HTML stands for **H**yper**t**ext **M**arkup **L**anguage, and it is the most widely used language to write Web Pages.

* **Hypertext** refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a webpage is called Hypertext.
* As its name suggests, HTML is a **Markup Language** which means you use HTML to simply "mark-up" a text document with tags that tell a Web browser how to structure it to display.

Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers.

Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

HTML Tags

|  |  |
| --- | --- |
| **Sr.No** | **Tag & Description** |
| 1 | **<!DOCTYPE...>**  This tag defines the document type and HTML version. |
| 2 | **<html>**  This tag encloses the complete HTML document and mainly comprises of document header which is represented by <head>...</head> and document body which is represented by <body>...</body> tags. |
| 3 | **<head>**  This tag represents the document's header which can keep other HTML tags like <title>, <link> etc. |
| 4 | **<title>**  The <title> tag is used inside the <head> tag to mention the document title. |
| 5 | **<body>**  This tag represents the document's body which keeps other HTML tags like <h1>, <div>, <p> etc. |
| 6 | **<h1>**  This tag represents the heading. |
| 7 | **<p>**  This tag represents a paragraph. |

## Heading Tags

You can use different sizes for your headings.

HTML also has six levels of headings, which use the elements **<h1>, <h2>, <h3>, <h4>, <h5>,** and **<h6>**.

## Paragraph Tag

An opening **<p>** and a closing **</p>**. To make different paragraphs.

## Line Break Tag

## <br /> element, anything following it starts from the next line. Empty element, where you do not need opening and closing tags.

## The <br /> tag has a space between the characters br and the forward slash. If you omit this space, older browsers will have trouble rendering the line break, *while if you miss the forward slash character and just use <br> it is not valid in XHTML*.

## Centering Content

## You can use <center> tag to put any content in the center of the page or any table cell.

## Horizontal Lines

## Horizontal lines are used to visually break-up sections of a document i.e. <hr> tag.

## <hr /> tag is an example of the empty element.

* Same description as <br /> tag.

## Preserve Formatting

Sometimes, you want your text to follow the exact format of how it is written in the HTML document. In these cases, you can use the preformatted tag **<pre>**.

## Nonbreaking Spaces

## Suppose you want to use the phrase "12 Angry Men." Here, you would not want a browser to split the "12, Angry" and "Men" across two lines

* Where you do not want the client browser to break text, you should use a nonbreaking space entity **&nbsp;** instead of a normal space.

Example:

<p>An example of this technique appears in the movie "12&nbsp;Angry&nbsp;Men."</p>

Result:

An example of this technique appears in the movie "12 Angry Men."

# **HTML - Elements**

An **HTML element** is defined by a starting tag. If the element contains other content, it ends with a closing tag.

|  |
| --- |
|  |
| **Start Tag** | **Content** | **End Tag** |
| <p> | This is paragraph content. | </p> |
| <h1> | This is heading content. | </h1> |
| <div> | This is division content. | </div> |
| <br /> |  |  |

There are some HTML elements which don't need to be closed, such as **<img.../>**, **<hr />** and **<br />** elements. These are known as **void elements**.

## HTML Tag vs. Element

* An HTML element is defined by a *starting tag*. If the element contains other content, it ends with a *closing tag*.
* For example, **<p>** is starting tag of a paragraph and **</p>** is closing tag of the same paragraph but **<p>This is paragraph</p>** is a paragraph element.

## Nested HTML Elements

## One HTML element inside another HTML element.

# **HTML – Attributes**

## An attribute is used to define the characteristics of an HTML element and is placed inside the element's opening tag. All attributes are made up of two parts − a name and a value.

* The **name** is the property you want to set. For example, the paragraph **<p>** element in the example carries an attribute whose name is **align**, which you can use to indicate the alignment of paragraph on the page.
* The **value** is what you want the value of the property to be set and always put within quotations. The below example shows three possible values of align attribute: **left, center** and **right**.

## Attribute names and attribute values are case-insensitive. However, the World Wide Web Consortium (W3C) recommends lowercase attributes/attribute values in their HTML 4 recommendation.

**Example:**

<p align = "left">This is left aligned</p>

<p align = "center">This is center aligned</p>

<p align = "right">This is right aligned</p>

**Result:**

This is left aligned

This is center aligned

This is right aligned

## Core Attributes

Used on the majority of HTML elements:

* **Id**

Id attribute as a unique identifier.

