

## **Day 5-Workshop-Learning-Content**

**Description:** This document will give knowledge about Cascading Style Sheet (CSS). You will learn what is CSS? Why and How to use CSS? What are selectors? Rendering order of CSS and properties like color, backgrounds, borders, margin, padding, height and width.

### **Table of Contents**

- 1. Introduction to Cascading Style Sheet (CSS)**
  - a. What is CSS
  - b. Why use CSS
  - c. How to use CSS
- 2. CSS Syntax and Selectors**
  - a. Syntax
  - b. Selectors – element, id and class
- 3. Comments in CSS**
- 4. Handling multiple Style Sheets**
- 5. CSS Color**
- 6. CSS Background**
- 7. CSS Border**
- 8. CSS Margin**
- 9. CSS Padding**
- 10. CSS Height and Width**
- 11. CSS Box Model**

#### **Disclaimer:**

- The information given here is open source information to help the students in learning & building skills.
- Knowledge is collected from the best available resources on the Internet and can still have mistaken all errors must be reported to the distributor.

## 1. Introduction to Cascading Style Sheet (CSS):

- a. **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 style sheets are stored in CSS files
- b. **Why use CSS:** CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.
- c. **How to use CSS:** There are three ways of inserting a style sheet:
  - i. External style sheet
  - ii. Internal style sheet
  - iii. Inline style

**External Style Sheet:** With an external style sheet, you can change the look of an entire website by changing just one file. Each page must include a reference to the external style sheet file inside the <link> element. The <link> element goes inside the <head> section.

For Example:

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

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

**Internal Style Sheet:** An internal style sheet may be used if one single page has a unique style. Internal styles are defined within the <style> element, inside the <head> section of an HTML page

For example:

```
<head>
<style>
body {
    background-color: linen;
}

h1 {
    color: maroon;
}
```

### Disclaimer:

- The information given here is open source information to help the students in learning & building skills.
- Knowledge is collected from the best available resources on the Internet and can still have mistaken all errors must be reported to the distributor.

```

        margin-left: 40px;
    }
</style>
</head>

```

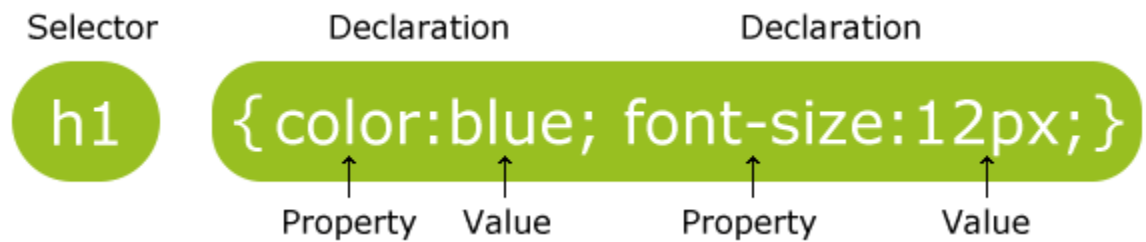
**Inline Styles:** An inline style may be used to apply a unique style for a single element. To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

For Example: The example below shows how to change the color and the left margin of a <h1> element

```
<h1 style="color:blue;margin-left:30px;">This is a heading</h1>
```

## 2. CSS Syntax and Selectors:

- a. **CSS 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.

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

```

p {
    color: red;
    text-align: center;
}

```

- b. **CSS Selectors:** CSS selectors are used to "find" (or select) HTML elements based on their element name, id, class, attribute, and more.

- i. **The element Selector:** The element selector selects elements based on the element name.

For example: You can select all <p> elements on a page like this (in this case, all

### Disclaimer:

- The information given here is open source information to help the students in learning & building skills.
- Knowledge is collected from the best available resources on the Internet and can still have mistaken all errors must be reported to the distributor.

<p> elements will be center-aligned, with a red text color)

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

- ii. **The id Selector:** The id selector uses the id attribute of an HTML element to select a specific element. The id of an element should be unique within a page, so the id selector is used to select one unique element. To select an element with a specific id, write a hash (#) character, followed by the id of the element.

For Example: The style rule below will be applied to the HTML element with id="container".

