Unit I. HTML

What is HTML ?

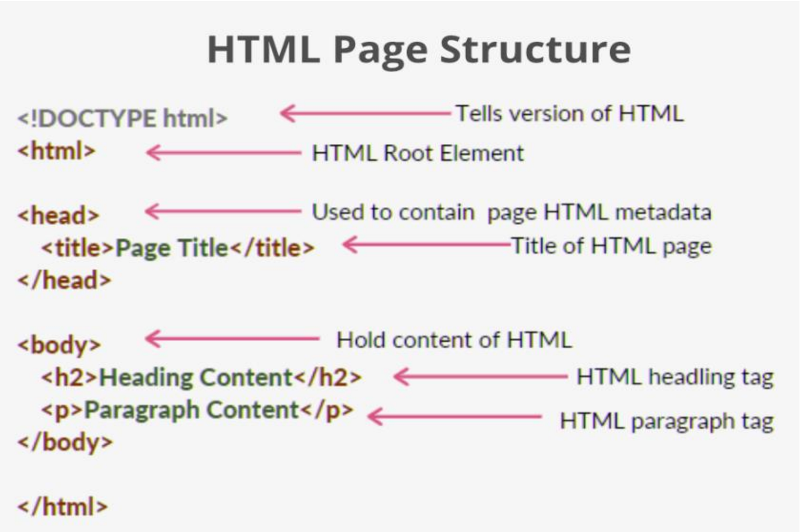
HTML stands for **HyperText Markup Language**. It is used to design web pages using the **markup language**. HTML is the combination of **Hypertext** and **Markup language**. Hypertext defines the link between the web pages and markup language defines the text document within the tag that define the structure of web pages.

What is HTML used for ?

HTML is used to create the structure of web pages that are displayed on the World Wide Web (www). It contains Tags and Attributes that are used to design the web pages. Also, we can link multiple pages using Hyperlinks.

HTML Basic Format Page Structure

The basic structure of an HTML page is laid out below. It contains the essential building-block elements (i.e. doctype declaration, HTML, head, title, and body elements) upon which all web pages are created.



HTML Page Structure

```
<!DOCTYPE html>
<html>

<head>
  <title>Page Title</title>
</head>

<body>
  <h2>Heading Content</h2>
  <p>Paragraph Content</p>
</body>

</html>
```

<DOCTYPE! html> – A doctype or document type declaration is an instruction that tells the web browser about the markup language in which the current page is written. It is not an element or tag. The doctype declaration is not case-sensitive.

<html> – This tag is used to define the root element of HTML document. This tag tells the browser that it is an HTML document. It is the second outer container element that contains all other elements within it.

<head> – This tag is used to define the head portion of the HTML document that contains information related to the document. Elements within the head tag are not visible on the front-end of a webpage.

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

To create an HTML document or a web page. To begin coding HTML you need only two stuff: a simple-text editor and a web browser.

Creating Your First HTML Document

An HTML file that displays "Hello world" message in your web browser.

Step 1: Creating the HTML file

Open up your computer's plain text editor and create a new file.

Tip: We suggest you to use Notepad (on Windows), TextEdit (on Mac) or some other simple text editor to do this; don't use Word or WordPad! Once you understand the basic principles, you may switch to more advanced tools such as Adobe Dreamweaver.

Step 2: Type some HTML code

Start with an empty window and type the following code:

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
   <title>A simple HTML document</title>
</head>
<body>
   <p>Hello World!<p>
</body>
</html>
```

Step 3: Saving the file

Now save the file on your desktop as "myfirstpage.html ".

Note: It is important that the extension .html is specified — some text editors, such as Notepad, will automatically save it as .txt otherwise.

To open the file in a browser. Navigate to your file then double click on it. It will open in your default Web browser. If it does not, open your browser and drag the file to it.

Explanation of code

You might think what that code was all about. Well, let's find out.

- The first line <!DOCTYPE html> is the document type declaration. It instructs the web browser that this document is an HTML5 document. It is case-insensitive.

- The <head> element is a container for the tags that provides information about the document, for example, <title> tag defines the title of the document.
- The <body> element contains the document's actual content (paragraphs, links, images, tables, and so on) that is rendered in the web browser and displayed to the user.

Note: A DOCTYPE declaration appears at the top of a web page before all other elements; however the doctype declaration itself is not an HTML tag. Every HTML document requires a document type declaration to insure that your pages are displayed correctly.

Tip: The <html>, <head>, and <body> tags make up the basic skeleton of every web page. Content inside the <head> and </head> are invisible to users with one exception: the text between <title> and </title> tags which appears as the title on a browser tab.

HTML Tags and Elements

HTML is written in the form of HTML elements consisting of markup tags. These markup tags are the fundamental characteristic of HTML. Every markup tag is composed of a keyword, surrounded by angle brackets, such as <html>, <head>, <body>, <title>, <p>, and so on.

HTML tags normally come in pairs like <html> and </html>. The first tag in a pair is often called the opening tag (or start tag), and the second tag is called the closing tag (or end tag).

An opening tag and a closing tag are identical, except a slash (/) after the opening angle bracket of the closing tag, to tell the browser that the command has been completed.

In between the start and end tags you can place appropriate contents. For example, a paragraph, which is represented by the p element, would be written as:

Example

```
<p>This is a paragraph.</p>
<!-- Paragraph with nested element -->
<p>
   This is <b>another</b> paragraph.
</p>
```

Heading Tags

Any document starts with a heading. 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>**. While displaying any heading, browser adds one line before and one line after that heading.

