

HTML

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 is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning HTML:

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

```
<!DOCTYPE html>
<html>
  <head>
    <title>This is document title</title>
  </head>
  <body>
    <h1>This is a heading</h1>
    <p>Hello World!</p>
  </body>
</html>
```

Applications of HTML

As mentioned before, HTML is one of the most widely used language over the web. I'm going to list few of them here:

- **Web pages development** - HTML is used to create pages which are rendered over the web. Almost every page of web is having html tags in it to render its details in browser.
- **Internet Navigation** - HTML provides tags which are used to navigate from one page to another and is heavily used in internet navigation.
- **Responsive UI** - HTML pages now-a-days works well on all platform, mobile, tabs, desktop or laptops owing to responsive design strategy.
-
- **Offline support** HTML pages once loaded can be made available offline on the machine without any need of internet.
- **Game development**- HTML5 has native support for rich experience and is now useful in gaming development arena as well.

HTML stands for Hypertext Markup Language, 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.

Basic HTML Document

In its simplest form, following is an example of an HTML document –

```
<!DOCTYPE html>
<html>

  <head>
    <title>This is document title</title>
  </head>

  <body>
    <h1>This is a heading</h1>
    <p>Document content goes here.....</p>
  </body>

</html>
```

HTML Tags

As told earlier, HTML is a markup language and makes use of various tags to format the content. These tags are enclosed within angle braces **<Tag Name>**. Except few tags, most of the tags have their corresponding closing tags. For example, **<html>** has its closing tag **</html>** and **<body>** tag has its closing tag **</body>** tag etc.

Above example of HTML document uses the following 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.

To learn HTML, you will need to study various tags and understand how they behave, while formatting a textual document. Learning HTML is simple as users have to learn the usage of different tags in order to format the text or images to make a beautiful webpage.

World Wide Web Consortium (W3C) recommends to use lowercase tags starting from HTML 4.

HTML Document Structure

A typical HTML document will have the following structure –

```
<html>
```

```
  <head>
```

```
    Document header related tags
```

```
  </head>
```

```
  <body>
```

```
    Document body related tags
```

```
  </body>
```

```
</html>
```

The <!DOCTYPE> Declaration

The <!DOCTYPE> declaration tag is used by the web browser to understand the version of the HTML used in the document. Current version of HTML is 5 and it makes use of the following declaration –

```
<!DOCTYPE html>
```

There are many other declaration types which can be used in HTML document depending on what version of HTML is being used.

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.

Example

```
<!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>
```

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>
```

Line Break Tag

Whenever you use the **
** 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 **
** 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 **
** 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 />
    Mahnaz</p>
```

```
</body>
```

```
</html>
```

Centering Content

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

Example

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Centring Content Example</title>
```

```
</head>
```

```
<body>
```

```
<p>This text is not in the center.</p>
```

```
<center>
```

```
<p>This text is in the center.</p>
```

```
</center>
```

```
</body>
```

```
</html>
```

Horizontal Lines

Horizontal lines are used to visually break-up sections of a document. The **<hr>** tag creates a line from the current position in the document to the right margin and breaks the line accordingly.

For example, you may want to give a line between two paragraphs as in the given example below –

Example

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Horizontal Line Example</title>
```

```
</head>
```

```
<body>
```

```
<p>This is paragraph one and should be on top</p>
```

```
<hr />
```

```
<p>This is paragraph two and should be at bottom</p>
```

```
</body>
```

```
</html>
```

Again **<hr />** tag is an example of the **empty** element, where you do not need opening and closing tags, as there is nothing to go in between them.

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

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 –

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

In cases, where you do not want the client browser to break text, you should use a nonbreaking space entity ** **; instead of a normal space. For example, when coding the "12 Angry Men" in a paragraph, you should use something similar to the following code –

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Nonbreaking Spaces Example</title>
  </head>

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

</html>
```

HTML - Elements

An **HTML element** is defined by a starting tag. If the element contains other content, it ends with a closing tag, where the element name is preceded by a forward slash as shown below with few tags –

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

So here <p>...</p> is an HTML element, <h1>...</h1> is another HTML element. There are some HTML elements which don't need to be closed, such as <img.../>, <hr /> and
 elements. These are known as **void elements**.

HTML documents consists of a tree of these elements and they specify how HTML documents should be built, and what kind of content should be placed in what part of an HTML document.

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

It is very much allowed to keep one HTML element inside another HTML element –

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Nested Elements Example</title>
  </head>
```

```
<body>
  <h1>This is <i>italic</i> heading</h1>
  <p>This is <u>underlined</u> paragraph</p>
</body>

</html>
```

HTML - Formatting

Bold Text

Anything that appears within **...** element, is displayed in bold as shown below –

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Bold Text Example</title>
  </head>

  <body>
    <p>The following word uses a <b>bold</b> typeface.</p>
  </body>

</html>
```

Italic Text

Anything that appears within *<i>...</i>* element is displayed in italicized as shown below –

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Italic Text Example</title>
  </head>

  <body>
    <p>The following word uses an <i>italicized</i> typeface.</p>
  </body>

</html>
```

This will produce the following result –

Underlined Text

Anything that appears within <u>...</u> element, is displayed with underline as shown below –

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Underlined Text Example</title>
  </head>

  <body>
    <p>The following word uses an <u>underlined</u> typeface.</p>
```

</body>

</html>

Strike Text

Anything that appears within **<strike>...</strike>** element is displayed with strikethrough, which is a thin line through the text as shown below –

Example

<!DOCTYPE html>

<html>

<head>

<title>Strike Text Example</title>

</head>

<body>

<p>The following word uses a <strike>strikethrough</strike> typeface.</p>

</body>

</html>

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.

Example

<!DOCTYPE html>

<html>

<head>

<title>Monospaced Font Example</title>

</head>

<body>

<p>The following word uses a <tt>monospaced</tt> typeface.</p>

</body>

</html>

Superscript Text

The content of a **^{...}** 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.

Example

<!DOCTYPE html>

<html>

<head>

<title>Superscript Text Example</title>

</head>

<body>

<p>The following word uses a ^{superscript} typeface.</p>

</body>

</html>

Subscript Text

The content of a **_{...}** 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.

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Subscript Text Example</title>
  </head>

  <body>
    <p>The following word uses a <sub>subscript</sub> typeface.</p>
  </body>

</html>
```

Inserted Text

Anything that appears within **<ins>...</ins>** element is displayed as inserted text.

