Unit 5: Cascading Style Sheet (CSS)

5.1. Introduction to Cascading Style Sheets (CSS) and advantages of using CSS

- A style sheet is a syntactic mechanism for specifying style information. In other words, a style sheet language is a computer language that expresses the presentation of structured documents.
- The best example of style sheet is Cascading Style Sheet (CSS). Word processors and desktop publishing systems have long used style sheets to impose a particular style on documents.
- Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in markup language.
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media.
- XHTML/HTML style sheets are called Cascading Style Sheets (CSS) because they can be defined at three different levels (inline level, internal or document level & external level) to specify the style of a document.

Advantages of CSS

- Consistency and Global Styling
- Easier to Maintain and Efficient Updates
- Time-Saving and Faster Loading Times
- Enhanced Design Options and Responsive Design
- SEO-Friendly
- Accessibility and Print-Friendly Pages
- Platform Independence and Device Compatibility
- CSS Minification and Page Load Efficiency
- Cascade, Inheritance, and Class/ID Usage
- CSS Sprites, Animations, and Effects

Disadvantages of CSS

- Cross-Browser Issues and Browser Compatibility
- Learning Curve
- Security Concerns
- Complex Layouts and Limited Layout Control
- File Size and Performance Impact
- Over-Specificity and Overriding Styles
- Maintenance Challenges

5.2. Basic Syntax

Syntax:

- A CSS rule has two min parts: a **selector** and one or more **declarations**.
- Selector is normally the HTML element we want to style and each declaration consists of a property and value.
- The property is the style attribute we want to use and each property has a value associated with it.
- A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.
- CSS selectors are used to “find” (or select) HTML elements based on their element name, id, class, attributes and more.

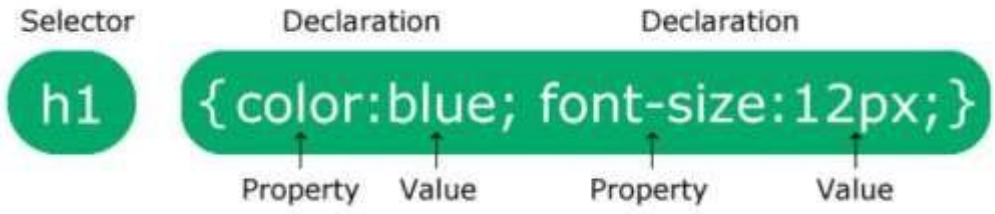


Figure 1: CSS Style Specification Format

CSS Syntax:

1. **Inline CSS**
   - ❖ Syntax: <h1 style="color: red; text-align: center;">Inline CSS</h1>
2. **Internal CSS**
   - ❖ Syntax: <head>
     <style> body{ background-color: wheat;} h1{ color: red; text-align: center;}
     </style>
     </head>
3. **External CSS**
   - ❖ Syntax:
     <head>
     <link rel="stylesheet" type="text/css" href="externalCSS.css"> </head>

5.2.1. Creating Cascading Style Sheets (CSS) using STYLE tag

HTML <style> Tag:

- ✞ The <style> tag is used to define style information (CSS) for a document.
- ✞ Inside the <style> element we specify how HTML elements should render in a browser.
- ✞ The <style> element must be included inside the <head> section of the HTML document.
- ✞ Use of the <style> element to apply a simple style sheet to an HTML document:
- ✞ The <style> tag comes in pairs. The content is written between the opening (<style>) and closing (</style>) tags.
- ✞ Example of the HTML <style> tag:

```
<!DOCTYPE html>
<html>
<head>
  <title>Style Tag</title>
  <style>
    body{
    background-color: bisque;
    }
   h1{
     color: red;
      text-align: center;
```

```
        }
        p{
      color:darkblue;
        text-align: justify;
      }
    </style>
</head>
<body>
   <h1>Style Tag</h1>
<p>The style tag is used to define style information (CSS) for a document.
</p>
</body>
</html>
```

5.3. Types of Style Sheets

- Cascading Style Sheet (CSS) is used to set the style in web pages that contain HTML elements.
- It sets the background color, font-size, font-family, color, …. etc.
- **There are three types of CSS / There are three levels of CSS** which are given below:
  1) Inline CSS
  2) Internal or Embedded CSS 3) External CSS