```
<!DOCTYPE html>
<html>

  <head>
    <title>Heading Example</title>
```

```
   </head>

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

</html>
```

Output

This is heading 1

This is heading 2

This is heading 3

This is heading 4

This is heading 5

This is heading 6

Paragraph Tag

The **<p>** tag offers a way to structure your text into different paragraphs. Each paragraph of text should go in between an opening <p> and a closing </p> tag as shown below in the example −

Example

```
<!DOCTYPE html>
<html>

   <head>
      <title>Paragraph Example</title>
   </head>

   <body>
      <p>Here is a first paragraph of text.</p>
      <p>Here is a second paragraph of text.</p>
      <p>Here is a third paragraph of text.</p>
   </body>

</html>
```

This will produce the following result −

Here is a first paragraph of text.

Here is a second paragraph of text.

Here is a third paragraph of text.

Line Break Tag

Whenever you use the **<br />** element, anything following it starts from the next line. This tag is an example of an **empty** element, where you do not need opening and closing tags, as there is nothing to go in between them.

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.

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Line Break  Example</title>
  </head>

  <body>
    <p>Hello<br />
      You delivered your assignment ontime.<br />
      Thanks<br />
      Mitali</p>
  </body>

</html>
```

This will produce the following result −

Hello
You delivered your assignment on time.
Thanks
Mitali

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>

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 Text Formatting

HTML contains several elements for defining text with a special meaning.

HTML Formatting Elements

Formatting elements were designed to display special types of text:

- <b> - Bold text
- <strong> - Important text
- <i> - Italic text
- <em> - Emphasized text
- <mark> - Marked text
- <small> - Smaller text
- <del> - Deleted text
- <ins> - Inserted text

- <sub> - Subscript text
- <sup> - Superscript text

HTML <b> and <strong> Elements

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

Example

```
<b>This text is bold</b>
```

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

Example

```
<strong>This text is important!</strong>
```

HTML <i> and <em> Elements

The HTML <i> element defines a part of text in an alternate voice or mood. The content inside is typically displayed in italic.

Tip: The <i> tag is often used to indicate a technical term, a phrase from another language, a thought, a ship name, etc.

Example

```
<i>This text is italic</i>
```

The HTML <em> element defines emphasized text. The content inside is typically displayed in italic.

Tip: A screen reader will pronounce the words in <em> with an emphasis, using verbal stress.

Example

```
<em>This text is emphasized</em>
```

HTML <small> Element

The HTML <small> element defines smaller text:

Example

```
<small>This is some smaller text.</small>
```

HTML <mark> Element

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

Example

```
<p>Do not forget to buy <mark>milk</mark> today.</p>
```

HTML <del> Element

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

Example

```
<p>My favorite color is <del>blue</del> red.</p>
```

HTML <ins> Element

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

Example

```
<p>My favorite color is <del>blue</del> <ins>red</ins>.</p>
```

HTML <sub> Element

The HTML <sub> element defines subscript text. Subscript text appears half a character below the normal line, and is sometimes rendered in a smaller font. Subscript text can be used for chemical formulas, like H_2O:

Example

```
<p>This is <sub>subscripted</sub> text.</p>
```

HTML <sup> Element

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

Example

```
<p>This is <sup>superscripted</sup> text.</p>
```

Unordered HTML List

An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.

The list items will be marked with bullets (small black circles) by default:

Example

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

Ordered HTML List

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

The list items will be marked with numbers by default:

Example

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

An unordered HTML list:

- Item
- Item
- Item
- Item

An ordered HTML list:

1. First item
2. Second item
3. Third item
4. Fourth item

HTML Description Lists

HTML also supports description lists.

A description list is a list of terms, with a description of each term.

The <dl> tag defines the description list, the <dt> tag defines the term (name), and the <dd> tag describes each term:

Example

```
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
```

Output

A Description List

Coffee

- black hot drink

Milk

- white cold drink

HTML List Tags

Tag	Description
<ul>	Defines an unordered list
<ol>	Defines an ordered list
<li>	Defines a list item
<dl>	Defines a description list
<dt>	Defines a term in a description list
<dd>	Describes the term in a description list

Unordered HTML List

An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.

The list items will be marked with bullets (small black circles) by default:

Example

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

Unordered HTML List - Choose List Item Marker

The CSS list-style-type property is used to define the style of the list item marker. It can have one of the following values:

Value	Description
disc	Sets the list item marker to a bullet (default)
circle	Sets the list item marker to a circle
square	Sets the list item marker to a square
none	The list items will not be marked

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.

