**Computer Multimedia & Animation**

**Unit I**

**Web Design**: Origins and evolution of HTML, Basic syntax, Basic text markup, Images, Lists, Tables, Forms, Frame, Overview and features of HTML5.

**CSS:** Introduction, Levels of style sheets, Style specification formats, Selector forms, Property value forms, Font properties, List properties, Color, Alignment of text, The and tags; Overview and features of CSS3.

**JavaScript:** Object orientation and JavaScript; General syntactic characteristics; Primitives, operations, and expressions; Screen output and keyboard input.

**1.1 Web Design**

**1.1.1 Origins and evolution of HTML**

HTML (Hypertext Markup Language) is the markup language for creating web pages. It provides some titles, headings, paragraphs, lists, tables, embedded images, etc., to describe the structure of text-based and multimedia information in HTML documents. HTML is the basis of a web page, and the web page is the basis of a website. HTML uses '[tags](https://www.w3schools.in/html/tags)' to create web documents.

* HTML (Hypertext Markup Language) is a language for publishing text-based and multimedia information on the World Wide Web.
* HTML was created by **Tim Berners-Lee in late 1991,**but was not officially released. It was published in 1995 as HTML 2.0.
* HTML 4.01 was published in late 1999 and was a major version of HTML. It was developed in the 1990.  **HTML 5** can be said for an extended version of HTML 4.01, which was published in the year 2012.
* **HTML** is the first language of web designing. **CSS** is also used along with HTML to improve web page design further. **JavaScrip**t is used with HTML to make web pages dynamic.
* Web browsers (Chrome, Internet Explorer, Firefox, Safari, and other web browsers) are software' to read HTML and display web page design as output.

We can write HTML in any simple editor, such as **Notepad**. And other software, such as **Adobe Dreamweaver**, **Sublime**, **NetBeans**, **Notepad ++**, etc., are mainly used for writing and editing HTML.

HTML file should be stored in "**.html**" or "**.htm**" .we can write HTML code in any text editor and save it as "**filename.html**" or "**filename.htm**".

**Getting started**

Notepad is a basic application that comes with the Windows operating system. It is commonly used to write and save texts. Below are four steps to write, save, and run HTML using Windows Notepad:

**Step 1**:

open the Notepad and write HTML code on it

***Example HTML Code***

<!DOCTYPE html>

<html>

<head>

This is page heading

</head>

<body>

<p> Hyper text Markup Language</p>

</body>

</html>

**Step 2 :**

save file name with the .html extension , to save the file as an HTML document.

**Step 3:**

Run a saved HTML file by open the file in any web browser.

[**<!DOCTYPE html>**](https://www.geeksforgeeks.org/html-doctypes/)

This is the document type declaration (not technically a tag). It declares a document as being an HTML document. The doctype declaration is not case-sensitive.

[**<html>**](https://www.geeksforgeeks.org/html-html-tag/)

This is called the HTML root element. All other elements are contained within it.

[**<head>**](https://www.geeksforgeeks.org/html-head-tag/#:~:text=The%20tag%20in%20HTML,head%3E%20element%20can%20be%20omitted.)

The head tag contains the “behind the scenes” elements for a webpage. Elements within the head aren’t visible on the front-end of a webpage. HTML elements used inside the <head> element are as follows,

**1.** [**<style>**](https://www.geeksforgeeks.org/html-style-tag/)

This tag allows us to insert styling into our WebPages and make them appealing to look at with the help of CSS.

**2.** [**<title>**](https://www.geeksforgeeks.org/html-title-tag/)

The tag is used to display title in the web browser.

**3.**[**<base>**](https://www.geeksforgeeks.org/html-base-tag/)

It specifies the base URL for all relative URL’s in a document.

**4.** [**<noscript>**](https://www.geeksforgeeks.org/html-noscript-tag/)

Defines a section of HTML that is inserted when the scripting has been turned off in the users browser.

**5.** [**<script>**](https://www.geeksforgeeks.org/html-script-tag/)

This tag is used to add functionality in the website with the help of JavaScript.

**6.** [**<meta>**](https://www.geeksforgeeks.org/html-meta-tag/#:~:text=The%20tag%20in%20HTML,keywords%2C%20document%20author%2C%20etc.)

This tag encloses the meta data of the website that must be loaded every time the website is visited. For eg, the metadata charset allows us to use the standard UTF-8 encoding in our website. This in turn allows the users to view our webpage in the language of their choice. It is a self closing tag.

**7.** [**<link>**](https://www.geeksforgeeks.org/html-link-tag/)

The ‘link’ tag is used to tie together HTML, CSS, and JavaScript. It is a self closing tag.

[**<body>**](https://www.geeksforgeeks.org/html-body-tag/#:~:text=The%20tag%20in%20HTML,well%20as%20an%20ending%20tag.)

The body tag is used to enclose all the visible content of a webpage. In other words, the body content is what the browser will show on the front-end.

**1.1.2 Overview and features of HTML**

**HTML** stands for Hyper Text Markup Language. It is used to design web pages using a markup language. HTML is a combination of Hypertext and Markup language. Hypertext defines the link between web pages. A markup language is used to define the text document within the tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most markup languages (e.g. HTML) are human-readable. The language uses tags to define what manipulation has to be done on the text.

HTML is a markup language used by the browser to manipulate text, images, and other content, in order to display it in the required format.

**Features of HTML**

* It is easy to learn and easy to use.
* It is platform-independent.
* Images, videos, and audio can be added to a web page.
* Hypertext can be added to the text.
* It is a markup language.

**Advantages**

* HTML is simple and easy Language to build websites.
* It is supported by all browsers.
* It can be integrated with other languages like CSS, JavaScript, etc

**Disadvantages**

* HTML can only create static web pages. For dynamic web pages, other languages have to be used.
* A large amount of code has to be written to create a simple web page.
* The security feature is not good.

**HTML History**

Since the early days of the World Wide Web, there have been many versions of HTML

|  |  |
| --- | --- |
| Year | Version |
| 1989 | Tim Berners-Lee invented www |
| 1991 | Tim Berners-Lee invented HTML |
| 1993 | Dave Raggett drafted HTML+ |
| 1995 | HTML Working Group defined HTML 2.0 |
| 1997 | W3C Recommendation: HTML 3.2 |
| 1999 | W3C Recommendation: HTML 4.01 |
| 2000 | W3C Recommendation: XHTML 1.0 |
| 2008 | WHATWG HTML5 First Public Draft |
| 2012 | [WHATWG HTML5 Living Standard](http://whatwg.org/html/) |
| 2014 | [W3C Recommendation: HTML5](http://www.w3.org/TR/html5/) |
| 2016 | W3C Candidate Recommendation: HTML 5.1 |
| 2017 | [W3C Recommendation: HTML5.1 2nd Edition](http://www.w3.org/TR/html51/) |
| 2017 | [W3C Recommendation: HTML5.2](http://www.w3.org/TR/html52/) |

**1.1.3 Basic syntax**

**HTML tags**

HTML tag can define how the browser will display the content and format of the web page. Most tags of HTML have two sections

1) an opening portion

2) a closing portion

any text is written within that has its effect based on the working of the tag. For Example ,

<b> </b> 🡪 any text is written within these tag will be displayed as bold

Tags with opening and closing tags are called **container tags**, and tags with only one opening tag and no closing tag are called **empty tags** (For example, paragraph tag <p> )

**HTML attributes**

Each HTML tag element may have one or more attributes used to define different properties of a particular HTML element.

An attribute is a property name used to provide supplementary information about HTML elements. Some common examples of HTML attributes are id, class, align, etc.

Many attributes are defined globally and are applied to any element, whereas we use some of them only for specific HTML elements.

It takes the general format,

***<element\_name attribute\_name="value"> .... </element\_name>***

where

< element\_name> 🡪opening of tag

attribute\_name 🡪 name of the attribute

value🡪 value given to that attribute

< /element\_name> 🡪 closing of tag

An attribute should be placed within the opening tag. For example,

<body ***bgcolor="green">***  will change the ***background color*** of the web page body to ***green.***

Names of attributes and their values are not case-sensitive. But according to the World Wide Web Consortium (W3C), it is recommended to use lowercase for name and values.

**Core Attributes**

There are four essential attributes that we can apply to almost all HTML elements. These attributes are called core attributes.

1. id
2. title
3. class
4. style

**id attribute**

The ***id*** attribute can be applied to assign a unique identity to any element. It provides a unique identifier for a particular element. When we have two elements of the same name within the same script, the id attribute helps to distinguish two similar elements through the unique ID

**title attribute**

The ***title***attribute gives a recommended title for a element. Its behavior depends on the element upon which it's implemented. It is often implemented to display a **tooltip** if the cursor hovers (comes over) the element.