1. Inline CSS
   - Inline CSS contains the CSS property in the body section attached to the element is known as inline CSS. This kind of style is specified within an HTML tag using the style attribute.
   - Inline style sheets apply to the content of a single XHTML/HTML element.
   - Inline style specifications appear within the opening tag and apply only to the content of that tag.
   - <style> attribute is used to style a particular HTML tag.
   - Inline CSS is mostly used if we don't have access to external CSS files or need to apply style for a single element only
   - An example of HTML page with inline CSS would be as follows

```
<!DOCTYPE html>
<html>
<head>
   <title>Inline CSS</title>
</head>
<body style="background-color:wheat;">
   <h1 style="color: red; text-align: center;">Inline Cascading Style Sheets (CSS)</h1>
   <p style="color: yellowgreen; text-align: justify;">This is a paragraph.</p>
</body>
</html>
```

2. Internal or Embedded CSS
   - This can be used when a single HTML document must be styled uniquely ✞ Internal or document-level CSS apply to the whole body of a single document.

- Internal CSS code is put in between <style></style> tags inside the <head> section of a particular page.
- The classes and IDs can be used to refer to the CSS code, but they are only active on that particular page.
- Internal CSS is useful for sending someone a page template – as everything is in one page, it is easier to see a preview.
- An example of internal stylesheet:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <title>Internal CSS</title>
   <style>
      body{
         background-color: wheat;
      }
      h1{
         color: red;
         text-align: center;
      }
      p{
         color: blue;
         text-align: justify;
      }
   </style>
</head>
<body>
   <h1>Internal Cascading Style Sheets (CSS)</h1>
   <p>This is a paragraph. </p>
</body>
</html>
```

3. External CSS

- External CSS contains separate CSS files that contain only style properties with the help of tag attributes (For example class, id, heading, … etc.). CSS property is written in a separate file with a .css extension and should be linked to the HTML document using a link tag.
- Any changes made to an external CSS file will be reflected on our website globally.
- A reference to an external CSS file is put in the <head> section of the page
- External style sheets are written as text files with the MIME type text/css
- External style sheets can be stored on any computer on the Web. The browser fetches external style sheets just as it fetches documents.
- The <link> tag is used to specify external style sheets. Within <link>, the rel attribute is used to specify the relationship of the linked-to document to the document in which the link appears.
- The href attribute of <link> is used to specify the URL of the style sheet document, as in the following example:

externalCss.css

```
body {
      background-color: wheat;
  }
   h1 {
       color: red;
      text-align: center;
  }
    p {
       color: blue;
      text-align: justify;
  }
```

externalCSS.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <title>Internal CSS</title>
   <link rel="stylesheet" type="text/css" href="externalCSS.css">
</head>
<body>
   <h1>Internal Cascading Style Sheets (CSS)</h1>
   <p>This is a paragraph. </p>
</body>
</html>
```

5.4.1. Backgrounds and Color Styles and Attributes

CSS background:

- CSS background property is used to define the background effects on element. There are 5 CSS background properties that affects the HTML elements:
  1. background-color
  2. background-image
  3. background-repeat
  4. background-attachment
  5. background-position

1. CSS background-color

- The background-color property specifies the background color of an element.

- Syntax:

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

2. CSS background-image

- The background-image property specifies an image to use as the background of an element.
- syntax:

```
body {background-image: url("paper.gif");}
```

3. CSS background-repeat

- By default, the background-image property repeats the background image horizontally and vertically. Some images are repeated only horizontally or vertically.
- The background looks better if the image repeated horizontally only.
- Syntax:

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

- Showing the background image only once is also specified by the background-repeat property:
- Syntax:

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

4. CSS background-position

- The background-position property is used to specify the position of the background image.
- syntax:

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

5. CSS background-attachment

- The background-attachment property specifies whether the background image should scroll or be fixed (will not scroll with the rest of the page):
- syntax:

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

CSS Colors:

- The color property in CSS is used to set the color of HTML elements. Typically, this property is used to set the background color or the font color of an element.
- We can define the color of an element by using the following ways:
  - RGB format.
  - RGBA format.
  - Hexadecimal notation.
  - HSL.
  - HSLA.

