Unit-I

What is HTML?

- ✓ HTML stands for Hyper Text Markup Language
- ✓ HTML is the standard markup language for creating Web pages and Web applications
- ✓ HTML describes the structure of a Web page
- ✓ HTML was created by Berners-Lee in late 1991
- ✓ HTML is a Markup language rather than a programming language.

Some of the key advantages of learning HTML:

- ✓ **Create Web site -** You can create a website or customize an existing web template if you know HTML well.
- ✓ **Become a web designer** - If you want to start a career as a professional web designer, HTML and CSS designing is a must skill.
- ✓ **Understand web** - If you want to optimize your website, to boost its speed and performance, it is good to know HTML to yield best results.
- ✓ **Learn other languages** - Once you understand the basic of HTML then other related technologies like JavaScript, PHP are become easier to understand

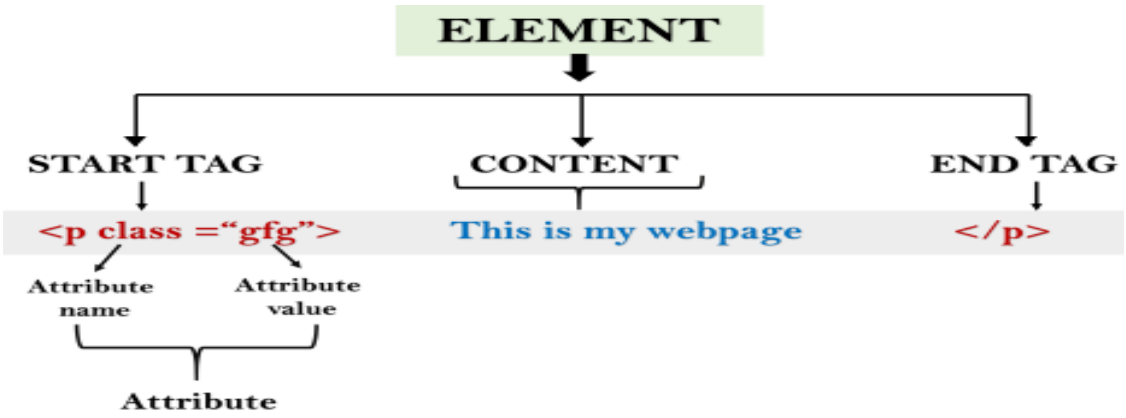
Origins and evolution of HTML

- ✓ HTML stands for "Hypertext Markup Language" and is the standard language used to produce web pages and applications.
- ✓ HTML was first presented by Tim Berners-Lee, the creator of the World Wide Web, in 1989. Berners-Lee originally created the Web to allow physicists to collaborate and share their work with each other
- ✓ **HTML1.0:** -HTML 1.0 was the first version of HTML, used from 1989 to 1994. It was a very limited version and included only 20 elements.
- ✓ **HTML2.0: -**Created in 1995, this version was a significant improvement to HTML 1.0. "HTML 2.0 becomes the first official set of standards for HTML
- ✓ **HTML3.0:** - where Dave Raggett introduced a fresh paper or draft on HTML. It included improved new features of HTML, giving more powerful characteristics for web masters in designing web pages. But these powerful features of the new HTML slowed down the browser in applying further improvements.

- ✓ **HTML4.01:** -This version of HTML, created in 1999, included cascading style sheets (CSS) which allowed aspects such as text, colour, font and backgrounds to be easily altered
- ✓ **HTML5:** - HTML 5.0, which is currently released and used worldwide. HTML 5 can be said for an extended version of HTML 4.01, which was published in the year 2012.

Basic Syntax

- ✓ The fundamental syntactic units of HTML are called **tags**. In general, tags are used to specify categories of content.
- ✓ For each category, a browser has default presentation specifications for the specified content.
- ✓ The syntax of a tag is the tag ‘s name surrounded by angle brackets (< and >). Tag names must be written in all lowercase letters. Most tags appear in pairs: an opening tag and a closing tag.
- ✓ The name of a closing tag is the name of its corresponding opening tag with a slash attached to the beginning. Whatever appears between a tag and its closing tag is the **content** of the tag.
- ✓ The opening tag and its closing tag together specify a container for the content they enclose. The container and its content together are called an **element**.
- ✓ For example, consider the following element: This is simple stuff.
- ✓ The paragraph tag, , marks the beginning of the content; the tag marks the end of the content of the paragraph element.



Attributes: -

- ✓ HTML attributes are special words which provide additional information about the elements or attributes are the modifier of the HTML element.
- ✓ Each element or tag can have attributes, which defines the behavior of that element.
- ✓ Attributes should always be applied with start tag
- ✓ Attributes usually come in name/value pairs like name="value"
- ✓ Attribute names, like tag names, are written in lowercase letters. Attribute values must be delimited by double quotes.

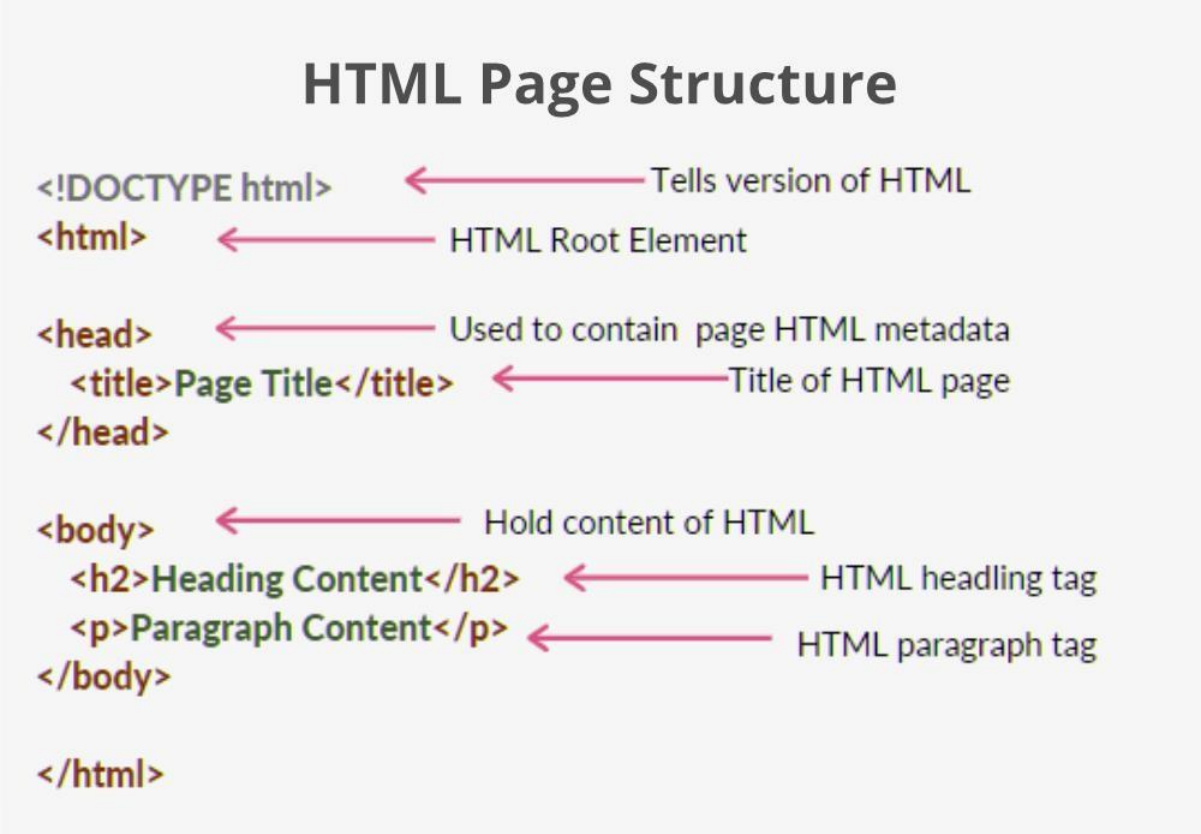
Syntax

```
<element attribute name="value">content</element>
```

Example

```
<!DOCTYPE html>
<html>
<body>
<h2>The href Attribute</h2>
<p>HTML links are defined with a tag. The link address is specified in the href attribute:</p>
<a href="https://jsscwchn.com">JSS College </a>
</body>
</html>
```

Standard HTML Document Structure



<!DOCTYPE...>	This tag defines the document type and HTML version.
<html>	This tag encloses the complete HTML document and mainly comprises of document header which is represented by <head>...</head> and document body which is represented by <body>...</body> tags.
<head>	This tag represents the document's header which can keep other HTML tags like <title>, <link> etc.
<title>	The <title> tag is used inside the <head> tag to mention the document title.
<body>	This tag represents the document's body which keeps other HTML tags like <h1>, , <p> etc
<h2>	This tag represents the heading.

<p>	This tag represents a paragraph.