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

- iii. **The class Selector:** The class selector selects elements with a specific class attribute. To select elements with a specific class, write a period (.) character, followed by the name of the class.

For Example: In the example below, all HTML elements with class="center" will be red and center-aligned

```
.center {  
    text-align: center;  
    color: red;  
}
```

- iv. **Grouping Selectors:** If you have elements with the same style definitions, like this:

```
h1 {  
    text-align: center;  
    color: red;  
}  
  
h2 {  
    text-align: center;  
    color: red;  
}
```

**Disclaimer:**

- The information given here is open source information to help the students in learning & building skills.
- Knowledge is collected from the best available resources on the Internet and can still have mistaken all errors must be reported to the distributor.

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

It will be better to group the selectors, to minimize the code. To group selectors, separate each selector with a comma. In the example below we have grouped the selectors from the code above

```
h1, h2, p {
    text-align: center;
    color: red;
}
```

3. **Comments in CSS:** Comments are used to explain the code, and may help when you edit the source code at a later date. Comments are ignored by browsers.

A CSS comment starts with `/*` and ends with `*/`. Comments can also span multiple lines

For Example:

```
p {
    color: red;
    /* This is a single-line comment */
    text-align: center;
}

/* This is
a multi-line
comment */
```

4. **Handling multiple Style Sheets:** If some properties have been defined for the same selector (element) in different style sheets, the value from the last read style sheet will be used.

For Example:

Assume that an external style sheet has the following style for the `<h1>` element

```
h1 {
    color: navy;
}
```

then, assume that an internal style sheet also has the following style for the `<h1>` element

```
h1 {
    color: orange;
}
```

**Disclaimer:**

- The information given here is open source information to help the students in learning & building skills.
- Knowledge is collected from the best available resources on the Internet and can still have mistaken all errors must be reported to the distributor.

If the internal style is defined after the link to the external style sheet, the <h1> elements will be "orange"

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
<style>
h1 {
    color: orange;
}
</style>
</head>
```

However, if the internal style is defined before the link to the external style sheet, the <h1> elements will be "navy"

```
<head>
<style>
h1 {
    color: orange;
}
</style>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

**Cascading Order:** when there is more than one style specified for an HTML element, then all the styles will "cascade" into a new "virtual" style sheet by the following rules, where number one has the highest priority:

- I. Inline style (inside an HTML element)
- II. External and internal style sheets (in the head section)
- III. Browser default

5. **CSS Color:** Colors in CSS are most often specified by:

- I. a valid color name - like "red"
- II. an RGB value - like "rgb(255, 0, 0)"
- III. a HEX value - like "#ff0000"

6. **CSS Background:** The CSS background properties are used to define the background effects for elements.

**Disclaimer:**

- The information given here is open source information to help the students in learning & building skills.
- Knowledge is collected from the best available resources on the Internet and can still have mistaken all errors must be reported to the distributor.

CSS background properties:

- I. **background-color:** The background-color property specifies the background color of an element.

**For Example:**

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

- II. **background-image:** The background-image property specifies an image to use as the background of an element. By default, the image is repeated so it covers the entire element.

**For Example:**

```
body {  
    background-image: url("desert.jpg");  
}
```

- III. **background-repeat:** By default, the background-image property repeats an image both horizontally and vertically. To repeat the image above horizontally, set background-repeat: repeat-x;

```
body {  
    background-image: url("desert.jpg");  
    background-repeat: repeat-x;  
}
```

To repeat an image vertically, set background-repeat: repeat-y;

```
body {  
    background-image: url("desert.jpg");  
    background-repeat: repeat-y;  
}
```

To use the image only once, set background-repeat: no-repeat;

```
body {  
    background-image: url("desert.jpg");  
    background-repeat: no-repeat;  
}
```

**Disclaimer:**

- The information given here is open source information to help the students in learning & building skills.
- Knowledge is collected from the best available resources on the Internet and can still have mistaken all errors must be reported to the distributor.

