

## MODULE-1

### Introduction:

Web technologies related to the interface between web servers and their clients. This information includes markup languages, programming interfaces and languages, and standards for document identification and display. In general web technology incorporates tools and techniques for web development.

### Web browsers

Web browsers are software applications used to access and display websites on the internet. They retrieve web pages from servers using the Hypertext Transfer Protocol (HTTP) or its secure version, HTTPS, and render them for users. The information could be web sites, video or audio information.

Ex: Chrome, Mozilla firefox, internet explorer, safari, netscape, opera etc.

#### How a Web Browser Works

- User enters a URL (e.g., www.example.com).
- Browser sends a request to a web server.
- Server responds with the requested web page.
- Browser renders the page using HTML, CSS, and JavaScript.

### Web Servers

A **web server** is software or hardware that stores, processes, and delivers web pages to users over the internet. It handles requests from web browsers and sends the requested content (such as HTML files, images, or videos). This exchange takes place using Hypertext Transfer Protocol (HTTP).

The most commonly used Web servers are Apache, which has been implemented for a variety of computer platforms, and Microsoft's Internet Information Server (IIS), which runs under Windows operating systems.

The primary task of a Web server is to monitor a communications port on its host machine, accept HTTP commands through that port, and perform the operations specified by the commands.

#### How a Web Server Works

- User enters a URL (e.g., www.example.com) in a web browser.
- Browser sends an HTTP request to the web server.
- Web server processes the request and locates the requested file or page.
- Server sends a response (usually an HTML page) back to the browser.
- Browser renders the page and displays it to the user.

### Types of Web Servers

1. **Static Web Server** – Delivers fixed content (HTML, images, CSS).
2. **Dynamic Web Server** – Generates content dynamically using server-side languages (PHP, Python, Java, etc.).

**Apache:**

It is an excellent server because it's both fast and reliable. Apache is capable of providing a long list of services beyond the basic process of serving documents to clients. When Apache begins execution, it reads its configuration information from a file and sets its parameters to operate accordingly.

Apache is a freely available Web server that is distributed under an "open source" license. Version 2.0 runs on most UNIX-based operating systems (such as Linux, Solaris, Digital UNIX, and AIX), on other UNIX/POSIX-derived systems (such as Rhapsody, BeOS, and BS2000/OSD), on Amiga OS, and on Windows 2000.

**IIS (Internet Information server):**

IIS (**Internet Information Services**) is a web server developed by **Microsoft** for hosting websites, applications, and services on **Windows Server** operating systems. It supports **HTTP, HTTPS, FTP, FTPS, SMTP, and NNTP** protocols.

The Microsoft IIS server is supplied as part of windows- and because it is reasonably good server. With IIS, server behaviour is modified by changes made through a window-based management program, named the IIS snap-in, which controls both IIS and ftp. Under Windows XP, the IIS snap-in is accessed by going to control panel, administrative tools, and IIS Admin.

**How IIS Works**

- User sends a request via a web browser (e.g., www.example.com).
- IIS receives the request and processes it.
- IIS checks for authentication and authorization (if required).
- IIS retrieves the requested content (static files or dynamic pages).
- IIS sends the response (HTML, CSS, JavaScript, or data from databases).

**IIS Components**

- IIS Manager – Graphical interface for managing the server.
- Application Pools – Separates web applications for better stability.
- Virtual Directories – Allows access to files outside the main web folder.
- Logging & Diagnostics – Provides detailed logs for troubleshooting.

**MIME (Multi-Purpose Internet Mail Extensions)**

MIME (**Multi-Purpose Internet Mail Extensions**) is an **internet standard** that extends the capabilities of email and web communication by supporting different file types such as images, videos, audio, and documents. It allows emails and web browsers to handle non-text content.

**Purpose of MIME**

- Allows emails to include attachments (e.g., images, PDFs, audio files).
- Helps web browsers understand different file formats.
- Supports different character encodings for non-English text.

### How MIME Works

- A sender attaches a file (e.g., image or document) to an email.
- MIME encodes the file into text format for transmission.
- Recipient's email client decodes the MIME message and displays the file.
- Web servers also use MIME types to tell browsers how to handle files.