Basic text markup

- ✓ Paragraphs
- ✓ Line Breaks
- ✓ Preserving white space
- ✓ Headings
- ✓ Block Quotes
- ✓ Font styles & size
- ✓ Character Entities
- ✓ Horizontal Rules
- ✓ Meta Element

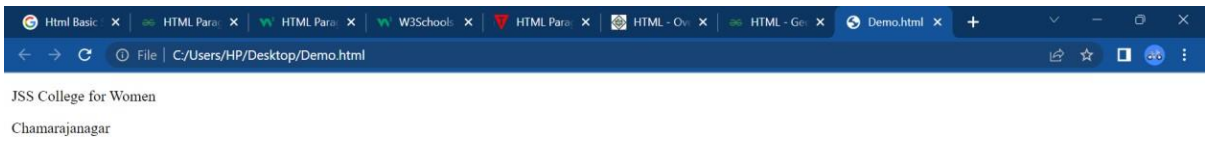
Paragraphs

- ✓ The HTML <p> element defines a paragraph.
- ✓ A paragraph always starts on a new line, and browsers automatically add some white space (a margin) before and after a paragraph

Example

```
<!DOCTYPE html>
<html>
<body>
<p>JSS College for Women </p>
<p>Chamarajanagar</p>
</body>
</html>
```

Output



Line Breaks

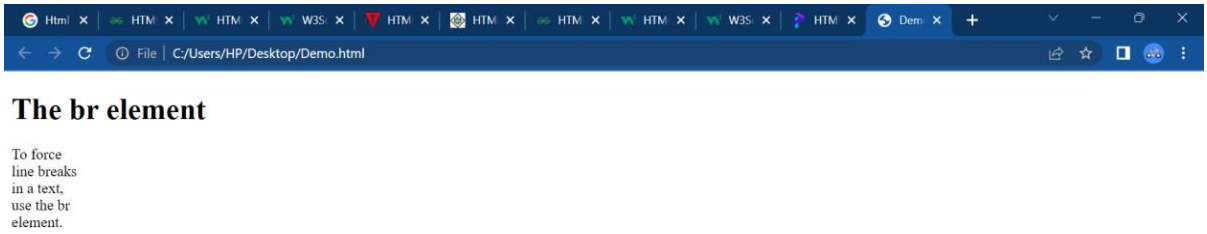
- ✓ The HTML Line Break tag
 is used to apply a line break and start your text in the new line

- ✓ The <br> tag is an empty tag which means that it has no end tag.

Example

```
<!DOCTYPE html>
<html>
<body>
<h1>The br element</h1>
<p>To force<br> line breaks<br> in a text,<br> use the br<br> element.</p>
</body>
</html>
```

Output



Preserving white space

```
<p><pre>
Mary
    had a
        little
            lamb
</pre>
```

This markup would be displayed as shown in Figure 2.5. Notice that the content of the `pre` element is shown in monospace, rather than in the default font.

```
Mary
    had a
        little
            lamb
```

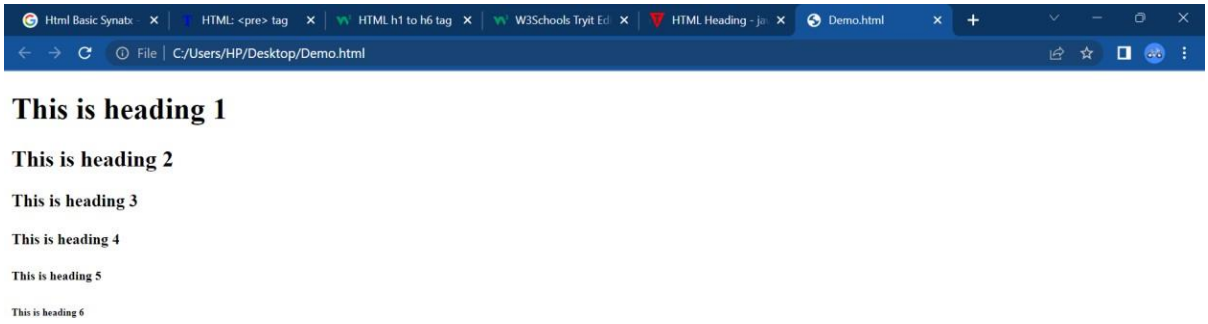
- ✓ The HTML **<pre> tag** defines preformatted text preserving both whitespace and line breaks in the HTML document.
- ✓ This tag is also commonly referred to as the <pre> element.

Example

Headings

- ✓ A HTML heading or HTML <h> tag can be defined as a title or a subtitle which you want to display on the webpage.
- ✓ There are six different HTML headings which are defined with the <h1> to <h6> tags, from highest level h1 (main heading) to the least level h6 (least important heading).
- ✓ h1 is the largest heading tag and h6 is the smallest one. So h1 is used for most important heading and h6 is used for least important

Output



Block Quotes

- ✓ HTML <blockquote> tag is used to define a block of text which is quoted from another source.
- ✓ The Browser usually displays the content within <blockquote> tag as indented text.

Example


```
<!DOCTYPE html>
<html>
<body>
<h1>The blockquote element</h1>
<p>JSS College for Women Chamarajanagar</p>
<blockquote cite="http://jsscwchn.com">
JSS College of Arts, Science and Commerce was established in 1967 with B.Sc programme. It was a co-education institution till it was converted into Women's College in the year 2000. </blockquote>
</body>
</html>
```

Font styles & size

Element name	Description
<b>	This is a physical tag, which is used to bold the text written between it.
	This is a logical tag, which tells the browser that the text is important.
<i>	This is a physical tag which is used to make text italic.
	This is a logical tag which is used to display content in italic.
<mark>	This tag is used to highlight text.
<u>	This tag is used to underline text written between it.
<strike>	This tag is used to draw a strikethrough on a section of text. (Not supported in HTML5)
<sup>	It displays the content slightly above the normal line.
<sub>	It displays the content slightly below the normal line.
<del>	This tag is used to display the deleted content.
<ins>	This tag displays the content which is added
<big>	This tag is used to increase the font size by one conventional unit.
<small>	This tag is used to decrease the font size by one unit from base font size.

tags are categorized into two

- ✓ Block
- ✓ Inline

Block tags

- ✓ breaks the current line so that its content appears in a new line
- ✓ E.g.: heading and blockquote tag

Inline tags

- ✓ does not implicitly include a line break.
- ✓ The content of the inline tag appears on the current line
- ✓ Eg: and

Note: - Inline versus block elements – Block elements CANNOT be nested in inline element

Character Entities

HTML provides a collection of special characters that are sometimes needed in a document but cannot be typed as themselves. These special characters are defined as entities, which are

codes for the characters. An entity in a document is replaced by its associated character by the browser.

E.g, ©

Will display ©

Table lists some of the most used entities

Character	Entity	Meaning
&	&	Ampersand
<	<	Is less than
>	>	Is greater than
"	"	Double quote
'	'	Single quote (apostrophe)
¼	¼	One-quarter
½	½	One-half
¾	¾	Three-quarters
°	°	Degree
(space)		Nonbreaking space

Horizontal Rules

The parts of a document can be separated from each other, making the document easier to read, by placing horizontal lines between them. Such lines are called horizontal rules, and the lock tag that creates them is <hr />. The <hr /> tag causes a line break (ending the current line) and places a line across the screen.

Images

HTML img tag is used to display image on the web page. HTML img tag is an empty tag that contains attributes only, closing tags are not used in HTML image element.

Attributes of HTML img tag

The **src** and **alt** are important attributes of HTML img tag. All attributes of HTML image tag are given below: -

- ✓ **src: -**It is a necessary attribute that describes the source or path of the image. It instructs the browser where to look for the image on the server.
- ✓ **alt**: -The alt attribute defines an alternate text for the image if it can't be displayed. The value of the alt attribute describes the image in words.
- ✓ **Width:** which is used to specify the width to display the image.
- ✓ **Height:** which is used to specify the height to display the image

Example

```
<!DOCTYPE html>
<html>
<body>
<h2>HTML Image</h2>
<img src="D:/images/4.jpg" alt="it is alternative" width="460" height="345">
<body>
<Html>
```

Output

Image Formats:

1. GIF (Graphic Interchange Format) 8-bit color (256 different colors)
2. JPEG (Joint Photographic Experts Group) It has 24-bit color representation (16 million different colors)
3. Portable Network Graphics (PNG)

Note: -Both use compression, but JPEG compression is better.

Lists

HTML Lists are used to specify lists of information. All lists may contain one or more list elements. There are three different types of HTML lists:

1. Ordered List or Numbered List (ol)
2. Unordered List or Bulleted List (ul)
3. Description List or Definition List (dl)

Ordered List or Numbered List (ol)

HTML Ordered List or Numbered List displays elements in numbered format. The HTML <ol> tag is used for ordered list.

There can be different types of numbered list:

- ✓ Numeric Number (1, 2, 3)
- ✓ Capital Roman Number (I II III)
- ✓ Small Roman Number (i ii iii)
- ✓ Capital Alphabet (A B C)
- ✓ Small Alphabet (a b c)

Example

```
<!DOCTYPE html>
<html>
<body>
<h2>An ordered HTML list</h2>
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
</body>
</html>
```

Output

The type Attribute for Order List

- ✓ You can use type attribute for <ol> tag to specify the type of numbering you like. By default, it is a number. Following are the possible options

Type	Description
Type "1"	This is the default type. In this type, the list items are numbered with numbers.
Type "I"	In this type, the list items are numbered with upper case roman numbers.
Type "i"	In this type, the list items are numbered with lower case roman numbers.
Type "A"	In this type, the list items are numbered with upper case letters.
Type "a"	In this type, the list items are numbered with lower case letters.

Example

```
<html>
   <head>
      <title>HTML Ordered List</title>
   </head>
   <body>
      <h1>List of Courses Offered By JSS College for women
Chamarajangar</h1>
      <ol type="I">
         <li >BA</li>
         <li>B.Com</li>
         <li>BBA</li>
         <li>B.Sc</li>
         <li>BCA</li>
      </ol>
   </body>
</html>
```

Output

List of Courses Offered By JSS College for women Chamarajangar

I. BA
II. B.Com
III. BBA
IV. B.Sc
V. BCA

The start Attribute

- ✓ You can use start attribute for <ol> tag to specify the starting point of numbering you need. Following are the possible options –

<ol type = "1" start = "4"> - Numerals starts with 4.
<ol type = "I" start = "4"> - Numerals starts with IV.
<ol type = "i" start = "4"> - Numerals starts with iv.
<ol type = "a" start = "4"> - Letters starts with d.
<ol type = "A" start = "4"> - Letters starts with D.

