**HTML:**

* HTML stands for Hyper Text Markup Language.
* A markup language is a computer language that is used to apply layout and formatting conventions to a text document. Markup language makes text more interactive and dynamic. It can turn text into images, tables, links, etc.
* HTML is used to create web pages and web applications.
* HTML is widely used language on the web.
* We can create a static website by HTML.
* A web page is a document which is commonly written in HTML and translated by a web browser. A web page can be identified by entering an URL. A Web page can be of the static or dynamic type.
* Technically, HTML is a Markup language rather than a programming language.

<!DOCTYPE**>**

**<html>**

**<head>**

**<title>**Web page title**</title>**

**</head>**

**<body>**

**<h1>**Write Your First Heading**</h1>**

**<p>**Write Your First Paragraph.**</p>**

**</body>**

**</html>**

## Description

**<!DOCTYPE>:** It defines the document type or it instruct the browser about the version of HTML.

**<html >** :This tag informs the browser that it is an HTML document. Text between html tag describes the web document. It is a container for all other elements of HTML except <!DOCTYPE>

**<head>:** It should be the first element inside the <html> element, which contains the metadata(information about the document). It must be closed before the body tag opens.

**<title>:** As its name suggested, it is used to add title of that HTML page which appears at the top of the browser window. It must be placed inside the head tag and should close immediately. (Optional)

**<body>**: Text between body tag describes the body content of the page that is visible to the end user. This tag contains the main content of the HTML document.

**<h1>** : Text between <h1> tag describes the first level heading of the webpage.

**HTML5 :** HTML5 is the newest version of HyperText Markup language. The first draft of this version was announced in January 2008. There are two major organizations one is W3C (World Wide Web Consortium), and another one is WHATWG( Web Hypertext Application Technology Working Group) which are involved in the development of HTML 5 version, and still, it is under development.

Features of HTML

1) It is a very **easy and simple language**. It can be easily understood and modified.

2) It is very easy to make an **effective presentation** with HTML because it has a lot of formatting tags.

3) It is a **markup language**, so it provides a flexible way to design web pages along with the text.

4) It facilitates programmers to add a **link** on the web pages (by html anchor tag), so it enhances the interest of browsing of the user.

5) It is **platform-independent** because it can be displayed on any platform like Windows, Linux, and Macintosh, etc.

6) It facilitates the programmer to add **Graphics, Videos, and Sound** to the web pages which makes it more attractive and interactive.

7) HTML is a case-insensitive language, which means we can use tags either in lower-case or upper-case.

# HTML text Editors

* An HTML file is a text file, so to create an HTML file we can use any text editors.
* Text editors are the programs which allow editing in a written text, hence to create a web page we need to write our code in some text editor.
* There are various types of text editors available which you can directly download, but for a beginner, the best text editor is Notepad (Windows) or TextEdit (Mac).
* After learning the basics, you can easily use other professional text editors which are, **Notepad++, Sublime Text, & Microsoft Visual Studio Code etc**.

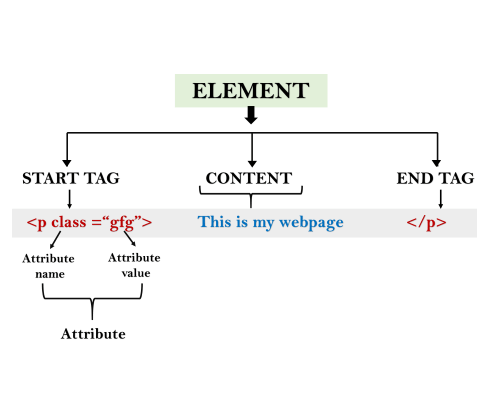
# Building blocks of HTML

An HTML document consist of its basic building blocks:

* **Tags:** An HTML tag surrounds the content and apply meaning to it. It is written between < and > brackets.
* **Attribute:** An attribute in HTML provides extra information about the element, and it is applied within the start tag. An HTML attribute contains two fields: name & value.

**<tag** name attribute\_name= "attr\_value"**>** content **</** **tag** name**>**

* **Elements:** An HTML element is an individual component of an HTML file. In an HTML file, everything written within tags are termed as HTML elements.



<!DOCTYPE html**>**

**<html>**

**<head>**

**<title>**The basic building blocks of HTML**</title>**

**</head>**

**<body>**

**<h2>**The building blocks**</h2>**

**<p>**This is a paragraph tag**</p>**

**<p** style="color: red"**>**The style is attribute of paragraph tag**</p>**

**<span>**The element contains tag, attribute and content**</span>**

**</body>**

**</html>**

# HTML Tags

HTML tags are like keywords which defines that how web browser will format and display the content. With the help of tags, a web browser can distinguish between an HTML content and a simple content. HTML tags contain three main parts: opening tag, content and closing tag. But some HTML tags are unclosed tags.

When a web browser reads an HTML document, browser reads it from top to bottom and left to right. HTML tags are used to create HTML documents and render their properties. Each HTML tags have different properties.

An HTML file must have some essential tags so that web browser can differentiate between a simple text and HTML text. You can use as many tags you want as per your code requirement.

* All HTML tags must enclosed within < > these brackets.
* Every tag in HTML perform different tasks.
* If you have used an open tag <tag>, then you must use a close tag </tag> (except some tags)

## Syntax

<tag> content </tag>

## HTML Tag Examples

<p> Paragraph Tag </p>

## <h2> Heading Tag </h2>

<b> **Bold Tag** </b>

<i> *Italic Tag* </i>

<u> Underline Tag</u>

## Unclosed HTML Tags

Some HTML tags are not closed, for example br and hr.

**<br> Tag**: br stands for break line, it breaks the line of the code.

**<hr> Tag**: hr stands for Horizontal Rule. This tag is used to put a line across the webpage.

## HTML Meta Tags

DOCTYPE, title, link, meta and style

## HTML Text Tags

<p>, <h1>, <h2>, <h3>, <h4>, <h5>, <h6>, <strong>, <em>, <abbr>, <acronym>, <address>, <bdo>, <blockquote>, <cite>, <q>, <code>, <ins>, <del>, <dfn>, <kbd>, <pre>, <samp>, <var> and <br>

## HTML Link Tags

<a> and <base>

## HTML Image and Object Tags

<img>, <area>, <map>, <param> and <object>

## HTML List Tags

<ul>, <ol>, <li>, <dl>, <dt> and <dd>

|  |  |
| --- | --- |
| **Tag name** | **Description** |
| [<!-- -->](https://www.javatpoint.com/html-comments) | This tag is used to apply comment in an HTML document. |
| [<!DOCTYPE>](https://www.javatpoint.com/doctype-html) | This tag is used to specify the version of HTML |
| A | |
| [<a>](https://www.javatpoint.com/html-anchor) | It is termed as anchor tag and it creates a hyperlink or link. |
| [<abbr>](https://www.javatpoint.com/html-abbr-tag) | It defines an abbreviation for a phrase or longer word. |
| [<acronym>](https://www.javatpoint.com/html-acronym-tag) | It defines acronym for a word. **(Not supported in HTML5)** |
| [<address>](https://www.javatpoint.com/html-address-tag) | It defines the author's contact information of the HTML article |
| [<applet>](https://www.javatpoint.com/html-applet-tag) | It defines an embedded Java applet. **(Not supported in HTML5)** |
| [<area>](https://www.javatpoint.com/html-area-tag) | It defines the area of an image map. |
| [<article>](https://www.javatpoint.com/html-article-tag)  HTML Tags List | It defines the self-contained content. |
| [<aside>](https://www.javatpoint.com/html-aside-tag)  HTML Tags List | It defines content aside from main content. Mainly represented as sidebar. |
| [<audio>](https://www.javatpoint.com/html-audio)  HTML Tags List | It is used to embed sound content in HTML document. |
| B | |
| [<b>](https://www.javatpoint.com/html-bold-tag) | It is used to make a text bold. |
| [<base>](https://www.javatpoint.com/html-base-tag) | This tag defines the base URL for all relative URL within the document. |
| [<basefont>](https://www.javatpoint.com/html-basefont-tag) | This tag is used to set default font, size and color for all elements of document.  **(Not supported in HTML5)** |
| [<bdi>](https://www.javatpoint.com/html-bdi-tag)  HTML Tags List | This tag is used to provide isolation for that part of text which may be formatted  in different directions from its surrounding text. |
| [<bdo>](https://www.javatpoint.com/html-bdo-tag) | It is used to override the current text direction. |
| [<big>](https://www.javatpoint.com/html-big-tag) | This tag is used to make font size one level larger than its surrounding content.  **(Not supported in HTML5)** |
| [<blockquote>](https://www.javatpoint.com/html-blockquote-tag) | It is used to define a content which is taken from another source. |
| [<body>](https://www.javatpoint.com/html-body-tag) | It is used to define the body section of an HTML document. |
| [<br>](https://www.javatpoint.com/html-br-tag) | It is used to apply single line break. |
| [<button>](https://www.javatpoint.com/html-button-tag) | It is used to represent a clickable button |
| C | |
| [<canvas>](https://www.javatpoint.com/html-canvas)  HTML Tags List | It is used to provide a graphics space within a web document. |
| [<caption>](https://www.javatpoint.com/html-caption-tag) | It is used to define a caption for a table. |
| [<center>](https://www.javatpoint.com/html-center-tag) | It is used to align the content in center. **(Not supported in HTML5)** |
| [<cite>](https://www.javatpoint.com/html-cite-tag) | It is used to define the title of the work, book, website, etc. |
| [<code>](https://www.javatpoint.com/html-code-tag) | It is used to display a part of programming code in an HTML document. |
| [<col>](https://www.javatpoint.com/html-col-tag) | It defines a column within a table which represent common properties of columns  and used with the <colgroup> element. |
| [<colgroup>](https://www.javatpoint.com/html-colgroup-tag) | It is used to define group of columns in a table. |
| D | |
| [<data>](https://www.javatpoint.com/html-data-tag)  HTML Tags List | It is used to link the content with the machine-readable translation. |
| [<datalist>](https://www.javatpoint.com/html-datalist-tag)  HTML Tags List | It is used to provide a predefined list for input option. |
| [<dd>](https://www.javatpoint.com/html-description-list) | It is used to provide definition/description of a term in description list. |
| [<del>](https://www.javatpoint.com/html-del-tag) | It defines a text which has been deleted from the document. |
| [<details>](https://www.javatpoint.com/html-details-tag)  HTML Tags List | It defines additional details which user can either view or hide. |
| [<dfn>](https://www.javatpoint.com/html-dfn-tag) | It is used to indicate a term which is defined within a sentence/phrase. |
| [<dialog>](https://www.javatpoint.com/html-dialog-tag)  HTML Tags List | It defines a dialog box or other interactive components. |
| [<dir>](https://www.javatpoint.com/html-dir-tag) | It is used as container for directory list of files. **(Not supported in HTML5)** |
| [<div>](https://www.javatpoint.com/html-div-tag) | It defines a division or section within HTML document. |
| [<dl>](https://www.javatpoint.com/html-description-list) | It is sued to define a description list. |
| [<dt>](https://www.javatpoint.com/html-description-list) | It is used to define a term in description list. |
| E | |
| [<em>](https://www.javatpoint.com/html-em-tag) | It is used to emphasis the content applied within this element. |
| [<embed>](https://www.javatpoint.com/html-embed-tag)  HTML Tags List | It is used as embedded container for external file/application/media, etc. |
| F | |
| [<fieldset>](https://www.javatpoint.com/html-fieldset-tag) | It is used to group related elements/labels within a web form. |
| [<figcaption>](https://www.javatpoint.com/html-figcaption-tag)  HTML Tags List | It is used to add a caption or explanation for the <figure> element. |
| [<figure>](https://www.javatpoint.com/html-figure-tag)  HTML Tags List | It is used to define the self-contained content, and s mostly refer as single unit. |
| [<font>](https://www.javatpoint.com/html-font-tag) | It defines the font, size, color, and face for the content. **(Not supported in HTML5)** |
| [<footer>](https://www.javatpoint.com/html-footer-tag)  HTML Tags List | It defines the footer section of a webpage. |
| [<form>](https://www.javatpoint.com/html-form) | It is used to define an HTML form. |
| [<frame>](https://www.javatpoint.com/html-frame-tag) | It defines a particular area of webpage which can contain another HTML file.  **(Not supported in HTML5)** |
| [<frameset>](https://www.javatpoint.com/html-frameset-tag) | It defines group of Frames. **(Not supported in HTML5)** |
| H | |
| [<h1> to <h6>](https://www.javatpoint.com/html-heading) | It defines headings for an HTML document from level 1 to level 6. |
| [<head>](https://www.javatpoint.com/html-head) | It defines the head section of an HTML document. |
| [<header>](https://www.javatpoint.com/html-header-tag)  HTML Tags List | It defines the header of a section or webpage. |
| [<hr>](https://www.javatpoint.com/html-hr-tag) | It is used to apply thematic break between paragraph-level elements. |
| [<html>](https://www.javatpoint.com/html-html-tag) | It represents root of an HTML document. |
| I | |
| [<i>](https://www.javatpoint.com/html-i-tag) | It is used to represent a text in some different voice. |
| [<iframe>](https://www.javatpoint.com/html-iframes) | It defines an inline frame which can embed other content. |
| [<img>](https://www.javatpoint.com/html-image) | It is used to insert an image within an HTML document. |
| [<input>](https://www.javatpoint.com/html-input-tag) | It defines an input field within an HTML form. |
| [<ins>](https://www.javatpoint.com/html-ins-tag) | It represent text that has been inserted within an HTML document. |
| [<isindex>](https://www.javatpoint.com/html-isindex-tag) | It is used to display search string for current document. **(Not supported in HTML5)** |
| K | |
| [<kbd>](https://www.javatpoint.com/html-kbd-tag) | It is used to define keyboard input. |
| L | |
| [<label>](https://www.javatpoint.com/html-label-tag) | It defines a text label for the input field of form. |
| [<legend>](https://www.javatpoint.com/html-legend-tag) | It defines a caption for content of <fieldset> |
| [<li>](https://www.javatpoint.com/html-lists) | It is used to represent items in list. |
| [<link>](https://www.javatpoint.com/html-link-tag) | It represents a relationship between current document and an external resource. |
| M | |
| [<main>](https://www.javatpoint.com/html-main-tag)  HTML Tags List | It represents the main content of an HTML document. |
| [<map>](https://www.javatpoint.com/html-map-tag) | It defines an image map with active areas. |
| [<mark>](https://www.javatpoint.com/html-mark-tag)  HTML Tags List | It represents a highlighted text. |
| [<marquee>](https://www.javatpoint.com/marquee-html) | It is used to insert the scrolling text or an image either horizontally or vertically.  **(Not supported in HTML5)** |
| [<menu>](https://www.javatpoint.com/html-menu-tag) | It is used for creating a menu list of commands. |
| [<meta>](https://www.javatpoint.com/html-meta-tag) | It defines metadata of an HTML document. |
| [<meter>](https://www.javatpoint.com/html-meter-tag)  HTML Tags List | It defines scalar measurement with known range or fractional value. |
| N | |
| [<nav>](https://www.javatpoint.com/html-nav-tag)  HTML Tags List | It represents section of page to represent navigation links. |
| [<noframes>](https://www.javatpoint.com/html-noframes-tag) | It provides alternate content to represent in browser which does not support  the <frame> elements. **(Not supported in HTML5)** |
| [<noscript>](https://www.javatpoint.com/html-noscript-tag) | It provides an alternative content if a script type is not supported in browser. |
| O | |
| [<object>](https://www.javatpoint.com/html-object-tag) | It is used to embed an object in HTML file. |
| [<ol>](https://www.javatpoint.com/html-ordered-list) | It defines an ordered list of items. |
| [<optgroup>](https://www.javatpoint.com/html-optgroup-tag) | It is used to group the options of a drop-down list. |
| [<option>](https://www.javatpoint.com/html-option-tag) | It is used to define options or items in a drop-down list. |
| [<output>](https://www.javatpoint.com/html-output-tag)  HTML Tags List | It is used as container element which can show result of a calculation. |
| P | |
| [<p>](https://www.javatpoint.com/html-paragraph) | It represents a paragraph in an HTML document. |
| [<param>](https://www.javatpoint.com/html-param-tag) | It defines parameter for an <object> element |
| [<picture>](https://www.javatpoint.com/html-picture-tag)  HTML Tags List | It defines more than one source element and one image element. |
| [<pre>](https://www.javatpoint.com/html-pre-tag) | It defines preformatted text in an HTML document. |
| [<progress>](https://www.javatpoint.com/html-progress-tag)  HTML Tags List | It defines the progress of a task within HTML document. |
| *Q* | |
| [<q>](https://www.javatpoint.com/html-quotes) | It defines short inline quotation. |
| *R* | |
| [<rp>](https://www.javatpoint.com/html-rp-tag)  HTML Tags List | It defines an alternative content if browser does not supports ruby annotations. |
| [<rt>](https://www.javatpoint.com/html-rt-tag) | It defines explanations and pronunciations in ruby annotations. |
| [<ruby>](https://www.javatpoint.com/html-ruby-tag) | It is used to represent ruby annotations. |
| S | |
| [<s>](https://www.javatpoint.com/html-s-tag) | It render text which is no longer correct or relevant. |
| [<samp>](https://www.javatpoint.com/html-samp-tag) | It is used to represent sample output of a computer program. |
| [<script>](https://www.javatpoint.com/html-script-tag) | It is used to declare the JavaScript within HTML document. |
| [<section>](https://www.javatpoint.com/html-section-tag)  HTML Tags List | It defines a generic section for a document. |
| [<select>](https://www.javatpoint.com/html-select-tag) | It represents a control which provides a menu of options. |
| [<small>](https://www.javatpoint.com/html-small-tag) | It is used to make text font one size smaller than document?s base font size. |
| [<source>>[HTML Tags List](https://www.javatpoint.com/html-source-tag)](https://www.javatpoint.com/html-source-tag) | It defines multiple media recourses for different media element such as  <picture>, <video>, and <audio> element. |
| [<span>](https://www.javatpoint.com/html-span-tag) | It is used for styling and grouping inline. |
| [<strike>](https://www.javatpoint.com/html-strike-tag) | It is used to render strike through the text. **(Not supported in HTML5)** |
| [<strong>](https://www.javatpoint.com/html-strong-tag) | It is used to define important text. |
| [<style>](https://www.javatpoint.com/html-style) | It is used to contain style information for an HTML document. |
| [<sub>](https://www.javatpoint.com/html-sub-tag) | It defines a text which displays as a subscript text. |
| [<summary>](https://www.javatpoint.com/html-summary-tag)  HTML Tags List | It defines summary which can be used with <details> tag. |
| [<sup>](https://www.javatpoint.com/html-sup-tag) | It defines a text which represent as superscript text. |
| [<svg>](https://www.javatpoint.com/html-svg) | It is used as container of SVG (Scalable Vector Graphics). |
| T | |
| [<table>](https://www.javatpoint.com/html-table) | It is used to present data in tabular form or to create a table within HTML document. |
| [<tbody>](https://www.javatpoint.com/html-tbody-tag) | It represents the body content of an HTML table and used along with <thead>  and <tfoot>. |
| [<td>](https://www.javatpoint.com/html-td-tag) | It is used to define cells of an HTML table which contains table data |
| [<template>](https://www.javatpoint.com/html-template-tag) | It is used to contain the client side content which will not display at time of page  load and may render later using JavaScript. |
| [<textarea>](https://www.javatpoint.com/html-textarea) | It is used to define multiple line input, such as comment, feedback, and review, etc. |
| [<tfoot>](https://www.javatpoint.com/html-tfoot-tag) | It defines the footer content of an HTML table. |
| [<th>](https://www.javatpoint.com/html-th-tag) | It defines the head cell of an HTML table. |
| [<thead>](https://www.javatpoint.com/html-thead-tag) | It defines the header of an HTML table. It is used along with <tbody> and <tfoot>  tags. |
| [<time>](https://www.javatpoint.com/html-time-tag)  HTML Tags List | It is used to define data/time within an HTML document. |
| [<title>](https://www.javatpoint.com/html-title) | It defines the title or name of an HTML document. |
| [<tr>](https://www.javatpoint.com/html-tr-tag) | It defines the row cells in an HTML table |
| [<track>](https://www.javatpoint.com/html-track-tag) | It is used to define text tracks for <audio> and <video> elements. |
| [<tt>](https://www.javatpoint.com/html-tt-tag) | It is used to define teletype text. **(Not supported in HTML5)** |
| U | |
| [<u>](https://www.javatpoint.com/html-u-tag) | It is used to render enclosed text with an underline. |
| [<ul>](https://www.javatpoint.com/html-unordered-list) | It defines unordered list of items. |
| V | |
| [<var>](https://www.javatpoint.com/html-var-list) | It defines variable name used in mathematical or programming context. |
| [<video>](https://www.javatpoint.com/html-video)  HTML Tags List | It is used to embed a video content with an HTML document |
| W | |
| [<wbr>](https://www.javatpoint.com/html-wbr-tag)  HTML Tags List | It defines a position within text where break line is possible. |

## HTML Table Tags

table, tr, td, th, tbody, thead, tfoot, col, colgroup and caption

## HTML Form Tags

form, input, textarea, select, option, optgroup, button, label, fieldset and legend

## HTML Scripting Tags

script and noscript

# HTML Attribute

* 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 behaviour of that element.
* Attributes should always be applied with start tag.
* The Attribute should always be applied with its name and value pair.
* The Attributes name and values are case sensitive, and it is recommended by W3C that it should be written in Lowercase only.
* You can add multiple attributes in one HTML element, but need to give space between two attributes.

## Syntax

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

## Example

<!DOCTYPE html**>**

**<html>**

**<head>**

**</head>**

**<body>**

**<h1>** This is Style attribute**</h1>**

**<p** style="height: 50px; color: blue"**>**Adds style property in element**</p>**

**<p** style="color: red"**>**It will change the color of content**</p>**

**</body>**

**</html>**

## The title attribute in HTML

**Description:** The title attribute is used as text tooltip in most of the browsers. It display its text when user move the cursor over a link or any text. You can use it with any text or link to show the description about that link or text. In our example, we are taking this with paragraph tag and heading tag.

## Example

<!DOCTYPE html**>**

**<html>**

**<head>**

**</head>**

**<body>**

**<h1** title="This is heading tag"**>**Example of title attribute**</h1>**

**<p** title="This is paragraph tag"**>**Move the cursor over the heading and paragraph, and you will see a description as a tooltip**</p>**

**</body>**

**</html>**

## The href attribute in HTML

**Description:** The href attribute is the main attribute of <a> anchor tag. This attribute gives the link address which is specified in that link. **The href attribute provides the hyperlink, and if it is blank, then it will remain in same page**.

<!DOCTYPE html>

<html>

<head> </head>

<body>

<h1>Display of href attribute</h1>

<p>Below is the link of anchor tag, click the link and see the next page</p>

<a href="https://www.irctc.co.in/nget/train-search">This is a link</a>

</body>

</html>

## The src Attribute

The **src** attribute is one of the important and required attribute of **<img>** element. It is source for the image which is required to display on browser. This attribute can contain image in same directory or another directory. The image name or source should be correct else browser will not display the image.

<!DOCTYPE html>

<html>

<head> </head>

<body>

<h1>Example of src attribute</h1>

<p>HTML images can be diplayed with the help of image tag and its attribute src gives the source for that image</p>

<img src=”” height="400" width="600">

</body>

</html>

## Quotes: single quotes or double quotes?

In this chapter you have seen that, we have used attribute with double quotes, but some people might use single quotes in HTML. So use of single quotes with HTML attribute, is also allowed. The following both statements are absolutely fine.

**Note**: IN HTML5, you can also omit use of quotes around attribute values.

# HTML Elements

An HTML file is made of elements. These elements are responsible for creating web pages and define content in that webpage. An element in HTML usually consist of a start tag <tag name>, close tag </tag name> and content inserted between them. **Technically, an element is a collection of start tag, attributes, end tag, content between them**.

#### **Note**: Some elements does not have end tag and content, these elements are termed as empty elements or self-closing element or void elements.

## Example

<!DOCTYPE html**>**

**<html>**

**<head>**

**<title>**WebPage**</title>**

**</head>**

**<body>**

**<h1>**This is my first web page**</h1>**

**<h2>** How it looks?**</h2>**

**<p>**It looks Nice!!!!!**</p>**

**</body>**

**</html>**

* All the content written between body elements are visible on web page.
* **Void element:** All the elements in HTML do not require to have start tag and end tag, some elements does not have content and end tag such elements are known as Void elements or empty elements. **These elements are also called as unpaired tag**.
* **Some Void elements are <br> (represents a line break) , <hr>(represents a horizontal line), etc.**

**Nested HTML Elements:** HTML can be nested, which means an element can contain another element.

## Block-level and Inline HTML elements

For the default display and styling purpose in HTML, all the elements are divided into two categories:

* Block-level element
* Inline element

## Block-level element:

* These are the elements, which structure main part of web page, by dividing a page into coherent blocks.
* A block-level element always start with new line and takes the full width of web page, from left to right.
* These elements can contain block-level as well as inline elements.

Following are the block-level elements in HTML.

<address>, <article>, <aside>, <blockquote>, <canvas>, <dd>, <div>, <dl>, <dt>, <fieldset>, <figcaption>, <figure>, <footer>, <form>, <h1>-<h6>, <header>, <hr>, <li>, <main>, <nav>, <noscript>, <ol>, <output>, <p>, <pre>, <section>, <table>, <tfoot>, <ul> and <video>.

<!DOCTYPE html**>**

**<html>**

**<head>**  **</head>**

**<body>**

**<div** style="background-color: lightblue"**>**This is first div**</div>**

**<div** style="background-color: lightgreen"**>**This is second div**</div>**

**<p** style="background-color: pink"**>**This is a block level element**</p>**

**</body>**

**</html>**

In the above example we have used div tag, which defines a section in a web page, and takes full width of page. We have used style attribute which is used to styling the HTML content, and the background color are showing that it's a block level element.

## Inline elements:

* Inline elements are those elements, which differentiate the part of a given text and provide it a particular function.
* These elements does not start with new line and take width as per requirement.
* The Inline elements are mostly used with other elements.

<a>, <abbr>, <acronym>, <b>, <bdo>, <big>, <br>, <button>, <cite>, <code>, <dfn>, <em>, <i>, <img>, <input>, <kbd>, <label>, <map>, <object>, <q>, <samp>, <script>, <select>, <small>, <span>, <strong>, <sub>, <sup>, <textarea>, <time>, <tt>, <var>.

## Example:

<!DOCTYPE html**>**

**<html>**

**<head>**    **</head>**

**<body>**

**<a** href=" "**>**Click on link**</a>**

**<span** style="background-color: lightblue"**>**this is inline element**</span>**

**<p>**This will take width of text only**</p>**

**</body>**

**</html>**

Following is the list of the some main elements used in HTML:

|  |  |  |  |
| --- | --- | --- | --- |
| **Start tag** | **Content** | **End tag** | **Description** |
| <h1> ...... <h6> | These are headings of HTML | </h1>??..</h6> | These elements are used to provide  the headings of page. |
| <p> | This is the paragraph | </p> | Used to display a content  in the form of paragraph. |
| <div> | This is div section | </div> | Used to provide a section  In the web page. |
| <br> |  |  | This element is used to provide a  line break. ( void element) |
| <hr> |  |  | This element is used to provide a  horizontal line. (void element) |

# HTML Formatting

**HTML Formatting** is a process of formatting text for better look and feel. HTML provides us ability to format text without using CSS. There are many formatting tags in HTML. These tags are used to make text bold, italicized, or underlined. There are almost 14 options available that how text appears in HTML and XHTML.

In HTML the formatting tags are divided into two categories:

* Physical tag: These tags are used to provide the visual appearance to the text.
* Logical tag: These tags are used to add some logical or semantic value to the text.

#### **Note**: There are some physical and logical tags which may give same visual appearance, but they will be different in semantics.

|  |  |
| --- | --- |
| **Element name** | **Description** |
| <b> | This is a physical tag, which is used to bold the text written between it. |
| <strong> | 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. |
| <em> | 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. |
| <tt> | This tag is used to appear a text in teletype. (not supported in HTML5) |
| <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. |

Here, we are going to learn 14 HTML formatting tags. Following is the list of HTML formatting text.

## 1) Bold Text

HTML<b> and <strong> formatting elements

The HTML <b> element is a physical tag which display text in bold font, without any logical importance. If you write anything within <b>............</b> element, is shown in bold letters.

<!DOCTYPE>

<html>

<body>

<p> <b>Write Your First Paragraph in bold text.</b></p>

</body>

</html>

The HTML <strong> tag is a logical tag, which displays the content in bold font and informs the browser about its logical importance. If you write anything between <strong>???????. </strong>, is shown important text.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<title>**formatting elements**</title>**

**</head>**

**<body>**

**<h1>**Explanation of formatting element**</h1>**

**<p><strong>**This is an important content**</strong>**, and this is normal content**</p>**

**</body>**

**</html>**

## 2) Italic Text

**HTML <i> and <em> formatting elements**

The HTML <i> element is physical element, which display the enclosed content in italic font, without any added importance. If you write anything within <i>............</i> element, is shown in italic letters.

<!DOCTYPE>

<html>

<body>

<p> <i>Write Your First Paragraph in italic text.</i></p>

</body>

</html>

The HTML <em> tag is a logical element, which will display the enclosed content in italic font, with added semantics importance

<!DOCTYPE html**>**

**<html>**

**<head>**

**<title>**formatting elements**</title>**

**</head>**

**<body>**

**<h1>**Explanation of italic formatting element**</h1>**

**<p><em>**This is an important content**</em>**, which displayed in italic font.**</p>**

**</body>**

**</html>**

## 3) HTML Marked formatting

If you want to mark or highlight a text, you should write the content within <mark>.........</mark>.

See this example:

<!DOCTYPE>

<html>

<body>

<h2>I want to put a <mark> Mark</mark> on your face</h2>

</body>

</html>

## 4) Underlined Text

If you write anything within <u>.........</u> element, is shown in underlined text.

See this example:

<!DOCTYPE>

<html>

<body>

<p> <u>Write Your First Paragraph in underlined text.</u></p>

</body>

</html>

## 5) Strike Text

Anything written within <strike>.......................</strike> element is displayed with strikethrough. It is a thin line which cross the statement.

See this example:

<!DOCTYPE>

<html>

<body>

<p> <strike>Write Your First Paragraph with strikethrough</strike>.</p>

</body>

</html>

## 6) Monospaced Font

If you want that each letter has the same width then you should write the content within <tt>.............</tt> element.

Note: We know that most of the fonts are known as variable-width fonts because different letters have different width. (for example: 'w' is wider than 'i'). Monospaced Font provides similar space among every letter.

See this example:

<!DOCTYPE>

<html>

<body>

<p>Hello <tt>Write Your First Paragraph in monospaced font.</tt></p>

</body>

</html>

## 7) Superscript Text

If you put the content within <sup>..............</sup> element, is shown in superscript; means it is displayed half a character's height above the other characters.

See this example:

<!DOCTYPE>

<html>

<body>

<p>Hello <sup>Write Your First Paragraph in superscript.</sup></p>

</body>

</html>

## 8) Subscript Text

If you put the content within <sub>..............</sub> element, is shown in subscript ; means it is displayed half a character's height below the other characters.

See this example:

<!DOCTYPE>

<html>

<body>

<p>Hello <sub>Write Your First Paragraph in subscript.</sub></p>

</body>

</html>

## 9) Deleted Text

Anything that puts within <del>..........</del> is displayed as deleted text.

See this example:

<!DOCTYPE>

<html>

<body>

<p>Hello <del>Delete your first paragraph.</del></p>

</body>

</html>

## 10) Inserted Text

Anything that puts within <ins>..........</ins> is displayed as inserted text.

See this example:

<!DOCTYPE>

<html>

<body>

<p><del>Delete your first paragraph.</del><ins>Write another paragraph.</ins></p>

</body>

</html>

## 11) Larger Text

If you want to put your font size larger than the rest of the text then put the content within <big>.........</big>. It increase one font size larger than the previous one.

See this example:

<!DOCTYPE>

<html>

<body>

<p>Hello <big>Write the paragraph in larger font.</big></p>

</body>

</html>

## 12) Smaller Text

If you want to put your font size smaller than the rest of the text then put the content within <small>.........</small>tag. It reduces one font size than the previous one.

See this example:

<!DOCTYPE>

<html>

<body>

<p>Hello <small>Write the paragraph in smaller font.</small></p>

</body>

</html>

# HTML Heading

A HTML heading or HTML h tag can be defined as a title or a subtitle which you want to display on the webpage. When you place the text within the heading tags <h1>.........</h1>, it is displayed on the browser in the bold format and size of the text depends on the number of heading.

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.

**Headings in HTML helps the search engine to understand and index the structure of web page.**

#### **Note**: The main keyword of the whole content of a webpage should be display by h1 heading tag.

<!DOCTYPE html>

<html>

<body>

<h1>Heading no. 1</h1>

<h2>Heading no. 2</h2>

<h3>Heading no. 3</h3>

<h4>Heading no. 4</h4>

<h5>Heading no. 5</h5>

<h6>Heading no. 6</h6>

</body>

</html>

#### **Note**: Heading elements (h1....h6) should be used for headings only. They should not be used just to make text bold or big.

* **HTML headings can also be used with nested elements. Following are different codes to display the way to use heading elements.**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<title>**Heading elements**</title>**

**</head>**

**<body>**

**<h1>**This is main heading of page. **</h1>**       **<p>**h1 is the most important heading, which is used to display the keyword of page **</p>**

**<h2>**This is first sub-heading**</h2>**

**<p>**h2 describes the first sub heading of page. **</p>**

**<h3>**This is Second sub-heading**</h3>**

**<p>**h3 describes the second sub heading of page.**</p>**

**<p>**We can use h1 to h6 tag to use the different sub-heading with their paragraphs if required.

**</p>**

**</body>**

**</html>**

# HTML Paragraph

HTML paragraph or HTML p tag is used to define a paragraph in a webpage. Let's take a simple example to see how it work. It is a notable point that a browser itself add an empty line before and after a paragraph. An HTML <p> tag indicates starting of new paragraph.

#### **Note**: If we are using various <p> tags in one HTML file then browser automatically adds a single blank line between the two paragraphs.

<!DOCTYPE html>

<html>

<body>

<p>This is first paragraph.</p>

<p>This is second paragraph.</p>

<p>This is third paragraph.</p>

</body>

</html>

## Space inside HTML Paragraph

If you put a lot of spaces inside the HTML p tag, browser removes extra spaces and extra line while displaying the page. The browser counts number of spaces and lines as a single one.

<!DOCTYPE html>

<html>

<body>

<p>

I am

going to provide

you a tutorial on HTML

and hope that it will

be very beneficial for you.

</p>

<p>

Look, I put here a lot

of spaces but I know, Browser will ignore it.

</p>

<p>

You cannot determine the display of HTML</p>

<p>because resized windows may create different result.

</p>

</body>

</html>

## How to Use <br> and <hr> tag with paragraph?

An HTML <br> tag is used for line break and it can be used with paragraph elements. Following is the example to show how to use <br> with <p> element.

<!DOCTYPE html**>**

**<html>**

**<head>**

**</head>**

**<body>**

**<h2>** Use of line break with pragraph tag**</h2>**

**<p><br>**Papa and mama, and baby and Dot,

**<br>**Willie and me?the whole of the lot

**<br>**Of us all went over in Bimberlie's sleigh,

**<br>**To grandmama's house on Christmas day.

**</p>**

**</body>**

**</html>**

An HTML <hr> tag is used to apply a horizontal line between two statements or two paragraphs. Following is the example which is showing use of <hr> tag with paragraph.

<!DOCTYPE html**>**

**<html>**

**<head>**    **</head>**

**<body>**

**<h2>** Example to show a horizontal line with paragraphs**</h2>**

**<p>** An HTML hr tag draw a horizontal line and separate two paragraphs with that line.**<hr>** it will start a new paragraph.

**</p>**

**</body>**

**</html>**

# HTML Phrase tag

The HTML phrase tags are special purpose tags, which defines the structural meaning of a block of text or semantics of text. Following is the list of phrase tags, some of which we have already discussed in HTML formatting.

* Abbreviation tag : <abbr>
* Acronym tag: <acronym> (not supported in HTML5)
* Marked tag: <mark>
* Strong tag: <strong>
* Emphasized tag : <em>
* Definition tag: <dfn>
* Quoting tag: <blockquote>
* Short quote tag : <q>
* Code tag: <code>
* Keyboard tag: <kbd>
* Address tag: <address>

## 1. Text Abbreviation tag

This tag is used to abbreviate a text. To abbreviate a text, write text between <abbr> and </abbr> tag.

## Example

<!DOCTYPE html>

<html>

<head></head>

<body>

<h2> Hypertext Markup language </h2>

<p>An <abbr title = "Hypertext Markup language">HTML </abbr>language is used to create web pages.

</p>

</body>

</html>

## 2. Marked tag:

The content written between <mark> and </mark> tag will show as yellow mark on browser. This tag is used to highlight a particular text.

<!DOCTYPE html>

<html>

<head></head>

<body>

<h2>Example of mark tag</h2>

<p>This tag will <mark>highlight</mark> the text.</p>

</body>

</html>

## 3. Strong text:

This tag is used to display the important text of the content. The text written between <strong> and </strong> will be displayed as important text.

<!DOCTYPE html>

<html>

<head></head>

<body>

<h2>Example of strong tag</h2>

<p>In HTML, it is recommended to use <strong>lower-case</strong>, while writing a code.

</p>

</body>

</html>

## 4. Emphasized text

This tag is used to emphasize the text, and displayed the text in italic form. The text written between <em> and </em> tag will italicized the text.

<!DOCTYPE html>

<html>

<head></head>

<body>

<h2>Example of emphasized tag</h2>

<p>HTML is an <em>easy </em>to learn language.</p>

</body>

</html>

## 5. Definition tag:

When you use the <dfn> and </dfn> tags, it allow to specify the keyword of the content. Following is the example to show how to definition element.

<!DOCTYPE html>

<html>

<head></head>

<body>

<h2>Example of definition element</h2>

<p><dfn>HTML </dfn> is a markup language. </p>

</body>

</html>

## 6. Quoting text:

The HTML <blockquote> element shows that the enclosed content is quoted from another source. The Source URL can be given using the cite attribute, and text representation of source can display using **<cite> ..... </cite>element**.

<!DOCTYPE html>

<html>

<head></head>

<body>

<h2>Example of blockquote element</h2>

<blockquote cite="https://www.keepinspiring.me/famous-quotes/"><p>"The first step toward success is taken when you refuse to be a captive of the environment in which you first find yourself."</p></blockquote>

<cite>-Mark Caine</cite>

</body>

</html>

## 7. Short Quotations:

An HTML <q> ....... </q> element defines a short quotation. If you will put any content between <q> ....... </q>, then it will enclose the text in double quotes.

## Example:

<!DOCTYPE html>

<html>

<head></head>

<body>

<p>Great Motivational quote</p>

<p>Steve Jobs said: <q>If You Are Working On Something That You Really Care About, You Don?t Have To Be Pushed. The Vision Pulls You.</q>?</p>

</body>

</html>

## 8. Code tags

The HTML <code> </code> element is used to display the part of computer code. It will display the content in monospaced font.

<!DOCTYPE html>

<html>

<head></head>

<body>

<p>First Java program</p>

<p><code>class Simple{ public static void main(String args[]){

System.out.println("Hello Java"); }} </code>

</p>

</body>

</html>

## 9. Keyboard Tag

In HTML the keyboard tag, <kbd>, indicates that a section of content is a user input from keyboard.

<!DOCTYPE html>

<html>

<head></head>

<body>

<p>Keyboard input. </p>

<p>Please press <kbd>Ctrl</kbd> + <kbd>Shift</kbd> + t<kbd></kbd> to restore page on chrome.</p>

</body>

</html>

## 10. Address tag

An HTML <address> tag defines the contact information about the author of the content. The content written between <address> and </address> tag, then it will be displayed in italic font.

<html>

<head></head>

<body>

<p>Address Tag</p>

<address> You can ask your queries by contact us on <a href=" ">example123@newdomain.com</a>

<br> You can also visit at: <br>58 S. Garfield Street. Villa Rica, GA 30187.

</address>

</body>

</html>

# HTML Anchor

The **HTML anchor tag** defines *a hyperlink that links one page to another page*. It can create hyperlink to other web page as well as files, location, or any URL. The "href" attribute is the most important attribute of the HTML a tag. and which links to destination page or URL.

## href attribute of HTML anchor tag

The href attribute is used to define the address of the file to be linked. In other words, it points out the destination page.

The syntax of HTML anchor tag is given below.

<a href = "..........."> Link Text </a>

Let's see an example of HTML anchor tag.

<!DOCTYPE>

<html>

<body>

<a href="second.html">Click for Second Page</a>

</body>

</html>

## Specify a location for Link using target attribute

If we want to open that link to another page then we can use target attribute of <a> tag. With the help of this link will be open in next page.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<title></title>**

**</head>**

**<body>**

**<p>**Click on **<a** href="" target="\_blank"**>** this-link **</a>**to go on home page.**</p>**

**</body>**

**</html>**

**Note:**

* The **target** attribute can only use with href attribute in anchor tag.
* If we will not use target attribute then link will open in same page.

## Appearance of HTML anchor tag

An **unvisited link** is displayed underlined and blue.

A **visited link** displayed underlined and purple.

An **active link** is underlined and red.

# HTML Image

**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.

Let's see an example of HTML image.

<!DOCTYPE>

<html>

<body>

<h2>HTML Image Example</h2>

<img src="good-morning.jpg" alt="Good Friends"/>

</body>

</html>

## 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.

#### **1) 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.

The location of image may be on the same directory or another server.

#### **2) alt**

The alt attribute defines an alternate text for the image, if it can't be displayed. The value of the alt attribute describe the image in words.

#### **3) width**

It is an optional attribute which is used to specify the width to display the image. It is not recommended now. You should apply CSS in place of width attribute.

#### **4) height**

It h3 the height of the image. The HTML height attribute also supports iframe, image and object elements. It is not recommended now. You should apply CSS in place of height attribute.

## Use of height and width attribute with img tag

You have learnt about how to insert an image in your web page, now if we want to give some height and width to display image according to our requirement, then we can set it with height and width attributes of image.

## Example:

<!DOCTYPE html>

<html>

<head>

<title>Image tag</title>

</head>

<body>

<h2>HTML image example with height and width</h2>

<img src= height="80" width="100" alt="animal image">

</body>

</html>

#### **Note**: Always try to insert the image with height and width, else it may flicker while displaying on webpage.

## Use of alt attribute

We can use alt attribute with tag. It will display an alternative text in case if image cannot be displayed on browser. Following is the example for alt attribute:

**<img** src="animal.png" height="180" width="300" alt="animal image"**>**

## How to get image from another directory/folder?

To insert an image in your web, that image must be present in your same folder where you have put the HTML file. But if in some case image is available in some other directory then you can access the image like this:

**<img** src="E:/images/animal.png" height="180" width="300" alt="animal image"**>**

## Use <img> tag as a link

We can also link an image with other page or we can use an image as a link. To do this, put <img> tag inside the <a> tag.

## Example:

<!DOCTYPE html>

<html>

<head>

<title>Image tag</title>

</head>

<body>

<h2>Use image as a link</h2>

<p>Click on the image to know about robotics</p>

<a href=" "><img src=" " height="100" width="100"></a>

</body>

</html>

# **HTML 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)

#### Note: We can create a list inside another list, which will be termed as nested List.

## HTML Ordered List or Numbered List

In the ordered HTML lists, all the list items are marked with numbers by default. It is known as numbered list also. The ordered list starts with <ol> tag and the list items start with <li> tag.

<!DOCTYPE>

<html>

<body>

<ol>

<li>Aries</li>

<li>Bingo</li>

<li>Leo</li>

<li>Oracle</li>

</ol>

</body>

</html>

## HTML Unordered List or Bulleted List

In HTML Unordered list, all the list items are marked with bullets. It is also known as bulleted list also. The Unordered list starts with <ul> tag and list items start with the <li> tag.

<!DOCTYPE>

<html>

<body>

<ul>

<li>Aries</li>

<li>Bingo</li>

<li>Leo</li>

<li>Oracle</li>

</ul>

</body>

</html>

## HTML Description List or Definition List

HTML Description list is also a list style which is supported by HTML. It is also known as definition list where entries are listed like a dictionary or encyclopedia.

The definition list is very appropriate when you want to present glossary, list of terms or other name-value list.

The HTML definition list contains following three tags:

1. **<dl> tag** defines the start of the list.
2. **<dt> tag** defines a term.
3. **<dd> tag** defines the term definition (description).

<!DOCTYPE>

<html>

<body>

<dl>

<dt>Aries</dt>

<dd>-One of the 12 horoscope sign.</dd>

<dt>Bingo</dt>

<dd>-One of my evening snacks</dd>

<dt>Leo</dt>

<dd>-It is also an one of the 12 horoscope sign.</dd>

<dt>Oracle</dt>

<dd>-It is a multinational technology corporation.</dd>

</dl>

</body>

</html>

## HTML Nested List

A list within another list is termed as nested list. If you want a bullet list inside a numbered list then such type of list will called as nested list.

<!DOCTYPE html>

<html>

<head>

<title>Nested list</title>

</head>

<body>

<p>List of Indian States with thier capital</p>

<ol>

<li>Delhi

<ul>

<li>NewDelhi</li>

</ul>

</li>

<li>Haryana

<ul>

<li>Chandigarh</li>

</ul>

</li>

<li>Gujarat

<ul>

<li>Gandhinagar</li>

</ul>

</li>

<li>Rajasthan

<ul>

<li>Jaipur</li>

</ul>

</li>

<li>Maharashtra

<ul>

<li>Mumbai</li>

</ul>

</li>

<li>Uttarpradesh

<ul>

<li>Lucknow</li></ul>

</li>

</ol>

</body>

</html>

# **HTML Ordered List | HTML Numbered List**

**HTML Ordered List** or Numbered List displays elements in numbered format. The HTML ol tag is used for ordered list. We can use ordered list to represent items either in numerical order format or alphabetical order format, or any format where an order is emphasized. There can be different types of numbered list:

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

To represent different ordered lists, there are 5 types of attributes in <ol> tag.

|  |  |
| --- | --- |
| **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. |

## HTML Ordered List Example

Let's see the example of HTML ordered list that displays 4 topics in numbered list. Here we are not defining type="1" because it is the default type.

<!DOCTYPE html>

<html>

<body>

<ol>

<li>HTML</li>

<li>Java</li>

<li>JavaScript</li>

<li>SQL</li>

</ol>

</body>

</html>

## ol type="I"

Let's see the example to display list in roman number uppercase.

<!DOCTYPE html>

<html>

<body>

<ol type="I">

<li>HTML</li>

<li>Java</li>

<li>JavaScript</li>

<li>SQL</li>

</ol>

</body>

</html>

## ol type="i"

Let's see the example to display list in roman number lowercase.

<!DOCTYPE html>

<html>

<body>

<ol type="i">

<li>HTML</li>

<li>Java</li>

<li>JavaScript</li>

<li>SQL</li>

</ol>

</body>

</html>

## ol type="A"

Let's see the example to display list in alphabet uppercase.

<!DOCTYPE html>

<html>

<body>

<ol type="A">

<li>HTML</li>

<li>Java</li>

<li>JavaScript</li>

<li>SQL</li>

</ol>

</body>

</html>

## ol type="a"

Let's see the example to display list in alphabet lowercase.

<!DOCTYPE html>

<html>

<body>

<ol type="a">

<li>HTML</li>

<li>Java</li>

<li>JavaScript</li>

<li>SQL</li>

</ol>

</body>

</html>

The start attribute is used with ol tag to specify from where to start the list items.