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Inserted Text Example</title>
  </head>

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

</html>
```

Deleted Text

Anything that appears within **...** element, is displayed as deleted text.

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Deleted Text Example</title>
  </head>

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

</html>
```

Larger Text

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

Example

```
<!DOCTYPE html>
```

```
<html>

<head>
  <title>Larger Text Example</title>
</head>

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

</html>
```

Smaller Text

The content of the `<small>...</small>` element is displayed one font size smaller than the rest of the text surrounding it as shown below –

Example

```
<!DOCTYPE html>
<html>

<head>
  <title>Smaller Text Example</title>
</head>

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

</html>
```

HTML - Phrase Tags

The phrase tags have been desicolgned for specific purposes, though they are displayed in a similar way as other basic tags like ``, `<i>`, `<pre>`, and `<tt>`, you have seen in previous chapter. This chapter will take you through all the important phrase tags, so let's start seeing them one by one.

Emphasized Text

Anything that appears within `...` element is displayed as emphasized text.

Example

```
<!DOCTYPE html>
<html>

<head>
  <title>Emphasized Text Example</title>
</head>

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

</html>
```

Marked Text

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

Example

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

```
  <head>
    <title>Marked Text Example</title>
  </head>
```

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

```
</html>
```

Strong Text

Anything that appears within **...** element is displayed as important text.

Example

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

```
  <head>
    <title>Strong Text Example</title>
  </head>
```

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

```
</body>
```

```
</html>
```

Quoting Text

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

Text inside a **<blockquote>** element is usually indented from the left and right edges of the surrounding text, and sometimes uses an italicized font.

Example

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

```
  <head>
    <title>Blockquote Example</title>
  </head>
```

```
  <body>
    <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>
  </body>
```

```
</html>
```

Text Citations

If you are quoting a text, you can indicate the source placing it between an opening **<cite>** tag and closing **</cite>** tag

As you would expect in a print publication, the content of the **<cite>** element is rendered in italicized text by default.

Example

<!HTML - Comments

Comment is a piece of code which is ignored by any web browser. It is a good practice to add comments into your HTML code, especially in complex documents, to indicate sections of a document, and any other notes to anyone looking at the code. Comments help you and others understand your code and increases code readability.

HTML comments are placed in between **<!-- ... -->** tags. So, any content placed with-in **<!-- ... -->** tags will be treated as comment and will be completely ignored by the browser.

Example

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head> <!-- Document Header Starts -->
```

```
    <title>This is document title</title>
```

```
  </head> <!-- Document Header Ends -->
```

```
  <body>
```

```
    <p>Document content goes here.....</p>
```

```
  </body>
```

```
</html>
```

This will produce the following result without displaying the content given as a part of comments –

Valid vs Invalid Comments

Comments do not nest which means a comment cannot be put inside another comment. Second the double-dash sequence **"--"** may not appear inside a comment except as part of the closing **-->** tag. You must also make sure that there are no spaces in the start-of comment string.

Example

Here, the given comment is a valid comment and will be wiped off by the browser.

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Valid Comment Example</title>
```

```
  </head>
```

```
  <body>
```

```
    <!-- This is valid comment -->
```

```
    <p>Document content goes here.....</p>
```

```
  </body>
```

```
</html>
```

But, following line is not a valid comment and will be displayed by the browser. This is because there is a space between the left angle bracket and the exclamation mark.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
  <title>Invalid Comment Example</title>
</head>

<body>
  <!-- This is not a valid comment -->
  <p>Document content goes here.....</p>
</body>

</html>
```

Multiline Comments

So far we have seen single line comments, but HTML supports multi-line comments as well.

You can comment multiple lines by the special beginning tag `<!--` and ending tag `-->` placed before the first line and end of the last line as shown in the given example below.

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Multiline Comments</title>
  </head>

  <body>
    <!--
      This is a multiline comment and it can
      span through as many as lines you like.
    -->

    <p>Document content goes here.....</p>
  </body>

</html>
```

Conditional Comments

Conditional comments only work in Internet Explorer (IE) on Windows but they are ignored by other browsers. They are supported from Explorer 5 onwards, and you can use them to give conditional instructions to different versions of IE.

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Conditional Comments</title>

    <!--[if IE 6]>
      Special instructions for IE 6 here
    <![endif]-->
  </head>

  <body>
    <p>Document content goes here.....</p>
  </body>
```

</html>

Using Comment Tag

There are few browsers that support <comment> tag to comment a part of HTML code.

Note – The <comment> tag deprecated in HTML5. Do not use this element.

Example

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Using Comment Tag</title>
```

```
  </head>
```

```
  <body>
```

```
    <p>This is <comment>not</comment> Internet Explorer.</p>
```

```
  </body>
```

```
</html>
```

```
<html>
```

```
  <head>
```

```
    <title>Citations Example</title>
```

```
  </head>
```

```
  <body>
```

```
    <p>This HTML tutorial is derived from <cite>W3 Standard for HTML</cite>.</p>
```

```
  </body>
```

```
</html>
```

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 **** tag. Following is the simple syntax to use this tag.