- IV. **background-position:** The position of the image is specified by the background-position property:

```
body {  
    background-image: url("desert.jpg");  
    background-repeat: no-repeat;  
    background-position: right top;  
}
```

- V. **background-attachment:** To specify that the background image should be fixed (will not scroll with the rest of the page), use the background-attachment property.

```
body {  
    background-image: url("desert.jpg");  
    background-repeat: no-repeat;  
    background-position: right top;  
    background-attachment: fixed;  
}
```

**Background - Shorthand property:** To shorten the code, it is also possible to specify all the background properties in one single property. This is called a shorthand property. The shorthand property for background is background:

```
body {  
    background: #ffffff url("desert.jpg") no-repeat right  
top;  
}
```

7. **CSS Border:** The CSS border properties allow you to specify the style, width, and color of an element's border.

- a. **Border-style:** The border-style property specifies what kind of border to display.

**The following values are allowed:**

- I. dotted - Defines a dotted border
- II. dashed - Defines a dashed border
- III. solid - Defines a solid border
- IV. double - Defines a double border
- V. groove - Defines a 3D grooved border. The effect depends on the border-color value
- VI. ridge - Defines a 3D ridged border. The effect depends on the border-color value
- VII. inset - Defines a 3D inset border. The effect depends on the border-color

**Disclaimer:**

- The information given here is open source information to help the students in learning & building skills.
- Knowledge is collected from the best available resources on the Internet and can still have mistaken all errors must be reported to the distributor.



- value
- VIII. outset - Defines a 3D outset border. The effect depends on the border-color value
- IX. none - Defines no border
- X. hidden - Defines a hidden border

For Example:

```
p {  
    border-style: solid;  
}
```

- b. **border-width:** The border-width property specifies the width of the four borders. The width can be set as a specific size (in px, pt, cm, em, etc) or by using one of the three pre-defined values: thin, medium, or thick. The border-width property can have from one to four values (for the top border, right border, bottom border, and the left border).

For Example:

```
p {  
    border-style: solid;  
    border-width: 5px;  
}
```

- c. **border-color:** The border-color property is used to set the color of the four borders. The color can be set by:

- I. name - specify a color name, like "red"
- II. Hex - specify a hex value, like "#ff0000"
- III. RGB - specify a RGB value, like "rgb(255,0,0)"
- IV. transparent

The border-color property can have from one to four values (for the top border, right border, bottom border, and the left border).

If border-color is not set, it inherits the color of the element.

For Example:

```
p {  
    border-style: solid;  
    border-width: 5px;  
    border-color: red;  
}
```

**Disclaimer:**

- The information given here is open source information to help the students in learning & building skills.
- Knowledge is collected from the best available resources on the Internet and can still have mistaken all errors must be reported to the distributor.

**Border - Shorthand Property:** The border property is a shorthand property for the following individual border properties:

- I. border-width
- II. border-style (required)
- III. border-color

**For Example:**

```
p {  
    border: 5px solid red;  
}
```

**8. CSS Margin:** The CSS margin properties are used to generate space around elements. The margin properties set the size of the white space outside the border.

CSS has properties for specifying the margin for each side of an element:

- I. margin-top
- II. margin-right
- III. margin-bottom
- IV. margin-left

**For Example:**

```
p {  
    margin-top: 100px;  
    margin-bottom: 100px;  
    margin-right: 150px;  
    margin-left: 80px;  
}
```

**Margin - Shorthand Property:** The sequence is a default one, browser will follow the same as given below:

- I. margin-top
- II. margin-right
- III. margin-bottom
- IV. margin-left

All the margin properties can have the following values:

- I. auto - the browser calculates the margin
- II. length - specifies a margin in px, pt, cm, etc.
- III. % - specifies a margin in % of the width of the containing element
- IV. inherit - specifies that the margin should be inherited from the parent element

**For Example:**

**Disclaimer:**

- The information given here is open source information to help the students in learning & building skills.
- Knowledge is collected from the best available resources on the Internet and can still have mistaken all errors must be reported to the distributor.

```
p {
    margin: 100px 150px 100px 80px;
}
```

9. **CSS Padding:** The CSS padding properties are used to generate space around content. The padding clears an area around the content (inside the border) of an element.