**<ol type="1" start="5">** : It will show numeric values starting with "5".

**<ol type="A" start="5">** : It will show capital alphabets starting with "E".

**<ol type="a" start="5">** : It will show lower case alphabets starting with "e".

**<ol type="I" start="5">** : It will show Roman upper case value starting with "V".

**<ol type="i" start="5">** : It will show Roman lower case value starting with "v".

Reversed Attribute:

This is a Boolean attribute of HTML <ol> tag, and it is new in HTML5 version. If you use the reversed attribute with

tag then it will numbered the list in descending order (7, 6, 5, 4......1).

Example:

<!DOCTYPE html>

<html>

<head>

</head>

<body>

<ol reversed>

<li>HTML</li>

<li>Java</li>

<li>JavaScript</li>

<li>SQL</li>

</ol>

</body>

</html>

# **HTML Unordered List | HTML Bulleted List**

**HTML Unordered List** or Bulleted List displays elements in bulleted format . We can use unordered list where we do not need to display items in any particular order. The HTML ul tag is used for the unordered list. There can be 4 types of bulleted list:

* disc
* circle
* square
* none

To represent different ordered lists, there are 4 types of attributes in <ul> tag.

|  |  |
| --- | --- |
| **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 . |

## HTML Unordered List Example

<!DOCTYPE html>

<html>

<body>

<ul>

<li>HTML</li>

<li>Java</li>

<li>JavaScript</li>

<li>SQL</li>

</ul>

</body>

</html>

## ul type="circle"

1. **<ul** type="circle"**>**
2. **<li>**HTML**</li>**
3. **<li>**Java**</li>**
4. **<li>**JavaScript**</li>**
5. **<li>**SQL**</li>**
6. **</ul>**

## ul type="square"

1. **<ul** type="square"**>**
2. **<li>**HTML**</li>**
3. **<li>**Java**</li>**
4. **<li>**JavaScript**</li>**
5. **<li>**SQL**</li>**
6. **</ul>**

## ul type="none"

1. **<ul** type="none"**>**
2. **<li>**HTML**</li>**
3. **<li>**Java**</li>**
4. **<li>**JavaScript**</li>**
5. **<li>**SQL**</li>**
6. **</ul>**

#### **Note**: The type attribute is not supported in HTML5, instead of type you can use CSS property of list-style-type. Following is the example to show the CSS property for ul tag.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **</head>**
5. **<body>**
6. **<h2>**The type attribute with CSS property**</h2>**
7. **<ul** style="list-style-type: square;"**>**
8. **<li>**HTML**</li>**
9. **<li>**Java**</li>**
10. **<li>**JavaScript**</li>**
11. **<li>**SQL**</li>**
12. **</ul>**
13. **</body>**
14. **</html>**

# **HTML Description List | HTML Definition List**

# **HTML Description List | HTML Definition List**

**HTML Description List** or Definition List displays elements in definition form like in dictionary. The <dl>, <dt> and <dd> tags are used to define description list.

The 3 HTML description list tags are given below:

1. **<dl> tag** defines the description list.
2. **<dt> tag** defines data term.
3. **<dd> tag** defines data definition (description).

<!DOCTYPE html>

<html>

<body>

<dl>

<dt>HTML</dt>

<dd>is a markup language</dd>

<dt>Java</dt>

<dd>is a programming language and platform</dd>

<dt>JavaScript</dt>

<dd>is a scripting language</dd>

<dt>SQL</dt>

<dd>is a query language</dd>

</dl>

</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. .

## Why use HTML Form

HTML forms are required if you want to collect some data from of the site visitor.

For example: If a user want to purchase some items on internet, he/she must fill the form such as shipping address and credit/debit card details so that item can be sent to the given address.

HTML Form Syntax

1. **<form** action="server url" method="get|post"**>**
2. //input controls e.g. textfield, textarea, radiobutton, button
3. **</form>**

## HTML Form Tags

Let's see the list of HTML 5 form tags.

|  |  |
| --- | --- |
| **Tag** | **Description** |
| <form> | It defines an HTML form to enter inputs by the used side. |
| <input> | It defines an input control. |
| <textarea> | It defines a multi-line input control. |
| <label> | It defines a label for an input element. |
| <fieldset> | It groups the related element in a form. |
| <legend> | It defines a caption for a <fieldset> element. |
| <select> | It defines a drop-down list. |
| <optgroup> | It defines a group of related options in a drop-down list. |
| <option> | It defines an option in a drop-down list. |
| <button> | It defines a clickable button. |

HTML 5 Form Tags

Let's see the list of HTML 5 form tags.

|  |  |
| --- | --- |
| **Tag** | **Description** |
| <datalist> | It specifies a list of pre-defined options for input control. |
| <keygen> | It defines a key-pair generator field for forms. |
| <output> | It defines the result of a calculation. |

## HTML <form> element

The HTML <form> element provide a document section to take input from user. It provides various interactive controls for submitting information to web server such as text field, text area, password field, etc.

#### **Note**: The <form> element does not itself create a form but it is container to contain all required form elements, such as <input>, <label>, etc.

## HTML <input> element

The HTML <input> element is fundamental form element. It is used to create form fields, to take input from user. We can apply different input filed to gather different information form user. Following is the example to show the simple text input.

1. **<body>**
2. **<form>**
3. Enter your name  **<br>**
4. **<input** type="text" name="username"**>**
5. **</form>**
6. **</body>**

## HTML TextField Control

The type="text" attribute of input tag creates textfield control also known as single line textfield control. The name attribute is optional, but it is required for the server side component such as JSP, ASP, PHP etc.

1. **<form>**
2. First Name: **<input** type="text" name="firstname"**/>** **<br/>**
3. Last Name:  **<input** type="text" name="lastname"**/>** **<br/>**
4. **</form>**

#### **Note**: If you will omit 'name' attribute then the text filed input will not be submitted to server.

## HTML <textarea> tag in form

The <textarea> tag in HTML is used to insert multiple-line text in a form. The size of <textarea> can be specify either using "rows" or "cols" attribute or by CSS.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Form in HTML**</title>**
5. **</head>**
6. **<body>**
7. **<form>**
8. Enter your address:**<br>**
9. **<textarea** rows="2" cols="20"**></textarea>**
10. **</form>**
11. **</body>**
12. **</html>**

## Label Tag in Form

It is considered better to have label in form. As it makes the code parser/browser/user friendly.

If you click on the label tag, it will focus on the text control. To do so, you need to have for attribute in label tag that must be same as id attribute of input tag.

#### **NOTE**: It is good to use <label> tag with form, although it is optional but if you will use it, then it will provide a focus when you tap or click on label tag. It is more worthy with touchscreens.

## HTML Password Field Control

The password is not visible to the user in password field control.

1. **<form>**
2. **<label** for="password"**>**Password: **</label>**
3. **<input** type="password" id="password" name="password"**/>** **<br/>**
4. **</form>**

## HTML 5 Email Field Control

The email field in new in HTML 5. It validates the text for correct email address. You must use @ and . in this field.

1. **<form>**
2. **<label** for="email"**>**Email: **</label>**
3. **<input** type="email" id="email" name="email"**/>** **<br/>**
4. **</form>**

#### **Note**: If we will not enter the correct email, it will display error like:

## Radio Button Control

The radio button is used to select one option from multiple options. It is used for selection of gender, quiz questions etc.

If you use one name for all the radio buttons, only one radio button can be selected at a time.

Using radio buttons for multiple options, you can only choose a single option at a time.

1. **<form>**
2. **<label** for="gender"**>**Gender: **</label>**
3. **<input** type="radio" id="gender" name="gender" value="male"**/>**Male
4. **<input** type="radio" id="gender" name="gender" value="female"**/>**Female **<br/>**
5. **</form>**

## Checkbox Control

The checkbox control is used to check multiple options from given checkboxes.

1. **<form>**
2. Hobby:**<br>**
3. **<input** type="checkbox" id="cricket" name="cricket" value="cricket"**/>**
4. **<label** for="cricket"**>**Cricket**</label>** **<br>**
5. **<input** type="checkbox" id="football" name="football" value="football"**/>**
6. **<label** for="football"**>**Football**</label>** **<br>**
7. **<input** type="checkbox" id="hockey" name="hockey" value="hockey"**/>**
8. **<label** for="hockey"**>**Hockey**</label>**
9. **</form>**

#### **Note**: These are similar to radio button except it can choose multiple options at a time and radio button can select one button at a time, and its display.

## Submit button control

HTML **<input type="submit">** are used to add a submit button on web page. When user clicks on submit button, then form get submit to the server.

Syntax:

1. **<input** type="submit" value="submit"**>**

The type = submit, specifying that it is a submit button

The value attribute can be anything which we write on button on web page.

The name attribute can be omit here.

**Example:**

1. **<form>**
2. **<label** for="name"**>**Enter name**</label><br>**
3. **<input** type="text" id="name" name="name"**><br>**
4. **<label** for="pass"**>**Enter Password**</label><br>**
5. **<input** type="Password" id="pass" name="pass"**><br>**
6. **<input** type="submit" value="submit"**>**
7. **</form>**

## HTML <fieldset> element:

The <fieldset> element in HTML is used to group the related information of a form. This element is used with <legend> element which provide caption for the grouped elements.

**Example:**

1. **<form>**
2. **<fieldset>**
3. **<legend>**User Information:**</legend>**
4. **<label** for="name"**>**Enter name**</label><br>**
5. **<input** type="text" id="name" name="name"**><br>**
6. **<label** for="pass"**>**Enter Password**</label><br>**
7. **<input** type="Password" id="pass" name="pass"**><br>**
8. **<input** type="submit" value="submit"**>**
9. **</fieldset>**
10. </form**>**

## HTML Form Example

Following is the example for a simple form of registration.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Form in HTML**</title>**
5. **</head>**
6. **<body>**
7. **<h2>**Registration form**</h2>**
8. **<form>**
9. **<fieldset>**
10. **<legend>**User personal information**</legend>**
11. **<label>**Enter your full name**</label><br>**
12. **<input** type="text" name="name"**><br>**
13. **<label>**Enter your email**</label><br>**
14. **<input** type="email" name="email"**><br>**
15. **<label>**Enter your password**</label><br>**
16. **<input** type="password" name="pass"**><br>**
17. **<label>**confirm your password**</label><br>**
18. **<input** type="password" name="pass"**><br>**
19. **<br><label>**Enter your gender**</label><br>**
20. **<input** type="radio" id="gender" name="gender" value="male"**/>**Male  **<br>**
21. **<input** type="radio" id="gender" name="gender" value="female"**/>**Female **<br/>**
22. **<input** type="radio" id="gender" name="gender" value="others"**/>**others **<br/>**
23. **<br>**Enter your Address:**<br>**
24. **<textarea></textarea><br>**
25. **<input** type="submit" value="sign-up"**>**
26. **</fieldset>**
27. **</form>**
28. **</body>**
29. **</html>**

## HTML Form Example

Let's see a simple example of creating HTML form.

1. **<form** action="#"**>**
2. **<table>**
3. **<tr>**
4. **<td** class="tdLabel"**><label** for="register\_name" class="label"**>**Enter name:**</label></td>**
5. **<td><input** type="text" name="name" value="" id="register\_name" style="width:160px"**/></td>**
6. **</tr>**
7. **<tr>**
8. **<td** class="tdLabel"**><label** for="register\_password" class="label"**>**Enter password:**</label></td>**
9. **<td><input** type="password" name="password" id="register\_password" style="width:160px"**/></td>**
10. **</tr>**
11. **<tr>**
12. **<td** class="tdLabel"**><label** for="register\_email" class="label"**>**Enter Email:**</label></td>**
13. **<td**
14. **><input** type="email" name="email" value="" id="register\_email" style="width:160px"**/></td>**
15. **</tr>**
16. **<tr>**
17. **<td** class="tdLabel"**><label** for="register\_gender" class="label"**>**Enter Gender:**</label></td>**
18. **<td>**
19. **<input** type="radio" name="gender" id="register\_gendermale" value="male"**/>**
20. **<label** for="register\_gendermale"**>**male**</label>**
21. **<input** type="radio" name="gender" id="register\_genderfemale" value="female"**/>**
22. **<label** for="register\_genderfemale"**>**female**</label>**
23. **</td>**
24. **</tr>**
25. **<tr>**
26. **<td** class="tdLabel"**><label** for="register\_country" class="label"**>**Select Country:**</label></td>**
27. **<td><select** name="country" id="register\_country" style="width:160px"**>**
28. **<option** value="india"**>**india**</option>**
29. **<option** value="pakistan"**>**pakistan**</option>**
30. **<option** value="africa"**>**africa**</option>**
31. **<option** value="china"**>**china**</option>**
32. **<option** value="other"**>**other**</option>**
33. **</select>**
34. **</td>**
35. **</tr>**
36. **<tr>**
37. **<td** colspan="2"**><div** align="right"**><input** type="submit" id="register\_0" value="register"**/>**
38. **</div></td>**
39. **</tr>**
40. **</table>**
41. **</form>**

# **HTML Form Input Types**

In HTML <input type=" "> is an important element of HTML form. The "type" attribute of input element can be various types, which defines information field. Such as <input type="text" name="name"> gives a text box.

## Following is a list of all types of <input> element of HTML.

|  |  |
| --- | --- |
| **type=" "** | **Description** |
| text | Defines a one-line text input field |
| password | Defines a one-line password input field |
| submit | Defines a submit button to submit the form to server |
| reset | Defines a reset button to reset all values in the form. |
| radio | Defines a radio button which allows select one option. |
| checkbox | Defines checkboxes which allow select multiple options form. |
| button | Defines a simple push button, which can be programmed to perform a task on an event. |
| file | Defines to select the file from device storage. |
| image | Defines a graphical submit button. |

**HTML5 added new types on <input> element. Following is the list of types of elements of HTML5**

|  |  |
| --- | --- |
| **type=" "** | **Description** |
| color | Defines an input field with a specific color. |
| date | Defines an input field for selection of date. |
| datetime-local | Defines an input field for entering a date without time zone. |
| email | Defines an input field for entering an email address. |
| month | Defines a control with month and year, without time zone. |
| number | Defines an input field to enter a number. |
| url | Defines a field for entering URL |
| week | Defines a field to enter the date with week-year, without time zone. |
| search | Defines a single line text field for entering a search string. |
| tel | Defines an input field for entering the telephone number. |

**Following is the description about types of <input> element with examples.**

### **1. <input type="text">:**

<input> element of type "text" are used to define a single-line input text field.

### **Example:**

1. **<form>**
2. **<label>**Enter first name**</label><br>**
3. **<input** type="text" name="firstname"**><br>**
4. **<label>**Enter last name**</label><br>**
5. **<input** type="text" name="lastname"**><br>**
6. **<p><strong>**Note:**</strong>**The default maximum cahracter lenght is 20.**</p>**
7. **</form>**

### **2. <input type="password">:**

The <input> element of type "password" allow a user to enter the password securely in a webpage. The entered text in password filed converted into "\*" or ".", so that it cannot be read by another user.

### **Example:**

1. **<form>**
2. **<label>**Enter User name**</label><br>**
3. **<input** type="text" name="firstname"**><br>**
4. **<label>**Enter Password**</label><br>**
5. **<input** type="Password" name="password"**><br>**
6. **<br><input** type="submit" value="submit"**>**
7. **</form>**

### **3. <input type="submit">:**

The <input> element of type "submit" defines a submit button to submit the form to the server when the "click" event occurs.

### **Example:**

1. **<form** action="https://www.amazon.in"**>**
2. **<label>**Enter User name**</label><br>**
3. **<input** type="text" name="firstname"**><br>**
4. **<label>**Enter Password**</label><br>**
5. **<input** type="Password" name="password"**><br>**
6. **<br><input** type="submit" value="submit"**>**
7. **</form>**

### **4. <input type="reset">:**

The <input> type "reset" is also defined as a button but when the user performs a click event, it by default reset the all inputted values.

### **Example:**

1. **<form>**
2. **<label>**User id: **</label>**
3. **<input** type="text" name="user-id" value="user"**>**
4. **<label>**Password: **</label>**
5. **<input** type="password" name="pass" value="pass"**><br><br>**
6. **<input** type="submit" value="login"**>**
7. **<input** type="reset" value="Reset"**>**
8. **</form>**

### **5. <input type="radio">:**

The <input> type "radio" defines the radio buttons, which allow choosing an option between a set of related options. At a time only one radio button option can be selected at a time.

### **Example:**

1. **<form>**
2. **<p>**Kindly Select your favorite color**</p>**
3. **<input** type="radio" name="color" value="red"**>** Red **<br>**
4. **<input** type="radio" name="color" value="blue"**>** blue **<br>**
5. **<input** type="radio" name="color" value="green"**>**green **<br>**
6. **<input** type="radio" name="color" value="pink"**>**pink **<br>**
7. **<input** type="submit" value="submit"**>**
8. **</form>**

### **6. <input type="checkbox">:**

The <input> type "checkbox" are displayed as square boxes which can be checked or unchecked to select the choices from the given options.

#### **Note**: The "radio" buttons are similar to checkboxes, but there is an important difference between both types: radio buttons allow the user to select only one option at a time, whereas checkbox allows a user to select zero to multiple options at a time.

### **Example:**

1. **<form>**
2. **<label>**Enter your Name:**</label>**
3. **<input** type="text" name="name"**>**
4. **<p>**Kindly Select your favourite sports**</p>**
5. **<input** type="checkbox" name="sport1" value="cricket"**>**Cricket**<br>**
6. **<input** type="checkbox" name="sport2" value="tennis"**>**Tennis**<br>**
7. **<input** type="checkbox" name="sport3" value="football"**>**Football**<br>**
8. **<input** type="checkbox" name="sport4" value="baseball"**>**Baseball**<br>**
9. **<input** type="checkbox" name="sport5" value="badminton"**>**Badminton**<br><br>**
10. **<input** type="submit" value="submit"**>**
11. **</form>**

### **7. <input type="button">:**

The <input> type "button" defines a simple push button, which can be programmed to control a functionally on any event such as, click event.

#### **Note**: It mainly works with JavaScript.

### **Example:**

1. **<form>**
2. **<input** type="button" value="Click me " onclick="alert('you are learning HTML')"**>**
3. **</form>**

### **8. <input type="file">:**

The <input> element with type "file" is used to select one or more files from user device storage. Once you select the file, and after submission, this file can be uploaded to the server with the help of JS code and file API.

### **Example:**

1. **<form>**
2. **<label>**Select file to upload:**</label>**
3. **<input** type="file" name="newfile"**>**
4. **<input** type="submit" value="submit"**>**
5. **</form>**

### **9. <input type="image">:**

The <input> type "image" is used to represent a submit button in the form of image.

### **Example:**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<body>**
4. **<h2>**Input "image" type.**</h2>**
5. **<p>**We can create an image as submit button**</p>**
6. **<form>**
7. **<label>**User id:**</label><br>**
8. **<input** type="text" name="name"**><br><br>**
9. **<input** type="image" alt="Submit" src="login.png"  width="100px"**>**
10. **</form>**
12. **</body>**
13. **</html>**

## HTML5 newly added <input> types element

### **1. <input type="color">:**

The <input> type "color" is used to define an input field which contains a colour. It allows a user to specify the colour by the visual colour interface on a browser.

#### **Note**: The "color" type only supports color value in hexadecimal format, and the default value is #000000 (black).

### **Example:**

1. **<form>**
2. Pick your Favorite color: **<br><br>**
3. **<input** type="color" name="upclick" value="#a52a2a"**>** Upclick**<br><br>**
4. **<input** type="color" name="downclick" value="#f5f5dc"**>** Downclick
5. **</form>**

### **2. <input type="date">:**

The <input> element of type "date" generates an input field, which allows a user to input the date in a given format. A user can enter the date by text field or by date picker interface.

### **Example:**

1. **<form>**
2. Select Start and End Date: **<br><br>**
3. **<input** type="date" name="Startdate"**>** Start date:**<br><br>**
4. **<input** type="date" name="Enddate"**>** End date:**<br><br>**
5. **<input** type="submit"**>**
6. **</form>**

### **3. <input type="datetime-local">:**

The <input> element of type "datetime-local" creates input filed which allow a user to select the date as well as local time in the hour and minute without time zone information.

### **Example:**

1. **<form>**
2. **<label>**
3. Select the meeting schedule: **<br><br>**
4. Select date & time: **<input** type="datetime-local" name="meetingdate"**>** **<br><br>**
5. **</label>**
6. **<input** type="submit"**>**
7. **</form>**

### **4. <input type="email">:**

The <input> type "email" creates an input filed which allow a user to enter the e-mail address with pattern validation. The multiple attributes allow a user to enter more than one email address.

### **Example:**

1. **<form>**
2. **<label><b>**Enter your Email-address**</b></label>**
3. **<input** type="email" name="email" required**>**
4. **<input** type="submit"**>**
5. **<p><strong>**Note:**</strong>**User can also enter multiple email addresses separating by comma or whitespace as following: **</p>**
6. **<label><b>**Enter multiple Email-addresses**</b></label>**
7. **<input** type="email" name="email"  multiple**>**
8. **<input** type="submit"**>**
9. **</form>**

### **5. <input type="month">:**

The <input> type "month" creates an input field which allows a user to easily enter month and year in the format of "MM, YYYY" where MM defines month value, and YYYY defines the year value. New

### **Example:**

1. **<form>**
2. **<label>**Enter your Birth Month-year: **</label>**
3. **<input** type="month" name="newMonth"**>**
4. **<input** type="submit"**>**
5. **</form>**

### **6. <input type="number">:**

The <input> element type number creates input filed which allows a user to enter the numeric value. You can also restrict to enter a minimum and maximum value using min and max attribute.

### **Example:**

1. **<form>**
2. **<label>**Enter your age: **</label>**
3. **<input** type="number" name="num" min="50" max="80"**>**
4. **<input** type="submit"**>**
5. **</form>**

### **7. <input type="url">:**

The <input> element of type "url" creates an input filed which enables user to enter the URL.

### **Example:**

1. **<form>**
2. **<label>**Enter your website URL: **</label>**
3. **<input** type="url" name="website" placeholder="http://example.com"**><br>**
4. **<input** type="submit" value="send data"**>**
5. **</form>**

### **8. <input type="week">:**

The <input> type week creates an input field which allows a user to select a week and year form the drop-down calendar without time zone.

### **Example:**

1. **<form>**
2. **<label><b>**Select your best week of year:**</b></label><br><br>**
3. **<input** type="week" name="bestweek"**>**
4. **<input** type="submit" value="Send data"**>**
5. **</form>**

### **9. <input type="search">:**

The <input> type "search" creates an input filed which allows a user to enter a search string. These are functionally symmetrical to the text input type, but may be styled differently.

### **Example:**

1. **<form>**
2. **<label>**Search here:**</label>**
3. **<input** type="search" name="q"**>**
4. **<input** type="submit" value="search"**>**
5. **</form>**

### **10. <input type="tel">:**

The <input> element of type ?tel? creates an input filed to enter the telephone number. The "tel" type does not have default validation such as email, because telephone number pattern can vary worldwide.

### **Example:**

1. **<form>**
2. **<label><b>**Enter your Telephone Number(in format of xxx-xxx-xxxx):**</b></label>**
3. **<input** type="tel" name="telephone" pattern="[0-9]{3}-[0-9]{3}-[0-9]{4}" required**>**
4. **<input** type="submit"**><br><br>**
5. **</form>**

# **HTML form Attribute**

## HTML <form> element attributes

In HTML there are various attributes available for <form> element which are given below:

## HTML action attribute

The action attribute of <form> element defines the process to be performed on form when form is submitted, or it is a URI to process the form information.

The action attribute value defines the web page where information proceed. It can be .php, .jsp, .asp, etc. or any URL where you want to process your form.

#### **Note**: If action attribute value is blank then form will be processed to the same page.

<!DOCTYPE html>

<html>

<body>

<h2>Demo of action attribute of form element</h2>

<form action="action.html" method="post">

<label>User Name:</label><br>

<input type="text" name="name"><br><br>

<label>User Password</label><br>

<input type="password" name="pass"><br><br>

<input type="submit">

</form>

<p><b>It will redirect to a new page "action.html" when you click on submit button</b></p>

</body>

</html>

### **HTML method attribute**

The method attribute defines the HTTP method which browser used to submit the form. The possible values of method attribute can be:

* **post:** We can use the post value of method attribute when we want to process the sensitive data as it does not display the submitted data in URL.

### **Example:**

1. **<form** action="action.html" method="post"**>**

* **get:** The get value of method attribute is default value while submitting the form. But this is not secure as it displays data in URL after submitting the form.

### **Example:**

1. **<form** action="action.html" method="get"**>**

When submitting the data, it will display the entered data in the form of:

1. file:///D:/HTML/action.html?name=QualityThought&pass=123

## HTML target attribute

The target attribute defines where to open the response after submitting the form. The following are the keywords used with the target attribute.

* **\_self:** If we use \_self as an attribute value, then the response will display in current page only.

### **Example:**

1. **<form** action="action.html" method="get" target="\_self"**>**

* **\_blank:** If we use \_blank as an attribute it will load the response in a new page.

### **Example:**

1. **<form** action="action.html" method="get" target="\_blank"**>**

## HTML autocomplete attribute

The HTML autocomplete attribute is a newly added attribute of HTML5 which enables an input field to complete automatically. It can have two values "on" and "off" which enables autocomplete either ON or OFF. The default value of autocomplete attribute is "on".

### **Example:**

1. **<form** action="action.html" method="get" autocomplete="on"**>**

### **Example:**

1. **<form** action="action.html" method="get" autocomplete="off"**>**

#### **Note**: it can be used with <form> element and <input> element both.

## HTML enctype attribute

The HTML enctype attribute defines the encoding type of form-content while submitting the form to the server. The possible values of enctype can be:

* **application/x-www-form-urlencoded:** It is default encoding type if the enctype attribute is not included in the form. All characters are encoded before submitting the form.

### **Example:**

1. **<form** action="action.html" method="post" enctype="application/x-www-form-urlencoded" **>**

* **multipart/form-data:** It does not encode any character. It is used when our form contains file-upload controls.

### **Example:**

1. **<form** action="action.html" method="post" enctype="multipart/form-data"**>**

* **text/plain (HTML5):** In this encoding type only space are encoded into + symbol and no any other special character encoded.

### **Example:**

1. **<form** action="action.html" method="post" enctype="text/plain" **>**

## HTML novalidate attribute HTML5

The novalidate attribute is newly added Boolean attribute of HTML5. If we apply this attribute in form then it does not perform any type of validation and submit the form.

### **Example:**

<!DOCTYPE html>

<html>

<body>

<h2>Fill the form</h2>

<form action = "action.html" method = "get" novalidate>

Enter name:<br><input type="name" name="name"><br>

Enter age:<br><input type="number" name="age"><br>

Enter email:<br><input type="email" name="email"><br>

<input type="submit" value="Submit">

</form>

<p><b>Try to change the form detials with novalidate atttribute and without novalidate attribute and see the difference.</b></p>

</body>

</html>

## HTML <input> element attribute

## HTML name attribute

The HTML name attribute defines the name of an input element. The name and value attribute are included in HTTP request when we submit the form.

#### **Note**: One should not omit the name attribute as when we submit the form the HTTP request includes both name-value pair and if name is not available it will not process that input field.

<!DOCTYPE html>

<html>

<body>

<h2>Fill the form</h2>

<form action = "action.html" method = "get">

Enter name:<br><input type="name" name="uname"><br>

Enter age:<br><input type="number" name="age"><br>

Enter email:<br><input type="email"><br>

<input type="submit" value="Submit">

</form>

<p><b>Note: If you will not use name attribute in any input field, then that input field will not be submitted, when submit the form.</b></p>

<p>Click on submit and see the URL where email is not included in HTTP request as we have not used name attribute in the email input field</p>

</body>

</html>

## HTML value attribute

The HTML value attribute defines the initial value or default value of an input field.

### **Example:**

<!DOCTYPE html>

<html>

<body>

<h2>Fill the form</h2>

<form>

<label>Enter your Name</label><br>

<input type="text" name="uname" value="Enter Name"><br><br>

<label>Enter your Email-address</label><br>

<input type="text" name="uname" value="Enter email"><br><br>

<label>Enter your password</label><br>

<input type="password" name="pass" value=""><br><br>

<input type="submit" value="login">

</form>

<p><b>Note: In password input filed the value attribute will always unclear</b></p>

</body>

</html>

## HTML required attribute HTML5

HTML required is a Boolean attribute which specifies that user must fill that filed before submitting the form.

### **Example:**

<!DOCTYPE html>

<html>

<body>

<h2>Fill the form</h2>

<form>

<label>Enter your Email-address</label><br>

<input type="text" name="uname" required><br><br>

<label>Enter your password</label><br>

<input type="password" name="pass"><br><br>

<input type="submit" value="login">

</form>

<p><b> If you will try to submit the form without completing email field then it will give an error pop up.</b></p>

</body>

</html>

## HTML autofocus attribute HTML5

The autofocus is a Boolean attribute which enables a field automatically focused when a webpage loads.

### **Example:**

1. **<form>**
2. **<label>**Enter your Email-address**</label><br>**
3. **<input** type="text" name="uname" autofocus**><br><br>**
4. **<label>**Enter your password**</label><br>**
5. **<input** type="password" name="pass"**><br><br>**
6. **<input** type="submit" value="login"**>**
7. **</form>**

## HTML placeholder attribute HTML5

The placeholder attribute specifies a text within an input field which informs the user about the expected input of that filed.

The placeholder attribute can be used with text, password, email, and URL values.

When the user enters the value, the placeholder will be automatically removed.

### **Example:**

<!DOCTYPE html>

<html>

<body>

<h3>Registration form</h3>

<form>

<label>Enter your name</label><br>

<input type="text" name="uname" placeholder="Your name"><br><br>

<label>Enter your Email address</label><br>

<input type="email" name="email" placeholder="example@gmail.com"><br><br>

<label>Enter your password</label><br>

<input type="password" name="pass" placeholder="your password"><br><br>

<input type="submit" value="login">

</form>

</body>

</html>

## HTML disabled attribute

The HTML disabled attribute when applied then it disable that input field. The disabled field does not allow the user to interact with that field.

The disabled input filed does not receive click events, and these input value will not be sent to the server when submitting the form.

### **Example:**

<!DOCTYPE html>

<html>

<body>

<h3>Registration form</h3>

<form>

<label>Enter User name</label><br>

<input type="text" name="uname" value="USER" disabled><br><br>

<label>Enter your Email address</label><br>

<input type="email" name="email" placeholder="example@gmail.com"><br><br>

<label>Enter your password</label><br>

<input type="password" name="pass" placeholder="your password"><br><br>

<input type="submit" value="login">

</form>

</body>

</html>

## HTML size attribute

The size attribute controls the size of the input field in typed characters.

### **Example:**

<!DOCTYPE html>

<html>

<body>

<h3>Registration form with disbaled attribute</h3>

<form>

<label>Account holder name</label><br>

<input type="text" name="uname" size="40" required><br><br>

<label>Account number</label><br>

<input type="text" name="an" size="30" required><br><br>

<label>CVV</label><br>

<input type="text" name="cvv" size="1" required><br><br>

<input type="submit" value="login">

</form>

</body>

</html>

## HTML form attribute

HTML form attribute allows a user to specify an input filed outside the form but remains the part of the parent form.

### **Example:**

<!DOCTYPE html>

<html>

<body>

<form id="fcontrol">

User Name:<br><input type="text" name="uname"><br><br>

User password:<br><input type="password" name="pass"><br><br>

</form>

<p>The email field is outside the form but still it will remain part of the form</p>

User email: <br><input type="email" name="email" form="fcontrol" required><br>

<input type="submit" form="fcontrol">

</body>

</html>

# HTML Table

**HTML table tag** is used to display data in tabular form (row \* column). There can be many columns in a row.

We can create a table to display data in tabular form, using <table> element, with the help of <tr>, <td>, and <th> elements.

|  |  |
| --- | --- |
| **Tag** | **Description** |
| <table> | It defines a table. |
| <tr> | It defines a row in a table. |
| <th> | It defines a header cell in a table. |
| <td> | It defines a cell in a table. |
| <caption> | It defines the table caption. |
| <colgroup> | It specifies a group of one or more columns in a table for formatting. |
| <col> | It is used with <colgroup> element to specify column properties for each column. |
| <tbody> | It is used to group the body content in a table. |
| <thead> | It is used to group the header content in a table. |
| <tfooter> | It is used to group the footer content in a table. |

In Each table, table row is defined by <tr> tag, table header is defined by <th>, and table data is defined by <td> tags.

HTML tables are used to manage the layout of the page e.g. header section, navigation bar, body content, footer section etc. But it is recommended to use div tag over table to manage the layout of the page.

**<html>**

**<body>**

**<table>**

**<tr><th>**First\_Name**</th><th>**Last\_Name**</th><th>**Marks**</th></tr>**

**<tr><td>**Sonoo**</td><td>**Jaiswal**</td><td>**60**</td></tr>**

**<tr><td>**James**</td><td>**William**</td><td>**80**</td></tr>**

**<tr><td>**Swati**</td><td>**Sironi**</td><td>**82**</td></tr>**

**<tr><td>**Chetna**</td><td>**Singh**</td><td>**72**</td></tr>**

**</table>**

**</body>**

**</html>**

## HTML Table with Border

There are two ways to specify border for HTML tables.

1. By border attribute of table in HTML
2. By border property in CSS

## 1) HTML Border attribute

You can use border attribute of table tag in HTML to specify border. But it is not recommended now.

<!DOCTYPE>

<html>

<body>

<table border="1">

<tr><th>First\_Name</th><th>Last\_Name</th><th>Marks</th>

</tr>

<tr><td>Sonoo</td><td>Jaiswal</td><td>60</td></tr>

<tr><td>James</td><td>William</td><td>80</td></tr>

<tr><td>Swati</td><td>Sironi</td><td>82</td></tr>

<tr><td>Chetna</td><td>Singh</td><td>72</td></tr>

</table>

</body>

</html>

## 2) CSS Border property

It is now recommended to use border property of CSS to specify border in table.

<!DOCTYPE>

<html>

<head>

<style>

table, th, td {

border: 1px solid black;

}

</style>

</head>

<body>

<table>

<tr><th>First\_Name</th><th>Last\_Name</th><th>Marks</th>

</tr>

<tr><td>Sonoo</td><td>Jaiswal</td><td>60</td></tr>

<tr><td>James</td><td>William</td><td>80</td></tr>

<tr><td>Swati</td><td>Sironi</td><td>82</td></tr>

<tr><td>Chetna</td><td>Singh</td><td>72</td></tr>

</table>

</body>

</html>

You can collapse all the borders in one border by border-collapse property. It will collapse the border into one.

**<style>**

table, th, td {

  border: 2px solid black;

  border-collapse: collapse;

}

**</style>**

## HTML Table with cell padding

You can specify padding for table header and table data by two ways:

1. By cellpadding attribute of table in HTML
2. By padding property in CSS

The cellpadding attribute of HTML table tag is obselete now. It is recommended to use CSS. So let's see the code of CSS.

<!DOCTYPE>

<html>

<head>

<style>

table, th, td {

border: 1px solid black;

border-collapse: collapse;

}

th, td {

padding: 10px;

}

</style>

</head>

<body>

<table>

<tr><th>First\_Name</th><th>Last\_Name</th><th>Marks</th>

</tr>

<tr><td>Sonoo</td><td>Jaiswal</td><td>60</td></tr>

<tr><td>James</td><td>William</td><td>80</td></tr>

<tr><td>Swati</td><td>Sironi</td><td>82</td></tr>

<tr><td>Chetna</td><td>Singh</td><td>72</td></tr>

</table>

</body>

</html>

## HTML Table width:

We can specify the HTML table width using the **CSS width** property. It can be specify in pixels or percentage.

We can adjust our table width as per our requirement. Following is the example to display table with width.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<title>**table**</title>**

**<style>**

        table{

             border-collapse: collapse;

             width: 100%;

        }

th,td{

        border: 2px solid green;

        padding: 15px;

    }

**</style>**

**</head>**

**<body>**

**<table>**

**<tr>**

**<th>**1 header**</th>**

**<th>**1 header**</th>**

**<th>**1 header**</th>**

**</tr>**

**<tr>**

**<td>**1data**</td>**

**<td>**1data**</td>**

**<td>**1data**</td>**

**</tr>**

**<tr>**

**<td>**2 data**</td>**

**<td>**2 data**</td>**

**<td>**2 data**</td>**

**</tr>**

**<tr>**

**<td>**3 data**</td>**

**<td>**3 data**</td>**

**<td>**3 data**</td>**

**</tr>**

**</table>**

**</body>**

**</html>**

## HTML Table with colspan

If you want to make a cell span more than one column, you can use the colspan attribute.

It will divide one cell/row into multiple columns, and the number of columns depend on the value of colspan attribute.

Let's see the example that span two columns.<!DOCTYPE>

<html>

<head>

<style>

table, th, td {

border: 1px solid black;

border-collapse: collapse;

}

th, td {

padding: 5px;

text-align: left;

}

</style>

</head>

<body>

<table style="width:100%">

<tr>

<th>Name</th>

<th colspan="2">Mobile No.</th>

</tr>

<tr>

<td>Ajeet Maurya</td>

<td>7503520801</td>

<td>9555879135</td>

</tr>

</table>

</body>

</html>

## HTML Table with rowspan

If you want to make a cell span more than one row, you can use the rowspan attribute.

It will divide a cell into multiple rows. The number of divided rows will depend on rowspan values.

Let's see the example that span two rows.

<!DOCTYPE>

<html>

<head>

<style>

table, th, td {

border: 1px solid black;

border-collapse: collapse;

}

th, td {

padding: 10px;

}

</style>

</head>

<body>

<table>

<tr><th>Name</th><td>Ajeet Maurya</td></tr>

<tr><th rowspan="2">Mobile No.</th><td>7503520801</td></tr>

<tr><td>9555879135</td></tr>

</table>

</body>

</html>

## HTML table with caption

HTML caption is displayed above the table. It must be used after table tag only.

<!DOCTYPE>

<html>

<head>

<style>

table, th, td {

border: 1px solid black;

border-collapse: collapse;

}

th, td {

padding: 10px;

}

</style>

</head>

<body>

<table>

<caption>Student Records</caption>

<tr><th>First\_Name</th><th>Last\_Name</th><th>Marks</th>

</tr>

<tr><td>Vimal</td><td>Jaiswal</td><td>70</td></tr>

<tr><td>Mike</td><td>Warn</td><td>60</td></tr>

<tr><td>Shane</td><td>Warn</td><td>42</td></tr>

<tr><td>Jai</td><td>Malhotra</td><td>62</td></tr>

</table>

</body>

</html>

## Styling HTML table even and odd cells

<!DOCTYPE>

<html>

<head>

<style>

table, th, td {

border: 1px solid black;

border-collapse: collapse;

}

th, td {

padding: 10px;

}

table#alter tr:nth-child(even) {

background-color: #eee;

}

table#alter tr:nth-child(odd) {

background-color: #fff;

}

table#alter th {

color: white;

background-color: gray;

}

</style>

</head>

<body>

<table id="alter">

<tr><th>First\_Name</th><th>Last\_Name</th><th>Marks</th></tr>

<tr><td>Sonoo</td><td>Jaiswal</td><td>60</td></tr>

<tr><td>James</td><td>William</td><td>80</td></tr>

<tr><td>Swati</td><td>Sironi</td><td>82</td></tr>

<tr><td>Chetna</td><td>Singh</td><td>72</td></tr>

</table>

</body>

# </html>

# **HTML style using CSS**

Let's suppose we have created our web page using a simple HTML code, and we want something which can present our page in a correct format, and visibly attractive. So to do this, we can style our web page with CSS (Cascading Stylesheet) properties.

CSS is used to apply the style in the web page which is made up of HTML elements. It describes the look of the webpage.

CSS provides various style properties such as background color, padding, margin, border-color, and many more, to style a webpage.

Each property in CSS has a name-value pair, and each property is separated by a semicolon (;).

<!DOCTYPE html>

<html>

<head>

<title></title>

</head>

<body style="text-align: center;">

<h2 style="color: red;">Welcome to iHub institute</h2>

<p style="color: blue; font-size: 25px; font-style: italic;">This is a great institute to learn technologies in very simple way.</p>

</body>

</html>

Three ways to apply CSS

To use CSS with HTML document, there are three ways:

* **Inline CSS:** Define CSS properties using style attribute in the HTML elements.
* **Internal or Embedded CSS:** Define CSS using <style> tag in <head> section.
* **External CSS:** Define all CSS property in a separate .css file, and then include the file with HTML file using tag in section.

## Inline CSS:

Inline CSS is used to apply CSS in a single element. It can apply style uniquely in each element.

To apply inline CSS, you need to use style attribute within HTML element. We can use as many properties as we want, but each property should be separated by a semicolon (;).

<!DOCTYPE html>

<html>

<head>

<title></title>

</head>

<body>

<h3 style="color: red;

font-style: italic;

text-align: center;

font-size: 50px;

padding-top: 25px;">Learning HTML using Inline CSS</h3>

</body>

</html>

## Internal CSS:

An Internal stylesheets contains the CSS properties for a webpage in <head> section of HTML document. To use Internal CSS, we can use class and id attributes.

We can use internal CSS to apply a style for a single HTML page.

### **Example:**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. /\*Internal CSS using element name\*/
6. body{background-color:lavender;
7. text-align: center;}
8. h2{font-style: italic;
9. font-size: 30px;
10. color: #f08080;}
11. p{font-size: 20px;}
12. /\*Internal CSS using class name\*/
13. .blue{color: blue;}
14. .red{color: red;}
15. .green{color: green;}
16. **</style>**
17. **</head>**
18. **<body>**
19. **<h2>**Learning HTML with internal CSS**</h2>**
20. **<p** class="blue"**>**This is a blue color paragraph**</p>**
21. **<p** class="red"**>**This is a red color paragraph**</p>**
22. **<p** class="green"**>**This is a green color paragraph**</p>**
23. **</body>**
24. **</html>**

## External CSS:

An external CSS contains a separate CSS file which only contains style code using the class name, id name, tag name, etc. We can use this CSS file in any HTML file by including it in HTML file using <link> tag.

If we have multiple HTML pages for an application and which use similar CSS, then we can use external CSS.

There are two files need to create to apply external CSS

* First, create the HTML file
* Create a CSS file and save it using the .css extension (This file only will only contain the styling code.)
* Link the CSS file in your HTML file using tag in header section of HTML document.

### **Example:**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<link** rel="stylesheet" type="text/css" href="style.css"**>**
5. **</head>**
6. **<body>**
7. **<h2>**Learning HTML with External CSS**</h2>**
8. **<p** class="blue"**>**This is a blue color paragraph**</p>**
9. **<p** class="red"**>**This is a red color paragraph**</p>**
10. **<p** class="green"**>**This is a green color paragraph**</p>**
11. **</body>**
12. **</html>**

**CSS file:**

body{  
background-color:lavender;  
text-align:center;  
}  
h2{  
font-style:italic;  
size:30px;  
color:#f08080;  
}  
p{  
font-size:20px;  
}  
  
.blue{  
color:blue;  
}  
.red{  
color:red;  
}  
.green{  
color:green;  
}

## Commonly used CSS properties:

|  |  |  |
| --- | --- | --- |
| **Properties-name** | **Syntax** | **Description** |
| background-color | background-color:red; | It defines the background color of that element. |
| color | color: lightgreen; | It defines the color of text of an element |
| padding | padding: 20px; | It defines the space between content and the border. |
| margin | margin: 30px; margin-left: | It creates space around an element. |
| font-family | font-family: cursive; | Font-family defines a font for a particular element. |
| Font-size | font-size: 50px; | Font-size defines a font size for a particular element. |
| text-align | text-align: left; | It is used to align the text in a selected position. |

# **HTML Classes**

## Class Attribute in HTML

The HTML class attribute is used to specify a single or multiple class names for an HTML element. The class name can be used by CSS and JavaScript to do some tasks for HTML elements. You can use this class in CSS with a specific class, write a period (.) character, followed by the name of the class for selecting elements.

A class attribute can be defined within <style> tag or in separate file using the (.) character.

In an HTML document, we can use the same class attribute name with different elements.

## Defining an HTML class

To create an HTML class, firstly define style for HTML class using <style> tag within <head> section as following example:

### **Example:**

1. **<head>**
2. **<style>**
3. .headings{
4. color: lightgreen;
5. font-family: cursive;
6. background-color: black; }
7. **</style>**
8. **</head>**

We have define style for a class name "headings", and we can use this class name with any of HTML element in which we want to provide such styling. We just need to follow the following syntax to use it.

Syntax: **<tag** class="ghf"**>** content **</tag>**

### **Example 1:**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. .headings{
6. color: lightgreen;
7. font-family: cursive;
8. background-color: black; }
9. **</style>**
10. **</head>**
11. **<body>**
12. **<h1** class="headings"**>**This is first heading**</h1>**
13. **<h2** class="headings"**>**This is Second heading**</h2>**
14. **<h3** class="headings"**>**This is third heading**</h3>**
15. **<h4** class="headings"**>**This is fourth heading**</h4>**
16. **</body>**
17. **</html>**

## Another Example with different class name

### **Example:**

Let's use a class name "Fruit" with CSS to style all elements.

1. **<style>**
2. .fruit {
3. background-color: orange;
4. color: white;
5. padding: 10px;
6. }
7. **</style>**
9. **<h2** class="fruit"**>**Mango**</h2>**
10. **<p>**Mango is king of all fruits.**</p>**
12. **<h2** class="fruit"**>**Orange**</h2>**
13. **<p>**Oranges are full of Vitamin C.**</p>**
15. **<h2** class="fruit"**>**Apple**</h2>**
16. **<p>**An apple a day, keeps the Doctor away.**</p>**

#### **Note**: You can use class attribute on any HTML element. The class name is case-sensitive.

## Class Attribute in JavaScript

You can use JavaScript access elements with a specified class name by using the getElementsByClassName() method.

### **Example:**

Let's hide all the elements with class name "fruit" when the user click on the button.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<body>**
5. **<h2>**Class Attribute with JavaScript**</h2>**
6. **<p>**Click the button, to hide all elements with the class name "fruit", with JavaScript:**</p>**
8. **<button** onclick="myFunction()"**>**Hide elements**</button>**

11. **<h2** class="fruit"**>**Mango**</h2>**
12. **<p>**Mango is king of all fruits.**</p>**
14. **<h2** class="fruit"**>**Orange**</h2>**
15. **<p>**Oranges are full of Vitamin C.**</p>**
17. **<h2** class="fruit"**>**Apple**</h2>**
18. **<p>**An apple a day, keeps the Doctor away.**</p>**
20. **<script>**
21. function myFunction() {
22. var x = document.getElementsByClassName("fruit");
23. for (var i = 0; i **<** **x.length**; i++) {
24. x[i].style.display = "none";
25. }
26. }
27. **</script>**
29. **</body>**
30. **</html>**

## Multiple Classes

You can use multiple class names (more than one) with HTML elements. These class names must be separated by a space.

### **Example:**

Let's style elements with class name "fruit" and also with a class name "center".

1. <!DOCTYPE html**>**
2. **<html>**
3. **<style>**
4. .fruit {
5. background-color: orange;
6. color: white;
7. padding: 10px;
8. }
10. .center {
11. text-align: center;
12. }
13. **</style>**
14. **<body>**
16. **<h2>**Multiple Classes**</h2>**
17. **<p>**All three elements have the class name "fruit". In addition, Mango also have the class name "center", which center-aligns the text.**</p>**
19. **<h2** class="fruit center"**>**Mango**</h2>**
20. **<h2** class="fruit"**>**Orange**</h2>**
21. **<h2** class="fruit"**>**Apple**</h2>**
23. **</body>**
24. **</html>**

## Same class with Different Tag

You can use the same class name with different tags like <h2> and <p> etc. to share the same style.