**class attribute**

The ***class*** attribute is implemented by combining an element through a stylesheet (CSS) and identifying its class element. It is used to specify the rules for Cascading Style Sheet (CSS) in our element.

**style attribute**

The style attribute specifies an inline style for an element. The style attribute will override any style set globally, e.g. styles specified in the <style> tag or in an external style sheet.

**Some other popular HTML attributes are**

|  |  |
| --- | --- |
| **Attribute** | **Usage** |
| align | It is used to align HTML tags horizontally, left, right, or center. |
| background | It is used to set a background image behind an HTML element |
| accesskey | It is used to add a specific keyboard shortcut to access an HTML element. |
| contextmenu | It is used to specify the context menu of an HTML element. |
| item | It is used to group HTML elements. |
| hidden | It is used to specify whether the HTML element should be visible or not. |

**HTML documents**

* All HTML documents begin with <! DOCTYPE html>
* The HTML document begins with <html> and ends with </html>
* HTML code that is written inside the <head> and </head> tags of an HTML document are visible in the web browser title bar.
* Only the content inside the <body> and </body> tags are displayed in the browser

**1.1.3 Basic text markup**

Text content of an HTML document can be formatted with HTML tag. Formatting elements were designed to display special types of text HTML Text Formatting Elements are given in the table.

|  |  |
| --- | --- |
| **Tag** | **Description** |
| [<b>](https://www.w3schools.com/tags/tag_b.asp) | Defines bold text |
| [<em>](https://www.w3schools.com/tags/tag_em.asp) | Defines emphasized text (looks like italic text) |
| [<i>](https://www.w3schools.com/tags/tag_i.asp) | Defines a part of text in an alternate voice or mood |
| [<small>](https://www.w3schools.com/tags/tag_small.asp) | Defines smaller text |
| [<strong>](https://www.w3schools.com/tags/tag_strong.asp) | Defines important text |
| [<sub>](https://www.w3schools.com/tags/tag_sub.asp) | Defines subscripted text |
| [<sup>](https://www.w3schools.com/tags/tag_sup.asp) | Defines superscripted text |
| [<ins>](https://www.w3schools.com/tags/tag_ins.asp) | Defines inserted text |
| [<del>](https://www.w3schools.com/tags/tag_del.asp) | Defines deleted text |
| [<mark>](https://www.w3schools.com/tags/tag_mark.asp) | Defines marked/highlighted text |
| <hr> | Defines Horizontal rule |

The **<h1>** to **<h6>** tags are used to define HTML headings. **<h1>** defines the most important heading. **<h6>** defines the least important heading. For example,

<h1>This is heading 1</h1>  
<h2>This is heading 2</h2>  
<h3>This is heading 3</h3>  
<h4>This is heading 4</h4>  
<h5>This is heading 5</h5>  
<h6>This is heading 6</h6>

**<b> tag** The HTML <b> element defines bold text, without any extra importance.

**<strong> tag** The HTML <strong> element defines text with strong importance. The content inside is typically displayed in bold.

**<i> tag** The HTML <i> element defines a part of text in an alternate voice or mood. The content inside is typically displayed in italic. **italic**  is often used to indicate a technical term, a phrase from another language, a thought, a ship name, etc.

**<em> tag** The HTML element defines emphasized text. The content inside is typically displayed in italic. Usually, a screen reader will pronounce the words in <em> with an emphasis, using verbal stress.

**<small> tag** The HTML <small> element defines smaller text

**<mark> tag** The HTML <mark> element defines text that should be marked or highlighted

**<del> tag** The HTML <del> element defines text that has been deleted from a document. Browsers will usually strike a line through deleted text

**<ins> tag** The HTML <ins> element defines a text that has been inserted into a document. Browsers will usually underline inserted text

**<sub> tag** The HTML <sub> element defines subscript text. Subscript text appears half a character below the normal line, and is sometimes rendered in a smaller font. Subscript text can be used for chemical formulas, like H2O

**<sup> tag**

The HTML <sup> element defines superscript text. Superscript text appears half a character above the normal line, and is sometimes rendered in a smaller font. Superscript text can be used for footnotes, like WWW[1]

## HTML Quotation and Citation Elements

|  |  |
| --- | --- |
| Tag | Description |
| [<abbr>](https://www.w3schools.com/tags/tag_abbr.asp) | Defines an abbreviation or acronym |
| [<address>](https://www.w3schools.com/tags/tag_address.asp) | Defines contact information for the author/owner of a document |
| [<bdo>](https://www.w3schools.com/tags/tag_bdo.asp) | Defines the text direction |
| [<blockquote>](https://www.w3schools.com/tags/tag_blockquote.asp) | Defines a section that is quoted from another source |
| [<cite>](https://www.w3schools.com/tags/tag_cite.asp) | Defines the title of a work |
| [<q>](https://www.w3schools.com/tags/tag_q.asp) | Defines a short inline quotation |

**<blockquote> tag**

The HTML <blockquote> element defines a section that is quoted from another source. Browsers usually indent <blockquote> elements.

**<q> tag**

The HTML <q> tag defines a short quotation. Browsers normally insert quotation marks around the quotation.

**<abbr> tag**

The HTML <abbr> tag defines an abbreviation or an acronym, like "HTML", "CSS", "Mr.", "Dr.", "ASAP", "ATM". Marking abbreviations can give useful information to browsers, translation systems and search-engines.

**<address> tag**

The HTML <address> tag defines the contact information for the author/owner of a document or an article. The contact information can be an email address, URL, physical address, phone number, social media handle, etc. The text in the <address> element usually renders in *italic,* and browsers will always add a line break before and after the <address> element.

**<cite> tag**

The HTML <cite> tag defines the title of a creative work (e.g. a book, a poem, a song, a movie, a painting, a sculpture, etc.).The text in the <cite> element usually renders in *italic*.

**<bdo> tag**

BDO stands for Bi-Directional Override. The HTML <bdo> tag is used to override the current text direction

comments in HTML document

**Comment <! > tag**

whatever in-between <! and > is considered as a comment. Comments are not displayed by the browser. It is for reference only and we can see them in HTML source code.

**Example 1.2 - Displaying heading in webpages**

<html>

<head>

<title>

</title>

</head>

<body bgcolor="palegreen">

Heading Tags

<hr>

<h1>Eye is the lamp of the body</h1>

<h2>Eye is the lamp of the body</h2>

<h3>Eye is the lamp of the body</h3>

<h4>Eye is the lamp of the body</h4>

<h5>Eye is the lamp of the body</h5>

<h6>Eye is the lamp of the body</h6>

</body>

</html>

**Example 1.3 - Formatting tags in webpage**

<html>

<head>

<title> Formatting tags

</title>

</head>

<body bgcolor="cyan ">

Formatting tags

<br><b>This is Bold</b>

<br><i>This is Italic</i>

<br><u>This is Underline</u>

<br><tt> This is type writer</tt>

<br><sub >This text is subscripted </sub>

<br><sup >This text is superscripted </sup>

</body>

</html>

# HTML Colors

HTML colors are specified with predefined color names, or with RGB, HEX, HSL, RGBA, or HSLA values. HTML supports [140 standard color names](https://www.w3schools.com/colors/colors_names.asp).

**RGB - rgb(red, green, blue)**

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 colors are possible. For example,

rgb(255, 0, 0) represents red

rgb(0, 255, 0) represents green

rgb(0, 0, 0) represents black

rgb(255, 255, 255) represents white

**RGBA Color Values - rgba(red, green, blue, alpha)**

RGBA color values are an extension of RGB color values with an Alpha channel - which specifies the opacity for a color. The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all)

**HEX Color Values** **#rrggbb**

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 represents red

#00ff00 represents green

#000000 represents black

#ffffff represents white

**HSL Color-** **hsl(*hue*, *saturation*, *lightness*)**

HSL stands for hue, saturation, and lightness. Hue is a degree on the color wheel from 0 to 360. for example, 0 is red 120 is green 240 is blue

**Saturation** is a percentage value. For example,

0% means a shade of gray 100% is the full color

**Lightness** is also a percentage value. For example,

0% is black 100% is white

**HSLA color**

HSLA color values are an extension of HSL with an Alpha channel (opacity).

**1.1.4 Images**

Images can improve the design and the appearance of a web page. <img> tag is used to embed an image in a web page.

