HTML Tutorial

HTML stands for **H**yper **T**ext **M**arkup **L**anguage, which is the most widely used language on Web to develop web pages.

Hypertext refers to the way in which Web pages (HTML documents) are linked together.

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.

Example: 1

```
<!DOCTYPE html>
<html>
<body>
  <h1>Good Morning</h1>
</body>
</html>
```

Example: 2

```
<!DOCTYPE html>
<html>
<head>
<title>Our First Title</title>
</head>
<body>
<h1>This is a heading</h1>
<p>Our First Paragraph</p>
</body>
</html>
```

HTML Tags

HTML is a markup language which 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.

Tag	Description
<!DOCTYPE...>	This tag defines the document type and HTML version.
<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.
<head>	This tag represents the document's header which can keep other HTML tags like <title>, <link> etc.

<title>	The **<title>** tag is used inside the <head> tag to mention the document title.
<body>	This tag represents the document's body which keeps other HTML tags like <h1>, , etc.
<h1>	This tag represents the heading.
	This tag represents a paragraph.

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 : 3

```
<!DOCTYPE html>
<html>
<head>
<title>Heading Example</title>
</head>
<body>
<h1>KANGKAN</h1>
<h2>SUBROT</h2>
<h3>BAGASHREE</h3>
<h4>RUBEEKA</h4>
<h5>JYOTI</h5>
<h6>IMRAN</h6>
</body>
</html>
```

Paragraph Tag

The ** ** tag offers a way to structure your text into different paragraphs. Each paragraph of text should go in between an opening and a closing 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.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Paragraph Example</title>
</head>
<body>
<p>Here is a first paragraph <br/>of text.</p>
<p>Here is a second paragraph of text.</p>
<p>Here is a third paragraph of text.</p>
</body>
</html>
```

Centering Content

You can use ** ** 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.

```
<!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.

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
	This is paragraph content.	
<h1>	This is heading content.	</h1>
	This is division content.	

So here .... 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 consist 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 is starting tag of a paragraph and is closing tag of the same paragraph but ** This is paragraph ** 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 Attributes

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

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

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

```
<!DOCTYPE html>
<html>
<head>
<title>Align Attribute  Example</title>
</head>
<body>
<p align="left">This is left aligned</p>
<p align="center">This is center aligned</p>
<p align="right">This is right aligned</p>
</body>
</html>
```

Attribute	Options	Function
Align	right, left, center	Horizontally aligns tags

Valign	top, middle, bottom	Vertically aligns tags within an HTML element.
bgcolor	numeric, hexidecimal, RGB values	Places a background color behind an element
background	URL	Places a background image behind an element
Id	User Defined	Names an element for use with Cascading Style Sheets.
Class	User Defined	Classifies an element for use with Cascading Style Sheets.
Width	Numeric Value	Specifies the width of tables, images, or table cells.
Height	Numeric Value	Specifies the height of tables, images, or table cells.
Title	User Defined	"Pop-up" title of the elements.

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>Rubeeka</b> typeface.</p>
</body>
</html>
```

Italic Text

Anything that appears within **...** 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 a <i>italicized</i> typeface.</p>
</body>
</html>
```

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 a <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>
```

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 <sup>superscript</sup> 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>
```

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>
```

Grouping Content

The ** ** and **** elements allow you to group together several elements to create sections or subsections of a page.

```
<!DOCTYPE html>
<html>
<head>
<title>Div Tag Example</title>
</head>
<body>
<div id="menu" align="middle" >
<a href="/index.htm">HOME</a> |
<a href="/about/contact_us.htm">CONTACT</a> |
<a href="/about/index.htm">ABOUT</a>
</div>

<div id="content" align="left" bgcolor="white">
<h5>Content Articles</h5>
<p>Actual content goes here.....</p>
</div>
</body>
</html>
```

The element feeds on a whole line (if its width and height is not fixed) whereas the element, on the other hand, can be used to group inline elements only.

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>
```