* **Title**

The title attribute gives a suggested title for the element.

Example:

<h3 title = "Hello HTML!">Titled Heading Tag Example</h3>

Result:

* Titled Heading Tag Example

Now try to bring your cursor over "Titled Heading Tag Example" and you will see that whatever title you used in your code is coming out as a tooltip of the cursor i.e. Hello HTML!

* **Class**

The class attribute is used to associate an element with a style sheet.

For example −

class = "className1 className2 className3"

* **Style**

The style attribute allows you to specify Cascading Style Sheet (CSS) rules within the element.

Example:

<p style = "font-family:arial; color:#FF0000;">Some text...</p>

Result:

Some text...

## Internationalization Attributes

There are three internationalization attributes, which are available for most (although not all) XHTML elements.

* dir
* lang
* xml:lang

### **The dir Attribute**

The **dir** attribute allows you to indicate to the browser about the direction in which the text should flow. The dir attribute can take one of two values, as you can see in the table that follows −

|  |  |
| --- | --- |
| **Value** | **Meaning** |
| ltr | Left to right (the default value) |
| rtl | Right to left (for languages such as Hebrew or Arabic that are read right to left) |

Example:

<!DOCTYPE html>

<html dir = "rtl">

<head>

<title>Display Directions</title>

</head>

<body>

This is how IE 5 renders right-to-left directed text.

</body>

</html>

Result –

.This is how IE 5 renders right-to-left directed text

When *dir* attribute is used within the <html> tag, it determines how text will be presented within the entire document. When used within another tag, it controls the text's direction for just the content of that tag.

### **The lang Attribute**

The **lang** attribute allows you to indicate the main language used in a document, but this attribute was kept in HTML only for backwards compatibility with earlier versions of HTML. This attribute has been replaced by the **xml:lang** attribute in new XHTML documents.

<html lang = "en">

**en** means that the page is using English Language.

## The xml:lang Attribute

The *xml:lang* attribute is the XHTML replacement for the *lang* attribute.

### **Generic Attributes**

Here's a table of some other attributes that are readily usable with many of the HTML tags.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Options** | **Function** |
| align | right, left, center | Horizontally aligns tags |
| valign | top, middle, bottom | Vertically aligns tags within an HTML element. |
| bgcolor | numeric, hexidecimal, RGB values | Places a background color behind an element |
| background | URL | Places a background image behind an element |
| id | User Defined | Names an element for use with Cascading Style Sheets. |
| class | User Defined | Classifies an element for use with Cascading Style Sheets. |
| width | Numeric Value | Specifies the width of tables, images, or table cells. |
| height | Numeric Value | Specifies the height of tables, images, or table cells. |
| title | User Defined | "Pop-up" title of the elements. |

# **HTML - Formatting**

## Bold Text

Anything that appears within **<b>...</b>** element. Eg: **bold**

## Italic Text

Anything that appears within **<i>...</i>** element. Eg: *italicized*

## Underlined Text

Anything that appears within **<u>...</u>** element. Eg: underlined

## Strike Text

Anything that appears within **<strike>...</strike>** element.

Eg: ~~strikethrough~~

## Monospaced Font

The content of a **<tt>...</tt>** element is written in monospaced font. Most of the fonts are known as variable-width fonts because different letters are of different widths (for example, the letter 'm' is wider than the letter 'i'). In a monospaced font, however, each letter has the same width.

Eg: monospaced

## Superscript Text

The content of a **<sup>...</sup>** element is written in superscript; the font size used is the same size as the characters surrounding it but is displayed half a character's height above the other characters.

Eg: Uses a superscript typeface.

## Subscript Text

The content of a **<sub>...</sub>** element is written in subscript; the font size used is the same as the characters surrounding it, but is displayed half a character's height beneath the other characters.

Eg: Uses a subscript typeface.

## Inserted Text

Anything that appears within **<ins>...</ins>** element

Eg: <p>I want to drink <del>cola</del> <ins>wine</ins></p>

Result :

I want to drink  wine

## Deleted Text

Anything that appears within **<del>...</del>** element.

Eg: <p>I want to drink <del>cola</del> <ins>wine</ins></p>

Result :

I want to drink  wine

## Larger Text

The content of the **<big>...</big>** element is displayed one font size larger than the rest of the text surrounding.