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 Table Tags

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.

HTML Table Example

Let's see the example of HTML table tag. It output is shown above.

```
<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>
```

output

```
First_Name Last_Name Marks
Sonoo      Jaiswal   60
James      William   80
Swati      Sironi    82
Chetna     Singh     72
```

HTML Frames

Html frames are useful at dividing the web page or the web browser into multiple sections; they separate sections then load differently.

A-frame displays content independent of its container. Multiple frames form a collection and are known as a frameset. The arrangement can be considered similar to the orientation of rows and columns in a table. The frame tag has been **deprecated in HTML5**.

Each frame has a frame tag to indicate. The horizontal frame is defined by the row attribute of the frame tag, and the vertical attribute is defined by the column attribute of **the frame tag in the HTML document**. The same window in one frame might display a static banner, a second navigation menu, etc.

Syntax of Frames

In HTML, there is no end tag for the <frame> tag. When we use frames in a webpage, we use <frameset> tag instead of a <body> tag. Using <frameset> is how we instruct the webpage to divide it into frames The <frame> tag defines which HTML document should open as frame.

The attribute cols take the value in pixels, and they help at specifying the number and size of **columns in a frameset**.

```
<cols ="10%,80%,10%">
```

Similarly, as we saw above, the attribute rows take the value in pixels, and they help at specifying the number and size of rows in a frameset.

```
<rows ="10%,80%,10%">
```

However, it should be duly noted that the <frame> tag has deprecated in HTML5 and it is no longer used.

Different Type of Frames in HTML

One of the uses of frames has always been to load navigation bars in one frame and then load main pages into a separate frame. A-frame tag defines one single window frame within a <frameset>.

<frame> has browser support in Chrome, Internet Explorer, Mozilla, Safari and Opera Mini.

The different and some commonly used attributes in the frame are attributes such as border, scrolling, src, name, etc.

Types of Frame Tag

Following are the different types of frame tags:

The <frameset> Tag Attribute

There are five important attributes of the <frameset>, and we discuss them here :

1. Col: The col attribute gives the vertical frames. The width, however, can be specified in 4 ways:

- Pixels: absolute values can be mentioned in pixels. If we have to create three vertical frames, we can give the value: cols=”100,50,100”.
- Percentage: The percentage of the browser window can be mentioned. If we have to create three vertical frames, we can give the value: cols=”40%,20%,40%.”

We can also make use of the wildcard here (*) and let the wildcard take the remaining of the window, which remains un-mentioned.

cols=”30%,*,30%”

2. Rows: The row attribute gives the horizontal frames. It specifies the rows in a frameset. If we have to create three horizontal frames we use:

Eg: Rows=”10%,80%,10%’.

We can also set the height of each row as we had done in the case of columns previously.

3. Border: It specifies the width of the border of each frame in pixels.

E.g., border=”4”. If border=”0”, it means there is no border.

4. Frameborder: If a three-dimensional border needs to be displayed between frames, then we use this attribute. The value which the attribute takes is either 1 or 0 (Yes or No).

E.g.: frameborder=”0” means no border.

5. Framespacing: This attribute specifies the amount of space between frames in a frameset. Any integer value can be given here for this attribute.

E.g., framespacing=”12” means between the frames, there should be the spacing of 12 pixels.

The <frame> Tag Attribute

There are 8 attributes that can be listed here in this section.

1. src: We provide the file name to this attribute that is supposed to be loaded into the frame. The value of this attribute can be any URL.

Eg: src= www.facebook.com

2. name: This attribute provides a name to the frame. It defines which frame a document should be loaded into. If you have multiple links in one frame that loads into another frame, we use this attribute. Then the second frame needs a name to identify itself as the target of the link.

E.g., name=”abc.htm.”

3. frameborder: This attribute is specified to show if the borders are to be shown or not. The value takes up is 1 or 0 (Yes or No).

4. marginwidth: The attribute is helpful at specifying the width of the space between the left and right of the frame’s border and also the content of the frame. The value is given in pixel.

E.g., marginwidth=”10”.

5. marginheight: The attribute is helpful at specifying the height of the space between the top and the bottom of the frame’s border and also the content of the frame. The value is given in pixel.

E.g., marginheight=”10”.

6. noresize: This attribute basically prevents the user from making any changes to the already present frames. In the absence of this attribute, any frame can be resized.

E.g., noresize=”noresize.”

7. scrolling: This attribute takes control of the appearance of the scroll bars, which are present on the frame. The value either a ‘yes, ‘no’, or ‘auto’.

E.g., scrolling=” no” means it should not have scroll bars.

8. longdesc: This attribute lets you provide a link to another page containing a long description of the content of the frame.

E.g., longdesc=”framedesc.htm.”

Frames and Browser Support

Old browsers do not support frames, then <noframes> elements should be displayed to the user.

In practice the <body> element should be placed inside <noframes> element because the <frameset> element is supposed to replace the <body> element. If the browser fails to understand <frameset> element then it understand the contents of the <body> element which is contained in <noframes> element.