Text Abbreviation

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

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Text Abbreviation</title>
</head>
<body>
<p>My best friend's name is  <abbr title="Abhishek">ggg</abbr>.</p>
</body>
</html>
```

HTML Metadata

HTML lets you specify metadata - additional important information about a document in a variety of ways. The META elements can be used to include name/value pairs describing properties of the HTML document, such as author, expiry date, a list of keywords, document author etc.

The **<meta>** tag is used to provide such additional information. This tag is an empty element and so does not have a closing tag but it carries information within its attributes.

You can include one or more meta tags in your document based on what information you want to keep in your document but in general, meta tags do not impact physical appearance of the document so from appearance point of view, it does not matter if you include them or not.

Adding Meta Tags to Your Documents

You can add metadata to your web pages by placing <meta> tags inside the header of the document which is represented by **<head>** and **</head>** tags. A meta tag can have following attributes in addition to core attributes:

Attribute	Description
Name	Name for the property. Can be anything. Examples include, keywords, description, author, revised, generator etc.
content	Specifies the property's value.
scheme	Specifies a scheme to interpret the property's value (as declared in the content attribute).
http-equiv	Used for http response message headers. For example http-equiv can be used to refresh the page or to set a cookie. Values include content-type, expires, refresh and set-cookie.

Specifying Keywords

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

Example

Following is an example where we are adding HTML, Meta Tags, Metadata as important keywords about the document.

```
<!DOCTYPE html>
<html>
<head>
<title>Meta Tags Example</title>
<meta name="keywords" content="HTML, Meta Tags, Metadata" />
</head>
<body>
<p>Hello HTML5!</p>
</body>
</html>
```

This will produce following result:

Hello HTML5!

Document Description

You can use <meta> tag to give a short description about the document. This again can be used by various search engines while indexing your webpage for searching purpose.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Meta Tags Example</title>
<meta name="keywords" content="HTML, Meta Tags, Metadata" />
<meta name="description" content="Learning about Meta Tags." />
</head>
<body>
<p>Hello HTML5!</p>
</body>
</html>
```

Document Revision Date

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

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Meta Tags Example</title>
<meta name="keywords" content="HTML, Meta Tags, Metadata" />
<meta name="description" content="Learning about Meta Tags." />
<meta name="revised" content="Version KU-001" />
</head>
<body>
<p>Hello KU</p>
</body>
</html>
```

Document Refreshing

A <meta> tag can be used to specify a duration after which your web page will keep refreshing automatically.

Example

If you want your page keep refreshing after every 5 seconds then use the following syntax.

```
<!DOCTYPE html>
<html>
<head>
<title>Meta Tags Example</title>
<meta name="keywords" content="HTML, Meta Tags, Metadata" />
<meta name="description" content="Learning about Meta Tags." />
<meta name="revised" content="KU version 234" />
<meta http-equiv="refresh" content="5" />
</head>
<body>
<p>Hello HTML5!</p>
</body>
</html>
```

Page Redirection

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

Example

Following is an example of redirecting current page to another page after 5 seconds. If you want to redirect page immediately then do not specify *content* attribute.

```
<!DOCTYPE html>
<html>
<head>
<title>Meta Tags Example</title>
<meta name="keywords" content="HTML, Meta Tags, Metadata" />
<meta name="description" content="Learning about Meta Tags." />
<meta name="revised" content="KU Version 0001" />
<meta http-equiv="refresh" content="5; url=http://www.kazirangauniversity.in" />
</head>
<body>
<p>Hello HTML5!</p>
</body>
</html>
```

Setting Author Name

You can set an author name in a web page using meta tag. See an example below:

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Meta Tags Example</title>
```

```
<meta name="keywords" content="HTML, Meta Tags, Metadata" />
<meta name="description" content="Learning about Meta Tags." />
<meta name="author" content="Mr. XYZ " />
</head>
<body>
<p>Hello HTML5!</p>
</body>
</html>
```

HTML Comment

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>
```

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="./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 our 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="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="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="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="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.

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 **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.

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>
```

Cellpadding and Cellspacing Attributes

There are two attribiutes called *cellpadding* and *cellspacing* which you will use to adjust the white space in your table cells. The cellspacing attribute defines the width of the border, 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 cellpadding="50" >Ramesh Raman</td>
<td>5000</td>
</tr>
<tr>
<td>Shabbir Hussein</td>
<td>7000</td>
</tr>
```

```
</table>
</body>
</html>
```

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.

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>
```

Table Height and Width

You can set a table width and height using **width** and **height** attrubutes. 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>
```

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:

- **<ul>** - An unordered list. This will list items using plain bullets.
- **<ol>** - 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 **<ul>** 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 <ul> 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>
```

HTML Definition Lists

HTML and XHTML support 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 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 **<ol>** tag. The numbering starts at one and is incremented by one for each successive ordered list element tagged with <li>.

