles which can be applied as user choice.

Choices can be applied to a single type of element or may be anonymous and hence applicable to any element.

If you want to apply a style to a particular element use the following syntax

```
Selector.classname{property: value; property: value}  
<selector class="classname">
```

```
<!DOCTYPE html>  
<html>  
<head>  
<style> p.center { text-align: center; color: red; } </style>  
</head>  
<body>  
<h1 class="center">This heading will not be affected</h1>  
<p class="center">This paragraph will be red and center-aligned.</p>  
</body>  
</html>
```

## Anonymous classes

Sometimes you want to apply a piece of formatting too many different elements within a page but not necessarily to the entire page.

Instead of defining styles to individual elements we are creating a style class that can be applicable any number of tags in your HTML document.

Such classes are called anonymous classes. It has the following syntax

```
.classname { Property: value; Property: value; }  
<selector1 class="classname"> ..... <selector1>  
<selector2 class="classname"> ..... <selector2>
```

Example:

```
<!DOCTYPE html>  
<html>  
<head>  
<style> .center { text-align: center; color: red; } </style>  
</head>  
<body>  
<h1 class="center">Red and center-aligned heading</h1>  
<p class="center">Red and center-aligned paragraph.</p>  
</body>  
</html>
```

## Using <div> tag

An element in an HTML document is either a block element or an inline element.

A block would be something like a paragraph, while an inline might be something like text, a figure or an individual character i.e., part of block.

Each of these can be manipulated separately, rather than applying the formatting to the element itself, a <div> ..... </div> pair of tags are wrapped around the elements.

Any formatting that needs adding is placed inside the <div> tag. The <div> tag has the following syntax:

```
<div class="any element"> <p> ..... </p> <p> ..... </p> <hr> </div>
```

Example:

```
<html>  
<head>  
<title> div tag css demo </title>  
<head>  
  <style type="text / css">  
    .fred { color: white; background-color: 009900; margin: 2px; font-size: 25px; }  
  </style>
```



```

</head>
<body>
  <div class "fred">
    <h3> this is heading inside division </h3>
    <p> The div tag is known as Division tag. The Div tag is used in HTML to make divisions
    of content in the web page like (text, images, header, footer, navigation bar etc).</p>
  </div>
</body>
</html>

```

### Using <span> tag:

The <span> tag has very similar properties to the <div> tag. If you want to group text without using elements use a <span> tag.

It has the following syntax <span [id="....."][class="....." | style="....."]>..... </span>

What is the difference between <span> and <div>?

Span as an in-line element and a div as a block level element. Span:






1. The span element basically occupies only the amount of space required by the element , it will not take the whole space/width of the web page .
2. Second thing is span doesn't breaks the line , i.e. all the spans will be shown in single line in a web page as shown in the screenshot below.
3. Div: The div element occupies all the width available to it on the web page and every time a new div element is made it breaks the line and starts in a new line.

```

<!DOCTYPE html>
<html>
<head>
<title>HTML span Tag</title>
</head>
<body>
<p>This is a paragraph <span style = "color:#FF0000;"> This is a paragraph</span>This is a
paragraph</p>
<p><span style = "color:#8866ff;"> This is another paragraph</span></p>
</body>
</html>

```

## Layers in HTML

-  The page layout that a browser creates results from layering text and images on the top of each other.
-  This lets web designers use images as the backgrounds of their pages and then place further images and text over them.
-  By extending the idea slightly we can place text items and images on top of each other in multiple layers. Netscape has extended the HTML standard by adding the layer tag.
-  The layer tag is browser specific and it use leads to confusion with the more general idea of layers.
-  The following are the properties and values of layers.

Z-index: n The browser maintains a stack of layers of content.

Position: absolute | relative

The placement of the layer may be absolute or relative. The default is absolute.

Left: n,

Top: n

The location of the division in pixels. We can specify the position of their top-left corner.

Width: n,

Height: m

The size of division in pixels defaults to the document of space needed to display the content of the division.

Example:

```

<!DOCTYPE html>
<html>

```

```
<head>
<title>HTML layer Tag</title>
</head>
<body>
  <div style="z-index:2; left:50px; top:250px; position:absolute; color:red; background-color:white;
    font-size:36pt; border:thin-groove;">
    <p> This is the higher layer </p>
  </div>
  <div style="z-index:1; left:100px; top:255px; position:absolute; color:magenta; background-
    color:green; font-size:46pt; border:thin-groove;">
    <p> some mor text</p>
  </div>
  <div style="z-index:4; left:10px; top:40px; width:150px; position:absolute; color:black; background-
    color:yellow; font-size:18pt;">
    <p> some mor text plcaed in a box that doesn't go right across the screen</p>
  </div>
  <div style=" left:500px; top:300px; width:25px; position:absolute; color:blue; background-
    color:#aeae00; font-size:16pt; font-style:italic;z-index:2; ">
    <p> And in the bottom right corner.....</p>
  </div>
</body>
</html>
```