### **Example:**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<style>**
4. .fruit {
5. background-color: orange;
6. color: white;
7. padding: 10px;
8. }
9. **</style>**
10. **<body>**
11. **<h2>**Same Class with Different Tag**</h2>**
12. **<h2** class="fruit"**>**Mango**</h2>**
13. **<p** class="fruit"**>**Mango is the king of all fruits.**</p>**
14. **</body>**
15. **</html>**

# **HTML Id Attribute**

The **id attribute** is used to specify the unique ID for an element of the HTML document. It allocates the unique identifier which is used by the **CSS** and the **JavaScript** for performing certain tasks.

#### **Note**: In the Cascading Style sheet (CSS), we can easily select an element with the specific id by using the # symbol followed by id.

#### **Note**: JavaScript can access an element with the given ID by using the getElementById() method.

1. Syntax: **<tag** id="value"**>**

**Example 1:** The following example describes how to use the id attribute in CSS document:

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**
5. Example of Id attribute in CSS
6. **</title>**
7. **<style>**
8. #Cars {
9. padding: 40px;
10. background-color: lightblue;
11. color: black;
12. text-align: center;
13. }
15. #Bikes
16. {
17. padding: 50px;
18. background-color: lightGreen;
19. text-align: center;
20. }
21. **</style>**
22. **</head>**
23. **<body>**
24. **<p>** Use CSS to style an element with the id: **</p>**
25. **<h1** id="Cars"**>** Cars **</h1>**
26. **<h1** id="Bikes"**>** Bikes **</h1>**
27. **</body>**
28. **</html>**

# **Example 2:** The following example describes how to use the ID attribute in JavaScript.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>** Date Attribute **</title>**
5. **<script>**
6. function viewdate() {
7. var x = document.getElementById("dob").value;
8. document.getElementById("demo").innerHTML = x;
9. **</script>**
10. **</head>**
11. **<body>**
12. Employee Name: **<input** type="text" placeholder="Your Good name"**/>**
13. **<br>**
14. **<br>**
15. Date of Joining:
16. **<input** type="date" id="dob"**>**
17. **<br>**
18. **<button** onclick="viewdate()"**>** Submit
19. **</button>**
20. **<br>**
21. **<h2** id="demo"**>** **</h2>**
22. **</body>**
23. **</html>**

# **HTML List Box**

The **list box** is a graphical control element in the HTML document that allows a user to select one or more options from the list of options.

### **Syntax**

To create a list box, use the [HTML element](https://www.javatpoint.com/html-elements) **<select>** which contains two attributes **Name** and **Size**. The **Name** attribute is used to define the name for calling the list box, and **size** attribute is used to specify the numerical value that shows the how many options it contains.

1. **<select** Name="Name\_of\_list\_box" Size="Number\_of\_options"**>**
2. **<option>** List item 1 **</option>**
3. **<option>** List item 2 **</option>**
4. **<option>** List item 3 **</option>**
5. **<option>** List item N **</option>**
6. **</select>**

### **Examples:**

**Example 1:** Consider the below example that creates a simple list box.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<title>**
4. Example of List Box
5. **</title>**
6. **<body>**
7. Customer Name:  **<input** type="text" Placeholder="Enter the Customer Name"**/>**
8. **<br>**
9. **<br>**
10. **<select** name="Cars" size="5"**>**
11. **<option** value="Merceders"**>** Merceders **</option>**
12. **<option** value="BMW"**>** BMW **</option>**
13. **<option** value="Jaguar"**>** Jaguar **</option>**
14. **<option** value="Lamborghini"**>** Lamborghini **</option>**
15. **<option** value="Ferrari"**>** Ferrari **</option>**
16. **<option** value="Ford"**>** Ford **</option>**
17. **</select>**
18. **</body>**
19. **</html>**

**Example 2:** Below example uses the **multiple** attribute for selecting the multiple options in a list. We can select multiple options from list box by holding the ctrl key.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<title>**
4. Example of List Box with multiple attribute
5. **</title>**
6. **<body>**
7. Customer Name:  **<input** type="text" Placeholder="Enter the Customer Name"**/>**
8. **<br>**
9. **<br>**
10. **<select** name="Cars" size="5" multiple="multiple"**>**
11. **<option** value="Merceders"**>** Merceders **</option>**
12. **<option** value="BMW"**>** BMW **</option>**
13. **<option** value="Jaguar"**>** Jaguar **</option>**
14. **<option** value="Lamborghini"**>** Lamborghini **</option>**
15. **<option** value="Ferrari"**>** Ferrari **</option>**
16. **<option** value="Ford"**>** Ford **</option>**
17. **</select>**
18. **</body>**
19. **</html>**

# **Add JavaScript to HTML**

There are following three ways through which we can add the JavaScript code into the HTML document:

1. Include the JavaScript code in <head>…</head> tag.
2. Include the JavaScript code between the <Body> …</Body> tag and after the closing of the body tag.
3. Link the separate file of JavaScript in HTML

## Include the JavaScript Code in <head> tag.

In this section, you will learn to include the JavaScript code between the <head> and </head> tag.

**Syntax**

## Include the JavaScript Code in <head> tag.

In this section, you will learn to include the JavaScript code between the <head> and </head> tag.

**Syntax**

1. **<html>**
2. **<head>**
3. **<script>**
4. JavaScript Code
5. Statement 1
6. Statement 2
7. ......
8. Statement N
9. **</script>**
10. **</head>**
11. **<body>**
12. **</body>**
13. **</html>**

In the above syntax, the [JavaScript](https://www.javatpoint.com/javascript-tutorial) code written between the <script>……. </script> tag is put between the <head> and </head> tag in HTML file.

**Example**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<meta** name="viewport" content="width=device-width, initial-scale=1"**>**
5. **<title>**
6. Include JavaScript in head tag
7. **</title>**
9. **<script>**
10. function check()
11. {
12. /\* The following statement is used to display a Confirm dialog box on a webpage with the statement which is enclosed in the brackets. \*/
13. confirm("Your JavaScript Code Run");
14. }
15. **</script>**
16. **<style>**
17. /\* The following tag selector button use the different properties for the Button. \*/
18. button {
19. background-color: red;
20. padding: 16px 20px;
21. margin: 8px 0;
22. border: none;
23. cursor: pointer;
24. color: white;
25. width: 100%;
26. }
27. **</style>**
28. **</head>**
29. **<body>**
30. **<form>**
31. <!-- The following statement use the Button type which is used to call a function of JavaScript when this button is clicked. -->
32. **<button** type="button" onclick="check()"**>** Click Me for running a JavaScript Code **</button>**
34. **</form>**
35. **</body>**
36. **</html>**

## 2. i) Include the JavaScript code in the <body> tag.

In this section, you will learn about how to include the JavaScript code in-between the <body> and </body> tag.

**Syntax**

1. **<html>**
2. **<head>**
3. **</head>**
4. **<body>**
6. **<script>**
7. JavaScript Code
8. Statement 1
9. Statement 2
10. ......
11. Statement N
12. **</script>**
13. **</body>**
14. **</html>**

In the above syntax, the JavaScript code written between the <script>……. </script> tag is put in-between of the <body> and </body> tag in HTML file.

**Example**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<meta** name="viewport" content="width=device-width, initial-scale=1"**>**
5. **<title>**
6. Include JavaScript in body tag
7. **</title>**
8. **<style>**
9. /\* The following tag selector button use the different properties for the Button. \*/
10. button {
11. background-color: red;
12. padding: 16px 20px;
13. margin: 8px 0;
14. border: none;
15. cursor: pointer;
16. color: white;
17. width: 100%;
18. }
19. **</style>**
20. **</head>**
21. **<body>**
22. **<form>**
23. **<script>**
24. function bdy\_JS ()
25. {
26. /\* The following statement is used to display a Confirm dialog box on a webpage with the statement which is enclosed in the brackets. \*/
27. confirm("Your JavaScript Code Run which is used in the Body tag");
28. }
29. **</script>**
31. <!-- The following statement use the Button type which is used to call a function of JavaScript when this button is clicked. -->
32. **<button** type="button" onclick="bdy\_JS()"**>** Click Me for running a JavaScript Code **</button>**
33. **</form>**
34. **</body>**
35. **</html>**

## ii) Include the JavaScript code after the <body> tag.

In this section, you will learn to include the JavaScript code after the <body> tag.

**Syntax**

1. **html>**
2. **<head>**
3. **</head>**
4. **<body>**
5. **</body>**
6. **<script>**
7. JavaScript Code
8. Statement 1
9. Statement 2
10. ......
11. Statement N
12. **</script>**
14. **</html>**
15. In the above syntax, the JavaScript code written between the <script>……. </script> tag is put after the <body>…</body> tag in HTML file.

**Example**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<meta** name="viewport" content="width=device-width, initial-scale=1"**>**
5. **<title>**
6. Include JavaScript code after the body tag
7. **</title>**
8. **<style>**
9. /\* The following tag selector button use the different properties for the Button. \*/
10. button {
11. background-color: red;
12. color: white;
13. margin: 8px 0;
14. border: none;
15. cursor: pointer;
16. width: 100%;
17. padding: 16px 20px;
18. }
19. **</style>**
20. **</head>**
21. **<body>**
22. **<form>**
23. <!-- The following statement use the Button type which is used to call a function of JavaScript when this button is clicked. -->
24. **<button** type="button" onclick="bdy\_JS()"**>** Click Me for running a JavaScript Code **</button>**
26. **</form>**
27. **</body>**
28. **<script>**
29. function bdy\_JS ()
30. {
31. /\* The following statement is used to display a Confirm dialog box on a webpage with the statement which is enclosed in the brackets. \*/
32. confirm("Your JavaScript Code Run which is used after the Body tag");
33. }
34. **</script>**
35. **</html>**

## Link the Separate file of JavaScript in HTML

In this section, you will learn to include the file of JavaScript code in the HTML file.

**Syntax**

1. **<html>**
2. **<head>**
3. **<script** src="Name\_of\_JavaScript\_file**>**
4. **</script>**
5. **</head>**
6. **<body>**
7. **</body>**
8. **</html>**

In the above syntax, the **src** is an attribute of <script> tag used for specifying the name of the JavaScript file.

**Example**

The following code is written in JavaScript (inc), which is saved by **.js** extension.

1. function funcjs ()
2. {
3. /\* The following statement displays a Confirm dialog box on a webpage with the statement which is enclosed in the brackets. \*/
4. confirm("Your JavaScript Code Run which is used after the Body tag");
5. }

The following code is written in an HTML file which uses the src attribute in the <script> tag to specify the above JavaScript file.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<meta** name="viewport" content="width=device-width, initial-scale=1"**>**
5. **<title>**
6. Include JavaScript file in head tag of HTML file
7. **</title>**
8. **<script** src="inc.js"**>**
9. **</script>**
10. **<style>**
11. /\* The following tag selector button use the different properties for the Button. \*/
12. button {
13. background-color: red;
14. color: white;
15. margin: 8px 0;
16. border: none;
17. cursor: pointer;
18. width: 100%;
19. padding: 16px 20px;
20. }
21. **</style>**
22. **</head>**
23. **<body>**
24. **<form>**
25. <!-- The following statement use the Button type which is used to call a function of JavaScript when this button is clicked. -->
27. **<button** type="button" onclick="funcjs()"**>** Click Me for running a JavaScript Code **</button>**
28. **</form>**
29. **</body>**
30. **</html>**

# **HTML Background Image**

The **<background>** attribute in the **HTML** document is used to specify the background image on a HTML page or a table. You can pass the path of an image as a value of **background attribute** to set the image of your HTML page or table.

### **Syntax**

**<tag** background="Path\_of\_an\_image"**>**

1. **<tag** style="background-image:Path\_of\_an\_image)"**>**  (Using CSS)

**Examples:** The following examples set the background image in different styles:

**Example 1:** This example uses the **background** attribute with the [body tag](https://www.javatpoint.com/html-body-tag) for displaying the image as a background of the web page.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**
5. Example of Background image
6. **</title>**
7. **</head>**
8. **<body** background="https://previews.123rf.com/images/kotenko/kotenko1403/kotenko140300008/26854631-snowy-winter-in-a-mountain-forest-sunny-cold-day-with-snow-covered-trees-carpathian-mountains-ukrain.jpg"**>**
10. **<font** color="red"**>**
11. **<h1>** click on this **<a** href="https://www.amazon.in/"**>** link **</a>** for the Home page of Our Website **</h1>**
12. **</font>**
13. **</body>**
14. **</html>**

**Example 2:** This example uses the **background** attribute with the [table tag](https://www.javatpoint.com/html-table) for displaying the image as a background of [html](https://www.javatpoint.com/html-tutorial) table.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**
5. Example of Background image
6. **</title>**
7. **</head>**
8. **<body>**
9. **<center>**
11. In this example, we use the background attribute with the table tag, which displays the image as a background of the HTML table.
13. **<table** background="https://images.pexels.com/photos/237272/pexels-photo-237272.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500" height="300" border="1" width="500"**>**
15. **<tr>**
16. **<td>** Roll No. **</td>**
17. **<td>** Name **</td>**
18. **</tr>**
19. **<tr>**
20. **<td>** 101 **</td>**
21. **<td>** Abhay **</td>**
22. **</tr>**
23. **<tr>**
24. **<td>** 102 **</td>**
25. **<td>** Chetan **</td>**
26. **</tr>**
27. **<tr>**
28. **<td>** 103 **</td>**
29. **<td>** Manpreet **</td>**
30. **</tr>**
31. **<tr>**
32. **<td>** 104 **</td>**
33. **<td>** Rakesh **</td>**
34. **</tr>**
35. **<tr>**
36. **<td>** 105 **</td>**
37. **<td>** Sumit **</td>**
38. **</tr>**
39. **</table>**
40. **</center>**
41. **</body>**
42. **</html>**

**Example 3:** This example uses the [CSS syntax](https://www.javatpoint.com/css-syntax) for the **background-image** property of the style attribute with the body tag.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**
5. Example of Background image
6. **</title>**
7. **</head>**
9. **<body** style="background-image:url('https://previews.123rf.com/images/kotenko/kotenko1403/kotenko140300008/26854631-snowy-winter-in-a-mountain-forest-sunny-cold-day-with-snow-covered-trees-carpathian-mountains-ukrain.jpg');"**>**
11. **<font** size="4" color="green"**>**
13. In this example, we use the **<b>** background-image **</b>** property in the **<b>**style**</b>** attribute with the **<b>** body tag **</b>** which display the image as a background on the web page.
15. **</font>**
16. **<center>**
17. **<font** size="4" color="red"**>**
18. **<table** height="300" border="1" width="500"**>**
19. **<tr>**
20. **<td>** Roll No. **</td>**
21. **<td>** Name **</td>**
22. **</tr>**
23. **<tr>**
24. **<td>** 101 **</td>**
25. **<td>** Abhay **</td>**
26. **</tr>**
27. **<tr>**
28. **<td>** 102 **</td>**
29. **<td>** Chetan **</td>**
30. **</tr>**
31. **<tr>**
32. **<td>** 103 **</td>**
33. **<td>** Manpreet **</td>**
34. **</tr>**
35. **<tr>**
36. **<td>** 104 **</td>**
37. **<td>** Rakesh **</td>**
38. **</tr>**
39. **<tr>**
40. **<td>** 105 **</td>**
41. **<td>** Sumit **</td>**
42. **</tr>**
43. **</table>**
44. **</font>**
45. **</center>**
46. **</body>**
47. **</html>**

**Example 4:** This example uses the [CSS](https://www.javatpoint.com/css-tutorial) syntax for the background-image property in the style attribute with the table tag.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**
5. Example of Background image
6. **</title>**
7. **</head>**
8. **<body>**
9. **<font** size="4" color="orange"**>**
10. In this example, we use the **<b>** background-image property **</b>** in the **<b>** style attribute **</b>** with the **<b>** table tag **</b>** which display the image as a background of HTML table.
11. **</font>**
12. **<font** size="4" color="red"**>**
13. **<center>**
15. **<table** style="background-image:url('https://images.pexels.com/photos/237272/pexels-photo-237272.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500');"**>**
16. **<tr>**
17. **<td>** Roll No. **</td>**
18. **<td>** Name **</td>**
19. **</tr>**
21. **<tr>**
22. **<td>** 101 **</td>**
23. **<td>** Abhay **</td>**
24. **</tr>**
25. **<tr>**
26. **<td>** 102 **</td>**
27. **<td>** Chetan **</td>**
28. **</tr>**
29. **<tr>**
30. **<td>** 103 **</td>**
31. **<td>** Manpreet **</td>**
32. **</tr>**
33. **<tr>**
34. **<td>** 104 **</td>**
35. **<td>** Rakesh **</td>**
36. **</tr>**
37. **<tr>**
38. **<td>** 105 **</td>**
39. **<td>** Sumit **</td>**
40. **</tr>**
41. **</table>**
42. **</center>**
43. **</font>**
44. **</body>**
45. **</html>**

**Example 5:** This example uses the CSS syntax for the **background-repeat and background-size** property of the style attribute with the body tag.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**
5. Example of Background image
6. **</title>**
7. **</head>**
9. **<body** style="background-image:url('https://images.pexels.com/photos/237272/pexels-photo-237272.jpeg?auto=compress&cs=tinysrgb&dpr=1&w=500');
10. background-repeat:no-repeat; background-size: 100% 100%"**>**
12. **<font** size="4" color="green"**>**
14. In this example, we use the **<b>** background-repeat and background-size property **</b>** in the **<b>** style attribute **</b>** with the **<b>** body **</b>** tag
15. which display the image without repeatition.
17. **</font>**
18. **<center>**
20. **<font** size="4" color="red"**>**
21. **<table** height="300" border="1" width="500"**>**
22. **<tr>**
23. **<td>** Roll No. **</td>**
24. **<td>** Name **</td>**
25. **</tr>**
26. **<tr>**
27. **<td>** 101 **</td>**
28. **<td>** Abhay **</td>**
29. **</tr>**
30. **<tr>**
31. **<td>** 102 **</td>**
32. **<td>** Chetan **</td>**
33. **</tr>**
34. **<tr>**
35. **<td>** 103 **</td>**
36. **<td>** Manpreet **</td>**
37. **</tr>**
38. **<tr>**
39. **<td>** 104 **</td>**
40. **<td>** Rakesh **</td>**
41. **</tr>**
42. **<tr>**
43. **<td>** 105 **</td>**
44. **<td>** Sumit **</td>**
45. **</tr>**
46. **</table>**
47. **</font>**
48. **</center>**
49. **</body>**
50. **</html>**

# **HTML iframes**

# **HTML iframes**

HTML Iframe is used to display a nested webpage (a webpage within a webpage). The HTML <iframe> tag defines an inline frame, hence it is also called as an Inline frame.

An HTML iframe embeds another document within the current HTML document in the rectangular region.

The webpage content and iframe contents can interact with each other using JavaScript.

### **Iframe Syntax**

An HTML iframe is defined with the <iframe> tag:

1. <iframe src="URL"></iframe>

Here, "src" attribute specifies the web address (URL) of the inline frame page.

## Set Width and Height of iframe

You can set the width and height of iframe by using "width" and "height" attributes. By default, the attributes values are specified in pixels but you can also set them in percent. i.e. 50%, 60% etc.

### **Example: (Pixels)**

1. <!DOCTYPE html>
2. <html>
3. <body>
4. <h2>HTML Iframes example</h2>
5. <p>Use the height and width attributes to specify the size of the iframe:</p>
6. <iframe src="https://www.flipkart.com/" height="300" width="400"></iframe>
7. </body>
8. </html>

### **Example: (Percentage)**

1. <!DOCTYPE html>
2. <html>
3. <body>
4. <h2>HTML Iframes</h2>
5. <p>You can use the height and width attributes to specify the size of the iframe:</p>
6. <iframe src="https://www.flipkart.com/" height="50%" width="70%"></iframe>
7. </body>
8. </html>

You can also use CSS to set the height and width of the iframe.

### **Example:**

1. <!DOCTYPE html>
2. <html>
3. <body>
4. <h2>HTML Iframes</h2>
5. <p>Use the CSS height and width properties to specify the size of the iframe:</p>
6. <iframe src="https://www.flipkart.com/" style="height:300px;width:400px"></iframe>    </body>
7. </html>

## Remove the border of iframe

By default, an iframe contains a border around it. You can remove the border by using <style> attribute and CSS border property.

1. <!DOCTYPE html>
2. <html>
3. <body>
4. <h2>Remove the Iframe Border</h2>
5. <p>This iframe example doesn't have any border</p>
6. <iframe src="https://www.flipkart.com/" style="border:none;"></iframe>
7. </body>
8. </html>

You can also change the size, color, style of the iframe's border.

### **Example:**

1. <!DOCTYPE html>
2. <html>
3. <body>
4. <h2>Custom Iframe Border</h2>
5. <iframe src="https://www.flipkart.com/" style="border:2px solid tomato;"></iframe>
6. </body>
7. </html>

## Iframe Target for a link

You can set a target frame for a link by using iframe. Your specified target attribute of the link must refer to the name attribute of the iframe.

### **Example:**

1. <!DOCTYPE html>
2. <html>
3. <body>
5. <h2>Iframe - Target **for** a Link</h2>
6. <iframe height="300px" width="100%" src="new.html" name="iframe\_a"></iframe>
7. <p><a href="https://www.flipkart.com" target="iframe\_a">flipkart.com</a></p>
8. <p>The name of iframe and link target must have same value **else** link will not open as a frame. </p>
10. </body>
11. </html>

**new.hmtl output code:**

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. p{ font-size: 50px;
6. color: red;}
7. </style>
8. </head>
9. <body style="background-color: #c7f15e;">
10. <p>This is a link below the ifarme click on link to open new iframe. </p>
11. </body>
12. </html>

## Embed YouTube video using iframe

You can also add a YouTube video on your webpage using the <iframe> tag. The attached video will be played at your webpage and you can also set height, width, autoplay, and many more properties for the video.

Following are some steps to add YouTube video on your webpage:

* Goto YouTube video which you want to embed.
* Click on SHARE ➦ under the video.
* Click on Embed <> option.
* Copy HTML code.
* Paste the code in your HTML file
* Change height, width, and other properties (as per requirement).

<!DOCTYPE html>

<html>

<head>

<style>

h3{text-align: center;

font-size: 50px;

color: red;}

p{font-size: 20px;

color: green;

font-weight: bold;}

</style>

</head>

<body style="background-color: #f0f8ff">

<h3>Play videos using iframe</h3>

<iframe width="550" height="315" src="https://www.youtube.com/ " frameborder="0" allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-in-picture" allowfullscreen style="padding:20px;"></iframe>

<iframe width="550" height="315" src="https://www.youtube.com/ " frameborder="0" allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-in-picture" style="padding:20px;">></iframe>

<p>In first video full screen is available and in second video full screen is not available</p>

</body>

</html>

## Attributes of <iframe>

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Value** | **Description** |
| allowfullscreen |  | If true then that frame can be opened in full screen. |
| height | Pixels | It defines the height of the embedded iframe, and the default height is 150 px. |
| name | text | It gives the name to the iframe. The name attribute is important if you want to create a link in one frame. |
| frameborder | 1 or 0 | It defines whether iframe should have a border or not. (Not supported in HTML5). |
| Width | Pixels | It defines the width of embedded frame, and default width is 300 px. |
| src | URL | The src attribute is used to give the path name or file name which content to be loaded into iframe. |
| sandbox |  |  |
|  | This attribute is used to apply extra restrictions for the content of the frame |
| allow-forms | It allows submission of the form if this keyword is not used then form submission is blocked. |
| allow-popups | It will enable popups, and if not applied then no popup will open. |
| allow-scripts | It will enable the script to run. |
| allow-same-origin | If this keyword is used then the embedded resource will be treated as downloaded from the same source. |
| srcdoc |  | The srcdoc attribute is used to show the HTML content in the inline iframe. It overrides the src attribute (if a browser supports). |
| scrolling |  |  |
|  | It indicates that browser should provide a scroll bar for the iframe or not. (Not supported in HTML5) |
| auto | Scrollbar only shows if the content of iframe is larger than its dimensions. |
| yes | Always shows scroll bar for the iframe. |
| no | Never shows scrollbar for the iframe. |

# **HTML JavaScript**

A Script is a small program which is used with HTML to make web pages more attractive, dynamic and interactive, such as an alert popup window on mouse click. Currently, the most popular scripting language is JavaScript used for websites.

### **Example:**

1. <!DOCTYPE html>
2. <html>
3. <body>
4. <h1>JavaScript Date and Time example</h1>
5. <button type="button"
6. onclick="document.getElementById('demo').innerHTML = Date()">
7. Click me to display Date and Time.</button>
8. <p id="demo"></p>
9. </body>
10. </html>

## HTML <script> Tag

The HTML <script> tag is used to specify a client-side script. It may be an internal or external JavaScript which contains scripting statements, hence we can place <script> tag within <body> or <head> section.

It is mainly used to manipulate images, form validation and change content dynamically. JavaScript uses document.getElementById() method to select an HTML element.

1. <!DOCTYPE html>
2. <html>
3. <body>
4. <h2>Use JavaScript to Change Text</h2>
5. <p id="demo"></p>
6. <script>
7. document.getElementById("demo").innerHTML = "Hello iHub Students";
8. </script>
9. </body>
10. </html>

## HTML events with JavaScript

An event is something which user does, or browser does such as mouse click or page loading are examples of events, and JavaScript comes in the role if we want something to happen on these events.

HTML provides event handler attributes which work with JavaScript code and can perform some action on an event.

### **Syntax:**

1. <element event = "JS code">

Example:

1. <!DOCTYPE html>
2. <html>
3. <body>
4. <h2>Click Event Example</h2>
5. <p>Click on the button and you csn see a pop-up window with a message</p>
6. <input type="button" value="Click" onclick="alert('Hi,how are you')">
7. </body>
8. </html>

HTML can have following events such as:

* **Form events:** reset, submit, etc.
* **Select events:** text field, text area, etc.
* **Focus event:** focus, blur, etc.
* **Mouse events:** select, mouseup, mousemove, mousedown, click, dblclick, etc.

Following are the list for Window event attributes:

|  |  |  |
| --- | --- | --- |
| **Event Event Name** | **Handler Name** | **Occurs when** |
| onBlur | blur | When form input loses focus |
| onClick | click | When the user clicks on a form element or a link |
| onSubmit | submit | When user submits a form to the server. |
| onLoad | load | When page loads in a browser. |
| onFocus | focus | When user focuses on an input field. |
| onSelect | select | When user selects the form input filed. |

Let's see what JavaScript can do:

**1) JavaScript can change HTML content.**

### **Example:**

1. <!DOCTYPE html>
2. <html>
3. <body>
4. <p>JavaScript can change the content of an HTML element:</p>
5. <button type="button" onclick="myFunction()">Click Me!</button>
6. <p id="demo"></p>
7. <script>
8. function myFunction() {
9. document.getElementById("demo").innerHTML = "Hello iHub Students!";
10. }
11. </script>
12. </body>
13. </html>

**2) JavaScript can change HTML style**

### **Example:**

1. <!DOCTYPE html>
2. <html>
3. <body>
4. <p id="demo">JavaScript can change the style of an HTML element.</p>
5. <script>
6. function myFunction() {
7. document.getElementById("demo").style.fontSize = "25px";
8. document.getElementById("demo").style.color = "brown";
9. document.getElementById("demo").style.backgroundColor = "lightgreen";
10. }
11. </script>
12. <button type="button" onclick="myFunction()">Click Me!</button>
13. </body>
14. </html>

**3) JavaScript can change HTML attributes.**

### **Example:**

1. <!DOCTYPE html>
2. <html>
3. <body>
4. <script>
5. function light(sw) {
6. var pic;
7. **if** (sw == 0) {
8. pic = "pic\_lightoff.png"
9. } **else** {
10. pic = "pic\_lighton.png"
11. }
12. document.getElementById('myImage').src = pic;
13. }
14. </script>
15. <img id="myImage" src="pic\_lightoff.png" width="100" height="180">
16. <p>
17. <button type="button" onclick="light(1)">Light On</button>
18. <button type="button" onclick="light(0)">Light Off</button>
19. </p>
20. </body>
21. </html>

## Use External Script

Suppose, you have various HTML files which should have same script, then we can put our JavaScript code in separate file and can call in HTML file. Save JavaScript external files using .js extension.

#### **Note**: Do not add <script> tag in the external file, and provide the complete path where you have put the JS file**.**

### **Syntax:**

1. <script type="text/javascript" src="URL "></script>

### **Example:**

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <script type="text/javascript" src="external.js"></script>
5. </head>
6. <body>
7. <h2>External JavaScript Example</h2>
8. <form onsubmit="fun()">
9. <label>Enter your name:</label><br>
10. <input type="text" name="uname" id="frm1"><br>
11. <label>Enter your Email-address:</label><br>
12. <input type="email" name="email"><br>
13. <input type="submit">
14. </form>
15. </body>
16. </html>

**JavaScript code:**

1. function fun() {
2. var x = document.getElementById("frm1").value;
3. alert("Hi"+" "+x+ "you have successfully submitted the details");
4. }

## HTML <noscript> Tag

HTML <noscript> tag is used to write disabled script in the browser. The text written within <noscript></noscript> tag is not displayed on the browser.

### **Example:**

1. <!DOCTYPE html>
2. <html>
3. <body>
4. <p id="demo"></p>
5. <script>
6. document.getElementById("demo").innerHTML = "Hello JavaScript!";
7. </script>
8. <noscript>This text is not visible in the browser.</noscript>
9. </body>
10. </html>

# **HTML Comments**

Comments are some text or code written in your code to give an explanation about the code, and not visible to the user. Comments which are used for HTML file are known as HTML comments. Anything written between these tags will be ignored by the browser, so comments will not be visible on the webpage.

Comments of any code make code easy to understand and increase readability of code.

Comments are also part of the code, which gives an explanation of the code.

## How to add comment In HTML

You can add comments in your HTML file using <! -- ... --> tag. So if you will write anything between theses comment tag that will be treated as comment and browser will not read it.

### **Syntax**

1. <! -- Write commented text here -->

#### **Note**: The commented code will not be visible to a webpage, and hence you can use comment tag for documentation purpose, and debugging purpose:

Such as:

1. <!--   <p>There is some text</p>
2. <p>There is second text</p> -->

### **Example:**

1. <!DOCTYPE html>
2. <html>
3. <!-- This is Header section -->
4. <head>
5. <!-- Internal CSS -->
6. <style>
7. body{
8. text-align: center;
9. background-color: #f0f8ff;
10. font-size: 30px;
11. color: red;
12. }
13. </style>
14. </head>
16. <!-- This is body section, write code here which you want to display on web-page -->
17. <body>
18. <!-- heading tag -->
19. <h2>First WebPage</h2>
21. <!-- Paragraph tag -->
22. <p>Write your Content here!!!</p>
23. </body>
24. </html>

## Multiline Comment

In HTML code, we can also comments multiple lines at a time. In multiline comment we can use any description about code or multiple line code to debug, etc.

### **Syntax**

1. <!---
2. Your code is commented.
3. Write description of code.
4. It will not display at webpage.
5. -->

### **Example**

1. <h2>Cake Gallery</h2>
2. <!-- This is image **for** a yummy cake
3. you can see it on your web-page
4. of your favorite browser -->
5. <img src=" " alt="cake image" height="300px"
6. width="300px">

# **HTML File Paths**

An HTML file path is used to describe the location of a file in a website folder. File paths are like an address of file for a web browser. We can link any external resource to add in our HTML file with the help of file paths such as images, file, CSS file, JS file, video, etc.

The src or href attribute requires an attribute to link any external source to HTML file.

Following are the different types to specify file paths:

1. **<img src="picture.jpg">**It specifies that picture.jpg is located in the same folder as the current page.
2. **<img src="images/picture.jpg">**It specifies that picture.jpg is located in the images folder in the current folder.
3. **<img src="/images/picture.jpg">**It specifies that picture.jpg is located in the images folder at the root of the current web.
4. **<img src="../picture.jpg">** It specifies that picture.jpg is located in the folder one level up from the current folder.

File paths are used on webpages to link external files like:

1. Web pages
2. Images
3. Style sheets
4. JavaScript

There are two types of File Paths:

1. Absolute File Paths
2. Relative File Paths

## Absolute File Paths

Absolute file path specifies full URL address.

### **Example:**

1. <!DOCTYPE html>
2. <html>
3. <body>
4. <h2>Using a Full URL File Path</h2>
5. <img src="https://www.flipkart.com" alt="image" style="width:300px">
6. </body>
7. </html>

## Relative File Paths

The relative file path specifies to a file which is related to the location of current page.

### **Example:**

Let's take an example to see how the file path points to a file in the images folder located at the root of the current web.

1. <!DOCTYPE html>
2. <html>
3. <body>
4. <h2>Using a Relative File Path</h2>
5. <img src="/images/nature-2.jpg" alt="Mountain" style="width:300px">
6. </body>
7. </html>

## Important Points for File path:

* Always remember to use proper URL, file name, image name, else it will not display on the webpage.
* Try to use relative file paths, so that your code will be independent of URL.

# **HTML Head**

The HTML <head> element is used as a container for metadata (data about data). It is used between <html> tag and <body> tag.

The head of an HTML document is a part whose content is not displayed in the browser on page loading. It just contains metadata about the HTML document which specifies data about the HTML document.

An HTML head can contain lots of metadata information or can have very less or no information, it depends on our requirement. But head part has a crucial role an HTML document while creating a website.

Metadata defines the document title, character set, styles, links, scripts, and other meta information.

Following is a list of tags used in metadata:

* <title>
* <style>
* <meta>
* <link>
* <script>
* <base>

## HTML <title> Element

The HTML <title> element is used to define the title of the document. It is used in all HTML/XHTML documents. The <title> element must be placed between <head> element, and one document can only have one title element.

**What does <title> element do?**

1. It defines a title in the browser tab.
2. It provides a title for the page when it is added to favorites.
3. It displays a title for the page in search engine results.

#### **Note**: The title element must be specific about the document and its recommended length is 65 to 70 characters including spaces.

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>This Page Title</title>
5. </head>
6. <body>
7. <p>The body's content is displayed in the browser window.</p>
8. <p>The content of the title element is displayed in the browser tab, in favorites and in search engine results.</p>
9. </body>
10. </html>

## HTML <style> Element

The HTML <style> element is used to style the HTML page. The <style> element can have CSS properties for that HTML page only. If we want to apply CSS for multiple pages then we should use separate CSS file.

**Example:**

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>This is Page Title</title>
5. <style>
6. body {background-color: pink;}
7. h1 {color: red;}
8. p {color: blue;}
9. </style>
10. </head>
11. <body>
12. <h1>This is a Heading</h1>
13. <p>This is a paragraph.</p>
14. </body>
15. </html>

## HTML <link> Element

The HTML <link> element is used to link an external style sheet to your webpage. The <link> element contains main two attributes which are "rel" and "href". The rel attribute indicates that it is a stylesheet, and href gives the path to that external file.

**Example:**

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>This is title</title>
5. <link rel="stylesheet" href="style.css">
6. </head>
7. <body>
8. <h2>Web-page with external CSS</h2>
9. <p>This is looking a cool page</p>
10. </body>
11. </html>

## HTML <meta> Element

The HTML <meta> element is used to specify the character set, page description, keywords, authors and other metadata on the webpage.

Metadata is mainly used by browsers, search engines, and other web services to rank your webpage better.

**Let's see how to use metadata:**

**To define a character set:**

1. <meta charset="UTF-8">

The charset attribute specifies the character encoding. In this example we have set it to "UTF-8" which means it can handle to display any language.

### **Example:**

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="UTF-8">
5. </head>
6. <body>
7. <p>This is written in English language<span style="color: blue"> My friend's name is.......</span></p>
8. <p>This is Chinese language <span style="color: red">Wǒ de péngyǒu jiào</span></p>
9. </body>
10. </html>

**To define a description of your webpage:**

1. <meta name="description" content="Free Web tutorials">

If you give a meta description then it will be useful for the relevant search to perform by search engines.

**To define keywords for search engines:**

1. <meta name="keywords" content="HTML, CSS, XML, JavaScript">

The keyword value is also used to provide keywords for a search engine, but it may ignore by browser due to spammers.

**To define author of the webpage:**

1. <meta name="author" content="Akon">

The author value specifies the name of the person who wrote the page content, and it is useful to automatically extract author information by some content management systems.

**To refresh document every 30 seconds:**

1. <meta http-equiv="refresh" content="30">

Meta refresh is used to provide instructions to the browser to automatically refresh the page after the given time interval. As in above example it will automatically refresh after 30 sec

1. <meta http-equiv="refresh" content="10; url=https://www.flipkart.com/html-head>

If you add an URL with content value, then it will redirect to that page after the time limit will over.

**Example:**

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta http-equiv="refresh" content="5; url=https://www.flipkart.com">
5. </head>
6. <body>
7. <h2>Meta element Example</h2>
8. <p style="color: green;">Kindly wait **for** 5 seconds and after 5 seconds it will automatically redirect to URL specified in meta tag</p>
9. </body>
10. </html>

**Following is an example to show how to use all Meta elements within HTML head**

**Example:**

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="UTF-8">
5. <meta name="description" content="Free Web tutorials">
6. <meta name="keywords" content="HTML,CSS,XML,JavaScript">
7. <meta name="author" content="Akon">
8. </head>
9. <body>
10. <p>All the meta information are set.</p>
11. </body>
12. </html>

## Use <meta> tag to set the Viewport

This method is introduced in HTML5 to take control over the viewport by using <meta> tag.

Viewport is the user's visible area of a webpage. It changes from device to device and appears smaller on mobile phones than computer screens.

**Syntax for <meta> viewport element:**

1. <meta name="viewport" content="width=device-width, initial-scale=1.0">
2. Here, the **<meta> viewport** element specifies how to control the page's dimensions and scaling.
3. The **width=device-width** is used to set the width of the page to follow the screen-width of the device (which will vary depending on the device).
4. The **initial-scale=1.0** is used to set the initial zoom level when the page is first loaded by the browser.
5. **Example of a web page without the viewport <meta> tag:**
6. <!DOCTYPE html>
7. <html>
8. <body>
10. <p><b>To understand **this** example, you should open **this** page on a phone or a tablet.</b></p>
12. <img src="image.jpg" alt="image" width="460" height="345">
14. <p>
15. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut
16. laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation
17. ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel
18. eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu
19. feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum
20. zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis
21. eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum.
22. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat
23. facer possim assum.
24. </p>
26. </body>
27. </html>

**Example of a web page with the viewport <meta> tag:**

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
5. <style>
6. img {
7. max-width: 100%;
8. height: auto;
9. }
10. </style>
11. </head>
12. <body>
13. <p><b>To understand **this** example, you should open **this** page on a phone or a tablet.</b></p>
15. <img src="image.jpg" alt="image" width="460" height="345">
17. <p>
18. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut
19. laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation
20. ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel
21. eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu
22. feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum
23. zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis
24. eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum.
25. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat
26. facer possim assum.
27. </p>
29. </body>
30. </html>

#### **Note**: To see the difference clearly, open this page on smartphone or tablet.

## HTML <base> Element

The HTML <base> element is used to specify the base URL and base target for all relative URLs in a page.

**Example:**

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>Page Title</title>
5. <base href="https://static.flipkart.com/htmlpages/images/" target="\_blank">
6. </head>
7. <body>
8. <img src="html5.png">
9. <p>We have specified a base URL, the browser will look **for** the image "html5.png"
10. at "https://static.flipkart.com/htmlpages/images/html5.png"</p>
11. <p><a href=" https://www.flipkart.com">flipkart</a></p>
12. <p>The link above will open in a **new** window because base target is set to "\_blank".</p>
13. </body>
14. </html>

## HTML <script> element

HTML <script> element is used to apply client side JavaScript for the same page or to add an external JavaScript file to current page.

**Example:**

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <script>
5. function fun() {
6. document.getElementById("p").style.color="green";
7. }
8. </script>
9. </head>
10. <body>
11. <h2>Script within Head element</h2>
12. <p id="p">This will change the color</p>
13. <button type="button" onclick="fun()">Click me</button>
14. </body>
15. </html>

If we want to use some external JavaScript file then it can be applied by:

1. <script src=".js file\_path">

## Excluding <html>, <head> and <body> elements

HTML 5 facilitates us to omit the <html>, the <body>, and the <head> tag.

**Example:**

1. <!DOCTYPE html>
2. <title>Page Title</title>
3. <h1>This is a heading</h1>
4. <p>This is a paragraph.</p>

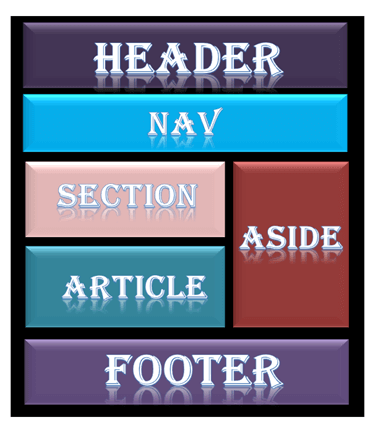
#### **Note**: It is not recommended to omit the <html> and <body> tags. Omitting these tags can crash DOM or XML software and produce errors in older browsers (IE9).

However, you can omit the <head> tag.

# **HTML Layouts**

HTML layouts provide a way to arrange web pages in well-mannered, well-structured, and in responsive form or we can say that HTML layout specifies a way in which the web pages can be arranged. Web-page layout works with arrangement of visual elements of an HTML document.

**Web page** layout is the most important part to keep in mind while creating a website so that our website can appear professional with the great look. You can also use CSS and JAVASCRIPT based frameworks for creating layouts for responsive and dynamic website designing



Every website has a specific layout to display content in a specific manner.

Following are different HTML5 elements which are used to define the different parts of a webpage.

* <header>: It is used to define a header for a document or a section.
* <nav>: It is used to define a container for navigation links
* <section>: It is used to define a section in a document
* <article>: It is used to define an independent self-contained article
* <aside>: It is used to define content aside from the content (like a sidebar)
* <footer>: It is used to define a footer for a document or a section
* <details>: It is used to define additional details
* <summary>: It is used to define a heading for the <details> element

#### **NOTE**: HTML layouts create an individual space for every part of the web page. So that every element can arrange in a significant order.

## Description of various Layout elements

### **HTML <header>**

The <header> element is used to create header section of web pages. The header contains the introductory content, heading element, logo or icon for the webpage, and authorship information.

### **Example:**

<!DOCTYPE html>

<html>

<head>

<title>First Webpage</title>

</head>

<body>

<header style="background-color: #303030; height: 80px;width: 100%">

<h1 style="font-size: 30px; color: white;text-align: center; padding-top: 15px;">Welcome to MyFirstWebpage</h1>

</header>

</body>

</html>

### **HTML <nav>**

The <nav> elements is a container for the main block of navigation links. It can contain links for the same page or for other pages.

### **Example:**

<!DOCTYPE html>

<html>

<head>

<style>

li{ display: inline-block;

padding: 10px}

</style>

</head>

<body>

<nav style="background-color:#bcdeef;">

<h1 style="text-align: center;">Navgation Links</h1>

<ul>

<li><a href="#">link1</a></li>

<li><a href="#">link2</a></li>

<li><a href="#">link3</a></li>

<li><a href="#">link4</a></li>

</ul>

</nav>

</body>

</html>

### **HTML <section>**

HTML <section> elements represent a separate section of a web page which contains related element grouped together. It can contain: text, images, tables, videos, etc.

### **Example:**

<!DOCTYPE html>

<html>

<head>

<title>Page Section</title>

</head>

<body>

<section style="background-color:#ff7f50; width: 100%; border: 1px solid black;">

<h2>Introduction to HTML</h2>

<p>HTML is a markup language which is used for creating attractive web pages with the help of styling, and which looks in a nice format on a web browser.</p>

</section>

</body>

</html>

### **HTML <article>**

The HTML

tag is used to contain a self-contained article such as big story, huge article, etc.

### **Example:**

<!DOCTYPE html>

<html>

<head>

<title>Article Example</title>

</head>

<body>

<article style="width: 100%; border:2px solid black; background-color: #fff0f5;">

<h2>History of Computer</h2>

<p>Write your content here for the history of computer</p>

</article>

</body>

</html>

### **HTML <aside>**

HTML <aside> define aside content related to primary content. The <aside> content must be related to the primary content. It can function as side bar for the main content of web page.

### **Example:**

<!DOCTYPE html>

<html>

<head>

<title>Aside Example</title>

</head>

<body>

<aside style="background-color:#e6e6fa">

<h2>Sidebar information</h2>

<p>This conatins information which will represent like a side bar for a webpage</p>

</aside>

</body>

</html>

### **HTML <footer>**

HTML <footer> element defines the footer for that document or web page. It mostly contains information about author, copyright, other links, etc.

### **Example:**

<!DOCTYPE html>

<html>

<head>

<title>Footer Section</title>

</head>

<body>

<footer style="background-color:#f0f8ff; width: 100%; text-align: center;">

<h3>Footer Example</h3>

<p>© Copyright 2018-2020. </p>

</footer>

</body>

</html>

### **HTML <details>**

HTML <details> element is used to add extra details about the web page and use can hide or show the details as per requirement.

### **Example:**

<!DOCTYPE html>

<html>

<head>

<title>Deatils element</title>

</head>

<body>

<details style="background-color: #f5deb3">

<summary>This is visible section: click to show other details</summary>

<p>This section only shows if user want to see it. </p>

</details>

</body>

</html>

### **HTML <summary>**

HTML <summary> element is used with the <details> element in a web page. It is used as summary, captions about the content of <details> element.

### **Example:**

<!DOCTYPE html>

<html>

<head>

<title>Summary Example</title>

</head>

<body>

<details>

<summary>HTML is acronym for?</summary>

<p style="color: blue; font-size: 20px;">Hypertext Markup Language</p>

</details>

</body>

</html>

# **HTML Layout Techniques**

Creating layouts are the most important things while designing a website, as it will ensure that your website looks in a well-arranged way and the content appears easy to understand. There are various techniques, and frameworks available for creating layouts, but here we will learn about simple techniques. You can use the following methods to create multicolumn layouts:

* HTML tables (Try not to use)
* CSS float property
* CSS framework
* CSS flexbox
* Layout using div

## HTML Tables (Not Recommended)

HTML table-based layout is one of the easiest ways for creating a layout, as table use only rows and column-based format, but HTML tables are not recommended for your page layout. The

element is designed to display tabular data. It is not good for a layout. Although first creating a layout is easy, but if you want to change or redesign your website, then it will be a complicated task.

Following is an example for the creation of a simple web page layout using HTML table.

### **Example:**

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. li{
6. display: inline-block;
7. padding: 10px;}
8. a{
9. color:#20b2aa;
10. }
11. </style>
12. </head>
13. <body>
14. <!-- Header Section -->
15. <table width="100%" style="border-collapse:collapse;">
16. <tr>
17. <td colspan="2" style="background-color:#1a1a1a; text-align: center;">
18. <h3 style="font-size: 30px; color: #ff6a6a;">iHub Table-layout</h3>
19. </td>
20. </tr>
21. <!-- Nav Section -->
22. <tr>
23. <td colspan="2" style="background-color:#666666;">
24. <ul>
25. <li><a href="#">Home</a></li>
26. <li><a href="#">Menu</a></li>
27. <li><a href="#">About-us</a></li>
28. <li><a href="#">Contact us</a></li>
29. </ul>
30. </td>
31. </tr>
32. <!-- Main Section -->
33. <tr>
34. <td style="background-color:#e6e6fa; width:80%; height: 400px; text-align: center;">
35. <p>Write your content Here</p>
36. </td>
37. <td style="background-color:#a7e6fb; height: 400px;">
38. <p>This is your side bar</p>
39. </td>
40. </tr>
41. <!-- Footer Section -->
42. <tr>
43. <td colspan="2" style="background-color:#2e2e2e; text-align: center;">
44. <p style="color:#f08080">©<strong>Copyright iHub.com</strong></p>
45. </td>
46. </tr>
47. </table>
48. </body>
49. </html>

## CSS Frameworks

CSS provides many frameworks like W3.CSS, Bootstrap, and many more, to create your layout fast. Using CSS frameworks you can easily create a responsive and attractive web layout. You just need to add a link for these frameworks, and you can use all properties available in the framework.