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

The start Attribute

You can use **start** attribute for <ol> 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.

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

Link Text

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="http://www.tutorialspoint.com" target="_self">Tutorials Point</a>
</body>
</html>
```

The target Attribute

We have used **target** attribute in our previous example. This attribute is used to specify the location where linked document is opened. Following are possible options:

Option	Description
_blank	Opens the linked document in a new window or tab.
_self	Opens the linked document in the same frame.
_parent	Opens the linked document in the parent frame.
_top	Opens the linked document in the full body of the window.
targetframe	Opens the linked document in a named *targetframe*.

Example

Try following example to understand basic difference in few options given for target attribute.

```
<!DOCTYPE html>
<html>
<head>
<title>Hyperlink Example</title>
<base href="http://www.tutorialspoint.com/">
</head>
<body>
<p>Click any of the following links</p>
<a href="/html/index.htm" target="_blank">Opens in New</a> |
<a href="/html/index.htm" target="_self">Opens in Self</a> |
<a href="/html/index.htm" target="_parent">Opens in Parent</a> |
<a href="/html/index.htm" target="_top">Opens in Body</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.

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.

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="http://www.tutorialspoint.com/">
</head>
<body alink="#green" link="#white" vlink="#red">
<p>Click following link</p>
<a href="/html/index.htm" target="_blank" >HTML Tutorial</a>
</body>
</html>
```

Download Links

You can create text link to make your PDF, or DOC or ZIP files downloadable. This is very simple, you just need to give complete URL of the downloadable file as follows:

```
<!DOCTYPE html>
<html>
<head>
<title>Hyperlink Example</title>
</head>
<a href="http://www.tutorialspoint.com/page.pdf">Download PDF File</a>
</body>
</html>
```

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="http://www.tutorialspoint.com" target="_self">
  <img src="/images/logo.png" alt="Tutorials Point" border="0"/>
</a>
</body>
</html>
```

HTML Email Tag

HTML <a> tag provides you option to specifiy an email address to send an email. While using <a> tag as an email tag, you will use **mailto:email address** along with *href* attribute. Following is the syntax of using **mailto** instead of using http.

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

This code will generate following link which you can use to send email.

Send Email

Now if a user clicks this link, it launches one Email Client (like Lotus Notes, Outlook Express etc.) installed on your user's computer. There is another risk to use this option to send email because if user do not have email client installed on their computer then it would not be possible to send email.

Default Settings

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

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

This code will generate following link which you can use to send email.

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.

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 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:

Attribute	Description
cols	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 four ways: • Absolute values in pixels. For example to create three vertical frames, use *cols="100, 500,100"*. • A percentage of the browser window. For example to create three vertical frames, use *cols="10%, 80%,10%"*. • Using a wildcard symbol. For example to create three vertical frames, use *cols="10%, *,10%"*. In this case wildcard takes remainder of the window. • As relative widths of the browser window. For example to create three vertical frames, use *cols="3*,2*,1*"*. 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.
rows	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 *rows="10%, 90%"*. You can specify the height of each row in the same way as explained above for columns.
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.
frameborder	This attribute specifies whether a three-dimensional border should be displayed between frames. This attrubute takes value either 1 (yes) or 0 (no). For example frameborder="0" specifies no border.
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 important attributes of <frame> tag:

Attribute	Description
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.
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.
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).
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 marginwidth="10".
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 marginheight="10".
noresize	By default you can resize any frame by clicking and dragging on the borders of a frame. The noresize attribute prevents a user from being able to resize the frame. For example noresize="noresize".
scrolling	This attribute controls the appearance of the scrollbars that appear on the frame. This takes values either "yes", "no" or "auto". For example scrolling="no" means it should not have scroll bars.
longdesc	This attribute allows you to provide a link to another page containing a long description of the contents of the frame. For example longdesc="framedescription.htm"

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 menubar 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 menubar, we have mentioned target frame as **main_page**, so whenever you click any of the links in menubar, 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>
So now click any link and see the result.
</body>
</html>

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

Option	Description
_self	Loads the page into the current frame.
_blank	Loads a page into a new browser window.opening a new window.
_parent	Loads the page into the parent window, which in the case of a single frameset is the main browser window.
_top	Loads the page into the browser window, replacing any current frames.
targetframe	Loads the page into a named targetframe.

