**Course Code: SIT1302**
**Course Name: Internet Programming**
**Chapter Name: Markup Language**

# UNIT 1
# MARKUP LANGUAGE

**UNIT 1:** Introduction to HTML - Structure of HTML, HTML elements - Mark up tags for inserting URL, Images, Tables, Frames - Form and its controls - Image maps - Client and Server Side – CSS – Inline – Internal and External - Multimedia components - Audio and Video - Dynamic HTML.

## 1.1 Introduction to HTML

### 1.1.1 Brief History of HTML

In the late 1980's , A physicist, Tim Berners-Lee who was a contractor at CERN, proposed a system for CERN researchers. In 1989, he wrote a memo proposing an internet based hypertext system.

**Tim Berners-Lee** is known as father of HTML. The first available description of HTML was a document called "HTML Tags" proposed by Tim in late 1991.

### 1.1.2 Features of HTML

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

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

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

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

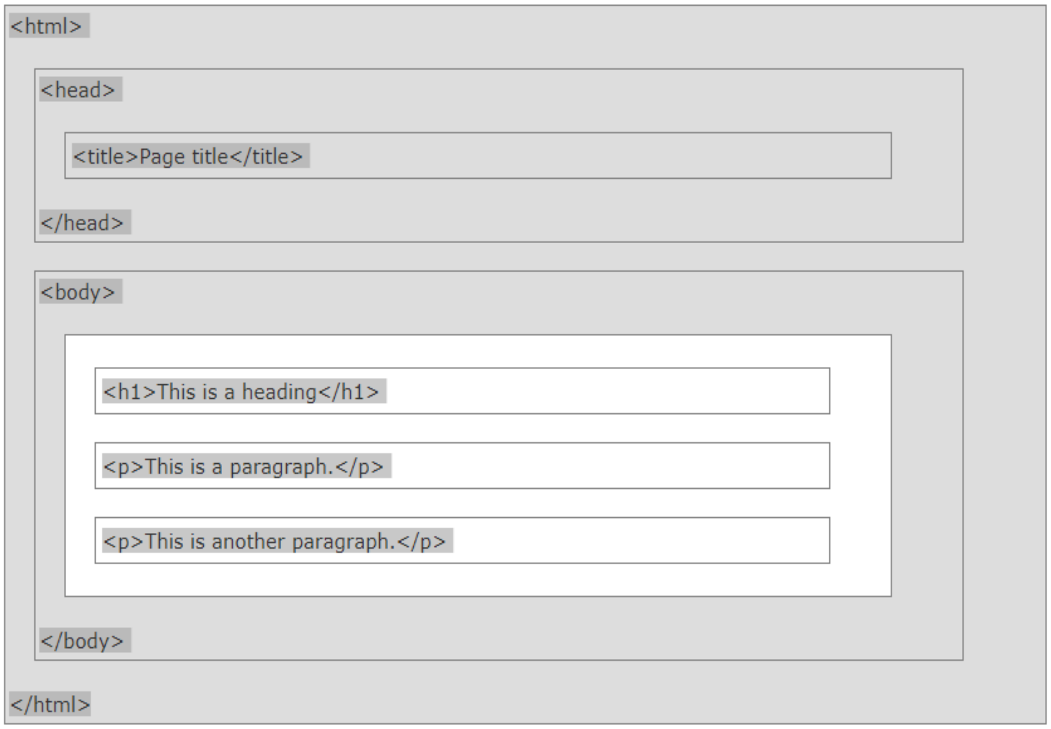
5) It is **platform-independent** because it can be displayed on any platform like Windows, Linux and Macintosh etc.

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

- HTML stands for Hyper Text Markup Language.
- HTML is used to create web pages.
- HTML is widely used language on the web.
- We can create static website by HTML only.
- HTML is an acronym which stands for Hyper Text Markup Language. Let's see what is Hyper Text and what is Markup Language?
- **Hyper Text:** Hyper Text simply means "Text within Text". A text has a link within it, is a hypertext. Every time when you click on a word which brings you to a new webpage, you have clicked on a hypertext.
- **Markup language:** A markup language is a programming language that is used make text more interactive and dynamic. It can turn a text into images, tables, links etc.

An HTML document is made of many HTML tags and each HTML tag contains different content.