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

The 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.

Example

```
<!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 - Tables

The HTML tables allow web authors to arrange data like text, images, links, other tables, etc. into rows and columns of cells.

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

Example

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>HTML Tables</title>
```

```
  </head>
```

```
  <body>
```

```
    <table border = "1">
```

```
      <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>
```


</html>

Here, the **border** is an attribute of <table> tag and it is used to put a border across all the cells. If you do not need a border, then you can use border = "0".

Table Heading

Table heading can be defined using <th> tag. This tag will be put to replace <td> tag, which is used to represent actual data cell. Normally you will put your top row as table heading as shown below, otherwise you can use <th> element in any row. Headings, which are defined in <th> tag are centered and bold by default.

Example

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>HTML Table Header</title>
```

```
  </head>
```

```
  <body>
```

```
    <table border = "1">
```

```
      <tr>
```

```
        <th>Name</th>
```

```
        <th>Salary</th>
```

```
      </tr>
```

```
      <tr>
```

```
        <td>Ramesh Raman</td>
```

```
        <td>5000</td>
```

```
      </tr>
```

```
      <tr>
```

```
        <td>Shabbir Hussein</td>
```

```
        <td>7000</td>
```

```
      </tr>
```

```
    </table>
```

```
  </body>
```

```
</html>
```

This will produce the following result –

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.

Example

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>HTML Table Cellpadding</title>
```

```
  </head>
```

```
  <body>
```

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

```
      <tr>
```

```
        <th>Name</th>
```

```
        <th>Salary</th>
```

```

</tr>
<tr>
  <td>Ramesh Raman</td>
  <td>5000</td>
</tr>
<tr>
  <td>Shabbir Hussein</td>
  <td>7000</td>
</tr>
</table>
</body>

```

```

</html>

```

This will produce the following result –

Colspan and Rowspan Attributes

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

Example

```

<!DOCTYPE html>
<html>

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

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

</html>

```

Tables Backgrounds

You can set table background using one of the following two ways –

- **bgcolor** attribute – You can set background color for whole table or just for one cell.
- **background** attribute – You can set background image for whole table or just for one cell.

You can also set border color also using **bordercolor** attribute.

Note – The *bgcolor*, *background*, and *bordercolor* attributes deprecated in HTML5. Do not use these attributes.

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>HTML Table Background</title>
  </head>

  <body>
    <table border = "1" bordercolor = "green" bgcolor = "yellow">
      <tr>
        <th>Column 1</th>
        <th>Column 2</th>
        <th>Column 3</th>
      </tr>
      <tr>
        <td rowspan = "2">Row 1 Cell 1</td>
        <td>Row 1 Cell 2</td>
        <td>Row 1 Cell 3</td>
      </tr>
      <tr>
        <td>Row 2 Cell 2</td>
        <td>Row 2 Cell 3</td>
      </tr>
      <tr>
        <td colspan = "3">Row 3 Cell 1</td>
      </tr>
    </table>
  </body>

</html>
```

Here is an example of using **background** attribute. Here we will use an image available in /images directory.

```
<!DOCTYPE html>
<html>

  <head>
    <title>HTML Table Background</title>
  </head>

  <body>
    <table border = "1" bordercolor = "green" background = "/images/test.png">
      <tr>
        <th>Column 1</th>
        <th>Column 2</th>
        <th>Column 3</th>
      </tr>
      <tr>
        <td rowspan = "2">Row 1 Cell 1</td>
        <td>Row 1 Cell 2</td><td>Row 1 Cell 3</td>
      </tr>
      <tr>
```

```

        <td>Row 2 Cell 2</td>
        <td>Row 2 Cell 3</td>
    </tr>
    <tr>
        <td colspan = "3">Row 3 Cell 1</td>
    </tr>
</table>
</body>

```

```
</html>
```

Table Height and Width

You can set a table width and height using **width** and **height** attributes. You can specify table width or height in terms of pixels or in terms of percentage of available screen area.

Example

```

<!DOCTYPE html>
<html>

    <head>
        <title>HTML Table Width/Height</title>
    </head>

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

</html>

```

Table Caption

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

Example

```

<!DOCTYPE html>
<html>

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

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

            <tr>

```

```

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

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

```

</html>

Table Header, Body, and Footer

Tables can be divided into three portions – a header, a body, and a foot. The head and foot are rather similar to headers and footers in a word-processed document that remain the same for every page, while the body is the main content holder of the table.

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.

A table may contain several <tbody> elements to indicate *different pages* or groups of data. But it is notable that <thead> and <tfoot> tags should appear before <tbody>

Example

```

<!DOCTYPE html>
<html>

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

    <body>
        <table border = "1" width = "100%">
            <thead>
                <tr>
                    <td colspan = "4">This is the head of the table</td>
                </tr>
            </thead>

            <tfoot>
                <tr>
                    <td colspan = "4">This is the foot of the table</td>
                </tr>
            </tfoot>

            <tbody>
                <tr>
                    <td>Cell 1</td>
                    <td>Cell 2</td>
                    <td>Cell 3</td>
                    <td>Cell 4</td>
                </tr>
            </tbody>

        </table>
    </body>

```

</body>

</html>

Nested Tables

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

Example

Following is the example of using another table and other tags inside a table cell.

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>HTML Table</title>
```

```
  </head>
```

```
  <body>
```

```
    <table border = "1" width = "100%">
```

```
      <tr>
```

```
        <td>
```

```
          <table border = "1" width = "100%">
```

```
            <tr>
```

```
              <th>Name</th>
```

```
              <th>Salary</th>
```

```
            </tr>
```

```
            <tr>
```

```
              <td>Ramesh Raman</td>
```

```
              <td>5000</td>
```

```
            </tr>
```

```
            <tr>
```

```
              <td>Shabbir Hussein</td>
```

```
              <td>7000</td>
```

```
            </tr>
```

```
          </table>
```

```
        </td>
```

```
      </tr>
```

```
    </table>
```

```
  </body>
```

```
</html>
```

HTML - Lists

HTML offers web authors three ways for specifying lists of information. All lists must contain one or more list elements. Lists may contain –

- **** – An unordered list. This will list items using plain bullets.
- **** – An ordered list. This will use different schemes of numbers to list your items.
- **<dl>** – A definition list. This arranges your items in the same way as they are arranged in a dictionary.