Example Following is an example where we used <ol type = "i" start = "4" >

```
<html>
   <head>
      <title>HTML Ordered List</title>
   </head>
   <body>
      <ol type = "i" start = "4">
         <li>Beetroot</li>
         <li>Ginger</li>
         <li>Potato</li>
         <li>Radish</li>
         </ol>
         </body>
         </html>
```

The Reversed Attribute: This is a Boolean attribute of HTML <ol> tag, and it is new in HTML5 version.

```
<html>
 <head>
  </head>
 <body>
    <ol reversed>
   <li>HTML</li>
    <li>Java</li>
    <li>JavaScript</li>
    <li>SQL</li>
  </ol>
 </body>
</html>
```


output

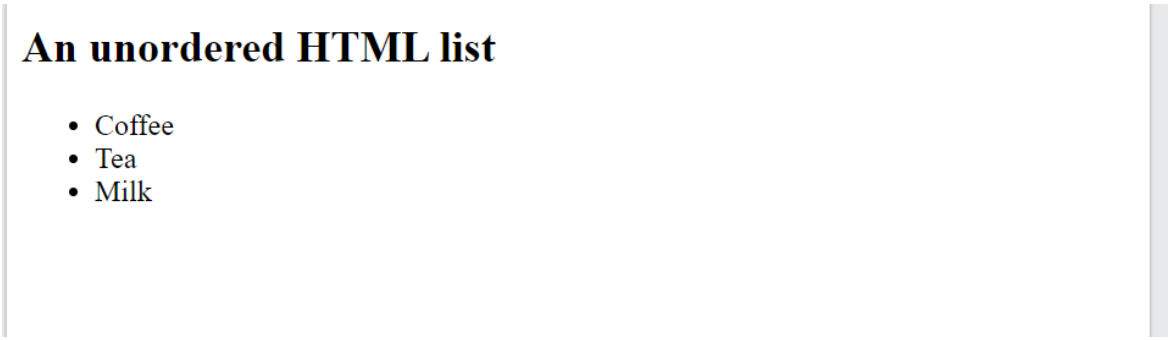
Unordered List or Bulleted List (ul)

- ✓ An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.
- ✓ The list items will be marked with bullets (small black circles) by default:

Example

```
<!DOCTYPE html>
<html>
<body>
<h2>An unordered HTML list</h2>
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
  </ul>
  </body>
  </html>
```

Output



The type Attribute

- ✓ You can use type attribute for <ul> tag to specify the type of bullet you like. By default, it is a disc. Following are the possible options –

Type	Description
Type "disc"	This is the default style. In this style, the list items are marked with bullets.
Type "circle"	In this style, the list items are marked with circles.
Type "square"	In this style, the list items are marked with squares.
Type "none"	In this style, the list items are not marked .

Example

```
<html>
   <head>
      <title>HTML Unordered List</title>
   </head>
   <body>
      <ul type = "circle">
         <li>Beetroot</li>
         <li>Ginger</li>
         <li>Potato</li>
         <li>Radish</li>
      </ul>
   </body>
</html>
```

Description List or Definition List (dl)

- ✓ HTML also supports description lists.
- ✓ A description list is a list of terms, with a description of each term.
- ✓ The <dl> tag defines the description list,the<dt> tag defines the term, and the <dd> tag describes each term(description details)

Example

```
<html>
<body>
<h2>A Description List</h2>
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
</body>
</html>
```

output

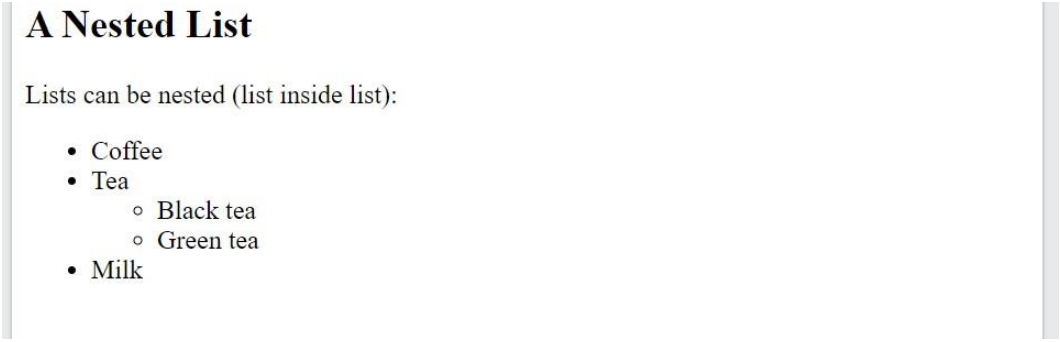
Nested List

- ✓ List can be nested within another List is Called Nested List

Example

```
<!DOCTYPE html>
<html>
<body>
<h2>A Nested List</h2>
<p>Lists can be nested (list inside list):</p>
<ul>
  <li>Coffee</li>
  <li>Tea
    <ul>
      <li>Black tea</li>
      <li>Green tea</li>
    </ul>
  </li>
  <li>Milk</li>
</ul>
</body>
</html>
```

output



A Nested List

Lists can be nested (list inside list):

- Coffee
- Tea
  - Black tea
  - Green tea
- Milk

Table Tag

- ✓ **HTML table tag** is used to display data in tabular form (row * column). There can be many columns in a row.
- ✓ The HTML tables are created using the **<table>** tag in which the **<tr>** tag is used to create table rows and **<td>** tag is used to create data cells. table header is defined by <th>
- ✓ The elements under <td> are regular and left aligned by default
- ✓ We can also mention the table caption with help of <caption> tag

Example

```
<html>
   <head>
      <title>HTML table Tag</title>
   </head>
   <body>
      <table border = "1">
         <tr>
            <th>Team</th>
            <th>Ranking</th>
         </tr>
         <tr>
            <td>India</td>
            <td>1</td>
         </tr>
         <tr>
            <td>South Africa</td>
            <td>2</td>
         </tr>
         <tr>
            <td>Australia</td>
            <td>3</td>
         </tr>
      </table>
      </body>
      </html>
```

Table its attributes

The HTML <table> tag also supports the following additional attributes

- ✓ align: - which is used to align the table to right, left, center and justify
- ✓ bgcolor: - You can set background colour for whole table or just for one cell.
- ✓ border: - Specifies the border width. A value of "0" means no border
- ✓ cellpadding: - Specifies the space between the cell borders and their contents
- ✓ cells pacing: - Specifies the space between cells.
- ✓ width: - Specifies the width of the table.
- ✓ height: - Specifies the height of the table

background: You can set background image for whole table or just for one cell

```
<html>
   <head>
      <title>HTML Table Cellpadding</title>
   </head>
   <body>
<table border = "1" cellpadding = "5" cellspacing = "5"
bgcolor="red" background="D:/images/1.jpg" align="center"
width="100%" height="400px">
 <tr>
            <th>Name</th>
            <th>Salary</th>
         </tr>
         <tr>
            <td>Mallesha R</td>
            <td>50000</td>
         </tr>
         <tr>
            <td>Vinay R U</td>
            <td>70000</td>
         </tr>
      </table>
   </body>
</html>
```

Colspan and Rowspan Attributes

- ✓ You will use colspan attribute if you want to merge two or more columns into a single column. Similar way you will use rowspan if you want to merge two or more rows

Example

```
<html>
   <head>
      <title>HTML Table Colspan/Rowspan</title>
   </head>

   <body>
      <table border = "1">
         <tr>
            <th>Column 1</th>
            <th>Column 2</th>
            <th>Column 3</th>
         </tr>
         <tr>
            <td rowspan = "2">Row 1 Cell 1</td>
            <td>Row 1 Cell 2</td>
            <td>Row 1 Cell 3</td>
         </tr>
         <tr>
            <td>Row 2 Cell 2</td>
            <td>Row 2 Cell 3</td>
         </tr>
         <tr>
            <td colspan = "3">Row 3 Cell 1</td>
         </tr>
      </table>
   </body>
</html>
```

Table Caption

- ✓ The caption tag will serve as a title or explanation for the table and it shows up at the top of the table. This tag is deprecated in newer version of HTML/XHTML.

Example

```
<html>

   <head>
      <title>HTML Table Caption</title>
   </head>

   <body>
      <table border = "1" width = "100%">
         <caption>This is the caption</caption>

         <tr>
            <td>row 1, column 1</td><td>row 1, columnn 2</td>
            </tr>

         <tr>
            <td>row 2, column 1</td><td>row 2, columnn 2</td>
            </tr>
            </table>
            </body>
            </html>
```

Nested Tables

- ✓ You can use one table inside another table. Not only tables you can use almost all the tags inside table data tag <td>.

Example

```
<html>
   <head>
      <title>HTML Table</title>
   </head>
   <body>
      <table border = "1" width = "100%">
         <tr>
            <td>
               <table border = "1" width = "100%">
                  <tr>
                     <th>Name</th>
                     <th>Contact/th>
                  </tr>
                  <tr>
                     <td>mallesha R</td>
                     <td>7349555800</td>
                  </tr>
                  <tr>
                     <td>Vinay R U</td>
                     <td>7777777777</td>
                  </tr>
               </table>
            </td>
         </tr>
      </table>
   </body>
</html>
```


HTML Form

- ✓ An HTML form is a section of a document which contains controls such as text fields, password fields, checkboxes, radio buttons, submit button, menus etc.
- ✓ An HTML form facilitates the user to enter data that is to be sent to the server for processing such as name, email address, password, phone number, etc.
- ✓ The HTML <form> tag is used to create an HTML form

Why we use Form

- ✓ HTML Forms are required when you want to collect some data from the site visitor. For example, during user registration you would like to collect information such as name, email address, credit card, etc.
- ✓ A form will take input from the site visitor and then will post it to a back-end application such as CGI, ASP Script, or PHP script etc. The back-end application will perform required processing on the passed data based on defined business logic inside the application.