## 1.2 Structure of HTML


<html>
<head>
<title>Page title</title>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
</body>
</html>

### The <!DOCTYPE> Declaration

- The <!DOCTYPE> declaration represents the document type, and helps browsers to display web pages correctly.
- It must only appear once, at the top of the page (before any HTML tags).
- The <!DOCTYPE> declaration is not case sensitive.

Let's see a simple example of HTML.

```
<!DOCTYPE>
<html>
<body>
<h1>Write Your First Heading</h1>
<p>Write Your First Paragraph.</p>
</body>
</html>
```

**Description of HTML example:**

**DOCTYPE:** It defines the document type.

**html** : Text between html tag describes the web document.

**body** : Text between body tag describes the body content of the page that is visible to the end user.

**h1** : Text between h1 tag describes the heading of the webpage.

**p** : Text between p tag describes the paragraph of the webpage.

## 1.3 HTML Elements:

HTML tags contain three main parts: opening tag, content and closing tag. But some HTML tags are unclosed tags.

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

**Syntax of HTML Elements:**

<tagname> content </tagname>

Note: HTML Tags are always written in lowercase letters. The basic HTML tags are given below:

### 1.3.1 HTML Formatting Tags:

Formatting is *a process of formatting text for better look and feel*. There are many formatting tags in HTML. These tags are used to make text bold, italicized, or underlined. There are almost 15 options available that how text appears in HTML.

## 1) Bold Text

If you write anything within <b>............</b> element, is shown in bold letters.

See this example:

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

Output:

```
Write a First Paragraph in bold text.
```

## 2) Italic Text

If you write anything within <i>............</i> element, is shown in italic letters.

example:

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

Output:

```
Write a First Paragraph in italic text.
```

## 3) Marked formatting

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

example:

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

Output:

```
I want to put a Mark on your face
```

## 4) Underlined Text

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

See this example:

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

Output:

```
Write Your First Paragraph in underlined text.
```

**5) Strike Text**

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

example:

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

Output:

~~Write Your First Paragraph with strikethrough.~~

**6) Monospaced Font**

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

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

See this example:

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

Output:

```
Hello Write Your First Paragraph in monospaced font.
```

**7) Superscript Text**

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

example:

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

Output:

```
Hello Write Your First Paragraph in superscript.
```

**8) Subscript Text**

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

example:

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

Output:

```
Hello Write Your First Paragraph in subscript.
```

**9) Deleted Text**

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

example:

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

Output:

```
Hello
```

**10) Inserted Text**

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

example:

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

Output:

```
Write another paragraph.
```

**11) Larger Text**

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

example:

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

Output:

```
Hello Write the paragraph in larger font.
```

**12) Smaller Text**

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

example:

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

Output:

```
Hello Write the paragraph in smaller font.
```

**13) Paragraph**

If you want to display the text content in paragraph format then use <p>…….</p> tag.

Example:
<p>This tag is used to display the contents in paragraph format. i.e., the sentences are printed continuously without any line breaks.</p>

Output:
This tag is used to display the contents in paragraph format. i.e., the sentences are printed continuously without any line breaks.

**Space inside HTML Paragraph**

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

## 14) Ordered & Unordered 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 UnorderedLists:

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>
```

This will produce following result:

- Beetroot
- Ginger
- Potato
- Radish

### 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>
```

This will produce following result:

- Beetroot
- Ginger
- Potato
- Radish

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>
```

This will produce following result:

- ○ Beetroot
- ○ Ginger
- ○ Potato
- ○ Radish

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>
```

This will produce following result:

- ○ Beetroot
- ○ Ginger
- ○ Potato
- ○ Radish

**HTML OrderedLists:**

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>
```

This will produce following result:

1. Beetroot
2. Ginger
3. Potato
4. Radish

**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.
```

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>
```

This will produce following result:

1. Beetroot
2. Ginger
3. Potato
4. Radish

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>
```

This will produce following result:

I. Beetroot
II. Ginger
III. Potato
IV. Radish

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>
```

This will produce following result:

i. Beetroot
ii. Ginger
iii. Potato
iv. Radish

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 following result:

A. Beetroot
B. Ginger
C. Potato
D. Radish

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 following result:

a. Beetroot
b. Ginger
c. Potato
d. Radish

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.
```

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>
```

This will produce following result:

```
  iv.    Beetroot
   v.    Ginger
  vi.    Potato
 vii.    Radish
```

**15) HTML DefinitionLists:**

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>
```

This will produce following result:

**HTML**

This stands for Hyper Text Markup Language

**HTTP**

This stands for Hyper Text Transfer Protocol

## 16) Marquee HTML:

The Marquee HTML tag is a non-standard HTML element which is used to scroll a image or text horizontally or vertically.