HTML Unordered Lists

An unordered list is a collection of related items that have no special order or sequence. This list is created by using HTML **** tag. Each item in the list is marked with a bullet.

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>HTML Unordered List</title>
  </head>

  <body>
    <ul>
      <li>Beetroot</li>
      <li>Ginger</li>
      <li>Potato</li>
      <li>Radish</li>
    </ul>
  </body>

</html>
```

The type Attribute

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

```
<ul type = "square">
<ul type = "disc">
<ul type = "circle">
```

Example

Following is an example where we used <ul type = "square">

```
<!DOCTYPE html>
<html>

  <head>
    <title>HTML Unordered List</title>
  </head>

  <body>
    <ul type = "square">
      <li>Beetroot</li>
      <li>Ginger</li>
      <li>Potato</li>
      <li>Radish</li>
    </ul>
  </body>

</html>
```

Example

Following is an example where we used <ul type = "disc"> –

```
<!DOCTYPE html>
<html>

  <head>
    <title>HTML Unordered List</title>
  </head>
```

```
<body>
  <ul type = "disc">
    <li>Beetroot</li>
    <li>Ginger</li>
    <li>Potato</li>
    <li>Radish</li>
  </ul>
</body>

</html>
```

Example

Following is an example where we used `<ul type = "circle">` –

```
<!DOCTYPE html>
<html>

  <head>
    <title>HTML Unordered List</title>
  </head>

  <body>
    <ul type = "circle">
      <li>Beetroot</li>
      <li>Ginger</li>
      <li>Potato</li>
      <li>Radish</li>
    </ul>
  </body>

</html>
```

HTML Ordered Lists

If you are required to put your items in a numbered list instead of bulleted, then HTML ordered list will be used. This list is created by using `` tag. The numbering starts at one and is incremented by one for each successive ordered list element tagged with ``.

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>HTML Ordered List</title>
  </head>

  <body>
    <ol>
      <li>Beetroot</li>
      <li>Ginger</li>
      <li>Potato</li>
      <li>Radish</li>
    </ol>
  </body>

</html>

The type Attribute
```


You can use **type** attribute for tag to specify the type of numbering you like. By default, it is a number. Following are the possible options –

<ol type = "1"> - Default-Case Numerals.

<ol type = "I"> - Upper-Case Numerals.

<ol type = "i"> - Lower-Case Numerals.

<ol type = "A"> - Upper-Case Letters.

<ol type = "a"> - Lower-Case Letters.

Example

Following is an example where we used <ol type = "1">

```
<!DOCTYPE html>
<html>

  <head>
    <title>HTML Ordered List</title>
  </head>

  <body>
    <ol type = "1">
      <li>Beetroot</li>
      <li>Ginger</li>
      <li>Potato</li>
      <li>Radish</li>
    </ol>
  </body>

</html>
```

Example

Following is an example where we used <ol type = "I">

```
<!DOCTYPE html>
<html>

  <head>
    <title>HTML Ordered List</title>
  </head>

  <body>
    <ol type = "I">
      <li>Beetroot</li>
      <li>Ginger</li>
      <li>Potato</li>
      <li>Radish</li>
    </ol>
  </body>

</html>
```

Example

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

```
<!DOCTYPE html>
<html>

  <head>
```

```
<title>HTML Ordered List</title>
</head>

<body>
  <ol type = "i">
    <li>Beetroot</li>
    <li>Ginger</li>
    <li>Potato</li>
    <li>Radish</li>
  </ol>
</body>

</html>
```

Example

Following is an example where we used `<ol type = "A" >`

```
<!DOCTYPE html>
<html>

  <head>
    <title>HTML Ordered List</title>
  </head>

  <body>
    <ol type = "A">
      <li>Beetroot</li>
      <li>Ginger</li>
      <li>Potato</li>
      <li>Radish</li>
    </ol>
  </body>

</html>
```

This will produce the following result –

Example

Following is an example where we used `<ol type = "a">`

```
<!DOCTYPE html>
<html>

  <head>
    <title>HTML Ordered List</title>
  </head>

  <body>
    <ol type = "a">
      <li>Beetroot</li>
      <li>Ginger</li>
      <li>Potato</li>
      <li>Radish</li>
    </ol>
  </body>

</html>
```

The start Attribute

You can use **start** attribute for tag to specify the starting point of numbering you need. Following are the possible options –

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

Example

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

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

HTML Definition Lists

HTML and XHTML supports a list style which is called **definition lists** where entries are listed like in a dictionary or encyclopedia. The definition list is the ideal way to present a glossary, list of terms, or other name/value list.

Definition List makes use of following three tags.

- <dl> – Defines the start of the list
- <dt> – A term
- <dd> – Term definition
- </dl> – Defines the end of the list

Example

```
<!DOCTYPE html>  
<html>  
  
  <head>  
    <title>HTML Definition List</title>  
  </head>  
  
  <body>  
    <dl>  
      <dt><b>HTML</b></dt>  
      <dd>This stands for Hyper Text Markup Language</dd>  
      <dt><b>HTTP</b></dt>  
      <dd>This stands for Hyper Text Transfer Protocol</dd>
```

```
</dl>
</body>

</html>
```

HTML - Text Links

A webpage can contain various links that take you directly to other pages and even specific parts of a given page. These links are known as hyperlinks.