<img> has the attributes as follows,

**src**  defines the URL of the image

**alt** defines an alternate text for an image, if it cannot be displayed

**width** defines the width of the image

**height**  defines the width of the image

**float**  to let the image float to the left or to the right

Always specify the width and height of an image. If width and height are not specified, the web page might flicker while the image loads.

**Example 1.4 Displaying Image in web page**

<!DOCTYPE html>

<html>

<body>

<img src="html5.png" alt="HTML5 Icon" width="128" height="128">

</body>

</html>

**Common Image Formats**

Here are the most common image file types, which are supported in all browsers (Chrome, Edge, Firefox, Safari, Opera)

|  |  |  |
| --- | --- | --- |
| **Abbreviation** | **File Format** | **File Extension** |
| APNG | Animated Portable Network Graphics | .apng |
| GIF | Graphics Interchange Format | .gif |
| ICO | Microsoft Icon | .ico, .cur |
| JPEG | Joint Photographic Expert Group image | .jpg, .jpeg, .jfif, .pjpeg, .pjp |
| PNG | Portable Network Graphics | .png |
| SVG | Scalable Vector Graphics | .svg |

**1.1.5 Lists**

HTML lists are used to display a set of related items in lists. List tag and their Description are listed below.

## HTML List Tags

|  |  |
| --- | --- |
| Tag | Description |
| [<ul>](https://www.w3schools.com/tags/tag_ul.asp) | Defines an unordered list |
| [<ol>](https://www.w3schools.com/tags/tag_ol.asp) | Defines an ordered list |
| [<li>](https://www.w3schools.com/tags/tag_li.asp) | Defines a list item |
| [<dl>](https://www.w3schools.com/tags/tag_dl.asp) | Defines a description list |
| [<dt>](https://www.w3schools.com/tags/tag_dt.asp) | Defines a term in a description list |
| [<dd>](https://www.w3schools.com/tags/tag_dd.asp) | Describes the term in a description list |

## Unordered List

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.

## Ordered List

An ordered list starts with the <ol> tag. Each list item starts with the <li> tag.

The list items will be marked with numbers by default.

## Description Lists

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 (name), and the <dd> tag describes each term

**Example 1. 5 Displaying list in web page**

<html>

<head>

<title> ordered and unordered list

</title>

</head>

<body bgcolor="olivegreen ">

<h1> Programming Languages</h1>

<ol type=i>

<li> C </li>

<li> C++ </li>

<li> C# </li>

<li> JAVA </li>

<li>Python</li>

<h1>Operating System</h1>

<ul type=disc>

<li> Windows</li>

<li> Unix </li>

<li> Linux</li>

<li> Mac OS</li>

</ol>

</body>

</html>

**1.1.6 Tables**

HTML tables allow web developers to arrange data into rows and columns. A table in HTML consists of table cells inside rows and columns.

**HTML Table Tags**

|  |  |
| --- | --- |
| Tag | Description |
| [<table>](https://www.w3schools.com/tags/tag_table.asp) | Defines a table |
| [<th>](https://www.w3schools.com/tags/tag_th.asp) | Defines a header cell in a table |
| [<tr>](https://www.w3schools.com/tags/tag_tr.asp) | Defines a row in a table |
| [<td>](https://www.w3schools.com/tags/tag_td.asp) | Defines a cell in a table |
| [<caption>](https://www.w3schools.com/tags/tag_caption.asp) | Defines a table caption |
| [<colgroup>](https://www.w3schools.com/tags/tag_colgroup.asp) | Specifies a group of one or more columns in a table for formatting |
| [<col>](https://www.w3schools.com/tags/tag_col.asp) | Specifies column properties for each column within a <colgroup> element |
| [<thead>](https://www.w3schools.com/tags/tag_thead.asp) | Groups the header content in a table |
| [<tbody>](https://www.w3schools.com/tags/tag_tbody.asp) | Groups the body content in a table |
| [<tfoot>](https://www.w3schools.com/tags/tag_tfoot.asp) | Groups the footer content in a table |

## Table Cells

Each table cell is defined by a **<td>** and a **</td>** tag. td stands for table data. Everything between <td> and </td> are the content of the table cell. A table cell can contain all sorts of HTML elements such as text, images, lists, links and tables.

**Table rows**

Each table row starts with a <tr> and ends with a </tr> tag. tr stands for table row.

**Table Headers**

Table header is used to make a cell as header. In those cases use the <th> tag instead of the <td> tag. th stands for table header.

**Table Attributes**

***Rowspan, colspan, border, bgcolor, cell padding***

***rowspan***attribute is used to make a cell span over multiple rows

***Cell padding*** is the space between the cell edges and the cell content. By default the padding is set to 0.

***Cell spacing*** is the space between each cell. By default the space is set to 2 pixels.

***colspan*** attribute is used to span cells over multiple columns

**Example 1. 6 Displaying table in web page**

<html>

<body bgcolor=yellow>

<table border=2 bgcolor=cyan>

<th colspan="2" bgcolor=red>Student List</th>

<tr><th >sno</th><th>name</th></tr>

<tr><td>1</td><td>Azad</td></tr>

<tr><td>2</td><td>Farook</td></tr>

<tr><td>3</td><td>Jack</td></tr>

<tr><td>4</td><td>Thilak</td></tr>

<tr><td>5</td><td>mahesh</td></tr>

</table>

</body>

**1.1.7 Forms**

An HTML form is used to collect user input. The user input is most often sent to a server for processing. The **<form>** element is a container for different types of input elements such as text fields, checkboxes, radio buttons, submit buttons, etc. The HTML <form> element can contain one or more of the following form elements

## HTML Form Elements

|  |  |
| --- | --- |
| **Tag** | **Description** |
| [<form>](https://www.w3schools.com/tags/tag_form.asp) | Defines an HTML form for user input |
| [<input>](https://www.w3schools.com/tags/tag_input.asp) | Defines an input control |
| [<textarea>](https://www.w3schools.com/tags/tag_textarea.asp) | Defines a multiline input control (text area) |
| [<label>](https://www.w3schools.com/tags/tag_label.asp) | Defines a label for an <input> element |
| [<fieldset>](https://www.w3schools.com/tags/tag_fieldset.asp) | Groups related elements in a form |
| [<legend>](https://www.w3schools.com/tags/tag_legend.asp) | Defines a caption for a <fieldset> element |
| [<select>](https://www.w3schools.com/tags/tag_select.asp) | Defines a drop-down list |
| [<optgroup>](https://www.w3schools.com/tags/tag_optgroup.asp) | Defines a group of related options in a drop-down list |
| [<option>](https://www.w3schools.com/tags/tag_option.asp) | Defines an option in a drop-down list |
| [<button>](https://www.w3schools.com/tags/tag_button.asp) | Defines a clickable button |
| [<datalist>](https://www.w3schools.com/tags/tag_datalist.asp) | Specifies a list of pre-defined options for input controls |
| [<output>](https://www.w3schools.com/tags/tag_output.asp) | Defines the result of a calculation |

## <label> Element

The **<label>** element defines a label for several form elements. The **<label>** element is useful for screen-reader users, because the screen-reader will read out loud the label when the user focus on the input element.

The **<label>** element also help users who have difficulty clicking on very small regions (such as radio buttons or checkboxes) - because when the user clicks the text within the **<label>**element, it toggles the radio button/checkbox.

**The <select> Element**

The **<select>** element defines a drop-down list. The **<option>** elements defines an option that can be selected. By default, the first item in the drop-down list is selected.. It has the properties **selected**, **size**, **multiple** etc.,

***size***  attribute is used to specify the number of visible values

***multiple***  attribute is used to select more than one value

***selected*** attribute is used to define a pre-selected option

**Example 1.7 HTML -select tag**

<html>

<body>

<label for="cars">Choose a car:</label>

<select id="cars" name="cars" size="4" multiple>

<option value="volvo">Volvo</option>

<option value="saab">Saab</option>

<option value="fiat" selected>Fiat</option>

<option value="audi">Audi</option>

</select>

</body>

</html>

**<textarea> Element**

The **<textarea>** element defines a multi-line input field (a text area). The rows attribute specifies the visible number of lines in a text area. The cols attribute specifies the visible width of a text area. For example,

<textarea name="message" rows="10" cols="30">  
Write Your comment here..  
</textarea>

## The <button> Element

## The <button> element defines a clickable button. for example.,

<button type="button" onclick="alert('Hello World!')">Click Me!</button>

## The <fieldset> and <legend> Elements

## The <fieldset> element is used to group related data in a form. The <legend> element defines a caption for the <fieldset> element.