All the HTML elements can be categorized into two categories **(a)** Block Level Elements **(b)** Inline Elements

Block Elements

Block elements appear on the screen as if they have a line break before and after them. For example the , <h1>, <h2>, <h3>, <h4>, <h5>, <h6>, <ul>, <ol>, <dl>, <pre>, <hr />, <blockquote>, and <address> elements are all block level elements. They all start on their own new line, and anything that follows them appears on its own new line.

Inline Elements

Inline elements, on the other hand, can appear within sentences and do not have to appear on a new line of their own. The <b>, <i>, <u>, , , <sup>, <sub>, <big>, <small>, <li>, <ins>, <del>, <code>, <cite>, <dfn>, <kbd>, and <var> elements are all inline elements.

Grouping HTML Elements

There are two important tags which we use very frequently to group various other HTML tags (i) tag and (ii) tag

The tag

This is the very important block level tag which plays a big role in grouping various other HTML tags and applying CSS on group of elements. Even now tag can be used to create webpage layout where we

define different parts (Left, Right, Top etc) of the page using tag. This tag does not provide any visual change on the block but this has more meaning when it is used with CSS.

Example

Following is a simple example of tag. We will learn Cascading Style Sheet (CSS) in a separate chapter but we used it here to show the usage of tag:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML div Tag</title>
</head>
<body>

<!-- First group of tags -->
<div style="color:red">
  <h4>This is first group</h4>
  <p>Following is a list of vegetables</p>
  <ul>
  <li>Beetroot</li>
  <li>Ginger</li>
  <li>Potato</li>
  <li>Radish</li>
  </ul>
</div>

<!-- Second group of tags -->
<div style="color:green">
  <h4>This is second group</h4>
  <p>Following is a list of fruits</p>
  <ul>
  <li>Apple</li>
  <li>Banana</li>
  <li>Mango</li>
  <li>Strawberry</li>
  </ul>
</div>

</body>
</html>
```

The tag

The HTML is an inline element and it can be used to group inline-elements in an HTML document. This tag also does not provide any visual change on the block but has more meaning when it is used with CSS.

The difference between the tag and the tag is that the tag is used with inline elements where as the tag is used with block-level elements.

Example

Following is a simple example of tag. We will learn Cascading Style Sheet (CSS) in a separate chapter but we used it here to show the usage of tag:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML span Tag</title>
</head>
<body>

<p>This is <span style="color:red">red</span> and this is <span style="color:green">green</span></p>

</body>
</html>
```

By default, your webpage background is white in color. You may not like it, but no worries. HTML provides you following two good ways to decorate your webpage background.

- Html Background with Colors
- Html Background with Images

Now let's see both the approaches one by one using appropriate examples.

Html Background with Colors

The **bgcolor** attribute is used to control the background of an HTML element, specifically page body and table backgrounds. Following is the syntax to use bgcolor attribute with any HTML tag.

```
<tagname bgcolor="color_value"...>
```

This color_value can be given in any of the following formats:

```
<!-- Format 1 - Use color name -->
<table bgcolor="lime" >
```

```
<!-- Format 2 - Use hex value -->
<table bgcolor="#f1f1f1" >

<!-- Format 3 - Use color value in RGB terms -->
<table bgcolor="rgb(0,0,120)" >
```

Example

Here are the examples to set background of an HTML tag:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Background Colors</title>
</head>
<body>

<!-- Format 1 - Use color name -->
<table bgcolor="yellow" width="100%">
<tr><td>
This background is yellow
</td></tr>
</table>

<!-- Format 2 - Use hex value -->
<table bgcolor="#6666FF" width="100%">
<tr><td>
This background is sky blue
</td></tr>
</table>

<!-- Format 3 - Use color value in RGB terms -->
<table bgcolor="rgb(255,0,255)"  width="100%">
<tr><td>
This background is green
</td></tr>
</table>