In simple words, you can say that it scrolls the image or text up, down, left or right automatically.

Marquee tag was first introduced in early versions of Microsoft's Internet Explorer. It is compared with Netscape's blink element.

**Example:**

<marquee>This is an example of html marquee </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. |
| scrollamount | defines scroll amount in number. |
| loop | defines loop for marquee content in number. |
| bgcolor | defines background color. It is now deprecated. |

**HTML Scroll Marquee:**

It is a by default property. It is used to scroll the text from right to left, and restarts at the right side of the marquee when it is reached to the end of left side. After the completion of loop text disappears.

```
<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>
```

**Disadvantages HTML marquee:**

1) Marquee may be distracting because human eyes are attracted towards movement and marquee text constantly.

2) Since Marquee text moves, so it is more difficult to click static text, depending on the scrolling speed.

3) It is a non-standard HTML element.

4) It draws user's attention needlessly and makes the text harder to read.

## <div> Element

- **<div>** is used for defining a section of your document. With the div tag, you can group large sections of HTML elements together and format them with CSS.
- The difference between the div tag and the span tag is that the div tag is used with block-level elements whilst the span tag is used with inline elements

**Example:**

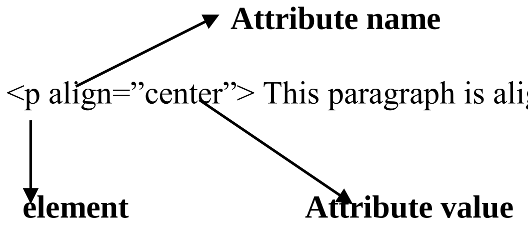
```
<html>
<head> <title>HTML div Tag</title>
</head>
<body>
<div id="contentinfo">
<p>Welcome to our  page. We provide  HTML  notes.</p>
</div>
</body>
</html>
```

**HTML Attributes**

Attributes provide additional information about HTML elements.

- HTML elements can have attributes
- Attributes provide additional information about an element
- Attributes are always specified in the start tag
- Attributes come in name/value pairs like: name="value"

Example:



```
<p align=”center”> This paragraph is aligned using center alignment.</p>
```

**HTML Styling:**

Every HTML element has a default style (background color is white, text color is black, text-size is 12px ...)

Changing the default style of an HTML element, can be done with the style attribute. This example changes the default background color from white to lightgrey.

Example:

```
<body style="background-color:lightgrey">
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
```

**HTML Style Attribute:**

The HTML style attribute has the following syntax:

style="property:value"

The property is a CSS property. The value is a CSS value.

**HTML Text Color:**

The color property defines the text color to be used for an HTML element:

Example:

```
<!DOCTYPE  html>
<html>
<body>
<h1  style="color:blue">This  is  a  heading</h1>
<p  style="color:red">This  is  a  paragraph.</p>
</body>
</html>
```

**HTML Text Fonts:**

The font-family property defines the font to be used for an HTML element:

Example:

```
<!DOCTYPE  html>
<html>
<body>
<h1  style="font-family:verdana">This  is  a  heading</h1>
<p  style="font-family:courier">This  is  a  paragraph.</p>
</body>
</html>
```

**HTML Text Size:**

The font-size property defines the text size to be used for an HTML element:

Example:

```
<!DOCTYPE  html>
<html>
<body>
<h1  style="font-size:300%">This  is  a  heading</h1>
<p  style="font-size:160%">This  is  a  paragraph.</p>
</body>
</html>
```

**HTML Text Alignment:**

The text-align property defines the horizontal text alignment for an HTML element:

Example:

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

```
<h1  style="text-align:center">Centered  Heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

## 1.4 Markup Tags for inserting Images, URL, Tables and Frames:

### a. Images:

HTML images are defined with the <img> tag.

Example:

```
<img  src="Sathyabama.jpg"  alt="www.sathyabamauniversity.ac.in"  width="104"  height
="142">
```

- The source file (src), alternative text (alt), and size (width and height) are attributes.
- The alt attribute specifies an alternative text to be used, when an HTML element cannot be displayed.

### b. URL:

The **HTML anchor tag** defines *a hyperlink that links one page to another page* i.e., an element in an electronic document that links to another place in the same document or to an entirely different document.

The "href" attribute is the most important attribute of the HTML a tag.

**href attribute of HTML anchor tag:**

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

The syntax of HTML anchor tag is given below.

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

Let's see an example of HTML anchor tag.

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

<a href ="https://www.google.co.in">Visit Google India</a>
```

- The href attribute specifies the destination address (https://www.google.co.in) of the link.
- The link text is the visible part (Visit Google India).