<p>The following word uses a <big>big</big> typeface.</p>

Result:

The following word uses a big typeface.

## Smaller Text

The content of the **<small>...</small>** element is displayed one font size smaller than the rest of the text surrounding.

<p>The following word uses a <small>small</small> typeface.</p>

Result:

The following word uses a small typeface.

## Grouping Content

* The **<div>** and **<span>** elements allow you to group together several elements to create sections or subsections of a page.
* The <span> element, on the other hand, can be used to group inline elements only. So, if you have a part of a sentence or paragraph which you want to group together, you could use the <span> element as follows.

# **HTML - Phrase Tags**

## Emphasized Text

Anything that appears within **<em>...</em>** element is displayed as emphasized text.

<p>The following word uses an <em>emphasized</em> typeface.</p>

Result :

The following word uses an emphasized typeface.

## Marked Text

Anything that appears with-in **<mark>...</mark>** element, is displayed as marked with yellow ink.

<p>The following word has been <mark>marked</mark> with yellow</p>

Result:

The following word has been marked with yellow

## Strong Text

Anything that appears within **<strong>...</strong>** element is displayed as important text.

<p>The following word uses a <strong>strong</strong> typeface.</p>

Result:

The following word uses a **strong** typeface.

## Text Abbreviation

You can abbreviate a text by putting it inside opening <abbr> and closing </abbr> tags. If present, the title attribute must contain this full description and nothing else.

<p>My best friend's name is <abbr title = "Abhishek">Abhy</abbr>.</p>

Result:

My best friend's name is Abhy. (It shows Abhishek when you point the mouse)

## Acronym Element

The **<acronym>** element allows you to indicate that the text between <acronym> and </acronym> tags is an acronym.

<p>This chapter covers marking up text in <acronym>XHTML</acronym>.</p>

Result:

This chapter covers marking up text in XHTML. (It shows full form of XHTML when you keep the mouse on that)

## Text Direction

The **<bdo>...</bdo>** element stands for Bi-Directional Override and it is used to override the current text direction.

<p><bdo dir = "rtl">This text will go right to left.</bdo></p>

Result:

.tfel ot …

## Special Terms

The **<dfn>...</dfn>** element (or HTML Definition Element) allows you to specify that you are introducing a special term. It's usage is similar to italic words in the midst of a paragraph.

<p>The following word is a <dfn>special</dfn> term.</p>

Result:

The following word is a special term.

## Quoting Text

When you want to quote a passage from another source, you should put it in between **<blockquote>...</blockquote>** tags.

<p>The following description of XHTML is taken from the W3C Web site:</p>

<blockquote>XHTML 1.0 is the W3C's first Recommendation for XHTML,following on

from earlier work on HTML 4.01, HTML 4.0, HTML 3.2 and HTML 2.0.</blockquote>

Result:

The following description of XHTML is taken from the W3C Web site:

XHTML 1.0 is the W3C's first Recommendation for XHTML, following on from earlier work on HTML 4.01, HTML 4.0, HTML 3.2 and HTML 2.0.

## Short Quotations

The **<q>...</q>** element is used when you want to add a double quote within a sentence.

<p>Amit is in Spain, <q>I think I am wrong</q>.</p>

Result:

Amit is in Spain,”I think I am wrong”.

## Computer Code

Any programming code to appear on a Web page should be placed inside **<code>...</code>** tags.

<p>Regular text. <code>This is code.</code> Regular text.</p>

Result:

Regular text. This is code. Regular text.

## Programming Variables

This element is usually used in conjunction with the **<pre>** and **<code>**elements to indicate that the content of that element is a variable.

<p><code>document.write("<var>user-name</var>")</code></p>

Result:

document.write("user-name")

## Program Output

The **<samp>...</samp>** element indicates sample output from a program, and script etc. Again, it is mainly used when documenting programming or coding concepts.

<p>Result produced by the program is <samp>Hello World!</samp></p>

Result:

Result produced by the program is Hello World!

## Address Text

The **<address>...</address>** element is used to contain any address.

<address>388A, Road No 22, Jubilee Hills - Hyderabad</address>

Result:

*388A, Road No 22, Jubilee Hills – Hyderabad*

# **HTML - Meta Tags**

The **<meta>** tag is used to provide such additional important information about a document in a variety of ways.

Does not have a closing tag but it carries information within its attributes.