Hyperlinks allow visitors to navigate between Web sites by clicking on words, phrases, and images. Thus you can create hyperlinks using text or images available on a webpage.

Linking Documents

A link is specified using HTML tag `<a>`. This tag is called **anchor tag** and anything between the opening `<a>` tag and the closing `` tag becomes part of the link and a user can click that part to reach to the linked document.

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.

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.

- **Following is the simple syntax to use `<a>` tag.**

```
<a href = "Document URL" ... attributes-list>Link Text</a>
```

Example

Let's try following example which links <http://www.tutorialspoint.com> at your page –

```
<!DOCTYPE html>
<html>

  <head>
    <title>Hyperlink Example</title>
  </head>

  <body>
    <p>Click following link</p>
    <a href = "https://www.tutorialspoint.com" target = "_self">Tutorials Point</a>
  </body>

</html>
```

Linking to a Page Section

You can create a link to a particular section of a given webpage by using **name** attribute. This is a two-step process.

Note – The *name* attribute deprecated in HTML5. Do not use this attribute. Use *id* and *title* attribute instead.

First create a link to the place where you want to reach with-in a webpage and name it using <a...> tag as follows –

```
<h1>HTML Text Links <a name = "top"></a></h1>
```

Second step is to create a hyperlink to link the document and place where you want to reach –

```
<a href = "/html/html_text_links.htm#top">Go to the Top</a>
```

This will produce following link, where you can click on the link generated **Go to the Top** to reach to the top of the HTML Text Link tutorial.

[Go to the Top](#)

Setting Link Colors

You can set colors of your links, active links and visited links using **link**, **alink** and **vlink** attributes of <body> tag.

Example

Save the following in test.htm and open it in any web browser to see how **link**, **alink** and **vlink** attributes work.

```
<!DOCTYPE html>
<html>

  <head>
    <title>Hyperlink Example</title>
    <base href = "https://www.tutorialspoint.com/">
  </head>

  <body alink = "#54A250" link = "#040404" vlink = "#F40633">
    <p>Click following link</p>
    <a href = "/html/index.htm" target = "_blank" >HTML Tutorial</a>
  </body>

</html>
```

HTML - Image Links

We have seen how to create hypertext link using text and we also learnt how to use images in our webpages. Now, we will learn how to use images to create hyperlinks.

Example

It's simple to use an image as hyperlink. We just need to use an image inside hyperlink at the place of text as shown below –

```
<!DOCTYPE html>
<html>

  <head>
    <title>Image Hyperlink Example</title>
  </head>

  <body>
    <p>Click following link</p>
    <a href = "https://www.tutorialspoint.com" target = "_self">
      <img src = "/images/logo.png" alt = "Tutorials Point" border = "0"/>
    </a>
  </body>
```

</html>

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.

Disadvantages of Frames

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

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

Creating Frames

To use frames on a page we use <frameset> tag instead of <body> tag. The <frameset> tag defines, how to divide the window into frames. The **rows** attribute of <frameset> tag defines horizontal frames and **cols** attribute defines vertical frames. Each frame is indicated by <frame> tag and it defines which HTML document shall open into the frame.

Note – The <frame> tag deprecated in HTML5. Do not use this element.

Example

Following is the example to create three horizontal frames –

```
<!DOCTYPE html>
<html>

  <head>
    <title>HTML Frames</title>
  </head>

  <frameset rows = "10%,80%,10%">
    <frame name = "top" src = "/html/top_frame.htm" />
    <frame name = "main" src = "/html/main_frame.htm" />
    <frame name = "bottom" src = "/html/bottom_frame.htm" />

    <noframes>
      <body>Your browser does not support frames.</body>
    </noframes>

  </frameset>

</html>
```

Example

Let's put the above example as follows, here we replaced rows attribute by cols and changed their width. This will create all the three frames vertically –

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

```

<head>
  <title>HTML Frames</title>
</head>

<frameset cols = "25%,50%,25%">
  <frame name = "left" src = "/html/top_frame.htm" />
  <frame name = "center" src = "/html/main_frame.htm" />
  <frame name = "right" src = "/html/bottom_frame.htm" />

  <noframes>
    <body>Your browser does not support frames.</body>
  </noframes>
</frameset>

</html>

```

The <frameset> Tag Attributes

Following are important attributes of the <frameset> tag –

Sr.No	Attribute & Description
1	<p>cols</p> <p>Specifies how many columns are contained in the frameset and the size of each column. You can specify the width of each column in one of the four ways –</p> <p>Absolute values in pixels. For example, to create three vertical frames, use <i>cols = "100, 500, 100"</i>.</p> <p>A percentage of the browser window. For example, to create three vertical frames, use <i>cols = "10%, 80%, 10%"</i>.</p> <p>Using a wildcard symbol. For example, to create three vertical frames, use <i>cols = "10%, *, 10%"</i>. In this case wildcard takes remainder of the window.</p> <p>As relative widths of the browser window. For example, to create three vertical frames, use <i>cols = "3*, 2*, 1*"</i>. This is an alternative to percentages. You can use relative widths of the browser window. Here the window is divided into sixths: the first column takes up half of the window, the second takes one third, and the third takes one sixth.</p>
2	<p>rows</p> <p>This attribute works just like the cols attribute and takes the same values, but it is used to specify the rows in the frameset. For example, to create two horizontal frames, use <i>rows = "10%, 90%"</i>. You can specify the height of each row in the same way as explained above for columns.</p>

3	border This attribute specifies the width of the border of each frame in pixels. For example, border = "5". A value of zero means no border.
4	frameborder This attribute specifies whether a three-dimensional border should be displayed between frames. This attribute takes value either 1 (yes) or 0 (no). For example frameborder = "0" specifies no border.
5	framespacing This attribute specifies the amount of space between frames in a frameset. This can take any integer value. For example framespacing = "10" means there should be 10 pixels spacing between each frames.