## CSS Float

You can create an entire web layout using CSS float property.

**Advantage:** It is very easy to learn and use. You just learn how the float and clear properties work.

**Disadvantage:** Floating elements are tied to the document flow, which may harm the flexibility.

### **Example:**

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. div.container {
6. width: 100%;
7. border: 1px solid gray;
8. }
10. header, footer {
11. padding: 1em;
12. color: white;
13. background-color: #000080;
14. clear: left;
15. text-align: center;
16. }
18. nav {
19. **float**: left;
20. max-width: 160px;
21. margin: 0;
22. padding: 1em;
23. }
25. nav ul {
26. list-style-type: none;
27. padding: 0;
28. }
30. nav ul a {
31. text-decoration: none;
32. }
34. article {
35. margin-left: 170px;
36. border-left: 1px solid gray;
37. padding: 1em;
38. overflow: hidden;
39. }
40. </style>
41. </head>
42. <body>
44. <div **class**="container">
46. <header>
47. <h1>Tutorials Gallery</h1>
48. </header>
50. <nav>
51. <ul>
52. <li><a href="#">HTML</a></li>
53. <li><a href="#">CSS</a></li>
54. <li><a href="#">JavaScript</a></li>
55. </ul>
56. </nav>
57. <article>
58. <h1>HTML</h1>
59. <p>HTML tutorial or HTML 5 tutorial provides basic and advanced concepts of html. Our HTML tutorial is
60. developed **for** beginners and professionals.</p>
61. <p>TML is an acronym which stands **for** Hyper Text Markup Language. Let's see what is Hyper Text and what is Markup Language?</p>
62. </article>
63. <footer>Copyright © iHub.com</footer>
64. </div>
65. </body>
66. </html>

## CSS Flexbox

Flexbox is a new layout mode in CSS3.

**Advantage:** It ensures that the page layout must accommodate different screen sizes and different display devices.

**Disadvantages:** It does not work in IE10 and earlier.

### **Example:**

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. .flex-container {
6. display: -webkit-flex;
7. display: flex;
8. -webkit-flex-flow: row wrap;
9. flex-flow: row wrap;
10. text-align: center;
11. }
13. .flex-container > \* {
14. padding: 15px;
15. -webkit-flex: 1 100%;
16. flex: 1 100%;
17. }
19. .article {
20. text-align: left;
21. }
23. header {background: #000080;color:white;}
24. footer {background: #000080;color:white;}
25. .nav {background:#eee;}
27. .nav ul {
28. list-style-type: none;
29. padding: 0;
30. }
31. .nav ul a {
32. text-decoration: none;
33. }
35. @media all and (min-width: 768px) {
36. .nav {text-align:left;-webkit-flex: 1 auto;flex:1 auto;-webkit-order:1;order:1;}
37. .article {-webkit-flex:5 0px;flex:5 0px;-webkit-order:2;order:2;}
38. footer {-webkit-order:3;order:3;}
39. }
40. </style>
41. </head>
42. <body>
44. <div **class**="flex-container">
45. <header>
46. <h1>City Gallery</h1>
47. </header>
49. <nav **class**="nav">
50. <ul>
51. <li><a href="#">HTML</a></li>
52. <li><a href="#">CSS</a></li>
53. <li><a href="#">JavaScript</a></li>
54. </ul>
55. </nav>
57. <article **class**="article">
58. <h1>HTML</h1>
59. <p>HTML tutorial or HTML 5 tutorial provides basic and advanced concepts of html. Our HTML tutorial is
60. developed **for** beginners and professionals.</p>
61. <p>TML is an acronym which stands **for** Hyper Text Markup Language. Let's see what is Hyper Text and what is Markup Language?</p>
62. <p><strong>Resize **this** page to see what happens!</strong></p>
63. </article>
65. <footer>Copyright © iHub.com</footer>
66. </div>
68. </body>
69. </html>

### **Layout using div**

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>Webpage using div</title>
5. <style>
6. body{
7. margin:0px;
8. }
9. .header{
10. padding: 10px;
11. background-color:#455e64;
12. text-align: center;
13. }
14. .header h2{
15. color: black; }
16. /\*===============[Nav CSS]==========\*/
17. .nav{
18. background-color:#243238;
19. padding: 5px;
20. }
22. .nav li{
23. list-style: none;
24. display: inline-block;
25. padding: 8px;
26. }
27. .nav a{
28. color: #fff;
29. }
31. .nav ul li a:hover{
32. text-decoration: none;
33. color: #7fffd4;
34. }
35. .lside{
36. **float**: left;
37. width: 80%;
38. min-height: 440px;
39. background-color: #f0f8ff;
40. text-align: center;
41. }
42. .rside
43. {
44. text-align: center;
45. **float**: right;
46. width: 20%;
47. min-height: 440px;
48. background-color:  #c1cdcd;
49. }
50. .footer{
51. height: 44px;
52. background-color:#455e64;
53. text-align: center;
54. padding-top: 10px;}
56. .footer p{
57. color:  #8fbc8f;
58. }
60. </style>
61. </head>
62. <body>
63. <div>
64. <div **class**="header">
65. <h2>iHub Div Layout</h2>
66. </div>
67. <!-- Nav -->
68. <div **class**="nav">
69. <ul>
70. <li><a href="#">HOME</a></li>
71. <li><a href="#">MENU</a></li>
72. <li><a href="#">ABOUT</a></li>
73. <li><a href="#">CONTACT</a></li>
74. <li style="float: right;"><a href="#">LOGIN</a></li>
75. <li style="float: right;"><a href="#">SIGN-UP</a></li>
76. </ul>
77. </div>
79. <!-- main -->
80. <div style="height:440px">
81. <div **class**="lside">
82. <p>Write your content here</p>
83. </div>
84. <!-- side -->
85. <div **class**="rside">
86. <p>This is side</p>
87. </div>
88. </div>
89. <!-- footer -->
90. <div **class**="footer">
91. <p>©<strong>Copyright iHub.com</strong></p>
92. </div>
93. </div>
94. </body>
95. </html>

# **HTML Responsive**

**Responsive Web design**

Responsive web design is used to make your web page look appropriate, good, and well placedon all devices (desktop, tablet, smartphone etc.)

Responsive web design uses HTML and CSS to resize, hide, shrink, enlarge, or move the content. It makes the content look good on any screen.

**Set the viewport**

Let's see how to set the viewport.

[**how to set the viewport**](https://www.javatpoint.com/html-viewport)

## Responsive Images

Images which can be scaled nicely to fit any browser size are known as responsive images.

## How to make Image Responsive?

**By using the width property**

Set the CSS width property to 100% to make the image responsive and scale up and down.

**Example**

1. <!DOCTYPE html>
2. <html>
3. <meta name="viewport" content="width=device-width, initial-scale=1.0">
4. <body>
5. <h2>Responsive Image</h2>
6. <p>When we set the CSS width property to 100%, it makes the image responsive.
7. Resize the browser window to see the effect.</p>
8. <img src="img\_girl.jpg" style="width:100%;">( change image)
9. </body>
10. </html>

#### **Note**: A problem with the above method (width: 100%) is that the image can be scaled up to be larger than its original size. So, it is better to use the max-width property instead.

## By using the max-width Property

This method is best and most used because it facilitates that the image will scale down if it has to, but never scale up to be larger than its original size.

**Example**

1. <!DOCTYPE html>
2. <html>
3. <meta name="viewport" content="width=device-width, initial-scale=1.0">
4. <body>
5. <h2>Responsive Image</h2>
6. <p>"max-width:100%" makes the image responsive and also ensures that the image
7. doesn't get bigger than its original size.</p>
8. <p>Resize the browser window to see the effect.</p>
9. <img src="img\_girl.jpg" style="max-width:100%;height:auto;"> (Change the image)
10. </body>
11. </html>

## Change images according to the browser width

By using the HTML <picture> element, you can set two or more imagesaccording to the browser width. It will change the picture when you change the browser-size. i.e. desktop and phone.

**Example**

1. <!DOCTYPE html>
2. <html>
3. <meta name="viewport" content="width=device-width, initial-scale=1.0">
4. <body>
5. <h2>Change Images Depending on Browser Width</h2>
6. <p>Resize the browser width and the image will change at 600px and 1500px.</p>
7. <picture>
8. <source srcset="img\_smallflower.jpg" media="(max-width: 600px)">(Change image)
9. <source srcset="img\_flowers.jpg" media="(max-width: 1500px)">(Change image)
10. <source srcset="flowers.jpg">
11. <img src="img\_flowers.jpg" alt="Flowers" style="width:auto;">
12. </picture>
13. </body>
14. </html>

## Responsive Text-size

We can make the text size responsive by using the "uv" unit. It means viewport-width. By using this, we can make the text size to follow the browserwindow screen.

**Example**

1. <!DOCTYPE html>
2. <html>
3. <meta name="viewport" content="width=device-width, initial-scale=1.0">
4. <body>
5. <h1 style="font-size:10vw;">Here size is 10vw.</h1>
6. <p style="font-size:6vw;">Here size is 6vw.</p>
7. <p style="font-size:4vw;">Here size is 4vw.</p>
8. <p>Resize the browser window to see how the text size changes.</p>
9. </body>
10. </html>

#### **Note**: viewport specifies the browser window size. 1vw = 1% of viewport width. Means, if the viewport is 100cm wide, 1vw is 1.0cm.

# **HTML Computer code**

When we are programming, sometimes it is mandatory to show the Output result, error message, or coding part to user on a webpage. Hence to solve this issue HTML uses different tags for the user inputs, codes, programs, etc. With the help of these tags, you will be able to write codes to display on your webpage.

Following is a list of some tags which are used in HTML for this task.

* <code>
* <kbd>
* <samp>
* <var>
* <pre>

## HTML <code> element

It is used to represent some programming code on your website. The content written between tag will be displayed in default monospace font.

**Example:**

1. <!DOCTYPE html>
2. <html>
3. <body>
4. <h2>Computer Code</h2>
5. <p>This is a programming code:</p>
6. <code>
7. x = 5;<br>
8. y = 6;<br>
9. z = x + y;
10. </code>
11. </body>
12. </html>

## HTML <kbd> Element

It is used to represent user input, keyboard input, voice command etc. Text written within <kbd>.....</kbd> tags is typically displayed in the browser's default monospace font.

**Example:**

1. <!DOCTYPE html>
2. <html>
3. <body>
4. <h2>The kbd Element</h2>
5. <kbd>This is how  content written within kbd element looks like.</kbd></p>
6. </body>
7. </html>

## HTML <samp> Element

The HTML <samp> element is used to represent a program's output. Text written within samp element is typically displayed in the browser's default monospace font.

**Example:**

1. <!DOCTYPE html>
2. <html>
3. <body>
4. <h2>The samp Element</h2>
5. <samp>This is how the content within samp element looks like. </samp>
6. </body>
7. </html>

## HTML <var> element

The HTML <var> element is used to define a variable. The variable could be a variable in a mathematical expression or a variable in programming context.

**Example:**

1. <!DOCTYPE html>
2. <html>
3. <body>
4. <h2>The var Element</h2>
5. <p>This is a famous formula: <var>E</var> = <var>mc</var><sup>2</sup>.</p>
6. </body>
7. </html>

## HTML <pre> element

The <pre> element defines preformatted text, which displays the content within it in a fixed-width font. It keeps the content into its original format and ignores all formatting.

**Example:**

1. <!DOCTYPE html>
2. <html>
3. <body>
4. <h3>Example of pre tag</h3>
5. <pre>
6. This is content written
7. within pre tag, and  pre tag will ignore all
8. spaces,     **break** lines, and will display the content
9. as in original format.
10. </pre>
11. </body>
12. </html>

# **HTML Entities**

HTML character entities are used as a replacement of reserved characters in HTML. You can also replace characters that are not present on your keyboard by entities.

These characters are replaced because some characters are reserved in HTML. HTML entities provide a wide range of characters which can allow you to add icons, geometric shapes, mathematical operators, etc.

**For example:** if you use less than (<) or greater than (>) symbols in your text, the browser can mix them with tags that's why character entities are used in HTML to display reserved characters.

## How to use an entity:

You can use an entity in your HTML document by name or by a numerical character reference. Each entity starts with symbol ampersand (&) and ends with a semicolon (;).

**Syntax:**

1. &entity\_name;
2. OR
3. &#entity\_number;

### **Most used HTML Character Entities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Result** | **Description** | **Entity Name** | **Entity Number** |
|  | non-breaking space | &nbsp; | 160 |
| < | less than | &lt; | 60 |
| > | greater than | &gt; | 62 |
| & | ampersand | &amp; | 38 |
| " | double quotation mark | &quot; | 34 |
| ' | single quotation mark (apostrophe) | &apos; | 39 |
| ¢ | cent | &cent; | 162 |
| £ | pound | &pound; | 163 |
| ¥ | yen | &yen; | 165 |
| € | Euro | &euro; | 8364 |
| © | copyright | &copy; | 169 |
| ® | registered trademark | &reg; | 174 |

#### **Note**: Entity names are case sensitive.

**Advantage of entity name:** An entity name is easy to remember.

**Disadvantage of entity name:** Browsers may not support all entity names, but the support for numbers is good.

**Example:**

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title></title>
5. </head>
6. <body>
7. <h3>HTML entity example</h3>
8. <p> "This is the content written within entity"</p>
9. <p> <p> Paragraph tag </p>
10. </body>
11. </html>

## Diacritical Marks in HTML

There are some special types of letters used in HTML whichhave some glyph added to the top or below the letters. These glyphs are called diacritical mark.

Some diacritical marks, like grave (   ̀) and acute (   ́) are called accents.Diacritical marks can be used both above and below a letter, inside a letter, and between two letters.

**Following is a list of some diacritical marks:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Mark** | **Character** | **Construct** | **Result** |
| ̀ | a | a&#768; | à |
| ́ | a | a&#769; | á |
| ̂ | a | a&#770; | â |
| ̃ | a | a&#771; | ã |
| ̀ | O | O&#768; | Ò |
| ́ | O | O&#769; | Ó |
| ̂ | O | O&#770; | Ô |
| ̃ | O | O&#771; | Õ |

# **HTML Symbols**

There are many mathematical, technical and currency symbols which are not present on a normal keyboard. We have to use HTML entity names to add such symbols to an HTML page.

If there no entity name exists, you can use an entity number, a decimal, or hexadecimal reference.

**Example:**

1. <!DOCTYPE html>
2. <html>
3. <body>
4. <h3>The Currency Symbols</h3>
5. <p>This is Indian Rupee symbol <b>₹<b></p>
6. <p>This is Euro symbol <b>€</b></p>
7. <p> This is Dollar symbol <b>#36;</b></p>
8. </body>
9. </html>
10. Mathematical Symbols Supported by HTML

|  |  |  |  |
| --- | --- | --- | --- |
| **Char** | **Number** | **Entity** | **Description** |
| ∀ | &#8704; | &forall; | FOR ALL |
| ∂ | &#8706; | &part; | PARTIAL DIFFERENTIAL |
| ∃ | &#8707; | &exist; | THERE EXISTS |
| ∅ | &#8709; | &empty; | EMPTY SETS |
| ∇ | &#8711; | &nabla; | NABLA |
| ∈ | &#8712; | &isin; | ELEMENT OF |
| ∉ | &#8713; | &notin; | NOT AN ELEMENT OF |
| ∋ | &#8715; | &ni; | CONTAINS AS MEMBER |
| ∏ | &#8719; | &prod; | N-ARY PRODUCT |
| ∑ | &#8721; | &sum; | N-ARY SUMMATION |

Greek Symbols Supported by HTML

|  |  |  |  |
| --- | --- | --- | --- |
| **Char** | **Number** | **Entity** | **Description** |
| Α | &#913; | &Alpha; | GREEK CAPITAL LETTER ALPHA |
| Β | &#914; | &Beta; | GREEK CAPITAL LETTER BETA |
| Γ | &#915; | &Gamma; | GREEK CAPITAL LETTER GAMMA |
| Δ | &#916; | &Delta; | GREEK CAPITAL LETTER DELTA |
| Ε | &#917; | &Epsilon; | GREEK CAPITAL LETTER EPSILON |
| Ζ | &#918; | &Zeta; | GREEK CAPITAL LETTER ZETA |

Some Important Symbols Supported by HTML

|  |  |  |  |
| --- | --- | --- | --- |
| **Char** | **Number** | **Entity** | **Description** |
| © | &#169; | &copy; | COPYRIGHT SIGN |
| ® | &#174; | &reg; | REGISTERED SIGN |
| € | &#8364; | &euro; | EURO SIGN |
| ™ | &#8482; | &trade; | TRADEMARK |
| ← | &#8592; | &larr; | LEFTWARDS ARROW |
| ↑ | &#8593; | &uarr; | UPWARDS ARROW |
| → | &#8594; | &rarr; | RIGHTWARDS ARROW |
| ↓ | &#8595; | &darr; | DOWNWARDS ARROW |
| ♠ | &#9824; | &spades; | BLACK SPADE SUIT |
| ♣ | &#9827; | &clubs; | BLACK CLUB SUIT |
| ♥ | &#9829; | &hearts; | BLACK HEART SUIT |
| ♦ | &#9830; | &diams; | BLACK DIAMOND SUIT |

# **HTML Charset**

HTML Charset is also called HTML Character Sets or HTML Encoding. It is used to display an HTML page properly and correctly because for displaying anything correctly, a web browser must know which character set (character encoding) to use.

## HTML Character Encoding

There are various types of Character Encoding which are given below:

## ASCII Character Set

ASCII stands for American Standard Code for Information Interchange. In HTML, the first ever character encoding standard is the ASCII standard. ASCII provides 128 different alphanumeric characters that could be used on the internet: numbers (0-9), English letters (A-Z), and some special characters like! $ + - ( ) @ <> .

The main problem with ASCII encoding was it contains a limited range of characters. It contains mainly 128 characters.

## ANSI Character Set

ANSI stands for **American National Standard Institute**. It is character set standard which is an extended version of standard ASCII character set. It supports 256 character set. ANSI also called as Windows-1252, and it was the default character set for Windows up to Windows 95.

## ISO-8859-1 Character Set

ISO-8859-1 was the default character encoding in HTML 2.0. It was also an extension of ASCII standard with International characters. It also used full bytes (8-bits) to show characters.

## UTF-8 Character Set

UTF-8 is a variable width character encoding which covers almost all of the characters and symbols in the world. ANSI (Windows-1252) was the original Windows character set, which supported 256 different character codes.

ISO-8859-1 was the default character set for HTML 4. This character set also supported 256 different character codes.

Why UTF 8 is also supported in HTML4?

Because ANSI and ISO-8859-1 were so limited, HTML 4 also supported UTF-8.The default character encoding for HTML5 is UTF-8.

**UTF-8 syntax for HTML4:**

1. <meta http-equiv="Content-Type" content="text/html;charset=ISO-8859-1">

**UTF-8 syntax for HTML5:**

1. <meta charset="UTF-8">

# **HTML URL Encode**

## What is URL?

URL stands for Uniform Resource Locator. It is actually a web address. A URL can contain words i.e. (iHub.com) or an Internet Protocol (IP) address i.e.195.201.68.81. But most of the user use URL in the form of words because it is easy to remember than numbers.

**Syntax of a URL:**

1. scheme://prefix.domain:port/path/filename

Here,

* **scheme** is used to define the type of Internet service (most common is http or https).
* **prefix** is used to define a domain prefix (default for http is www).
* **domain** is used to define the Internet domain name (like iHub.com).
* **port** is used to define the port number at the host (default for http is 80).
* **path** is used to define a path at the server (If omitted: the root directory of the site).
* **filename** is used to define the name of a document or resource.

**Following is a list of some common types of schemes used in URL:**

* http(HyperText Transfer Protocol):Common web pages. Not encrypted.
* https (Secure HyperText Transfer Protocol):Secure web pages. Encrypted.
* ftp(File Transfer Protocol): Downloading or uploading files.
* file: A file on your computer.

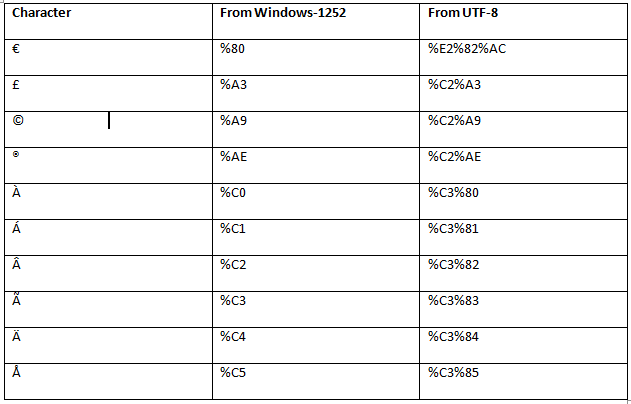
## URL Encoding

URL encoding is used to convert non-ASCII characters into a format that can be used over the Internet because a URL is sent over the Internet by using the ASCII character-set only. If a URL contains characters outside the ASCII set, the URL has to be converted.

In URL encoding, the non-ASCII characters are replaced with a "%" followed by hexadecimal digits.

URLs cannot contain spaces. URL encoding normally replaces a space with a plus (+) sign, or %20.

**Following is a list of some character sets which are encoded by browser after submitting the text.**



# **HTML Attributes**

# HTML Global Attributes

HTML global attributes are those attributes which are common for all HTML elements. The global attributes are supported by both standard and non-standard element.

The global attributes can be used with all elements, although it may not have any effect on some elements.

Following is the complete list of global attributes with their description:

HereHTML Tags Listrepresent new in HTML5.

|  |  |  |
| --- | --- | --- |
| **Attributes** | **value** | **Description** |
| accesskey | character | It is used to generate keyboard shortcuts for the current element. |
| class | classname | It is used to provide the class name for the current element. It is mainly used with the stylesheet. |
| ContenteditableHTML Tags List | true false | It determines whether the content within an element is editable or not. |
| contextmenu | menu\_id | It defines the id for the <menu> element which is used as a context menu (a menu appear on right click) for an element. |
| data-\*HTML Tags List | somevalue | It is used to store element-specific private data which can be accessed by Javascript. |
| dir | rtl ltr auto | It specifies the direction of the content inside the current element. |
| draggableHTML Tags List | true false auto | It specifies whether the content within an element is movable or not using Drag and Drop API. |
| dropzoneHTML Tags List | copy move link | It specifies the action is taken on the dragged element when it is dropped, ¬¬ such as whether it is copied, moved or linked. |
| hiddenHTML Tags List |  | It is used to hide the element from view. |
| id | id | It specifies a unique id for the element. It can be used with CSS and JavaScript. |
| lang | language\_code | It specifies the primary language for the content of an element. |
| style | style | It is used to apply inline CSS to the current element. |
| spellcheckHTML Tags List | true false | It specifies whether the content should be checked for spelling errors or not. |
| tabindex | number | It determines the tabbing order of an element. |
| title | text | It is used to provide the title, name, or some extra information about the element. |
| translateHTML Tags List | yes no | It specifies whether the content of the element should be translated when the page is localized or not. |

# HTML Event Attributes

When a browser reacts on user action, then it is called as an event. For example, when you click on the submit button, then if the browser displays an information box.

In HTML5 there are lots of event attributes available which can be activated using a programming language such as JavaScript.

Following is a table of event attributes, using these attributes you can perform several events.

## Windows Event Attributes

Windows events are related for the window object, and it can only be applied with <body> tag.

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| onafterprint | Executed the script after the document is printed. |
| onbeforeprint | Executed the script before the document is printed. |
| onbeforeunload | Executed the script before a document being unloaded. |
| onerror | Executed the script when an error occurs. |
| onhashchange | Executed the script when the anchor part in URL of the webpage is changed. |
| onload | Executed the script when the webpage is entirely loaded. |
| onmessage | Executed the script when a message event occurs. |
| onoffline | Executed the script when the network connection is disconnected, and browser started working offline. |
| ononline | Executed the script when the browser started working online |
| onpagehide | Executed the script when the current webpage is hidden such as if the user has moved away from the current webpage. |
| onpageshow | Executed the script when the current webpage is focused. |
| onpopstate | Executed the script when the window's active history is changed. |
| onresize | Executed the script when the window is resized. |
| onstorage | Executed the script when web storage is updated. |
| onunload | Executed the script when the current webpage is unloaded, or window is closed. |

## Form Event Attributes

Form event occurs when the user performs some action within the form such as submitting the form, selecting input field, etc.

The form events can be used with any element, but these are mainly used with HTML form elements.

Following is the list of all Form Event attributes:

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| onblur | Executed the script when form element loses the focus. |
| onchange | Executed the script when the value of the element is changed. |
| onfocus | Trigger an event when the element gets focused. |
| oninput | Executed the script when the user enters input to the element. |
| oninvalid | Executed the script when the element does not satisfy its predefined constraints. |
| onreset | Triggers the event when user reset the form element values. |
| onsearch | Triggers the event when a search field receives some input. |
| onselect | Triggers the event when the user has selected some text. |
| onsubmit | Triggers the event when a form is submitted. |

Keyboard Event Attributes

Keyboard event occurs when a user interacts with the keyboard. Following is a list of the Keyboard event.

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| onkeydown | Triggers the event when the user presses down a key on the keyboard. |
| onkeypress | Trigger the event when the user presses the key which displays some character. |
| onkeyup | Trigger the event when the user releases the currently pressed key. |

Mouse Event Attributes

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| onclick | Trigger the event when the mouse clicks on the element. |
| ondblclick | Trigger the event when mouse double-click occurs on the element. |
| onmousedown | Trigger the event when the mouse button is pressed on the element. |
| onmousemove | Trigger the event when the mouse pointer moves over the element. |
| onmouseout | Trigger the event when the mouse moves outside the element. |
| onmouseover | Trigger the event when the mouse moves onto the element. |
| onmouseup | Trigger the event when the mouse button is released. |
| onmousewheel | Deprecated. Use the onwheel attribute. |
| onwheel | Trigger the event when the mouse wheel rolls up or down on the element |

Clipboard Event Attributes

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| oncopy | Trigger the event when the user copies the content to the system clipboard. |
| oncut | Trigger the event when the content of an element is cut and copy to the clipboard. |
| onpaste | Trigger the event when the user pastes some content in an element. |

Media Event Attributes

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| onabort | Executed the script when media playback is aborted. |
| oncanplay | Executed the script when the media file is ready to play. |
| oncanplaythrough | Executed the script when the media file is ready to play without buffering or stopping. |
| oncuechange | Executed the script text cue of <track> element is changed. |
| ondurationchange | Executed the script when the media file duration is changed. |
| onemptied | Executed the script if media occurs some fatal error, and the file becomes unavailable. |
| onended | Executed the script when the media file occurs its end point. |
| onerror | Executed the script when some error occurred while fetching the media data. |
| onloadeddata | Executed the script when media data is loaded. |
| onloadedmetadata | Executed the script when metadata of media file is loaded. |
| onloadstart | Executed the script when loading of media file starts. |
| onpause | Executed the script when media playback is paused. |
| onplay | Executed the script when media file ready to play after being paused. |
| onplaying | Executed the script when media file is started playing. |
| onprogress | Executed the script when the browser is in the process of getting the media data. |
| onratechange | Executed the script when playback speed changed. |
| onseeked | Executed the script when seek operation is ended and seeking attribute is set to false. |
| onseeking | Executed the script when seek operation is active and seeking attribute is set to true. |
| onstalled | Executed the script when browser unexpectedly stopped fetching the data media. |
| onsuspend | Executed the script if fetching of media data is intentionally stopped. |
| ontimeupdate | Executed the script when playback position is changed, such as if a user fasts forward the track. |
| onvolumechange | Executed the script when media volume is changed (muted or unmuted). |
| onwaiting | Executed the script if playback pause to wait for loading more data. |

# **HTML 5 Tags**

There is a list of newly included tags in HTML 5. These HTML 5 tags (elements) provide a better document structure. This list shows all HTML 5 tags in alphabetical order with description.

List of HTML 5 Tags

|  |  |
| --- | --- |
| **Tag** | **Description** |
| <article> | This element is used to define an independent piece of content in a document, that may be a blog, a magazine or a newspaper article. |
| <aside> | It specifies that article is slightly related to the rest of the whole page. |
| <audio> | It is used to play audio file in HTML. |
| <bdi> | The bdi stands for bi-directional isolation. It isolates a part of text that is formatted in other direction from the outside text document. |
| <canvas> | It is used to draw canvas. |
| <data> | It provides machine readable version of its data. |
| <datalist> | It provides auto complete feature for textfield. |
| <details> | It specifies the additional information or controls required by user. |
| <dialog> | It defines a window or a dialog box. |
| <figcaption> | It is used to define a caption for a <figure> element. |
| <figure> | It defines a self-contained content like photos, diagrams etc. |
| <footer> | It defines a footer for a section. |
| <header> | It defines a header for a section. |
| <main> | It defines the main content of a document. |
| <mark> | It specifies the marked or highlighted content. |
| <menuitem> | It defines a command that the user can invoke from a popup menu. |
| <meter> | It is used to measure the scalar value within a given range. |
| <nav> | It is used to define the navigation link in the document. |
| <progress> | It specifies the progress of the task. |
| <rp> | It defines what to show in browser that don't support ruby annotation. |
| <rt> | It defines an explanation/pronunciation of characters. |
| <ruby> | It defines ruby annotation along with <rp> and <rt>. |
| <section> | It defines a section in the document. |
| <summary> | It specifies a visible heading for <detailed> element. |
| <svg> | It is used to display shapes. |
| <time> | It is used to define a date/time. |
| <video> | It is used to play video file in HTML. |
| <wbr> | It defines a possible line break. |

# **HTML <!DOCTYPE> tag**

On the HTML document you have often seen that there is a <!DOCTYPE html> declaration before the <html> tag. HTML <!DOCTYPE> tag is used to inform the browser about the version of HTML used in the document. It is called as the document type declaration (DTD).

Technically <!DOCTYPE > is not a tag/element, it just an instruction to the browser about the document type. It is a null element which does not contain the closing tag, and must not include any content within it.

Actually, there are many type of HTML e.g. HTML 4.01 Strict, HTML 4.01 Transitional, HTML 4.01 Frameset, XHTML 1.0 Strict, XHTML 1.0 Transitional, XHTML 1.0 Frameset, XHTML 1.1 etc.

The <!DOCTYPE> declaration refers Document Type Declaration (DTD) in HTML 4.01; because HTML 4.01 was based on SGML. But HTML 5 is not SGML based language.

#### DTD defines the rules for the markup languages so that the browsers recognize the content correctly.

The doctype declaration differs between HTML versions. The HTML 5 doctype declaration is given below.

### **Syntax**

<!DOCTYPE html>

**Following are some specifications about the HTML <!DOCTYPE>**

|  |  |
| --- | --- |
| **Display** | **None** |
| **Start tag/End tag** | Start tag only |
| **Usage** | Structural |

Let's see an example of HTML document with doctype declaration.

# **HTML Anchor**

The **HTML anchor tag** defines *a hyperlink that links one page to another page*. It can create hyperlink to other web page as well as files, location, or any URL. The "href" attribute is the most important attribute of the HTML a tag. and which links to destination page or URL.

## href attribute of HTML anchor tag

The href attribute is used to define the address of the file to be linked. In other words, it points out the destination page.

The syntax of HTML anchor tag is given below.

<a href = "..........."> Link Text </a>

Let's see an example of HTML anchor tag.

1. **<a** href="second.html"**>**Click for Second Page**</a>**

## Specify a location for Link using target attribute

If we want to open that link to another page then we can use target attribute of <a> tag. With the help of this link will be open in next page.

## Example:

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title></title>**
5. **</head>**
6. **<body>**
7. **<p>**Click on **<a** href="https://www.iHub.com/" target="\_blank"**>** this-link **</a>**to go on home page of iHub.**</p>**
8. **</body>**
9. **</html>**

**Note:**

* The **target** attribute can only use with href attribute in anchor tag.
* If we will not use target attribute then link will open in same page.

## Appearance of HTML anchor tag

An **unvisited link** is displayed underlined and blue.

A **visited link** displayed underlined and purple.

An **active link** is underlined and red.

# **HTML <abbr> tag**

HTML <abbr> tag is used to represent an acronym or abbreviation of a longer word or phrase, such as www, HTML, HTTP, etc. The content written between <abbr> tags renders with dotted underline in some browser.

This tag can be used with "title" attribute (optional), and the value of title attribute will be pop-up when the mouse hovers over the content written between <abbr> tag.

### **Syntax**

1. **<abbr** title="HyperText Markup language"**>**HTML**</abbr>**

**Following are some specifications about the HTML <abbr> tag**

|  |  |
| --- | --- |
| **Display** | **Inline** |
| **Start tag/End tag** | Both Start tag and End tag |
| **Usage** | Semantic/Textual |

### **Example**

**With title Attribute:**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Abbreviation Tag**</title>**
5. **</head>**
6. **<body>**
7. **<h2>**Abbreviation tag example**</h2>**
8. **<p><b>**Hover mouse over the content and see the abbreviation**</b></p>**
9. **<abbr** title="Hyper Text Transfer protocol"**>**HTTP**</abbr>**
10. **</body>**
11. **</html>**

### **Example**

**Without title Attribute:**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Abbreviation tag**</title>**
5. **</head>**
6. **<body>**
7. **<h2>**Abbreviation tag Example**</h2>**
8. **<p><abbr>**CPU**</abbr>** is brain of a Computer**</p>**
9. **</body>**
10. **</html>**

# **HTML <acronym> tag**

HTML <acronym> tag is used with title attribute to contain a full explanation of an acronym content. When you hover the mouse on content, then it will show the explanation of word.

#### **Note: The <acronym> tag has been depreciated in HTML5 and we can use <abbr> tag instead of <acronym>.**

### **Syntax**

1. **<acronym** title="Indian Space Research Organisation"**>**ISRO**</acronym>**

**Following are some specifications about the HTML <acronym> tag**

|  |  |
| --- | --- |
| **Display** | **Inline** |
| **Start tag/End tag** | Both Start tag and End tag |
| **Usage** | Semantic/Textual |

### **Example**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Acronym tag**</title>**
5. **</head>**
6. **<body>**
7. **<h2>**Acronym tag Example**</h2>**
8. **<acronym** title="Indian Space Research Organisation"**>**ISRO**</acronym>**
9. **</body>**
10. **</html>**

## Difference between <acronym> and <abbr> tag

Although <acronym> tag is not supported by HTML5 but instead of that we can use <abbr> tag and we will get the same result using both tag.

### **Example**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **</head>**
5. **<body>**
6. **<h2>**Acronym and Abbreviation Example**</h2>**
7. **<acronym** title="Indian Space Research Organisation"**>**ISRO**</acronym>**
8. **<abbr** title="National Aeronautics and Space Administration"**>**NASA**</abbr>**
9. **</body>**
10. **</html>**

# **HTML <address> tag**

HTML <address> tag is used to specify the authorship information of the article or webpage. It can contain any type of information which is needed such as, URL, physical address, phone number, email, other links, etc.

The <address> tag is useful for various contexts such as business contact information in the header of the page, or author related contact information, etc.

The contact information written between <address> tags mostly renders in the italic form on the browser.

#### **Note: To represent a random address use <p> tag instead of <address> tag, as it should contain the main contact information.**

### **Syntax**

1. <address>Contact Author at:<br>
2. <a href="mailto:example@gmail.com">Example@gmail.com</a></address>

If you want to specify the information of the author for an article, you must place the <address> tag inside the <article> element.

### **Example**

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>Address Tag</title>
5. </head>
6. <body>
7. <h2>Example of Address tag</h2>
8. <address>The article is written by:<b>Harshita</b> <br>Contact Author at:
9. <a href="mailto:example@gmail.com">Example@gmail.com</a><br>You can also visit      our blog:
10. <a href="http://iHub.blogspot.com/">iHub blog</a>
11. </address>
12. </body>
13. </html>

# **HTML <applet> tag (Not supported in HTML5)**

HTML <applet> tag was used to embed the Java applet in an HTML document. This element has been deprecated in HTML 4.0 and instead of it we can use <object> and newly added element <embed>.

The use of Java applet is also deprecated, and most browsers do not support the use of plugins.

#### **Note: The <applet> tag is deprecated in HTML4.0 and not supported in HTML5. So you can use <object> tag or <embed> tag instead of <applet>.**

### **Syntax**

1. <applet code="URL" height="200" width="100">.............</applet>

**Following are some specifications about <applet> tag**

|  |  |
| --- | --- |
| **Display** | **Block** |
| **Start tag/End tag** | Both Start tag and End tag |
| **Usage** | Embed Applets |

### **Example**

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>Applet Tag</title>
5. </head>
6. <body>
7. <p>Example of Applet Tag</p>
8. <applet code="Shapes.class" align="right" height="200" width="300">
9. <b>Sorry! you need Java to see **this**</b>
10. </applet>
11. </body>
12. </html>

## Attributes

**Specific Attributes**

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Value** | **Description** |
| code | URL | It specifies the URL of Java applet class file. |
| width | pixels | It specifies the display width of the applet panel. |
| height | pixels | It specifies the display height of applet panel |
| align | * left * right * top * middle * bottom | It specifies the position of applet application relative to surrounding content. |
| alt | text | It is used to display alternative text in case browser does not support Java. |
| archive | URL | This specifies the archived or compressed version of an applet application. |
| object | name | It specifies the URL or reference to a serialized representation of an applet. |
| codebase | URL | It specifies the exact or relative URL of applets .class file specified in the code attribute. |
| hspace | pixels | It specifies the horizontal space around the applet. |
| vspace | pixels | It specifies the vertical space around the applet. |
| name | name | It specifies the name for the applet |

# **HTML <area> tag**

## Description:

The <area> tag defines the clickable areas or active areas inside the image-map which are associated with the hyperlinks. If you click on those areas then it will perform some action such as open a new image, new URL, etc. This tag is always used with <map> element.

Inside an image map different areas can be hyperlinked to various locations using multiple <area> elements in a single <map> element.

The <area> element is defined with (required) attributes **shape** and **coords**. The shape attribute specifies the shape of the area such as **rectangle**, **circle**, **square**, and **polygon**. The coords attribute defines the coordinates of areas inside the image.

## What is Image-map

An image-map is defined as a graphical image with active areas so that when user click on those area, it can link to different destinations. **To define an image-map, we require the following things:**

* An HTML <img> element with **usemap** attribute which defines a valid map name.
* HTML <map> element with **name** attribute whose value must be same as **usemap**
* One or more <area> elements inside a <map> element which create clickable areas in an image-map.

### **Syntax**

1. **<area** shape="" coords="" href=""**>**

**Following are some specifications about the HTML <area> tag**

|  |  |
| --- | --- |
| **Display** | **Block** |
| **Start tag/End tag** | Only start tag(End tag forbidden) |
| **Usage** | Image Map |

### **Example**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**HTML Area tag**</title>**
5. **<style>**
6. body{
7. margin-left: 250px;}
8. **</style>**
9. **</head>**
10. **<body>**
11. **<h2>**Example of HTML Area tag**</h2>**
12. **<img** src="image1.png" usemap="#web"**>**
13. **<map** name="web"**>**
14. **<area** shape="rect" coords="66,117,131,168" href="https://www.iHub.com/html-tutorial"**>**
15. **<area** shape="rect" coords="199,36,277,85" href="https://www.iHub.com/css-tutorial"**>**
16. **<area** shape="rect" coords="330,107,406,159" href="https://www.iHub.com/bootstrap-tutorial"**>**
17. **<area** shape="rect" coords="199,185,267,236" href="https://www.iHub.com/javascript-tutorial"**>**
18. **</map>**
19. **</body>**
20. **</html>**

## Attribute:

### **Tag-specific attributes:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| alt | text | An alternative text String to display on the browser if it does not display the image. |
| coords | x1,y1,x2,y2(rect) | Defines coordinates for the upper left and lower right of a rectangle. |
| x,y, radius(circle) | Defines coordinates for the circle. |
| x1,y1,x2,y2,x3,y3,..(polygon) | Defines the polygon vertices. |
| href | href | URL It determines the hyperlink destination for the active area. |
| target | \_blank | Open link in a new window |
| \_parent | Open link in the parent frame |
| \_self | Open link in current window |
| \_top | Open link with full width in the same window |
| frame\_name | In the frame. **(Not supported in HTML5)** |
| shape | default | It defines the default area(rectangular). |
| rect | It defines the rectangular area. |
| circle | Defines the circular area. |
| poly | Defines the polygonal. |
| download | filename | Defines that hyperlink, which is used for downloading the resource. |
| relHTML Tags List | alternate author bookmark help license next nofollow noreferrer prefetch prev search tag | It defines the relationship between current and linked document. |
| hreflangHTML Tags List | language\_code | It specifies the language of the linked resource. |
| typeHTML Tags List | media\_type | It specifies the MIME type of linked source.**(Not supported in HTML5)** |

# **HTML Article Tag**

The **HTML <article> tag** defines an independent self-contained content in a document, page, application or a site.

The article tag content makes sense on its own. It is independent and complete from other content shown on the page. This tag is generally used on Forum post, Blog post, News story, comment etc.

## HTML article tag example

<!DOCTYPE>

<html>

<body>

<article>

<h2>Narendra Modi</h2>

<i>(Naam to suna hi hoga) </i>

<p>Narendra DamodarDas Modi is the 15th and current Prime Minister of India,

Modi, a leader of the Bharatiya Janata Party (BJP), previously served as the Chief Minister

of Gujarat state from 2001 to 2014. He is currently the Member of Parliament (MP) from Varanasi. </p>

</article>

</body>

</html>

# **HTML Aside Tag**

The HTML <aside> tag provides information about the main content. According to W3C definition, the <aside> element represents content that forms the main textual flow of a document.

HTML aside is a new tag introduced in HTML5.

## HTML aside tag example

1. **<p>**I don't want to live in Ghaziabad, I wish; I could buy a flat in New Delhi.**</p>**
2. **<aside>**
3. **<h3>**New Delhi**</h3>**
4. **<p>**New Delhi is the capital of India.**</p>**
5. **</aside>**

# **HTML Audio Tag**

**HTML audio tag** is used to define sounds such as music and other audio clips. Currently there are three supported file format for HTML 5 audio tag.

1. mp3
2. wav
3. ogg

HTML5 supports <video> and <audio> controls. The Flash, Silverlight and similar technologies are used to play the multimedia items.

This table defines that which web browser supports which audio file format.

## HTML Audio Tag Example

Let's see the code to play mp3 file using HTML audio tag.

<!DOCTYPE>

<html>

<body>

<audio controls>

<source src="koyal.mp3" type="audio/mpeg">

Your browser does not support the html audio tag.

</audio>

</body>

</html>

Attributes of HTML Audio Tag

There is given a list of HTML audio tag.

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| controls | It defines the audio controls which is displayed with play/pause buttons. |
| autoplay | It specifies that the audio will start playing as soon as it is ready. |
| loop | It specifies that the audio file will start over again, every time when it is completed. |
| muted | It is used to mute the audio output. |
| preload | It specifies the author view to upload audio file when the page loads. |
| src | It specifies the source URL of the audio file. |

## HTML Audio Tag Attribute Example

Here we are going to use controls, autoplay, loop and src attributes of HTML audio tag.

<!DOCTYPE>

<html>

<body>

<audio controls autoplay loop>

<source src="koyal.mp3" type="audio/mpeg"></audio>

</body>

</html>

MIME Types for HTML Audio format

The available MIME type HTML audio tag is given below.

|  |  |
| --- | --- |
| **Audio Format** | **MIME Type** |
| mp3 | audio/mpeg |
| ogg | audio/ogg |
| wav | audio/wav |

# **HTML bold tag**

HTML bold tag is represented by <b> tag.

**HTML <b> tag** is used to display the written text in bold format. It is strictly a presentational element. If you want to show your text in bold letters and not have real semantic meaning, then put it within <b>.......</b> tag.

## Difference between HTML <b> and <strong> tag

The b tag is explicit whereas strong tag is semantic.The <strong> tag adds extra semantic meaning to the HTML document.

It is recommended to use strong tag for bold format now.

## HTML bold tag example

<!DOCTYPE>

<html>

<body>

<p> Hello guys, <b>this is the method to write bold text.</b></p>

</body>

</html>

#### **Note: According to HTML5 specification, b tag should be used only if no other tag is appropriate. For example: If you want to write a heading, you must use the header tag <h1> to <h6>. Important statement should be denoted within <strong>.....</strong> tag, and the text you want to mark or highlight, must be put within <mark>...</mark> tag.**

There is also a CSS 'font-weight' property to set bold text.

# **HTML <base> tag**

HTML <base> tag is used to specify a base URI or URL to use for all relative links contained within an HTML document.

Only one <base> element can be specified within a document, and it must be placed within <head> element. We can also specify how other links should open (in the same window, in another window, etc.) using the target attribute.

### **Syntax**

1. <base href="https://www.iHub.com">

In HTML <base> element does not contain closing tag but in XHTML end tag </base> is required.

**Following are some specifications about the <base> tag**

|  |  |
| --- | --- |
| **Display** | **None** |
| **Start tag/End tag** | Only start-tag |
| **Usage** | Anchors and links |

### **Example**

Open in same window (by default)

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>Base tag</title>
5. <style>
6. a{text-decoration: none;
7. color:black;}
8. a:hover{color: green;
9. font-size: 18px;}
10. </style>
11. <base href="https://www.iHub.com">
12. </head>
13. <body>
14. <h2 style="color: red; font-size: 30px; font-style: italic;">List of Web developement Tutorial</h2>
15. <a href="/html-tutorial">HTML tutorial</a><br>
16. <a href="/css-tutorial">CSS tutorial</a><br>
17. <a href="/bootstrap-tutorial">Bootstrap tutorial</a><br>
18. <a href="/javascript-tutorial">JavaScript tutorial</a><br>
19. <a href="/jquery-tutorial">JavaScript tutorial</a>
20. </body>
21. </html>

In the above example we have used base URL =" <https://www.iHub.com>" which all other relative links will treat as starting URL. Here (/) defines the root URL of the current document.

## Open in new window (using \_blank in <base> tag):

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>Base tag</title>
5. <style>
6. a{text-decoration: none;
7. color:black;}
8. a:hover{color: green;
9. font-size: 18px;}
10. </style>
11. <base href="https://www.iHub.com" target="\_blank">
12. </head>
13. <body>
14. <h2 style="color: red; font-size: 30px; font-style: italic;">List of Web developement Tutorial</h2>
15. <a href="/html-tutorial">HTML tutorial</a><br>
16. <a href="/css-tutorial">CSS tutorial</a><br>
17. <a href="/bootstrap-tutorial">Bootstrap tutorial</a><br>
18. <a href="/javascript-tutorial">JavaScript tutorial</a><br>
19. <a href="/jquery-tutorial">JavaScript tutorial</a>
20. </body>
21. </html>

In the above example we have used **target="\_blank"** in only <base> tag but it is applied on whole document links.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Definition** |
| href | URL | It specifies the base URL for all relative links. |
| target |  |  |
|  | \_blank | Open the relative link in the new window |
|  | \_self | Open the relative link in current window |
|  | \_parent | Open the relative link in the parent frame |
|  | \_top | Open the links in full width of the page |

# **HTML <basefont> tag**

(Not Supported in HTML5)

HTML <basefont> tag was used to specify the default value of font-size, color, and font-family for all content written within an HTML document.