HTML Links - Hyperlinks

HTML links are hyperlinks.

You can click on a link and jump to another document.

When you move the mouse over a link, the mouse arrow will turn into a little hand.

Note: A link does not have to be text. A link can be an image or any other HTML element!

HTML Links - Syntax

The HTML <a> tag defines a hyperlink. It has the following syntax:

<a href="*url*">*link text*</a>

The most important attribute of the <a> element is the href attribute, which indicates the link's destination.

The *link text* is the part that will be visible to the reader.

Clicking on the link text, will send the reader to the specified URL address.

Example

This example shows how to create a link to W3Schools.com:

```
<a href="https://www.w3schools.com/">Visit W3Schools.com!</a>
```

By default, links will appear as follows in all browsers:

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red

Tip: Links can of course be styled with CSS, to get another look!

HTML Links - The target Attribute

By default, the linked page will be displayed in the current browser window. To change this, you must specify another target for the link.

The target attribute specifies where to open the linked document.

The target attribute can have one of the following values:

- _self - Default. Opens the document in the same window/tab as it was clicked
- _blank - Opens the document in a new window or tab
- _parent - Opens the document in the parent frame
- _top - Opens the document in the full body of the window

Example

Use target="_blank" to open the linked document in a new browser window or tab:

```
<a href="https://www.w3schools.com/" target="_blank">Visit W3Schools!</a
```

Absolute URLs vs. Relative URLs

Both examples above are using an **absolute URL** (a full web address) in the href attribute.

A local link (a link to a page within the same website) is specified with a **relative URL** (without the "https://www" part):

Example

```
<h2>Absolute URLs</h2>
<p><a href="https://www.w3.org/">W3C</a></p>
<p><a href="https://www.google.com/">Google</a></p>

<h2>Relative URLs</h2>
<p><a href="html_images.asp">HTML Images</a></p>
<p><a href="/css/default.asp">CSS Tutorial</a></p>
```

HTML Links - Use an Image as a Link

To use an image as a link, just put the <img> tag inside the <a> tag:

Example

```
<a href="default.asp">
<img src="smiley.gif" alt="HTML tutorial" style="width:42px;height:42px;">
</a>
```

Link to an Email Address

Use mailto: inside the href attribute to create a link that opens the user's email program (to let them send a new email):

Example

```
<a href="mailto:someone@example.com">Send email</a>
```

Button as a Link

To use an HTML button as a link, you have to add some JavaScript code.

JavaScript allows you to specify what happens at certain events, such as a click of a button:

Example

```
<button onclick="document.location='default.asp'">HTML Tutorial</button>
```

Tip: Learn more about JavaScript in our JavaScript Tutorial.

Link Titles

The title attribute specifies extra information about an element. The information is most often shown as a tooltip text when the mouse moves over the element.

Example

```
<a href="https://www.w3schools.com/html/" title="Go to W3Schools HTML section">Visit our
HTML Tutorial</a>
```

More on Absolute URLs and Relative URLs

Example

Use a full URL to link to a web page:

```
<a href="https://www.w3schools.com/html/default.asp">HTML tutorial</a>
```

Example

Link to a page located in the html folder on the current web site:

```
<a href="/html/default.asp">HTML tutorial</a>
```

Example

Link to a page located in the same folder as the current page:

```
<a href="default.asp">HTML tutorial</a>
```

You can read more about file paths in the chapter HTML File Paths.

Summary

- Use the <a> element to define a link
- Use the href attribute to define the link address
- Use the target attribute to define where to open the linked document
- Use the <img> element (inside <a>) to use an image as a link
- Use the mailto: scheme inside the href attribute to create a link that opens the user's email program

HTML Link Tags

Tag	Description
<a>	Defines a hyperlink

HTML - Images

Images are very important to beautify as well as to depict many complex concepts in simple way on your web page. This tutorial will take you through simple steps to use images in your web pages.

Insert Image

You can insert any image in your web page by using **<img>** tag. Following is the simple syntax to use this tag.

```
<img src = "Image URL" ... attributes-list/>
```

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

Example

To try following example, let's keep our HTML file test.htm and image file test.png in the same directory −

```
<!DOCTYPE html>
<html>
```

```
  <head>
    <title>Using Image in Webpage</title>
  </head>

  <body>
    <p>Simple Image Insert</p>
    <img src = "/html/images/test.png" alt = "Test Image" />
  </body>

</html>
```

You can use PNG, JPEG or GIF image file based on your comfort but make sure you specify correct image file name in **src** attribute. Image name is always case sensitive.

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

Set Image Location

Usually we keep all the images in a separate directory. So let's keep HTML file test.htm in our home directory and create a subdirectory **images** inside the home directory where we will keep our image test.png.