The <frame> Tag Attributes

Following are the important attributes of <frame> tag –

Sr.No	Attribute & Description
1	src This attribute is used to give the file name that should be loaded in the frame. Its value can be any URL. For example, src = "/html/top_frame.htm" will load an HTML file available in html directory.
2	name This attribute allows you to give a name to a frame. It is used to indicate which frame a document should be loaded into. This is especially important when you want to create links in one frame that load pages into an another frame, in which case the second frame needs a name to identify itself as the target of the link.
3	frameborder This attribute specifies whether or not the borders of that frame are shown; it overrides the value given in the frameborder attribute on the <frameset> tag if one is given, and this can take values either 1 (yes) or 0 (no).
4	marginwidth

	This attribute allows you to specify the width of the space between the left and right of the frame's borders and the frame's content. The value is given in pixels. For example <code>marginwidth = "10"</code> .
5	marginheight This attribute allows you to specify the height of the space between the top and bottom of the frame's borders and its contents. The value is given in pixels. For example <code>marginheight = "10"</code> .
6	noresize By default, you can resize any frame by clicking and dragging on the borders of a frame. The <code>noresize</code> attribute prevents a user from being able to resize the frame. For example <code>noresize = "noresize"</code> .
7	scrolling This attribute controls the appearance of the scrollbars that appear on the frame. This takes values either "yes", "no" or "auto". For example <code>scrolling = "no"</code> means it should not have scroll bars.
8	longdesc This attribute allows you to provide a link to another page containing a long description of the contents of the frame. For example <code>longdesc = "framedescription.htm"</code>

Browser Support for Frames

If a user is using any old browser or any browser, which does not support frames then `<noframes>` element should be displayed to the user.

So you must place a `<body>` element inside the `<noframes>` element because the `<frameset>` element is supposed to replace the `<body>` element, but if a browser does not understand `<frameset>` element then it should understand what is inside the `<body>` element which is contained in a `<noframes>` element.

You can put some nice message for your user having old browsers. For example, *Sorry!! your browser does not support frames.* as shown in the above example.

Frame's name and target attributes

One of the most popular uses of frames is to place navigation bars in one frame and then load main pages into a separate frame.

Let's see following example where a `test.htm` file has following code –

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

```

<head>
  <title>HTML Target Frames</title>
</head>

<frameset cols = "200, *">
  <frame src = "/html/menu.htm" name = "menu_page" />
  <frame src = "/html/main.htm" name = "main_page" />

  <noframes>
    <body>Your browser does not support frames.</body>
  </noframes>
</frameset>

</html>

```

Here, we have created two columns to fill with two frames. The first frame is 200 pixels wide and will contain the navigation menu bar implemented by **menu.htm** file. The second column fills in remaining space and will contain the main part of the page and it is implemented by **main.htm** file. For all the three links available in menu bar, we have mentioned target frame as **main_page**, so whenever you click any of the links in menu bar, available link will open in main page.

Following is the content of menu.htm file

```

<!DOCTYPE html>
<html>

  <body bgcolor = "#4a7d49">
    <a href = "http://www.google.com" target = "main_page">Google</a>
    <br />
    <br />

    <a href = "http://www.microsoft.com" target = "main_page">Microsoft</a>
    <br />
    <br />

    <a href = "http://news.bbc.co.uk" target = "main_page">BBC News</a>
  </body>

</html>

```

Following is the content of main.htm file –

```

<!DOCTYPE html>
<html>

  <body bgcolor = "#b5dcb3">
    <h3>This is main page and content from any link will be displayed here.</h3>
    <p>So now click any link and see the result.</p>
  </body>

</html>

```

When we load **test.htm** file, it produces following result –

Now you can try to click links available in the left panel and see the result. The *targetattribute* can also take one of the following values –

Sr.No	Option & Description
-------	----------------------

1	_self Loads the page into the current frame.
2	_blank Loads a page into a new browser window. Opening a new window.
3	_parent Loads the page into the parent window, which in the case of a single frameset is the main browser window.
4	_top Loads the page into the browser window, replacing any current frames.
5	targetframe Loads the page into a named targetframe.

HTML - Fonts

Fonts play a very important role in making a website more user friendly and increasing content readability. Font face and color depends entirely on the computer and browser that is being used to view your page but you can use HTML **** tag to add style, size, and color to the text on your website. You can use a **<basefont>** tag to set all of your text to the same size, face, and color.

The font tag is having three attributes called **size**, **color**, and **face** to customize your fonts. To change any of the font attributes at any time within your webpage, simply use the **** tag. The text that follows will remain changed until you close with the **** tag. You can change one or all of the font attributes within one **** tag.

Note –The *font* and *basefont* tags are deprecated and it is supposed to be removed in a future version of HTML. So they should not be used rather, it's suggested to use CSS styles to manipulate your fonts. But still for learning purpose, this chapter will explain font and basefont tags in detail.