#### **Note: The <basefont> was deprecated in HTML 4 and completely removed from HTML5 so do not use this tag, instead of it you can use CSS to style the document.**

### **Syntax**

1. <basefont color="blue" size="5" face="arial">

In HTML the closing tag </basefont> is not required but in XHTML the end tag is required.

**Following are some specifications about the <base> tag**

|  |  |
| --- | --- |
| **Display** | **None (empty tag)** |
| **Start tag/End tag** | Only start-tag |
| **Usage** | Font styling |

### **Example**

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>Basefont tag</title>
5. <basefont color="blue" size="5" face="arial">
6. </head>
7. <body>
8. <h2>Example of Basefont tag</h2>
9. <p>The basefornt tag is not supported in HTML5 use CSS to style the document</p>
10. </body>
11. </html>

## Right way is to Use CSS to set the base font:

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. p{
6. font-size: 20px;
7. color: #67dfee;
8. font-family: Helvetica;
9. }
10. </style>
11. </head>
12. <body>
13. <h2>CSS to Style the document</h2>
14. <p>This is Paragraph</p>
15. </body>
16. </html>
17. Attributes
18. **Tag specific attributes:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| color | color | Specify the default color of all text within an HTML document. (Not Supported in HTML5) |
| face | font-family | Specify the default font face for the text in the document. (Not Supported in HTML5) |
| size | number | Specify the size of the text. (Not Supported in HTML5) |

# **HTML <bdi> tag**

HTML <bdi> tag stands for Bidirectional Isolate Element. It is used to inform the browser to isolate the span of text which may be formatted in opposite directions than the surrounding text.

This tag is new in HTML5 and useful for languages such as Arabic or Hebrew, or if the browser dynamically inserts some text without knowing the directionality of text.

#### **Note: The directionality of content inside <bdi> does not influence by the directionality of surrounding content.**

### **Syntax**

1. <bdi>Content</bdi>

**Following are some specifications about the <bdi> tag**

|  |  |
| --- | --- |
| **Display** | **Inline** |
| **Start tag/End tag** | Both start and End tag |
| **Usage** | Semantic/textual |

### **Example**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Bdi tag**</title>**
5. **<style>**
6. bdi{
7. font-size: 25px;
8. color: red; }
9. **</style>**
10. **</head>**
11. **<body>**
12. **<h2>**Example of HTML bdi tag**</h2>**
13. **<p>**
14. This **<bdi>** ???? ????? **</bdi>**  Content is written in Arabic language
15. **</p>**
16. **</body>**
17. **</html>**

## Difference between <bdi> and <bdo>

HTML <bdi> tag is new element of HTML5 and it is very similar to HTML <bdo> element. But the main difference in both the elements is that <bdi> tag isolates the content from its surrounding content but <bdo> tag reveres the direction. Sometimes <bdo> tag may cause some unexpected rendering errors so try to use <bdi> if required.

# **HTML <bdo> tag**

HTML <bdo> tag stands for "bidirectional override" which is used to overrides the current/default text direction. This tag sets the direction of content within it to render on browser from left to right or right to left.

The <bdo> tag is useful for the languages which are written from right to left such as Arabic and Hebrew.

### **Syntax**

1. <bdo dir=" "> Content......</bdo>

**Following are some specifications about the <bdo> tag**

|  |  |
| --- | --- |
| **Display** | **Inline** |
| **Start tag/End tag** | Both start and End tag |
| **Usage** | textual |

### **Example**

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>Bdo tag</title>
5. </head>
6. <body>
7. <h2>Example of HTML bdo tag</h2>
8. <p style="color:#ff4040">This is Normal Content with **default** directionality</p>
9. <p>This is with left to right directionality:
10. <bdo dir="ltr" style="color: #5f9ea0">A beautiful Flower</bdo>
11. </p>
12. <p>This is with right to left directionality:
13. <bdo dir="rtl" style="color:#68228b">A beautiful Flower</bdo>
14. </p>
15. </body>
16. </html>
17. Attributes
18. **Tag-specific attributes**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| dir |  |  |
|  | ltr | Defines the directionality from left to right. |
|  | rtl | Defines the directionality from right to left. |

# **HTML <big> tag (Not supported in HTML5)**

HTML <big> tag was used to increase the text font size one level bigger than the document's base font size or surrounding text size, such as small to medium, medium to large, etc.

#### **NOTE: HTML <big> tag is no longer supported by HTML5, so you can use CSS to increase the font size.**

### **Syntax**

1. <big>Content... </big>

**Following are some specifications about the <big> tag**

|  |  |
| --- | --- |
| **Display** | **Inline** |
| **Start tag/End tag** | Both start and End tag |
| **Usage** | Fonts and Web Typography |

### **Example**

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>Big tag</title>
5. </head>
6. <body>
7. <h2>Example of HTML big tag</h2>
8. <p style="color:#ff4040">This is paragraph with normal font size</p>
9. <p style="color:#00bfff">
10. <big>This is paragraph with font size greater than the previous content</big>
11. </p>
12. </body>
13. </html>

## Increasing Font-size using CSS property:

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. .fsize{
6. font-size: 18px;
7. color: red;
8. }
9. </style>
10. </head>
11. <body>
12. <h2>Example to change font size using CSS</h2>
13. <p>The paragraph content with **default** font size</p>
14. <p **class**="fsize">The paragraph content with CSS to increase font size</p>
15. </body>
16. </html>

# **HTML <blockquote> tag**

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.

If you want to insert a long quote then use <blockquote> and for short or inline quote use <q> tag.

### **Syntax**

1. <blockquote> quoted text......</blockquote>

**Following are some specifications about the <blockquote> tag**

|  |  |
| --- | --- |
| **Display** | **Block** |
| Start tag/End tag | Both start and End tag |
| Usage | Semantic/textual |

### **Example**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Blockquote tag**</title>**
5. **</head>**
6. **<body>**
7. **<h2>**Example of blockquote tag**</h2>**
8. **<p>**A Great Motivational Quote :**</p>**
9. **<blockquote** cite="https://www.brainyquote.com/authors/erin\_cummings"**>**
10. **<p>**
11. At the end of the day, you are solely responsible for your success and your failure. And the sooner you realize that, you accept that, and integrate that into your work ethic, you will start being successful. As long as you blame others for the reason you aren't where you want to be, you will always be a failure.
12. **</p>**
13. **</blockquote>**
14. **<cite>**Erin Cummings**</cite>**
15. **</body>**
16. **</html>**

## Styling <blockquote> tag

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Blockquote tag**</title>**
5. **<style>**
6. blockquote{
7. text-align: justify;
8. background-color: #F7EAE9;
9. border-radius: 4px;
10. margin-right: 25px;}
11. cite{
12. margin-left: 15px;}
13. **</style>**
14. **</head>**
15. **<body>**
16. **<h2>**Example of blockquote tag**</h2>**
17. **<p>**A Great Motivational Quote:**</p>**
18. **<blockquote** cite="https://www.brainyquote.com/authors/erin\_cummings"**>**
19. **<p>**
20. At the end of the day, you are solely responsible for your success and your failure. And the sooner you realize that, you accept that, and integrate that into your work ethic, you will start being successful. As long as you blame others for the reason you aren't where you want to be, you will always be a failure.
21. **</p>**
22. **</blockquote>**
23. **<cite>**-Erin Cummings**</cite>**
24. **</body>**
25. **</html>**

## Attribute

### **Tag specific attribute**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **value** | **Description** |
| citeHTML blockquote tag | URL | It is used to specify the URL of the source from where the quote is taken. |

# **HTML <body> tag**

HTML <body> tag defines the main content of an HTML document which displays on the browser. It can contain text content, paragraphs, headings, images, tables, links, videos, etc.

The <body> must be the second element after the <head> tag or it should be placed between </head> and </html> tags. This tag is required for every HTML document and should only use once in the whole HTML document.

### **Syntax**

1. <body> Place your Content here........</body>

**Following are some specifications about the <body> tag**

|  |  |
| --- | --- |
| **Display** | **Inline** |
| **Start tag/End tag** | Both start and End tag |
| **Usage** | Structural |

### **Example**

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>Body Tag</title>
5. </head>
6. <body>
7. <h2>Example of body tag</h2>
8. <p>This paragraph is written between the body tag</p>
9. </body>
10. </html>

## Attribute

### **Tag specific Attributes**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| alink | color | It defines the color of the active link in a document. (Not supported in HTML5) |
| background | URL | It determines the background image for the document. (Not supported in HTML5) |
| bgcolor | color | It determines the background color of the content. (Not supported in HTML5) |
| link | color | It determines the color of the unvisited link. (Not supported in HTML5) |
| text | color | It determines the color of the text in the document. (Not supported in HTML5) |
| vlink | color | It determines the color of the visited link. (Not supported in HTML5) |
| onload |  | Function call on page loading |
| onunload |  | Function call when user leaves the page |
| onfocus |  | Function call when document receives focus by user. |
| onblur |  | Function call when document loses focus by user. |

# **HTML <br> tag**

The **<br> tag** in **HTML** document is used to create a line break in a text.

It is generally used in poem or address where the division of line is necessary. It is an empty tag, which means it does not need a company of end tag. If you place the <br> tag in the HTML code, then it works the same as pressing the enter key in a word processor.

#### **Note: Don't use br tag for the margin between two paragraphs, use CSS margin property instead.**

### **Syntax**

1. Text **<br>** Text

## Difference between HTML <br> and <br/>

You can use HTML br tag two ways: <br> or <br/>. It is recommended to use closed br tag <br/> because it is supported in HTML and XHTML both.

**Example 1:**

1. <!DOCTYPE HTML**>**
2. **<html>**
3. **<head>**
4. **<title>**
5. Example of BR tag
6. **</title>**
7. **</head>**
8. **<body>**
9. **<p>**If you want to break a line **<br>** in a paragraph, **<br>** use the BR element in **<br>** your HTML document. **</p>**
10. **</body>**
11. **</html>**

**Example 2:** The following example uses the <br> tag in the address:

1. <!DOCTYPE HTML**>**
2. **<html>**
3. **<head>**
4. **<title>**
5. Example of BR tag in Address
6. **</title>**
7. **</head>**
8. **<body>** **<font** color="green"**>** <!-The color attribute displays the color of following address in green --**>**
9. **<p>**Central park **<br>**
10. D.M Road, **<br>**
11. New Delhi, **<br>**
12. India **<br>**
13. **</body>**
14. **</html>**

**Example 2:** The following example uses the <br> tag in the poem:

1. <!DOCTYPE HTML**>**
2. **<html>**
3. **<head>**
4. **<title>**
5. Example of BR tag in Poem
6. **</title>**
7. **</head>**
8. **<body>** **<font** color="green"**>**
9. **<center>** Dreams
11. **</center>**
12. Hold fast to dreams **<br>**
14. For if dreams die **<br>**
16. Life is a broken-winged bird **<br>**
18. That cannot fly. **<br>**
20. Hold fast to dreams **<br>**
22. For when dreams go **<br>**
24. Life is a barren field **<br>**
26. Frozen with snow. **<br>**
27. **</body>**
28. **</html>**

# **HTML Button Tag**

The <button> tag is used to create a clickable button within HTML form on your webpage. You can put content like text or image within the <button>........</button> tag.

You should always specify the type attribute for a <button> tag. Different browsers use different default type for the button element.

HTML Button tag can be used inside and outside the form.

If you use it **inside the form**, it works as the submit button. You can also use it as reset button.

If you use it **outside the form**, you can call JavaScript function on it.

## HTML Button Tag Example

Let's see the code to display the button.

<!DOCTYPE>

<html>

<body>

<form>

<button name="button" value="OK" type="button">Click Here</button>

</form>

</body>

</html>

## HTML Button Example: Calling JavaScript Function

Let's see the code to call JavaScript function on button click.

1. **<button** name="button" value="OK" type="button" onclick="hello()"**>**Click Here**</button>**
2. **<script>**
3. function hello(){
4. alert("hello iHub user");
5. }
6. **</script>**

## HTML Button Example: Submit Form

Let's see the code to submit form on button click.

1. **<form>**
2. Enter Name:**<input** type="text" name="name"**/><br/>**
3. **<button>**Submit**</button>**
4. **</form>**

## HTML Button Example: Reset Form

Let's see the code to submit form on button click.

1. **<form>**
2. Enter Name:**<input** type="text" name="name"**/><br/>**
3. **<button** type="reset"**>**reset**</button>**
4. **</form>**
5. Attributes of HTML Button Tag
6. <button> tag supports all global attributes and some specific additional attributes.
7. There is given a list of HTML button tag attributes.

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| autofocus | It specifies that a button should automatically get focus while the loading of the page. |
| disabled | It specifies that a button shuld be disabled. |
| form | It specifies one or more forms that the button belongs to. |
| formaction | It is used for submit type. It specifies where to send the form data when form is submitted. |
| formmethod | It specifies how to send form-data. |
| formenctype | It specifies how form-data should be encoded before sending it to server. |
| formnovalidate | It specifies that the form data should not be validated on submission. |
| formtarget | It specifies that where to display the response after submitting the form. |
| name | It specifies the name of the button. |
| type | It specifies the type of the button. |
| value | It specifies the value of the button. |

# **HTML Canvas Tag**

The **HTML canvas element** provides HTML a bitmapped surface to work with. It is used to draw graphics on the web page.

The **HTML 5 <canvas> tag** is used to draw graphics using scripting language like JavaScript.

The <canvas> element is only a container for graphics, you must need a scripting language to draw the graphics. The <canvas> element allows for dynamic and scriptable rendering of 2D shapes and bitmap images.

It is a low level, procedural model that updates a bitmap and does not have a built-in scene. There are several methods in canvas to draw paths, boxes, circles, text and add images.

## How to create a HTML canvas?

A canvas is a rectangle like area on an HTML page. It is specified with canvas element. By default, the <canvas> element has no border and no content, it is like a container.

1. **<canvas** id = "mycanvas" width ="200" height ="100"**>** **</canvas>**

## HTML 5 Canvas Tag Example

<!DOCTYPE>

<html>

<body>

<canvas id="myCanvas" width="300" height="200" style="border:2px solid;">

Your browser does not support the HTML5 canvas tag.

</canvas>

</body>

</html>

#### **Note: It is always necessary to specify the id attribute and the height & width attribute to define the size of the canvas. You can have multiple canvas elements on one HTML page.**

## HTML Canvas Tag with JavaScript

A canvas is a two dimensional grid.

Coordinates (0,0) defines the upper left corner of the canvas. The parameters (0,0,200,100) is used for fillRect() method. This parameter will fill the rectangle start with the upper-left corner (0,0) and draw a 200 \* 100 rectangle.

1. **<canvas** id="myCanvas" width="250" height="150" style="border:1px solid #c3c3c3;"**>**
2. Your browser does not support the HTML5 canvas tag.
3. **</canvas>**
4. **<script>**
5. var c = document.getElementById("myCanvas");
6. var cctx = c.getContext("2d");
7. ctx.fillStyle = "#FF0000";
8. ctx.fillRect(0,0,200,100);
9. **</script>**

## Drawing Line on Canvas

If you want to draw a straight line on the canvas, you can use the following two methods.

**moveTo(x,y):** It is used to define the starting point of the line.

**lineTo(x,y):** It is used to define the ending point of the line.

If you draw a line which starting point is (0,0) and the end point is (200,100), use the stroke method to draw the line.

1. **<canvas** id="myCanvasLine" width="200" height="100" style="border:1px solid #d3d3d3;"**>**
2. Your browser does not support the HTML5 canvas tag.**</canvas>**
3. **<script>**
4. var c = document.getElementById("myCanvasLine");
5. var cctx = c.getContext("2d");
6. ctx.moveTo(0,0);
7. ctx.lineTo(200,100);
8. ctx.stroke();
9. **</script>**

## Drawing Circle on Canvas

If you want to draw a circle on the canvas, you can use the arc() method:

1. arc(x, y, r, start, stop)

To sketch circle on HTML canvas, use one of the ink() methods, like stroke() or fill().

1. **<canvas** id="myCanvasCircle" width="200" height="100" style="border:1px solid #d3d3d3;"**>**
2. Your browser does not support the HTML5 canvas tag.**</canvas>**
3. **<script>**
4. var c = document.getElementById("myCanvasCircle");
5. var cctx = c.getContext("2d");
6. ctx.beginPath();
7. ctx.arc(95,50,40,0,2\*Math.PI);
8. ctx.stroke();
9. **</script>**

## Drawing text on canvas

There are property and methods used for drawing text on the canvas.

**font property:** It is used to define the font property for the text.

**fillText(text,x,y) method:** It is used to draw filled text on the canvas. It looks like bold font.

**strokeText(text,x,y) method:** It is also used to draw text on the canvas, but the text is unfilled.

Let's see **fillText()** method example.

1. **<canvas** id="myCanvasText1" width="300" height="100" style="border:1px solid #d3d3d3;"**>**
2. Sorry! Your browser does not support the HTML5 canvas tag.**</canvas>**
3. **<script>**
4. var c = document.getElementById("myCanvasText1");
5. var cctx = c.getContext("2d");
6. ctx.font = "30px Arial";
7. ctx.fillText("Hello iHub",10,50);
8. **</script>**

Let's see **strokeText()** method example.

1. **<canvas** id="myCanvasText2" width="300" height="100" style="border:1px solid #d3d3d3;"**>**
2. Sorry!Upgrade your browser. It does not support the HTML5 canvas tag.**</canvas>**
3. **<script>**
4. var c = document.getElementById("myCanvasText2");
5. var cctx = c.getContext("2d");
6. ctx.font = "30px Arial";
7. ctx.strokeText("Hello iHub",10,50);
8. **</script>**

# **HTML <caption> tag**

HTML <caption> tag is used to add a caption or title of an HTML table. It should be used inside the <table> element and just after the <table> start tag. A table may contain only one <caption> element.

### **Syntax**

1. **<caption>**Table title...**</caption>**

**Following are some specifications about the <caption> tag**

|  |  |
| --- | --- |
| **Display** | **Inline** |
| **Start tag/End tag** | Both start and End tag |
| **Usage** | textual |

### **Example 1**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Caption Tag**</title>**
5. **<style>**
6. table, td, th {
7. border: 3px solid gray;
8. border-collapse: collapse;}
9. **</style>**
10. **</head>**
11. **<body>**
12. **<h2>**Example of Caption tag**</h2>**
13. **<table** width="800"**>**
14. **<caption>**Employee Details**</caption>**
15. **<thead>**
16. **<tr>**
17. **<th>**Sr. No.**</th>**
18. **<th>**Name**</th>**
19. **<th>**Email**</th>**
20. **</tr>**
21. **</thead>**
22. **<tbody>**
23. **<tr>**
24. **<td>**1.**</td>**
25. **<td>**Ankit Pandey**</td>**
26. **<td>**ankit2@gmail.com**</td>**
27. **</tr>**
28. **<tr>**
29. **<td>**2.**</td>**
30. **<td>**Ashvini Kumar**</td>**
31. **<td>**ashvini@gmail.com**</td>**
32. **</tr>**
33. **<tr>**
34. **<td>**3.**</td>**
35. **<td>**Swati Sharma**</td>**
36. **<td>**swati8@gmail.com**</td>**
37. **</tr>**
38. **</tbody>**
39. **</table>**
40. **</body>**
41. **</html>**

## Attribute

### **Tag-specific Attributes**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| align | * top * bottom * left * right | It aligns the caption with respect to table |

# **HTML <center> tag (Not Supported in HTML5)**

The HTML <center> is a block level element which contains both block level and inline contents within it. The content written between the <center> elements will be displayed at the middle of the page.

The <center> tag has been deprecated in HTML 4 and obsolete in HTML5.

#### **NOTE: The <center> tag is deprecated now so instead of that you can use CSS property text-align: center; to achieve the same result.**

### **Syntax**

1. **<center>**Add Content Here....**</center>**

**Following are some specifications about the <center> tag**

|  |  |
| --- | --- |
| **Display** | **Inline** |
| **Start tag/End tag** | Both start and End tag |
| **Usage** | textual |

### **Example**

**Using <center> tag**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Center tag**</title>**
5. **</head>**
6. **<body>**
7. **<h2>**Example of center tag**</h2>**
8. **<center>**This content is displayed in the middle of page, but try to use CSS property to align the content as this tag is deprecated now.**</center>**
9. **</body>**
10. **</html>**

**Using CSS**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Center tag**</title>**
5. **<style** **>**
6. h2{
7. text-align: center;}
8. **</style>**
9. **</head>**
10. **<body>**
11. **<h2>**Example of center tag**</h2>**
12. **<p>**This content is align centered using CSS property**</p>**
13. **</body>**
14. **</html>**

## Attribute

HTML <center> tag does not contain any specific attribute in HTML but it supports the Global Attribute (till <center> tag is not removed completely).

# **HTML <cite> tag**

Refer to <cite> examples above in this document

# **HTML <code> tag**

Refer to <cite> examples above in this document

# **HTML <col> tag**

HTML <col> tag is used to specify common column properties for each column within <colgroup> element in an HTML table.

The <col> tag is useful when it requires to apply common style rules for each column cell rather than applying individually.

### **Syntax**

1. **<col** span=" "   style=" "**>**

#### **Note: In HTML <col> tag does not require a closing tag but in XHTML it require a closing </col> tag.**

**Following are some specifications about the HTML <col> tag**

|  |  |
| --- | --- |
| **Display** | **Inline** |
| **Start tag/End tag** | Start tag(Empty tag) |
| **Usage** | HTML tables |

### **Example 1**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Col tag**</title>**
5. **</head>**
6. **<body>**
7. **<h2>**Example of Col Tag**</h2>**
8. **<table** border="1"**>**
9. **<colgroup>**
10. **<col**  span="2" style="background-color: #7fffd4"**>**
11. **<col** span="2" style="background-color:   #98f5ff"**>**
12. **</colgroup>**
13. **<tr>**
14. **<th>**Roll No.**</th>**
15. **<th>**Name**</th>**
16. **<th>**Maths**</th>**
17. **<th>**Science**</th>**
18. **</tr>**
19. **<tr>**
20. **<td>**1**</td>**
21. **<td>**Avinash**</td>**
22. **<td>**68**</td>**
23. **<td>**54**</td>**
24. **</tr>**
25. **<tr>**
26. **<td>**2**</td>**
27. **<td>**Neha**</td>**
28. **<td>**65**</td>**
29. **<td>**61**</td>**
30. **</tr>**
31. **<tr>**
32. **<td>**3**</td>**
33. **<td>**Jenny**</td>**
34. **<td>**55**</td>**
35. **<td>**78**</td>**
36. **</tr>**
37. **<tr>**
38. **<td>**4**</td>**
39. **<td>**Vivek**</td>**
40. **<td>**75**</td>**
41. **<td>**52**</td>**
42. **</tr>**
43. **</table>**
44. **</body>**
45. **</html>**

## Attribute:

**Tag-specific Attribute**

element should contain. The default value of span attribute is 1.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| align | * left * center * right * justify * char | It specifies the horizontal alignment of each column cell. (Not supported in HTML5). |
| char | character | It specifies the alignment of content based on Characters in a column. It will be ignored if align is not set to char. (Not supported in HTML5). |
| charoff | number | It sets the number of characters to offset the column data from the alignment character specified by the char attribute. (Not supported in HTML5). |
| span | number | It specifies the number of the columns which a |
| valign | * top * middle * bottom * baseline | It specifies the vertical alignment of the columns. (Not supported in HTML5). |
| width | * % * Pixels * relative\_length | It specifies the width of the column. (Not supported in HTML5). |

# **HTML <colgroup> tag**

HTML <colgroup> tag specifies group of column within an HTML table. It is used a parent container of one or more <col> elements to apply different properties in an HTML table.

#### **NOTE: The <colgroup> tag must be used with <table> element after the <caption> and before <thead> or <tbody> elements.**

### **Syntax**

1. **<colgroup>**......**</colgroup>**

**Following are some specifications about the HTML <colgroup> tag**

|  |  |
| --- | --- |
| **Display** | **None** |
| **Start tag/End tag** | Start tag and End Tag |
| **Usage** | HTML tables |

### **Example**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Colgroup tag**</title>**
5. **</head>**
6. **<body>**
7. **<h2>**Example of Colgroup Tag**</h2>**
8. **<table** border="1"**>**
9. **<colgroup>**
10. **<col**  style="background-color: green" width="40"**>**
11. **<col** span="2" style="background-color:   #ff7256" width="80"**>**
12. **</colgroup>**
13. **<tr>**
14. **<th>**Sr.No**</th>**
15. **<th>**Product**</th>**
16. **<th>**Price**</th>**
17. **</tr>**
18. **<tr>**
19. **<td>**1**</td>**
20. **<td>**Rice**</td>**
21. **<td>**85**</td>**
22. **</tr>**
23. **<tr>**
24. **<td>**2**</td>**
25. **<td>**Butter**</td>**
26. **<td>**260**</td>**
27. **</tr>**
28. **<tr>**
29. **<td>**3**</td>**
30. **<td>**Mango**</td>**
31. **<td>**125**</td>**
32. **</tr>**
33. **</table>**
34. **</body>**
35. **</html>**

## Attributes

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| align | * left * center * right * justify * char | It specifies the alignment of the column content. (Not supported in HTML5). |
| char | character | It specifies the alignment of the content in a column group to the character. (Not supported in HTML5). |
| charoff | number | It sets the number of characters to offset the column data from the alignment character specified by the char attribute. (Not supported in HTML5). |
| span | number | It specifies the number of columns a colgroup should span. |
| valign | * top * middle * bottom * baseline | It specifies the vertical alignment of the column group. (Not supported in HTML5). |
| width | * % * Pixels * relative\_length | It specifies the width of the group of column. (Not supported in HTML5). |

# **HTML Data Tag**

The HTML <data> tag is used to provide a machine readable version of its own contents. It displays the data in a special format. It is useful in the case where your data needs to be in a certain format because it may be processed by a script, but this might not be the format that you'd like your users to see.

Let's take a scenario to understand it well:

Suppose you have to represent some numbers to your users with letter (i.e. one, two, three and so on) but you have script that sorts the numbers in ascending or descending orders. So your script needs numbers in a format like 1, 2, 3... etc.

The <data> tag is here to solve your problem.

For the users: Write within the <data>....</data> tag.

For the script: Use value attribute.

1. **<data** value="1"**>**One **</data>**

## HTML data tag attribute

There is only one required attribute. The data tag supports global and event attributes also.

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| value | It is required attribute. It is used to provide the machine-readable version of tag's content. |

## HTML data tag example

<!DOCTYPE>

<html>

<body>

<ul>

<li><data value="101">Java Tutorial</data></li>

<li><data value="111">SQL tutorial</data></li>

<li><data value="121">HTML tutorial</data></li>

</ul>

</body>

</html>

# **HTML Datalist Tag**

The HTML <datalist> tag is is used to provide an auto complete feature on form element. It provides a list of predefined options to the users to select data.

The datalist tag is introduced in HTML5.

The <datalist> tag should be used with an <input< element that contains a "list" attribute. The value of "list" attribute is linked with the datalist id.

#### **The HTML datalist tag supports global and event attributes also.**

## HTML datalist tag example

Let's see the simple example of HTML5 datalist tag. If you press A, it will show a list of cricketers starting with A letter.

1. **<label>**
2. Enter your favorite cricket player: Press any character**<br** **/>**
3. **<input** type="text" id="favCktPlayer" list="CktPlayers"**>**
4. **<datalist** id="CktPlayers"**>**
5. **<option** value="Sachin Tendulkar"**>**
6. **<option** value="Brian Lara"**>**
7. **<option** value="Jacques Kallis"**>**
8. **<option** value="Ricky Ponting"**>**
9. **<option** value="Rahul Dravid"**>**
10. **<option** value="Shane Warne"**>**
11. **<option** value="Rohit Sharma"**>**
12. **<option** value="Donald Bradman"**>**
13. **<option** value="Saurav Ganguly "**>**
14. **<option** value="AB diVilliers"**>**
15. **<option** value="Mahendra Singh Dhoni"**>**
16. **<option** value="Adam Gilchrist"**>**
17. **</datalist>**
18. **</label>**

# **HTML Details Tag**

HTML <details> tag is used to specify the additional details on the web page that the user can view or hide on demand.

According to W3C HTML specification, it is used as a disclosure widget from which user can retrieve additional information or control.

It is used together with a relevant tag known as <summary>. Technically, there is no need of summary tag, but if you ignore this then the browser will use some default text.

The <details> tag wraps all the content which you want to show or hide and the <summary> tag contains the summary and the title of the section.

HTML details is a new tag introduced in HTML5.

Let's take an example to understand this clearly.

## HTML details tag attribute

HTML details tag also supports global and event attributes in HTML.

The details tag provides one specific attribute open.

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| open | It specifies that the details will be displayed (open) to the user. |

## HTML details tag example

<!DOCTYPE>

<html>

<body>

<details>

<summary>Copyright 2011-2014.</summary>

<p> - by iHub. All Rights Reserved.</p>

<p>All content and graphics on this web site are the property of the iHub.com</p>

</details>

<p><b>Note:</b> The details tag is currently only supported in Opera, Chrome, and in Safari 6.</p>

</body>

</html>

# **HTML Dialog Tag**

**HTML <dialog> tag** is used to create a new popup dialog on a web page. This tag represents a dialog box or other interactive component like window.

The <dialog> element uses a boolean attribute called open that activate element and facilitate user to interact with it.

HTML dialog is a new tag introduced in HTML5.

## HTML dialog tag example

<!DOCTYPE>

<html>

<body>

<div>

<dialog id="myFirstDialog" style="width:50%;background-color:#F4FFEF;border:1px dotted black;">

<p><q>I am so clever that sometimes I don't understand a single word of what I am saying.

</q> - <cite>Oscar Wilde</cite></p>

<button id="hide">Close</button>

</dialog>

<button id="show">Show Dialog</button>

</div>

<!-- JavaScript to provide the "Show/Close" functionality -->

<script type="text/JavaScript">

(function() {

var dialog = document.getElementById('myFirstDialog');

document.getElementById('show').onclick = function() {

dialog.show();

};

document.getElementById('hide').onclick = function() {

dialog.close();

};

})();

</script>

</body>

</html>

# **HTML <dir> tag (Not Supported in HTML5)**

HTML <dir> tag was used as a container for the list of files/folders or content of the directory. The <dir> element is used with the <li> tags, and list of directories renders in bullets by default.

#### **Note: Do not use HTML <dir> tag as it is deprecated now and not supported in HTML5. You can use <ul> tag and CSS property instead.**

### **Syntax**

1. <dir>Directory list... </dir>

**Following are some specifications about the HTML <dir> tag**

|  |  |
| --- | --- |
| **Display** | **Block** |
| **Start tag/End tag** | Both Start and End tag |
| **Usage** | HTML lists |

### **Example 1**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Dir tag**</title>**
5. **<style>**
6. h2{
7. color: #b22222;}
8. p{
9. color: green;}
10. **</style>**
11. **</head>**
12. **<body>**
13. **<h2>**Example of dir tag**</h2>**
14. **<p>**List of iHub popular Tutorials**</p>**
15. **<dir>**
16. **<li>**Java-tutorial**</li>**
17. **<li>**DBMS-tutorial**</li>**
18. **<li>**DataStructure-tutorial**</li>**
19. **<li>**HTML-tutorial**</li>**
20. **</dir>**
21. **</body>**
22. **</html>**

Attribute

Tag-specific Attribute

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| compact | compact | It specifies that list should display smaller than normal. (Not supported in HTML5) |

# **HTML Div Tag**

The **HTML <div> tag** is used *to group the large section of HTML elements together*.

We know that every tag has a specific purpose e.g. p tag is used to specify paragraph, <h1> to <h6> tag are used to specify headings but the <div> tag is just like a container unit which is used to encapsulate other page elements and divides the HTML documents into sections.

The div tag is generally used by web developers to group HTML elements together and apply CSS styles to many elements at once. For example: If you wrap a set of paragraph elements into a div element so you can take the advantage of CSS styles and apply font style to all paragraphs at once instead of coding the same style for each paragraph element.

Difference between HTML div tag and span tag

|  |  |
| --- | --- |
| **div tag** | **span tag** |
| HTML div is a **block** element. | HTML span is an **inline** element |
| HTML div element is used to **wrap large sections of elements**. | HTML span element is used to **wrap small portion of texts, image** etc. |

## HTML div example: Login Form

In this example, we are creating box using div tag. There is a login form inside the box. Let's see the CSS and HTML code.

CSS Code:

1. .loginform{
2. padding:10px;
3. border:1px solid pink;
4. border-radius:10px;
5. float:right;
6. margin-top:10px;
7. }
8. .formheading{
9. background-color:red;
10. color:white;
11. padding:4px;
12. text-align:center;
13. }
14. .sub{
15. background-color:blue;
16. padding: 7px 40px 7px 40px;
17. color:white;
18. font-weight:bold;
19. margin-left:70px;
20. border-radius:5px;
21. }

HTML Code:

<!DOCTYPE>

<html>

<head>

<style>

.loginform{

padding:10px;

border:1px solid pink;

border-radius:10px;

width:300px;

margin-top:10px;

}

.formheading{

background-color:red;

color:white;

padding:4px;

text-align:center;

}

.sub{

background-color:red;

padding: 7px 40px 7px 40px;

color:white;

font-weight:bold;

margin-left:70px;

border-radius:5px;

}

</style>

</head>

<body>

<div class="loginform">

<h3 class="formheading">Please Login</h3>

<form action="LoginServlet" method="post">

<table>

<tr><td>Email:</td><td><input type="email" name="email"/></td></tr>

<tr><td>Password:</td><td><input type="password" name="password"/></td></tr>

<tr><td colspan="2" style="text-align:center"><input class="sub" type="submit" value="login"/></td></tr>

</table>

</form>

</div>

</body>

</html>

# **HTML <embed> tag**

HTML <embed> tag is used to embed an external document in an HTML file which can be a third party application, multimedia, plugin(interactive content), etc. This element is new in HTML5.

#### **Note: The <embed> element can be used to contain third party applications as well multimedia documents but in HTML5 the newly added <audio> and <video> elements are especially used to embed multimedia in HTML document.**

### **Syntax**

1. **<embed** src=" " **></embed>**
2. **Following are some specifications about the HTML <embed> tag**

|  |  |
| --- | --- |
| **Display** | **block** |
| **Start tag/End tag** | Both Start and End tag |
| **Usage** | Embed third-party applications |

### **Example 1**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Embed Tag**</title>**
5. **<style>**
6. h2{
7. color: #cd5c5c;}
8. p{
9. color: #20b2aa;
10. font-style: italic;}
11. **</style>**
12. **</head>**
13. **<body>**
14. **<h2>**Example of embed tag**</h2>**
15. **<p>**The Dancing Penguin**</p>**
16. **<embed** src="giphy.gif" height="200" width="300"**></embed>**
17. **</body>**
18. **</html>**

Attribute

Tag-specific attribute

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| height | pixels | It specifies the height of the embedded content |
| src | URL | It defines the resource location of the embedded document. |
| type | media-type | It specifies the MIME type. |
| width | pixels | It determines the width of the embedded content. |

# **HTML figcaption tag**

The <figcaption> element is used to provide a caption to an image.

It is an optional tag and can appear before or after the content within the <figure> tag.

Only one <figcaption> element can be nested within a <figure> tag although the <figure> element itself may contain multiple other elements like <img> or <code>.

The <figcaption> element is used with <figure> element and it can be placed as the first or last child of the <figure> element.

## HTML figure tag example

<!DOCTYPE>

<html>

<body>

<figure>

<img src="/htmlpages/images/tajmahal.jpg" alt="Taj Mahal"/>

<figcaption>Fig.1.1 - A front view of the great Taj Mahal in Agra.</figcaption>

</figure>

</body>

</html>

# **HTML figure tag**

**HTML <figure> tag** is used to mark up a photo in the document on a web page.

As we know image tag is already available in HTML to display the pictures on web pages. But HTML 5 <figure> tag is used to handle the group of diagrams, photos, code listing etc. with some embedded content. You can also add a caption for the photo with the help of <figcaption> tag.

HTML figure is new a new tag introduced in HTML5.

## HTML figure tag example

<!DOCTYPE>

<html>

<body>

<p>The Taj Mahal is widely recognized as "the jewel of Muslim art in India and

one of the universally admired masterpieces of the world's heritage". It is regarded

by many as the finest example of Mughal architecture, a style that combines elements

from Islamic, Persian, Ottoman, Turkish and Indian architectural styles.</p>

<figure>

<img src="/htmlpages/images/tajmahal.jpg" alt="Taj Mahal"/>

</figure>

</body>

</html>

#### **Note: The content you put within <figure>.......</figure> tag is related to the main flow, but its position is independent of the main flow and does not affect the flow the document when removed.**

# **HTML <font> tag (Not Supported in HTML5)**

HTML <font> tag is used to define the font style for the text contained within it. It defines the font size, color, and face or the text in an HTML document.

#### **NOTE: Do not use HTML <font> tag as it is deprecated in HTML5, so you can use CSS properties to change the font size, face, color, etc.**

### **Syntax**

1. **<font** size=" " color=" " face=" "**>** Content....**</font>**

**Following are some specifications about the HTML <font> tag**

|  |  |
| --- | --- |
| **Display** | **Inline** |
| **Start tag/End tag** | Both Start and End tag |
| **Usage** | Font Style |

### **Example 1**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Font Tag**</title>**
5. **</head>**
6. **<body>**
7. **<h2>**Example of font tag**</h2>**
8. **<p>**This is normal text without any font styling**</p>**
9. **<p>**
10. **<font** color="blue"**>**Text with normal size and default face**</font>**
11. **</p>**
12. **<p>**
13. **<font** size="5" color="green"**>**Text with Increased size and default face**</font>**
14. **</p>**
15. **<p>**
16. **<font** color="red" face="cursive"**>**Text with Changed face**</font>**
17. **</p>**
18. **</body>**
19. **</html>**

### **Using CSS**

The same effect can be achieved using CSS properties as in below example:

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Font Tag**</title>**
5. **</head>**
6. **<body>**
7. **<h2>**Change font using CSS**</h2>**
8. **<p>**This is normal text without any font styling**</p>**
9. **<p** style="color: blue;"**>**Text with normal size and default face**</p>**
10. **<p** style="font-size: 25px; color: green;"**>**Text with Increased size and default face **</p>**
11. **<p** style="font-family: cursive; color: red;"**>**Text with Changed face**</p>**
12. **</body>**
13. **</html>**

## Attribute

### **Tag-specific attribute**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| colod | rgb(X,X,X) #xxxxx color\_name | It specifies the color of the content. (Not Supported in HTML5) |
| face | font\_family | It specifies the typeface of the content. (Not Supported in HTML5) |
| size | number | It specifies the size of the content. (Not Supported in HTML5) |

# **HTML <frame> tag (Not supported in HTML5)**

HTML <frame> tag define the particular area within an HTML file where another HTML web page can be displayed.

A <frame> tag is used with <frameset>, and it divides a webpage into multiple sections or frames, and each frame can contain different web pages.

#### **Note: Do not use HTML <frame> tag as it is not supported in HTML5, instead you can use <iframe> or <div> with CSS to achieve similar effects in HTML.**

### **Syntax**

1. **<** **frame** src = "URL" **>**

**Following are some specifications about the HTML <frame> tag**

|  |  |
| --- | --- |
| **Display** | **Block** |
| **Start tag/End tag** | Start tag(required), End tag(forbidden) |
| **Usage** | Frames |

### **Example 1**

#### **Create Vertical frames:**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Frame tag**</title>**
5. **</head>**
6. **<frameset** cols="25%,50%,25%"**>**
7. **<frame** src="frame1.html" **>**
8. **<frame** src="frame2.html"**>**
9. **<frame** src="frame3.html"**>**
10. **</frameset>**
11. **</html>**

**Frame1.html**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. div{
6. background-color: #7fffd4;
7. height: 500px;
8. }
9. **</style>**
10. **</head>**
11. **<body>**
12. **<div>**
13. **<h2>**This is first frame**</h2>**
14. **</div>**
15. **</body>**
16. **</html>**

**Frame2.html**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. div{
6. background-color: #2f4f4f;
7. height: 500px;
9. }
10. **</style>**
11. **</head>**
12. **<body>**
13. **<div>**
14. **<h2>**This is Second frame**</h2>**
15. **</div>**
16. **</body>**
17. **</html>**

**Frame3.html**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. div{
6. background-color: #c1ffc1;
7. height: 500px;
8. }
9. **</style>**
10. **</head>**
11. **<body>**
12. **<div>**
13. **<h2>**This is Third frame**</h2>**
14. **</div>**
15. **</body>**
16. **</html>**

### **Example 2:**

#### **Create Horizontal frames:**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Frame tag**</title>**
5. **</head>**
6. **<frameset** rows="30%, 40%, 30%"**>**
7. **<frame** name="top" src="frame1.html" **>**
8. **<frame** name="main" src="frame2.html"**>**
9. **<frame** name="bottom" src="frame3.html"**>**
10. **</frameset>**
11. **</html>**

## Attribute

### **Tag-specific attribute**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| frameborder | 0 1 | It specifies whether to display a border around the frame or not, and its default value is 1 |
| longdsec | URL | It specifies a page which contains the long description of the content of the frame. |
| marginheight | pixels | It specifies the top and bottom margins of the frame. |
| marginwidth | pixels | It defines the height of the margin between frames. |
| name | text | It is used to assign the name to the frame. |
| noresize | noresize | It is used to prevent resizing of the frame by the user. |
| scrolling | yes no auto | It specifies the existence of the scrollbar for overflowing content. |
| src | URL | It specifies the URL of the document which we want to display in a frame. |

# **HTML <frameset> tag (Not supported in HTML5)**

HTML <frameset> tag is used to contain the group of frames which can be controlled and styled as a unit. The <frameset> element also specifies the number of rows and columns in the frameset, and how much space they will occupy in a frame.

#### **Note: Do not use HTML <frameset> element as it is deprecated and not supported by HTML5, but you can use <iframe> tag instead.**

### **Syntax**

1. **<frameset** cols=" "**>**............**</frameset>**

**Following are some specifications about the HTML <frameset> tag**

|  |  |
| --- | --- |
| **Display** | **Block** |
| **Start tag/End tag** | Both Start and End Tag |
| **Usage** | Frames |

### **Example 1**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Frame tag**</title>**
5. **</head>**
6. **<frameset** cols="50%,50%"**>**
7. **<frame**  src="https://www.iHub.com/html-table"**>**
8. **<frame**  src="https://www.iHub.com/css-table"**>**
9. **</frameset>**
10. **</html>**

## Attribute

### **Tag-specific attribute**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| cols | Pixels % \* | It specifies the number and size of column spaces in the frameset. (Not Supported in HTML5) |
| rows | Pixels % \* | It specifies the number and size of the rows spaces in the frameset. (Not Supported in HTML5) |

# **HTML <isindex> tag (Not supported in HTML5)**

HTML <isindex> tag is used to provide a single line text input in a page to query a document. If user sent input to the server then server returns the list of page matching with the query.

The <isindex> tag can be used anywhere in the document, but it would be preferable to use it within <head> tag.

#### **Note: Do not use <isindex> tag as it is not supported by HTML5, instead of it you can use form <input> which will do the same work.**

### **Syntax**

1. **<isindex** prompt = "Search your document here" **/>**

**Following are some specifications about the HTML <isindex> tag**

|  |  |
| --- | --- |
| **Display** | **None** |
| **Start tag/End tag** | Only Start tag (End tag forbidden) |
| **An element of** | Document Structure |

### **Example**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<isindex** prompt = "Search your document here" **/>**
5. **</head>**
6. **</html>**

## Attribute

### **Tag-specific Attribute**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| prompt | hint-text | It adds value as a label for the search text field. |
| action | URL | It is used to send the query to different URL instead of current document. |

# **HTML Main Tag**

**HTML <main> tag** is used to represent the main content of the <body> tag.

The <main> tag is written within <body> tag. It is used to accurately describe the primary content of a page.

The content of the main tag is directly related to the central topic of the document.

HTML <main> is a new tag and introduced in HTML5.

## Points to remember:

Author should not include more than one <main> tag within a document.

The <main> element should not used as a child of an <article>, <aside>, <header>, <footer>, or <nav> element.

#### **HTML main tag and its sub elements can be easily styled by CSS.**

## HTML main tag example

The <main> tag also supports global attributes and event attributes in HTML 5.

# **HTML <map> tag**

HTML <map> tag is used with <area> tag to define a client-side image map.

An image map is consist of an image with clickable areas, where you can click on the image, and it will open to new or the provided destination.

The <map> tag can consist of more than one <area> elements which define the coordinates and type of the area.

With the help of <map> tag, you can easily link any part of the image to other documents, without dividing the image.

### **Syntax**

1. **<map** name=" "**>**

**Following are some specifications about the HTML <map> tag**

|  |  |
| --- | --- |
| **Display** | **None** |
| **Start tag/End tag** | Empty Tag(Only Start tag) |
| **Usage** | Image map |

### **Example**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**HTML map tag**</title>**
5. **<style>**
6. body{
7. margin-left: 250px;}
8. **</style>**
9. **</head>**
10. **<body>**
11. **<h2>**Example of HTML Map tag**</h2>**
12. **<img** src="image1.png" usemap="#web"**>**
13. **<map** name="web"**>**
14. **<area** shape="rect" coords="66,117,131,168" href="https://www.iHub.com/html-tutorial"**>**
15. **<area** shape="rect" coords="199,36,277,85" href="https://www. iHub.com/css-tutorial"**>**
16. **<area** shape="rect" coords="330,107,406,159" href="https://www.iHub.com/bootstrap-tutorial"**>**
17. **<area** shape="rect" coords="199,185,267,236" href="https://www.iHub.com/javascript-tutorial"**>**
18. **</map>**
19. **</body>**
20. **</html>**

## Attribute:

### **Tag-specific attributes:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| name | mapname | It defines the name of the image-map. |

# **Marquee HTML**

The **Marquee HTML** tag is a non-standard HTML element which is used to scroll a image or text horizontally or vertically.

In simple words, you can say that it scrolls the image or text up, down, left or right automatically.

Marquee tag was first introduced in early versions of Microsoft's Internet Explorer. It is compared with Netscape's blink element.

## Marquee HTML Example

<!DOCTYPE html>

<html>

<body>

<marquee>This is an example of html marquee </marquee>

</body>

</html>

HTML Marquee Attributes

Marquee's element contains several attributes that are used to control and adjust the appearance of the marquee.

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| behavior | It facilitates user to set the behavior of the marquee to one of the three different types: scroll, slide and alternate. |
| direction | defines direction for scrolling content. It may be left, right, up and down. |
| width | defines width of marquee in pixels or %. |
| height | defines height of marquee in pixels or %. |
| hspace | defines horizontal space in pixels around the marquee. |
| vspace | defines vertical space in pixels around the marquee. |
| scrolldelay | defines scroll delay in seconds. |
| scrollamount | defines scroll amount in number. |
| loop | defines loop for marquee content in number. |
| bgcolor | defines background color. It is now *deprecated*. |

## HTML Scroll Marquee

It is a by default property. It is used to scroll the text from right to left, and restarts at the right side of the marquee when it is reached to the end of left side. After the completion of loop text disappears.

<!DOCTYPE html>

<html>

<body>

<marquee width="100%" behavior="scroll" bgcolor="pink">

This is an example of a scroll marquee...

</marquee>

</body>

</html>

## HTML Slide Marquee

In slide marquee, all the contents to be scrolled will slide the entire length of marquee but stops at the end to display the content permanently.

<!DOCTYPE html>

<html>

<body>

<marquee width="100%" behavior="slide" bgcolor="pink">

This is an example of a slide marquee...

</marquee>

</body>

</html>

## HTML Alternate Marquee

It scrolls the text from right to left and goes back left to right.

<!DOCTYPE html>

<html>

<body>

<marquee width="100%" behavior="alternate" bgcolor="pink">

This is an example of a alternate marquee...