A browser needs some way of determining the format of a document it receives from a Web server. Without knowing the form of the document, the browser would be unable to render it, because different document formats require different rendering tools. The forms of these documents are specified with Multipurpose Internet Mail Extensions (MIME).

### Type Specifications

MIME was developed to specify the format of different kinds of documents to be sent via Internet mail. These documents could contain various kinds of text, video data, or sound data.

A Web server attaches a MIME format specification to the beginning of the document that it is about to provide to a browser. When the browser receives the document from a Web server, it uses the included MIME format specification to determine what to do with the document.

MIME specifications have the following form: type/subtype

The most common MIME types are text, image, and video. The most common text subtypes are plain and html. Some common image subtypes are gif and jpeg. Some common video subtypes are mpeg and quicktime.

### URL (Uniform Resource Locators)

A **URL (Uniform Resource Locator)** is the web address used to access resources on the internet, such as websites, images, or files. It specifies the location of a resource and how to retrieve it. There are many different kinds of resources, identified by different forms of URLs.

**URL Formats:** scheme:object-address

**Ex:** <https://www.example.com:80/path/page.html?query=abc#section1>

### Types of URLs

1. Absolute URL – A complete web address including protocol and domain.

Example: <https://www.example.com/page.html>

2. Relative URL – A partial address that depends on the base URL.

Example: /page.html (works within the same website).

### HTTP (Hypertext Transport Protocol)

HTTP means HyperText Transfer Protocol. HTTP is the underlying protocol used by the World Wide Web and this protocol defines how messages are formatted and transmitted, and what actions Web servers and browsers should take in response to various commands.

HTTP is an application layer protocol. It supports communication between web browser and web server.

HTTP consists of two phases:

- i. The Request phase (Client sends a request).
- ii. The Response phase (Server returns a reply).

### Introduction to XHTML5

XHTML5 (**Extensible Hypertext Markup Language 5**) is a reformulation of **HTML5** using **XML (Extensible Markup Language) rules**. It combines the flexibility of HTML5 with the strict structure of XML, making it more reliable for web development.

XHTML (Extensible Hypertext Markup Language) is a family of XML markup languages that mirror or extend versions of the widely used Hypertext Markup Language (HTML), the language in which web pages are written. While HTML (prior to HTML5) was defined as an application of Standard Generalized Markup Language (SGML), a very flexible markup language framework, XHTML is an application of XML, a more restrictive subset of SGML. Because XHTML documents need to be well-formed, they can be parsed using standard XML parsers—unlike HTML, which requires a lenient HTML-specific parser.

XHTML 1.0 became a World Wide Web Consortium (W3C) Recommendation on January 26, 2000. XHTML 1.1 became a W3C Recommendation on May 31, 2001. XHTML5 is undergoing development as of September 2009, as part of the HTML5 specification.

### Why Use XHTML5?

- Better structure and consistency due to XML rules.
- Improved error handling, making pages more reliable.
- Compatibility with modern browsers that support HTML5.
- Easier integration with other XML-based technologies (e.g., SVG, MathML).

### HTML Versus XHTML

Feature	HTML	XHTML
Syntax	Flexible, allows some errors	Strict, follows XML rules
Case Sensitivity	Tags are case-insensitive	Tags must be lowercase
Closing Tags	Optional for some elements	Required for all elements
Attribute Values	Can be unquoted in some cases	Must always be quoted
Nesting	More forgiving	Must be properly nested

### Standard XHTML Document Structure

An XHTML document consists of three main parts:

1. DOCTYPE
2. Head
3. Body

The basic document structure is:

```
<!DOCTYPE ...>

<html>

<head> ... </head>

<body> ... </body>

</html>
```

An XHTML document must have an XHTML DOCTYPE declaration.

A complete list of all the XHTML Doctypes is found in our HTML Tags Reference. The <html>, <head>, <title>, and <body> elements must also be present, and the xmlns attribute in <html> must specify the xml namespace for the document.

This example shows an XHTML document with a minimum of required tags:

An XHTML example

```
1. <?xml version="1.0"?>

2. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "DTD/xhtml1-transitional.dtd">

3. <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

4. <head>

   <title>My XHTML Sample Page</title>

   </head>

5. <body bgcolor="white">

   <center><h1>Welcome to XHTML !</h1>

   </center>

   </body>

6. </html>
```

**Line 1:** Since XHTML is HTML expressed in an XML document, it must include the initial XML declaration <?xml version="1.0"?> at the top of the document.