- Built-in color.

Colors in CSS can be specified by the following methods:

```
Selector { color: red; }                    /* color keyword   */
Selector { color: rgb(0 255 0); }           /* RGB range 0-255 */
Selector { color: rgb(0% 100% 0%); }  /* RGB range 0%-100% */
Selector { color: #ff6347; }          /* hexadecimal code (#rrggbb) All values must be
                                                between 00 and FF. */
```

1. **CSS Background Color:**
   ✞ We can set the background color for HTML elements: ✞ Example:

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>
```

2. **CSS Text Color:**
   ✞ We can set the color of text: ✞
   Example:

```
<h1 style="color:Tomato;">Hello World</h1>
```

3. **CSS Border Color**
   ✞ We can set the color of borders: ✞
   Example:

```
<h1 style="border:2px solid Tomato;">Hello World</h1>
```

5.4.2. Fonts and Text Styles and Attributes

CSS Fonts:

- ✞ CSS Font property is used to control the look of texts. By the use of CSS font property, We can change the text size, color, style and more.
- ✞ These are some important font attributes:
  1. CSS Font family: This property is used to change the face of the font.
  2. CSS Font size: This property is used to increase or decrease the size of the font.
  3. CSS Font style: This property is used to make the font bold, italic or oblique.
  4. CSS Font variant: This property creates a small-caps effect.
  5. CSS Font weight: This property is used to increase or decrease the boldness and lightness of the font.

1. CSS font-family

- In CSS, we use the font-family property to specify the font of a text.
- Example:

```
.p1 {font-family: "Times New Roman", Times, serif;}
```

```
.p2 {font-family: Arial, Helvetica, sans-serif;}
```

2. CSS Font Size

- The font-size property sets the size of the text.
- Example:

```
h1 {font-size: 40px;}
```

3. CSS Font Style

- The font-style property is mostly used to specify italic text.
- This property has three values:
  - normal - The text is shown normally
  - italic - The text is shown in italics
  - oblique - The text is "leaning" (oblique is very similar to italic, but less supported)
- Example:

```
p.normal {font-style: normal;}
p.italic {font-style: italic;}
p.oblique {font-style: oblique;}
```

4. Font Variant

- The font-variant property specifies whether or not a text should be displayed in a smallcaps font.
- Example:

```
p.normal {font-variant: normal;}
p.small {font-variant: small-caps;}
```

5. Font Weight

- The font-weight property specifies the weight of a font:
- Example:

```
p.normal {font-weight: normal;}
p.thick {font-weight: bold;}
```

CSS Text:

- ❖ CSS provides a number of properties to format text in beautiful way.
- ❖ Following properties provided by CSS can be used to format the text:

1. Text Color

- The color property is used to set the color of the text. The color is specified by:
  - ○ a color name - like "red" ○ a HEX value - like "#ff0000" ○ an RGB value - like "rgb(255,0,0)"
- Example: h1 {color: green;}

2. CSS Text Alignment

- The text-align property is used to set the horizontal alignment of a text.
- Example: h1 {text-align: center;}

Text Align Last

- The text-align-last property specifies how to align the last line of a text.
- Example: p.a {text-align-last: right;} **Text Direction**

- The direction and unicode-bidi properties can be used to change the text direction of an element.
- Example: p {direction: rtl; unicode-bidi: bidi-override;}

Vertical Alignment

- The vertical-align property sets the vertical alignment of an element.
- Example: Set the vertical alignment of an image in a text: img.a {vertical-align: baseline;} img.b {vertical-align: text-top;} img.c {vertical-align: sub;}
  img.d {vertical-align: super;}

3. CSS - Text Decoration

- The CSS text-decoration property helps in adding extra decoration to the text, such as, adding a line (underline, strikethrough, overline) and color, style and thickness to the line.
- It is a shorthand property to the following properties:
  - text-decoration-line: Sets the type of decoration line (underline, strikethrough or overline).
  - text-decoration-color: Sets the color to the text decoration.
  - text-decoration-style: Sets the style to the text decoration (dotted, dashed, solid, wavy, double, etc.)
  - text-decoration-thickness: Sets the thickness to the text decoration.
- syntax: text-decoration: overline|line-through|underline|non

4. CSS Text Transformation

- The text-transform property is used to specify uppercase and lowercase letters in a text.
- It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word:
- Example:

```
p.uppercase {text-transform: uppercase;}
p.lowercase { text-transform: lowercase;}
p.capitalize {text-transform: capitalize;}
```

5. CSS Text Spacing Text Indentation

- The text-indent property is used to specify the indentation of the first line of a text:
- Example

```
p {  text-indent: 50px;}
```

Letter Spacing

- The letter-spacing property is used to specify the space between the characters in a text.
- Example: h1 {letter-spacing: 5px;}

Word Spacing

- The word-spacing property is used to specify the space between the words in a text.
- Example: p.one { word-spacing: 10px;}

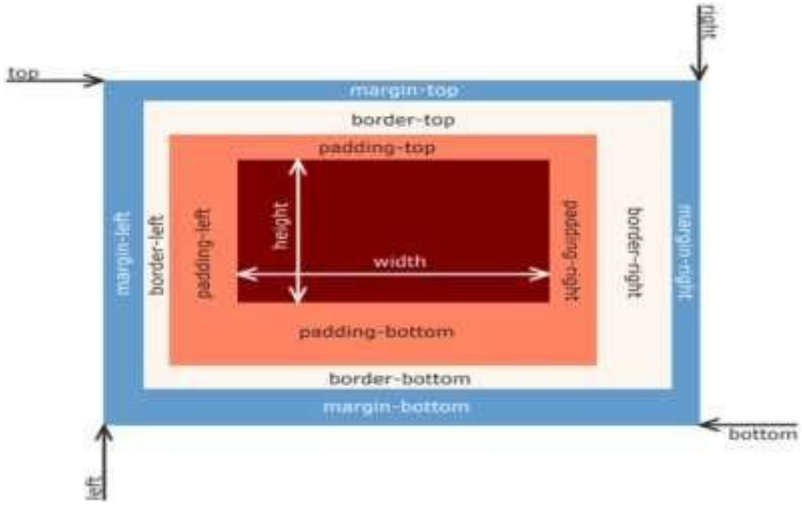
White Space

- The white-space property specifies how white-space inside an element is handled.
- Example: p {white-space: nowrap;}

5.4.3. Margin, Padding and Border Styles and Attributes

The Box Model:

- All HTML elements can be considered as boxes.
- The CSS box model is essentially a box that wraps around every HTML element.
- It consists of margins, borders, paddings and the actual content.
- It can be used as a toolkit for customizing the layout of different elements. The web browser renders every element as a rectangular box according to the CSS box model.
- Virtually all document elements can have borders with various styles, such as color and width.
- The amount of space between the content of an element and its border, is known as padding, and the space between the border and an adjacent element, is known as the margin.
- When there is no border, the margin plus the padding is the space between the content of an element and its neighbors.
- Although margin and padding seem similar when no border, but the background extends into the padding, but not into the margin.
- Padding creates extra space within an element, while margin creates extra space around an element.
- This model is shown below:



1. Borders

- The CSS border properties allow us 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:
  - dotted - Defines a dotted border
  - dashed - Defines a dashed border
  - solid - Defines a solid border
  - double - Defines a double border

- groove - Defines a 3D grooved border. The effect depends on the border-color value
- ridge - Defines a 3D ridged border. The effect depends on the border-color value
- inset - Defines a 3D inset border. The effect depends on the border-color value
- outset - Defines a 3D outset border. The effect depends on the border-color value
- none - Defines no border
- hidden - Defines a hidden border

- The border-style property can have from one to four values (for the top border, right border, bottom border, and the left border).
- The styles of one particular side of an element can also be set with border-top- style, border-bottom-style, border-left-style, and borderright-style.
- CSS Syntax border-style: none|dotted|dashed|solid|double|; Example:

```
<!DOCTYPE html>
<html>
<head>  <style>
    p.dotted {
        border-style: dotted;
    }

    p.dashed {
        border-style: dashed;
    }

    p.solid {
        border-style: solid;
    }

    p.double {
        border-style: double;
    }

    p.groove {
        border-style: groove;
    }

    p.ridge {
        border-style: ridge;
    }

    p.inset {
        border-style: inset;
    }

    p.outset {
        border-style: outset;
    }
```

```
        p.none {
            border-style: none;
        }

        p.hidden {
            border-style: hidden;
        }

        p.mix {
            border-style: dotted dashed solid double;
        }
    </style>
</head>
<body>
    <h2>The border-style Property</h2>
    <p>This property specifies what kind of border to display:</p>
    <p class="dotted">A dotted border.</p>
    <p class="dashed">A dashed border.</p>
    <p class="solid">A solid border.</p>
    <p class="double">A double border.</p>
    <p class="groove">A groove border.</p>
    <p class="ridge">A ridge border.</p>
    <p class="inset">An inset border.</p>
    <p class="outset">An outset border.</p>
    <p class="none">No border.</p>
    <p class="hidden">A hidden border.</p>
    <p class="mix">A mixed border.</p>
</body>
</html>
```

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 does not work if it is used alone. Always specify the "border-style" property to set the borders first.
- CSS Syntax: border-width: medium|thin|thick|length|; ❍ Example:

```
<!DOCTYPE html>
<html>
<head>  <style>
        p.one {
            border-style: solid;
            border-width: 5px;
        }
        p.two {
            border-style: solid;
            border-width: medium;
        }
        p.three {
```

```
            border-style: dotted;
            border-width: 2px;
        }
        p.four {
            border-style: dotted;
            border-width: thick;
        }
        p.five {
            border-style: double;
            border-width: 15px;
        }
        p.six {
            border-style: double;
            border-width: thick;
        }
    </style>
</head>

<body>

    <h2>The border-width Property</h2>
    <p>This property specifies the width of the four borders:</p>
    <p class="one">Some text.</p>
    <p class="two">Some text.</p>
    <p class="three">Some text.</p>
    <p class="four">Some text.</p>
    <p class="five">Some text.</p>
    <p class="six">Some text.</p>
</body>
</html>
```

c) Border Color

- The border-color property is used to set the color of the four borders.
- The color can be set by:
  - name - specify a color name, like "red"
  - HEX - specify a HEX value, like "#ff0000"
  - RGB - specify a RGB value, like "rgb(255,0,0)"
  - HSL - specify a HSL value, like "hsl(0, 100%, 50%)"
- The individual borders of an element can be colored differently through the properties: border-top-color, border-bottom-color, border-left-color, and border-right-color.
- Always border-style property must be declared before the border-color property. An element must have borders before we can change the color.
- CSS Syntax:

  border-color: color|transparent|;
- Example:

```
<!DOCTYPE html>
<html>
<head>    <style>
      p.one {
          border-style: solid;
          border-color: red;
      }
      p.two {
          border-style: solid;
          border-color: green;
      }
      p.three {
          border-style: dotted;
          border-color: blue;
      }
   </style>
</head>
<body>
   <h2>The border-color Property</h2>
   <p>This property specifies the color of the four borders:</p>
   <p class="one">A solid red border</p>
   <p class="two">A solid green border</p>
   <p class="three">A dotted blue border</p>
</body>
</html>
```

2. Margin

- The space between the border and an adjacent element, is known as the margin.
- Margins are used to create space around elements, outside of any defined borders.
- CSS has properties for specifying the margin for each side of an element:
  - margin-top
  - margin-right
  - margin-bottom
  - margin-left
- All the margin properties can have the following values:
  - auto - the browser calculates the margin • length - specifies a margin in px, pt, cm, etc.
  - % - specifies a margin in % of the width of the containing element
  - inherit - specifies that the margin should be inherited from the parent element ✞ Example:

```
<!DOCTYPE html>

<html>

<head>

    <style>
```

```
        div {
            border: 1px solid black;
            margin-top: 30px;
            margin-bottom: 30px;
            margin-right: 30px;
            margin-left: 30px;
            background-color: lightblue;
        }
    </style>
</head>
<body>
    <h2>Using individual margin properties</h2>
    <div>
     This div element has a top margin of 30px,
     a right margin of 30px, a bottom margin of 30px,
     and a left margin
     of 30px.
    </div>
</body>
</html>
```

3. Padding

- The amount of space between the content of an element and its border, is known as padding
- The CSS padding properties are used to generate space around an element's content, inside of any defined borders.
- CSS has properties for specifying the padding for each side of an element:
    - padding-top
    - padding-right
    - padding-bottom
    - padding-left
- All the padding properties can have the following values:
    - length - specifies a padding in px, pt, cm, etc.
    - % - specifies a padding in % of the width of the containing element
    - inherit - specifies that the padding should be inherited from the parent element ✞ Example:

```
<!DOCTYPE html>
```

```
<html>
<head>
    <style>
        div {
            border: 1px solid black;
            background-color: lightblue;
            padding-top: 50px;
            padding-right: 30px;
            padding-bottom: 50px;
            padding-left: 80px;
        }
    </style>
</head>
<body>
    <h2>Using individual padding properties</h2>
    <div>
        This div element has a top padding of 50px,
        a right padding of 30px, a bottom padding of 50px,
        and a left
        padding of 80px.
    </div>
</body>
</html>
```

5.4.4. List Styles and Table Layouts

CSS List Properties:

- The CSS list properties allow us to:
  - Set different list item markers for ordered lists
  - Set different list item markers for unordered lists
  - Set an image as the list item marker
  - Add background colors to lists and list items
- The list-style-type property is used to specify styles for both ordered and unordered lists.

- The list-style-type property of an unordered list can be set to disc, circle, square, or none. A disc is a small filled circle, a circle is an unfilled circle, and a square is a filled square. The default property value for bullets is disc
- Example:

```
<!DOCTYPE html>
<html>
<head>
    <style>
      ul.a {
         list-style-type: circle;
      }

      ul.b {
         list-style-type: square;
      }
    </style>
</head>
<body>
    <h2>The list-style-type Property</h2>
    <p>Example of unordered lists:</p>
    <ul class="a">
      <li>Coffee</li>
      <li>Tea</li>
      <li>Coca Cola</li>
    </ul>
    <ul class="b">
      <li>Coffee</li>
      <li>Tea</li>
      <li>Coca Cola</li>
    </ul>
</body>
</html>
```

- Any image can be used in a list item bullet. Such a bullet is specified with the list-style-image property, whose value is specified with the url form.
- The list-style-image property specifies an image as the list item marker.
- Syntax: ul { list-style-image: url('sqpurple.gif');} ✞ Example:

```
<!DOCTYPE html>
<html>
<head>
    <style>
      ul {
        list-style-image: url('sqpurple.gif');
      }
    </style>
</head>
<body>
    <h2>The list-style-image Property</h2>
```

```
    <p>The list-style-image property specifies an image as the list item    marker:</p>
    <ul>
      <li>Coffee</li>
      <li>Tea</li>
      <li>Coca Cola</li>
    </ul>
  </body>
  </html>
```

- The list-style-type property of an order list can be used to specify the types of sequence values.

Property value	*Sequence type*	*First four*
decimal	Arabic numerals	1, 2, 3, 4
upper-alpha	Uc letters	A, B, C, D
lower-alpha	Lc letters	a, b, c, d
upper-roman	Uc Roman	I, II, III, IV
lower-roman	***Lc Roman***	***i, ii, iii, iv***

- Example:

```
    <!DOCTYPE html>
  <html>
  <head>
    <style>
      ol.a {
        list-style-type: upper-roman;
      }
      ol.b {
        list-style-type: lower-alpha;
      }
    </style>
  </head>
  <body>
    <h2>The list-style-type Property</h2>
    <p>Example of ordered lists:</p>
    <ol class="a">
      <li>Coffee</li>
      <li>Tea</li>
      <li>Coca Cola</li>
    </ol>
    <ol class="b">
      <li>Coffee</li>
      <li>Tea</li>
      <li>Coca Cola</li>
    </ol>
  </body>
  </html>
```

Example: Ordered List