</marquee>

</body>

</html>

## Direction in HTML marquee

This is used to change the direction of scrolling text. Let's take an example of marquee scrolling to the right. The direction can be left, right, up and down.

<!DOCTYPE html>

<html>

<body>

<marquee width="100%" direction="right">

This is an example of a right direction marquee...

</marquee>

</body>

</html>

## Nested marquee example

<!DOCTYPE html>

<html>

<body>

<marquee width="400px" height="100px" behavior="alternate" style="border:2px solid red">

<marquee behavior="alternate">

Nested marquee...

</marquee>

</marquee>

</body>

</html>

## Disadvantages HTML marquee

1) Marquee may be distracting because human eyes are attracted towards movement and marquee text constantly.

2) Since Marquee text moves, so it is more difficult to click static text, depending on the scrolling speed.

3) It is a non-standard HTML element.

4) It draws user's attention needlessly and makes the text harder to read.

# **HTML <menu> tag**

HTML <menu> tag specifies a list or menu of commands that a user can perform or activate. It is used for creating context menu as well as lists menu.

A <menu> element can contain one or more <li> or <menuitem> elements within it.

#### **Note: The <menu> tag was deprecated in HTML 4.01 and again included in HTML 5.1 specification. But it will better to ignore it as it is experimental and not supported by many browsers.**

### **Syntax**

1. **<menu>**........**<menu>**

**Following are some specifications about the HTML <menu> tag**

|  |  |
| --- | --- |
| **Display** | **Inline** |
| **Start tag/End tag** | Both Start and End tag |
| **Usage** | Semantic and textual |

### **Example**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Menu Tag**</title>**
5. **</head>**
6. **<body>**
7. **<h2>**Example of Menu Tag**</h2>**
8. **<menu>**
9. **<li>**Home**</li>**
10. **<li>**Registration**</li>**
11. **<li>**Contact-us**</li>**
12. **<li>**About-us**</li>**
13. **</menu>**
14. **</body>**
15. **</html>**

## Attribute:

### **Tag-specific attributes:**

* popup
* toolbar
* context

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| labelHTML Tags List | text | It specifies the label for the menu. |
| typeHTML Tags List | It specifies the type of the menu in an HTML document. |  |

# **HTML Meter Tag**

**HTML <meter> tag** is used to measure data within a given range. It defines a scalar measurement with range. It is also known as a gause.

It should be used to display disk usage, voting population etc.

The HTML meter tag is new in HTML5 so you must use new browsers.

## Attributes of HTML Meter Tag

HTML <meter> tag supports the global and event attributes as well as some specific attributes.

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| value | It is a mandatory attribute which is used to specify a value in numbers. The number may be integer or floating point number. |
| form | It specifies one or more forms to which meter element belongs to. |
| high | It specifies a range that is considered as high value. |
| low | It specifies a range that is considered as low value. |
| max | It specifies the maximum value defined in the range. |
| min | It specifies the minimum value defined in the range. |
| optimum | It specifies the optimum value for the gauge.It is an optional attribute. |

## HTML Meter Tag Example

Let's see the progress example with min, max and value attributes.

<!DOCTYPE html>

<html>

<body>

<p>Display a gauge:</p>

<meter value="30" min="1" max="100">30 out of 100</meter><br>

<meter value="0.8">80%</meter>

</body>

</html>

## Styling Meter

You can apply CSS code on meter tag.

1. meter{
2. width: 300px;
3. height: 30px;
4. }

# **HTML <nav> tag**

HTML <nav> tag is used to represent a section which contains navigation links, either within current document or to another document. Examples of some navigation links are menu, table of contents, indexes, etc.

The <nav> tag is newly added tag in HTML5.

#### **Tips: You can use multiple links within a page, but it's not necessary to put all the links withintag. It must contain only major block of navigation links.**

### **Syntax**

1. **<nav>**.....**</nav>**

**Following are some specifications about the HTML <nav> tag**

|  |  |
| --- | --- |
| **Display** | **Block** |
| **Start tag/End tag** | Both Start and End tag |
| **Usage** | Structural |

### **Example**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Nav Tag**</title>**
5. **<style>**
6. a{
7. text-decoration: none;
8. color: green;
9. font-size: 20px;
10. padding: 8px;
11. }
12. a:hover{
13. color: blue;
14. }
15. h1{
16. text-align: center;
17. }
18. **</style>**
19. **</head>**
20. **<body>**
21. **<h1>**Example of Nav tag**</h1>**
22. **<nav>**
23. **<a** href="#"**>**Home**</a>** |
24. **<a** href="#"**>**Courses**</a>** |
25. **<a** href="#"**>**About-us**</a>** |
26. **<a** href="#"**>**Contact-us**</a>** |
27. **</nav>**
28. **</body>**
29. **</html>**

# **HTML <noframes> tag (Not supported in HTML5)**

HTML <noframes> tag is used to contain an alternative text to display if the browser does not support the <frame> content. It will only work, if the browser does not support the frame else, it will be ignored by the browser.

The <noframes> tag should be used within <frameset> element.

#### **Note: Do not use <noframes> as it is deprecated and currently not supported by HTML5.**

### **Syntax**

1. **<noframes>**........**</noframes>**

**Following are some specifications about the HTML <noframes> tag**

|  |  |
| --- | --- |
| **Display** | **Block** |
| **Start tag/End tag** | Both Start and End tag |
| **Usage** | With Frames |

### **Example**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Noframe tag**</title>**
5. **</head>**
6. **<frameset** cols="50%,50%"**>**
7. **<frame** src="https://www. iHub.com/html-table"**>**
8. **<frame** src="https://www. iHub.com/css-table"**>**
9. **<noframes>**Sorry! Your browser does not support frames. **</noframes>**
10. **</frameset>**
11. **</html>**

# **HTML <object> tag**

HTML <object> tag is used to embed multimedia files on webpage. The <object> tag can include multimedia files such as video, audio, image, PDF, Java Applets, or another page on your page.

HTML <param> tag also used with <object> tag to pass parameters to plugin which has been included with <object> tag.

If you insert text between the <object> and </object> tags, then it will only be displayed if the browser does not support the <object> tag.

### **Syntax**

1. **<object** data="" type=""**></object>**

**Following are some specifications about the HTML <object > tag**

|  |  |
| --- | --- |
| **Display** | **Inline** |
| **Start tag/End tag** | Both Start and End tag |
| **Usage** | Embed external resource |

### **Example**

**Embedding a Video:**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Obejct Tag**</title>**
5. **</head>**
6. **<body>**
7. **<h2>**Example of Object tag**</h2>**
8. **<object** height="250" width="500" data="https://www.youtube.com "**></object>**
9. **</body>**
10. **</html>**

**Embedding a gif image:**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Obejct Tag**</title>**
5. **</head>**
6. **<body>**
7. **<h2>**Example of Object tag**</h2>**
8. **<object** height="250" width="500" data="circle.gif"**>**Sorry!,Your browser does not support**</object>**
9. **</body>**
10. **</html>**

## Attribute:

### **Tag-specific attributes:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| data | URL | It specifies the address of the resource. |
| type | content\_type | It determines the content type of the resource specified by data. |
| archive | URL | It specifies the space-separated list of URL's for archives of resources for the object. **(Not supported in HTML5)**. |
| border | pixels | It sets the width of the border around the <object> **(Not supported in HTML5)**. |
| classid | URL | It specifies the URL for the object implementation. **(Not supported in HTML5)**. |
| codebase | URL | It specifies the base path where to find the code for the object. **(Not supported in HTML5)**. |
| form | form\_id It | specifies the form element that the object element is associated with. |
| height | pixels | It defines the height of the object. |
| width | pixels | It determines the width of the object. |
| typemustmatch | boolean | It specifies that resource should be embedded if type attribute matches with the actual content type of the resources provided on the data attribute. |
| name | name | It defines the name for the object. |

HTML <optgroup> tag is used to group related <options> in a drop down list within <select> element.

Using <optgroup> tag with <select> makes easier to access the dropdown list especially if list has large number of options.

### **Syntax**

1. **<optgroup** label=" "**>**........**</optgroup>**

**Following are some specifications about the HTML <optgroup> tag**

|  |  |
| --- | --- |
| **Display** | **Inline** |
| **Start tag/End tag** | Both Start and End tag |
| **Usage** | Forms and Inputs |

### **Example**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Optgroup Tag**</title>**
5. **<style>**
6. body{
7. text-align: center;
8. }
9. select{
10. font-size:20px;
11. font-weight:bold;
12. color:green;
13. }
14. **</style>**
15. **</head>**
16. **<body>**
17. **<h2>**Example of optgroup tag**</h2>**
18. **<label>**Select your favourite brand**</label><br>**
19. **<select>**
20. **<optgroup** label="Laptop Maufacturur"**>**
21. **<option** value="dell"**>**Dell**</option>**
22. **<option** value="hp"**>**HP**</option>**
23. **<option** value="lenovo"**>**Lenovo**</option>**
24. **<option** value="acer"**>**Acer**</option>**
25. **</optgroup>**
26. **<optgroup** label="Mobile Manufacturer"**>**
27. **<option** value="apple"**>**Apple**</option>**
28. **<option** value="nokia"**>**Nokia**</option>**
29. **<option** value="samsung"**>**Samsung**</option>**
30. **<option** value="coolpad"**>**Coolpad**</option>**
31. **</optgroup>**
32. **</select>**
33. **</body>**
34. **</html>**

## Attribute:

### **Tag-specific attributes:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| disabled | disabled | If it is set then options of that optgroup will be disabled. |
| label | text | It defines the label for the group which will be displayed in the drop-down list. It is required attribute. |

# **HTML <output> tag**

HTML <output> tag is used to display the result of some calculation (performed by JavaScript) or the outcome of a user action (such as Input data into a form element).

The <output> tag is a newly added tag and was introduced in HTML5.

### **Syntax**

1. **<output>**......**</output>**

**Following are some specifications about the HTML <output> tag**

|  |  |
| --- | --- |
| **Display** | **Inline** |
| **Start tag/End tag** | Both Start and End tag |
| **Usage** | Forms and Input |

### **Example**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Output Tag**</title>**
5. **</head>**
6. **<body>**
7. **<p>**Calculate the Sum of the two Numbers**</p>**
8. **<form** oninput="res.value=parseInt(a.value)+parseInt(b.value);"**>**
9. **<label>**Enter First Value.**</label><br>**
10. **<input** type="number" name="a" value=""**/><br>**
11. +**<br/>**
12. **<label>**Enter First Value.**</label><br>**
13. **<input** type="number" name="b" value=""**><br>**
14. =**<br>**
15. Output is:**<output** name="res"**></output>**
16. **</form>**
17. **</body>**
18. **</html>**

## Attribute:

### **Tag-specific attributes:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| for | element\_id | It specifies the list of other element's ids which indicates the relationship between the result of the calculation and the input elements. |
| form | form\_id | It specifies the form element to which this element is associated with. |
| name | name | It defines the name for the output element. |

# **HTML <param> tag**

HTML <param> tag is used to pass the parameters to the object that has been embedded using <object> element.

We can use more than one <param> tag within an <object> element in any order, but each tag must contain name and value attribute and should be placed at the start of the content.

The <param> tag controls the behaviour of the <object> element using a different pair of the name of value attributes, such as autoplay, controller, etc.

### **Syntax**

1. **<param** name=" " value=""**>**

**Following are some specifications about the HTML <param> tag**

|  |  |
| --- | --- |
| **Display** | **Inline** |
| **Start tag/End tag** | Empty Tag(End tag forbidden) |
| **Usage** | Programming |

### **Example**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Param Tag**</title>**
5. **</head>**
6. **<body>**
7. **<h2>**Example of Param Tag**</h2>**
8. **<object** data="https://www.youtube.com/ " **>**
9. **<param** name="controller" value="true"**>**
10. **</object>**
11. **</body>**
12. **</html>**

## Attribute:

### **Tag-specific attributes:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| Name | text | It determines the name of the parameter. |
| value | text | It determines the value of the parameter. |
| type | content\_type | It specifies the media type of the parameter, and it only used if the valuetype is set to "ref." (Not Supported in HTML5) |
| valuetype | * data * ref * object | It determines the type of the value attribute. (Not Supported in HTML5) |

# **HTML <picture> tag**

HTML <picture> tag is used in responsive web designing where we need to load the different images based on their viewport, height, width, orientation, and pixel density.

The <picture> tag contains one or more <source> elements and one <img> elements.

According to the viewport, the matching image will be loaded from different <source> tag, and if no source contains the matching image, then the default image present in <img> tag will be displayed on the browser.

This tag is a new tag in HTML5.

### **Syntax**

1. **<picture>**
2. **<source** srcset="" media=""**>**
3. **<img** src=""**>**
4. **</picture>**

**Following are some specifications about the HTML <picture> tag**

|  |  |
| --- | --- |
| **Display** | **Inline** |
| **Start tag/End tag** | Both Start and End tag |
| **Usage** | Image |

### **Example**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Picture Tag**</title>**
5. **<style>**
6. body{
7. text-align: center;
8. }
9. p{
10. font-size: bold;
11. font-size: 20px;
12. color: green;
13. }
14. **</style>**
15. **</head>**
16. **<body>**
17. **<h2>**Example of picture tag**</h2>**
18. **<p>**Resize the page to see the different versions of the images at different viewports, and as per viewport image will be automatically changed.**</p>**
19. **<picture>**
20. **<source** srcset="download1.jpg" media="(min-width: 750px)"**>**
21. **<source** srcset="pic2.jpg" media="(min-width: 450px)"**>**
22. **<img** srcset="rose.jpg" alt="default image" style="width: auto;"**>**
23. **</picture>**
24. **</body>**
25. **</html>**

## Attribute:

### **Tag-specific attributes:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| media | media\_query | It defines and accept any media query which can be defined in CSS. |
| srcsetHTML Tags List | URL | It defines the URL of the image which can be used for different situations. (Required) |
| type | video/ogg video/mp4 video/webM audio/ogg audio/mpeg | It determines the MIME type |
| src | URL | It specifies the location of the image. |

# **HTML Progress Tag**

**HTML <progress> tag** is used to display the progress of a task. It provides an easy way for web developers to create progress bar on the website. It is mostly used to show the progress of a file uploading on the web page.

The HTML progress tag is new in HTML5 so you must use new browsers.

## Attributes of HTML Progress Tag

HTML <progress> tag supports the global and event attributes as well as 2 specific attributes.

|  |  |
| --- | --- |
| **Tag** | **Description** |
| value | It defines that how much work the task has been completed. |
| max | It defines that how much work the task requires in total. |

#### **The progress tag should be used to represent progress of a task only, not for just a gauge (disk pace usage). For such purpose, <meter> element is used.**

## HTML Progress Tag Example

Let's see HTML progress example without attribute.

<!DOCTYPE>

<html>

<body>

<progress></progress>

</body>

</html>

Let's see the progress example with value and max attributes.

<!DOCTYPE>

<html>

<body>

Downloading progress:

<progress value="43" max="100"></progress>

</body>

</html>

## Styling Progress Bar

You can apply CSS code on progress tag.

1. progress{
2. width: 300px;
3. height: 30px;
4. }

## HTML Progress Tag with JavaScript

The <progress> tag should be used in conjunction with JavaScript to display the progress of a task.

1. **<script>**
2. var gvalue=1;
3. function abc(){
4. var progressExample = document.getElementById ('progress-javascript-example');
5. setInterval (function ()
6. {
7. if(gvalue**<100**){
8. gvalue++;
9. progressExample.value =gvalue;
10. }
11. abc();
12. }, 1000);
13. }
14. **</script>**
15. **<progress** id="progress-javascript-example" min="1" max="100"**></progress>**
16. **<br/><br/>**
17. **<button** onclick="abc()"**>**click me**</button>**

# **HTML <rp> tag**

HTML <rp> tag is used to provide fall back parenthesis for the content which is to be shown in the browser, if browser does not support display of ruby annotations.

The ruby annotations are used to display certain characters in several Asian languages, using <ruby> element.

The <rp> tag must enclose the <rt> element with parenthesis which contains the annotation's text.

The <rp> tag is new in HTML5.

### **Syntax**

1. **<rp>**.....**</rp>**

**Following are some specifications about the HTML <rp> tag**

|  |  |
| --- | --- |
| **Display** | **Inline** |
| **Start tag/End tag** | Both Start and End tag |
| **Usage** | Textual |

### **Example**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**HTML rp tag**</title>**
5. **<style>**
6. body{
7. text-align: center;
8. }
9. rt{ font-size: 18px;
10. color: green;
11. }
12. **</style>**
13. **</head>**
14. **<body>**
15. **<h1>**Example of rp tag**</h1>**
16. **<ruby>**
17. 漢**<rp>**(**</rp><rt>**Kan**</rt><rp>**)**</rp>**
18. 字 **<rp>**(**</rp><rt>**ji**</rt><rp>**)**</rp>**
19. **</ruby>**
20. **</body>**
21. **</html>**

# **HTML <rt> tag**

HTML <rt> tag provides translation, pronunciations, or transliteration information for the east Asian characters present in ruby language annotation.

The <rt> tag must be enclosed within <ruby> and <rp> (optional) element.

The <rt> tag is new in HTML5.

### **Syntax**

1. **<rt>**........ **</rt>**

Following are some specifications about the HTML <rt> tag

|  |  |
| --- | --- |
| **Display** | **Inline** |
| **Start tag/End tag** | Both Start and End tag |
| **Usage** | Formatting |

### **Example**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**HTML rt tag**</title>**
5. **<style>**
6. body{
7. text-align: center;
8. }
9. rt{ font-size: 10px;
10. color: green;
11. }
12. **</style>**
13. **</head>**
14. **<body>**
15. **<h1>**Example of rt tag**</h1>**
16. **<ruby>**
17. 大哥**<rt>**Big Brother**</rt>**
18. **</ruby>**
19. **</body>**
20. **</html>**

# **HTML <ruby> tag**

HTML <ruby> tag is used to represent ruby annotations on the web page. Ruby annotations are useful if we want to show the pronunciation of East Asian characters like Chinese and Japanese Chinese.

The <ruby> tag enclosed one or more <rt> elements which give the pronunciation of ruby annotations and displays above the annotation characters. It can also contain <rp> (optional) element which is used as a fall-back parenthesis for the browser which does not support the ruby annotation.

HTML <ruby> tag can also be used to represent small annotation which is relevant to the main content, apart from East Asian language.

The <ruby> tag is new in HTML5.

### **Syntax**

1. **<ruby>**......**</ruby>**

**Following are some specifications about the HTML <ruby> tag**

|  |  |
| --- | --- |
| **Display** | **Inline** |
| **Start tag/End tag** | Both Start and End tag |
| **Usage** | formatting |

### **Example**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**HTML ruby tag**</title>**
5. **<style>**
6. body{
7. text-align: center;
8. }
9. rt{ font-size: 10px;
10. color: green;
11. }
12. **</style>**
13. **</head>**
14. **<body>**
15. **<h1>**Example of ruby tag**</h1>**
16. **<p>**East Asian Langugae Annotation**</p>**
17. **<ruby>**
18. 大哥**<rt>**Big Brother**</rt>**
19. **</ruby>**
21. **<p>**Noraml Annotaion Representation of Expiry date**</p>**
22. **<ruby>**
23. 2022**<rp>**(**</rp><rt>**Year**</rt><rp>**)**</rp>**
24. 12**<rp>**(**</rp><rt>** Month**</rt><rp>**)**</rp>**
25. 06**<rp>**(**</rp><rt>**Date**</rt><rp>**)**</rp>**
26. **</ruby>**
27. **</body>**
28. **</html>**

# **HTML <s> tag**

HTML <s> tag is used to represent the range of content which is no longer accurate or relevant in some way. The text written between <s> (start) and </s> (end) tag, renders as a strike a line through the text.

#### **Note: Do not confuse with <del> and <s> as <del> is used for the text which is deleted or removed from document and <s> represent the text which is no longer accurate.**

### **Syntax**

1. **<s>** Write your content here.....**</s>**

**Following are some specifications about the HTML <s> tag**

|  |  |
| --- | --- |
| **Display** | **Inline** |
| **Start tag/End tag** | Both Start and End tag |
| **Usage** | Formatting |

### **Example**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**HTML s tag**</title>**
5. **<style>**
6. body{
7. text-align: center;}
8. s{
9. color: green;
10. }
11. **</style>**
12. **</head>**
13. **<body>**
14. **<h2>**Example of s tag**</h2>**
15. **<p><s>**The last date of apply for online classes is 12/02/2019**</s></p>**
16. **<p>**The last date of apply for online classes is 13/03/2019**</p>**
17. **</body>**
18. **</html>**

# **HTML <samp> tag**

HTML <samp> tag is a phrase tag which is used to represent the sample output from a computer program or a script, in the browser.

The content written between the <samp> and </samp>, tags renders in a browser?s default monospace font.

#### **Note: The <samp> tag is not deprecated yet, but you can also use CSS property to achieve the same result.**

### **Syntax**

1. **<samp>**Write sample output here.....**</samp>**

**Following are some specifications about the HTML <samp> tag**

|  |  |
| --- | --- |
| **Display** | **Inline** |
| **Start tag/End tag** | Both Start and End tag |
| **Usage** | textual |

### **Example**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**HTML samp tag**</title>**
5. **<style>**
6. body{
7. text-align: center;
8. }
9. **</style>**
10. **</head>**
11. **<body>**
12. **<h2>**Example of samp tag**</h2>**
13. **<p>**I am trying to install a software on my system but it is continuously giving an error message:
14. **</p>**
15. **<p><samp>**Error 5: Access is denied.**</samp></p>**
16. **</body>**
17. **</html>**

# **HTML Section Tag**

The HTML <section> tag is used to define sections in a document. When you put your content on a web page, it may contains many chapters, headers, footers, or other sections on a web page that is why HTML <section> tag is used.

HTML <section> is a new tag introduced in HTML5.

## HTML section tag example

**CSS code:**

1. section{
2. border:1px solid pink;
3. padding:15px;
4. margin:10px;
5. }

**HTML code:**

<!DOCTYPE>

<html>

<head>

<style>

section{

border:1px solid pink;

padding:15px;

margin:10px;

}

</style>

</head>

<body>

<h2> Indian Leader</h2>

<section>

<h3> Jawaharlal Nehru </h3>

<p> Jawaharlal Nehru was the first Prime Minister of India and a central figure in

Indian politics for much of the 20th century. He emerged as the paramount leader of the Indian

independence movement under the tutelage of Mahatma Gandhi. -Source Wikipedia </p>

</section>

<section>

<h3>Subhas Chandra Bose </h3>

<p>Subhas Chandra Bose was an Indian nationalist whose attempt during World War II to rid India of

British rule with the help of Nazi Germany and Japan left a troubled legacy.

The honorific Netaji (Hindustani language: "Respected Leader"), first applied to Bose in Germany,

by the Indian soldiers of the Indische Legion and by the German and Indian officials

in the Special Bureau for India in Berlin, in early 1942, is now used widely throughout India. -source Wikipedia</p>

</section>

</body>

</html>

# **HTML <small> tag**

HTML <small> tag makes text font by one size smaller than the document?s base font size (Such as large to medium, medium to small, etc.)

In HTML5, <small> tag is used for identifying secondary importance such as copyright, side comments, and legal notices.

#### **Tips: The <small> tag can be nested which means we can use <small> tag inside one another for multiple times, and it will continue to decrease the font size than its surrounding text.**

### **Syntax**

1. **<small>**Write your content here...... **<small>**

**Following are some specifications about the HTML <small> tag**

|  |  |
| --- | --- |
| **Display** | **Inline** |
| **Start tag/End tag** | Both Start and End tag |
| **Usage** | textual |

### **Example**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**HTML small tag**</title>**
5. **</head>**
6. **<body>**
7. **<h2>**Example of small tag**</h2>**
8. **<p** style="color: green;"**>**This is normal font size.....
9. **<small** style="color: blue;"**>**It is smaller than previous...
10. **<small** style="color: red;"**>**It is smallest.**</small>**
11. **</small>**
12. **</p>**
13. **</body>**
14. **</html>**

# **HTML <source > tag**

HTML <source> tag is used as a child element to define more than one media resources for <audio>, <video>, and <picture> element.

It is used to provide the same media content with different formats such as mp3, mp4, etc.

When we embed multiple resources with the same content but different format then the browser may choose the most compatible format and display or play that media file.

The <source> tag was introduced in HTML5.

### **Syntax**

1. **<source** src=" " type=" "**>**

**Following are some specifications about the HTML <source> tag**

|  |  |
| --- | --- |
| **Display** | **Inline** |
| **Start tag/End tag** | Empty tag( Only start tag) |
| **Usage** | Media resource |

### **Example**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**HTML source tag**</title>**
5. **</head>**
6. **<body>**
7. **<h2>**Example of source tag**</h2>**
8. **<video** controls="controls" height="200" width="300"**>**
9. **<source** src="flower.webm" type="video/webm" **>**
10. **<source** src="flower.mp4" type="video/mp4"**>**
11. Your browser does not support the HTML5 video element.
12. **</video>**
13. **</body>**
14. **</html>**

## Attribute:

### **Tag-specific attributes:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| media | Media\_query | It determines that for which media/device the linked source is optimized. |
| sizes |  | It specifies acceptable image sizes for different page layout. |
| src |  | It determines the URL of media files. |
| srcset |  | Specifies the URL of the image for different situations. It is only used when <source> is child of <picture> element. |
| type | video/ogg video/mp4 video/webm audio/ogg audio/mpeg | It determines the media type of resource. |

# **HTML SVG**

The **HTML SVG** is an acronym which stands for Scalable Vector Graphics.

HTML SVG is a modularized language which is used to describe graphics in XML. It describe two-dimensional vector and mixed vector/raster graphics in XML. It is a W3C recommendation. SVG images and their behaviors are defined in XML text files. So as XML files, you can create and edit an SVG image with text editor, but generally drawing programs like inkspace are preferred to create it.

SVG is mostly used for vector type diagrams like pie charts, 2-Dimensional graphs in an X,Y coordinate system etc.

The <svg> element specifies the root of a SVG fragment. You can animate every element and every attribute in SVG files.

## HTML SVG Circle Example

Let's see the example to draw circle by svg tag.

<!DOCTYPE html>

<html>

<body>

<svg width="100" height="100">

<circle cx="50" cy="50" r="40" stroke="yellow" stroke-width="4" fill="red" />

</svg>

</body>

</html>

Here, we are using cx, cy and r attributes of circle tag. These attributes can't be used with svg rect tag.

## HTML SVG Rectangle Example

Let's see the example to draw rectangle by svg tag.

<!DOCTYPE html>

<html>

<body>

<svg width="200" height="100">

<rect width="200" height="100" stroke="yellow" stroke-width="4" fill="red" />

</svg>

</body>

</html>

## HTML SVG polygon Example

Let's see the example to draw polygon by svg tag.

<!DOCTYPE html>

<html>

<body>

<svg height="210" width="500">

<polygon points="100,10 40,198 190,78 10,78 160,198"

style="fill:red;stroke:yellow;stroke-width:5;fill-rule:nonzero;" />

</svg>

</body>

</html>

## Why SVG is preferred over other image formats?

SVG images can be saved as the smallest size possible. Unlike bitmap image formats like JPG or PNG, it does not contain a fixed set of dots. So it is also easy to print with high quality at any resolution.

SVG images can be zoomed to a certain level without degradation of the picture quality.

SVG images and their behaviors are defined in XML text files, so they can be created and edited with any text editor.

# HTML <tbody> tag

HTML <tbody> tag is used to group the table rows (<tr>) together, which indicates that this is body part of a table (<table>).

The <tbody> tag must be a child of <table> element.

The <tbody> is used along with <thead> and <tfoot> which shows the different part of the table that are table head, table body, and table footer, however, it does not affect the layout of the table.

These elements can be used for providing semantic information which can be helpful in accessibility purpose, or rendering the header at top and footer at the bottom while printing a large table.

#### The <tbody> tag must contain one or more <tr> elements.

### Syntax

**<tbody>**............**</tbody>**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**HTML tbody tag**</title>**
5. **<style>**
6. body{
7. margin-left: 195px;"
8. }
9. **</style>**
10. **</head>**
11. **<body>**
12. **<h2>**Example of the tbody tag**</h2>**
13. **<table** border="1" bgcolor="#98f5ff"**>**
14. **<thead>**
15. **<tr>**
16. **<th>**EmpId**</th>**
17. **<th>**Name**</th>**
18. **<th>**Email-Id**</th>**
19. **</tr>**
20. **</thead>**
21. **<tbody>**
22. **<tr>**
23. **<td>**121**</td>**
24. **<td>**John**</td>**
25. **<td>**john123@gmail.com**</td>**
26. **</tr>**
28. **<tr>**
29. **<td>**122**</td>**
30. **<td>**William **</td>**
31. **<td>**william56@gmail.com**</td>**
32. **</tr>**
34. **<tr>**
35. **<td>**123**</td>**
36. **<td>**Amit**</td>**
37. **<td>**amitk98@gmail.com**</td>**
38. **</tr>**
39. **</tbody>**
40. **</table>**
41. **</body>**
42. **</html>**

## Attribute:

### Tag-specific attributes:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute** | | **Value** | | **Description** | |
| align | right left center justify char | | It determines the alignment of the content inside the <tbody> element. | |
| char | character | | It specifies the alignment of the <tbody> content to the character. | |
| charoff | Number | | It specifies the number of characters the content will be aligned from the character specified by the char attribute. | |
| valign | top middle bottom baseline | | It determines the vertical alignment of the content inside the <tbody> element. | |

# HTML <td> tag

HTML <td> tag is used to specify the cells of an HTML table which contain data of the table. The <td> tag must be the child element of <tr> (table row) tag. Each table row can contain multiple <td> data elements.

The grouped <td> elements of a <tr> tag renders as a single row in the table. The content of the <td> elements is regular and left-aligned in the table by default.

### Syntax

**<td>**.......**</td>**

**Note**: Please refer to the above example.

## Attribute:Tag-specific attributes:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute** | | **Value** | | **Description** | |
| abbr | text | | It defines the abbreviated version of content of the cell. **(Not Supported in HTML5)** | |
| align | left right center justify char | | It specifies the alignment of the content of the cell. **(Not Supported in HTML5)** | |
| axis | category\_name | | It Categorizes Cells. . **(Not Supported in HTML5)** | |
| bgcolor | rgb(x,x,x) #xxxxxx Color\_name | | It sets the background color of the cell. **(Not Supported in HTML5)** | |
| char | character | | It specifies the alignment of the content of cell to the character. **(Not Supported in HTML5)** | |
| charoff | number | | It determines the number of characters the content aligned from the character specified by the char attribute. **(Not Supported in HTML5)** | |
| colspan | number | | It determines the number of columns a cell should span. | |
| headers | header\_id | | It determines one or more header cells to which a cell is related. | |
| height | % pixels | | It determines the height of a table cell. **(Not Supported in HTML5)** | |
| nowrap | nowrap | | If it sets then content inside the cell should not wrap. **(Not Supported in HTML5)** | |
| rowspan | number | | It determines the number of rows a cell should span. | |
| scope | col colgroup row rowgroup | | It specifies the cells that the header element relates to. **(Not Supported in HTML5)** | |
| valign | top middle bottom baseline | | It determines the vertical alignment of the cell content. **(Not Supported in HTML5)** | |
| width | % pixels | | It determines the width of the cell.**(Not Supported in HTML5)** | |

# **HTML <template> tag**

HTML <template> tag is used to hold the client-side content that will not render at the time of page load, but it can be instantiated during runtime using JavaScript.

The content of the template will not be displayed until it is not activated using JavaScript. The browser processes the content of the <template> element while loading the page to ensure that the content is valid, the contents are not rendered, however.

It can also be useful when you want to use same content multiple times in your HTML document without any change.

The <template> tag can be placed anywhere inside of <head>, <body>, <frameset>, or <table> elements.

The <template> tag is newly added element in HTML5.

### **Syntax**

**<template>**.........**</template>**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**HTML Template tag**</title>**
5. **<style>**
6. body{
7. background-color: #e6e6fa;
8. }
9. **</style>**
10. **</head>**
11. **<body>**
12. **<h2>**Example of template tag**</h2>**
13. **<button** onclick="clickMe()"**>**Click Me**</button><br>**
15. **<template** id="mytemplate"**>**
16. **<img** src="bird.jpg" alt="bird's image" height="100" width="100"**>**
17. **<script>**
18. alert("Thank you for choosing template. Click OK for image.")
19. **</script>**
20. **</template>**
22. **<script>**
23. function clickMe() {
24. var x= document.getElementsByTagName("template")[0];
25. var clon = x.content.cloneNode(true);
26. document.body.appendChild(clon);}
27. **</script>**
28. **</body>**
29. **</html>**

## Attribute:

### **Tag-specific attributes:**

The <template> tag does not contain any specific attribute.

### **Global attribute:**

The <template> tag supports the Global attributes in HTML.

# **HTML Textarea**

The **HTML <textarea> tag** is used to define a multi-line text input control.

It can hold unlimited number of characters and the texts are displayed in a fixed-width font (usually courier).

The size of the HTML textarea is defined by <cols> and <rows> attribute, or it can also be defined through CSS height and width properties.

## HTML Textarea Example

<!DOCTYPE>

<html>

<body>

<textarea rows="9" cols="70">

Textarea tag example with rows and columns.

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| autofocus | It specifies that a text area should be automatically get focused when the page is loaded. |
| form | It specifies one or more forms the textarea belongs to. |
| maxlength | It specifies the maximum number of characters allowed in the text area. |
| placeholder | It specifies a short hint that describes the expected value of a textarea. |
| required | It specifies that textarea must be filled out. |
| wrap | It specifies that how the texts in the textarea are wrapped at the time of the submission of the form. |

## HTML Textarea form attribute

The form attribute specifies one or more forms the text area belongs to.

1. **<form** action="updates.jsp" id="usrform"**>**
2. Name: **<input** type="text" name="usrname"**>**
3. **<input** type="submit"**>**
4. **</form>**
5. **<br>**
6. **<textarea** rows="9" cols="70" name="comment" form="usrform"**>**
7. Enter text here...**</textarea>**
8. **<p>**The text area above is outside the form element, but should still be a part of the form.**</p>**
9. **<p><b>**Note:**</b>** The form attribute is not supported in Internet Explorer.**</p>**

# **HTML <tfoot> tag**

HTML <tfoot> tag is used to define the set of rows which represents footer of an HTML table. The <tfoot> tag must contain one or more <tr> element.

The <tfoot> tag is used as a child element of HTML table (<table>) along with <thead> and <tbody> elements, where <thead> defines table header and <tbody> defines the table body.

#### **Tips**: The <thead>, <tbody>, and <tfoot> elements do not affect the table layout, and if you want to apply the change in table layout then use CSS properties.

### **Syntax**

**<tfoot>**

**<tr></tr>**

**<tr></tr>**

</tfoot**>**

Example:

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**HTML tfoot Tag**</title>**
5. **<style>**
6. table{
7. border-collapse: collapse;
8. }
9. thead,tfoot{
10. background-color:#3f87a6;
11. }
12. tbody{
13. background-color:#97ffff;
14. }
15. **</style>**
16. **</head>**
17. **<body>**
18. **<h1>**Example of tfoot tag**</h1>**
19. **<table** border="1" **>**
20. **<thead>**
21. **<tr>**
22. **<th>**Items**</th>**
23. **<th>**Quantity**</th>**
24. **<th>**Expenditure**</th>**
25. **</tr>**
26. **</thead>**
27. **<tfoot>**
28. **<tr>**
29. **<th>**Total**</th>**
30. **<th>**90**</th>**
31. **<th>**4175**</th>**
32. **</tr>**
33. **</tfoot>**
34. **<tbody>**
35. **<tr>**
36. **<td>**Books**</td>**
37. **<td>**5**</td>**
38. **<td>**1500**</td>**
39. **</tr>**
40. **<tr>**
41. **<td>**Drawing-Paper**</td>**
42. **<td>**50**</td>**
43. **<td>**800**</td>**
44. **</tr>**
45. **<tr>**
46. **<td>**Marker**</td>**
47. **<td>**35**</td>**
48. **<td>**1875**</td>**
49. **</tr>**
50. **</tbody>**
51. **</table>**
52. **</body>**
53. **</html>**

## Attribute:

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| align | right left center justify char | It determines the alignment of the content inside the <tfoot> element. |
| char | Character | It specifies the alignment of the content inside the <tfoot> element to the character. |
| charoff | Number | It specifies the number of characters the content will be aligned from the character specified by the char attribute. |
| valign | top middle bottom baseline | It determines the vertical alignment of the content inside the <tfoot> element. |

# **HTML <th> tag**

In an HTML table there are two types of cells:

* **Header cell** - It contains the header information (created using <th> element)
* **Data Cells** - It contains the main data of the table (created using <td> element).

HTML <th> tag is used to define the header cells of an HTML table. The header cell renders as bold and centered by default on the browser, but you can change its default style using CSS properties.

The <th> tag must be used as a child element of the <tr> element within <table> element. The size of the table is auto-adjustable as per the content size.

### **Syntax**

**<th>**Content....... **</th>**

Refer to the above Examples.

## Attribute:

### **Tag-specific attributes:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| abbr | text | It defines the abbreviated version of content of the header cell. |
| align | left right center justify char | It specifies the alignment of the content of the header cell. |
| axis | category\_name | It Categorizes header Cells |
| bgcolor | rgb(x,x,x) #xxxxxx Color\_name | It sets the background color of the header cell |
| char | character | It specifies the alignment of the content of the header cell to the character |
| charoff | number | It specifies the number of characters the header cell content will be aligned from the character specified by the char attribute. |
| colspan | number | It determines the number of columns a header cell should span. |
| headers | header\_id | It determines a space-separated list of header cells which  contains information of the cells is related. |
| height | % pixels | It determines the height of a table header cell. |
| nowrap | nowrap | If it sets then content inside the header cell should not wrap. |
| rowspan | number | It determines the number of rows a cell should span. |
| scope | col colgroup row rowgroup | It specifies the cells that the header element relates to. |
| valign | top middle bottom baseline | It determines the vertical alignment of the cell content. |
| width | % pixels | It determines the width of the header cell |

# **HTML <thead> tag**

HTML <thead> elements is used to define header of an HTML table. The <thead> tag is used along with <tbody> and <tfoot> tags which defines table header, table body, and table footer in an HTML table.

The <thead> tag must be child of <table> element, and it must be used before any <tbody>, <tr>, or <tfoot> elements.

The <thead> tag should contain at least one row <tr> element inside it.

Refer to above examples.

## Attribute:

### **Tag-specific attributes:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| align | right left center justify char | It determines the alignment of the content inside the <thead> element |
| char | Character | It specifies the alignment of the content inside the <thead> element  to the character. |
| charoff | Number | It specifies the number of characters the content that will be aligned from the character specified by the char attribute. |
| valign | top middle bottom baseline | It determines the vertical alignment of the content inside the <thead> element |

# **HTML Time Tag**

**HTML <time> tag** is used to define date and time. It displays time value in a 24 hour clock or a precise date in a Gregorian calendar in HTML.

It is used to encode dates and times in a machine-readable way to make easy to mark or schedule your task.

It also helps search engines to produce smarter search results.

HTML <time> is a new tag and introduced in HTML5.

Let's see the syntax to define date and time.

**<time>**Define Time and Date here**</time>**

Attribute

There is only one specific attribute of HTML5 time tag.

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| datetime | It is used to define machine-readable date/time within the time element. |

<!DOCTYPE>

<html>

<body>

<p>We open our shop at <time>09:00</time> am.</p>

<p>The business meeting is scheduled on <time datetime="2009-02-18">next wednesday</time>.</p>

<p>The wedding of Salman's sister was scheduled at <time datetime="2014-11-19 T0 7:00-09:00">7pm last wednesday </time>.</p>

</body>

</html>

# **HTML <track> tag**

HTML <track> tag is used to define time-based text tracks for a media file. The <track> tag must use as child element of <audio> and <video> elements.

The <track> tag is used to add subtitle, caption, or any other form of text which displayed when a media file plays.

HTML <track> is new tag in HTML5.

### **Syntax**

1. **<track** src=" " kind=" " srclang=" " label=" "**>**

**Following are some specifications about the HTML <track> tag**

|  |  |
| --- | --- |
| **Display** | **None** |
| **Start tag/End tag** | Only start tag(End tag forbidden) |
| **Usage** | HTML media |

### **Example**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**HTML track Tag**</title>**
5. **</head>**
6. **<body>**
7. **<h2>**Example of track tag**</h2>**
8. **<video** controls="controls"**>**
9. **<source** src="flower.mp4" type="video/mp4"**>**
10. **<track** src="flower.vtt" kind="subtitles" srclang="en" label="English"**>**
11. Sorry!Your browser does not support the track
12. **</video>**
13. **</body>**
14. **</html>**

## How to create WEBVTT file:

Following are some basic steps to create WEBVTT file for <track> tag:

1. Open text editor in your PC such as Notepad
2. Write WEBVTT as the first line in the editor
3. Leave a blank line
4. Specify the time duration in the proper format (you can also provide numbering and add CSS).
5. Enter and write your text which you want to add a subtitle or caption, and repeat step 3 to 5 until you finish it.
6. Save it using .vtt extension.

Now your WEBVTT file is ready to use.

## Attribute:

### **Tag-specific attributes:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| default | default | It specifies that the track should be enabled unless the user?s preferences indicate that another track is more important. |
| kind | captions chapters descriptions metadata subtitles | It specifies that which type of text track you want to add. |
| label | text | It specifies the title of the text track. |
| src | URL | It defines the URL of the track file. |
| srclang | language\_code | It defines the language of the track text content, such as English, Germany, etc. |

# **HTML <var> tag**

HTML <var> tag is a phrase tag which is used to define the variable for a mathematical equation, or in the programming context.

The content within <var> tag renders in italic font in most of the browsers, but it can be overridden using appropriate CSS.

Following are some related elements of <var> tag, which can also be used for the same context:

* <code>: To determine the computer programming code.
* <kbd>: To determine the keyboard input.
* <samp>: To determine the sample output.

### **Syntax**

1. **<var>**........**</var>**

**Following are some specifications about the HTML <var> tag**

|  |  |
| --- | --- |
| **Display** | **Inline** |
| **Start tag/End tag** | Start and End tag |
| **Usage** | Formatting |

### **Example**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**HTML var tag**</title>**
5. **</head>**
6. **<body>**
7. **<h2>**Example of var tag**</h2>**
8. **<p>**Following is equation for distributive law**</p>**
9. **<p><var>**a**</var>**(**<var>**b**</var>**+**<var>**c**</var>**)=**<var>**ab**</var>**+**<var>**ac**</var></p>**
10. **</body>**
11. **</html>**

# **HTML Video Tag**

HTML 5 supports <video> tag also. The HTML video tag is used for streaming video files such as a movie clip, song clip on the web page.

Currently, there are three video formats supported for HTML video tag:

1. mp4
2. webM
3. ogg

Let's see the table that defines which web browser supports video file format.

|  |  |  |  |
| --- | --- | --- | --- |
| **Browser** | **mp4** | **webM** | **ogg** |
| ie browser Internet Explorer | yes | no | no |
| chrome browser Google Chrome | yes | yes | yes |
| firefox browser Mozilla Firefox | yes | yes | yes |
| opera browser Opera | no | yes | yes |
| safari browser Apple Safari | yes | no | no |

#### **Android also supports mp4 format.**

## HTML Video Tag Example

Let's see the code to play mp4 file using HTML video tag.

<!DOCTYPE>

<html>

<body>

<video controls>

<source src="movie.mp4" type="video/mp4">

Your browser does not support the html video tag.

</video>

</body>

</html>

Let's see the example to play ogg file using HTML video tag.

1. **<video** controls**>**
2. **<source** src="movie.ogg" type="video/ogg"**>**
3. Your browser does not support the html video tag.
4. **</video>**

Attributes of HTML Video Tag

Let's see the list of HTML 5 video tag attributes.

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| controls | It defines the video controls which is displayed with play/pause buttons. |
| height | It is used to set the height of the video player. |
| width | It is used to set the width of the video player. |
| poster | It specifies the image which is displayed on the screen when the video is not played. |
| autoplay | It specifies that the video will start playing as soon as it is ready. |
| loop | It specifies that the video file will start over again, every time when it is completed. |
| muted | It is used to mute the video output. |
| preload | It specifies the author view to upload video file when the page loads. |
| src | It specifies the source URL of the video file. |

HTML Video Tag Attribute Example

Let's see the example of video tag in HTML where are using height, width, autoplay, controls and loop attributes.

<!DOCTYPE>

<html>

<body>

<video width="320" height="240" controls autoplay loop>

<source src="movie.mp4" type="video/mp4">

Your browser does not support the html video tag.

</video>

</body>

</html>

MIME Types for HTML Video format

The available MIME type HTML video tag is given below.

|  |  |
| --- | --- |
| **Video Format** | **MIME Type** |
| mp4 | video/mp4 |
| ogg | video/ogg |
| webM | video/webM |

# **HTML Wbr Tag**

**HTML <wbr> tag** is used to specify a line break opportunity within an HTML document.

Without wbr tag, it is very difficult to read a long single word or a sentence. Without wbr tag, single long word can wrap or not wrap at all, it creates problem for the layout of the page.

Without <wbr> tag, users will have to scroll right to read a long complete word or sentence.

The <wbr> tag is new and introduced in HTML 5.

#### **The wbr tag doesn't break line.**

## Difference between <wbr> and <br> tag

The <br> tag forces a line break while <wbr> tag only represents a line break opportunity. It only facilitate the browser to line break if necessary (for example at the end of page .

## How to test <wbr> tag

To test HTML wbr tag, write too many characters without wbr tag, it will enable a scroll bar for you. But if you write too many characters within wbr tag, it will not enable a scroll bar but break the line.

## HTML wbr tag example

<html>

<body>

<p> This is a world record for the largest word,

a 45-letter word appears in a major dictionary

<i>pneumonoultramicroscopicsilicovolcanoconiosis</i>

</p>

<p>Here's what it looks like without using the <code>wbr</code> tag...<br/>

<i> pneumonoultramicroscopicsilicovolcanoconiosis </i></p>

<p> It will look like this after using wbr tag. </p>

<i> pneu<wbr>monoultra<wbr>microsc<wbr>opicsili<wbr>covolcan<wbr>iosis</i>

</body>

</html>

# **HTML Event Attributes**

When a browser reacts on user action, then it is called as an event. For example, when you click on the submit button, then if the browser displays an information box.

In HTML5 there are lots of event attributes available which can be activated using a programming language such as JavaScript.

Following is a table of event attributes, using these attributes you can perform several events.

## Windows Event Attributes

Windows events are related for the window object, and it can only be applied with <body> tag.

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| onafterprint | Executed the script after the document is printed. |
| onbeforeprint | Executed the script before the document is printed. |
| onbeforeunload | Executed the script before a document being unloaded. |
| onerror | Executed the script when an error occurs. |
| onhashchange | Executed the script when the anchor part in URL of the webpage is changed. |
| onload | Executed the script when the webpage is entirely loaded. |
| onmessage | Executed the script when a message event occurs. |
| onoffline | Executed the script when the network connection is disconnected, and browser started working offline. |
| ononline | Executed the script when the browser started working online |
| onpagehide | Executed the script when the current webpage is hidden such as if the user has moved away from the current webpage. |
| onpageshow | Executed the script when the current webpage is focused. |
| onpopstate | Executed the script when the window's active history is changed. |
| onresize | Executed the script when the window is resized. |
| onstorage | Executed the script when web storage is updated. |
| onunload | Executed the script when the current webpage is unloaded, or window is closed. |

## Form Event Attributes

Form event occurs when the user performs some action within the form such as submitting the form, selecting input field, etc.