**Line 2:** XHTML documents must be identified by one of three standard sets of rules.

These rules are stored in a separate document called a Document Type Declaration (DTD), and are utilized to validate the accuracy of the XHTML document structure. The purpose of a DTD is to describe, in precise terms, the language and syntax allowed in XHTML.

**Line 3:** The second tag in an XHTML document must include the opening <html> tag with the XML namespace identified by the xmlns=http://www.w3.org/1999/xhtml attribute. The XML namespace

identifies the range of tags used by the XHTML document. It is used to ensure that names used by one DTD don't conflict with user-defined tags or tags defined in other DTDs.

**Line 4:** XHTML documents must include a full header area. This area contains the opening `<head>` tag and the title tags (`<title></title>`), and is then completed with the closing `</head>` tag.

**Line 5:** XHTML documents must include opening and closing `<body></body>` tags. Within these tags you can place your traditional HTML coding tags. To be XHTML conformant, the coding of these tags must be well-formed.

**Line 6:** Finally, the XHTML document is completed with the closing `</html>` tag.

### HTML Elements:

An HTML element is a building block of a webpage. It consists of a start tag, content, and an end tag. HTML documents are defined by HTML elements.

Syntax: `<tagname>Content</tagname>`

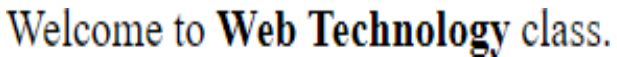
For example, `<p>My first paragraph</p>`.

<code>&lt;b&gt;</code>	Bold text
<code>&lt;strong&gt;</code>	Important text
<code>&lt;i&gt;</code>	Italic text
<code>&lt;em&gt;</code>	Emphasized text
<code>&lt;mark&gt;</code>	Marked text
<code>&lt;small&gt;</code>	Smaller text
<code>&lt;del&gt;</code>	Deleted text
<code>&lt;ins&gt;</code>	Inserted text
<code>&lt;sub&gt;</code>	Subscript text
<code>&lt;sup&gt;</code>	Superscript text

### Bold `<b>` tag:

HTML `<b>` and `<strong>` Elements

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

<pre> &lt;!doctype html&gt;   &lt;html dir="ltr" lang="en"&gt;     &lt;head&gt;       &lt;meta charset="utf-8"&gt;     &lt;body&gt;       Welcome to &lt;b&gt; Web Technology&lt;/b&gt; class.     &lt;/body&gt;   &lt;/html&gt; </pre>	
---	--

**Headings: -**

Headings are defined with the <h1> to <h6> tags. <h1> defines the largest heading while <h6> defines the smallest.

<pre> &lt;!doctype html&gt;   &lt;html dir="ltr" lang="en"&gt;     &lt;head&gt;       &lt;meta charset="utf-8"&gt;     &lt;/head&gt;     &lt;title&gt; Sample&lt;/title&gt;   &lt;/head&gt;   &lt;body&gt;     &lt;h1&gt;Heading 1&lt;/h1&gt;     &lt;h2&gt;Heading 2&lt;/h2&gt;     &lt;h3&gt;Heading 3&lt;/h3&gt;     &lt;h4&gt;Heading 4&lt;/h4&gt;     &lt;h5&gt;Heading 5&lt;/h5&gt;     &lt;h6&gt;Heading 6&lt;/h6&gt;   &lt;/body&gt; &lt;/html&gt; </pre>	<h1>Heading 1</h1> <h2>Heading 2</h2> <h3>Heading 3</h3> <h4>Heading 4</h4> <h5>Heading 5</h5> <h6>Heading 6</h6>
---	---

**Line Breaks <br>**

The HTML <br> element defines a line break.

<pre> &lt;!doctype html&gt;   &lt;html dir="ltr" lang="en"&gt;     &lt;head&gt;       &lt;meta charset="utf-8"&gt;     &lt;/head&gt;     &lt;body&gt;       This is&lt;br&gt;a paragraph&lt;br&gt; with line breaks.     &lt;/body&gt;   &lt;/html&gt; </pre>	<p>This is a paragraph with line breaks.</p>
---	--

**Paragraphs <p>: -**

Paragraphs are defined with the <p> tag. Think of a paragraph as a block of text. You can use the align attribute with a paragraph tag as well.