- ➢ Clicking on the link text will send you to the specified address.

**The target Attribute**

- ➢ The target attribute specifies where to open the linked document.
- ➢ The target attribute can have one of the following values:

  - _blank - Opens the linked document in a new window or tab
  - _self - Opens the linked document in the same window/tab as it was clicked (this is default)
  - _parent - Opens the linked document in the parent frame
  - _top - Opens the linked document in the full body of the window
  - framename - Opens the linked document in a named frame

**Example**

```
<a href="https://www.google.co.in/" target="_blank">Visit Google India</a>
```

This example will open the linked document in a new browser window/tab.

**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**.

**c. Table:**

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

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

| Tag | Description |
|---|---|
| &lt;table&gt; | It defines a table. |
| &lt;tr&gt; | It defines a row in a table. |
| &lt;th&gt; | It defines a header cell in a table. |

| | |
|---|---|
| &lt;td&gt; | It defines a cell in a table. |
| &lt;caption&gt; | It defines the table caption. |
| &lt;colgroup&gt; | It specifies a group of one or more columns in a table for formatting. |
| &lt;col&gt; | It is used with &lt;colgroup&gt; element to specify column properties for each column. |
| &lt;tbody&gt; | It is used to group the body content in a table. |
| &lt;thead&gt; | It is used to group the header content in a table. |
| &lt;tfooter&gt; | It is used to group the footer content in a table. |

Example:

```
<html>
<body>
<table border="1">
<tr>
<th>Table Header</th>
<th>Table Header</th>
</tr>
<tr>
<td>Table cell 1</td>
<td>Table cell 2</td>
</tr>
<tr>
<td>Table cell 3</td>
<td>Table cell 4</td>
</tr>
</table>
</body>
</html>
```

Output:

| **Table Header** | **Table Header** |
|---|---|
| Table cell 1 | Table cell 2 |
| Table cell 3 | Table cell 4 |

**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>
```

This will produce following result:

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

**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.

```
<!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>

</html>
```

## <Frameset> element and target attribute:

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.

<frameset> tag defines a frameset.

- Each <frameset> holds one or more <frame> elements. Each <frame> element can hold a separate document.
- <frameset> element specifies HOW MANY columns or rows there will be in the frameset, and HOW MUCH percentage/pixels of space will occupy each of them.

**Example:**

A simple two-framed (row frames) page:

**sampleframe.html:**

```
<html>
<frameset rows="25%,*">
  <frame src="frame1.html">
  <frame src="frame2.html">
</frameset>
</html>
```

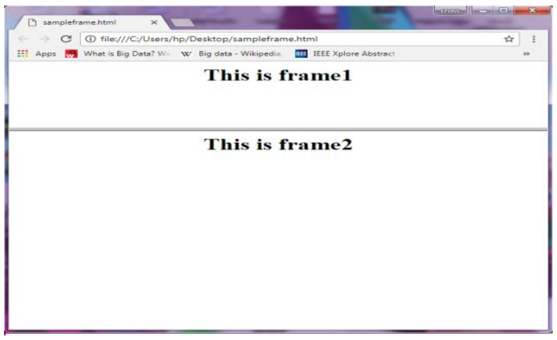
**frame1.html:**

```
<html>
<body>
<center><h1>This is frame1</h1></center>
</body>
</html>
```

**frame2.html:**

```
<html>
<body>
<center><h1>This is frame2</h1></center>
</body>
</html>
```

**Output:**



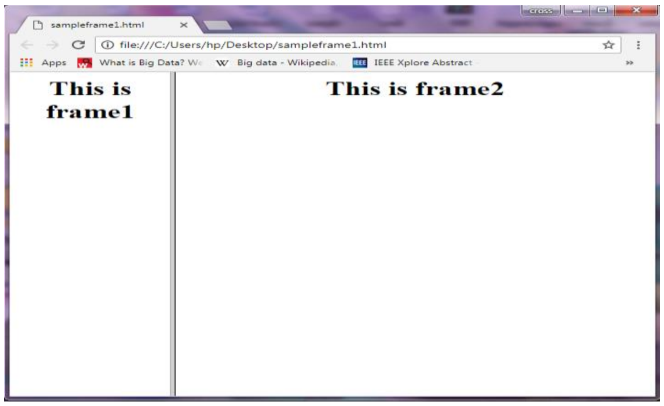
```
This is frame1