The META elements can be used to include name/value pairs describing properties of the HTML document, such as author, expiry date, a list of keywords, document author etc.

## Adding Meta Tags to Your Documents

You can add metadata to your web pages by placing <meta> tags inside the header of the document which is represented by **<head>** and **</head>** tags.

A meta tag can have following attributes in addition to core attributes −

|  |  |
| --- | --- |
| **Sr.No** | **Attribute & Description** |
| 1 | **Name**  Name for the property. Can be anything. Examples include, keywords, description, author, revised, generator etc. |
| 2 | **content**  Specifies the property's value. |
| 3 | **scheme**  Specifies a scheme to interpret the property's value (as declared in the content attribute). |
| 4 | **http-equiv**  Used for http response message headers. For example, http-equiv can be used to refresh the page or to set a cookie. Values include content-type, expires, refresh and set-cookie. |

## Specifying Keywords

You can use <meta> tag to specify important keywords related to the document and later these keywords are used by the search engines while indexing your webpage for searching purpose.

## Document Description

You can use <meta> tag to give a short description about the document.

## Document Revision Date

You can use <meta> tag to give information about when last time the document was updated. This information can be used by various web browsers while refreshing your webpage.

## Page Redirection

You can use <meta> tag to redirect your page to any other webpage. You can also specify a duration if you want to redirect the page after a certain number of seconds.

## Setting Cookies

Cookies are data, stored in small text files on your computer and it is exchanged between web browser and web server to keep track of various information based on your web application need.

You can use <meta> tag to store cookies on client side and later this information can be used by the Web Server to track a site visitor.

**Note** : If you do not include the expiration date and time, the cookie is considered a session cookie and will be deleted when the user exits the browser.

## Setting Author Name

You can set an author name in a web page using meta tag.

## Specify Character Set

You can use <meta> tag to specify character set used within the webpage.

To serve the static page with traditional Chinese characters, the webpage must contain a <meta> tag to set Big5 encoding.

<!DOCTYPE html>

<html>

<head>

<title>Meta Tags Example</title>

<meta name = "keywords" content = "HTML, Meta Tags, Metadata" />

<meta name = "description" content = "Learning about Meta Tags." />

<meta name = "revised" content = "Tutorialspoint, 3/7/2014" />

<meta http-equiv = "refresh" content = "5; url = http://www.tutorialspoint.com" />

<meta http-equiv = "cookie" content = "userid = xyz;

expires = Wednesday, 08-Aug-15 23:59:59 GMT;" />

<meta name = "author" content = "Mahnaz Mohtashim" />

<meta http-equiv = "Content-Type" content = "text/html; charset = UTF-8" />

<meta http-equiv = "Content-Type" content = "text/html; charset = Big5" />

</head>

<body>

<p>Hello HTML5!</p>

</body>

</html>

Result:

Hello HTML5!

# **HTML - Images**

## Insert Image

You can insert any image in your web page by using **<img>** tag.

The **alt** attribute is a mandatory attribute which specifies an alternate text for an image, if the image cannot be displayed.

The <img> tag is an empty tag, which means that, it can contain only list of attributes and it has no closing tag.

<img src = "/html/images/test.png" alt = "Test Image" width = "150" height = "100" border = "3" align = "right"/>

Result :

**It will show the test.png image**

# **HTML - Tables**

The HTML tables are created using the **<table>** tag in which the **<tr>** tag is used to create table rows and **<td>** tag is used to create data cells. The elements under <td> are regular and left aligned by default

## Table Heading

Table heading can be defined using **<th>** tag.

## Cellpadding and Cellspacing Attributes

There are two attributes called *cellpadding* and *cellspacing* which you will use to adjust the white space in your table cells. The cellspacing attribute defines space between table cells, while cellpadding represents the distance between cell borders and the content within a cell.

body>

<table border = "1" cellpadding = "5" cellspacing = "5" >

<tr>

<th>Name</th>

<th>Salary</th>

</tr>

<tr>

<td>Row 1, Column 1</td>

<td>Row 1, Column 2</td>

</tr>

<tr>

<td>Row 2, Column 1</td>

<td>Row 2, Column 2</td>

</tr>

</table>

</body>

Result:

|  |  |
| --- | --- |
| Row 1, Column 1 | Row 1, Column 2 |
| Row 2, Column 1 | Row 2, Column 2 |

If you do not need a border, then you can use border = "0".