### Example 1.8 HTML - legend and fieldset

<form >   
  <fieldset>  
    <legend>Personal Info:</legend>  
    <label for="fname">First name:</label><br>  
    <input type="text" id="fname" name="fname" value="Christy"><br>  
    <label for="lname">Last name:</label><br>  
    <input type="text" id="lname" name="lname" value="V"><br><br>  
    <input type="submit" value="Submit">  
  </fieldset>  
</form>

Bottom of Form

## <datalist> Element

The <datalist> element specifies a list of pre-defined options for an <input> element. Users will see a drop-down list of the pre-defined options as they input data. The list attribute of the <input> element, must refer to the id attribute of the <datalist> element.

**Example 1.9 HTML - datalist tag**

<form action="/action\_page.php">  
Your's Default browser..

  <input list="browsers">  
  <datalist id="browsers">  
    <option value="Internet Explorer">  
    <option value="Firefox">  
    <option value="Chrome">  
    <option value="Opera">  
    <option value="Safari">  
  </datalist>  
</form>

## The <output> Element

The <output> element represents the result of a calculation (like one performed by a script).

### Example 1.10 Perform a calculation and display the result in an <output> element

<form   oninput="x.value=parseInt(a.value)+parseInt(b.value)">  
  0  
  <input type="range"  id="a" name="a" value="50">  
  100 +  
  <input type="number" id="b" name="b" value="50">  
  =  
  <output name="x" for="a b"></output>  
  <br><br>  
  <input type="submit">  
</form>

Bottom of Form

## The HTML <input> Element

An **<input>** element can be displayed in many ways, depending on the**type** attribute. There are the different input types available in HTML

1. <input type="button">
2. <input type="checkbox">
3. <input type="color">
4. <input type="date">
5. <input type="datetime-local">
6. <input type="email">
7. <input type="file">
8. <input type="hidden">
9. <input type="image">
10. <input type="month">
11. <input type="number">
12. <input type="password">
13. <input type="radio">
14. <input type="range">
15. <input type="reset">
16. <input type="search">
17. <input type="submit">
18. <input type="tel">
19. <input type="text">
20. <input type="time">
21. <input type="url">
22. <input type="week">

It is used to specify the type of <input> element to display. The default type of <input> type attribute is text.

|  |  |
| --- | --- |
| **Type value** | **Usage** |
| [button](https://www.geeksforgeeks.org/html-input-type-button/) | It is used to define a clickable Button in a Document. It is mostly used with the JavaScript to activate the script. |
| [checkbox](https://www.geeksforgeeks.org/html-input-typecheckbox/) | It is used to define a checkbox field. The checkbox is shown as a square box that is ticked when it is activated. It allows the user to select one or more option among all the limited choices. |
| [color](https://www.geeksforgeeks.org/html-input-typecolor/) | It is used to define a color picker. The value should be a seven-character hexadecimal notation. Its default value is #000000(black). |
| [date](https://www.geeksforgeeks.org/html-input-typedate/) | It is used to define a date picker or control field. The value will be the year, month and day. |
| [email](https://www.geeksforgeeks.org/html-input-typeemail/) | It is used to define a field for email address. The input email id is automatically validated to check the format of the email id is correct or not. |
| [file](https://www.geeksforgeeks.org/html-input-typefile/) | It is used to specify the file select field and add a button to choose a file for upload to the form. |
| [hidden](https://www.geeksforgeeks.org/html-input-typehidden/) | It is used to define an input hidden field. A hidden field also includes those data that could not be seen or modified by the users when submitted the form. A hidden field only stores those database records that need to be updated when submitting the form. |
| [image](https://www.geeksforgeeks.org/html-input-typeimage/) | It is used to define an image as the submit button. |
| [month](https://www.geeksforgeeks.org/html-input-typemonth/) | It is used to specify the control of month and year field. The value must be in the format of “YYYY-MM”. |
| [number](https://www.geeksforgeeks.org/html-input-typenumber/) | It is used to specify an input field for entering a number. |
| [password](https://www.geeksforgeeks.org/html-input-typepassword/) | It is used to specify the password field of input tag. Password should be served over the HTTPS pages because it include the sensitive information of the user. |
| [radio](https://www.geeksforgeeks.org/html-input-typeradio/) | It is used to define a Radio Button. Radio Buttons are used to let the user select exactly one option from a list of predefined options. Radio Button input controls are created by using the “input” element with a type attribute having value as “radio”. |
| [range](https://www.geeksforgeeks.org/html-input-typerange/) | It is used to define control for a number entered by the user. It can set restrictions on unimportant number or value which will be entered by the user. Its default range from 0 to 100. |
| [reset](https://www.geeksforgeeks.org/html-input-typereset/) | It is used to defines a reset button. The reset button is used to reset all the form values to its initial values. |
| [search](https://www.geeksforgeeks.org/html-input-typesearch/) | It is used to define a text field that entered a search string. |
| [submit](https://www.geeksforgeeks.org/html-input-typesubmit/) | It is used to define a submit button. It is used to submit all the user value to the form handler. The Form Handler is a server page that activates a script for processing all the input values. |
| [tel](https://www.geeksforgeeks.org/html-input-typetel/) | It is used to define a field that entering a user telephone Number. |
| [text](https://www.geeksforgeeks.org/html-input-typetext/) | It is used to define a single-line text field . The default width of the text field is 20 characters. |
| [time](https://www.geeksforgeeks.org/html-input-typetime/) | It is used to specify the entering time control field. |
| [url](https://www.geeksforgeeks.org/html-input-typeurl/) | It is used to define a field that entered a URL. This input value is automatically validated before submitted the form. |
| [week](https://www.geeksforgeeks.org/html-input-typeweek/) | It is used to define a week and a year control field. |

### Example 1.11 HTML form  <input> element

<!DOCTYPE html>

<html>

<head> <title>List Of Inputs in HTML</title>

</head>

<body>

<h1 style="color: green">

Student Information

</h1>

<h2> List Of Inputs in HTML </h2>

<input type="button">click Button if You agree<br><br>

<input type="checkbox">Check If you are 18+<br><br>

<input type="color">Choose the color of cloth<br><br>

<input type="date">Choose You birth

date<br><br>

<input type="email" placeholder="Enter Your Mail"><br><br>

<input type="file">Input the required file<br><br>

<input type="image">Input You Image<br><br>

<input type="month">Month Of admission<br><br>

<input type="number" placeholder="Enter Your Age"><br><br>

</body>

**<**/html>

# HTML Form Attributes

## Available form attributes are : *Action, target, method*

**action attribute**

The**action** attribute defines the action to be performed when the form is submitted. Usually, the form data is sent to a file on the server when the user clicks on the submit button. If the **action** attribute is omitted, the action is set to the current page.

The <input type="submit"> defines a button for submitting the form data to a form-handler. The form-handler is specified in the form's **action** attribute. The form-handler is typically a file on the server with a script for processing input data.

For example,

## <form action="/action\_page.php">

Here, the form data is sent to a file called "action\_page.php". This file contains a server-side script that handles the form data

## target Attribute

The**target** attribute specifies where to display the response that is received after submitting the form.

The**target** attribute can have one of the following values

|  |  |
| --- | --- |
| **Value** | **Description** |
| \_blank | The response is displayed in a new window or tab |
| \_self | The response is displayed in the current window |
| \_parent | The response is displayed in the parent frame |
| \_top | The response is displayed in the full body of the window |
| *framename* | The response is displayed in a named iframe |

The default value is **\_self** which means that the response will open in the current window.

For example,

<form action="/action\_page.php" target="\_blank">

Here, action\_page.php will be opened in new browser tab.

## Method Attribute

The**method** attribute specifies the HTTP method to be used when submitting the form data. It takes either **get** or **post** value. Default value is **get**.

for example,

<form action="/action\_page.php" method="post">

**Notes on GET**

* Appends the form data to the URL, in name/value pairs
* The length of a URL is limited up to 2048 characters
* Useful for form submissions where a user wants to bookmark the result
* GET is good for non-secure data, like query strings in Google
* GET is not useful to send sensitive data. Because, the submitted form data is visible in the URL

**Notes on POST:**

* Appends the form data inside the body of the HTTP request . So, the submitted form data is not shown in the URL.
* POST has no size limitations, and can be used to send large amounts of data.
* Form submissions with POST cannot be bookmarked

**1.1.8 Frame**

HTML Frames are used to divide the web browser window into multiple sections where each section can be loaded separately. A collection of frames in the browser window is known as a frameset.