The form events can be used with any element, but these are mainly used with HTML form elements.

Following is the list of all Form Event attributes:

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| onblur | Executed the script when form element loses the focus. |
| onchange | Executed the script when the value of the element is changed. |
| onfocus | Trigger an event when the element gets focused. |
| oninput | Executed the script when the user enters input to the element. |
| oninvalid | Executed the script when the element does not satisfy its predefined constraints. |
| onreset | Triggers the event when user reset the form element values. |
| onsearch | Triggers the event when a search field receives some input. |
| onselect | Triggers the event when the user has selected some text. |
| onsubmit | Triggers the event when a form is submitted. |

## Keyboard Event Attributes

Keyboard event occurs when a user interacts with the keyboard. Following is a list of the Keyboard event.

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| onkeydown | Triggers the event when the user presses down a key on the keyboard. |
| onkeypress | Trigger the event when the user presses the key which displays some character. |
| onkeyup | Trigger the event when the user releases the currently pressed key. |

## Mouse Event Attributes

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| onclick | Trigger the event when the mouse clicks on the element. |
| ondblclick | Trigger the event when mouse double-click occurs on the element. |
| onmousedown | Trigger the event when the mouse button is pressed on the element. |
| onmousemove | Trigger the event when the mouse pointer moves over the element. |
| onmouseout | Trigger the event when the mouse moves outside the element. |
| onmouseover | Trigger the event when the mouse moves onto the element. |
| onmouseup | Trigger the event when the mouse button is released. |
| onmousewheel | Deprecated. Use the onwheel attribute. |
| onwheel | Trigger the event when the mouse wheel rolls up or down on the element |

## Clipboard Event Attributes

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| oncopy | Trigger the event when the user copies the content to the system clipboard. |
| oncut | Trigger the event when the content of an element is cut and copy to the clipboard. |
| onpaste | Trigger the event when the user pastes some content in an element. |

## Media Event Attributes

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| onabort | Executed the script when media playback is aborted. |
| oncanplay | Executed the script when the media file is ready to play. |
| oncanplaythrough | Executed the script when the media file is ready to play without buffering or stopping. |
| oncuechange | Executed the script text cue of <track> element is changed. |
| ondurationchange | Executed the script when the media file duration is changed. |
| onemptied | Executed the script if media occurs some fatal error, and the file becomes unavailable. |
| onended | Executed the script when the media file occurs its end point. |
| onerror | Executed the script when some error occurred while fetching the media data. |
| onloadeddata | Executed the script when media data is loaded. |
| onloadedmetadata | Executed the script when metadata of media file is loaded. |
| onloadstart | Executed the script when loading of media file starts. |
| onpause | Executed the script when media playback is paused. |
| onplay | Executed the script when media file ready to play after being paused. |
| onplaying | Executed the script when media file is started playing. |
| onprogress | Executed the script when the browser is in the process of getting the media data. |
| onratechange | Executed the script when playback speed changed. |
| onseeked | Executed the script when seek operation is ended and seeking attribute is set to false. |
| onseeking | Executed the script when seek operation is active and seeking attribute is set to true. |
| onstalled | Executed the script when browser unexpectedly stopped fetching the data media. |
| onsuspend | Executed the script if fetching of media data is intentionally stopped. |
| ontimeupdate | Executed the script when playback position is changed, such as if a user fasts forward the track. |
| onvolumechange | Executed the script when media volume is changed (muted or unmuted). |
| onwaiting | Executed the script if playback pause to wait for loading more data. |

# **HTML Google Maps**

HTML Google Map is used to display maps on your webpage. You can simply add a map on your basic HTML page.

**Syntax:**

1. <!DOCTYPE html>
2. <html>
3. <body>
4. <h1>First Google Map Example</h1>
5. <div id="map">My map will go here...</div>
6. </body>
7. </html>

## Set the Map Size

You can set the map size by using the following syntax:

1. <div id="map" style="width:400px;height:400px;background:grey"></div>

**How to create a function to set the map properties?**

You can set the map properties by creating a function. Here, the function is myMap(). This example shows the Google map centered in London, England.

We have to use the functionalities of Google Maps API provided by a JavaScript library located at Google. Use the following script to refer to the Google Maps API with a callback to the myMap function.

1. <script src="https://maps.googleapis.com/maps/api/js?callback=myMap"></script>

### **Example:**

1. <!DOCTYPE html>
2. <html>
3. <body>
4. <h1>My First Google Map</h1>
5. <div id="map" style="width:400px;height:400px;background:grey"></div>
6. <script>
7. function myMap() {
8. var mapOptions = {
9. center: **new** google.maps.LatLng(51.5, -0.12),
10. zoom: 10,
11. mapTypeId: google.maps.MapTypeId.HYBRID
12. }
13. var map = **new** google.maps.Map(document.getElementById("map"), mapOptions);
14. }
15. </script>
16. <script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBu-916DdpKAjTmJNIgngS6HL\_kDIKU0aU&callback=myMap"></script>
17. </body>
18. </html>

## Example Explanation

**mapOptions:** It is a variable which defines the properties for the map.

**center:** It specifies where to center the map (using latitude and longitude coordinates).

**zoom:** It specifies the zoom level for the map (try to experiment with the zoom level).

**mapTypeId:** It specifies the map type to display. The following map types are supported: ROADMAP, SATELLITE, HYBRID, and TERRAIN.

**var map=new google.maps.Map(document.getElementById("map"), mapOptions):** It creates a new map inside the

element with id="map", using the parameters that are passed (mapOptions).

## HTML Multiple Maps

You can use different map types in a single example.

### **Example:**

1. <!DOCTYPE html>
2. <html>
3. <body>
4. <div id="googleMap1" style="width:400px;height:300px;"></div>
5. <br>
6. <div id="googleMap2" style="width:400px;height:300px;"></div>
7. <br>
8. <div id="googleMap3" style="width:400px;height:300px;"></div>
9. <br>
10. <div id="googleMap4" style="width:400px;height:300px;"></div>
11. <script>
12. function myMap() {
13. var mapOptions1 = {
14. center: **new** google.maps.LatLng(51.508742,-0.120850),
15. zoom:9,
16. mapTypeId: google.maps.MapTypeId.ROADMAP
17. };
18. var mapOptions2 = {
19. center: **new** google.maps.LatLng(51.508742,-0.120850),
20. zoom:9,
21. mapTypeId: google.maps.MapTypeId.SATELLITE
22. };
23. var mapOptions3 = {
24. center: **new** google.maps.LatLng(51.508742,-0.120850),
25. zoom:9,
26. mapTypeId: google.maps.MapTypeId.HYBRID
27. };
28. var mapOptions4 = {
29. center: **new** google.maps.LatLng(51.508742,-0.120850),
30. zoom:9,
31. mapTypeId: google.maps.MapTypeId.TERRAIN
32. };
33. var map1 = **new** google.maps.Map(document.getElementById("googleMap1"),mapOptions1);
34. var map2 = **new** google.maps.Map(document.getElementById("googleMap2"),mapOptions2);
35. var map3 = **new** google.maps.Map(document.getElementById("googleMap3"),mapOptions3);
36. var map4 = **new** google.maps.Map(document.getElementById("googleMap4"),mapOptions4);
37. }
38. </script>
39. <script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBu-916DdpKAjTmJNIgngS6HL\_kDIKU0aU&callback=myMap"></script>
40. </body>
41. </html>

# **HTML5 Semantics**

In any language, it is essential to understand the meaning of words during communication. And if this is a computer communication then it becomes more critical. So HTML5 provides more semantic elements which make easy understanding of the code.

Hence Semantics defines the meaning of words and phrases, i.e.

Semantic elements= elements with a meaning. Semantic elements have a simple and clear meaning for both, the browser and the developer.

### **For example:**

In HTML4 we have seen <div>, <span> etc. are which are non-semantic elements. They don't tell anything about its content.

On the other hand, <form>, <table>, and <article> etc. are semantic elements because they clearly define their content.

HTML5 semantic elements are supported by all major browsers.

## Why to use semantic elements?

In HTML4, developers have to use their own id/class names to style elements: header, top, bottom, footer, menu, navigation, main, container, content, article, sidebar, topnav, etc.

This is so difficult for search engines to identify the correct web page content. Now in HTML5 elements (<header> <footer> <nav> <section> <article>), this will become easier. It now allows data to be shared and reused across applications, enterprises, and communities."

Semantic elements can increase the accessibility of your website, and also helps to create a better website structure.

## Semantic Elements in HTML5

|  |  |  |
| --- | --- | --- |
| **Index** | **Semantic Tag** | **Description** |
| 1. | <article> | Defines an article |
| 2. | <aside> | Defines content aside from the page content |
| 3. | <details> | Defines additional details that the user can view or hide |
| 4. | <figcaption> | Defines a caption for a <figure> element |
| 5. | <figure> | Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc. |
| 6. | <footer> | Defines a footer for a document or section |
| 7. | <header> | Specifies a header for a document or section |
| 8. | <main> | Specifies the main content of a document |
| 9. | <mark> | Defines marked/highlighted text |
| 10. | <nav> | Defines navigation links |
| 11. | <section> | Defines a section in a document |
| 12. | <summary> | Defines a visible heading for a <details> element |
| 13. | <time> | Defines a date/time |

## Some important semantic elements in HTML5

### **HTML5 <article> Element**

HTML <article> element defines article content within a document, page, application, or a website. It can be used to represent a forum post, a magazine, a newspaper article, or a big story.

### **Example:**

<!DOCTYPE html>

<html>

<body>

<nav>

<a href="https://www.iHub.com/html-tutorial">HTML</a> |

<a href="https://www. iHub.com/java-tutorial">Java</a> |

<a href="https://www. iHub.com/php-tutorial">PHP</a> |

<a href="https://www. iHub.com/css-tutorial">CSS</a>

</nav>

</body>

</html>

### **HTML5 <aside> Element**

The <aside> element represent the content which is indirectly giving information to the main content of the page. It is frequently represented as a sidebar.

### **Example:**

1. <body>
2. <h2>My last year memories</h2>
3. <p>I have visited Paris with my friends last month. This was the memorable journey and i wish to go there again.</p>
4. <aside>
5. <h4>Paris</h4>
6. <p>Paris, France's capital, is a major European city and a global center **for** art, fashion, gastronomy and culture</p>
7. </aside>
8. </body>

### **HTML5 <section> Element**

The <section> element is used to represent the standalone section within an HTML document. A page can have various sections and each section can contain any content, but headings for each section is not mandatory.

### **Example:**

1. <h2>Web designing Tutorial</h2>
2. <section>
3. <h3>HTML</h3>
4. <p>HTML is an acronym which stands **for** Hyper Text Markup Language which is used **for** creating web pages and web applications.</p>
5. </section>
6. <section>
7. <h3>CSS</h3>
8. <p>CSS stands **for** Cascading Style Sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML.</p>
9. </section>

## Nesting <article> tag in <section> tag or Vice Versa?

We know that the<article> element specifies independent, self-contained content and the <section> element defines section in a document.

In HTML, we can use <section> elements within <article> elements, and <article> elements within <section> elements.

We can also use <section> elements within <section> elements, and <article> elements within <article> elements.

#### **For example:**

In a newspaper, the sport <article> in the sport section, may have a technical section in each <article>.

### **HTML5 <nav> Element**

The HTML <nav> element is used to define a set of navigation links.

### **Example:**

1. <!DOCTYPE html>
2. <html>
3. <body>
4. <nav>
5. <a href="https://www. iHub.com/html-tutorial">HTML</a> |
6. <a href="https://www. iHub.com/java-tutorial">Java</a> |
7. <a href="https://www. iHub.com/php-tutorial">PHP</a> |
8. <a href="https://www. iHub.com/css-tutorial">CSS</a>
9. </nav>
10. </body>
11. </html>

### **HTML5 <header> Element**

The <header> element represent the header of the document which can contain introductory content or navigation links.

### **Example:**

1. <header>
2. <h1>Welcome to Web123.com</h1>
3. <nav>
4. <ul>
5. <li>Home |</li>
6. <li>About us |</li>
7. <li>Contact us</li>
8. </ul>
9. </nav>
10. </header>

### **HTML5 <footer> Element**

The <footer> tag defines the footer of an HTML document or page.

### **Example:**

1. <footer>
2. <p>© Copyright 2019. All rights reserved. </p>
3. </footer>

# **HTML5 Migration**

HTML5 migration specifies that how to migrate from HTML4 to HTML5. Let?s see how to convert HTML4 page into HTML5 page without any problem in content or structure.

**Table:**

|  |  |
| --- | --- |
| **In HTML4** | **In HTML5** |
| <div id="header"> | <header> |
| <div id="menu"> | <nav> |
| <div id="content"> | <section> |
| <div class="article"> | <article> |
| <div id="footer"> | <footer> |

**Let's see a typical HTML4 page.**

### **Example:**

1. <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
2. <html lang="en">
3. <head>
4. <meta http-equiv="Content-Type" content="text/html;charset=utf-8">
5. <title>HTML4</title>
6. <style>
7. body {
8. font-family: Verdana,sans-serif;
9. font-size: 0.9em;
10. }
12. div#header, div#footer {
13. padding: 10px;
14. color: white;
15. background-color: black;
16. }
18. div#content {
19. margin: 5px;
20. padding: 10px;
21. background-color: lightgrey;
22. }
24. div.article {
25. margin: 5px;
26. padding: 10px;
27. background-color: white;
28. }
30. div#menu ul {
31. padding: 0;
32. }
34. div#menu ul li {
35. display: inline;
36. margin: 5px;
37. }
38. </style>
39. </head>
40. <body>
42. <div id="header">
43. <h1>iHub Times</h1>
44. </div>
46. <div id="menu">
47. <ul>
48. <li>Tutorials</li>
49. <li>Technology</li>
50. <li>Blog</li>
51. </ul>
52. </div>
54. <div id="content">
55. <h2>Tutorials Section</h2>
56. <div **class**="article">
57. <h2>Tutorial1</h2>
58. <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in porta lorem.
59. Morbi condimentum est nibh, et consectetur tortor feugiat at.</p>
60. <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in porta lorem.
61. Morbi condimentum est nibh, et consectetur tortor feugiat at.</p>
62. </div>
63. <div **class**="article">
64. <h2>Tutorial2</h2>
65. <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in porta lorem.
66. Morbi condimentum est nibh, et consectetur tortor feugiat at.</p>
67. <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in porta lorem.
68. Morbi condimentum est nibh, et consectetur tortor feugiat at.</p>
69. </div>
70. </div>
72. <div id="footer">
73. <p>© 2018 iHub Times. All rights reserved.</p>
74. </div>
76. </body>
77. </html>

## Change HTML4 Doctype to HTML5 Doctype

**HTML4 Doctype Syntax:**

1. <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

**HTML5 Doctype Syntax:**

1. <!DOCTYPE html>

### Example:

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4. <meta http-equiv="Content-Type" content="text/html;charset=utf-8">
5. <title>HTML5</title>
6. <style>
7. body {
8. font-family: Verdana,sans-serif;
9. font-size: 0.9em;
10. }
11. div#header, div#footer {
12. padding: 10px;
13. color: white;
14. background-color: black;
15. }
17. div#content {
18. margin: 5px;
19. padding: 10px;
20. background-color: lightgrey;
21. }
22. div.article {
23. margin: 5px;
24. padding: 10px;
25. background-color: white;
26. }
27. div#menu ul {
28. padding: 0;
29. }
30. div#menu ul li {
31. display: inline;
32. margin: 5px;
33. }
34. </style>
35. </head>
36. <body>
37. <div id="header">
38. <h1>iHub Times</h1>
39. </div>
41. <div id="menu">
42. <ul>
43. <li>Tutorials</li>
44. <li>Technology</li>
45. <li>Blog</li>
46. </ul>
47. </div>
48. <div id="content">
49. <h2>Tutorials Section</h2>
50. <div **class**="article">
51. <h2>Tutorial1</h2>
52. <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in porta lorem.
53. Morbi condimentum est nibh, et consectetur tortor feugiat at.</p>
54. <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in porta lorem.
55. Morbi condimentum est nibh, et consectetur tortor feugiat at.</p>
56. </div>
57. <div **class**="article">
58. <h2>Tutorial2</h2>
59. <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in porta lorem.
60. Morbi condimentum est nibh, et consectetur tortor feugiat at.</p>
61. <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque in porta lorem.
62. Morbi condimentum est nibh, et consectetur tortor feugiat at.</p>
63. </div>
64. </div>
65. <div id="footer">
66. <p>© 2018 iHub Times. All rights reserved.</p>
67. </div>
69. </body>
70. </html>

# **HTML Space Code**

There are multiple ways to insert spaces between the words or characters:

## Using Spacebar

To add a space between the words, first we have to click where we want to add a space and then press space bar. The HTML document use only one space between the words. In simple words, if we press spacebar more than one time, then it will show only one space between the words on the browser as described in the following example:

### **Example**

1. **<head>**
2. **<title>**
3. Example of space
4. **</title>**
5. **<head>**
6. **<body>**
7. iHub Institute
8. **</body>**
9. **</html>**

In this example, we press the spacebar more than one time, but the following output shows only one space between the iHub and Institute.

### **Using &nbsp;**

The **&nbsp;** is a non-breaking space that is used in the HTML document. It is most commonly used method for spacing, which prevents a line from breaking. The following example determines how to use  :

### **Example**

1. **<html>**
2. **<head>**
3. **<title>**
4. Example of space
5. **</title>**
6. **<head>**
7. **<body>**
8. iHub  &nbsp; &nbsp; &nbsp; &nbsp;  Institute
9. **</body>**
10. **</html>**

# **HTML Background-color**

The **<bgcolor>** is the attribute to set the background color of an HTML element. This attribute is used with the following tags:

* **<body>**
* **<table>**
* **<marquee>**
* **<td>**
* **<th>**
* **<tr>**

### **Syntax**

1. **<**"tag" bgcolor="Color\_name|rgb number|Hex number"**>**

**Examples:** The following examples use the <bgcolor> attribute in the different tags.

**Example 1:** Use the <bgcolor> attribute with the <body> tag

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**
5. Example of Background color Attribute
6. **</title>**
7. **</head>**
8. **<body**  bgcolor="lightblue"**>**
9. <!-- The attribute bgcolor use with the body tag to set the background of web page as lightblue -->
10. **<center>**
11. **<h1>** iHub Institute**</h1>**
12. **<br>** **<br>**
13. **<center>**
14. **<h2>** Hyper Text Markup Language **</h2>**
15. **</center>**
16. **</body>**
17. **</html>**

**Example 2:** Use the <bgcolor> attribute with the <tr> tag

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**
5. Example of Bgcolor Attribute
6. **</title>**
7. **</head>**
8. **<body**  bgcolor="red"**>**
9. <!-- The attribute bgcolor use with the 'body' tag to set the background of web page as red -->
10. **<center>**
11. **<h1>** iHub**</h1>**
12. **<br>**
13. **<br>**
14. **<center>**
15. **<table>**
16. <!-- The attribute bgcolor use with the 'tr' tag to set the background of table rows by different colors -->
17. **<tr** bgcolor="blue"**>**
18. **<td>** Roll No. **</td>**
19. **<td>** Name **</td>**
20. **</tr>**
21. **<tr** bgcolor="lightblue"**>**
22. **<td>** 101 **</td>**
23. **<td>** Abhay **</td>**
24. **</tr>**
25. **<tr** bgcolor="brown"**>**
26. **<td>** 102 **</td>**
27. **<td>** Chetan **</td>**
28. **</tr>**
29. **<tr** bgcolor="orange"**>**
30. **<td>** 103 **</td>**
31. **<td>** Manpreet **</td>**
32. **</tr>**
33. **<tr** bgcolor="yellow"**>**
34. **<td>** 104 **</td>**
35. **<td>** Rakesh **</td>**
36. **</tr>**
37. **<tr** bgcolor="lightgreen"**>**
38. **<td>** 105 **</td>**
39. **<td>** Sumit **</td>**
40. **</tr>**
41. **</table>**
42. **</center>**
43. **</body>**
44. **</html>**

**Example 3:** The following example use the **<bgcolor>** attribute with the **<marquee>** tag.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<meta** name="viewport" content="width=device-width, initial-scale=1"**>**
5. **<title>**
6. Example of Background color Attribute
7. **</title>**
8. **</head>**
9. **<body**  bgcolor="lightgreen"**>**
10. <!-- The attribute bgcolor use with the body tag to set the background of web page as lightgreen -->
12. **<h1>** **<font** color="blue"**>**
13. **<marquee** bgcolor="orange"**>**
14. <!-- The attribute bgcolor use with the marquee tag to set the background of marquee as orange -->
15. iHub
16. **</marquee>**
17. **</font>**
18. **</h1>**
19. **</body>**
20. **</html>**

# **HTML Hide Element**

You can hide an element by using the Boolean attribute **hidden** with the element. When you specify the hidden attribute in the [HTML](https://www.javatpoint.com/html-tutorial) file, then the browser will not display that element, which is specified with this attribute.

### **Syntax**

1. **<element** or tag hidden**>** Any statement or content **</element** or tag**>**

**Examples:**T he following examples are specified to understand easily how to use the hidden attribute with different elements or tags:

**Example 1:** This example uses the hidden attribute with the [paragraph tag](https://www.javatpoint.com/html-paragraph).

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**
5. First Example of Hidden attribute
6. **</title>**
7. **</head>**
8. **<body>**
9. **<center>**
10. **<h1>** iHub **</h1>**
11. **</center>**
12. **<p** hidden**>** This paragraph should be hidden.
13. **</p>**
14. **</body>**
15. **</html>**

**Example 2:** The following example uses the hidden attribute with the <input type=text> element.

In this example, we have used the hidden attribute with two [input tags](https://www.javatpoint.com/html-input-tag). These input fields will not display on the web page, when the following program executes.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<meta** name="viewport" content="width=device-width, initial-scale=1"**>**
5. **<title>**
6. Example of Hidden attribute in input tag
7. **</title>**
8. **<style>**
9. /\* The following tag selector body use the font-family and background-color properties for body of a page\*/
10. body {
11. font-family: Calibri, Helvetica, sans-serif;
12. background-color: pink;
13. }
14. /\* Following container class used padding for generate space around it, and also use a background-color for specify the color lightblue as a background \*/
15. .container {
16. padding: 50px;
17. background-color: lightblue;
18. }
19. /\* The following tag selector input use the different properties for the text filed. \*/
20. input[type=text] {
21. width: 100%;
22. padding: 15px;
23. margin: 5px 0 22px 0;
24. border: none;
25. background: #f1f1f1;
26. }
28. input[type=text]:focus {
29. background-color: orange;
30. outline: none;
31. }
32. /\* The following div tag selector is used to provide the space or gap between the content or elements on a web page. \*/
33. div {
34. padding: 10px 0;
35. }
36. hr {
37. border: 1px solid #f1f1f1;
38. margin-bottom: 25px;
39. }
40. **</style>**
41. **</head>**
42. **<body>**
43. **<form>**
44. **<div** class="container"**>**
45. **<center>**  **<h1>** Registration Form**</h1>** **</center>**
47. **<hr>**
48. **<label** for="fn"**>** Firstname: **</label>**
49. <!-- The following input field not display on the web page because the hidden attribute is used in this <input> tag. -->
50. **<input** hidden type="text"  name="firstname" id= "fn" size="15" required **/>**
51. **<div>**
52. **<label** for="mn"**>** Middlename: **</label>**
53. **<input** type="text" name="middlename" id="mn" size="15" required **/>**
54. **</div>**
55. **<label** for="Ln"**>** Lastname: **</label>**
56. **<input** type="text" name="lastname" id="Ln" size="15"required **/>**
58. **<label** for="pn"**>**
59. Phone :
60. **</label>**
61. **<input** type="text" name="country code" placeholder="Country Code"  value="+91" size="2"**/>**
62. <!-- The following input field not display on the web page because it is also use the hidden attribute-->
63. **<input** hidden type="text"  name="phone" id="pn" size="10"/ required**>**
64. **</hr>**
65. **</div>**
66. **</form>**
67. **</body>**
68. **</html>**

# **HTML Search Box**

The **HTML Search Box** allows a user to search the content. The **Search** is a value of **type** attribute of an **<input>** element.

### **Syntax**

1. **<input** type="search"**>**

**Examples:** The following examples have used the different CSS codes for displaying the different Search boxes on the web page.

**Example 1:**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**
5. Example of search box
6. **</title>**
7. **<style>**
8. /\* The following tag selector body use the text-align and background-color properties. The text-align property is used to align the text as center, and the background-color is used to specify the blue colour as a background of a page\*/
9. body {
10. text-align: center;
11. background-color: blue;
12. }
13. /\* The following tag selector form uses the different properties for displaying a form. \*/
15. form{
16. display: block;
17. left: 30%;
18. position: absolute;
19. top: 30%;
20. }
21. /\* The following tag selector input uses the different properties for the search box. \*/
22. input[type=search]{
23. border: 5px  orange;
24. box-sizing: border-box;
25. font-size:1em;
26. height: 2em;
27. margin-left: 10vw;
28. padding: .5em;
29. transition: all 2s ease-in;
30. width: 30vw;
31. z-index:1;
32. &:focus {
33. border: solid 3px #09f;
34. outline: solid #fc0 2000px;
35. }
36. }
37. **</style>**
38. **</head>**
39. **<body>**
40. **<form>**
41. **<input** type="search" placeholder="Search for any site"**/>**
42. **</form>**
43. **</body>**
44. **</html>**

**Example 2:** In this example, we have not used the search value of the type attribute of an [input](https://www.javatpoint.com/html-input-tag) element. But we have used the text input field using [CSS](https://www.javatpoint.com/css-tutorial), which is working as a search field.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**
5. Second Example of search box
6. **</title>**
7. **<style>**
8. \*
9. {
10. outline: none;
11. }
12. /\* The following tag selector form uses one property for specifying the height of a form. \*/
14. form
15. {
16. height: 96px;
17. }
18. /\* The following tag selector input use the different properties for specifying the text field. \*/
20. input[type="text"]
21. {
22. width: 100%;
23. height: 80px;
24. font-size: 40px;
25. line-height: 1;
26. }
27. /\* The following tag selector uses the color of placeholder as orange for the input text field. \*/
28. input[type="text"]::placeholder
29. {
30. color: orange;
31. }
33. .tb
34. {
35. display: table;
36. width: 100%;
37. }
38. .td
39. {
40. display: table-cell;
41. vertical-align: middle;
42. }
43. /\* The following tag selector uses the different properties for specifying all the inputs and buttons used in the body tag in this html file. \*/
45. input, button
46. {
47. color: #fff;
48. font-family: Arial;
49. padding: 0;
50. margin: 0;
51. border: 0;
52. background-color: transparent;
53. }
54. /\* the following id selector cover uses the different attribute which are used between the body tag \*/
55. #cover
56. {
57. position: absolute;
58. top: 50%;
59. width: 550px;
60. padding: 35px;
61. margin: -83px auto 0 auto;
62. background-color: #ff7575;
63. border-radius: 20px;
64. left: 0;
65. right: 0;
66. box-shadow: 0 10px 40px #ff7c7c, 0 0 0 20px #ffffffeb;
67. transform: scale(0.6);
68. }
69. /\* The following tag selector button uses the different proprties for specifying the button on a web page. \*/
70. button
71. {
72. position: relative;
73. display: block;
74. width: 84px;
75. height: 96px;
76. cursor: pointer;
77. }
78. /\* The following id selector uses the different properties for searching button \*/
79. #s-circle
80. {
81. position: relative;
82. top: -8px;
83. left: 0;
84. width: 43px;
85. height: 43px;
86. margin-top: 0;
87. border-width: 15px;
88. border: 15px solid #fff;
89. background-color: transparent;
90. border-radius: 50%;
91. transition: 0.5s ease all;
92. }
93. button span
94. {
95. position: absolute;
96. top: 68px;
97. left: 43px;
98. display: block;
99. width: 45px;
100. height: 15px;
101. background-color: transparent;
102. border-radius: 10px;
103. transform: rotateZ(52deg);
104. transition: 0.5s ease all;
105. }
106. button span:before, button span:after
107. {
108. content: '';
109. position: absolute;
110. bottom: 0;
111. right: 0;
112. width: 45px;
113. height: 15px;
114. background-color: #fff;
115. border-radius: 10px;
116. transform: rotateZ(0);
117. transition: 0.5s ease all;
118. }
119. #s-cover:hover #s-circle
120. {
121. top: -1px;
122. width: 67px;
123. height: 15px;
124. border-width: 0;
125. background-color: #fff;
126. border-radius: 20px;
127. }
128. /\* The following hover effect use the different properties when you take mouse to that element where thr s-cover is selector is used. \*/
129. #s-cover:hover span
130. {
131. top: 50%;
132. left: 56px;
133. width: 25px;
134. margin-top: -9px;
135. transform: rotateZ(0);
136. }
137. #s-cover:hover button span:before
138. {
139. bottom: 11px;
140. transform: rotateZ(52deg);
141. }
142. #s-cover:hover button span:after
143. {
144. bottom: -11px;
145. transform: rotateZ(-52deg);
146. }
147. #s-cover:hover button span:before, #s-cover:hover button span:after
148. {
149. right: 6px;
150. width: 40px;
151. background-color: #fff;
152. }
153. **</style>**
154. **</head>**
155. **<body>**
156. **<div** id="cover"**>**
157. **<form** method="get" action=""**>**
158. **<div** class="tb"**>**
159. **<div** class="td"**>**
160. <!-- The following tag is input for the text input field which is used as a seach field on a web page -->
161. **<input** type="text" placeholder="Search" required**></div>**
162. **<div** class="td" id="s-cover"**>**
163. **<button** type="submit"**>**
164. **<div** id="s-circle"**>** **</div>**
165. **<span>** **</span>**
166. **</button>**
167. **</div>**
168. **</div>**
169. **</form>**
170. **</div>**
171. **</body>**
172. **</html>**

# **HTML Nested Table**

The nested table in HTML means creating a table on a webpage inside another table on the same web page.

#### **Note: The inner table always has to be placed between the <td> .......... </td> of the outer table.**

**Example:** The following example describes how to create a nested table. In this example we create a table, which contains another table in the 2nd cell of first row.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**
5. Example of Nested Table
6. **</title>**
7. **</head>**
8. **<body>**
9. <!?The following table tag is used to create outer table on a webpage.
10. This table tag contains two attributes border and bordercolor.
11. The border attribute is used to specify the border (4) of the outer table, and the bordercolor is used to specify the green colour of border of an outer table. --**>**
12. **<table** border="4" bordercolor="green"**>**
13. **<tr>**
14. **<td>** It is a 1st Cell of 1st row in the 1st Table. **</td>**
15. **<td>** It is a 2nd Cell of 1st row in the 1st Table.
17. <!?The following table tag is used to create an inner table which is shown in the second cell of the first row in the outer table on a webpage.
18. This table tag also contains two attributes border and bordercolor.
19. The border attribute is used to specify the border (6) of the inner table, and the bordercolor is used to specify the blue colour of border of an inner table. --**>**
20. **<table** border="6" bordercolor="blue"**>**
21. **<tr>**
22. **<td>** It is a 1st Cell of 1st row in the 2nd Table. **</td>**
23. **</tr>**
24. **<tr>**
25. **<td>** It is a 2nd Cell of 2nd row in the 2nd Table. **</td></tr>**
26. **</table>**
27. **</td>**
28. **</tr>**
29. **<tr>**
30. **<td>** It is a 3rd Cell of 2nd row in the 1st Table. **</td>**
31. **<td>** It is a 4th Cell of 2nd row in the 1st Table. **</td>**
32. **</tr>**
33. **</table>**
34. **</body>**
35. **</html>**

# **How to add Space in Html**

In HTML, we can easily add the space in the document using the following different ways:

1. Using Html
2. Using Internal CSS

## Using Html

If we want to add the space in the document using [Html tag](https://www.javatpoint.com/html-tags) to show the space before and after the text then we have to follow the steps which are given below. Using these steps, we can easily add the space:

**Step 1:** Firstly, we have to type the [Html](https://www.javatpoint.com/html-tutorial) code in any text editor or open the existing Html file in the text editor in which we want to add the space.

1. <!Doctype Html**>**
2. **<Html>**
3. **<Head>**
4. **<Title>**
5. Add the space using Html tags
6. **</Title>**
7. **</Head>**
8. **<Body>**
9. This page helps you to understand how to add the space in Html document.
10. And, this section helps you to understand how to add the space using the Html tags.
11. **</Body>**
12. **</Html>**

**Step 2:** Now, place the cursor where we want to add the space. And, then we have to press the spacebar from the keyboard. Basically, web page shows only one space between the two words. It does not matter that how much we have pressed the spacebar.

**Step 3:** If we want to show more than one space before or after the text on the webpage then we have to give the following tag:

1. &nbsp;    This tag is used for displaying only one space between the text.

**Step 4:** If we want to insert/add the longer space then we also use the following tags for showing two or four spaces between the text on the web page:

1. &ensp;       This tag is used for displaying two spaces.
3. &emsp;       This tag is used for displaying four spaces.

**Step 5:** We can also add the <pre> tag, which displays the text on the web page as same as entered in the Html document.

1. **<pre>**   Any text or paragraph **</pre>**

**Step 6:** When we have added the space successfully in the document, then we have to save the Html file and run the file.

1. <!Doctype Html**>**
2. **<Html>**
3. **<Head>**
4. **<Title>**
5. Add the space using Html tags
6. **</Title>**
7. **</Head>**
8. **<Body>**
9. This page helps you to understand   <!-- This tag shows one space between the 'understand' and 'how' words. -->
10. how to add the &emsp; <!-- This tag shows two space between the 'the' and 'space' words. --> space in  &ensp; <!-- This tag shows four space between the 'in' and 'Html' words. -->
11. Html document.
12. **<pre>** <!-- This tag shows the same content which is written between the pre tags. -->
13. And, this   section   helps  you to           understand how to add the space using the        Html tags.**</pre>**
14. **</Body>**
15. **</Html>**

## Using Internal CSS

If we want to add the space in the document using the [Internal Cascading stylesheet](https://www.javatpoint.com/internal-css) to show the space before and after the text, we have to follow the steps which are given below. Using these steps, we can easily add the space.

**Step 1:** Firstly, we have to type the Html code in any text editor or open the existing Html file in the text editor in which we want to use the **Internal CSS** for adding the space.

1. <!Doctype Html**>**
2. **<Html>**
3. **<Head>**
4. **<Title>**
5. Add the space using the Internal Cascading StyleSheet
6. **</Title>**
7. **</Head>**
8. **<Body>**
9. This page helps you to understand how to add the space in Html document.
10. And, this section helps you to understand how to add the space using the Internal Cascading Stylesheet.
11. **</Body>**
12. **</Html>**

**Step 2:** Now, we have to place the cursor in the [**head tag**](https://www.javatpoint.com/html-head) of the Html document and then define the styles inside the [**<style>** tag](https://www.javatpoint.com/html-style) as shown in the following block. And, then we have to type the **text-indent** attribute into the [**paragraph**](https://www.javatpoint.com/html-paragraph) element.

1. **<Head>**
2. **<Title>**
3. Add the space using the Internal Cascading StyleSheet
4. **</Title>**
5. **<style>**
6. p
7. {
8. text-indent: 5em;
9. }
10. .tab1 {
11. tab-size: 2;
12. }
13. **</style>**
14. **</Head>**

**Step 3:** Now, we have to type the paragraph tag and the class tab1 in the [<pre> tag](https://www.javatpoint.com/html-pre-tag) of the Html. And, at last, we have to save the Html file and then run the file.

1. <!Doctype Html**>**
2. **<Html>**
3. **<Head>**
4. **<Title>**
5. Add the space using the Internal Cascading StyleSheet
6. **</Title>**
7. **<style>**
8. p
9. {
10. text-indent:5em;
11. }
12. .tab1 {
13. tab-size: 2;
14. }
15. **</style>**
16. **</Head>**
17. **<Body>**
18. **<p>**This page helps you to understand how to add the space in Html document.
19. **<pre** class="tab1"**>**
20. And, this section helps you to understand how to      add the  **<br>**space using the Internal Cascading Stylesheet.**</pre>** **</p>**
21. **</Body>**
22. **</Html>**

# **How to Rotate Image in Html**

If we want to rotate an image in Html, then we have to follow the steps which are given below. Using these steps, we can easily rotate an image.

**Step 1:** Firstly, we have to type the [Html](https://www.javatpoint.com/html-tutorial) code in any text editor or open the existing Html file in the text editor in which we want to rotate an image.

1. <!Doctype Html**>**
2. **<Html>**
4. **<Head>**
5. **<Title>**
6. Rotate an Image
7. **</Title>**
8. **</Head>**
9. **<Body>**
10. Hello User!... **<br>** **<center>**
11. **<img** src="https://www.iHub.com/images/logo/jtp\_logo.png" width="100" height="100" **>** **</center>**
12. **</Body>**
13. **</Html>**

**Step 2:** Now, we have to place the cursor in the head tag of the Html document and then define the styles inside the **<style>** tag as shown in the following block. And, then type the **different** attributes in any class. The following class in CSS [style tag](https://www.javatpoint.com/html-style) contains the different attributes which help for rotating the image.

1. **<Head>**
2. **<Title>**
3. Rotate an Image
4. **</Title>**
5. **<style>**
6. .rotate90 {
7. -webkit-transform:rotate(90deg);
8. -moz-transform: rotate(90deg);
9. -ms-transform: rotate(90deg);
10. -o-transform: rotate(90deg);
11. transform: rotate(90deg);
12. }
13. **</style>**
14. **</Head>**

**Step 3:** And then, we have to define the class in the <img> tag of that image which we want to rotate.

1. **<img** src="https://www.iHub.com/images/logo/jtp\_logo.png" width="100" height="100" class="rotate90"**>**

**Step 4:** And, at last, we have to save the Html file and then run the file in the [browser](https://www.javatpoint.com/browsers).

1. <!Doctype Html**>**
2. **<Html>**
3. **<Head>**
4. **<Title>**
5. Rotate an Image
6. **</Title>**
7. **<style>**
8. .rotate90 {
9. -webkit-transform:rotate(90deg);
10. -moz-transform: rotate(90deg);
11. -ms-transform: rotate(90deg);
12. -o-transform: rotate(90deg);
13. transform: rotate(90deg);
14. }
15. **</style>**
16. **</Head>**
17. **<Body>**
18. Hello User!... **<br>** **<center>**
19. **<img** src="https://www.iHub.com/images/logo/jtp\_logo.png" width="100" height="100" class="rotate90"**>** **</center>**
20. **</Body>**
21. **</Html>**

# **How to Wrap text in Html**

If we want to wrap the text in Html, then we have to follow the steps which are given below. Using these steps, any user can easily wrap any text, which is to be shown on the web page.

**Step 1:** Firstly, we have to type the [Html](https://www.javatpoint.com/html-tutorial) code in any text editor or open the existing Html file in the text editor in which we want to wrap the text.

1. <!Doctype Html**>**
2. **<Html>**
3. **<Head>**
4. **<Title>**
5. Wrap the text
6. **</Title>**
7. **</Head>**
8. **<Body>**
9. Hello User
10. Your are at IHub Site .
11. If we want to wrap the text in Html, then we have to follow the steps which are given on this page.
12. **</Body>**
13. **</Html>**

**Step 2:** Now, we have to use **word-wrap** property. So, we have to place the cursor between the head tag just after the title tag. And, then we have to define a class and [<div> tag](https://www.javatpoint.com/html-div-tag) sector in the [<style> tag](https://www.javatpoint.com/html-style) as shown in the following block.

1. **<Head>**
2. **<Title>**
3. Wrap the text
4. **</Title>**
5. **<style>**
6. div {
7. width: 100px;
8. border: 2px solid red;
9. }
10. div.class\_name {
11. word-wrap: normal;
12. }
13. **</style>**
14. **</Head>**

**Step 3:** Now, we have to define that class before the text, which we want to wrap.

1. **<div** class="a"**>**Any text which we want to wrap **</div>**

**Step 4:** And, at last, we have to save the file and run the file in the [browser](https://www.javatpoint.com/browsers).

1. <!Doctype Html**>**
2. **<Html>**
3. **<Head>**
4. **<Title>**
5. Wrap the text
6. **</Title>**
7. **<style>**
8. div {
9. width: 100px;
10. border: 2px solid red;
11. }
12. div.a {
13. word-wrap: normal;
14. }
15. div.b {
16. word-wrap: break-word;
17. }
18. **</style>**
19. **</Head>**
20. **<Body>**
21. Hello User
22. **<div** class="a"**>**Your are at IHubSite.Ifwe want to wrap the text in Html then we have to follow the steps which are given in this page. **</div>** **<br>**
23. **<div** class="b"**>**Your are at IHubSite.Ifwe want to wrap the text in Html then we have to follow the steps which are given in this page. **</div>**
24. **</Body>**
25. **</Html>**

# **How to make a Navigation Bar in Html**

If we want to make a navigation bar in Html, then we have to follow the steps which are given below. Using these steps, we can easily create the Navigation bar.

**Step 1:** Firstly, we have to type the [Html](https://www.javatpoint.com/html-tutorial) code in any text editor or open the existing Html file in the text editor in which we want to make a Navigation Bar.

1. <!Doctype Html**>**
2. **<Html>**
3. **<Head>**
4. **<Title>**
5. Make a Navigation Bar
6. **</Title>**
7. **</Head>**
8. **<Body>**
9. **</Body>**
10. **</Html>**

**Step 2:** Now, we have to define the [<nav> tag](https://www.javatpoint.com/html-nav-tag) in the [<body> tag](https://www.javatpoint.com/html-body-tag) where we want to make the bar.

1. **<Body>**
2. **<nav>**

5. **</nav>**
6. You are at IHub Site.....
7. **</Body>**
8. **</Html>**

**Step 3:** After then, we have to define the [<ul> tag](https://www.javatpoint.com/html-unordered-list), which is used to show the unordered list. And, then we have to define the list items in the <li> tag. We have to define those items which we want to show in the navigation bar.

1. **<Body>**
2. **<nav>**
3. **<ul>**
4. **<li>**
5. **<a** href="#"**>** Home **</a>**
6. **</li>**
7. **<li>**
8. **<a** href="#"**>** About **</a>**
9. **</li>**
10. **<li>**
11. **<a** href="#"**>** Contact **</a>**
12. **</li>**
13. **<li>** **<a** href="#"**>** Terms of use **</a>**
14. **</li>**
15. **<li>**
16. **<a** href="#"**>** Join Us **</a>**
17. **</li>**
18. **</ul>**
19. **</nav>**
20. You are at IHub Site.....
21. **</Body>**
22. **</Html>**

**Step 4:** After then, we have to place the cursor in the [<head>](https://www.javatpoint.com/html-head) just after the closing of the title tag. And then, we have to define the [<style> tag](https://www.javatpoint.com/html-style). Step 4: After then, we have to place the cursor in the <head> just after the closing of the title tag. And then, we have to define the <style> tag.

1. **<Head>**
2. **<Title>**
3. Make a Navigation Bar
4. **</Title>**
5. **<style** type=text/css**>**

8. **</style>**
9. **</Head>**

**Step 5:** Now, we have to specify different id attributes which are used to set the position, color of the navigation bar. So, we have to use the following code in the head tag. We can also change the value of properties according to our requirements.

1. **<style** type=text/css**>**
2. body
3. {
4. height: 125vh;
5. margin-top: 80px;
6. padding: 30px;
7. background-size: cover;
8. font-family: sans-serif;
9. }
10. header {
11. background-color: orange;
12. position: fixed;
13. left: 0;
14. right: 0;
15. top: 5px;
16. height: 30px;
17. display: flex;
18. align-items: center;
19. box-shadow: 0 0 25px 0 black;
20. }
21. header \* {
22. display: inline;
23. }
24. header li {
25. margin: 20px;
26. }
27. header li a {
28. color: blue;
29. text-decoration: none;
30. }
31. **</style>**

**Step 6:** After that, we have to type the [<header> tag](https://www.javatpoint.com/html-header-tag) just before the opening <nav> tag. And we have to also close this tag. And, at last, we have to save the Html file and then run the file in the [browser](https://www.javatpoint.com/browsers).

1. <!Doctype Html**>**
2. **<Html>**
3. **<Head>**
4. **<Title>**
5. Make a Navigation Bar
6. **</Title>**
7. **<style** type=text/css**>**
8. body
9. {
10. height: 125vh;
11. margin-top: 80px;
12. padding: 30px;
13. background-size: cover;
14. font-family: sans-serif;
15. }
16. header {
17. background-color: orange;
18. position: fixed;
19. left: 0;
20. right: 0;
21. top: 5px;
22. height: 30px;
23. display: flex;
24. align-items: center;
25. box-shadow: 0 0 25px 0 black;
26. }
27. header \* {
28. display: inline;
29. }
30. header li {
31. margin: 20px;
32. }
33. header li a {
34. color: blue;
35. text-decoration: none;
36. }
37. **</style>**
38. **</Head>**
39. **<Body>**
40. **<header>**
41. **<nav>**
42. **<ul>**
43. **<li>**
44. **<a** href="#"**>** Home **</a>**
45. **</li>**
46. **<li>**
47. **<a** href="#"**>** About **</a>**
48. **</li>**
49. **<li>**
50. **<a** href="#"**>** Contact **</a>**
51. **</li>**
52. **<li>** **<a** href="#"**>** Terms of use **</a>**
53. **</li>**
54. **<li>**
55. **<a** href="#"**>** Join Us **</a>**
56. **</li>**
57. **</ul>**
58. **</nav>**
59. **</header>**
60. You are at IHub Site.....
61. **</Body>**
62. **</Html>**

# **How to add Social Media Icons in Html**

If we want to add social media icons in the Html document using the [Internal CSS](https://www.javatpoint.com/internal-css), to show them on the web page then we have to follow the steps which are given below. Using these simple steps, we can easily add the social media icons.

**Step 1:** Firstly, we have to type the [Html](https://www.javatpoint.com/html-tutorial) code in any text editor or open the existing Html file in the text editor in which we want add the icons of social media.

1. <!Doctype Html**>**
2. **<Html>**
3. **<Head>**
4. **<Title>**
5. Add the icons of social media
6. **</Title>**
7. **</Head>**
8. **<Body>**
9. Hello User!...
10. You are at IHub Site...
11. **</Body>**
12. **</Html>**

**Step 2:** Now, we have to place the cursor in the head tag just after the title tag of the Html document and then define the **<style>** tag as shown in the following block.

**Step 3:** And, then we have to use the following code between the closing and opening of [<style> tag](https://www.javatpoint.com/html-style).

1. .fa {
2. padding: 20px;
3. text-align: center;
4. margin: 5px 2px;
5. font-size: 30px;
6. width: 50px;
7. }
8. .fa-facebook {
9. background: #3B5998;
10. color: white;
11. }
12. .fa-twitter {
13. background: #55ACEE;
14. color: white;
15. }
16. .fa-pinterest {
17. background: #cb2027;
18. color: white;
19. }
20. .fa-linkedin {
21. background: #007bb5;
22. color: white;
23. }
24. .fa-instagram {
25. background: #125688;
26. color: white;
27. }
28. .fa-youtube {
29. background: #bb0000;
30. color: white;
31. }
32. .fa-google {
33. background: #dd4b39;
34. color: white;
35. }
36. .fa-snapchat-ghost {
37. background: #fffc00;
38. color: white;
39. text-shadow: -1px 0 black, 0 1px black, 1px 0 black, 0 -1px black;
40. }
41. .fa-skype {
42. background: #00aff0;
43. color: white;
44. }
45. .fa:hover {
46. opacity: 0.9;
47. }

**Step 4:** When we added the above code successfully, then we have to link the css file in the <head> tag.

1. **<link** rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css"**>**

**Step 5:** And, then we have to use the following code in the [<body> tag](https://www.javatpoint.com/html-body-tag) of Html document.