Example

```
<form action = "Script URL" method = "GET|POST">
   form elements like input, textarea etc.
</form>
```

Form Attribute

- ✓ **Action: -**Backend script ready to process your passed data.
- ✓ **Method: -**Method to be used to upload data. The most frequently used are GET and POST methods.
- ✓ **Target: -**Specify the target window or frame where the result of the script will be displayed. It takes values like _blank, _self, _parent etc.
- ✓ **Name: -**You can mention the Name of the Form

HTML Form Controls

There are different types of form controls that you can use to collect data using HTML form

- ✓ Text Input Controls
- ✓ Checkboxes Controls
- ✓ Radio Box Controls
- ✓ Select Box Controls
- ✓ File Select boxes
- ✓ Submit and Reset Button

Text Input Controls

There are three types of text input used on forms −

- ✓ **Single-line text input controls** − This control is used for items that require only one line of user input. They are created using HTML <input> tag.
- ✓ **Password input controls** − This is also a single-line text input, but it masks the character as soon as a user enters it. They are also created using HTML <input> tag.
- ✓ **Multi-line text input controls** − This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML <textarea> tag.

Single-line text input controls

- ✓ This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML <input> tag.

Example

Here is a basic example of a single-line text input used to take first name and last name –

```
<html>
   <head>
      <title>Text Input Control</title>
   </head>

   <body>
      <form >
         First name: <input type = "text" name = "first_name" />
      </form>
   </body>
</html>
```

Attributes for Single-line text input controls

Following is the list of attributes for <input> tag for creating text field.

- ✓ **Type: -**Indicates the type of input control and for text input control it will be set to text.
- ✓ **Name: -**Used to give a name to the control which is sent to the server to be recognized and get the value.
- ✓ **Value: -** This can be used to provide an initial value inside the control.
- ✓ **Size: -** Allows to specify the width of the text-input control in terms of characters.
- ✓ **Maxlength: -**Allows to specify the maximum number of characters a user can enter the text box.

Password input controls

- ✓ This is also a single-line text input, but it masks the character as soon as a user enters it. They are also created using HTML <input>tag but type attribute is set to password.

Example

Here is a basic example of a single-line password input used to take user password

```
<html>
   <head>
      <title>Password Input Control</title>
   </head>
   <body>
      <form >
         User ID : <input type = "text" name = "user_id" />
         <br>
         Password: <input type = "password" name = "password" />
      </form>
      </body>
      </html>
```

Attributes for Password input controls

Following is the list of attributes for <input> tag for creating password field.

- ✓ **Type:** -Indicates the type of input control and for password input control it will be set to password.
- ✓ **Name:** -Used to give a name to the control which is sent to the server to be recognized and get the value.
- ✓ **Value:** -This can be used to provide an initial value inside the control.
- ✓ **Size:** -Allows to specify the width of the text-input control in terms of characters.
- ✓ **Maxlength:** -Allows to specify the maximum number of characters a user can enter the text box.

Multiple-Line Text Input Controls

- ✓ This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML <textarea> tag.

Example

Here is a basic example of a multi-line text input used to take item description –


```
<html>

   <head>
      <title>Multiple-Line Input Control</title>
   </head>
   <body>
      <form>
         Description: <br />
         <textarea rows = "5" cols = "50" name = "description">
            Enter description here...
         </textarea>
      </form>
   </body>
```

Attributes

Following is the list of attributes for <textarea> tag.

- ✓ **Name:** -Used to give a name to the control which is sent to the server to be recognized and get the value.
- ✓ **Rows:** -Indicates the number of rows of text area box.
- ✓ **Cols:** -Indicates the number of columns of text area box

Checkbox Control

- ✓ Checkboxes are used when more than one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to checkbox

Example

```
<html>

   <head>
   </head>
   <body>
      <form>
     <input type = "checkbox" name = "maths" value = "on"> Maths
  <input type = "checkbox" name = "physics" value = "on">
Physics
      </form>
   </body>
</html>
```

Attributes

Following is the list of attributes for <checkbox> tag.

- ✓ **Type: -**Indicates the type of input control and for checkbox input control it will be set to checkbox.
- ✓ **Name**: -Used to give a name to the control which is sent to the server to be recognized and get the value.
- ✓ **Value:** -The value that will be used if the checkbox is selected.
- ✓ **Checked:** -Set to checked if you want to select it by default.

Radio Button Control

- ✓ Radio buttons are used when out of many options, just one option is required to be selected.
- ✓ They are also created using HTML <input> tag but type attribute is set to radio.

Example

```
<html>
   <head>
   </head>
   <body>
      <form>
  <input type = "radio" name = "subject" value = "maths"> Maths
<input type = "radio" name = "subject" value = "physics">
Physics
      </form>
   </body>
</html>
```

Attributes

Following is the list of attributes for radio button.

- ✓ **Type:** -Indicates the type of input control and for checkbox input control it will be set to radio.
- ✓ **Name:** -Used to give a name to the control which is sent to the server to be recognized and get the value.
- ✓ **Value:** -The value that will be used if the radio box is selected.

Select Box Control

- ✓ A select box, also called drop down box which provides option to list down various options in the form of drop-down list, from where a user can select one or more options.

Example

```
<html>
   <head>
      <title>Select Box Control</title>
   </head>
   <body>
      <form>
         <select name = "dropdown">
            <option value = "Maths" selected>Maths</option>
            <option value = "Physics">Physics</option>
         </select>
      </form>
   </body>
</html>
```


Attributes

Following is the list of important attributes of <select> tag

- ✓ Name: -Defines a name for the drop-down list
- ✓ Size: -Defines the number of visible options in a drop-down list
- ✓ Multiple: -Specifies that multiple options can be selected at once.

Following is the list of important attributes of <option> tag

- ✓ Value: -The value that will be used if an option in the select box is selected.
- ✓ Selected: -Specifies that this option should be the initially selected value when the page loads.
- ✓ label: -An alternative way of labelling options

File Upload Box

- ✓ If you want to allow a user to upload a file to your web site, you will need to use a file upload box, also known as a file select box. This is also created using the <input> element but type attribute is set to file.

Example

Here is example HTML code for a form with one file upload box

```
<html>

   <head>
      <title>File Upload Box</title>
   </head>

   <body>
      <form>
         <input type = "file" name = "fileupload" accept =
"image/*" />
      </form>
   </body>

</html>
```

Attributes

Following is the list of important attributes of file upload box –

- ✓ Name: -Used to give a name to the control which is sent to the server to be recognized and get the value.
- ✓ Accept: -Specifies the types of files that the server accepts.

Button Controls

- ✓ There are various ways in HTML to create clickable buttons. You can also create a clickable button using <input>tag by setting its type attribute to button.

The type attribute can take the following values

- ✓ **Submit:** -This creates a button that automatically submits a form.
- ✓ **Reset:** -This creates a button that automatically resets form controls to their initial values.
- ✓ **Button:** -This creates a button that is used to trigger a client-side script when the user clicks that button.
- ✓ **image**: -This creates a clickable button but we can use an image as background of the button.

Example

Here is example HTML code for a form with three types of buttons –

```
<html>
   <head>
      <title>File Upload Box</title>
   </head>
   <body>
      <form>
         <input type = "submit" name = "submit" value = "Submit"/>
         <input type = "reset" name = "reset" value = "Reset" />
         <input type = "button" name = "ok" value = "OK" />
         <input type = "image" name = "imagebutton" src =
"/html/images/logo.png" />
      </form>
   </body>
</html>
```

Frame Tag

- ✓ The browser display window can be used to display more than one document at a time the window can be divided into rectangular areas each of which is a **frame.**
- ✓ Each frame is cable of displaying its own document.

Framesets

- ✓ The number of frames and their layout in the browser window are specified with the <frameset> tag.
- ✓ A frameset element takes the place of the body element in a document. A document has either a body or a frameset but cannot have both. The <frameset> tag must have either a rows or cols attribute and often they have both.
- ✓ The rows attribute specifies the number of rows of frames that will occupy the window. There are three kinds of values for rows: numbers, percentage and asterisks. Two or more values separated by commons are given in a quoted string when a number is used as a value it specifies the height of one row in pixels.
- ✓ A percentage is given as number followed immediately by a percent sign. When percentage value specifies the percentage of the total browser window height that a row should occupy. when an asterisk is used as the value of rows it means the remainder of the window height

Example
<frameset rows=”200,300,400”>
<frameset rows=”22%,33%,45%>
<frameset rows=”22%,33%,*>

- ✓ The cols attribute is very much like the rows attribute except that is specifies the number of Columns of frames

Example<frameset rows=" 33%,33%,33%" cols="25%,*">

Content of Frame1	Content of Frame2
Content of Frame 3	Content of Frame4
Content of Frame5	Content of Frame6

Frames

The Content of a frame is specified with the <frame> tag which can appear only in the content of a frameset element Each frame defined in a frameset has an associated <frame> tag that gives the filename of a document that supplies its content. The sequence of <frame> tags in a frameset is important because the order indicates which frame gets which content. The content of a frame is specified as the value of the src attribute in the <frame> tag. For example

<frame src=" jss.html">

If a <frame> tag has no src attribute the browser displays an empty frame. if the content of a frame does not fit into then given frame, scroll bars are implicitly included. If u want to scroll bar regardless of the size of its content the <frame> attribute scrolling can be set to yes. If a <frame> tag includes a name attribute the content of its associated frame can be changed by the selection of link in some other frame that specifies that name.

