

BCA/IMCA SEMESTER 1

Fundamentals of Web Technology (BC01001031)

UNIT 2 : HTML, XHTML & HTML5

1. Introduction to HTML

What is HTML?

- **HTML (HyperText Markup Language)** is the backbone of web development.
- It defines **the structure and content** of a web page.
- Uses **tags** (keywords inside angle brackets < >) to tell the browser how to display content.

Example:

```
<p>Hello, this is a paragraph.</p>
```

□ **Note:** HTML only structures content; it does not add advanced design (done by CSS) or interactivity (done by JavaScript).

HTML Tags and Attributes

- **Tags** are used to mark elements.
 - Example: <h1>Heading</h1>
 - **Attributes** give extra information about elements.
 - Example:
 - Here:
 - src="car.jpg" → tells where the image is stored.
 - alt="Car Image" → alternative text if image fails to load.
-

HTML File Structure

A proper HTML file follows a standard structure:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>My First Page</title>
  </head>
  <body>
    <h1>Welcome to HTML</h1>
    <p>This is a simple HTML document.</p>
  </body>
</html>
```

- **<!DOCTYPE html>** – Declares HTML5 version.
 - **<html>** – Root element.
 - **<head>** – Metadata (not visible to users, but useful for search engines/browsers).
 - **<meta>** – Defines character encoding, description, keywords.
 - **<title>** – Sets page title (shown in browser tab).
 - **<body>** – All visible content (text, images, videos, forms).
-

2. Common HTML Tags

HTML Headings (<h1> to <h6>)

Definition:

- Headings are used to define **titles and subtitles** in an HTML document.
 - They are written using tags <h1> to <h6>.
 - <h1> is the **largest and most important heading**, while <h6> is the **smallest and least important**.
-

Syntax:

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

Example Output (in a browser):

Heading 1
Heading 2
Heading 3
Heading 4
Heading 5
Heading 6

Features of Headings:

1. **Block-level elements** – each heading starts on a new line.
 2. **Automatically bold** – browsers display them in bold by default.
 3. **SEO Importance** – search engines (like Google) use `<h1>`, `<h2>`, etc., to understand page structure.
 4. **Accessibility** – screen readers use heading tags to help users navigate pages.
 5. **Hierarchy** – `<h1>` should be used once per page as the main title, while `<h2>`–`<h6>` are used for sub-sections.
-

Example – Article with Headings:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Heading Example</title>
  </head>
  <body>
    <h1>Web Technology</h1>
    <p>This is the main subject.</p>

    <h2>Introduction</h2>
    <p>Basics of the Internet, HTTP, and Web Servers.</p>

    <h3>History</h3>
    <p>Evolution of web technology from 1990s.</p>

    <h2>Applications</h2>
    <p>Websites, mobile apps, e-commerce, social networking.</p>
  </body>
</html>
```

•

Paragraphs & Line Breaks

- `<p>` defines paragraphs.
- `
` adds a line break.

- `<p>This is a paragraph.</p>`
- `Line 1
Line 2`

1. Paragraphs in HTML (`<p>`)

Definition:

- The **`<p>` tag** in HTML is used to define a **paragraph of text**.
- A paragraph is a **block-level element**, meaning it always starts on a new line and adds space before and after the text.

Syntax:

```
<p>This is a paragraph.</p>
```

Example:

```
<p>HTML is the standard language for creating web pages.</p>
<p>It uses tags to structure the content of a website.</p>
```

Output in Browser:

HTML is the standard language for creating web pages.

It uses tags to structure the content of a website.

□ Notice how each `<p>` starts on a **new line** with space above and below.

2. Line Breaks in HTML (`
`)

Definition:

- The **`
` tag** is used to insert a **line break** within text.
- Unlike `<p>`, it does not create extra spacing, just moves text to the next line.
- It is an **empty tag** (no closing tag required).

Syntax:

```
Line 1<br>Line 2
```

Example:

```
<p>Address:<br>
```

```
123 Main Street<br>
New York, USA</p>
```

Output in Browser:

Address:
123 Main Street
New York, USA

3. Difference Between `<p>` and `
`

Feature	<code><p></code> (Paragraph)	<code>
</code> (Line Break)
Purpose	Creates a new paragraph	Breaks line inside text
Extra Space	Adds margin before & after	No extra margin
Type	Block-level element	Inline element
Closing Tag	Requires <code></p></code>	Does not need closing tag
Example	<code><p>Text</p></code>	Line 1 Line 2

Anchors & Links (`<a>`)

- Used for hyperlinks.
- `Visit Wikipedia`
- **Attributes:**
 - `href` → link destination.
 - `target="_blank"` → open in new tab.