Example

Assuming our image location is "image/test.png", try the following example −

```
<!DOCTYPE html>
<html>

  <head>
    <title>Using Image in Webpage</title>
  </head>

  <body>
    <p>Simple Image Insert</p>
    <img src = "/html/images/test.png" alt = "Test Image" />
  </body>

</html>
```

Set Image Width/Height

You can set image width and height based on your requirement using **width** and **height** attributes. You can specify width and height of the image in terms of either pixels or percentage of its actual size.

Example

```
<!DOCTYPE html>
<html>
```

```
  <head>
    <title>Set Image Width and Height</title>
  </head>

  <body>
    <p>Setting image width and height</p>
    <img src = "/html/images/test.png" alt = "Test Image" width = "150" height = "100"/>
  </body>

</html>
```

Set Image Border

By default, image will have a border around it, you can specify border thickness in terms of pixels using border attribute. A thickness of 0 means, no border around the picture.

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Set Image Border</title>
  </head>

  <body>
    <p>Setting image Border</p>
    <img src = "/html/images/test.png" alt = "Test Image" border = "3"/>
  </body>

</html>
```

Set Image Alignment

By default, image will align at the left side of the page, but you can use **align** attribute to set it in the center or right.

```
<!DOCTYPE html>
<html>

  <head>
    <title>Set Image Alignment</title>
  </head>

  <body>
    <p>Setting image Alignment</p>
    <img src = "/html/images/test.png" alt = "Test Image" border = "3" align = "right"/>
  </body>

</html>
```

HTML Multimedia: HTML Media Tags with Examples

The term "media" refers to various means of communicating and disseminating information, such as text, images, graphics, audio, video, and animation. All these mediums of communication are collectively termed "multimedia."

A combination of video and audio files can also be used in websites to increase viewership or to provide users with information and entertainment.

HTML helps you add multimedia files to your website by providing various multimedia tags. So without any further delay, let's start describing the tags used to include multimedia elements in a web page.

HTML Multimedia Tags

HTML tags that fall under the category of "multimedia tags" are as follows:

- HTML AUDIO Tag
- HTML VIDEO Tag
- HTML EMBED Tag
- HTML OBJECT Tag

HTML Multimedia: The AUDIO tag

The AUDIO tag is used to display the audio file on the web page.

An audio file is a way for devices like computers, MP3 players, and mobile phones to store audio data.

To store audio data, you need to convert it into a digital format. The process of converting audio data into a digital file is called "encoding" of the raw audio data. It involves taking samples of audio data and storing them in a compressed format to reduce the file size.

An audio player decodes these compressed sample files to make the audio waves audible. The process of converting a digital file into audio data is known as "decoding." A codec performs the encoding and decoding of the raw audio data.

HTML5's AUDIO tag only supports the following three audio file formats:

- **.mp3**: The most popular of the three audio file formats supported by the HTML AUDIO tag is ".mp3." All web browsers support this audio file format.
- **.wav**: IBM and Microsoft created the ".wav" audio file format. It works well on all platforms, including Windows, Macintosh, and Linux.
- **.ogg**: The Xiph.Org Foundation created the ".ogg" audio file format.

Attributes of the AUDIO tag

The following table shows the attributes of the audio tag.

Attribute	Description
autoplay	plays the audio file as soon as the web page loads.
controls	displays the controls on the web page, such as play and pause buttons.
loop	replays the audio file.
preload	specifies whether the audio file is preloaded on the web page or not.
src	provides the location of the audio file to play.

HTML AUDIO tag example

Consider the following example to learn how to embed audio files into your website.

```
<audio controls>
  <source src="song.mp3" type="audio/mpeg">
  The AUDIO tag is not supported by your browser.
</audio>
```

When the AUDIO tag is not supported by the web browser, the text between the AUDIO tag is displayed. Because the text "The AUDIO tag is not supported by your browser." is available between the AUDIO tag in the above code fragment, it will be displayed if the AUDIO tag is not supported by the web browser.

You can provide multiple audio sources in multiple file formats. For example:

```
<audio controls>
```

```
  <source src="song.ogg" type="audio/ogg">
  <source src="song.mp3" type="audio/mpeg">
  The AUDIO tag is not supported by your browser.
</audio>
```

If there are multiple sources available between the AUDIO tag, the first supported source file will be used.

The "controls" attribute adds audio controls like play, pause, and volume.

HTML Multimedia: The VIDEO tag

The "VIDEO" tag, like "AUDIO," is used to display video files on a web page.

Following are the three video file formats supported in HTML:

- **.mp4**: This video file format has been developed by the "Moving Pictures Expert Group." This video file format is recommended by one of the biggest video sharing platforms, YouTube.
- **.webm**: This video file format has been developed by web giants Mozilla, Opera, Adobe, and Google.
- **.ogg**: The Xiph.Org Foundation created the ".ogg" video file format.

Attributes of the VIDEO tag

The following table describes the attributes of the Video tag.

Attribute	Description
audio	controls the audio channel's default state in the video.
autoplay	plays the audio file as soon as the web page loads.
controls	displays the controls on a web page, such as play and pause buttons.
height	specifies the height of the video tag.
loop	plays the video file again.
preload	specifies whether the video file is preloaded on the web page or not.

poster	provides an image to be displayed when the video file is not available.
src	provides the location of the video file to play.
width	specifies the width of the video tag.

HTML Video tag example

To provide the source of the video file, you can also use the SOURCE tag within the opening and closing tags of the VIDEO tag.

When the location of the video file cannot be determined, the SOURCE tag is used. In this case, the VIDEO tag plays the first video file found in the specified path. The VIDEO tag is demonstrated in the following code snippet:

```
<VIDEO src="video.ogv" autoplay="true" loop="3" controls>
</VIDEO>
```

In the src attribute of the preceding code snippet, we defined a video.ogv file. We've also set the autoplay attribute to true, which means the video will begin playing as soon as the web page loads. Because the loop attribute is set to 3, the video file will be played three times. Furthermore, the controls attribute shows the controls on the video player.

Consider the following example to learn how to embed video into your website.

```
<video width="320" height="240" controls>
  <source src="song.mp4" type="video/mp4">
  <source src="song.ogg" type="video/ogg">
   The VIDEO tag is not supported by your browser.
</video>
```

It is a good idea to always include width and height attributes. If height and width are not set, the browser does not know the size of the video. The effect will be that the page will change (or flicker) while the video loads. The remaining code works similarly to "AUDIO."

HTML Multimedia: The EMBED tag

HTML allows you to embed plug-ins in a web page using the EMBED tag. This tag lets you embed multimedia in a web page and play it while opening the page.

The EMBED tag is supported by Internet Explorer as well as Netscape Navigator. It is also supported across the Windows and Mac platforms.

The EMBED tag uses the three mandatory attributes, namely src, height, and width.

Attributes of the EMBED tag

The following table lists various attributes of the EMBED tag in HTML.

Attribute	Description
height	specifies the height of the embedded component.
hspace	sets the horizontal padding around the tag.
type	specifies the Multipurpose Internet Mail Extension (MIME) type for the components.
width	sets the width of the page's embedded component.

HTML EMBED tag example

Following is an example of using the EMBED tag.

```
<EMBED src="Music.mp3" width=600 height=100></EMBED>
```

In the previous example, we used the src attribute to tell the browser which multimedia file to play while the web page loads. We have also defined the height and width attributes to specify the height and width of the embedded multimedia component on the page, respectively.

HTML Multimedia: The OBJECT tag

Objects like images, sounds, videos, Java applets, ActiveX controls, Portable Document Format (PDF), and Flash objects can be added to a web page using the HTML OBJECT tag.

An object tag can also be used inside the body tag. The text between the starting and ending tags of the OBJECT tag is the alternate text for browsers that do not support this tag.

Attributes of the OBJECT tag

The following table shows various attributes of the OBJECT tag.

Attribute	Description
data	specifies the URL of the object's data.
form	specifies which form(s) the object belongs to.
height	specifies the height of the object in pixels.
name	specifies the object's name.
type	specifies the MIME type for the component.
usemap	specifies the URL.
width	sets the width of the page's embedded components.

HTML OBJECT tag example

The OBJECT tag initializes the object by passing the parameters to the object, which can be done using the PARAM tag.

In HTML, you can use the PARAM tag to define parameters or variables for an OBJECT tag. An OBJECT tag can contain multiple PARAM tags, as shown in the following code snippet.

```
<OBJECT data="movie.avi" type="video/quicktime" id="video" width="200" height="100">
  <PARAM name="BorderStyle" value="1" />
  <PARAM name="autoplay value=true />
</OBJECT>
```

In the above code snippet, we have defined an OBJECT tag, which includes a video file in a web page. We have also used the PARAM tags to pass the parameters for the OBJECT tag.

The following table shows the various attributes of the PARAM tag.

Attribute	Description

name	specifies the name of the parameter.
value	specifies the value of the parameter.

History of Browser Support for Multimedia

The first web browsers had support for text only and were limited to a single font in a single color. Later came browsers with support for colors, fonts, and even pictures.

The support for sounds, animations, and videos is handled differently by various browsers. Different types and formats are supported, and some formats require extra helper programs (plug-ins) to work.

Hopefully, this will become history. HTML5 multimedia promises an easier future for multimedia.

Multimedia Formats

The most common way to discover the type of a file, is to look at the file extension. When a browser sees the file extension ".htm" or ".html," it will treat the file as an HTML file. The **.xml** extension indicates an XML file, and the **.css** extension indicates a style sheet file. Images are identified by file extensions such as **.gif**, **.png**, and **.jpg**.

Multimedia files have their own formats and extensions as well, such as **.swf**, **.wav**, **.mp3**, **.mp4**, **.mpg**, **.wmv**, and **.avi**.

HTML object and param

- Attributes of the object and param

HTML <object> - embedding media

The **<object>** tag is used for embedding media contents (such as images, Java applets, Flash, movie, audio, or even another HTML file) on a web page.
The object element is partially supported in all major browsers, depend on the

object inserted into HTML document.
If the object element is not displayed, the code between the <object> and </object> tags will be executed, so you can add an alternative text or an image.

This element is designed to include any sort of document. It uses the **type** attribute to indicate the "mime type", and the **data** attribute to indicate the source of the document (its URI).
If the browser (or one of its plugin) can interpret the embedded object, will download and play /run it, otherwise the nested content of the object is displayed. Any HTML code can be inserted as an alternative content, for example a link to the document or an image.

Examples:

Embed a PDF document