Example

```
<html>
<head>
<title> First Frame</title>
</head>
<body>
<h1> Departments </h1>
<a href="cs.html" target="_content">Comupter Science</a><br />
<a href="math.html" target="_content">Mathematics</a><br />
<a href="kannada.html"   target="_content">Kannada</a><br />
<a href="commerce.html"   target="_content">Commerce</a><br />
</body>
</html>
```

Example

```
<html>
  <head>
    <title>Frames</title>
  </head>
  <frameset cols = "20%,80%">
    <frame name = "left" src = "Frame_First_Frame.html" ></frame>
    <frame name = "_content" src = "content.html" ></frame>
    <noframes>
      <body>Your browser does not support frames.</body>
    </noframes>
  </frameset>
</html>
```

```
<html >
<head>
   <title>Untitled Page</title>
</head>
<body>
<h1>Content page</h1>
</body>
</html>
```


Note: - frameset can be nested For example outermost frameset can define some number of columns. each of these columns then can be divided with a nested frameset into whatever collection of frames is useful.

Example

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title></title>
</head>
<frameset cols="40%,*">
  <frameset rows="50%,*">
    <frame src="frame1.html"></frame>
    <frame src="frame2.html"></frame>
  </frameset>
```

Overview and features of HTML5

HTML5 is a next version of HTML. Here, you will get some brand-new features which will make HTML much easier. These new introducing features make your website layout clearer to both website designers and users.

New Features

HTML5 introduces a number of new elements and attributes that can help you in building modern websites. Here is a set of some of the most prominent features introduced in HTML5.

- ✓ **New Semantic Elements** − These are like <header>, <footer>, and <section>.
- ✓ **Forms 2.0** − Improvements to HTML web forms where new attributes have been introduced for <input> tag.
- ✓ **Persistent Local Storage** − To achieve without resorting to third-party plugins.
- ✓ **WebSocket** − A next-generation bidirectional communication technology for web applications.
- ✓ **Server-Sent Events** − HTML5 introduces events which flow from web server to the web browsers and they are called Server-Sent Events (SSE).
- ✓ **Canvas** − This supports a two-dimensional drawing surface that you can program with JavaScript.
- ✓ **Audio & Video** − You can embed audio or video on your webpages without resorting to third-party plugins.
- ✓ **Geolocation** − Now visitors can choose to share their physical location with your web application.
- ✓ **Drag and drop** − Drag and drop the items from one location to another location on the same webpage.

CSS

Introduction to CSS

- ✓ **CSS** is used to control the style of a web document in a simple and easy way.
- ✓ CSS stands for Cascading Style Sheet.
- ✓ CSS is used to design HTML tags or whole document or even a whole collection of documents.
- ✓ CSS is used for web designing. It helps the web designers to apply style on HTML tags.

Why to Learn CSS?

- ✓ Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.
- ✓ Create Stunning Web site - CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs.
- ✓ Become a web designer - If you want to start a career as a professional web designer, HTML and CSS designing is a must skill.
- ✓ Control web - CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.
- ✓ Learn other languages - Once you understand the basic of HTML and CSS then other related technologies like JavaScript, PHP, or angular are become easier to understand.

Advantages of CSS

- ✓ CSS saves time − You can write CSS 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 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.
- ✓ Global web standards − Now HTML attributes are being deprecated and it is being recommended to use CSS. So, it's a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.

Versions

- ✓ The CSS1 specification was developed in 1996
- ✓ CSS2 was released in 1998
- ✓ CSS3 is current version

Standard Levels of Style Sheets

There are three levels of style sheets: in order from lowest level to highest level are

- ✓ Inline CSS
- ✓ Document level Internal Level CSS
- ✓ External Level CSS

1 Inline style sheet: -

- ✓ Inline style sheet applies to the content of the single element (Only for particular tag).
- ✓ If you want to use inline CSS, you should use the style attribute to the relevant tag.
- ✓ The style attribute can contain any CSS property
- ✓ Syntax: <html tag style="cssproperty1:value; cssproperty2:value;"> </html tag>

Example

```
<html>
<body>
<h1 style="color: blue;text-align:center;">This is a heading</h1>
</body>
</html>
```

Disadvantages of Inline CSS

- ✓ These styles cannot be reused anywhere else.
- ✓ These styles are tough to be edited because they are not stored at a single place.
- ✓ It is not possible to style pseudo-codes and pseudo-classes with inline CSS.

2. Document-level style sheets or Internal Style sheets: -

- ✓ Apply to the whole document in which they appear.
- ✓ Document-level style sheets appear in the head of HTML document.
- ✓ The internal style is defined inside the <style> element, inside the head section.

Example

```
<html>
<head>
<style>
body {
  background-color: linen;
}
h1 {
  color: maroon;
  margin-left: 40px;
}
</style>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph</p>
</body>
</html>
```


3. External style sheets: -

- ✓ Can be applied to any number of documents. External style sheets are in separate files having .css file extension.
- ✓ Each HTML page must include a reference to the external style sheet file inside the <link> element, inside the head section
- ✓ A <link> tag is used to specify that the browser is to fetch and use an external style sheet file
- ✓ <link rel = "stylesheet" type = "text/css" href =" "> </link>
- ✓ An external style sheet can be written in any text editor and must be saved with a **.css** extension.
- ✓ The external .css file should not contain any HTML tags.

Example

```
<html>
<head>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph</p>
</body>
</html>
```

Here is how the "mystyle.css" file looks:

```
Body
 {
  background-color: lightblue;
}
h1 {
  color: navy;
  margin-left: 20px;
}
```


Note: -

- ✓ Inline style sheets have precedence over document style sheets, which have precedence over external style sheets.
- ✓ If no style sheet information is specified, the browser default property values are used

The @import directive is an alternative way to use style specifications from other files. The form of this directive is

@import url(file name);

Notice that the file name is not quoted. There are two differences between link and @import:

- ✓ @import can appear only at the beginning of the content of a style element,
- ✓ the imported file can contain markup, as well as style, rules. In fact, sometimes the imported file contains other @import directives, along with some style rules

Style sheets are called **cascading style sheets** because they can be defined at three different levels to specify the style of a document. Lower-level style sheets can override higher level

style sheets, so the style of the content of a tag is determined, in effect, through a cascade of style-sheet applications. When more than one style sheet applies to a specific tag in a document, the lowest level style sheet has precedence.

Style Specification Formats

- ✓ The format of a style specification depends on the level of style sheet.
- ✓ Inline style specifications appear as values of the style attribute of a tag, the general form of which is as follows:

```
style = "property_1:value_1; property_2:value_2; ...;
         property_n:value_n;"
```

Note: -it is recommended that the last property–value pair be followed by a semicolon.

Document style specifications

- ✓ Appear as the content of a style element within the header of a document, although the format of the specification is quite different from that of inline style sheets.
- ✓ The general form of the content of a style element is as follows

```
<style type = "text/css">
  rule_list
</style>
```

- ✓ The type of attribute of the <style> tag tells the browser the type of style specification, which is always text/css.
- ✓ Each style rule in a rule list has two parts: a selector, which indicates the tag or tags affected by the rule, and a list of property–value pairs. The list has the same form as the quoted list for inline style sheets, except that it is delimited by braces rather than double quotes. So, the form of a style rule is as follows:

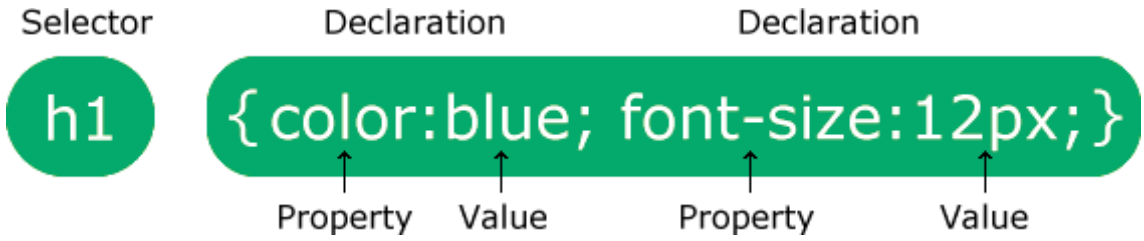
```
selector {property_1:value_1; property_2:value_2; ...;
          property_n:value_n;}
```

- ✓ If a property is given more than one value, those values usually are separated with spaces. For some properties, however, multiple values are separated with commas.
- ✓ Selector − A selector is an HTML tag at which a style will be applied. This could be any tag like <h1> or <table> etc.
- ✓ Property − A property is a type of attribute of HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be color, border etc.
- ✓ Value − Values are assigned to properties. For example, color property can have value

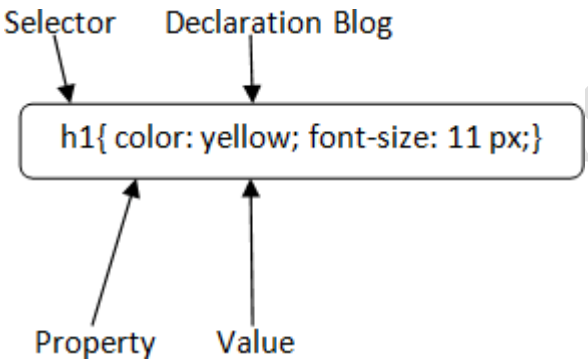
either red or #F1F1F1 etc.

You can put CSS Style Rule Syntax as follows –