```
<!DOCTYPE html>
<html>
<head>
    <style>
        ol.list {
            list-style-type: upper-alpha;
        }
    </style>
</head>
<body>
    <p>Example of ordered lists:</p>
    <ol class="list">
        <li>Coffee</li>
        <li>Tea</li>
        <li>Coca Cola</li>
    </ol>
</body>
</html>
```

CSS Tables Properties:

- With CSS table properties, you can set how tables and table columns should be displayed.

1. Table border
   - To specify table borders in CSS, use the **border** property.
     Example:

```
table, th, td {
        border: 1px solid black;
}
```

   - The **border-collapse** property sets whether the table borders should be collapsed into a single border or not.

     Example:

```
table {
  border-collapse: collapse;
}
```

2. Table height & width
   - Width and height of a table are defined by the width and height properties.
     Example:

```
table {
  width: 100%;
}

th {
```

```
    height: 50px;
  }
```

3. Text align
   - For the alignment of text either horizontally or vertically.
   - The **text-align** property sets the horizontal alignment (like left, right, or center) of the content in <th> or <td>.
   - The **vertical-align** property sets the vertical alignment (like top, bottom, or middle) of the content in <th> or <td>.

   Examples:

```
th {
  text-align: left;
}
td {
  vertical-align: bottom;
}
```

4. Table padding and spacing
   - To control the space between the border and the content in a table, use the **padding** property on <td> and <th> elements
   - The **border-spacing** property sets the distance between the borders of adjacent cells.

   Example:

```
th, td {
  padding: 15px;
}
#table2 {
    border-spacing: 15px 50px;
}
```

   //Using two values (the first sets the horizontal spacing and the second sets the vertical spacing)
   //If single value it sets both the horizontal spacing and the vertical spacing

5. Table color

   Example:

```
th {
   background-color: #4CAF50;
   color: white;
}
```

6. Table caption side
   - Specify the placement of table captions

   Example:

```
#table1 {
caption-side: top;

}
```

```
#table2 {
      caption-side: bottom;

}
```

   ✞ Example :

   <!DOCTYPE html>

```
<html>
<head>
    <style>
        table,
        td,
        th {
            border: 1px solid #ddd;
            text-align: left;
        }
        table {
            border-collapse: collapse;
            width: 100%;
        }
        th,
        td {
            padding: 15px;
            height: 70px;
            text-align: center
        }
    </style>
</head>
<body>
    <table>
        <tr>
            <th>Firstname</th>
            <th>Lastname</th>
            <th>Savings</th>
        </tr>
        <tr>
            <td>Peter</td>
            <td>Griffin</td>
            <td>$100</td>
        </tr>
        <tr>
            <td>Lois</td>
            <td>Griffin</td>
            <td>$150</td>
        </tr>
        <tr>
            <td>Joe</td>
            <td>Swanson</td>
            <td>$300</td>
        </tr>
        <tr>
            <td>Cleveland</td>
            <td>Brown</td>
            <td>$250</td>
```

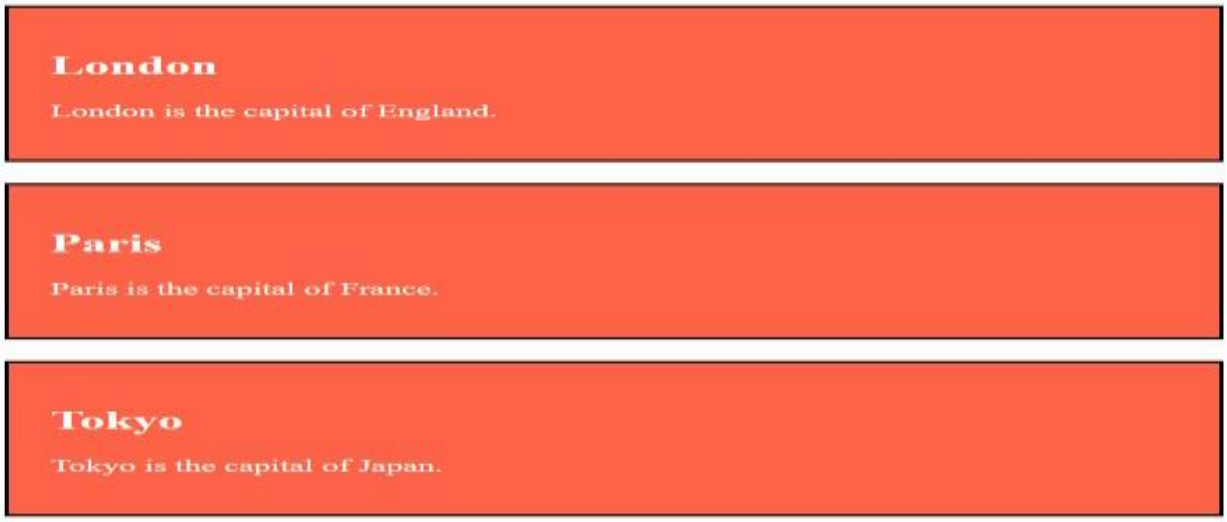
```
        </tr>
    </table>
</body>
</html>
```

HTML class Attribute:

- The HTML class attribute is used to specify a class for an HTML element.
- Multiple HTML elements can share the same class.
- The class attribute is often used to point to a class name in a style sheet. It can also be used by a JavaScript to access and manipulate elements with the specific class name.
- In the following example we have three elements with a class attribute with the value of "city". All of the three elements will be styled equally according to the .city style definition in the head section:

```
<!DOCTYPE html>
<html>
<head>
    <style>
        .city {
            background-color: tomato;
            color: white;
            border: 2px solid black;
            margin: 20px;
            padding: 20px;
        }
    </style>
</head>
<body>
    <div class="city">
        <h2>London</h2>
        <p>London is the capital of England.</p>
    </div>
    <div class="city">
        <h2>Paris</h2>
        <p>Paris is the capital of France.</p>
    </div>
    <div class="city">
        <h2>Tokyo</h2>
        <p>Tokyo is the capital of Japan.</p>
    </div>
</body>
</html>
```

Output :



HTML Tag

- The tag is an inline container used to mark up a part of a text, or a part of a document.
- The tag is easily styled by CSS or manipulated with JavaScript using the class or id attribute.
- It is used to group elements for styling purposes (by using the class or id attributes), A better way to use it when no other semantic element is available.
- The tag is much like the element, but is a block-level element and is an inline element
- Syntax:

Some Text

Example:

```
<!DOCTYPE html>
<html>
<head>
  <title>span tag</title>
  <!-- style for span tag  -->
  <style type=text/css>
    .span {
      color: green;
      text-decoration: underline;
      font-style: italic;
      font-weight: bold;
      font-size: 26px;
    }
  </style>
</head>
<body>
  <h2>
    Span Tag
  </h2>
```

```
    <span class="span">
       This is Span  tag
    </span><br />
    <span class="span">
       This is Span Tag
    </span>
  </body>
  </html>
```

HTML Div Tag

- The tag is known as the 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.
- The tag defines a division or a section in an HTML document.
- The tag is used as a container for HTML elements - which is then styled with CSS or manipulated with JavaScript.
- The tag is easily styled by using the class or id attribute.
- **Example:**

```
<!DOCTYPE html>
<html>
 <style>
    div.mycontainer {
       width: 100%;
       overflow: auto;
    }
    div.mycontainer div {
       width: 33%;
       float: left;
    }
  </style>
  <body>
    <div class="mycontainer">
      <div style="background-color:#FFF4A3;">
        <h2>London</h2>
        <p>London is the capital city of England.</p>
        <p>London has over 13 million inhabitants.</p>
             </div>

      <div style="background-color:#FFC0C7;">
        <h2>Oslo</h2>
        <p>Oslo is the capital city of Norway.</p>
        <p>Oslo has over 600.000 inhabitants.</p>
      </div>

      <div style="background-color:#D9EEE1;">
        <h2>Rome</h2>
        <p>Rome is the capital city of Italy.</p>
        <p>Rome has almost 3 million inhabitants.</p>
```

```
        </div>
    </div>
</body>
</html>
```

(END)