</body>
</html>
```

Html Background with Images

The **background** attribute can also be used to control the background of an HTML elmement, specifically page body and table backgrounds. You can specify an image to set background of your HTML page or table. Following is the syntax to use background attribute with any HTML tag.

Note: The *background* attribute is deprecated and it is recommended to use Style Sheet for background setting.

<tagname background="Image URL"...>

The most frequently used image formats are JPEG, GIF and PNG images.

Example

Here are the examples to set background images of a table.

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Background Images</title>
</head>
<body>

<!-- Set table background -->
<table background="/images/html.gif" width="100%" height="100">
<tr><td>
This background is filled up with HTML image.
</td></tr>
</table>

</body>
</html>
```

Patterned & Transparent Backgrounds

You might have seen many pattern or transparent backgrounds on various websites. This simply can be achieved by using patterned image or transparent image in the background.

It is suggested that while creating patterns or transparent GIF or PNG images, use the smallest dimensions possible even as small as 1x1 to avoid slow loading.

Example

Here are the examples to set background pattern of a table:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Background Images</title>
</head>
<body>

<!-- Set a table background using pattrern -->
<table background="/images/pattern1.gif" width="100%" height="100">
<tr><td>
This background is filled up with a pattern image.
</td></tr>
</table>

<!-- Another example on table background using pattrern -->
<table background="/images/pattern2.gif" width="100%" height="100">
<tr><td>
This background is filled up with a pattern image.
</td></tr>
</table>

</body>
</html>
```

HTML Colors

Colors are very important to give a good look and feel to your website. You can specify colors on page level using <body> tag or you can set colors for individual tags using **bgcolor** attribute.

The <body> tag has following attributes which can be used to set different colors:

- **bgcolor -** sets a color for the background of the page.
- **text -** sets a color for the body text.
- **alink -** sets a color for active links or selected links.
- **link -** sets a color for linked text.
- **vlink -** sets a color for *visited links* - that is, for linked text that you have already clicked on.

HTML Color Coding Methods

There are following three different methods to set colors in your web page:

- **Color names -** You can specify color names directly like green, blue or red.
- **Hex codes -** A six-digit code representing the amount of red, green, and blue that makes up the color.
- **Color decimal or percentage values -** This value is specified using the rgb( ) property.

Now we will see these coloring schemes one by one.

HTML Colors - Color Names

You can sepecify direct a color name to set text or background color. W3C has listed 16 basic color names that will validate with an HTML validator but there are over 200 different color names supported by major browsers.

Example

Here are the examples to set background of an HTML tag by color name:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Colors by Name</title>
</head>
<body text="blue" bgcolor="green">
<p>Use different color names for for body and table and see the result.</p>
<table bgcolor="black">
<tr>
<td>
<font color="white">This text will appear white on black background.</font>
</td>
</tr>
</table>
</body>
</html>
```

HTML Colors - Hex Codes

A hexadecimal is a 6 digit representation of a color. The first two digits(RR) represent a red value, the next two are a green value(GG), and the last are the blue value(BB).

Example

Here are the examples to set background of an HTML tag by color code in hexadecimal:

```
<!DOCTYPE html>
```

```
<html>
<head>
<title>HTML Colors by Hex</title>
</head>
<body text="#0000FF" bgcolor="#00FF00">
<p>Use different color hexa for for body and table and see the result.</p>
<table bgcolor="#000000">
<tr>
<td>
<font color="#FFFFFF">This text will appear white on black background.</font>
</td>
</tr>
</table>
</body>
</html>
```

HTML Colors - RGB Values

This color value is specified using the **rgb( )** property. This property takes three values, one each for red, green, and blue. The value can be an integer between 0 and 255 or a percentage.

Example

Here are the examples to set background of an HTML tag by color code using rgb() values:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Colors by RGB code</title>
</head>
<body text="rgb(0,0,255)" bgcolor="rgb(0,255,0)">
<p>Use different color code for for body and table and see the result.</p>
<table bgcolor="rgb(0,0,0)">
<tr>
<td>
<font color="rgb(255,255,255)">This text will appear white on black background.</font>
</td>
</tr>
</table>
</body>
</html>
```

HTML Fonts

Fonts play 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 **<font>** tag to add style, size, and color to the text on your website.

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.

Example