1. **<a** href="#" class="fa fa-facebook"**></a>**
2. **<a** href="#" class="fa fa-twitter"**></a>**
3. **<a** href="#" class="fa fa-pinterest"**></a>**
4. **<a** href="#" class="fa fa-linkedin"**></a>**
5. **<a** href="#" class="fa fa-instagram"**></a>**
6. **<a** href="#" class="fa fa-youtube"**></a>**
7. **<a** href="#" class="fa fa-google"**></a>**
8. **<a** href="#" class="fa fa-snapchat-ghost"**></a>**
9. **<a** href="#" class="fa fa-skype"**></a>**

**Step 6:** And, at last, we have to save the Html file and then run the file in the browser.

1. <!Doctype Html**>**
2. **<Html>**
3. **<Head>**
4. **<link** rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css"**>**
5. **<Title>**
6. Add the icons of social media
7. **</Title>**
8. **<style>**
9. .fa {
10. padding: 20px;
11. text-align: center;
12. margin: 5px 2px;
13. font-size: 30px;
14. width: 50px;
15. }
16. .fa-facebook {
17. background: #3B5998;
18. color: white;
19. }
20. .fa-twitter {
21. background: #55ACEE;
22. color: white;
23. }
24. .fa-pinterest {
25. background: #cb2027;
26. color: white;
27. }
28. .fa-linkedin {
29. background: #007bb5;
30. color: white;
31. }
32. .fa-instagram {
33. background: #125688;
34. color: white;
35. }
36. .fa-youtube {
37. background: #bb0000;
38. color: white;
39. }
40. .fa-google {
41. background: #dd4b39;
42. color: white;
43. }
44. .fa-snapchat-ghost {
45. background: #fffc00;
46. color: white;
47. text-shadow: -1px 0 black, 0 1px black, 1px 0 black, 0 -1px black;
48. }
49. .fa-skype {
50. background: #00aff0;
51. color: white;
52. }
53. .fa:hover {
54. opacity: 0.9;
55. }
56. **</style>**
57. **</Head>**
58. **<Body>**
59. Hello User!... **<br>**
60. You are at IHub Site... **<br>**
61. Following are the icons of social media
62. **</br>**
63. **<ul>**
64. **<li>**
65. **<a** href="#" class="fa fa-facebook"**>** **</a>**
66. **</li>**
67. **<li>**
68. **<a** href="#" class="fa fa-twitter"**>** **</a>**
69. **</li>**
70. **<li>**
71. **<a** href="#" class="fa fa-pinterest"**>** **</a>**
72. **</li>**
73. **<li>**
74. **<a** href="#" class="fa fa-linkedin"**>** **</a>**
75. **</li>**
76. **<li>**
77. **<a** href="#" class="fa fa-instagram"**>** **</a>**
78. **</li>**
79. **<li>**
80. **<a** href="#" class="fa fa-youtube"**>** **</a>**
81. **</li>**
82. **<li>**
83. **<a** href="#" class="fa fa-google"**>** **</a>**
84. **</li>**
85. **<li>**
86. **<a** href="#" class="fa fa-snapchat-ghost"**>** **</a>**
87. **</li>**
88. **<li>**
89. **<a** href="#" class="fa fa-skype"**>** **</a></li></ul></Body>** **</Html>**

## Add Sticky Social Media Icons

If we want to add the sticky social media icons in the Html document using the Internal [CSS](https://www.javatpoint.com/css-tutorial) and Html code, to show the icons on the web page then we have to follow the steps which are given below. Using these simple steps, we can easily add the sticky social media icons.

**Step 1:** Firstly, we have to type the Html code in any text editor or open the existing Html file in the text editor in which we want add the sticky icons of social media.

1. <!Doctype Html**>**
2. **<Html>**
3. **<Head>**
4. **<Title>**
5. Add the sticky icons of social media
6. **</Title>**
7. **</Head>**
8. **<Body>**
9. Hello User!...
10. You are at IHub Site...
11. **</Body>**
12. **</Html>**

**Step 2:** Now, we have to place the cursor just after the <title> tag in the <head> tag of the Html document and then attach the CDN link. So, we have to type the following <link> tag in the <head> tag.

1. **<link** rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css"**>**

**Step 3:** Now, we have to design the structure of sticky media icons using the CSS code. So, we have to add the code as shown in the following block just after the [<link> tag](https://www.javatpoint.com/html-link-tag) in the Html document.

1. **<style>**
2. /\* Style the Sticky icons on web page \*/
3. .stickyicon-list {
4. position: fixed;
5. top: 200px;
6. transform: translateY(-50%);
7. }
8. .stickyicon-list a {
9. display: block;
10. text-align: center;
11. padding: 8px;
12. transition: all 0.5s ease;
13. color: white;
14. font-size: 20px;
15. }
16. /\* HOver affect on sticky social media icons \*/
17. .stickyicon-list a:hover {
18. color: #000;
19. width:10px;
20. }
22. /\* Now we have to design each icon of social media \*/
23. .facebook {
24. background: #3b5998;
25. color: white;
26. }
27. .twitter {
28. color: white;
29. background: #00acee;
30. }
31. .linkedin {
32. background: #0e76a8;
33. color: white;
34. }
35. .google {
36. color: white;
37. background: #db4a39;
38. }
39. .instagram {
40. background: #3f729b;
41. color: white;
42. }
43. .youtube {
44. color: white;
45. background: #c4302b;
46. }
47. **</style>**

**Step 4:** Now, we have to add the following code in the <body> tag.

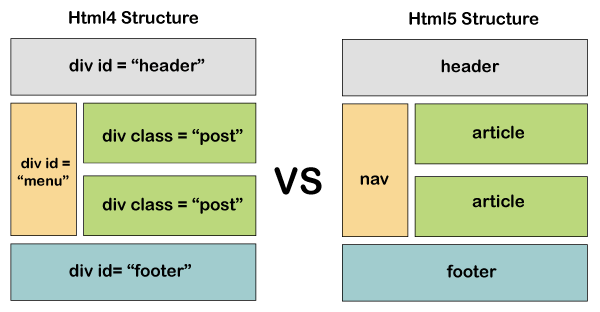
1. **<div** class="stickyicon-list"**>**
2. **<a** href="#facebook" class="facebook"**>**
3. **<i** class="fa fa-facebook"**></i>**
4. **</a>**
5. **<a** href="#twitter" class="twitter"**>**
6. **<i** class="fa fa-twitter"**></i>**
7. **</a>**
8. **<a** href="#linkedin" class="linkedin"**>**
9. **<i** class="fa fa-linkedin"**></i>**
10. **</a>**
11. **<a** href="#google" class="google"**>**
12. **<i** class="fa fa-google"**></i>**
13. **</a>**
14. **<a** href="#instagram" class="instagram"**>**
15. **<i** class="fa fa-instagram"**></i>**
16. **</a>**
17. **<a** href="#youtube" class="youtube"**>**
18. **<i** class="fa fa-youtube"**></i>**
19. **</a>**
20. **</div>**

**Step 5:** And, at last, we have to save the Html file and then run the file in the browser.

1. <!Doctype Html**>**
2. **<Html>**
3. **<Head>**
4. **<link** rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css"**>**
5. **<Title>**
6. Add the Sticky icons of social media
7. **</Title>**
8. **<style>**
9. /\* Style the Sticky icons on web page \*/
10. .stickyicon-list {
11. position: fixed;
12. top: 200px;
13. transform: translateY(-50%);
14. }
15. .stickyicon-list a {
16. display: block;
17. text-align: center;
18. padding: 8px;
19. transition: all 0.5s ease;
20. color: white;
21. font-size: 20px;
22. }
23. /\* HOver affect on sticky social media icons \*/
24. .stickyicon-list a:hover {
25. color: #000;
26. width:10px;
27. }
29. /\* Now we have to design each icon of social media \*/
30. .facebook {
31. background: #3b5998;
32. color: white;
33. }
34. .twitter {
35. color: white;
36. background: #00acee;
37. }
38. .linkedin {
39. background: #0e76a8;
40. color: white;
41. }
42. .google {
43. color: white;
44. background: #db4a39;
45. }
46. .instagram {
47. background: #3f729b;
48. color: white;
49. }
50. .youtube {
51. color: white;
52. background: #c4302b;
53. }
54. **</style>**
55. **</Head>**
56. **<Body>**
57. Hello User!... **<br>**
58. You are at IHub Site... **<br>**
59. **<div** class="stickyicon-list"**>**
60. **<a** href="#facebook" class="facebook"**>**
61. **<i** class="fa fa-facebook"**></i>**
62. **</a>**
63. **<a** href="#twitter" class="twitter"**>**
64. **<i** class="fa fa-twitter"**></i>**
65. **</a>**
66. **<a** href="#linkedin" class="linkedin"**>**
67. **<i** class="fa fa-linkedin"**></i>**
68. **</a>**
69. **<a** href="#google" class="google"**>**
70. **<i** class="fa fa-google"**></i>**
71. **</a>**
72. **<a** href="#instagram" class="instagram"**>**
73. **<i** class="fa fa-instagram"**></i>**
74. **</a>**
75. **<a** href="#youtube" class="youtube"**>**
76. **<i** class="fa fa-youtube"**></i>**
77. **</a>**
78. **</div>**
79. **</Body>**
80. **</Html>**

# **Difference between HTML and HTML5?**

HTML5 is more complete and easier than **HTML4**, it has lots of new tags like [**<header>**](https://www.javatpoint.com/html-header-tag)**,**[**<footer>**](https://www.javatpoint.com/html-footer-tag)**, <navy>,**[**<Audio>**](https://www.javatpoint.com/html-audio)**,**[**<video>**](https://www.javatpoint.com/html-video)**,**[**<main>**](https://www.javatpoint.com/html-main-tag) etc. It also supports graphics. In the following image, we have described all the essential terms related to HTML and HTML5.



HTML is referred to as the **primary** language of the [**World Wide Web**](https://www.javatpoint.com/what-is-world-wide-web)**. HTML** has many updates over time, and the latest [**HTML**](https://www.javatpoint.com/html-tutorial) version is [**HTML5**](https://www.javatpoint.com/html5-tutorial). There are some differences between the two versions:

* HTML5 supports both **audio** and **video** while none of them were part of
* HTML cannot allow JavaScript to run within the web browser, while **HTML5** provides full support for running JavaScript.
* In **HTML5**, inline **mathML** and **SVG** can be used in a text, while in HTML it is not possible.
* HTML5 supports new types of form controls, such as **date** and **time, email, number, category, title, Url, search, etc.**
* Many elements have been introduced in HTML5. Some of the most important are **time, audio, description, embed, fig, shape, footer, article, canvas, navy, output, section, source, track, video**, etc.

## Difference between Html and Html5

|  |  |  |
| --- | --- | --- |
| **Features** | **Html** | **Html5** |
| definition | A hypertext markup language (HTML) is the primary language for developing web pages. | HTML5 is a new version of HTML with new functionalities with markup language with Internet technologies. |
| Multimedia support | Language in **HTML** does not have support for video and audio. | **HTML5** supports both video and audio. |
| Storage | The HTML browser uses cache memory as temporary storage. | HTML5 has the storage options like:**application cache, SQL database,** and **web storage**. |
| Browser compatibility | HTML is compatible with almost all browsers because it has been present for a long time, and the browser made modifications to support all the features. | In HTML5, we have many new tags, elements, and some tags that have been **removed/modified**, so only some browsers are fully compatible with **HTML5**. |
| Graphics support | In HTML, vector graphics are possible with tools Like**Silver light, Adobe Flash, VML,** etc. | In HTML5, vector graphics are supported by default. |
| Threading | In HTML, the browser interface and JavaScript running in the same thread. | The HTML5 has the JavaScript Web Worker API, which allows the browser interface to run in multiple threads. |
| Storage | Uses cookies to store data. | Uses local storage instead of cookies |
| Vector and Graphics | Vector graphics are possible with the help of technologies like **VML, Silverlight, Flash,etc**. | Vector graphics is an integral part of **HTML5, SVG** and **canvas**. |
| Shapes | It is not possible to create shapes like **circles, rectangles, triangles**. | We can draw shapes like **circles, rectangles, triangles**. |
| Doc type | Doctype declaration in html is too long <! DOCTYPE HTML PUBLIC "- // W3C // DTD HTML 4.01 // EN" "http://www.w3.org/TR/html4/strict.dtd"> | The DOCTYPE declaration in html5 is very simple "<! DOCTYPE html> |
| Character Encoding | Character encoding in HTML is too long. <! DOCTYPE HTML PUBLIC "- // W3C // DTD HTML 4.0 Transitional // EN"> | Character encoding declaration is simple <meta charset = "UTF-8"> |
| Multimedia support | Audio and video are not the part of HTML4. | Audio and video are essential parts of HTML5,like: **<Audio>, <Video>**. |
| Vector Graphics | In HTML4, vector graphics are possible with the help of techniques like VML, Silver light and Flash. | Vector graphics are an integral part of **HTML5, SVG**, and **canvas**. |
|  | Html5 uses cookies. | It supplies local storage in place of cookies. |
| Shapes | It is not possible to draw shapes like circles, rectangles, triangles. | Using html5, you can draw shapes like **circles, rectangles, triangles**. |
| Browser Support | Works with all older browsers | A new browser supports this. |

# **mailto HTML Tag**

A **mailto** link is the type of HTML hyperlink that redirects the programmer to a default mail client with a predefined recipient address. When the user clicks on the **mailto** to generate a link, the default mail client opens on the user's computer to send an email. The mail client has predefined parameters such as Cc, Bcc, subject and body content, that are used to send email to one or more recipients. For example, if your system has Microsoft Outlook, Google, Yahoo! Mail, Apple Mail, Thunderbird and other installed accounts, the mail client redirects it to you as you press a mailto link on the browser.



### Syntax

1. **<a** href= "mailto:john@example.com ? {subject} = {subject}" **>** Click message **</a>**

## Creating a mailto tag in HTML

To create a mailto link in [HTML](https://www.javatpoint.com/html-tutorial), we need to use the [HTML <a> tag](https://www.javatpoint.com/html-anchor) with its **href** [attributes](https://www.javatpoint.com/html-attributes), and then use the mailto link.

Let's create a simple mailto program in the HTML.

**mail.html**

1. **<html>**
2. **<head>**
3. **<title>** Use of MailTo tag in HTML **</title>**
4. **<body>**
5. **<a** href="mailto:john@example.com? subject=Learn about MailTo tags in HTML"**>**
6. Click to send email
7. **</a>**
8. **</body>**
9. **</head>**
10. **</html>**

In the above example, when we run the **mail.html** file on the Chrome browser, it shows a **Click to send email** link. After that, as we click on the link, it redirects us to the installed mail account to send an email, as the above image shows.

If we want to send an email to more than one user, we need to separate each address with a **comma** or & (**Ampersand**) symbol, as shown in the below syntax of the mailto tag.

1. **<a** href = "mailto: {email address} ?{subject}= {subject} & cc ={cc addresses} & body = {body}" **>** {clickable text link or image} **</a>**

In the above syntax, the **mailto** attributes can be arranged in any order, but ensure that the {**email address**) is first, and then a question mark directly follows it.

### **Properties of the mailto HTML Tag**

Following are the various properties of the **mailto** [html tag](https://www.javatpoint.com/html-tags).

1. **mailto:** It is the main parameter that specifies the recipient's **email address,** including optional parameters such as CC (**Carbon copy**), BCC (**Blind Carbon Copy**), the subject and body of the message.
2. **Email address:** It is the main parameter of the mailto tag that defines the address of the recipient mail field.
3. **Cc:** It is an optional parameter of the **mailto** tag that holds the address of another mail and is received by the recipient as a carbon copy of the mail.
4. **Bcc:** It is an optional parameter of the mailto tag that holds the specific address of another mail and is received by the recipient as a blind carbon copy of the mail.
5. **Subject:** It is an optional parameter of the mailto tag. It is used to write the subject of the mail.
6. **Body:** It is an optional parameter of the mailto tag. It is used to fill or write the content for mail.
7. **?:** It is an optional parameter of the mailto tag that holds the first parameter of the delimiter.
8. **@:** It is also an optional parameter of the mailto tag that holds other delimiter parameters.
9. **Clickable text or image:** It is the HTML hyperlink field that is displayed to the user on the browser.

**Example 2:** Write a program to add multiple user addresses with subject and body using mailto tag in HTML to send an email.

**mail2.html**

1. **<html>**
2. **<head>**
3. **<title>** Use of MailTo tag in HTML **</title>**
4. **<body>**
5. **<a** href="mailto:john@example.com,jtp@gmail.com, iHub@gmail.com?
6. subject= Learn how to add multiple mail Addresses using the mailto tag in html&body=Welcome to the IHub.com"**>**
7. Give feedback
8. **</a>**
9. **</body>**
10. **</head>**
11. **</html>**

**Example 3**: Write a program to automatically populate CC, BCC, subject and body of mail using mailto tag in html.

**mailto.html**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**
5. Welcome to IHub **</title>**
6. **</head>**
8. **<body** align = "center"**>**
9. **<h1>** IHub **</h1>** **<br>**
10. **<b>** **<h2>** Use of mailto tag in html **</h2>** **</b>**
12. **<a** href = "mailto:abc@gmail.com?
13. cc=xyz@outlook.com&
14. bcc=abc@example.com
15. &subject= Learn Computer Science related technologies.&
16. body=IHub is a platform that facilitates better learning on various technologies related to Computer Science.
17. We assure you that you will not face any problem with our tutorials. But if there is any mistake, please post the problem in our contact form."**>**
18. Send your queries through mail
19. **</a>**
20. **</body>**
21. **</html>**

**Example 4:** Create a custom layout for sending an email using the mailto tag in the html.

**mail5.html**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Using mailto link**</title>**
5. **<style>**
6. h1 {
7. color: brown;
8. }
9. h3{
10. color: green
11. }
12. .container {
13. width: 600px;
14. border: 2px solid black;
15. padding: 25px;
17. }
18. **</style>**
19. **</head>**
21. **<body** align = center**>**
22. **<div** class="container"**>**
23. **<h1>**IHUB.COM**</h1>**
24. **<b>** **<h3>** Using mailto link **</h3>** **</b>**
25. **<form** method="POST"
26. action="mailto: iHub@gmail.com? Subject= Feedback/ Suggestion"
27. enctype="multipart/form-data"**>**
28. **<div** class="control"**>**
29. Name:
30. **<input** aria-required=""
31. id="name"
32. type="text" **/>**
33. **</div>**
34. **<br>**
35. **<div** class="control"**>**
36. Mob no:
37. **<input** aria-required=""
38. id="mobile\_number"
39. type="tel" **/>**
40. **</div>**
41. **<br>**
42. **<div** class="control"**>**
43. Suggestion/ Feedback:
44. **<br>**
45. **<textarea** rows="8" cols="25"
46. name="comment"**>**
48. **</textarea>**
49. **</div>**
50. **<br>**
51. **<div** class="control"**>**
52. **<input** type="submit"
53. value="Submit" **/>**
54. **</div>**
56. **</form>**
57. **</div>**
58. **</body>**
59. **</html>**

# **Download pdf file using HTML**

The general syntax of <a> tag is given below:

### **Syntax:**

1. **<a** href ="Document URL" attributes-list**>** Link Text **</a>**

In above syntax,

**<a>** tag is used for creating link,

**Href** specifies the location of the document, where the link will go.

**Types of links in html:**

* **Absolute hyperlink**: it is also called an external hyperlink. The absolute hyperlink uses a complete URL, i.e., the full website address. When we want to link our page to any other website on the web, we need to provide the full website address of the webpage. Such a type of address is called an absolute hyperlink.  
  **Example:**  
  <a href ="http://www.microsoft.com" attributes-list> Link Text </a>
* **Relative hyperlink**: It is also called an internal hyperlink. A relative URL points to a file within a web site. Relative links make pages, searching down all links and changing their names. A relative link is based on the fact that the server knows the location of the linked document.  
  **Example:**  
  <a href ="about.html" attributes-list> Link Text </a>
* <! DOCTYPE html**>**
* **<html>**
* **<meta** name="viewport" content="width=device-width, initial-scale=1"**>**
* **<head>**
* **<title>**
* Download image using **<a>** tag download attribute
* **</title>**
* **<style>**
* h1 {
* color: green;
* }
* h5 {
* color: green;
* }
* h2 {
* color: green;
* }
* p {
* color: red;
* }
* **</style>**
* **</head>**
* **<body>**
* **<center>**
* **<h2>** Example 1 **</h2>**
* **<h1>**download attribute used to download image**</h1>**
* **<h5>**Click on the image to download it :**</h5>**
* **<a** href="logo 2.png" download **>**
* **<img** src="logo 2.png" alt="javaTpoint" width="104" height="142"**>**
* **</a>**
* **</center>**
* **</body>**
* **</html>**

# **HTML API: (Will learn once JavaScript is done)**

# **HTML Drag and Drop**

**HTML Drag and Drop** (DnD) is a feature of HTML5. It is a powerful user interface concept which *is used to copy, reorder and delete items with the help of mouse*. You can hold the mouse button down over an element and drag it to another location. If you want to drop the element there, just release the mouse button.

If you want to achieve the Drag and Drop functionality in traditional HTML4, you must either have to use complex JavaScript programming or other JavaScript frameworks like jQuery etc.

## Events for Drag and Drop feature

|  |  |
| --- | --- |
| **Event** | **Description** |
| Drag | It fires every time when the mouse is moved while the object is being dragged. |
| Dragstart | It is a very initial stage. It fires when the user starts dragging object. |
| Dragenter | It fires when the user moves his/her mouse cursur over the target element. |
| Dragover | This event is fired when the mouse moves over an element. |
| Dragleave | This event is fired when the mouse leaves an element. |
| Drop | Drop It fires at the end of the drag operation. |
| Dragend | It fires when user releases the mouse button to complete the drag operation. |

## HTML5 Drag and Drop Example

Let's see an example of HTML 5 drag and drop feature.

#### **To understand this example, you must have the knowledge of JavaScript.**

1. **<script>**
2. function allowDrop(ev) {ev.preventDefault();}
3. function drag(ev) {ev.dataTransfer.setData("text/html", ev.target.id);}
4. function drop(ev) {
5. ev.preventDefault();
6. var data = ev.dataTransfer.getData("text/html");
7. ev.target.appendChild(document.getElementById(data));
8. }
9. **</script>**
10. **<p>**Drag the iHub image into the rectangle:**</p>**
11. **<div** id="div1" style="width:350px;height:100px;padding:10px;border:1px solid #aaaaaa;"
12. ondrop="drop(event)" ondragover="allowDrop(event)"**></div>**
13. **<br>**
14. **<img** id="drag1" src="/htmlpages/images/welcome.png" alt="welcome image"
15. draggable="true" ondragstart="drag(event)"**/>**

In the above example, we have used **ondrop and ondragover events on div** element, and **ondragstart event on img** tag.

#### **Note: MouseEvent is not fired during drag operation.**

## Stages during Drag and Drop operations

**1) Make an element draggable**

If you want to make an element draggable, set the draggable attribute to "true" on the element. For example:

1. **<img** draggable = "true"**>**

**2) What to drag:**

Use ondragstart and setData () methods.

Specify what should happen when the element is dragged.

**3) Where to Drop:**

Use ondragover event.

**4) Do the Drop:**

Use ondrop event.

# **HTML5 Geolocation**

The Geolocation is one of the best HTML5 API which is used to identify the user's geographic location for the web application.

This new feature of HTML5 allows you to navigate the latitude and longitude coordinates of the current website's visitor. These coordinates can be captured by JavaScript and send to the server which can show your current location on the website

Most of the geolocation services use Network routing addresses such as IP addresses, RFID, WIFI and MAC addresses or internal GPS devices to identify the user's location.

#### **Tips: To completely understand the concept of Geolocation API you must have some knowledge of JavaScript.**

## User privacy:

The user's location is the privacy concern, so geolocation API protects the user's privacy by taking the user's permission before getting the location. Geolocation API sends a notification prompt box which user can allow or deny, and if the user allows then only his location will be identified.

#### **Note: Your browser must support the geolocation to use it for the web application. Although most of the browsers and mobile devices support the Geolocation API, and this API is only available for HTTPS request.**

## Geolocation object

The Geolocation API is work with the navigation.geolocation object. Its read-only property returns a Geolocation object which identifies the location of the user and can generate a customized result based on user location.

### **Syntax:**

1. geo=navigator. geolocation;

If this object is present, then you can get the geolocation services.

## Geolocation Methods

The Geolocation API uses three methods of Geolocation interface which are given following:

|  |  |
| --- | --- |
| **Methods** | **Description** |
| getCurrentPosition() | It identifies the device or the user's current location and returns a position object with data. |
| watchPosition() | Return a value whenever the device location changes. |
| clearWatch() | It cancels the previous watchPosition() call |

## Checking for browser support:

The geolocation property of navigator.geolcation object helps to determine the browser support for the Geolocation API.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Geolocation API**</title>**
5. **</head>**
6. **<body>**
7. **<h1>**Find your Current location**</h1>**
8. **<button** onclick="getlocation()"**>**Click me**</button>**
9. **<div** id="location"**></div>**
10. **<script>**
11. var x= document.getElementById("location");
13. function getlocation() {
14. if(navigator.geolocation){
15. alert("your browser is supporting Geolocation API")
16. }
17. else
18. {
19. alert("Sorry! your browser is not supporting")
20. }
21. }
22. **</script>**
23. **</body>**
24. **</html>**

## Getting the User's current position:

To get the user's current location, getCurrentPosition() method of the navigator.geolocation object is used. This method accepts three parameters:

* **success:** A success callback function to get the location of the user
* **error:** An error callback function which takes "Position Error" object as input.
* **options:** It defines various options for getting the location.

The below example will return the longitude and latitude of the visitor's current location.

### **Example**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Geolocation API**</title>**
5. **</head>**
6. **<body>**
7. **<h1>**Find your Current location**</h1>**
8. **<button** onclick="getlocation()"**>**Click me**</button>**
9. **<div** id="location"**></div>**
10. **<script>**
11. var x= document.getElementById("location");
12. function getlocation() {
13. if(navigator.geolocation){
14. navigator.geolocation.getCurrentPosition(showPosition)
15. }
16. else
17. {
18. alert("Sorry! your browser is not supporting")
19. } }
21. function showPosition(position){
22. var x = "Your current location is (" + "Latitude: " + position.coords.latitude + ", " + "Longitude: " +    position.coords.longitude + ")";
23. document.getElementById("location").innerHTML = x;
24. }
25. **</script>**
26. **</body>**
27. **</html>**

### **Explanation:**

* First checking the browser support
* Getting current position with getCurrentPosition()
* Getting latitude and longitude values with showPosition() method which is call back method of getCurrentPosition().

## Handling Errors and Rejections: Using an Error callback function

The second parameter of getCurrentPosition is an error Callback function. It is an optional parameter and used to handle errors and user rejection while getting the user's location.

Following are the possible options for invoking the error call back function:

* Unknown random error Occurred
* If the user has denied for sharing location
* Location information is not available
* Request for location is timed-out.

### **Example**

1. function showError(error) {
2. switch(error.code){
3. case error.PERMISSION\_DENIED:
4. alert("User denied the request for Geolocation API.");
5. break;
6. case error.POSITION\_UNAVAILABLE:
7. alert("USer location information is unavailable.");
8. break;
9. case error.TIMEOUT:
10. alert("The request to get user location timed out.");
11. break;
12. case error.UNKNOWN\_ERROR:
13. alert("An unknown error occurred.");
14. break;
15. }
16. }

## Displaying location on Google Map

Till now, we have seen how to show your location using latitude and longitude values, but it is not sufficient. Hence we can also show the exact location on Google map with this API.

Following example showing the location using Google Map.

### **Example**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Geolocation API**</title>**
5. **</head>**
6. **<body>**
7. **<h2>**Find Your Location in below Map**</h2>**
8. **<button** onclick="getlocation();"**>** Show Position**</button>**
9. **<div** id="demo" style="width: 600px; height: 400px; margin-left: 200px;"**></div>**
11. **<script** src="https://maps.google.com/maps/api/js?sensor=false"**>** **</script>**
13. **<script** type="text/javascript"**>**
14. function getlocation(){
15. if(navigator.geolocation){
16. navigator.geolocation.getCurrentPosition(showPos, showErr);
17. }
18. else{
19. alert("Sorry! your Browser does not support Geolocation API")
20. }
21. }
22. //Showing Current Poistion on Google Map
23. function showPos(position){
24. latt = position.coords.latitude;
25. long = position.coords.longitude;
26. var lattlong = new google.maps.LatLng(latt, long);
27. var myOptions = {
28. center: lattlong,
29. zoom: 15,
30. mapTypeControl: true,
31. navigationControlOptions: {style:google.maps.NavigationControlStyle.SMALL}
32. }
33. var maps = new google.maps.Map(document.getElementById("demo"), myOptions);
34. var markers =
35. new google.maps.Marker({position:lattlong, map:maps, title:"You are here!"});
36. }
38. //Handling Error and Rejection
39. function showErr(error) {
40. switch(error.code){
41. case error.PERMISSION\_DENIED:
42. alert("User denied the request for Geolocation API.");
43. break;
44. case error.POSITION\_UNAVAILABLE:
45. alert("USer location information is unavailable.");
46. break;
47. case error.TIMEOUT:
48. alert("The request to get user location timed out.");
49. break;
50. case error.UNKNOWN\_ERROR:
51. alert("An unknown error occurred.");
52. break;
53. }
54. }        **</script>**
55. **</body>**
56. **</html>**

To learn more about Google Maps JavaScript API, you can click on the following link:

[https://developers.google.com/maps/documentation/javascript/reference.](https://developers.google.com/maps/documentation/javascript/reference)

## Location properties

The getCurrentPosition() method of Geolocation API returns callback methods which retrieve the user location information. This callback method returns a Position Object which contains all location information and specifies different properties. It always returns latitude and longitude properties, but the following table describes some other properties of Position object.

|  |  |
| --- | --- |
| **Properties** | **Description** |
| coords.latitude | It returns latitude of user location as a decimal number. |
| coords.longitude | It returns longitude of user location as a decimal number. |
| coords.altitude | It returns altitude in meters above the sea level (Only if available). |
| coords.accuracy | It returns the accuracy of the user's position. |
| coords.altitudeAccuracy | It returns the altitude accuracy of user location. (If available) |
| coords.heading | It returns headings as degree clockwise from North. (If available) |
| coords.speed | It returns the speed in meter per seconds. (If available). |
| timestamp | It returns data or time of response. (If available). |

## Watching the current location:

If we want to know the user location while he is moving and want accurate location at every changed position, then it can be achieved by using watchPosition() callback function.

This function has all three parameters which getCurrentPosition() contains.

### **Syntax:**

1. var id = navigator.geolocation.watchPosition(success[, error[, options]])

The watchPosition() method returns an ID that can be used to uniquely identifying the user?s position, and this ID can also be used with clearWatch() method to stop watching the location.

### **Syntax:**

1. navigator.geolocation.clearWatch(id);

# **HTML Web Storage**

The Web Storage is one of the great features of HTML5. With the Web Storage feature, web applications can locally store data within the browser on the client side. It stores data in the form of key/value pair on the browser. Web Storage sometimes also known as DOM storage.

Storing data with the help of web storage is similar to cookies, but it is better and faster than cookies storage.

In compared to cookies Web Storage has Following Advantages:

* Web Storage can use storage space upto 5MB per domain. (The browser software may prompt the user if the space limit is reached).
* It will not send data to the server side, hence it is faster than cookies storage.
* The data stored by local Storage never expires, but cookies data expires after some time or session.
* Web Storage is more secure than cookies.

## Types of Web Storage

There are two types of web storage with different scope and lifetime.

* **Local Storage:** Local Storages uses Windows.localStaorage object which stores data and available for every page. But data persist even if the browser is closed and reopened (Stores data with no Expiration).
* **Session Storage:** Session Storage uses Windows.sessionStorage object which stores data for one session and data will be lost if the window or browser tab will be closed.

#### **Note: For both storage type, web storage data will not be available for different browsers, and Storage size may vary from browser to browser.**

## Browser support for Web Storage

Before learning for web Storage we must check whether our browser is supporting the web Storage or not. So you can check by executing the following code:

1. <!DOCTYPE html**>**
2. **<html>**
3. **<body>**
4. **<div** id="result"**></div>**
5. **<script>**
6. if(typeof(Storage)!=="undefined") {
7. document.getElementById("result").innerHTML = "Hey, Your browser supports the Web Storage.";
8. }
9. else{
10. document.getElementById("result").innerHTML = "Sorry, your browser does not support Web Storage";
11. }
12. **</script>**
13. **</body>**
14. **</html>**

## The localStorage Object

The localStorage object stores data locally within the browser. The data stored by localStroage object does not have any expiration date. Hence the stored data will not be deleted if the browser is closed or reopened.

Each piece of data is stored in simple key-value pairs. The key/values are always stored as String, and can be accessed with localStorage.getItem() and localStorage.setItem() methods.

### **Example:**

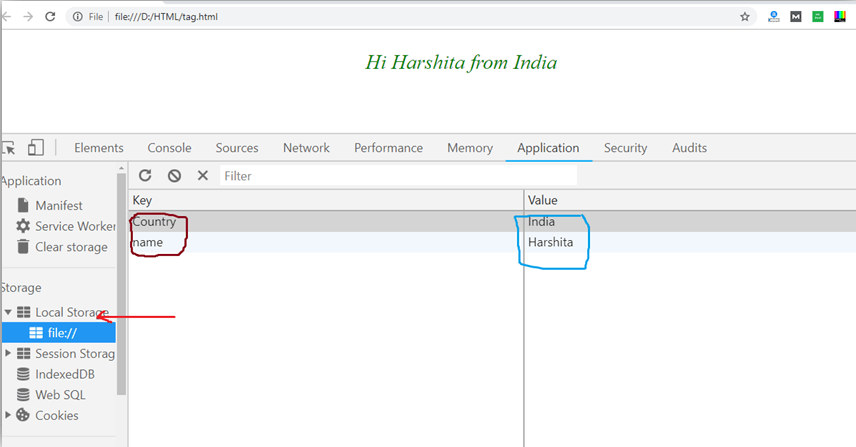
1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Web Storage API**</title>**
5. **<style>**
6. body{
7. color: green;
8. text-align: center;
9. font-size: 30px;
10. margin-top: 30px;
11. font-style: italic;
12. }
13. **</style>**
14. **</head>**
15. **<body>**
16. **<script>**
17. if(typeof(Storage)!=="undefined") {
18. localStorage.setItem("name","Harshita");
19. localStorage.setItem("Country", "India");
20. document.write("Hi"+" "+localStorage.name+" "+"from" +" "+ localStorage.Country);
21. }
22. else{
23. alert("Sorry! your browser is not supporting the browser")
24. }
25. **</script>**
26. **</body>**
27. **</html>**

### **Example Explanation:**

* In the above example, we have used **typeof(Storage)!=="undefined"** to check browser support.
* **localStorage.setItem("name","Harshita")** is used to set the key and value data where "name" is key and "Harshita" is value.
* The **localStorage.name** is used to retrieve the values using key. You can also use another method:  
  **localStorage.getItem** to retrieve the value.

#### **Note: You can check the local storage items in the form of key/value pair by inspecting elements on the web page and then go to the Application option where you will find the local storage and Session storage and can check stored items in the list.**

You can check the following screenshot with key/value pairs.



### **Example 2:**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. div{
6. background-color: pink;
7. height: 50px;
8. }
9. **</style>**
10. **</head>**
11. **<body>**
12. **<h2>**Example of counter Using Local Storage**</h2>**
13. **<button** onclick="counter();"**>**click me**</button>**
14. **<div** id="output"**>**See the result here :**</div>**
15. **<script** type="text/javascript"**>**
16. function counter() {
17. if(localStorage.hits){
18. localStorage.hits=Number(localStorage.hits)+1;
19. }
20. else{
21. localStorage.hits=1;
22. }
23. document.getElementById('output').innerHTML= "You have clicked counter button for"+ " "+ localStorage.hits +" "+"times";
24. }
25. **</script>**
26. **<p>**click the counter button to see the total counts. **</p>**
27. **<p>**If you will close the browser still it will not reset. **</p>**
28. **</body>**
29. **</html>**

### **Example Explanation:**

In the above example, we have shown a counter which will increase as you will click on the counter button.

We have used **localStorage.hits** to set a counter

#### **Note: It will show the total number of count even if you close the browser.**

## The sessionStorage Object:

The sessionStorage object is same as the localStorage object, but the difference is that it stores data only for one session. If you close the browser, then data will be lost or deleted.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. div{
6. background-color: pink;
7. height: 50px;
8. }
9. **</style>**
10. **</head>**
11. **<body>**
12. **<h2>**Example of counter Using Session Storage**</h2>**
13. **<button** onclick="counter();"**>**click me**</button>**
14. **<div** id="output"**>**See the result here:**</div>**
15. **<script** type="text/javascript"**>**
16. function counter() {
17. if(sessionStorage.hits){
18. sessionStorage.hits=Number(sessionStorage.hits)+1;
19. }
20. else{
21. sessionStorage.hits=1;
22. }
23. document.getElementById('output').innerHTML= "You have clicked counter button for"+ " "+ sessionStorage.hits +" "+"times";
24. }
25. **</script>**
26. **<p>**Click the counter button to see the total counts. **</p>**
27. **<p>**Now, if you close the browser then it will reset to initial value. **</p>**
28. **</body>**
29. **</html>**

### **Example Explanation:**

The above example is working same as local storage counter example, but the difference is we have used **sessionStorage.hits** for session storage.

Here the counter will reset if you close the browser and it will start from the initial value.

#### **Tips: You can make these examples more attractive and useful by using jQuery with JavaScript.**

## Remove Web Storage:

As we have seen the session storage data will automatically be deleted, when you close the browser but the data saved by local storage will remain in the browser even if you close it.

Hence to delete the local storage data, you need to call two methods:

* **localStorage.removeItem('key'):** If you want to delete the value on a particular key, then you can use the "key," and that value will be deleted.
* **localStorage.clear():** If you want to delete or clear all settings with key/value pair, then you can call this method.

### **Example**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Web Storage API**</title>**
5. **<style>**
6. body{
7. color: green;
8. text-align: center;
9. font-size: 30px;
10. margin-top: 30px;
11. font-style: italic;
12. }
13. **</style>**
14. **</head>**
15. **<body>**
16. **<button** onclick="remove();"**>**Remove item**</button>**
17. **<div** id="output"**></div>**
19. **<script>**
20. if(typeof(Storage)!=="undefined") {
21. localStorage.setItem("name","Harshita");
22. localStorage.setItem("Country", "India");
23. document.getElementById('output').innerHTML= "Hii, my name is"+ " "+ localStorage.name +" "+"and i belongs to"+" "+localStorage.Country;
24. }
25. else{
26. alert("Sorry! your browser is not supporting the browser")
27. }
28. function remove() {
29. localStorage.removeItem("name");
30. document.getElementById('output').innerHTML= "Hii, my name is"+ " "+ localStorage.name +" "+"and i belongs to"+" "+localStorage.Country;
31. }
32. **</script>**
33. **</body>**
34. **</html>**

### **Example Explanation:**

In the above example we have used **localStorage.removeItem("name");** Which will delete the value for the key "name".

You can remove id for a particular key, or you can also remove all data using **localStorage.clear()** method.

# **HTML5 Web Workers**

The Web Workers are the separate JavaScript code which runs in the background of the web page without affecting the user Interface.

## What is Web Worker?

Everyone wants a website or application which work fast and can execute multiple operations simultaneously without affecting the performance of the page. However, sometimes we experience some delay response or degraded performance of page while executing some large operations. So this problem can be solved using the Web Workers.

Web Workers are the multithreaded object which can execute multiple JavaScript in parallel without affecting the performance of the application or webpage.

**Following are some key features of the Web Workers:**

* Web-workers are threaded JavaScript.
* Web-workers are the kernel-level thread.
* Web-workers requires more space and CPU time.
* Web-workers enhances the speed of a website.
* Web-worker executes codes on the client side (not server-side).
* Web worker threads communicate with each other using postMessage() callback method

#### **Tips: Before working with HTML Web Workers you must have knowledge of JavaScript as the Web Worker depends on JavaScript.**

## Types of Web Workers:

In HTML5 Web Workers are of two types:

* **Dedicated Web Workers:**

The dedicated worker can be accessed by only one script which has called it. The dedicated worker thread end as its parent thread ends. Dedicated workers are only used by one or single main thread.

* **Shared Web Workers:**

It can be shared by multiple scripts and can communicate using the port. Shared workers can be accessed by different windows, iframes or workers.

#### **Note: In this section, we will use dedicated Web Workers.**

## Web Workers Browser Support:

Before learning the web Workers first, we need to check about the browser support. So following is the code which checks whether your browser is supporting or not.

### **Example**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**Web Worker API**</title>**
5. **</head>**
6. **<body>**
7. **<h2>**Example to check the browser support of Web Workers**</h2>**
8. **<div** id="supported"**></div>**
9. **<div** id="unsupported"**></div>**
10. **<button** onclick="worker();"**>**click me**</button>**
11. **<script** type="text/javascript"**>**
12. function worker()
13. {
14. if(typeof(Worker)!=="undefined"){
15. document.getElementById("supported").innerHTML="Supporting the browser";
16. }
17. else
18. {
19. document.getElementById("unsupported").innerHTML="Not supporting";}
20. }
21. **</script>**
22. **</body>**
23. **</html>**

## Creation of Web worker file:

To create a Web Worker file we need to create an external JavaScript file.

Here we have created a web worker file for calculating the square of the number. And saved it with the name "worker.js".

Below is the code for **worker.js** file.

1. onmessage =function(event){
2. var num= event.data;
3. var sqr=num\*num;
4. var result="";
5. for(var i=1;i**<**=sqr; i++)
6. {
7. result= "Sqaure result is:"+ " "+i;
8. }
9. postMessage(result);
10. }

## Creating the Web Worker Object:

In above "worker.js" file, we have created the JS file for web Worker now it needs to call on an HTML file. To create the web worker object, you need to call the Worker() constructor.

Following is the syntax to call and create the object of Web Worker:

1. var worker= new Worker('worker.js');

## Sending messages between the Worker thread and main thread:

All the communication of Worker thread depends on the **postMessage()** method and **onmessage** event handler.

## Terminating the Web Worker:

If you want to immediately terminate the currently running worker in the main thread, then you can terminate it by calling the terminate() method of Web Worker. Here is the syntax for web worker termination:

1. worker.terminate();

Web Worker can be terminate in worker thread also by calling the **close()** method.

### **Example**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. .div1{
6. margin-left: 350px;
7. }
8. **</style>**
9. **</head>**
10. **<body>**
11. <!-- Sqaure Output Result -->
12. **<div** class="div1"**>**
13. **<h2>**Example of Web Worker**</h2>**
14. **<label>**Enter the number to find the square**</label>**
15. **<br><input** type="text" name="num" id="num"**><br>**
16. **<br><button** id="submit"**>**Submit**</button>**
17. **<button** id="other"**>**Wait**</button>**
18. **<div** id="text"**></div>**
19. **</div>**
20. **<script** type="text/javascript"**>**
22. document.getElementById("other").onclick=function() {
23. alert("Hey! Web Worker is working, and you can wait for the result.");
24. }
26. //Web-worker Code.....
27. var worker= new Worker("worker.js");
28. worker.onmessage= function(event){
29. document.getElementById("text").innerText= event.data;}
30. document.getElementById("submit").onclick= function(){
31. var num= document.getElementById("num").value;
32. worker.postMessage(num);
33. }
34. **</script>**
35. **<p><b>**Note:Try to enter a big number, and then click on other button. The page will respond properly**</b></p>**
36. **</body>**
37. **</html>**

**Worker.js file:**

1. onmessage=function(event){
2. var num= event.data;
3. var sqr=num\*num;
4. var result="";
5. for(var i=1;i**<**=sqr; i++)
6. {
7. result= "Sqaure result is:"+ " "+i;
8. }
9. postMessage(result);
10. }

### **Example Explanation:**

In the above example in HTML file we have used

* **var worker= new Worker("worker.js");** To create the web Worker object.
* **worker.onmessage= function(event):** It is used to send the message between the main thread and Worker thread.
* **worker.postMessage(num);** This is the method used to communicate between the Worker thread and main thread. Using this method Worker thread return the result to the main thread.

# **HTML5 Server-Sent Event**

The HTML5 server-sent event enables a browser to receive automatic updates and data from a server via HTTP connections.

## What are the Server-Sent Events?

Whenever we perform some event and send it to the server such as by submitting the form to the server. So such type of event which flows from web browser to web-server are called as a client-side events. But if the server sent some updates or information to the browser, then such events are called server-sent events. **Hence A server sent event occurs when the browser automatically updated from the Server**.

The Server-sent events are mono-directional (always come from server to client). Or it may be called as one-way messaging.

## Receiving events from the server

The Server sent event uses the **EventSource** object to receive events from the server. It specifies the URI of the script which generates the events.

### **Example:**

1. if(typeof(EventSource) !== "undefined") {
2. var source = new EventSource("ServerUpdate.php");
3. source.onmessage = function(event) {
4. document.getElementById("output").innerHTML += event.data + "**<br>**";
5. }

### **Code Explanation:**

* **First, create the new EventSource object, and define the URI of the page which sends server updates. Here we have taken ServerUpdate.php** for sending the updates to the web browser.
* Each time when an update occurs from the server, then the onmessage event occurs and print the message on the web page.
* The occurred message can be displayed on div using id "output".

## Check browser support for Server-sent Event

First we need to check the browser support for server-sent event. So to check the browser support for Server-sent event we will check the EventSource object is true or not. If it is true then it will give alert for supporting else it will give alert for not supporting.

### **Example:**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<title>**HTML5 SSE API**</title>**
5. **</head>**
6. **<body>**
7. **<div** id="output"**></div>**
8. **<script** type="text/javascript"**>**
10. if(typeof(EventSource)!=="undefined"){
11. alert("Hey! Your browser is supporting.");
12. }
13. else{
14. alert("Sorry! Your browser is not supporting.");
15. }
16. **</script>**
17. **</body>**
18. **</html>**

## Sending events from the server:

To work with server-sent, we need a server which can send data updates to the web browser. For this, we need to create a file in ASP, PHP or any dynamic language.

Following is the example to show the server updates:

**ServerUpdate.php:**

### **Example:**

1. **<?php**
2. header('Content-Type: text/event-stream');
3. header('Cache-Control: no-cache');
4. /Get the current time of server
5. $time = date('r');
6. echo "data: The Current Server time is: {$time}\n\n";
7. flush();
8. **?>**

### **Code Explanation:**

* In the first line of the code, we have set the "Content-type" header to "text/event-stream". It is required for server-side event standard.
* The second line informs the server to turn off the caching else the server updates may be cached.
* echo "data: The Current Server time is: {$time}\n\n"; It creates the output of data to send, and it must always start with data: .
* Then, we have used the flush () method, which makes sure that data is sent right away to the web page.

## Complete Example:

### **Example:**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style** type="text/css"**>**
5. div{
6. text-align: center;
7. background-color: #98f5ff;
8. }
9. **</style>**
10. **</head>**
11. **<body>**
13. **<h1** align="center"**>**Dynamic Server Updates**</h1>**
14. **<div** id="output"**></div>**
15. **<script>**
16. if(typeof(EventSource) !== "undefined") {
17. var source = new EventSource("ServerUpdate.php");
18. source.onmessage = function(event) {
19. document.getElementById("output").innerHTML += event.data + "**<br>**";
20. }
21. } else {
22. alert("Sorry, your browser does not support the server sent updates");}
23. **</script>**
24. **</body>**
25. **</html>**

#### **Note: To execute the above code on your browser, you need a server installed on your system, and then run this on localhost. You can install any server such as MYSQL, XAMPP, etc.**