1. Introduction

- The **anchor tag** `<a>` is one of the most important tags in HTML.
 - It is used to **create hyperlinks**, which allow users to navigate from one page to another or from one section to another within the same page.
 - The anchor tag turns text, images, or other elements into clickable links.
-

2. Syntax

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

- `<a>` → Anchor tag.
- **href (Hypertext Reference)** → Specifies the destination of the link.

- **Link Text** → The clickable part visible to the user.
-

3. Example (Basic Link)

```
<a href="https://www.google.com">Visit Google</a>
```

Output in Browser:

□ Visit Google

When clicked, this opens Google's homepage.

Lists

- **Ordered list** () → Numbers/letters.
- **Unordered list** () → Bullets.
- **Definition list** (<dl>) → Term-definition pairs.

Example:

```
<ol>
  <li>HTML</li>
  <li>CSS</li>
</ol>
<ul>
  <li>Apple</li>
  <li>Banana</li>
</ul>
<dl>
  <dt>HTTP</dt>
  <dd>HyperText Transfer Protocol</dd>
</dl>
```

1. Ordered List ()

Definition:

- Displays list items in a **specific sequence** (numbered, alphabetically, or Roman numerals).
- Each item is defined using the (list item) tag.

Syntax:

```
<ol>
  <li>HTML</li>
  <li>CSS</li>
  <li>JavaScript</li>
</ol>
```

Default Output:

1. HTML
2. CSS
3. JavaScript

Attributes of :

- `type` → Defines numbering style.
 - `"1"` → Numbers (default)
 - `"A"` → Uppercase letters
 - `"a"` → Lowercase letters
 - `"I"` → Roman numerals (uppercase)
 - `"i"` → Roman numerals (lowercase)
- `start` → Defines starting value.

Example:

```
<ol type="A" start="3">
  <li>Option One</li>
  <li>Option Two</li>
</ol>
```

Output:

- C. Option One
D. Option Two
-

2. Unordered List ()**Definition:**

- Displays items with **bullet points** (no specific order).
- Each item is created using ``.

Syntax:

```
<ul>
  <li>Apple</li>
  <li>Banana</li>
  <li>Mango</li>
</ul>
```

Default Output:

- Apple
- Banana
- Mango

Attributes of :

- type (HTML4, deprecated in HTML5 but still works in some browsers):
 - "disc" → ● (default)
 - "circle" → ○
 - "square" → ■

Example:

```
<ul type="square">
  <li>Red</li>
  <li>Green</li>
  <li>Blue</li>
</ul>
```

Output:

- Red
- Green
- Blue

3. Definition List (<dl>)

Definition:

- Used for **term-definition pairs**, like dictionaries or glossaries.
- Consists of:
 - <dt> → Definition Term
 - <dd> → Definition Description

Syntax:

```
<dl>
```



```
<dt>HTTP</dt>
<dd>HyperText Transfer Protocol</dd>

<dt>HTML</dt>
<dd>HyperText Markup Language</dd>
</dl>
```

Output:

HTTP

HyperText Transfer Protocol

HTML

HyperText Markup Language

Nested Lists (Lists inside Lists)

Example:

```
<ol>
  <li>Frontend
    <ul>
      <li>HTML</li>
      <li>CSS</li>
    </ul>
  </li>
  <li>Backend
    <ul>
      <li>PHP</li>
      <li>Python</li>
    </ul>
  </li>
</ol>
```

Output:

1. Frontend
 - HTML
 - CSS
 2. Backend
 - PHP
 - Python
-

Tables

- Used for structured data.

```
<table border="1">
  <tr>
    <th>Name</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Rahul</td>
    <td>21</td>
  </tr>
</table>
```

- **<tr>** → Table row.
- **<th>** → Table header (bold).
- **<td>** → Table data (cell).

1. Introduction

The `<table>` tag in HTML is used to display data in **rows and columns**. It is useful for organizing structured information like timetables, marksheets, or product listings.

2. Basic Structure of a Table

A table is made up of several tags:

Tag	Meaning
<code><table></code>	Defines a table
<code><tr></code>	Table Row
<code><th></code>	Table Header Cell (bold, centered by default)
<code><td></code>	Table Data Cell

Syntax:

```
<table>
  <tr>
    <th>Header 1</th>
    <th>Header 2</th>
  </tr>
  <tr>
    <td>Data 1</td>
```

```

        <td>Data 2</td>
    </tr>
</table>

```

Output:

Header 1 Header 2

Data 1 Data 2

3. Attributes of <table>

□ Table-level Attributes

Attribute	Description	Example
border	Adds border around table cells	<table border="1">
cellpadding	Space inside each cell (padding)	<table cellpadding="10">
cellspacing	Space between cells	<table cellspacing="5">
width	Sets width of the table	<table width="500">
align (HTML4, replaced by CSS)	Aligns table (left/center/right)	<table align="center">

□ Row/Column Attributes

Attribute	Applies To	Description
rowspan	<td> / <th>	Merge cells vertically
colspan	<td> / <th>	Merge cells horizontally
bgcolor (HTML4, use CSS in HTML5)	<td>, <th>, <tr>	Sets background color
align	<td> / <th>	Aligns text (left/center/right)

4. Example 1 – Simple Table

```

<table border="1" cellpadding="5" cellspacing="2">
  <tr>
    <th>Name</th>
    <th>Age</th>
    <th>City</th>
  </tr>
  <tr>
    <td>Riya</td>
    <td>21</td>

```

```

        <td>Mumbai</td>
    </tr>
    <tr>
        <td>Aman</td>
        <td>22</td>
        <td>Delhi</td>
    </tr>
</table>

```

Output:

Name	Age	City
Riya	21	Mumbai
Aman	22	Delhi

5. Example 2 – Using rowspan and colspan

```

<table border="1" cellpadding="8" cellspacing="0">
    <tr>
        <th rowspan="2">Name</th>
        <th colspan="2">Marks</th>
    </tr>
    <tr>
        <th>Math</th>
        <th>Science</th>
    </tr>
    <tr>
        <td>Riya</td>
        <td>85</td>
        <td>90</td>
    </tr>
    <tr>
        <td>Aman</td>
        <td>78</td>
        <td>88</td>
    </tr>
</table>

```

Output:

Name	Math	Science
Riya	85	90
Aman	78	88

□ Note how **rowspan** merges rows and **colspan** merges columns.

Frames (Old HTML)

- Divides page into multiple sections.
- Example:
- `<frameset cols="50%,50%">`
- `<frame src="left.html">`
- `<frame src="right.html">`
- `</frameset>`

□ **Note:** Frames are **deprecated in HTML5** (replaced by **CSS Flexbox/Grids**).

Forms

- Used to collect user input.

```
<form action="submit.php" method="post">
  Name: <input type="text" name="username"><br>
  Email: <input type="email" name="email"><br>
  <input type="submit" value="Submit">
</form>
```

- **action** → Script where data is sent.
 - **method="get/post"** → Defines submission type.
 - **Form elements:** text box, radio button, checkbox, dropdown, submit button.
-

3. XHTML (Extensible HTML)

- Developed to overcome **loose HTML coding**.
- Combines **HTML + XML rules**.
- More **strict, cleaner, and portable**.

XHTML Rules

- All tags must be properly closed.
 - Wrong: `
` ✓ Correct: `
`
- Tags & attributes must be lowercase.
- Attribute values must be quoted.
- Documents must start with a proper DOCTYPE.

HTML Validator

- Tool to check correctness of code.
 - Example: W3C Validator.
-

Block vs Inline Elements

- **Block-level elements:** Start on new line, occupy full width.
 - Examples: <div>, <p>, <h1>, <table>.
 - **Inline elements:** Stay within text flow.
 - Examples: , <a>, , .
-

4. HTML5 – The Latest Version

- Released officially in **2014**.
- Supports **modern web apps** with multimedia, storage, graphics, and mobile features.

Key Features of HTML5

1. **Semantic Tags**
 - Provide meaning to content.
 - Examples: <header>, <footer>, <section>, <article>, <nav>, <aside>.

Example:

```
<article>
  <h2>News Article</h2>
  <p>Latest updates from technology world...</p>
</article>
```

2. Multimedia Support

- <video> and <audio> allow embedding media without plugins.
 - <video controls>
 - <source src="movie.mp4" type="video/mp4">
 - </video>
 - <audio controls>
 - <source src="song.mp3" type="audio/mpeg">
 - </audio>
-

3. Form Enhancements

- New input types:
 - email, url, date, time, range, color, number.

Example:

```
<form>
  Email: <input type="email"><br>
  Date: <input type="date"><br>
  Age: <input type="number" min="18" max="60">
</form>
```

4. Graphics & Animation

- `<canvas>`: Draw shapes, charts, and animations with JavaScript.
 - `<svg>`: Scalable vector graphics (resolution independent).
-

5. Offline Storage & APIs

- `localStorage` – stores data permanently in browser.
 - `sessionStorage` – stores data until tab is closed.
 - **Geolocation API**: Get user's location.
 - **Drag & Drop API**: Move items within a page.
-

6. Mobile-Friendly & Responsive Design

- Meta viewport tag for scaling on devices:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```