```
<object type="application/pdf" data="dir/html_course.pdf" width="300" height="200">

 <img src="dir/html_course.jpg" alt="HTML course" />

</object>
```

- Here the browser will show a PDF document if it supports the PDF format, otherwise it will show a JPG image (*html_course.jpg*).
The "*width*" and "height" attributes define the visible area of the object. The browser scale the object to these dimensions.

Embed an HTML file

```
<object type="text/html" data="dir/test.html" width="500" height="350">

 alt: <a href="dir/test.html" title="Test">test.html</a>

</object>
```

- If the browser can't use the object element to include an "*text/html*" file, will show the link added between the <object> and </object> tags (*alt: test.html*).

• The object element may also contain a number of **param** elements to define parameters or values for the object when it displays or plays.

Embed a SWF file

- This example includes a Flash game (in SWF format), and uses a <param> tag to define a background color for the area in which the game is displayed (set with the "width" and "height" attributes).
- Internet Explorer sometimes needs a <param> with a **src** parameter to understand the location.

```
<object type="application/x-shockwave-flash" data="games/cubilus.swf"
width="500" height="250">

 <param name="src" value="html/l/cubilus.swf" />

 <param name="bgcolor" value="#fbfbfe" />

 Your browser not support SWF.

</object>
```

If the browser supports SWF application, will display the "cubilus" game, otherwise shows the text: "*Your browser not support SWF.*"
Here is the result:

Embend a WAV file

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

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

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

 <param name="autoStart" value="false" />

 Alt : <a href="dir/test.wav" title="WAV file">test.wav</a>

</object>
```

- Depending on the object, the major browsers use different codes to load the same object type. QuickTime understand the "autoplay" parameter. Windows media Player and Real Audio understand "autoStart" parameter.

Embendding audio MP3

```
<object type="audio/mpeg" data="audio/test.mp3" width="200" height="20">

 <param name="src" value="audio/test.mp3">

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

 <param name="autoStart" value="false" />

 Your system can't play audio/mpeg files.

</object>
```

Embendding MOV files

- For some types, like QuickTime document (MOV files), IE needs a non-standard value to the **classid** attribute, an identifier to load an associated activeX.
We can nest another object as an alternative content, for the other browsers that use the standard code:

```
<object classid="clsid:02BF25D5-8C17-4B23-BC80-D3488ABDDC6B"
codebase="http://www.apple.com/qtactivex/qtplugin.cab" width="480"
height="360">

 <param name="src" value="dir/movie.mov">

 <param name="controller" value="true" />

 <param name="autoplay" value="false" />
 <object type="video/quicktime" data="dir/movie.mov" width="480" height="360">

 <param name="controller" value="true" />

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

 You do not have QuickTime Player installed.

 </object>

</object>
```

Attributes of the HTML object tag

The <object> tag supports the following attributes (along with: "id", "class", "style", "title"):

- **archive="*URLs*"** - Specifies a space-separated list of URLs for resources that are related to the object.
- **border="*pixels*"** - Defines a border around the object (*Deprecated*).
- **classid="*classid_ID*"** - Defines a class ID value as set in the Windows Registry or a URL.
- **codebase="*URL*"** - Defines where to find the code for the object. Used in case if the browser not support it.
- **codetype="*MIME_type*"** - The internet media type of the code referred to by the classid attribute.
- **data="*URL*"** - Specifies the address of the resource.
- **declare** (declare="*declare*" in XHTML) - Declares an object but restrains the browser from downloading and processing until need it.
- **height="*pixels*"** - Defines the height of the object, in pixels.
- **hspace="*pixels*"** - Sets a horizontal space around the object, in pixels (*Deprecated*).
- **name="*text*"** - Defines the name for an object (to use in scripts).
- **standby="*message*"** - Specifies a message to display while the object is loading.
- **type="*MIME_type*"** - Defines the media type of the resource specified in the "data" attribute.

- **usemap="*URL*"** - Specifies a URL of a image map to be used with the object.
- **vspace="*pixels*"** - Sets a vertical space around the object, in pixels (*Deprecated*).
- **width="*pixels*"** - Defines the width of the object, in pixels.