## Colspan and Rowspan Attributes

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

<body>

<table border = "1" width = "400" height = "150">

<caption>This is the caption</caption>

<tr>

<th>Column 1</th>

<th>Column 2</th>

<th>Column 3</th>

</tr>

<tr>

<td rowspan = "2">Row 1 Cell 1</td>

<td>Row 1 Cell 2</td>

<td>Row 1 Cell 3</td>

</tr>

<tr>

<td>Row 2 Cell 2</td>

<td>Row 2 Cell 3</td>

</tr>

<tr>

<td colspan = "3">Row 3 Cell 1</td>

</tr>

</table>

</body>

Result:

|  |  |  |
| --- | --- | --- |
| **Column 1** | **Column 2** | **Column 3** |
| Row 1 Cell 1 | Row 1 Cell 2 | Row 1 Cell 3 |
| Row 2 Cell 2 | Row 2 Cell 3 |
| Row 3 Cell 1 | | |

The three elements for separating the head, body, and foot of a table are −

* **<thead>** − to create a separate table header.
* **<tbody>** − to indicate the main body of the table.
* **<tfoot>** − to create a separate table footer.

## Nested Tables

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

# **HTML - Image Links**

We just need to use an image inside hyperlink

<a href = "https://www.tutorialspoint.com" target = "\_self">

<img src = "/images/logo.png" alt = "Tutorials Point" border = "0"/>

</a>

## Mouse-Sensitive Images

There are two ways to create image maps –

* **Server-side image maps** − This is enabled by the **ismap** attribute of the <img> tag and requires access to a server and related image-map processing applications.

Here you simply put your image inside a hyper link and use **ismap** attribute which makes it special image and when the user clicks some place within the image, the browser passes the coordinates of the mouse pointer along with the URL specified in the <a> tag to the web server. The server uses the mouse-pointer coordinates to determine which document to deliver back to the browser.

<a href = "/cgi-bin/ismap.cgi" target = "\_self">

<img ismap src = "/images/logo.png" alt = "Tutorials Point" border = "0"/>

</a>

Then the browser sends the following search parameters to the web server which can be processed by **ismap.cgi** script or **map file** and you can link whatever documents you like to these coordinates −

/cgi-bin/ismap.cgi?20,30

* **Client-side image maps** − This is created with the **usemap** attribute of the <img> tag, along with corresponding <map> and <area> tags.

<body>

<p>Search and click the hotspot</p>

<img src = /images/html.gif alt = "HTML Map" border = "0" usemap = "#html"/>

<!-- Create Mappings -->

<map name = "html">

<area shape = "circle" coords = "80,80,20"

href = "/css/index.htm" alt = "CSS Link" target = "\_self" />

<area shape = "rect" coords = "5,5,40,40" alt = "jQuery Link"

href = "/jquery/index.htm" target = "\_self" />

</map>

</body>

## Coordinate System

The actual value of coords is totally dependent on the shape in question. Here is a summary, to be followed by detailed examples −

* **rect = x1 , y1 , x2 , y2**

x1 and y1 are the coordinates of the upper left corner of the rectangle; x2 and y2 are the coordinates of the lower right corner.

* **circle = xc , yc , radius**

xc and yc are the coordinates of the center of the circle, and radius is the circle's radius. A circle centered at 200,50 with a radius of 25 would have the attribute *coords = "200,50,25"*

* **poly = x1 , y1 , x2 , y2 , x3 , y3 , ... xn , yn**

The various x-y pairs define vertices (points) of the polygon, with a "line" being drawn from one point to the next point. A diamond-shaped polygon with its top point at 20,20 and 40 pixels across at its widest points would have the attribute *coords = "20,20,40,40,20,60,0,40"*.

All coordinates are relative to the upper-left corner of the image (0,0). Each shape has a related URL. You can use any image software to know the coordinates of different positions.

# **HTML - Email Links**

## HTML Email Tag

<a href = "mailto: abc@example.com">Send Email</a>

## Default Settings

You can specify a default *email subject* and *email body* along with your email address. Following is the example to use default subject and body.

<a href = "mailto:abc@example.com?subject = Feedback&body = Message">

Send Feedback

</a>

# **HTML - Frames**

HTML frames are used to divide your browser window into multiple sections where each section can load a separate HTML document. A collection of frames in the browser window is known as a frameset. The window is divided into frames in a similar way the tables are organized: into rows and columns.