**Creating Frames**

 Instead of using body tag, use **frameset** tag in HTML to use frames in web browser. But this Tag is deprecated in HTML 5. The frameset tag is used to define how to divide the browser.

Each frame is indicated by **frame** tag and it basically defines which HTML document shall open into the frame. To define the horizontal frames use **row** attribute of frame tag in HTML document and to define the vertical frames use **col** attribute of frame tag in HTML document.

**Example 1. 12 Displaying list in web page**

<!DOCTYPE html>

<html>

    <head>

        <title>Example of HTML Frames using row attribute</title>

    </head>

        <frameset rows = "20%, 60%, 20%">

        <frame name = "top" src =  " attr1.png" />

        <frame name = "main" src = "gradient3.png" />

        <frame name = "bottom" src = "col\_last.png" />

    </frameset>

</html>

## Disadvantages of Frames

There are few drawbacks with using frames, so it's never recommended to use frames in r webpages −

* Some smaller devices cannot cooperate with frames often because their screen is not big enough to be divided up.
* Sometimes page will be displayed differently on different computers due to different screen resolution.
* The browser's *back* button might not work as the user hopes.
* There are still few browsers that do not support frame technology.

**<iframe> tag**

The <iframe> tag defines a rectangular region within the document in which the browser can display a separate document, including scrollbars and borders. An inline frame is used to embed another document within the current HTML document.

The **src** attribute is used to specify the URL of the document that occupies the inline frame.

**<iframe> Tag Attributes**

*name, class, frameborder, id, longdesc, marginheight, marginwidth, name, scrolling, style,* and *title* behave exactly like the corresponding attributes for the <frame> tag. These attributes are deprecated in HTML5.

**1.2 CSS**

**1.2.1 Introduction**

CSS stands for Cascading Style Sheet that describes how HTML elements should be displayed.

**1.2.1 Overview and features of CSS3**

**C**ascading **S**tyle **S**heets(**CSS)** is a designed language intended to simplify the process of making web pages presentable. CSS allows us to apply styles to web pages. More importantly, CSS enables us to do this independently of the HTML that makes up each web page. It describes how a webpage should look. It prescribes colors, fonts, spacing, and much more. In short, we can make our website look however we want. CSS lets developers and designers define how it behaves, including how elements are positioned in the browser.

While HTML uses tags, CSS uses rule sets. CSS is easy to learn and understand, but it provides powerful control over the presentation of an HTML document.

**Features of CSS**

**CSS saves time**

We can write CSS once and reuse the same sheet in multiple HTML page

**Easy Maintenance**

To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.

**Search Engines**

CSS is considered a clean coding technique, which means search engines won’t have to struggle to “read” its content.

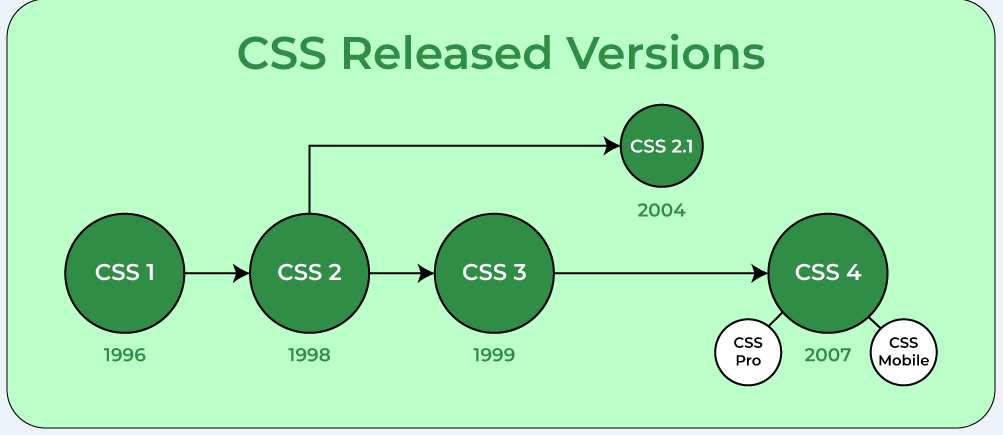
**Superior styles to HTML**

CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.

**Offline Browsing**

CSS can store web applications locally with the help of an offline cache. Using this we can view offline websites.

**CSS versions release years**



**CSS Syntax**

CSS comprises style rules that are interpreted by the browser and then applied to the corresponding elements in your document. A style rule set consists of a selector and declaration block. It takes the general format,

**Selector**

**{**

**property:value;**

**}**

where

selector 🡪 HTML element we want to style.

property 🡪 property name

value 🡪 value given to that property

CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces. Each declaration includes a CSS property name and a value, separated by a colon. The declaration block may contains one or more declarations separated by semicolons.

for example,

h1

{

color:blue;

font size:12px;

}

**color** is property and **blue** is value

**Css Effescts On Html Elements**

**Without CSS (plain HTML)**

<!DOCTYPE html>

<html>

<head>

    <title>Example</title>

</head>

<body>

    <main>

       <h1>HTML Page</h1>

       <p>This is a basic web page.</p>

    </main>

</body>

</html>

**With CSS(Cascaded Style Sheet)**

<!DOCTYPE html>

<html>

<head>

    <title>Example</title>

    <style>

        main

{

            width: 600px;

            height: 200px;

            padding: 10px;

            background: beige;

         }

        h1 {

             color: olivedrab;

             border-bottom: 1px dotted darkgreen;

         }

        p {

            font-family: sans-serif;

            color: orange;

         }

    </style>

</head>

<body>

    <main>

        <h1>My first Page</h1>

        <p>This is a basic web page.</p>

    </main>

</body>

</html>

**1.2.2 Levels of style sheets**

CSS property can be included in the HTML page in a number of different ways. HTML documents are formatted according to the information in the style sheet which is to be included.

**CSS style sheets**

CSS can be added to HTML documents in 3 ways

**Inline style sheets** - by using the **style** attribute inside HTML elements

**Internal style sheets**  - by using a **<style>** element in the **<head>** section

**External style sheets**  - by using a **<link>** element to link to an external CSS file

The most common way to add CSS, is to keep the styles in external CSS files

**External style sheet**

CSS property written in a separate file with .css extension and should be linked to the HTML document using link tag. These type of style sheets are called external style sheet. For each element, style can be set only once and that will be applied across web pages.

External style sheet can be included by using **<link>** tag or **@import**

**External style sheet using <link> tag**

The **<link>** tag is used to link the external style sheet with the html webpage. The <link> tag is placed inside of the HTML <head> element.

<link rel="stylesheet" href="style.css">

**External style sheet using the @import (At-Rule)**

 At-rule method must be included either within <style> tag or else inside the style sheet.  For example,

<style>

@import url(style.css);

</style>

**Example 1.13 External CSS**

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" href="mycss.css"/>

</head>

<body>

<div class = "main">

<div class ="heading">Multimedia and Animation</div>

<p > IV Semester BCA

</body>

</html>

**mycss.css**

body { background-color:powderblue; }

.main { text-align:center; }

.heading {

color:#009900;

font-size:50px;

font-weight:bold;

}

p

{

font-style:bold;

font-size:20px;

color:#785600;

}

**Internal style sheet**

An internal style sheet is defined in the **<head>** section of an HTML page, within a **<style>** element. The following example sets the text color of ALL the **<h1>** elements (on that page) to blue, and the text color of ALL the **<p>** elements to red. In addition, the page will be displayed with a "powderblue" background color.

**Example 1.14 Internal style sheet**

<!DOCTYPE html>  
<html>  
<head>  
<style>  
body {background-color: powderblue;}  
h1   {color: blue;}  
p    {color: red;}  
</style>  
</head>  
<body>  
<h1>This is a heading</h1>  
<p>This is a paragraph.</p>  
</body>  
</html>

**Inline style sheet**

An inline style sheet is used to apply a unique style to a single HTML element. It uses the **style** attribute of an HTML element. The following example sets the text color of the **<h1>** element to blue, and the text color of the **<p>** element to red

<h1 style="color:blue;">A Blue Heading</h1>  
<p style="color:red;">A red paragraph</p>

Note :

***The External Style Sheet (using HTML <link> Tag) is the best method which is used to link the element. Maintaining and re-using the CSS file across different pages is easy and efficient.***

To specify a media type=”text/css” for a Cascading Style Sheet <type> attribute which is used to ignore style sheet types that are not supported in a browser.