Set Font Size

You can set content font size using **size** attribute. The range of accepted values is from 1(smallest) to 7(largest). The default size of a font is 3.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Setting Font Size</title>
```

```
</head>
```

```
<body>
```

```
<font size = "1">Font size = "1"</font><br />
```

```
<font size = "2">Font size = "2"</font><br />
```

```
<font size = "3">Font size = "3"</font><br />
```

```
<font size = "4">Font size = "4"</font><br />
```

```
<font size = "5">Font size = "5"</font><br />
```

```
<font size = "6">Font size = "6"</font><br />
<font size = "7">Font size = "7"</font>
</body>
```

</html>

Setting Font Face

You can set font face using *face* attribute but be aware that if the user viewing the page doesn't have the font installed, they will not be able to see it. Instead user will see the default font face applicable to the user's computer.

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Font Face</title>
  </head>

  <body>
    <font face = "Times New Roman" size = "5">Times New Roman</font><br />
    <font face = "Verdana" size = "5">Verdana</font><br />
    <font face = "Comic sans MS" size = " 5">Comic Sans MS</font><br />
    <font face = "WildWest" size = "5">WildWest</font><br />
    <font face = "Bedrock" size = "5">Bedrock</font><br />
  </body>

</html>
```

Setting Font Color

You can set any font color you like using *color* attribute. You can specify the color that you want by either the color name or hexadecimal code for that color.

Example

```
<!DOCTYPE html>
<html>

  <head>
    <title>Setting Font Color</title>
  </head>

  <body>
    <font color = "#FF00FF">This text is in pink</font><br />
    <font color = "red">This text is red</font>
  </body>

</html>
```

HTML – Forms

HTML Forms are required to collect different kinds of user inputs, such as contact details like name, email address, phone numbers, or details like credit card information, etc.

Forms contain special elements called controls like inputbox, checkboxes, radio-buttons, submit buttons, etc. Users generally complete a form by modifying its controls e.g. entering text, selecting items, etc. and submitting this form to a web server for further processing.

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

The `<input>` Element

The HTML `<input>` element is the most used form element.

An `<input>` element can be displayed in many ways, depending on the type attribute.

Here are some examples:

Type	Description
<code><input type="text"></code>	Displays a single-line text input field
<code><input type="radio"></code>	Displays a radio button (for selecting one of many choices)
<code><input type="checkbox"></code>	Displays a checkbox (for selecting zero or more of many choices)
<code><input type="submit"></code>	Displays a submit button (for submitting the form)
<code><input type="button"></code>	Displays a clickable 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>

```

This will produce the following result –

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>

```

This will produce the following result –

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 <i>checked</i> 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 <i>checked</i> 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>
```

This will produce the following result –

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>
```

This will produce the following result –

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>

```

Frequently Used Form Attributes

The following table lists the most frequently used form element's attributes:

Attribute	Description
name	Specifies the name of the form.
action	Specifies the URL of the program or script on the web server that will be used for processing the information submitted via form.
method	Specifies the HTTP method used for sending the data to the web server by the browser. The value can be either get (the default) and post.
target	Specifies where to display the response that is received after submitting the form. Possible values are _blank, _self, _parent and _top.
enctype	Specifies how the form data should be encoded when submitting the form to the server. Applicable only when the value of the method attribute is post.

HTML - Embed Multimedia

Sometimes you need to add music or video into your web page. The easiest way to add video or sound to your web site is to include the special HTML tag called **<embed>**. This tag causes the browser itself to include controls for the multimedia automatically provided browser supports <embed> tag and given media type.

You can also include a **<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.

Example

Here is a simple example to play an embedded midi file –

```

<!DOCTYPE html>
<html>

  <head>
    <title>HTML embed Tag</title>
  </head>

  <body>
    <embed src = "/html/yourfile.mid" width = "100%" height = "60" >
      <noembed><img src = "yourimage.gif" alt = "Alternative Media" ></noembed>
    </embed>
  </body>

</html>

```

The <embed> Tag Attributes

Following is the list of important attributes which can be used with <embed> tag.

Note –The *align* and *autostart* attributes deprecated in HTML5. Do not use these attributes.

Sr.No	Attribute & Description
1	align Determines how to align the object. It can be set to either center, <i>left</i> or <i>right</i> .
2	autostart 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 <i>loop</i> if you are using 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.

- **.swf files** – are the file types created by Macromedia's Flash program.
- **.wmv files** – are Microsoft's Window's Media Video file types.
- **.mov files** – are Apple's Quick Time Movie format.
- **.mpeg files** – are movie files created by the Moving Pictures Expert Group.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>HTML embed Tag</title>
```

```
</head>
```

```
<body>
```

```
<embed src = "/html/yourfile.swf" width = "200" height = "200" >
```

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

```
</embed>
```

```
</body>
```

```
</html>
```

Background Audio

You can use HTML **<bgsound>** tag to play a soundtrack in the background of your webpage. This tag is supported by Internet Explorer only and most of the other browsers ignore this tag. It downloads and plays an audio file when the host document is first downloaded by the user and displayed. The background sound file also will replay whenever the user refreshes the browser.

Note – The bgsound tag is deprecated and it is supposed to be removed in a future version of HTML. So they should not be used rather, it's suggested to use HTML5 tag audio for adding sound. But still for learning purpose, this chapter will explain bgsound tag in detail.

This tag is having only two attributes *loop* and *src*. Both these attributes have same meaning as explained above.

Here is a simple example to play a small midi file –

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>HTML embed Tag</title>
```

```
</head>
```

```
<body>
```

```
<bgsound src = "/html/yourfile.mid">
```

```
<noembed><img src = "yourimage.gif" ></noembed>
```

```
</bgsound>
```

```
</body>
```

```
</html>
```

This will produce the blank screen. This tag does not display any component and remains hidden.

Internet Explorer can also handle only three different sound format files – wav, the native format for PCs; au, the native format for most Unix workstations; and MIDI, a universal music-encoding scheme.

HTML - Marquees

An HTML marquee is a scrolling piece of text displayed either horizontally across or vertically down your webpage depending on the settings. This is created by using HTML <marquee> tag.

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.

Syntax

A simple syntax to use HTML <marquee> tag is as follows –

```
<marquee attribute_name = "attribute_value"....more attributes>
```

```
    One or more lines or text message or image  
</marquee>
```

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. Sets the interval between each scroll movement in milliseconds. The default value is 85. Note that any value smaller than 60 is ignored and the value 60 is used instead, unless truespeed is specified.
scrollamount	defines scroll amount in number. The default value is 6
loop	defines loop for marquee content in number.
bgcolor	defines background color. It is now <i>deprecated</i> .

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.