```
selector {property: value}
```



Example



Selector: Selector indicates the HTML element you want to style. It could be any tag like <h1>, <title> etc.

Declaration Block: The declaration block can contain one or more declarations separated by a semicolon. For the above example, there are two declarations:

1. color: yellow.
2. font-size: 11 px.

Each declaration contains a property name and value, separated by a colon.

Property: A Property is a type of attribute of HTML element. It could be color, border etc.

Value: Values are assigned to CSS properties. In the above example, value "yellow" is assigned to color property.

Selector {Property1: value1; Property2: value2;..... ;}

Example

```
color: red.
text-align left;
font-size: 8pt
```

CSS Selector or Selector Forms

CSS selectors are used to select the content you want to style. Selectors are the part of CSS rule set. CSS selectors select HTML elements according to its id, class, type, attribute etc.CSS selectors are used to "find" (or select) the HTML elements you want to style.

There are several different types of selectors in CSS.

1. CSS Element Selector or Simple Selector Forms
2. CSS Id Selector
3. CSS Class Selector
4. Generic Selectors
5. CSS Universal Selector
6. CSS Group Selector
7. CSS Pseudo-classes

The CSS element Selector: -

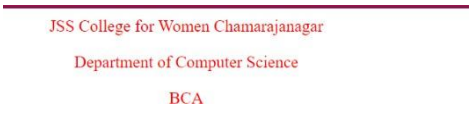
- ✓ The element selector selects HTML elements based on the element name.
- ✓ The selector could be a list of element names separated by commas; in which case the property values apply to all occurrences of all of the named elements

Example

```
<html>
<head>
<style>
p {
  text-align: center;
  color: red;
}
</style>
```

```
</head>
<body>
<p>JSS College for Women Chamarajanagar</p>
<p >Department of Computer Science </p>
<p>BCA</p>
</body>
</html>
```

Output



Selectors can also specify that the style should apply only to elements in certain positions in the document. This is done by listing the element hierarchy in the selector, with only white space separating the element names. For example, the rule

form em {font-size: 14pt;} applies its style only to the content of emphasis elements that are nested in a form element in the document. This is **a contextual selector** (sometimes called **a descendant selector**)

CSS Id Selector

The id selector selects the id attribute of an HTML element to select a specific element. An id is always unique within the page, so it is chosen to select a single, unique element.

The general form of an id selector is as follows

#Specific-id {property-value list}

It is written with the hash character (#), followed by the id of the element.

Let’s take an example with the id "para1".

```
<html>
<head>
<style>
#para1 {
   text-align: center;
   color: blue;
}
</style>
```

```
</head>
<body>
<p id="para1">JSS College for Women Chamarajanagr</p>
<p>Double Road </p>
</body>
</html>
```

Output

JSS College for Women Chamarajanagr

Double Road

Note: An id name cannot start with a number!

The CSS class Selector

- ✓ The class selector selects HTML elements with a specific class attribute.
- ✓ Class selectors are used to allow different occurrences of the same tag to use different style specifications. A style class is defined in a style element by giving the style class a name, which is attached to the tag's name with a period
- ✓ To select elements with a specific class, write a period (.) character, followed by the class name.

Example

```
<html>
<head>
<style>
. center {
  text-align: center;
  color: red;
}
</style>
</head>
<body>
<h1 class="center">JSS College for Women </h1>
<p class="center">Chamarajanagar </p>
</body>
</html>
```

Output

JSS College for Women

Chamarajanagar

You can also specify that only specific HTML elements should be affected by a class.

```
<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>
```

HTML elements can also refer to more than one class.

```
<html>
<head>
<style>
p.center {
  text-align: center;
  color: red;
}
p.large {
  font-size: 300%;
}
</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>
<p class="center large">This paragraph will be red, center-aligned, and in a large font-size.</p>
</body>
</html>
```


Note: A class name cannot start with a number!

Now, in the body of a document, you could have the following markup:

```
<h3 class = "sale"> Weekend Sale </h3>
...
<p class = "sale">
...
</p>
```

The CSS Universal Selector

The universal selector (*) selects all HTML elements on the page.

For example,

* {color: red;}

Example

The CSS rule below will affect every HTML element on the page:

```
<html>
<head>
<style>
* {
  text-align: center;
  color: blue;
}
</style>
</head>
<body>
<h1>Hello world!</h1>
<p>JSS College for women Chamarajanaga.</p>
<p >BCA</p>
<p>Department of Computer Science</p>
</body>
</html>
```


The CSS Grouping Selector

The grouping selector selects all the HTML elements with the same style definitions.

```
<html>
<head>
<style>
h1, h2, p {
  text-align: center;
  color: red;
}
</style>
</head>
<body>
<h1>Hello World!</h1>
<h2>Smaller heading!</h2>
<p>This is a paragraph.</p>
</body>
```

Cascading Order: -What style will be used when there is more than one style specified for an HTML element?

All the styles in a page will "cascade" into a new "virtual" style sheet by the following rules, where number one has the highest priority:

1. Inline style (inside an HTML element)
2. External and internal style sheets (in the head section)
3. Browser default

So, an inline style has the highest priority, and will override external and internal styles and browser default

CSS Comments

- ✓ CSS comments are not displayed in the browser, but they can help document your source code.
- ✓ 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 is placed inside the <style> element, and starts with /* and ends with */:

Example

```
<html>
<head>
<style>
/* This is a single-line comment */
p {
  color: red;
}
</style>
</head>
<body>
<p>Hello World!</p>
<p>This paragraph is styled with CSS.</p>
<p>CSS comments are not shown in the output.</p>
</body>
</html>
```

Property Value Forms

CSS includes 60 different properties in seven categories: fonts, lists, alignment of text, margins, colors, backgrounds, and borders. Property values can appear in a variety of forms. Keyword property values are used when there are only a few possible values and they are predefined—for example, large, medium, and small. Keyword values are not case sensitive, so Small, SmAlL, and SMALL are all the same as small. Number values are used when no meaningful units can be attached to a numeric property value. A number value can be either an integer or a sequence of digits with a decimal point and can be preceded by a sign (+ or -).

Length values are specified as number values that are followed immediately by a two-character abbreviation of a unit name. There can be no space between the number and the unit name. The possible unit names are px, for pixels; in, for inches; cm, for centimeters; mm, for millimeters; pt, for points (a point is 1/72 inch);and pc,

Percentage values are numbers that are followed immediately by a percent sign (%). For example, if the font size were set to 75%, it would make the new current size for the font 75 percent of its previous

URL property values use a form that is slightly different from references to URLs in links

Example **`url(tetons.jpg)`**

There can be no space between url and the left parenthesis.

Color property values can be specified as color names, as six-digit hexadecimal numbers, or in RGB form. RGB form is just the word rgb followed by a parenthesized list of three numbers that specify the levels of red, green, and blue, respectively. The RGB values can be given either as decimal numbers between 0 and 255 or as percentages. Hexadecimal numbers must be preceded with pound signs (#), as in #43AF00. For example, powder blue could be specified with blue or rgb(255, 0, 255) or #FF00FF

Some property values are inherited by elements nested in the element for which the values are specified. For example, the property background-color is not inherited, but font-size is. Using a style sheet to set a value for an inheritable property for the <body> tag effectively sets it as a default property value for the whole document, as in body {font-size: 16pt}

Unless overridden by a style sheet that applies to paragraph elements, every paragraph element in the body of this document would inherit the font size of 16 points.

CSS Fonts

CSS Font property is used to control the look of texts. By the use of CSS font property, you can change the text size, color, style and more.

These are some important font attributes:

1. CSS Font color: This property is used to change the color of the text. (Standalone attribute)
2. CSS Font family: This property is used to change the face of the font.
3. CSS Font size: This property is used to increase or decrease the size of the font.
4. CSS Font style: This property is used to make the font bold, italic or oblique.

5. CSS Font variant: This property creates a small-caps effect.
6. CSS Font weight: This property is used to increase or decrease the boldness and lightness of the font.

CSS Font Family

The font-family property is used to specify a list of font names. The browser uses the first font in the list that it supports. For example, the property: font-family: Arial, Helvetica, Futura tells the browser to use Arial if it supports that font. If not, it will use Helvetica if it supports it. If the browser supports neither Arial nor Helvetica, it will use Futura if it can. If the browser does not support any of the specified fonts, it will use an alternative of its choosing.

If a font name has more than one word, the whole name should be delimited by single quotes,9 as in the following example:

font-family: 'Times New Roman'

CSS font family can be divided in two types:

- ✓ Generic family: It includes Serif, Sans-serif, and Monospace.
- ✓ Font family: It specifies the font family name like Arial, New Times Roman etc.

Example

```
<html>
  <head>
  </head>
  <body>
    <p style = "font-family:georgia,garamond,serif;">
      This text is rendered in either georgia, garamond, or the
      default serif font depending on which font  you have at your system.
    </p>
  </body>
</html>
```

Some Font Examples

Generic Font Family	Examples of Font Names
Serif	Times New Roman,Georgia,Garamond
Sans-serif	Arial,Verdana,Helvetica
Monospace	Courier New,Lucida Console,Monaco
Cursive	Brush Script MT ,Lucida Handwriting
Fantasy	Copperplate,Papyrus

Example

```
<html>
<head>
<style>
.p1 {
  font-family: "Times New Roman", Times, serif;
}
.p2 {
  font-family: Arial, Helvetica, sans-serif;
}
.p3 {
  font-family: "Lucida Console", "Courier New", monospace;
}
```

```
</style>
</head>
<body>
<h1>CSS font-family</h1>
<p class="p1">This is a paragraph, shown in the Times New Roman font.</p>
<p class="p2">This is a paragraph, shown in the Arial font.</p>
<p class="p3">This is a paragraph, shown in the Lucida Console font.</p>
</body>
</html>
```

CSS Font Size

- ✓ CSS font size property is used to change the size of the font

These are the possible values that can be used to set the font size:

Font Size Value	Description
xx-small	used to display the extremely small text size.
x-small	used to display the extra small text size.
Small	used to display small text size.
Medium	used to display medium text size.
Large	used to display large text size.
x-large	used to display extra large text size.
xx-large	used to display extremely large text size.
Smaller	used to display comparatively smaller text size.
Larger	used to display comparatively larger text size.
size in pixels or %	used to set value in percentage or in pixels.

Example

```
<html>
<head>
<title>Practice CSS font-size property</title>
</head>
<body>
<p style="font-size:xx-small;">
This font size is extremely small.</p>
<p style="font-size:x-small;">
This font size is extra small</p>
<p style="font-size:small;">
This font size is small</p>
<p style="font-size:medium;">
This font size is medium. </p>
</body>
</html>
```

CSS Font Style

- ✓ CSS Font style property defines what type of font you want to display. It may be italic, oblique, or normal.