**1.2.3 Style specification formats**

CSS stands for Cascading Style Sheets is used to format the layout of a webpage. It controls the layout of multiple web pages all at once. CSS defines that in between elements, how elements are positioned and laid out, what background images or background colors are to be used, different displays for different devices and screen sizes.

The word cascading means that a style applied to a parent element will also apply to all children elements within the parent. So, if we set the color of the body text to blue, then all headings, paragraphs and other text elements within the body will also get the same color unless we specify something else.

**1.2.4 Selector forms and Property value**

CSS selectors are used to select HTML elements based on their element name, id, class, attribute, and more.

**Selector**

Selector represents CSS which elements to style. For example- P, h1 etc

**Property**

Property represents CSS what we have to style/change on that particular element . For Example font-size,line-height, margin,color etc.,

**Value**

Value represents CSS that element value. For example, 24px,black etc

**Types of Selectors**

**1.** **Universal Selectors**

The universal selectors are simply matches the name of any element type. For Example,

\*{

color: #0000FF;

}

This rule renders the content of every element in our document in black.

**2. Element Selectors**

The element selector selects elements based on the element name. For example,

p {

    text-align: center;

    color: red;

}

Here, all p elements will be center-aligned, with a red text color

**3. Descendant Selectors**

**T**o apply a style rule to a particular element only when it lies inside a particular element. For example,

ul em {

color: #000000;

}

Here, , the style rule will apply to the **em** element only when it lies inside the **ul** tag.

**4. Id Selectors**

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

The style rule below will be applied to the HTML element with id=”jk″

#jk {

    color: green;

    text-align: center;

}

**5. Class Selectors**

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

for example,

.jk {

    color: green;

    text-align: center;

}

Here all HTML elements with class=”jk” will be green and center-aligned

**6. Grouping Selectors**

If we have elements with the same style definitions, like

h1 {

    text-align: center;

    color: blue;

}

h2 {

    text-align: center;

    color: blue;

}

p {

    text-align: center;

    color: blue;

}

It will be better to group the selectors, to minimize the code. To group selectors, separate each selector with a comma. Here, we can group the selectors from the code above

h1,h2,p {

    text-align: center;

    color: red;

}

**Example 1.15 CSS examples heading tags**

<html><body>

<h1 style="background-color:DodgerBlue;">Multimedia and Animation</h1>  
<h2 style="color:Tomato;">BCA IV SEM</h2>

<h1 style="text-align:center">This is heading 1</h1>  
<h2 style="text-align:left">This is heading 2</h2>  
<h3 style="text-align:right">This is heading 3</h3>  
<h4 style="text-align:justify">This is heading 4</h4>

</body></html>

**Example 1.16 CSS Examples on <p> tag**

<!DOCTYPE html>

<html>

<head>

<title>Example</title>

<style>

h1

{

display: block;

font-size: 2em;

margin-top: 0.67em;

margin-bottom: 0.67em;

margin-left: 0;

margin-right: 0;

font-weight: bold;

}

p

{

color:red;

font-weight: bold;

}

</style>

<body bgcolor="cyan">

<h1> IV SEM SUBJECTS</h1>

<p>Python Programming

<p> Multimedia and Animation

<p> Operating System Concepts

</body>

</html>

**Example 1.17 CSS Examples - Table**

<html>

<head>

<style>

table, th, td

{

border: 1px solid black;

border-radius: 10px;

}

th, td

{

background-color: #96D4D4;

}

th, td

{

padding-top: 10px;

padding-bottom: 20px;

padding-left: 30px;

padding-right: 40px;

}

table

{

border-spacing: 30px;

}

</style>

</head>

<body bgcolor=yellow>

<table border=2 bgcolor=cyan>

<th colspan="2" bgcolor=red>Student List</th>

<tr><th >sno</th><th>name</th></tr>

<tr><td>1</td><td>Azad</td></tr>

<tr><td>2</td><td>Farook</td></tr>

<tr><td>3</td><td>Jack</td></tr>

<tr><td>4</td><td>Thilak</td></tr>

<tr><td>5</td><td>mahesh</td></tr>

</table>

</body>

</html>

## Example 1.18 CSS Border, padding and margin

The CSS border property defines a border around an HTML element. We can define a border for nearly all HTML elements.The CSS padding property defines a padding (space) between the text and the border. The CSS margin property defines a margin (space) outside the border.

<html>

<head>

<style>

p {

border: 2px solid powderblue;

padding: 30px;

margin: 50px;

}

</style>

</head>

<body>

<p>

A fool does not care whether he understands a thing or not; all he wants to do is to show how clever he is

</body>

</html>

</html>Top of Form

**1.2.5 Font properties**

## CSS Fonts

The CSS font is used to set the font’s content of the HTML element . There are many font properties in CSS which are used to set the font contents.

|  |  |
| --- | --- |
| CSS Property | Meaning |
| font-family | The font-family property specifies the font of an element |
| font-style | The font-style property is used to give designing to any type of text. It can be “normal, italic or oblique”. |
| font-weight | The font-weight property is used to set the weight or thickness of the font being used with the HTML Text. Its value can be “normal, bold, lighter, bolder”. |
| font-variant | The font-variant property is used to converted all lowercase letters into uppercase letters. It can be “normal or small-caps”. |
| font-size | The font-size property in CSS is used to set the font size of text in HTML document. The font-size can be set in different ways like in “pixels, percentage, em or we can set values like small, large” etc. |
| font-stretch | The font-stretch property in CSS is used to set the text wider or narrower. |
| font-kerning | This property is used to control the usage of the Kerning Information that has been stored in the Font |
| [**font-family**](https://www.geeksforgeeks.org/css-font-family-property/) | It is used to set the font type of an HTML element. It holds several font names as a fallback system |

### Example 1.19 Use of CSS color, font-family and font-size properties

<!DOCTYPE html>  
<html>  
<head>  
<style>  
h1 {  
  color: blue;  
  font-family: verdana;  
  font-size: 300%;  
}  
p {  
  color: red;  
  font-family: courier;  
  font-size: 160%;  
}  
</style>  
</head>  
<body>  
<h1>This is a heading</h1>  
<p>This is a paragraph.</p>  
</body>  
</html>

### Example 1.20 Use of font-family and font-size properties

<!DOCTYPE html>

<html>

<head>

<title>font-family property</title>

<style>

.jtk{

font-family: "Times New Roman";

font-weight: bold;

font-size: 40px;

color: #090;

text-align: center;

}

.jtl{

font-family: "Comic Sans MS", cursive, sans-serif;

text-align: center;

}

</style>

</head>

<body>

<div class="jtk">Web Designing</div>

<div class="jtl"> HTML#CSS #JavaScript </div>

</body>

</html>

### Example 1.21 Use of font-weight properties

<html>

<head>

<title>font-weight property</title>

<style>

h1{

font-weight: bold;

font-style: normal;

font-family: "Times New Roman";

font-size: 40px;

color: #090;

text-align: center;

}

p {

font-weight: normal;

text-align: center;

}

</style>

</head>

<body>

<h1> Golden Rule</h1>

<p> Love your neighbour as yourself

</body>

</html>

**1.2.7 List properties**

The **List** in CSS specifies the listing of the contents or items in a particular manner i.e., it can either be organized orderly or unordered way, which helps to make a clean webpage. It can be used to arrange the huge with a variety of content as they are flexible and easy to manage. The default style for the list is borderless. The list can be categorized into two types,

[**Unordered List**](https://www.geeksforgeeks.org/html-ul-tag/)

In unordered lists, the list items are marked with bullets i.e. small black circles by default.

[**Ordered List**](https://www.geeksforgeeks.org/html-ol-tag/)

In ordered lists, the list items are marked with numbers and an alphabet.