```
<!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>
```

Relative Font Size

You can specify how many sizes larger or how many sizes smaller than the preset font size should be. You can specify it like **<font size="+n"> or <font size="-n">**

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Relative Font Size</title>
</head>
<body>
<font size="-1">Font size="-1"</font><br />
<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>
</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 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:

Attribute	Description
action	Backend script ready to process your passed data.
method	Method to be used to upload data. The most frequently used are GET and POST methods.
target	Specify the target window or frame where the result of the script will be displayed. It takes values like _blank, _self, _parent etc.
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.

Note: You can refer to **Perl & CGI** for a detail on how form data upload works.

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.

Attribute	Description

Type	Indicates the type of input control and for text input control it will be set to **text**.
Name	Used to give a name to the control which is sent to the server to be recognized and get the value.
Value	This can be used to provide an initial value inside the control.
Size	Allows to specify the width of the text-input control in terms of characters.
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.

Attribute	**Description**

Type	Indicates the type of input control and for password input control it will be set to **password**.
Name	Used to give a name to the control which is sent to the server to be recognized and get the value.
Value	This can be used to provide an initial value inside the control.
Size	Allows to specify the width of the text-input control in terms of characters.
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.

Attribute	**Description**

Name	Used to give a name to the control which is sent to the server to be recognized and get the value.
Rows	Indicates the number of rows of text area box.
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.

Attribute	Description
Type	Indicates the type of input control and for checkbox input control it will be set to **checkbox**.
Name	Used to give a name to the control which is sent to the server to be recognized and get the value.

Value	The value that will be used if the checkbox is selected.
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.

Attribute	Description
Type	Indicates the type of input control and for checkbox input control it will be set to **radio**.
Name	Used to give a name to the control which is sent to the server to be recognized and get the value.
Value	The value that will be used if the radio box is selected.
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:

Attribute	Description
Name	Used to give a name to the control which is sent to the server to be recognized and get the value.
Size	This can be used to present a scrolling list box.
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:

Attribute	Description
Value	The value that will be used if an option in the select box box is selected.
Selected	Specifies that this option should be the initially selected value when the page loads.

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:

Attribute	Description
Name	Used to give a name to the control which is sent to the server to be recognized and get the value.
Accept	Specifies the types of files that the server accepts.

Button Controls

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:

Type	Description
submit	This creates a button that automatically submits a form.

reset	This creates a button that automatically resets form controls to their initial values.
button	This creates a button that is used to trigger a client-side script when the user clicks that button.
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 has 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>
```

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.

```
<!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>
```

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 <marquees> tag.

Note: The HTML <marquee> tag may not be supported by various browsers so its not recommended to rely on this tag, instead you can use Javascript and CSS to create such effects.

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>

The <marquee> Tag Attributes

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

Attribute	Description
Width	This specifies the width of the marquee. This can be a value like 10 or 20% etc.
Height	This specifies the height of the marquee. This can be a value like 10 or 20% etc.
direction	This specifies the direction in which marquee should scroll. This can be a value like *up*, *down*, *left* or *right*.
behavior	This specifies the type of scrolling of the marquee. This can have a value like *scroll*, *slide* and *alternate*.
scrolldelay	This specifies how long to delay between each jump. This will have a value like 10 etc.
scrollamount	This specifies the speed of marquee text. This can have a value like 10 etc.
Loop	This specifies how many times to loop. The default value is INFINITE, which means that the marquee loops endlessly.
Bgcolor	This specifies background color in terms of color name or color hex value.
Hspace	This specifies horizontal space around the marquee. This can be a value like 10 or 20% etc.
Vspace	This specifies vertical space around the marquee. This can be a value like 10 or 20% etc.

```
<!DOCTYPE html>
<html>
<head>
<title>HTML marquee Tag</title>
</head>
<body>
<marquee>This is basic example of marquee</marquee>
</body>
</html>
```

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

```
<head>
<title>HTML marquee Tag</title>
</head>
<body>
<marquee width="50%">This example will take only 50% width</marquee>
</body>
</html>


<!DOCTYPE html>
<html>
<head>
<title>HTML marquee Tag</title>
</head>
<body>
<marquee direction="right">This text will scroll from left to right</marquee>
</body>
</html>


<!DOCTYPE html>
<html>
<head>
<title>HTML marquee Tag</title>
</head>
<body>
<marquee direction="up">This text will scroll from bottom to up</marquee>
</body>
</html>
```