<p> is have a attribute called align it is used to align the text at left, right and center.

```

<!doctype html>
  <html dir="ltr" lang="en">
    <head>
      <meta charset="utf-8">
    <title> Sample</title>
  </head>
  <body>
    <p >This is a paragraph</p>
  </body>
</html>

```

This is a paragraph

### Italic <i> tag:

The HTML <i> element defines a part of text in italic.

```

<!doctype html>
  <html dir="ltr" lang="en">
    <head>
      <meta charset="utf-8">
    <body>
      Welcome to <i> Web Technology</i> class.
    </body>
  </html>

```

Welcome to *Web Technology* class.

### <mark> tag

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

```

<!doctype html>
  <html dir="ltr" lang="en">
    <head>
      <meta charset="utf-8">
    <body>
      THIS IS WEB TECHNOLOGY <mark>UNIT 1</mark> 2ND
      SEM MCA.

```

THIS IS WEB TECHNOLOGY **UNIT 1** 2ND SEM MCA.



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

**<del> tag**

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

```
<!doctype html>
  <html dir="ltr" lang="en">
    <head>
      <meta charset="utf-8">
    <body>
      My favourite colour is <del>blue</del> red.
    </body>
  </html>
```

My favourite colour is ~~blue~~ red.

**Subscript <sub> tag**

The HTML <sub> element defines subscript text. Subscript text appears half a character below the normal line. Subscript text can be used for chemical formulas, like H<sub>2</sub>O:

```
<!doctype html>
  <html dir="ltr" lang="en">
    <head>
      <meta charset="utf-8">
    <body>
      H<sub>2</sub>O</p>
    </body>
  </html>
```

H<sub>2</sub>O.

**Superscript <sup> tag**

The HTML <sup> element defines superscript text. Superscript text appears half a character above the normal line. Superscript text can be used for mathematical formulas, like ax<sup>2</sup>+bx+c=0.

```
<!doctype html>
  <html dir="ltr" lang="en">
    <head>
      <meta charset="utf-8">
    <body>
      ax<sup>2</sup>+bx+c=0</p>
    </body>
  </html>
```

ax<sup>2</sup>+bx+c=0

**IMAGE <img src>**

HTML <img> tag is used to display image on the web page. HTML img tag is an empty tag that contains attributes only, closing tags are not used in HTML image element.

```
<!doctype html>

    <html dir="ltr" lang="en">

    <head>

    <meta charset="utf-8">

<body>

<h2>HTML Image Example</h2>



</body>

</html>
```

**Attributes of HTML img tag****1) src**

It is a necessary attribute that describes the source or path of the image. It instructs the browser where to look for the image on the server. The location of image may be on the same directory or another server.

**2) alt**

The alt attribute defines an alternate text for the image, if it can't be displayed. The value of the alt attribute describes the image in words.

**3) width**

It is an optional attribute which is used to specify the width to display the image.

**4) height**

It defines the height of the image.

**Lists <li>: -**

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

**1. Ordered List or Numbered List (ol)**

2. Unordered List or Bulleted List (ul)

3. Description List or Definition List (dl)

### 1. Ordered List or Numbered List <ol>

An **Ordered List** in HTML is used to display a list of items in a **numbered** or **sequenced** format. It is created using the <ol> tag, with each item inside an <li> (list item) tag.

Syntax:

```
<ol>
```

```
<li>First item</li>
```

```
<li>Second item</li>
```

```
<li>Third item</li>
```

```
</ol>
```

Ex:

<pre>&lt;html dir="ltr" lang="en"&gt; &lt;head&gt; &lt;meta charset="utf-8"&gt; &lt;title&gt;HTML Ordered List&lt;/title&gt; &lt;/head&gt; &lt;body&gt; &lt;ol&gt; &lt;li&gt;DBMS&lt;/li&gt; &lt;li&gt;SOFTWARE ENGINEERING&lt;/li&gt; &lt;li&gt;JAVA&lt;/li&gt; &lt;li&gt;WEB TECHNOLOGY&lt;/li&gt; &lt;/ol&gt; &lt;/body&gt; &lt;/html&gt;</pre>	<hr/> <ol style="list-style-type: none"> <li>1. DBMS</li> <li>2. SOFTWARE ENGINEERING</li> <li>3. JAVA</li> <li>4. WEB TECHNOLOGY</li> </ol>
--	--

### 2. Unordered List or Bulleted List <ul>

An **unordered list** in HTML is used to display a list of items **without a specific order**. The items are usually marked with **bullets (•)**. It is created using the <ul> tag, with each item inside an <li> (list item) tag.