This is frame2
```

A simple two-framed (column frames) page:

**sampleframe1.html:**

```
<html>
<frameset cols="25%,*">
  <frame src="frame1.htm">
  <frame src="frame2.htm">
</frameset>
</html>
```

**Output:**



This is frame1

This is frame2

**Image Mapping:**

In Web page development, an image map is a graphic image defined so that a user can click on different areas of the image and be linked to different destinations. You make an image map by defining each of the sensitive areas in terms of their x and y coordinates (that is, a certain horizontal distance and a certain vertical distance from the left-hand corner of the image). With each set of coordinates, you specify a Uniform Resource Locator or Web address that will be linked to when the user clicks on that area.

Originally, the map file had to be sent to the server. Now the creator can place the map information either at the server or at the client (a "client-side map").

**Client side image Mapping:**

The X and Y coordinates are expressed in pixels in the same HTML file that contains the link to the image map.

**clientmap.html:**

```
<html>
<head>
```

```
<title>MAPPING</title>
</head>
<body>
<img src="color.jpg" usemap="#shapes">
<map name="shapes">
<area shape="circle" coords=101,114,100 href="circle.html">
<area shape="rect" coords=261,64,343,142 href="rect.html">
<area shape="poly" coords=223,154,171,204,171,256,221,284,271,256,273,204 href="poly.html">
</map>
</bodY>
</html>
```

**Circle.html*:***

```
<html>
<head>
<title> circle</title>
</head>
<body>
<h3> This is a circle
</h3>
</body>
</html>
```

**Rectangle.html:**

```
<html>
<head>
<title> rectangle</title>
</head>
<body>
<h3> This is a rectangle</h3>
</body>
</html>
```

**Polygon.html:**

```
<html>
<head>
<title> polygon</title>
</head>
<body>
<h3> This is a polygon</h3>
<a href="image.bmp" target="third">image</a>
</body>
</html>
```

**Server side image Mapping:**

The X and Y coordinates are expressed in pixels in a different file which has **.map** extension that contains the link to the image map.

**servermap.asp:**

```
<html>
<head><title>Server-side Image map Example</title></head>
<body>
        <h1 align="center">Server-side Image map Test</h1><hr/>
        <a href="mapper.map"> <img src="paradise.jpg" ismap alt="Shapes Map"
                border="0" width="400" height="200"></a>
</body>
</html>
```

**Mapper.map**

```
default a.jpg
poly b.jpg 16,358,216,378,206,556,66,574,14,358
circle c.jpg 200,200,100
rect Sunset.jpg 250,50,350,150
```

**Cascading Style Sheets:**

There are three ways of inserting a style sheet:

- External style sheet
- Internal style sheet
- Inline style

**External Style Sheet:**

An external style sheet is ideal when the style is applied to many pages. With an external style sheet, you can change the look of an entire Web site by changing just one file.

Each page must include a link to the style sheet with the <link> tag. The <link> tag goes inside the head section:

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

An external style sheet can be written in any text editor. The file should not contain any html tags. The style sheet file must be saved with a **.css** extension. An example of a style sheet file called **"myStyle.css"**, is shown below:

```
body {
    background-color: lightblue;
}
h1 {
    color: navy;
    margin-left: 20px;
}
```

**Internal Style Sheet:**

An internal style sheet should be used when a single document has a unique style. You define internal styles in the head section of an HTML page, inside the
<style> tag, like this:

**Example:**

```
<head>
<style>
body {
      background-color:  linen;
}
h1 {
      color:  maroon;
      margin-left:  40px;
}
</style>
</head>
```

**Inline Style Sheet:**

An inline style loses many of the advantages of a style sheet (by mixing content with presentation).

To use inline styles, add the style attribute to the relevant tag. The style attribute can contain any CSS property. The example shows how to change the color and the left margin of a h1 element:

**Example:**