CSS has properties for specifying the padding for each side of an element:-

- I. padding-top
- II. padding-right
- III. padding-bottom
- IV. padding-left

All the padding properties can have the following values:

- I. length - specifies a padding in px, pt, cm, etc.
- II. % - specifies a padding in % of the width of the containing element
- III. inherit - specifies that the padding should be inherited from the parent element

**For Example:**

```
p {
    padding-top: 50px;
    padding-right: 30px;
    padding-bottom: 50px;
    padding-left: 80px;
}
```

**Padding - Shorthand Property:** - The padding shorthand property is a smarter way of writing padding properties given below:-

- I. padding-top
- II. padding-right
- III. padding-bottom
- IV. padding-left

**For Example :**

```
p {
    padding: 50px 30px 50px 80px;
}
```

10. **CSS Height and Width:** The height and width properties are used to set the height and width of an element. The height and width can be set to auto (that is default. Means that the browser calculates the height and width), or be specified in length values, like px, cm, etc., or in percent (%) of the containing block.

**For Example:**

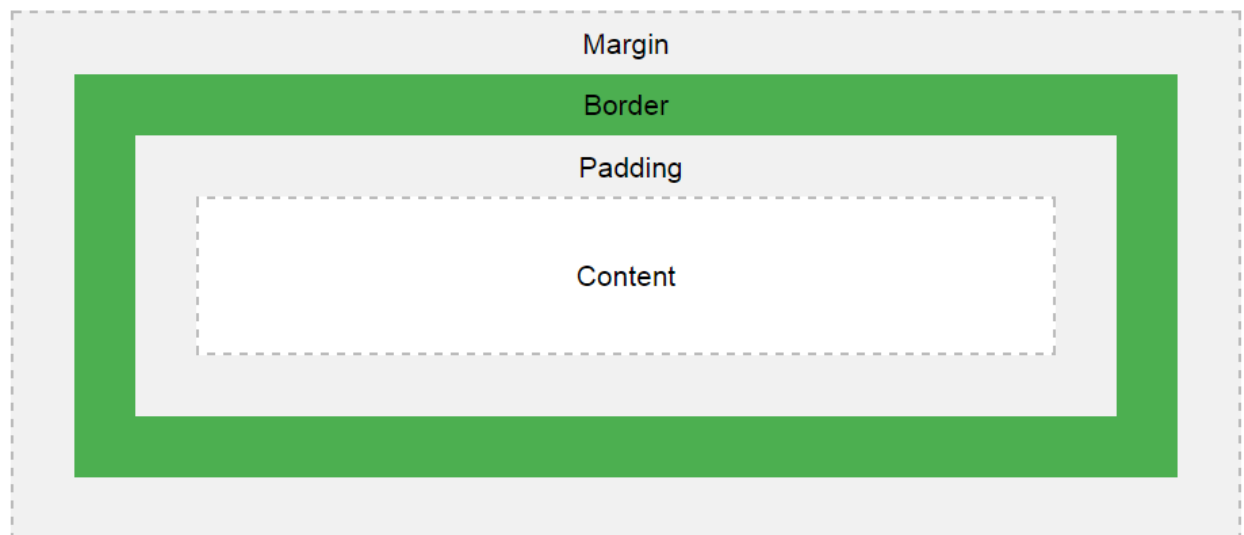
**Disclaimer:**

- The information given here is open source information to help the students in learning & building skills.
- Knowledge is collected from the best available resources on the Internet and can still have mistaken all errors must be reported to the distributor.

```
div {  
  height: 200px;  
  width: 50%;  
  background-color: blue;  
}
```

The height and width properties do not include padding, borders, or margins; they set the height/width of the area inside the padding, border, and margin of the element

11. **CSS Box Model:** All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout. 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:



Explanation of the different parts:

- I. **Content** - The content of the box, where text and images appear
- II. **Padding** - Clears an area around the content. The padding is transparent
- III. **Border** - A border that goes around the padding and content
- IV. **Margin** - Clears an area outside the border. The margin is transparent

The box model allows us to add a border around elements, and to define space between elements.

**Disclaimer:**

- The information given here is open source information to help the students in learning & building skills.
- Knowledge is collected from the best available resources on the Internet and can still have mistaken all errors must be reported to the distributor.