Syntax:

```
<ul>
```

```
<li>Item One</li>
```

```
<li>Item Two</li>
```

```
</li>Item Three</li>
```

```
</ul>
```

Ex:

```
<!doctype html>
  <html dir="ltr" lang="en">
    <head>
      <meta charset="utf-8">
    <title>HTML Unordered List</title>
  </head> <body>
    <ul>
      <li>DBMS</li>
      <li>SOFTWARE ENGINEERING</li>
      <li>JAVA</li>
      <li>WEB TECHNOLOGY</li>
    </ul>
  </body>
</html>
```

- DBMS
- SOFTWARE ENGINEERING
- JAVA
- WEB TECHNOLOGY

### 3. Description Lists <dl>

HTML also supports description lists. A Definition List in HTML is used to display terms and their corresponding definitions. It is commonly used for glossaries, descriptions, or metadata.

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

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

Structure:

A definition list consists of three tags:

Tag	Description
<dl>	Defines the <b>definition list</b> container
<dt>	Defines the <b>term (name)</b>
<dd>	Defines the <b>description (definition)</b>

Syntax

```
<dl>
```

```
<dt>HTML</dt>
```

```
<dd>HyperText Markup Language, used to create web pages.</dd>
```

```
<dt>CSS</dt>
```

```
<dd>Cascading Style Sheets, used for styling web pages.</dd>
```

```
</dl>
```

```
<!doctype html>
```

```
<html dir="ltr" lang="en">
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<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

This stands for Hyper Text Markup Language

## HTTP

This stands for Hyper Text Transfer Protocol

### HTML tables:

An **HTML table** is used to display data in a structured format using **rows and columns**. The HTML tables allow web authors to arrange data like text, images, links, other tables, etc. into rows and columns of cells.

It is created using the <table> tag and contains table rows (<tr>), table headers (<th>), and table data cells (<td>).

A border attribute in the <table> tag specifies a border between the cells. Rule specifies the lines that separate the cells. The border attribute can be set to a number, which will be the border width in pixels.

Structure:

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

```
<tr>
```

```
<th>Header 1</th>
```

```
<th>Header 2</th>
```

```
</tr>
```

```
<tr>
```

```

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

<tr>

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

</tr>
</table>

```

Ex:

```

<!doctype html>
    <html dir="ltr" lang="en">
    <head>
        <meta charset="utf-8">
    <title>HTML Tables</title>
    </head>
    <body>
        <table border="1">
            <tr>
                <th>DAY</th>
                <td>8.30-9.30</td>
                <td>9.30-10.30</td>
            </tr>
            <tr>
                <th> Monday</th>
                <td>DBMS</td>
                <td>WT</td>
            </tr>
            <tr>
                <th> Tuesday</th>
                <td> JAVA</td>
                <td>SE</td>
            </tr>
        </table>
    </body>
</html>

```

DAY	8.30-9.30	9.30-10.30
Monday	DBMS	WT
Tuesday	JAVA	SE

### Cellpadding and Cellspacing Attributes

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

```
<!doctype html>
```

```

<html dir="ltr" lang="en">
<head>
<meta charset="utf-8">
<title>HTML Table Cellpadding</title>
</head>
<body>
<table border = "1" cellpadding = "5" cellspacing = "5">
  <tr>
    <th>Name</th>
    <th>Salary</th>
  </tr>
  <tr>
    <td>Ramesh Raman</td>
    <td>5000</td>
  </tr>
  <tr>
    <td>Shabbir Hussein</td>
    <td>7000</td>
  </tr>
</table>
</body>
</html>

```

Name	Salary
Ramesh Raman	5000
Shabbir Hussein	7000

### Rowspan & Colspan

The rowspan attribute in HTML specifies the number of rows a cell should span. That is if a row spans two rows, it means it will take up the space of two rows in that table. It allows the single table cell to span the height of more than one cell or row. It provides the same functionality as “merge cell” in a spreadsheet program like Excel.

#### Syntax:

```
<td rowspan = "value">table content...</td>
```

```

<!doctype html>
<html dir="ltr" lang="en">
<head>
<meta charset="utf-8">

```

```

<title>HTML Tables</title>
</head>
<body>
<table border="1">
<tr>
<th>DAY</th>
<td>8.30-9.30</td>
<td>9.30-10.30</td>
<td>10.30-11.00</td>
</tr>
<tr>
<th> Monday</th>
<td>DBMS</td>
<td>WT</td>
<td rowspan="2"> TEA BREAK </td>
</tr>
<tr>
<th> Tuesday</th>
<td> JAVA</td>
<td>SE</td>
</tr>
</table>
</body>
</html>

```

DAY	8.30-9.30	9.30-10.30	10.30-11.00
Monday	DBMS	WT	TEA BREAK
Tuesday	JAVA	SE	

### Colspan:

The colspan attribute in HTML specifies the number of columns a cell should span. It allows the single table cell to span the width of more than one cell or column.

### Syntax:

```
<td colspan = "value">table content...</td>
```

```

<!doctype html>
<html dir="ltr" lang="en">
<head>
<meta charset="utf-8">
<title>HTML Tables</title>

```



```

</head>
<body>
<table border="1">
<tr>
<th>DAY</th>
<td>8.30-9.30</td>
<td>9.30-10.30</td>
</tr>
<tr>
<th> Monday</th>
<td>DBMS</td>
<td>WT</td>
</tr>
<tr>
<th> Tuesday</th>
<td colspan="2"> WT Lab</td>
</tr>
</table>
</body>
</html>

```

DAY	8.30-9.30	9.30-10.30
Monday	DBMS	WT
Tuesday	WT Lab	

## Progress

The HTML 5 <progress> Tag is used to represent the progress of a task. It is similar to an indicator that displays the progress of completing a task. It is not used to represent the disk space or relevant query.

### Syntax:


```
<progress attributes...> </progress>
```

```

<!doctype html>
<html dir="ltr" lang="en">
<head>
<meta charset="utf-8">
<body>
<h3>The progress element</h3>
<label>Downloading progress:</label>
<progress value="32" max="100"> 32% </progress>
</body>
</html>

```

### The progress element

Downloading progress: 

### Attributes

Value	Descriptions
max	It represents the total work that is to be done to complete a task.

value	It represents the amount of work that is already completed.
-------	---

## Media tags

HTML **media tags** allow embedding and controlling **audio, video, and images** directly in a webpage. These media element tags changed the entire development using HTML.

The key media elements include:

Media Tag	Description
<audio>	Embeds audio files
<video>	Embeds video files
<source>	Specifies multiple media sources
<embed>	Embeds external media (e.g., PDFs, Flash)
<track>	Adds subtitles/captions to videos
<object>	Embeds other objects like Flash, PDFs

### <audio> Tag:

It is a useful tag if you want to add audio such as songs, or any sound files into your webpage.

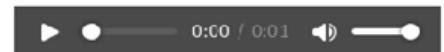
#### Syntax:

<audio>

`<source src="sample.mp3" type="audio/mp3">`

</audio>

```
<audio controls autoplay>
  <source src=" C:/Users/Public/Music/sample.mp3"
    type="audio/mp3">
</audio>
```



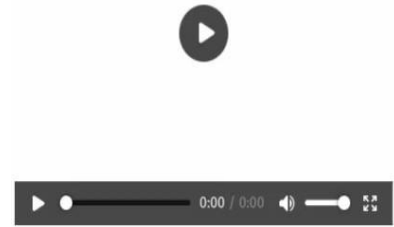
### <video> Tag:

It is a standard way to embed a video into your web page.

#### Syntax:

`<video src="" controls> </video>`

```
<video width="400" height="350" controls preload>
  <source src="C:/Users/Public/Video/sample.mp4"
type="video/mp4">
</video>
```

**<embed> Tag:**

It is used as a container for embedding plug-ins such as flash animations.

**Syntax:**

