

# NOTES

Grade: 10

Chap-6 – CSS-Cascading Style Sheet

Sub: Comp. Applications

## Static Vs Dynamic Webpage:

**Static Webpage or websites** appear or display the same content for every user that accesses them and only change when a developer modifies the source files. A static webpage remains the same or fixed, in terms of the content it displays.

A **Dynamic Webpage** is the opposite, its content changes according to the location of the user, or based on actions a user has made on the page before.

## DHTML (Dynamic HTML)

DHTML is a combination of HTML, CSS, JavaScript and DOM (Document Object Model). DHTML technologies are utilized to create interactive and dynamic web pages.

## Utility of Dynamic Websites

The utility of dynamic websites are as follows:

- Dynamic websites allow easy content updates.
- They tailor content based on user preferences.
- They enable feedback forms, comment sections, and social media integration.
- They can fetch and display real-time data from databases or APIs enabling live updates like news feeds, stock market information, weather updates, etc.
- They support online transactions and inventory management.
- They handle large amounts of content and user traffic.
- They streamline processes and save time.

## Some features of Dynamic Websites

- Some features of dynamic websites are as follows:
- User registration and authentication can be done.
- Content management system (CMS) provides for easy content updates.
- Search functionality to find specific content.
- Database integration for dynamic content generation.
- Interactive forms for user engagement.
- E-commerce capabilities for online transactions.
- Social media integration for sharing and interaction.
- Personalization based on user preferences.
- Dynamic content delivery for customized user experiences.
- Analytics and tracking to gather data for analysis and optimization.

## What do you understand by Stylesheets? How are these useful?

A style sheet is a file containing formatting guidelines that define the overall look of a document.

**Style sheets are useful in the following ways:**

- It helps to separate structure and presentation. The HTML file can include structure tags and style sheet takes care of the presentation of content.
- Web pages download much faster.
- Developers have to type less code, and the web pages are shorter and neater.
- The look of the site is kept consistent throughout all the pages that work off the same style sheet.
- Updating design and general site maintenance are made much easier.
- Errors caused by editing multiple HTML pages occur less often.

### What is CSS style rule? How do you define it?

A CSS rule is a single statement in a style sheet that identifies what should be styled (the selector) and how those styles should be applied (the declaration).

We define a rule by writing the selector tag without angle brackets. The properties and their values are written in the following syntax:

```
selector { propertyname : value ; propertyname : value ; ... }
```

For example, if we want <H3 tag to have font face Arial and red color then we define the style rule as follows:

```
h3 { font.family : Arial ; color : red ; }
```

### What is CSS?

- 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 stylesheets are stored in **CSS files**

### What can I do with CSS?

- CSS is a style language that defines layout of HTML documents. For example, CSS covers fonts, colours, margins, lines, height, width, background images, advanced positions, and many other things. Just wait and see!
- HTML can be used to add layout to websites. But CSS offers more options and is more accurate and sophisticated. CSS is supported by all browsers today.
- After only a few lessons of this tutorial you will be able to make your own style sheets using CSS to give your website a new great look.

### How to use CSS?

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



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

### **CSS can be used**

#### **1) Internal CSS - Inside HTML file by using <STYLE> </STYLE> tag and**

Syntax:

#### Syntax to use Internal CSS

```
<STYLE type="text/css">
  Tagname      {
                Property: value;
                Property: value
              }
</STYLE>
```

## 2) External CSS - Creating external CSS file by using saving with .CSS extension

Syntax:

### In CSS file:

```
Tagname    {  
                Property: value;  
                Property: value  
            }
```

### In HTML file

```
<link rel="stylesheet" type="text/css" href="mystyle.css">
```

## 3) Inline CSS – using STYLE attribute inside tag.

Syntax:

```
<H1 STYLE= "color : red ; font-family : arial "> Welcome to CSS <H1>
```

## CSS Text Properties

Text-align	: left   right   center   justify
Text-decoration	: none   underline   overline   line-through
Text-transform	: uppercase   lowercase   capitalize
Text-indent	: n px (n is any integer value)
Letter-spacing	: n px (n is any integer value)
Word-spacing	: n px (n is any integer value)

## CSS Font Properties

Font-family	: <i>fontname</i>
Font-style	: normal   italic   oblique
Font-weight	: normal   bold
Font-size	: x pt
Font-variant	: normal   small-caps

## CSS Display Property

Display	: inline   block   none
---------	-------------------------

## CSS Background Property

Background-color	: colorname
Background-image	: url(path) e.g. → Background-image : url("c:\\abc\\mypic.jpg")
Background-repeat	: repeat-x   repeat-y   no-repeat
Background-position	: left   right   center top   center   bottom
Background-attachment	: fixed   none

### **CSS Border Property**

Border-color	: colorname	
Border	: n px	(n is any integer value)
Border-style	: none   dotted   solid   dashed   double   groove   ridge   inset   outset	
Border-radius	: n px	(n is any integer value)
Border-width	: n px	(n is any integer value)

### **CSS Margin Property (for outside space of content)**

Margin-left	: n px	(n is any integer value)
Margin-right	: n px	(n is any integer value)
Margin-top	: n px	(n is any integer value)
Margin-bottom	: n px	(n is any integer value)

### **CSS Padding Property (for providing space between content and border)**

Padding-left	: n px	(n is any integer value)
Padding -right	: n px	(n is any integer value)
Padding -top	: n px	(n is any integer value)
Padding -bottom	: n px	(n is any integer value)

### **CSS List Property**

List-style-type	: circle   square   upper-roman   lower-roman   upper-alpha   lower-alpha	
List-style-position	: inside   outside	
List-style-image	: url(path) e.g. → Background-ima	List-style-image: url("c:\\abc\\mypic.jpg")

### **Grouping of Elements using Class Attribute:**

In CSS, we can identify specific elements and group them according to our requirements. This feature helps us in applying element-specific styling i.e., applying a style to particular element and other style to other element. This is made possible using the class and id attributes. These are also called CSS Selectors.

Identification of elements is done with class and id properties.

Identification and grouping of elements is useful when we want to divide our web page into different sections or sub sections.

We use class or id to find or select HTML elements to apply the required styles.

The advantage of using selectors is that we can present the same element differently depending on its class or id.

#### **Identification using class**

- We can find elements using a class name.
- We can use the same class on multiple elements.
- We can also use multiple classes on single element.
- In CSS, a class name is preceded by a period (.)

**An example would make you understand better,**

```
<html>
  <head>
    <style type="text/css">
      p.red {
          color: red;
      }
      p.blue {
          color: blue;
      }
      p.green{
          color: green;
      }
    </style>
  </head>
  <body>
    <p class="red"> The Color is red. Applied class name is also Red. </p>
    <p class="blue"> The Color is blue. Applied class name is also Blue. </p>
    <p class="green"> The Color is Green. Applied class name is also Green. </p>
  </body>
</html>
```

As you can see from the above example, we have applied different styles for different paragraphs using the attribute "class".

**The output of the above program would be,**

The Color is red. Applied class name is also Red.

The Color is blue. Applied class name is also Blue.

The Color is Green. Applied class name is also Green.