```
<html>
<head>
<style>
body {
font-size: 100%;
}
h2 { font-style: italic; }
h3 { font-style: oblique; }
h4 { font-style: normal; }
}
</style>
</head>
<body>
<h2>This heading is shown in italic font.</h2>
<h3>This heading is shown in oblique font.</h3>
<h4>This heading is shown in normal font.</h4>
</body>
</html>
```

CSS Font Variant

- ✓ CSS font variant property specifies how to set font variant of an element. It may be normal and small-caps.


```
<!DOCTYPE html>
<html>
<head>
<style>
p { font-variant: small-caps; }
h3 { font-variant: normal; }
</style>
</head>
<body>
<h3>This heading is shown in normal font.</h3>
<p>This paragraph is shown in small font.</p>
</body>
</html>
```

CSS Font Weight

- ✓ CSS font weight property defines the weight of the font and specify that how bold a font is. The possible values of font weight may be normal, bold, bolder, lighter or number (100, 200 ..... upto 900).

```
<html>
<body>
<p style="font-weight:bold;">This font is bold.</p>
<p style="font-weight:bolder;">This font is bolder.</p>
<p style="font-weight:lighter;">This font is lighter.</p>
<p style="font-weight:100;">This font is 100 weight.</p>
<p style="font-weight:900;">This font is 900 weight.</p>
</body>
</html>
```

Font shorthand

If more than one font property must be specified, the values can be stated in a list as the value of the font property. The browser then has the responsibility for determining which properties to assign from the forms of the values.

The font property is a shorthand property for:

- ✓ font-style
- ✓ font-variant
- ✓ font-weight
- ✓ font-size/line-height
- ✓ font-family

Example font shorthand **font: italic small-caps bold 12px/30px Georgia, serif;**

CSS Lists

- ✓ Lists can be classified as ordered lists and unordered lists. In ordered lists, marking of the list items is with alphabet and numbers, whereas in unordered lists, the list items are marked using bullets.

We can style the lists using CSS. CSS list properties allow us to:

- ✓ Set the distance between the text and the marker in the list.
- ✓ Specify an image for the marker instead of using the number or bullet point.
- ✓ Control the marker appearance and shape.
- ✓ Place the marker outside or inside the box that contains the list items.
- ✓ Set the background colors to list items and lists.

CSS List Properties are:

- ✓ **list-style-type:** This property is responsible for controlling the appearance and shape of the marker.
- ✓ **list-style-image:** It sets an image for the marker instead of the number or a bullet point.
- ✓ **list-style-position:** It specifies the position of the marker.
- ✓ **list-style:** It is the shorthand property of the above properties.
- ✓ **marker-offset:** It is used to specify the distance between the text and the marker. It is unsupported in IE6 or Netscape 7.

The list-style-type Property

- ✓ The list-style-type property allows you to control the shape or style of bullet point (also known as a marker) in the case of unordered lists and the style of numbering characters in ordered lists.

Here are the values which can be used for an unordered list −

Sr.No.	Value & Description
1	None
2	disc (default) A filled-in circle
3	Circle An empty circle
4	Square A filled-in square

Here are the values, which can be used for an ordered list −

Value	Description	Example
Decimal	Number	1,2,3,4,5
decimal-leading-zero	0 before the number	01, 02, 03, 04, 05
lower-alpha	Lowercase alphanumeric characters	a, b, c, d, e
upper-alpha	Uppercase alphanumeric characters	A, B, C, D, E
lower-roman	Lowercase Roman numerals	i, ii, iii, iv, v
upper-roman	Uppercase Roman numerals	I, II, III, IV, V
lower-greek	The marker is lower-greek	alpha, beta, gamma
lower-latin	The marker is lower-latin	a, b, c, d, e
upper-latin	The marker is upper-latin	A, B, C, D, E
Hiragana	The marker is hiragana	a, i, u, e, o, ka, ki
Katakana	The marker is katakana	A, I, U, E, O, KA, KI
hiragana-iroha	The marker is hiragana-iroha	i, ro, ha, ni, ho, he, to
katakana-iroha	The marker is katakana-iroha	I, RO, HA, NI, HO, HE, TO

Here is an example

```
<html>
   <head>
```

```
   </head>
   <body>
      <ul style = "list-style-type:circle;">
         <li>Maths</li>
         <li>Social Science</li>
         <li>Physics</li>
      </ul>

      <ul style = "list-style-type:square;">
         <li>Maths</li>
         <li>Social Science</li>
         <li>Physics</li>
      </ul>

      <ol style = "list-style-type:decimal;">
         <li>Maths</li>
         <li>Social Science</li>
         <li>Physics</li>
      </ol>

      <ol style = "list-style-type:lower-alpha;">
         <li>Maths</li>
         <li>Social Science</li>
         <li>Physics</li>
      </ol>

      <ol style = "list-style-type:lower-roman;">
         <li>Maths</li>
         <li>Social Science</li>
         <li>Physics</li>
      </ol>
   </body>
</html>
```


It will produce the following result –

- Maths
- Social Science
- Physics

- Maths
- Social Science
- Physics

1. Maths
2. Social Science
3. Physics

a. Maths
b. Social Science
c. Physics

i. Maths
ii. Social Science
iii. Physics

The list-style-position Property

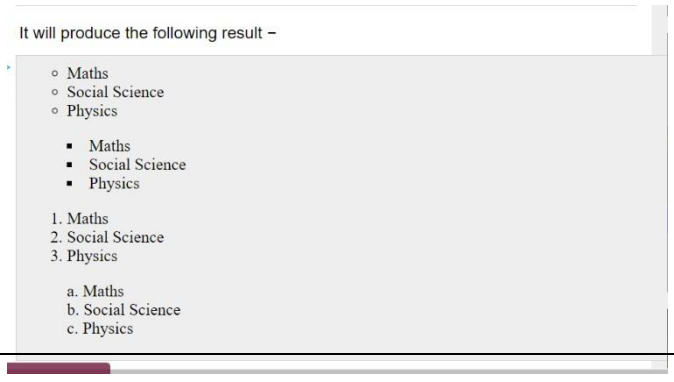
- ✓ The list-style-position property indicates whether the marker should appear inside or outside of the box containing the bullet points. It can have one the two values
- ✓ **None:** - NA

- ✓ **Inside**: -If the text goes onto a second line, the text will wrap underneath the marker. It will also appear indented to where the text would have started if the list had a value of outside.
- ✓ **Outside**: -If the text goes onto a second line, the text will be aligned with the start of the first line (to the right of the bullet).

Example

```
<html>
   <head>
   </head>

   <body>
      <ul style = "list-style-type:circle; list-stlye-
position:outside;">
         <li>Maths</li>
         <li>Social Science</li>
         <li>Physics</li>
      </ul>

      <ul style = "list-style-type:square;list-style-
position:inside;">
         <li>Maths</li>
         <li>Social Science</li>
         <li>Physics</li>
      </ul>

      <ol style = "list-style-type:decimal;list-stlye-
position:outside;">
         <li>Maths</li>
         <li>Social Science</li>
         <li>Physics</li>
      </ol>

      <ol style = "list-style-type:lower-alpha;list-style-
position:inside;">
         <li>Maths</li>
         <li>Social Science</li>
         <li>Physics</li>
      </ol>
   </body>
</html>
```


output

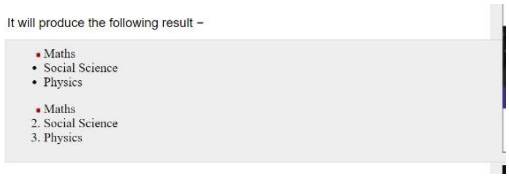


The list-style-image Property

- ✓ The list-style-image allows you to specify an image so that you can use your own bullet style.
- ✓ If it does not find the given image then default bullets are used.

Here is an example −

```
<html>
   <head>
   </head>

   <body>
      <ul>
         <li style = "list-style-image:
url(/images/bullet.gif);">Maths</li>
         <li>Social Science</li>
         <li>Physics</li>
      </ul>

      <ol>
         <li style = "list-style-image:
url(/images/bullet.gif);">Maths</li>
         <li>Social Science</li>
         <li>Physics</li>
      </ol>
   </body>
</html>
```

The list-style Property

- ✓ The *list-style* allows you to specify all the list properties into a single expression. These properties can appear in any order.

Here is an example

```
<html>
   <head>
   </head>
```

```
   <body>
      <ul style = "list-style: inside square;">
         <li>Maths</li>
         <li>Social Science</li>
         <li>Physics</li>
      </ul>