```
<embed src="C:/Users/Public/picture/sample.gif" width="300px" height="300px">
```

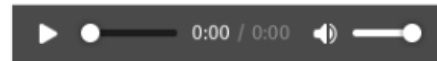
**<source> Tag:**

As you can observe that <audio>, <video> elements contain the <source> element, the <source> tag is used to attach multimedia files.

**Syntax:**

```
<source src="" type="">
...
</source>
```

```
<audio controls>
  <source src="audio.mp3"
type="audio/mp3">
</audio>
```

**<track> Tag:**

It is used to specify subtitles, caption files, or different files containing text, that ought to be visible once the media is taking part in it. Thus, it is a simple sector for the <audio> and <video> elements.

**Syntax:**

```
<track Attribute>
```

**FORM**

An HTML form is used to collect user input and send data to a web server. It contains form elements like text fields, buttons, checkboxes, radio buttons, dropdowns, etc.

An HTML form facilitates the user to enter data that is to be sent to the server for processing such as name, email address, password, phone number, etc.

The HTML `<form>` element is used to create an HTML form for user input:

**Syntax:**

```
form>  
.  
form elements  
.  
</form>
```

The `<form>` element is a container for different types of input elements, such as: text fields, checkboxes, radio buttons, submit buttons, etc.

Tag	Description
<code>&lt;form&gt;</code>	It defines an HTML form to enter inputs by the user side.
<code>&lt;input&gt;</code>	It defines an input control.
<code>&lt;textarea&gt;</code>	It defines a multi-line input control.
<code>&lt;label&gt;</code>	It defines a label for an input element.
<code>&lt;fieldset&gt;</code>	It groups the related elements in a form.
<code>&lt;legend&gt;</code>	It defines a caption for a <code>&lt;fieldset&gt;</code> element.
<code>&lt;select&gt;</code>	It defines a drop-down list.
<code>&lt;optgroup&gt;</code>	It defines a group of related options in a drop-down list.
<code>&lt;option&gt;</code>	It defines an option in a drop-down list.
<code>&lt;button&gt;</code>	It defines a clickable button.

**The `<input>` Element**

The HTML `<input>` element is a fundamental form element. It is used to create form fields, to take input from user. We can apply different input types to gather different information from user.

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

```

<!doctype html>
<html dir="ltr" lang="en">
<head>
<meta charset="utf-8">
<body>
<form>
  Enter your name <br>
  <input type="text" name="username">
</form>
</body>
</html>

```

Enter your name

### TextField Control

The `type="text"` attribute of `input` tag creates textfield control also known as single line textfield control. The `name` attribute is optional.

```

<form>
First Name: <input type="text" name="firstname"/> <br>
Last Name: <input type="text" name="lastname"/> <br>
</form>

```

First Name:

Last Name:

### `<textarea>` tag in form

The `<textarea>` tag in HTML is used to insert multiple-line text in a form. The size of `<textarea>` can be specify either using `"rows"` or `"cols"` attribute or by CSS.

```
<!doctype html>
```

```
<html dir="ltr" lang="en">
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<title>Form in HTML</title>
```

```
</head>
```

```
<body>
```

```
<form>
```

```
Enter your address:<br>
```

```
<textarea rows="2" cols="20"></textarea>
```

```
</form>
```

```
</body>
```

```
</html>
```

Enter your address:

### Password Field Control

The password is not visible to the user in password field control.

```
<form>
```

```
<label for="password">Password: </label>
```

```
<input type="password" id="password" name="password"/>
```

```
<br/>
```

```
</form>
```

Password:

### Radio Button Control

The radio button is used to select one option from multiple options. It is used for selection of gender, quiz questions etc. If you use one name for all the radio buttons, only one radio button can be selected at a time. Using radio buttons for multiple options, you can only choose a single option at a time.

```
<form>
```

```
<label for="gender">Gender: </label>
```

```
<input type="radio" id="gender" name="gender"
```

```
value="male"/>Male
```

```
<input type="radio" id="gender" name="gender"
```

```
value="female"/>Female <br/>
```

```
</form>
```

Gender: ☐ Male ☒ Female

### Checkbox Control

The checkbox control is used to check multiple options from given checkboxes.