We have the following CSS lists properties, which can be used to control the CSS lists.

|  |  |  |
| --- | --- | --- |
| **Property** | **Meaning** | **Examples** |
| [list-style-type](https://www.geeksforgeeks.org/css-list-style-type-property/) | This property is used to specify the appearance (such as disc, character, or custom counter style) of the list item marker | * circle * decimal, eg :1,2,3, etc * decimal-leading-zero,eg :01,02,03,04,etc * lower-roman * upper-roman * lower-alpha, eg: a,b,c, etc * upper-alpha, eg: A, B, C, etc * square |
| [list-style-image](https://www.geeksforgeeks.org/css-list-style-image-property/) | * This property is used to sets an image to be used as the list item marker. Its default value is “none”. | list-style-image: url; |
| [list-style-position](https://www.geeksforgeeks.org/css-list-style-position-property/) | * It specifies the position of the marker box with respect to the principal block box. |  |
| list-style | * This property is used to set the list style. |  |

**Example 1.22 CSS List with the various list-style**

<!DOCTYPE html>

<html>

<head>

<style>

ul.a {

list-style-type: square;

}

ol.c {

list-style-type: lower-alpha;

}

</style>

</head>

<body bgcolor="tomato">

<h1>Fruits</h1>

<p> Unordered lists </p>

<ul class="a">

<li>Apple</li>

<li>Orange</li>

<li>Mango</li>

</ul>

<ul class="b">

<li>Banana</li>

<li>Jack</li>

<li>Papaya</li>

</ul>

<h1>Vegetables </h1>

<p> Ordered Lists </p>

<ol class="c">

<li>Brinjal</li>

<li>Onion</li>

<li>Carrot</li>

</ol>

<ol class="d">

<li>Potato</li>

<li>Bitter Gaurd</li>

<li>Bottle Gaurd</li>

</ol>

</body>

</html>

**1.2.8 CSS Color**

CSS Color property is used to set the color of HTML elements. This property is used to set font color, background color etc.

Color of an element can be defined in the following ways

* Built-In Color
* RGB Format
* RGBA Format
* Hexadecimal Notation
* HSL
* HSLA

**Built-In Color**

 These are a set of predefined colors which are used by its name. For example: red, blue, green etc.

h1 {

color: color-name;

}

**RGB Format**

The RGB(Red, Green, Blue) format is used to define the color of an HTML element by specifying the R, G, B values range between 0 to 255. For example: RGB value of Red color is (255, 0, 0), Green color is (0, 255, 0), Blue color is (0, 0, 255) etc.  
**Syntax:**

h1 {

color: rgb(R, G, B);

}

**RGBA Format**

 The RGBA format is similar to the RGB, but the difference is RGBA contains A (Alpha) which specify the transparency of elements. The value of alpha lies between 0.0 to 1.0 where 0.0. represents fully transparent and 1.0 represents not transparent.  
**Syntax:**

h1 {

color:rgba(R, G, B, A);

}

**Hexadecimal Notation**

 The hexadecimal notation begins with # symbol followed by 6 characters each range from 0 to F. For example: Red #FF0000, Green #00FF00, Blue #0000FF etc.  
**Syntax:**

h1 {

color:#(0-F)(0-F)(0-F)(0-F)(0-F)(0-F);

}

**HSL**

HSL stands for Hue, Saturation, and Lightness respectively. This format uses the cylindrical coordinate system.

**Hue:** Hue is the degree of the color wheel. Its value lies between 0 to 360 where 0 represents red, 120 represents green and 240 represents blue color.

**Saturation:** It takes percentage value, where 100% represents completely saturated, while 0% represents completely unsaturated (gray).

**Lightness:** It takes percentage value, where 100% represents white, while 0% represents black.

**Syntax:**

h1 {

color:hsl(H, S, L);

}

**HSLA**

**HSLA** is similar to HSL property, but the difference is HSLA contains A (Alpha) which specify the transparency of elements. The value of alpha lies between 0.0 to 1.0 where 0.0. represents fully transparent and 1.0 represents not transparent.  
**Syntax:**

h1 {

color:hsla(H, S, L, A);

}

**1.2.9 Alignment of text**

The text-align property is used to set the horizontal alignment of a text. A text can be left or right aligned, centered, or justified. The CSS Text Alignment/Direction Properties are used to set text alignment and directions.

|  |  |
| --- | --- |
| **Property** | **Description** |
| [direction](https://www.w3schools.com/cssref/pr_text_direction.asp) | Specifies the text direction/writing direction |
| [text-align](https://www.w3schools.com/cssref/pr_text_text-align.asp) | Specifies the horizontal alignment of text |
| [text-align-last](https://www.w3schools.com/cssref/css3_pr_text-align-last.asp) | Specifies how to align the last line of a text |
| [unicode-bidi](https://www.w3schools.com/cssref/pr_text_unicode-bidi.asp) | Used together with the [direction](https://www.w3schools.com/cssref/pr_text_direction.asp) property to set or return whether the text should be overridden to support multiple languages in the same document |
| [vertical-align](https://www.w3schools.com/cssref/pr_pos_vertical-align.asp) | Sets the vertical alignment of an element |
|  |  |

**Example 1.23 CSS Text allignment**

<html>

<head>

<style>

h1 { text-align: center; }

h2 { text-align: left; }

h3 { text-align: right; }

</style>

</head>

<body bgcolor="paleyellow">

<h1> Do not forget your friends or your father's friends</h1>

<h2> Do not forget your friends or your father's friends</h2>

<h3> Do not forget your friends or your father's friends</h3>

</body>

<html>

**Example 1.24 CSS image**

<!DOCTYPE html>

<html>

<head>

<style>

img {

width: 100%;

}

</style>

</head>

<body>

<img src="html5.png" alt="HTML5 Icon" width="128" height="128">

<img src="html5.png" alt="HTML5 Icon" style="width:128px;height:128px;">

</body>

</html>

**CSS background**

Background image can be added to an HTML element, by using background-image property. To avoid the background image from repeating itself, we need to set ***background-repeat*** property to ***no-repeat.***

If we want the background image to cover the entire element, we can set the **background-size** property to ***cover***. Also, to make sure the entire element is always covered, set the ***background-attachment*** property to ***fixed***

This way, the background image will cover the entire element, with no stretching (the image will keep its original proportions)

<p style="background-image: url('flower.jpg');">

**To Specify the background image in the <style> element**

<style>  
 p {  
   background-image: url('img\_girl.jpg');  
 }  
</style>

**to Add a background image for the entire page**

<style>  
 body {

  background-image: url('img\_girl.jpg');  
 }  
</style>

To avoid the background image from repeating itself, set the **background-repeat** property to ***no-repeat***.

<style>  
body {  
  background-image: url('example\_img\_girl.jpg');  
  background-repeat: no-repeat;  
}  
</style>

## Background Cover

We can cover the entire element with background image, by setting ***background-size*** property to ***cover.*** By this way, the background image will cover the entire element, with no stretching (the image will keep its original proportions)

<style>  
body {  
  background-image: url('baby.jpg');  
  background-repeat: no-repeat;  
  background-attachment: fixed;  
  background-size: cover;  
}  
</style>

We can stretch the background image to fit the entire element, by setting the ***background-size*** property to ***100% 100%***

<style>  
body {  
  background-image: url('baby.jpg');  
  background-repeat: no-repeat;  
  background-attachment: fixed;  
  background-size: 100% 100%;  
}  
</style>

**1.3 JavaScript**

JavaScript is the programming language of the Web. [**HTML**](https://www.w3schools.com/html/default.asp)  is used to define the content of web pages. [**CSS**](https://www.w3schools.com/css/default.asp) is used to specify the layout of web pages. **JavaScript** is used to program the behavior of web pages.

JavaScript and [Java](https://www.w3schools.com/java/default.asp) are completely different languages, both in concept and design. JavaScript was invented by Brendan Eich in 1995, and became an ECMA standard in 1997.

In HTML, JavaScript code is inserted between <script> and </script> tags. For Example,

<script>  
document.write( "My First JavaScript");  
</script>

We can place any number of scripts in an HTML document. Scripts can be placed in the <body>, or in the <head> section of an HTML page, or in both.

### Example 1.25 JavaScript examples - JavaScript in <head>

<!DOCTYPE html>  
<html>  
<head>  
<script>  
function myFunction() {  
  document.getElementById("demo").innerHTML = "Paragraph changed.";  
}  
</script>  
</head>  
<body>

<h2>Demo JavaScript in Head</h2>  
  
<p id="demo">A Paragraph</p>  
<button type="button" onclick="myFunction()">Try it</button>

</body>  
</html>

Example 1. JavaScript examples -JavaScript in <body>

<!DOCTYPE html>  
<html>  
<body>  
<h2>Demo JavaScript in Body</h2>  
<p id="demo">A Paragraph</p>  
<button type="button" onclick="myFunction()">Try it</button>  
<script>  
function myFunction()

{  
  document.getElementById("demo").innerHTML = "Paragraph changed.";  
}  
</script>  
</body>  
</html>

Placing scripts at the bottom of the <body> element improves the display speed, because script interpretation slows down the display.

## External JavaScript

Scripts can also be placed in external files and can be used in any web pages. External scripts are useful, when the same code is used in many different web pages. JavaScript files have the file extension **.js**.