      <ol style = "list-style: outside upper-alpha;">
         <li>Maths</li>
         <li>Social Science</li>
         <li>Physics</li>
      </ol>
   </body>
</html>
```

It will produce the following result –

- Maths
- Social Science
- Physics

A. Maths
B. Social Science
C. Physics

Text Alignment

✓ The **text-align** property in CSS is used for the alignment of text. This CSS property is used to set the horizontal alignment of a table-cell box or the block element. It is similar to the vertical-align property but in the horizontal direction.

✓ The **text-align** property includes values like **justify, center, right, left,** It specifies the horizontal alignment of text in an element.

Syntax

1. text-align: justify | center | left | right;

The value **justify** stretches the content of the element to display the equal width of every line and generally used in magazines and newspapers. The value **left** aligns the text to the left, whereas the value **right** is used to align it to the right. The value ,**center** is used to center the inline text

Example

```
<html>
   <head>
   </head>

   <body>
      <p style = "text-align:right;">
```

```
         This will be right aligned.
      </p>

      <p style = "text-align:center;">
         This will be center aligned.
      </p>

      <p style = "text-align:left;">
         This will be left aligned.
      </p>
   </body>
</html>
```

The **text-indent** property can be used to indent the first line of a paragraph. This property takes either a length or a percentage value, as in the following markup:

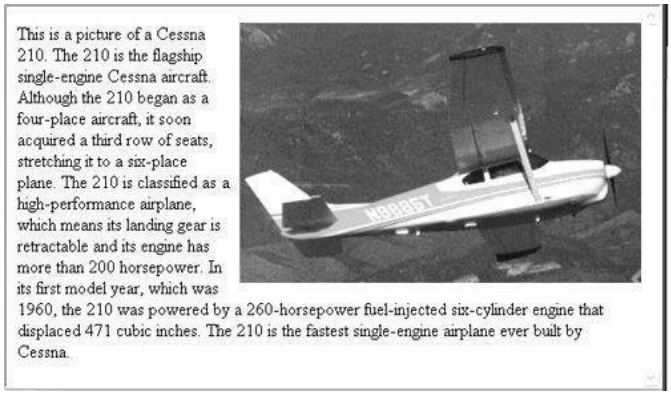
```
<style type = "text/css">
p.indent {text-indent: 0.5in}
</style>
<p class = "indent">
Now is the time for all good Web programmers to begin
using cascading style sheets for all presentation
details in their documents. No more deprecated tags
and attributes, just nice, precise style sheets.
</p>
```


Note :-The default value for text-align is left

The **float property** is used to specify that text should flow around some element, often an image or a table. The possible values for float are left, right, and none, which is the default.

For example

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!-- float.html
     An example to illustrate the float property
     -->
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title> The float property </title>
    <style type = "text/css">
      img {float: right}
    </style>
  </head>
  <body>
    <p>
      <img src = "c210new.jpg"  alt = "Picture of a Cessna 210" />
    </p>
    <p>
      This is a picture of a Cessna 210. The 210 is the flagship
      single-engine Cessna aircraft. Although the 210 began as a
      four-place aircraft, it soon acquired a third row of seats,
      stretching it to a six-place plane. The 210 is classified
      as a high-performance airplane, which means its landing
      gear is retractable and its engine has more than 200
      horsepower. In its first model year, which was 1960,
      the 210 was powered by a 260-horsepower fuel-injected
      six-cylinder engine that displaced 471 cubic inches.
      The 210 is the fastest single-engine airplane ever
      built by Cessna.
    </p>
  </body>
</html>
```



Colors

Types we can Mention the colors in HTML

- ✓ RGB
- ✓ HEX
- ✓ HSL

RGB Color Values

- ✓ In HTML, a color can be specified as an RGB value, using this formula:
- ✓ rgb(*red, green, blue*) Example rgb(250,0,13);
- ✓ Each parameter (red, green, and blue) defines the intensity of the color with a value between 0 and 255.
- ✓ This means that there are 256 x 256 x 256 = 16777216 possible colors!
- ✓ For example, rgb(255, 0, 0) is displayed as red, because red is set to its highest value (255), and the other two (green and blue) are set to 0.
- ✓ Another example, rgb(0, 255, 0) is displayed as green, because green is set to its highest value (255), and the other two (red and blue) are set to 0.
- ✓ To display black, set all color parameters to 0, like this: rgb(0, 0, 0).
- ✓ To display white, set all color parameters to 255, like this: rgb(255, 255, 255).
- ✓ Shade the colors for giving equal number for red, green and blue

Example

```
<html>
<body>
<h1 style="background-color:rgb(255, 0, 0);">rgb(255, 0, 0)</h1>
<h1 style="background-color:rgb(0, 0, 255);">rgb(0, 0, 255) </h1>
<p>In HTML, you can specify colors using RGB values. </p>
</body>
</html>
```

RGBA Color Values

- ✓ RGBA color values are an extension of RGB color values with an Alpha channel - which specifies the opacity for a color.
- ✓ An RGBA color value is specified with: rgba(red, green, blue, alpha)
- ✓ The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all):

Example

```
<html>
<body>
<h1 style="background-color:rgba(255, 99, 71, 0);">rgba(255, 99, 71, 0)</h1>
<h1 style="background-color:rgba(255, 99, 71, 0.2);">rgba(255, 99, 71, 0.2)</h1>
</body>
</html>
```

HTML HEX Colors

- ✓ A hexadecimal color is specified with: #RRGGBB, where the RR (red), GG (green) and BB (blue) hexadecimal integers specify the components of the color.
- ✓ In HTML, a color can be specified using a hexadecimal value in the form:
- ✓ #rrggbb
- ✓ Where rr (red), gg (green) and bb (blue) are hexadecimal values between 00 and ff (same as decimal 0-255).
- ✓ For example, #ff0000 is displayed as red, because red is set to its highest value (ff), and the other two (green and blue) are set to 00.
- ✓ Another example, #00ff00 is displayed as green, because green is set to its highest value (ff), and the other two (red and blue) are set to 00.
- ✓ To display black, set all color parameters to 00, like this: #000000.
- ✓ To display white, set all color parameters to ff, like this: #ffffff.
- ✓ Experiment by mixing the HEX values below: #787976 Shades of the color #404040

Example

```
<html>
<body>
<h1 style="background-color:#ff0000;">#ff0000</h1>
<h1 style="background-color:#0000ff;">#0000ff</h1>
<h1 style="background-color:#3cb371;">#3cb371</h1>
```

```
<p>In HTML, you can specify colors using Hex values.</p>
</body>
</html>
```

Digit HEX Value

- ✓ Sometimes you will see a 3-digit hex code in the CSS source.
- ✓ The 3-digit hex code is a shorthand for some 6-digit hex codes.
- ✓ The 3-digit hex code has the following form:

  #rgb Where r, g, and b represents the red, green, and blue components with values between 0 and f.
- ✓ The 3-digit hex code can only be used when both the values (RR, GG, and BB) are the same for each component. So, if we have #ff00cc, it can be written like this: #f0c.

HSL Color Values

- ✓ HSL stands for hue, saturation, and lightness.
- ✓ HSLA color values are an extension of HSL with an Alpha channel (opacity).
- ✓ HSL Color Values
- ✓ In HTML, a color can be specified using hue, saturation, and lightness (HSL) in the form: hsl(hue, saturation, lightness)
- ✓ Hue is a degree on the color wheel from 0 to 360. 0 is red, 120 is green, and 240 is blue.
- ✓ Saturation is a percentage value, 0% means a shade of gray, and 100% is the full color.
- ✓ Lightness is also a percentage value, 0% is black, and 100% is white.

Example

```
<html>
<body>
<h1 style="background-color:hsl(0, 100%, 50%);">hsl(0, 100%, 50%)</h1>
<p>In HTML, you can specify colors using HSL values.</p>
</body>
</html>
```

HSLA Color Values

- ✓ HSLA color values are an extension of HSL color values with an Alpha channel - which specifies the opacity for a color.
- ✓ An HSLA color value is specified with: hsla(hue, saturation, lightness, alpha)

- ✓ The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all):

Example

```
<html>
<body>
<h1 style="background-color:hsla(9, 100%, 64%, 0);">hsla(9, 100%, 64%, 0)</h1>
<p>You can make transparent colors by using the HSLA color value.</p>
</body>
</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.

The HTML tag is used for grouping and applying styles to inline elements

Example

```
<html>

<body>

<h1>The span element</h1>

<p>My mother has <span style="color:blue;font-weight:bold">blue</span> eyes and my father has <span style="color:darkolivegreen;font-weight:bold">dark green</span> eyes.</p>

</body>

</html>
```

<Div> Tag

- ✓ 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

```
<html>
```

```
<head>
<style>
.myDiv {
  border: 5px outset red;
  background-color: lightblue;
  text-align: center;
}
</style>
</head>
<body>

<h1>The div element</h1>

<div class="myDiv">
  <h2>This is a heading in a div element</h2>
  <p>This is some text in a div element.</p>
</div>

<p>This is some text outside the div element.</p>

</body>
</html>
```

Overview and features of CSS3

- ✓ CSS3, also known as Cascading Style Sheets Level 3, is a more advanced version of CSS and the successor of CSS2.
- ✓ CSS3 is used for the same thing as CSS, namely to style web pages and make them more attractive and user-friendly

features of CSS3

- ✓ New Selectors: More ways to target HTML Elements.
- ✓ Expanded color options and Gradient Effects.
- ✓ Rounded Corners.
- ✓ 2D and 3D transformations for shapes and text.
- ✓ Shadow effect.
- ✓ Transitions and Animations: Smooth and Dynamic Visual Effects.
- ✓ Setting Opacity.