**Never recommended to use frames in your webpages**

# **HTML - Iframes**

The **<iframe>** tag defines a rectangular region within the document in which the browser can display a separate document, including scrollbars and borders.

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

<iframe src = "/html/menu.htm" width = "555" height = "200">

Sorry your browser does not support inline frames.

</iframe>

# **HTML - Forms**

The HTML **<form>** tag is used to create an HTML form and it has following syntax −

<form action = "Script URL" method = "GET|POST">

form elements like input, textarea etc.

</form>

Form Attributes

Apart from common attributes, following is a list of the most frequently used form attributes −

|  |  |
| --- | --- |
| **Sr.No** | **Attribute & Description** |
| 1 | **action**  Backend script ready to process your passed data. |
| 2 | **method**  Method to be used to upload data. The most frequently used are GET and POST methods. |
| 3 | **target**  Specify the target window or frame where the result of the script will be displayed. It takes values like \_blank, \_self, \_parent etc. |
| 4 | **enctype**  You can use the enctype attribute to specify how the browser encodes the data before it sends it to the server. Possible values are −  **application/x-www-form-urlencoded** − This is the standard method most forms use in simple scenarios.  **mutlipart/form-data** − This is used when you want to upload binary data in the form of files like image, word file etc. |

## Text Input Controls

**Single-line text input controls** – **<input>**

**Password input controls** − **<input>**

**Multi-line text input controls** − **<textarea>**

First name: <input type = "text" name = "first\_name" />

## Attributes

* type
* name
* value
* size
* maxlength

## Password input controls

Password: <input type = "password" name = "password" />

## Multiple-Line Text Input Controls

<textarea rows = "5" cols = "50" name = "description">

Enter description here...

</textarea>

## Checkbox Control

<form>

<input type = "checkbox" name = "maths" value = "on"> Maths

<input type = "checkbox" name = "physics" value = "on"> Physics

</form>

## Radio Button Control

<form>

<input type = "radio" name = "subject" value = "maths"> Maths

<input type = "radio" name = "subject" value = "physics"> Physics

</form>

## Attributes (Same for Checkbox and Radio button)

* type
* name
* value
* checked

## Select Box Control

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

<form>

<select name = "dropdown">

<option value = "Maths" selected>Maths</option>

<option value = "Physics">Physics</option>

</select>

</form>

Attributes

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

|  |  |
| --- | --- |
| **Sr.No** | **Attribute & Description** |
| 1 | **name**  Used to give a name to the control which is sent to the server to be recognized and get the value. |
| 2 | **size**  This can be used to present a scrolling list box. |
| 3 | **multiple**  If set to "multiple" then allows a user to select multiple items from the menu. |

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

|  |  |
| --- | --- |
| **Sr.No** | **Attribute & Description** |
| 1 | **value**  The value that will be used if an option in the select box box is selected. |
| 2 | **selected**  Specifies that this option should be the initially selected value when the page loads. |
| 3 | **label**  An alternative way of labeling options |

## File Upload Box

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

<form>

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

</form>

## Attributes

* **name**

Used to give a name to the control which is sent to the server to be recognized and get the value.

* **Accept**

Specifies the types of files that the server accepts.

## Button Controls

|  |  |
| --- | --- |
| **Sr.No** | **Type & Description** |
| 1 | **submit**  This creates a button that automatically submits a form. |
| 2 | **reset**  This creates a button that automatically resets form controls to their initial values. |
| 3 | **button**  This creates a button that is used to trigger a client-side script when the user clicks that button. |
| 4 | **image**  This creates a clickable button but we can use an image as background of the button. |

<form>

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

<input type = "reset" name = "reset" value = "Reset" />

<input type = "button" name = "ok" value = "OK" />

<input type = "image" name = "imagebutton" src = "/html/images/logo.png" />

</form>

## Hidden Form Controls

## Hidden form controls are used to hide data inside the page which later on can be pushed to the server. This control hides inside the code and does not appear on the actual page. For example, following hidden form is being used to keep current page number. When a user will click next page then the value of hidden control will be sent to the web server and there it will decide which page will be displayed next based on the passed current page.