To use an external script, put the name of the script file in the src (source) attribute of a <script> tag. To add several script files to one page  - use several script tags:

**myfile.js**

document.write("Welcome to Java Script");

### Example 1.26 Webpage with external script

<html>

<head>

<script src="myfile.js"></script>

</script>

<body>

<font size="20">

Computer Multimedia and Animation

</font>

</body>

</html>

## External JavaScript Advantages

Placing scripts in external files has some advantages:

* It separates HTML and code
* It makes HTML and JavaScript easier to read and maintain
* Cached JavaScript files can speed up page loads

**1.3.1 Object orientation and JavaScript**

JavaScript is an Object Oriented Programming language. JavaScript is an excellent language to write object oriented web applications.

It can support OOP because it supports inheritance through prototyping as well as properties and methods

[**Object**](https://www.geeksforgeeks.org/objects-in-javascript/)

An Object is a **unique** entity that contains **properties** and **methods**. For example “a car” is a real-life Object, which has some characteristics like color, type, model, and horsepower and performs certain actions like driving. The characteristics of an Object are called **Properties** and actions are called **methods**.

An Object is an **instance** of a class. Objects are everywhere in JavaScript, almost every element is an Object whether it is a function, array, or string.  The object can be created in two ways in JavaScript

1. Object Literal
2. Object Constructor

[**Classes**](https://www.geeksforgeeks.org/javascript-classes/)

Classes are **blueprints** of an Object. A class can have many Objects because the class is a **template** while Objects are **instances** of the class or the concrete implementation.   
 Unlike other Object Oriented languages there are **no classes in JavaScript** we have only Object. To be more precise, JavaScript is a prototype-based Object Oriented Language, which means it doesn’t have classes, rather it defines behaviors using a constructor function and then reuses it using the prototype.

**Abstraction**

Abstraction means displaying only essential information and hiding the details. Data abstraction refers to providing only essential information about the data to the outside world, hiding the background details or implementation.

**Encapsulation**

The process of **wrapping properties and functions**within a **single unit** is known as encapsulation.

**Inheritance**

It is a concept in which some properties and methods of an Object are being used by another Object. Unlike most of the OOP languages where classes inherit classes, JavaScript Objects inherit Objects i.e. certain features (property and methods) of one object can be reused by other Objects.

**Polymorphism**

Polymorphism means the same function with different signatures is called many times. In real life, for example, a boy at the same time may be a student, a class monitor, etc., Polymorphism can be achieved by method overriding and method overloading

JavaScript is best known for web page development but it is also used in a variety of non-browser environments.

**Example 1.27 Javascript - object example**

### <html>

### <body bgcolor=pink>

### <script>

### var person = new Object();

### person.name = "Jeeva";

### person.age = 23;

### person.job = "Network Engineer";

### alert(person.name);

### </script>

### </body>

### </html>

### Object Properties in JavaScript

Object properties are variables used within the object's methods but can also be globally visible variables used throughout the page. The syntax for including any property to an object is

***object\_name. property=value***

For example,

Person.age=23

**1.3.2 General syntactic characteristics**

JavaScript scripts will be embedded in HTML documents either directly, as in  
<script type = "text/javascript">  
-- JavaScript script code -  
</script>

Or indirectly, as a file specified in the **src** attribute of **<script>,** as in  
<script type = "text/javascript"  
        src = "myscript.js">  
</script>

**Language Basics**

* Identifier begin with a letter or underscore, followed by any number of letters, underscores, and digits
* Java Script is case sensitive
* It has [25 reserved words](https://www.loc-cs.org/~chu/ITEC315/ch4/reserved_words.pdf)
* for Comments both // and /\* ... \*/ can be used
* We should put each statement on its own line whenever possible and terminate each statement with a semicolon.
* We can omit the semicolon between two statements if they are written on separate lines.
* We can omit a semicolon at the end of a program or if the next token in the program is a closing curly brace }.

**1.3.3 Primitives, operations, and expressions**

**JavaScript Identifiers / Names**

Identifiers are JavaScript names. Identifiers are used to name variables and keywords, and functions. The rules for legal names are the same in most programming languages.

A JavaScript name must begin with:

* A letter (A-Z or a-z)
* A dollar sign ($)
* Or an underscore (\_)

Subsequent characters may be letters, digits, underscores, or dollar signs.

**Operators**

Operators are used to do some operations. There are different types of JavaScript operators, such as

* Arithmetic Operators
* Assignment Operators
* Comparison Operators
* String Operators
* Logical Operators
* Bitwise Operators

|  |  |
| --- | --- |
| **Arithmetic Operator** | **Description** |
| + | Addition |
| - | Subtraction |
| \* | Multiplication |
| \*\* | Exponentiation ([ES2016](https://www.w3schools.com/js/js_2016.asp)) |
| / | Division |
| % | Modulus (Division Remainder) |
| ++ | Increment |
| -- | Decrement |

|  |  |
| --- | --- |
| **Relational Operator** | **Description** |
| == | equal to |
| === | equal value and equal type |
| != | not equal |
| !== | not equal value or not equal type |
| > | greater than |
| < | less than |
| >= | greater than or equal to |
| <= | less than or equal to |
| ?: | ternary operator |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Logical Operator** | **Description** | | | |
| && | logical and | | | |
| || | logical or | | | |
| ! | logical not | | | |
|  |  | | | |
|  |  | | | |
| **Bitwise Operator** | **Description** | Example | Same as | Result | | Decimal |
| & | AND | 5 & 1 | 0101 & 0001 | 0001 | | 1 |
| | | OR | 5 | 1 | 0101 | 0001 | 0101 | | 5 |
| ~ | NOT | ~ 5 | ~0101 | 1010 | | 10 |
| ^ | XOR | 5 ^ 1 | 0101 ^ 0001 | 0100 | | 4 |
| << | left shift | 5 << 1 | 0101 << 1 | 1010 | | 10 |
| >> | right shift | 5 >> 1 | 0101 >> 1 | 0010 | | 2 |
| >>> | unsigned right shift | 5 >>> 1 | 0101 >>> 1 | 0010 | | 2 |

**Expressions**

Expression is the combination of operators and operands. If the expression involves arithmetic operators, then it is called arithmetic expression. If the expression involves logical operators, then it is called logical expression.

for example

C=A+23 is an arithmetic expression

A>B is relational expression etc.,

**1.3.4 Screen output and Keyboard input**

**1.3.4.1 Screen Output**

In JavaScript, Output can be displayed in following ways,

* Writing into an HTML element, using innerHTML.
* Writing into the HTML output using document.write().It is used for testing purpose.
* Writing into an alert box, using window.alert().It displays the content using an alert box.
* Writing into the browser console, using console.log().It is used for debugging purposes.

**innerHTML**

To access an HTML element, JavaScript can use the document.getElementById(id) method.

The id attribute defines the HTML element. The innerHTML property defines the HTML content.

**Example 1.28 Java Script displaying data using innerHTML**

<html>

<body>

<h1>JavaScript</h1>

<h2>

JavaScript Display Possibilities

Using innerHTML

</h2>

<p id="jk"></p>

<!-- Script to use innerHTML -->

<script>

document.getElementById("jk").innerHTML

= 10 \* 5;

</script>

</body>

</html>

## document.write()

For testing purposes, it is convenient to use document.write() method

**Example 1.29 Java Script displaying data using document.write**

<html>

<body>

<script>

document.write(5 + 6);

</script>

</body>

</html>

**Example 1.30 Java Script displaying data using document.write()**

<html>  
<body>  
<button type="button" onclick="document.write(5 + 6)">click here</button>  
</body>  
</html>

## window.alert()

We can use an alert box to display data.

**Example 1.31 Java Script displaying data using window.alert()**

<html>  
<body>  
  
<script>  
window.alert("Hi… IV SEM BCA !!!");

alert("BYE BYE");  
</script>  
</body>  
</html>

We can skip the window keyword. In JavaScript, the window object is the global scope object. This means that variables, properties, and methods by default belong to the window object. This also means that specifying the window keyword is optional.

## console.log()

## For debugging purposes, we can call the console.log() method in the browser to display data.

**Example 1.32 Java Script displaying data using window.alert()**

<html>  
<body>  
<script>  
console.log(5 + 6);  
</script>  
</body>  
</html>

**1.3.4.2 Keyboard Input**

[window.prompt()](https://www.geeksforgeeks.org/javascript-window-prompt-method/)  it Allows to take input from user.

**Example 1.33 Java Script getting input from user**

<html>

<body bgcolor=paleyellow>

<script>

s= window.prompt("Enter your Name");

window.alert("Your name is "+s);

</script>

</body>

</html>