```
<h1  style="color:blue;margin-left:30px;">This  is  a  heading.</h1>
```

**HTML Forms:**

A form is simply an area that can contain form fields.

Form fields are objects that allow the visitor to enter information - for example text boxes, drop-down menus or radio buttons.

**<Form>      </Form>**

To let the browser know where to send the content we add these attributes to the <form> tag:

- action=address

  The **address** is the url of script the content should be sent to.

- method=post or method=get

  The **post** and **get** methods are simply two different methods for submitting data to the script

**POST** offers better security because the submitted data is not visible in the page address.

If the form submission is passive (like a search engine query), and without sensitive information.

When you use **GET**, the form data will be visible in the page address:

| | |
|---|---|
| enctype | Specifies the encoding of the submitted data (default: is url-encoded) |

enctype="application/x-www-form-urlencoded"

.

These fields can be added to your forms:

- **Text field**

  Text fields are one line areas that allow the user to input text.

  <input type="text" size="25" value="Enter your name here!">

The **size** option defines the width of the field. That is how many visible characters it can contain.

The **maxlength** option defines the maximum length of the field. That is how many characters can be entered in the field.
If you do not specify a maxlength, the visitor can easily enter more characters than are visible in the field at one time.

The **name** setting adds an internal name to the field so the program that handles the form can identify the fields.

The **value** setting defines what will appear in the box as the default value.

- **Password field**

Password fields are similar to text fields. The difference is that what is entered into a password field shows up as dots on the screen. This is, of course, to prevent others from reading the password on the screen.

```
<input type="password" size="25">
```

- **Hidden field**

Hidden fields are similar to text fields, with one very important difference!

The difference is that the hidden field does not show on the page

```
<input type="hidden" name="Language" value="English">
```

- **Text area**

Text areas are text fields that can span several lines.

```
<textarea cols="40" rows="5" name="myname">
```

- **Check box**

Check boxes are used when you want to let the visitor select one or more options from a set of alternatives. If only one option is to be selected at a time you should use radio buttons instead.

```
<input type="checkbox" name="option1" value="Milk"> Milk<br>
<input type="checkbox" name="option2" value="Butter" checked>
input type="checkbox" name="option3" value="Cheese">
Cheese<br>
```

- **Radio button**

Radio buttons are used when you want to let the visitor select one - and just one - option from a set of alternatives. If more options are to be allowed at the same time you should use
check boxes instead.

```
<input type="radio" name="group1" value="Milk"> Milk<br>
<input type="radio" name="group1" value="Butter" checked>
Butter<br>
<input type="radio Butter<br>
<" name="group1" value="Cheese"> Cheese
```

- **Drop-down menu**

Drop-down menus are probably the most flexible objects you can add to your forms.

Depending on your settings, drop-down menus can serve the same purpose as radio buttons (one selection only) or check boxes (multiple selections allowed).

The advantage of a drop-down menu, compared to radio buttons or check boxes, is that it takes up less space.
But that is also a disadvantage, because people can't see all options in the menu right away.

| HTML | EXPLANATION | EXAMPLE |
|---|---|---|
| select | Drop-down menu | |
| name= | Name of the field. | |
| size= | Visible items in list. | |
| multiple= | Allows multiple choices if yes. | |
| option | Individual items in the menu. | |
| selected | Default select the item. | |
| value= | Value to send if selected. | |

Drop-down menus combine <select> and <option>.
Both tags have an opening and a closing tag.

A typical example of the syntax would be:

```
<select>
  <option>Milk</option>
  <option>Coffee</option>
  <option>Tea</option>
</select>
```

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

- **Submit button**

When a visitor clicks a submit button, the form is sent to the address specified in the action of the <form> tag.
<input type="submit" value="Send me your name!">

- **Reset button**

When a visitor clicks a reset button, the entries are reset to the default values.
<input type="reset" value="Reset!">

- **Image button**

Image buttons have the same effect as submit buttons. When a visitor clicks an image button the form is sent to the address specified in the action setting of the <form> tag

```
<input type="image" src="rainbow.gif" name="image" width="60" height="60">
```

The HTML button Element represents a clickable button.