Attributes of the param tag

The <param> tag supports the following attributes:

- **id="*text*"** - Specifies a unique "id", it can be referenced from a script, or CSS style sheet.
- **name="*text*"** - Defines the name of the parameter (*Required*).
- **type="*MIME_type*"** - Specifies the media type of the resource only when the "valuetype" attribute is set to "*ref*".
- **value="*value*"** - Defines the value of the parameter.
- **valuetype="*data | ref | object*"** - Specifies the type of value:
    - **data** - indicates that the parameter's value is data (default)
    - **ref** - indicates that the parameter's value is a URL
    - **object** - indicates that the value is the URL of another object in the document

HTML - Forms

HTML Forms are required, when you want to collect some data from the site visitor. For example, during user registration you would like to collect information such as name, email address, credit card, etc.

A form will take input from the site visitor and then will post it to a back-end application such as CGI, ASP Script or PHP script etc. The back-end application will perform required processing on the passed data based on defined business logic inside the application.

There are various form elements available like text fields, textarea fields, drop-down menus, radio buttons, checkboxes, etc.

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.

HTML Form Controls

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

- Text Input Controls
- Checkboxes Controls
- Radio Box Controls
- Select Box Controls
- File Select boxes
- Hidden Controls

- Clickable Buttons
- Submit and Reset Button

Text Input Controls

There are three types of text input used on forms −

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

Single-line text input controls

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

Example

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

```
<!DOCTYPE html>
<html>

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

   <body>
      <form >
         First name: <input type = "text" name = "first_name" />
         <br>
         Last name: <input type = "text" name = "last_name" />
      </form>
   </body>

</html>
```

Attributes

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

Sr.No	Attribute & Description

1	**type** Indicates the type of input control and for text input control it will be set to **text**.
2	**name** Used to give a name to the control which is sent to the server to be recognized and get the value.
3	**value** This can be used to provide an initial value inside the control.
4	**size** Allows to specify the width of the text-input control in terms of characters.
5	**maxlength** Allows to specify the maximum number of characters a user can enter into the text box.

Password input controls

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

Example

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

```
<!DOCTYPE html>
<html>

   <head>
      <title>Password Input Control</title>
   </head>

   <body>
      <form >
         User ID : <input type = "text" name = "user_id" />
         <br>
         Password: <input type = "password" name = "password" />
      </form>
   </body>

</html>
```

Attributes

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

Sr.No	Attribute & Description
1	**type** Indicates the type of input control and for password input control it will be set to **password**.
2	**name** Used to give a name to the control which is sent to the server to be recognized and get the value.
3	**value** This can be used to provide an initial value inside the control.
4	**size** Allows to specify the width of the text-input control in terms of characters.
5	**maxlength** Allows to specify the maximum number of characters a user can enter into the text box.

Multiple-Line Text Input Controls

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

Example

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

```
<!DOCTYPE html>
<html>

  <head>
    <title>Multiple-Line Input Control</title>
  </head>

  <body>
    <form>
      Description : <br />
      <textarea rows = "5" cols = "50" name = "description">
        Enter description here...
      </textarea>
    </form>
  </body>
```

```

</html>
```

Attributes

Following is the list of attributes for <textarea> 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	**rows** Indicates the number of rows of text area box.
3	**cols** Indicates the number of columns of text area box

Checkbox Control

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

Example

Here is an example HTML code for a form with two checkboxes −

```
<!DOCTYPE html>
<html>

   <head>
      <title>Checkbox Control</title>
   </head>

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

</html>
```

Attributes

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

Sr.No	Attribute & Description
1	**type** Indicates the type of input control and for checkbox input control it will be set to **checkbox.**.
2	**name** Used to give a name to the control which is sent to the server to be recognized and get the value.
3	**value** The value that will be used if the checkbox is selected.
4	**checked** Set to *checked* if you want to select it by default.

Radio Button Control

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

Example

Here is example HTML code for a form with two radio buttons −

```
<!DOCTYPE html>
<html>

  <head>
    <title>Radio Box Control</title>
  </head>

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

</html>
```

Attributes

Following is the list of attributes for radio button.

Sr.No	Attribute & Description
1	**type** Indicates the type of input control and for checkbox input control it will be set to radio.
2	**name** Used to give a name to the control which is sent to the server to be recognized and get the value.
3	**value** The value that will be used if the radio box is selected.
4	**checked** Set to *checked* if you want to select it by default.

Select Box Control

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

Example

Here is example HTML code for a form with one drop down box

```
<!DOCTYPE html>
<html>

   <head>
      <title>Select Box Control</title>
   </head>

   <body>
      <form>
         <select name = "dropdown">
            <option value = "Maths" selected>Maths</option>
            <option value = "Physics">Physics</option>
      </select>
    </form>
  </body>

</html>
```

Attributes

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

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**.

Example

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

```
<!DOCTYPE html>
<html>

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

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

</html>
```

Attributes

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

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	**accept** Specifies the types of files that the server accepts.

Button Controls

There are various ways in HTML to create clickable buttons. You can also create a clickable button using <input>tag by setting its type attribute to **button**. The type attribute can take the following values −

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.

Example

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

```
<!DOCTYPE html>
<html>

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

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

</html>
```

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.

Example

Here is example HTML code to show the usage of hidden control −

```
<!DOCTYPE html>
<html>

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

   <body>
      <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>
   </body>

</html>
```