```
<marquee width="100%" behavior="scroll" bgcolor="pink">  
This is an example of a scroll marquee...  
</marquee>
```

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.

```
<marquee width="100%" behavior="slide" bgcolor="pink">  
This is an example of a slide marquee...  
</marquee>
```

HTML Alternate Marquee

It scrolls the text from right to left and goes back left to right.

```
<marquee width="100%" behavior="alternate" bgcolor="pink">  
This is an example of a alternate marquee...  
</marquee>
```

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.

```
<marquee width="100%" direction="right">  
This is an example of a right direction marquee...  
</marquee>
```

Nested marquee example

```
<marquee width="400px" height="100px" behavior="alternate" style="border:2px solid red">  
<marquee behavior="alternate">  
Nested marquee...  
</marquee>  
</marquee>
```

Disadvantages HTML marquee

- Marquee may be distracting because human eyes are attracted towards movement and marquee text constantly.
- Since Marquee text moves, so it is more difficult to click static text, depending on the scrolling speed.
- It is a non-standard HTML element.
- It draws user's attention needlessly and makes the text harder to read.

Publishing HTML Documents

Everything in this tutorial so far has been focussed on creating HTML documents. Now we're going to talk about *publishing* your HTML documents.

Saving your HTML Files

Before we discuss publishing your documents, let's talk about *saving* your documents.

HTML documents are normally saved with a `.html` or `.htm` extension. No special software is required to create an HTML file. You could use a simple text editor for example. However, there are specialized options such as Adobe Dreamweaver that make it easier to create HTML documents.

However, if you need to use a server-side scripting language (such as PHP or ColdFusion), you will need to save your files with the appropriate extension for that language. So you will typically save a PHP file with a `.php` extension and a ColdFusion file with a `.cfm` extension. Your HTML code will still work because these scripting languages were created with HTML in mind. These files will usually contain a mixture of HTML and server-side code.

Typically, if you save a file as `index.html`, `index.htm`, `index.php`, `index.cfm`, `default.html`, or `default.htm`, this will make it the default file for the directory that it's located in. This means that if someone navigates to a directory without specifying the file name, that file will be displayed.

For example, if you have a document located at `http://example.com/html/index.html`, the user could access that file by typing in either of the following:

```
http://example.com/html/index.html
```

```
http://example.com/html/
```

```
http://example.com/html
```

However, this depends on how the web server is configured. It could be configured to honor `.php` files over `.html` files, in which case, if there's a `http://example.com/html/index.php` file in the same directory, the user could only use the first URL above (`http://example.com/html/index.html`). This is because, typing the following would produce the PHP file:

```
http://example.com/html/index.php
```

```
http://example.com/html/
```

```
http://example.com/html
```

Most web hosting companies allow you to configure this to suit your own website. So you can decide which document names are the default pages for your directories.

Publishing Methods

There are various methods of publishing HTML documents but here are some of the most common methods:

Blogging platform

Blogging platforms usually allow you to create "blog posts" by logging in to an administration area and clicking a button such as New Post or similar. They usually provide an HTML editor that allows you to add HTML elements without needing to know HTML. All HTML code is automatically generated as you use the WYSIWYG editor. But you can usually switch to HTML View or similar if you prefer to add your own HTML code. Once a post is ready to go live, simply click a button called Publish or

similar. Using this method, all blog posts are stored as data in a database, so there is no need to create a .html document and save it to your local computer.

Online website builder

Similar to using a blogging platform, you simply log in to the admin area, create articles, then publish them when they're ready. Online website builders usually allow you to select from a range of templates, add images, and edit articles using a WYSIWYG editor.

Content Management System (CMS)

This is similar to using an online website builder. In fact, an online website builder basically *is* a content management system. The reason I separated this one is because, many organizations have their own CMS that staff can use to update their website. This may have been custom built for their organization or it could be an "off the shelf" product.

FTP your HTML document

Upload your HTML document to the "live" server. In this scenario, you have a copy of your website on your local computer, and a copy located on a "live" (or "production") server. The live server is located with your web hosting provider (although you could host it yourself if you choose). Typically, you upload files to your hosting provider via FTP (File Transfer Protocol) using FTP software (referred to as an *FTP Client*) such as FileZilla. You can't use this method if your articles are stored in a database. You can only use it to upload files from a file system.

Modify the file directly on the server

This is not recommended, however, it is certainly possible. If you have full access to the live server, there's nothing to stop you editing the files on your website directly. You can double-click on a file, edit it, save it, and your website will be updated with the new change. Just as you would editing a file on your own computer. However, it is better practice to use FTP to edit your website in this case.

Self-Hosted Websites

Most websites/blogs are hosted with a third party hosting company (regardless of which publishing method/s they use). Hosting is a 24/7 job and it's usually better to leave it to the experts (assuming your hosting company are experts!). However, it is possible to host your own website—after all, you are connected to the internet just like any hosting provider. To do this, you'll need to know how web servers work and you'll need to be well versed in internet security.

The method you choose to publish your HTML documents will largely be dictated by whether you use a CMS, blogging system, online website builder, or just a simple web hosting account.