```
<form>

Hobby:<br>

<input type="checkbox" id="cricket" name="cricket"
value="cricket"/> Cricket <br>

<input type="checkbox" id="football" name="football"
value="football"/> Football <br>

<input type="checkbox" id="hockey" name="hockey"
value="hockey"/> Hockey

</form>
```

Hobby:

☒ Cricket

☒ Football

☐ Hockey

### Selection/ Dropdown List

Dropdowns are used in web forms to provide a compact interface for selecting items from a list. HTML dropdowns are created using the **<select>** and **<option>** elements.

#### Syntax:

```
<select>

  <option value="option1">Option 1</option>

  <option value="option2">Option 2</option>

  <option value="option3">Option 3</option>

  <!-- Add more options as needed -->

</select>
```

```
<!doctype html>

<html dir="ltr" lang="en">

<head>

  <meta charset="utf-8">

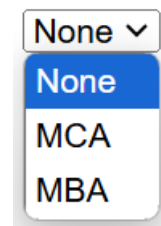
<body>

<form>

<select>

  <option value="None"> None </option>

  <option value="MCA" > MCA</option>
```



```
<option value="MBA" > MBA</option>

</form>

</body>

</html>
```

### Submit button control

HTML `<input type="submit">` are used to add a submit button on web page. When user clicks on submit button, then form get submit to the server.

#### Syntax:

```
<input type="submit" value="submit">
```

```
<form>

  <label for="name">Enter name</label><br>
  <input type="text" id="name" name="name"><br>
  <label for="pass">Enter Password</label><br>
  <input type="Password" id="pass" name="pass"><br>
  <input type="submit" value="submit">

</form>
```

Enter name

Enter Password

submit

### HTML Program to create a simple Form

```
<!doctype html>

<html dir="ltr" lang="en">

<head>

<meta charset="utf-8"> <title>Form in HTML</title>

</head>

<body>

<h2>Registration form</h2>

<form>

  <fieldset>

    <legend>User personal information</legend>

    <label>Enter your full name</label><br>

    <input type="text" name="name"><br> <br>

    <label>Enter your email</label><br>
```



```
<input type="email" name="email"><br> <br>
<label>Enter your password</label><br>
<input type="password" name="pass"><br> <br>
<label>confirm your password</label><br>
<input type="password" name="pass"><br> <br>
<br><label>Select gender</label><br>
<input type="radio" id="gender" name="gender" value="male"/>Male
<input type="radio" id="gender" name="gender" value="female"/>Female <br/>
<br>Enter your Address:<br>
<textarea></textarea><br> <br>
<label>Course: </label>
<select>
<option value="None"> None </option>
<option value="MCA" > MCA</option>
<option value="MBA" > MBA</option>
</select>
<br>
<br>
<input type="submit" value="Register" >
</fieldset>
</form>
</body>
</html>
```

## Registration form

User personal information

Enter your full name

Enter your email

Enter your password

confirm your password

Select gender  
☐ Male ☐ Female

Enter your Address:

Course:

### span and div tags

The **<span>** tag is an inline container used to group text or elements for styling and JavaScript manipulation. It does not break the line, making it useful for modifying part of a sentence.

#### Syntax:

```
<span>Some text</span>
```

Ex:

```
<p>Today's temperature is <span style="color: blue;">25°C</span>.</p>
```

The **<div>** (short for "division") tag is a block-level container used to group multiple elements together for styling, layout, and JavaScript manipulation.

It helps in

- Acts as a flexible layout tool
- Helps organize webpage sections
- Easier to apply CSS styles

**Syntax:**

<div> Some text </div>

**Difference Between <div> and <span>**

Feature	<div>	<span>
Display Type	Block	Inline
Use Case	Grouping sections	Styling part of a text
Line Break?	Yes	No

```
<!doctype html>
  <html dir="ltr" lang="en">
  <head>
    <meta charset="utf-8">
  <title>Span Tag</title>
</head>
<body>
  <h2>Example of span tag</h2>
  <div>
    <p>I have choosen only
      <span style="color: red;">red</span>,
      <span style="color: blue;">blue</span>, and
      <span style="color: green;">green</span>
    colors for my painting.
  </p>
  </div>
</body>
</html>
```

**Example of span tag**

I have choosen only red, blue, and green colors for my painting.