<form>

<p>This is page 10</p>

<input type = "hidden" name = "pagename" value = "10" />

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

<input type = "reset" name = "reset" value = "Reset" />

</form>

# **HTML - Embed Multimedia**

The easiest way to add video or sound to your web site is to include the special HTML tag called **<embed>.**

**<noembed>** tag for the browsers which don't recognize the <embed> tag. You could, for example, use <embed> to display a movie of your choice, and **<noembed>** to display a single JPG image if browser does not support <embed> tag.

<body>

<embed src = "/html/yourfile.mid" width = "100%" height = "60" >

<noembed><img src = "yourimage.gif" alt = "Alternative Media" ></noembed>

</embed>

</body>

## The <embed> Tag Attributes

|  |  |
| --- | --- |
| **Sr.No** | **Attribute & Description** |
| 1 | **Align** (Deprecated in HTML5)  Determines how to align the object. It can be set to either center, *left or right*. |
| 2 | **Autostart** (Deprecated in HTML5)  This boolean attribute indicates if the media should start automatically. You can set it either true or false. |
| 3 | **loop**  Specifies if the sound should be played continuously (set loop to true), a certain number of times (a positive value) or not at all (false) |
| 4 | **playcount**  Specifies the number of times to play the sound. This is alternate option for *loop* if you are usiong IE. |
| 5 | **hidden**  Specifies if the multimedia object should be shown on the page. A false value means no and true values means yes. |
| 6 | **width**  Width of the object in pixels |
| 7 | **height**  Height of the object in pixels |
| 8 | **name**  A name used to reference the object. |
| 9 | **src**  URL of the object to be embedded. |
| 10 | **volume**  Controls volume of the sound. Can be from 0 (off) to 100 (full volume). |

## Supported Video Types

You can use various media types like Flash movies (.swf), AVI's (.avi), and MOV's (.mov) file types inside embed tag.

## HTML Object tag

HTML 4 introduces the **<object>** element, which offers an all-purpose solution to generic object inclusion. The **<object>** element allows HTML authors to specify everything required by an object for its presentation by a user agent.

### Example - 1

You can embed an HTML document in an HTML document itself as follows −

<object data = "data/test.htm" type = "text/html" width = "300" height = "200">

alt : <a href = "data/test.htm">test.htm</a>

</object>

### Example - 2

You can embed a PDF document in an HTML document as follows −

<object data = "data/test.pdf" type = "application/pdf" width = "300" height = "200">

alt : <a href = "data/test.pdf">test.htm</a>

</object>

### Example - 3

You can specify some parameters related to the document with the **<param>**tag. Here is an example to embed a wav file −

<object data = "data/test.wav" type = "audio/x-wav" width = "200" height = "20">

<param name = "src" value = "data/test.wav">

<param name = "autoplay" value = "false">

<param name = "autoStart" value = "0">

alt : <a href = "data/test.wav">test.wav</a>

</object>

You can add a flash document,

You can add a Java applet into HTML document.

* The **classid** attribute identifies which version of Java Plug-in to use. You can use the optional *codebase* attribute to specify if and how to download the JRE.

# **HTML - Header**

The <head> tag is a container of various important tags like <title>, <meta>, <link>, <base>, <style>, <script>, and <noscript> tags.

## The HTML <meta> Tag

<!DOCTYPE html>

<html>

<head>

<title>HTML Meta Tag Example</title>

<!-- Provide list of keywords -->

<meta name = "keywords" content = "C, C++, Java, PHP, Perl, Python">

<!-- Provide description of the page -->

<meta name = "description" content = "Simply Easy Learning by Tutorials Point">

<!-- Author information -->

<meta name = "author" content = "Tutorials Point">

<!-- Page content type -->

<meta http-equiv = "content-type" content = "text/html; charset = UTF-8">

<!-- Page refreshing delay -->

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

<!-- Page expiry -->

<meta http-equiv = "expires" content = "Wed, 21 June 2006 14:25:27 GMT">

<!-- Tag to tell robots not to index the content of a page -->

<meta name = "robots" content = "noindex, nofollow">

</head>

<body>

<p>Hello, World!</p>

</body>

</html>

## The HTML <link> Tag

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

You can use CSS in three ways in your HTML document −

* External Style Sheet
* Internal Style Sheet
* Inline Style Sheet

# **HTML - JavaScript**

## External JavaScript

<script src = "/html/script.js" type = "text/javascript"/></script>

## Internal Script

<script type = "text/JavaScript">

function Hello() {

alert("Hello, World");

}

</script>

* **<noscript>** tag if browsers don't support scripts.