```
<button name="button"  value="OK" type="button">Click Me</button>
```

**Grouping Form Data with <fieldset>:**

The <fieldset> element groups related data in a form.

The <legend> element defines a caption for the <fieldset> element.
Example:

```
<html>

<body>

<form action="action_page.php">
<fieldset>
<legend>Personal  information:</legend> First
name:<br>
<input  type="text"  name="firstname"  value="Mickey"><br>
Last name:<br>
<input  type="text"  name="lastname"  value="Mouse"><br><br>
<input  type="submit"  value="Submit"></fieldset>
</form>
</body>
</html>
```

**HTML Audio Tag:**

HTML audio tag is used to define sounds such as music and other audio clips. Currently there are three supported file format for HTML 5 audio tag.

1. mp3
2. wav
3. ogg

HTML5 supports <video> and <audio> controls. The Flash, Silverlight and similar technologies are used to play the multimedia items.

This table defines that which web browser supports which audio file format.

Example:

Let's see the code to play mp3 file using HTML audio tag.

```
<audio controls>
<source src="koyal.mp3" type="audio/mpeg">
Your browser does not support the html audio tag.
</audio>
```

Example:

Let's see the example to play ogg file using HTML audio tag.

```
<audio controls>
<source src="koyal.ogg" type="audio/ogg">
Your browser does not support the html audio tag.
</audio>
```

**Attributes of HTML Audio Tag:**

There is given a list of HTML audio tag.

| Attribute | Description |
|---|---|
| controls | It defines the audio controls which is displayed with play/pause buttons. |
| autoplay | It specifies that the audio will start playing as soon as it is ready. |
| loop | It specifies that the audio file will start over again, every time when it is completed. |
| muted | It is used to mute the audio output. |
| preload | It specifies the author view to upload audio file when the page loads. |
| src | It specifies the source URL of the audio file. |

HTML Audio Tag Attribute Example

Here we are going to use controls, autoplay, loop and src attributes of HTML audio tag.

```
<audio controls autoplay loop>
<source src="koyal.mp3" type="audio/mpeg"></audio> Test it Now
```

**MIME Types for HTML Audio format:**

The available MIME type HTML audio tag is given below.

| Audio Format | MIME Type |
|---|---|
| mp3 | audio/mpeg |
| Ogg | audio/ogg |
| Wav | audio/wav |

**HTML Video Tag:**

HTML 5 supports <video> tag also. The HTML video tag is used for streaming video files such as a movie clip, song clip on the web page.

Currently, there are three video formats supported for HTML video tag:

1. mp4
2. webM
3. ogg

Example:

Let's see the code to play mp4 file using HTML video tag.

```
<video controls>
<source src="movie.mp4" type="video/mp4">
Your browser does not support the html video tag.
</video>
```

Let's see the example to play ogg file using HTML video tag.

```
<video controls>
<source src="movie.ogg" type="video/ogg">
Your browser does not support the html video tag.
</video>
```

**Attributes of HTML Video Tag:**

Let's see the list of HTML 5 video tag attributes.

| Attribute | Description |
|---|---|
| controls | It defines the video controls which is displayed with play/pause buttons. |
| height | It is used to set the height of the video player. |
| width | It is used to set the width of the video player. |
| poster | It specifies the image which is displayed on the screen when the video is not played. |
| autoplay | It specifies that the video will start playing as soon as it is ready. |
| loop | It specifies that the video file will start over again, every time when it is completed. |
| muted | It is used to mute the video output. |
| preload | It specifies the author view to upload video file when the page loads. |
| src | It specifies the source URL of the video file. |

Example:

Let's see the example of video tag in HTML where are using height, width, autoplay, controls and loop attributes.

```
<video width="320" height="240" controls autoplay loop>
<source src="movie.mp4" type="video/mp4">
Your browser does not support the html video tag.
</video>
```

**MIME Types for HTML Video format:**

The available MIME type HTML video tag is given below.

| Video Format | MIME Type |
|---|---|
| mp4 | video/mp4 |
| Ogg | video/ogg |
| webM | video/webM |

**HTML Canvas Tag:**

The HTML canvas element provides HTML a bitmapped surface to work with. It is used to draw graphics on the web page.

The HTML 5 <canvas> tag is used to draw graphics using scripting language like JavaScript.

The <canvas> element is only a container for graphics, you must need a scripting language to draw the graphics. The <canvas> element allows for dynamic and scriptable rendering of 2D shapes and bitmap images.

It is a low level, procedural model that updates a bitmap and does not have a built-in scene. There are several methods in canvas to draw paths, boxes, circles, text and add images.