# Module 2

# Mernstack – HTML

--------------------------------------------------------------------------------------------------------------------------------------Question 1: Define HTML. What is the purpose of HTML in web development?

Answer:

HTML stands for **HyperText Markup Language**.  
 It is the **standard markup language** used to create and structure content on the web.

**Purpose of HTML in web development:**

1. **Structure the content:**  
   HTML provides the basic **skeleton or layout** of a webpage by using elements like headings, paragraphs, lists, tables, and more.
2. **Embed media:**  
   It allows the inclusion of **images, videos, audios, and other media** using appropriate tags.
3. **Linking pages:**  
   HTML enables **navigation between web pages** using hyperlinks (<a> tags), forming the structure of websites.
4. **Semantic meaning:**  
   HTML5 introduces **semantic elements** (like <header>, <footer>, <article>, etc.) that describe the **meaning of content**, improving SEO and accessibility.
5. **Foundation for other technologies:**  
   HTML works together with **CSS (for styling)** and **JavaScript (for interactivity)**, forming the **core of front-end web development**.

Question 2: Explain the basic structure of an HTML document. Identify the mandatory tags and their purposes.

Answer:

<html>

<head>

<title>My Web Page</title>

</head>

<body>

<h1>Hello, World!</h1>

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

</body>

</html>

Mandatory Tags and Their Purpose:

**<html> ... </html>**

* **What it does:**  
  This is the **root element** of the HTML document.
* **Why it’s important:**  
  It wraps everything — both the head and body sections.
* **Note:** All HTML code must be written **inside** this tag.

**<head> ... </head>**

* **What it does:**  
  The head section contains **information about the document** (not visible on the webpage).
* **Why it’s important:**  
  It helps browsers, search engines, and other tools understand the page.

**<title>My Web Page</title>**

* **What it does:**  
  Defines the **title of the webpage** shown in the browser’s title/tab bar.
* **Why it’s important:**  
  It improves **SEO** and helps users identify the tab.

**body> ... </body>**

* **What it does:**  
  This is where all the **visible content** of the webpage goes — text, images, links, tables, etc.
* **Why it’s important:**  
  Everything written inside <body> is shown on the screen when a user opens the website.

| **Tag** | **Purpose** |
| --- | --- |
| <h1> to <h6> | Headings (h1 is largest, h6 is smallest). |
| <p> | Paragraphs of text. |
| <br> | Line break (no closing tag). |
| <hr> | Horizontal line (divider). |
| <b> | Bold text. |
| <i> | Italic text. |
| <u> | Underlined text. |
| <small> | Smaller font size. |
| <strong> | Strong importance (bold with meaning). |
| <em> | Emphasized text (italic with meaning). |
| Tag | Purpose |
| <a href="URL"> | Creates a hyperlink to another page or site. |
| target="\_blank" | Opens link in a new tab or window. |
| <name> / id | Used for linking to specific sections on the same page. |

| **Tag** | **Purpose** |
| --- | --- |
| <img src="path.jpg" alt="description"> | Displays an image. |
| width / height | Set image dimensions. |
| <embed> | Embeds multimedia like Flash or PDF. |
| <object> | Embeds external content like videos or plugins. |
| Tag | Purpose |
| <div> | Block-level container (used to group sections). |
| <span> | Inline-level container (used for styling small text pieces). |
| <center> | Centers content (deprecated but used in older HTML). |
| <table> | Creates a table. |
| <tr> | Table row. |
| <td> | Table data cell. |
| <th> | Table header cell. |
| <form> | Creates a form for user input. |
| Tag | Purpose |
| <form> | Container for form elements. |
| <input type="text"> | Text input field. |
| <input type="submit"> | Submit button. |
| <textarea> | Multiline text box. |
| <select> and <option> | Dropdown menu. |
| <label> | Label for input fields. |
| <fieldset> / <legend> | Groups related inputs with a caption. |

Question 3: What is the difference between block-level elements and inline elements inHTML? Provide examples of each.

Answer:

**1. Block-Level Elements**

**Definition:**

Block-level elements **start on a new line** and take up the **full width** available (by default).  
They are used to create **larger sections or blocks** of content.

Block Elements:

**lock Elements:**

* <div>
* <p>
* <h1> to <h6>
* <ul>, <ol>, <li>
* <table>, <tr>, <td>
* <form>

**Example:**

<html>

<head>

<title>block elements</title>

</head>

<body>

<div>This is a block element</div>

<p>This is a paragraph</p>

<h1>This is a heading</h1>

<ul>

<li>List item</li>

</ul>

<table>

<tr><td>Cell</td></tr>

</table>

</body>

</html>

**Definition:**

Inline elements **do not start on a new line** and take up **only as much width** as needed.  
They are used to **style or format** parts of text **within a block**.

**Inline Elements:**

* <span>
* <a>
* <b>, <i>, <u>
* <img>
* <input>
* <label>

Example:

<span>This is inline</span>

<a href="#">This is a link</a>

<b>Bold Text</b>

<i>Italic Text</i>

<img src="pic.jpg" alt="image">

Question 4: Discuss the role of semantic HTML. Why is it important for accessibility and SEO? Provide examples of semantic elements.

Answer:

**Semantic HTML** refers to the use of **HTML tags that clearly describe the meaning and structure of the content** they enclose.

Instead of using generic tags like <div> or <span>, semantic tags **define their purpose**, making the code more **readable**, **accessible**, and **search engine-friendly**.

Why Semantic HTML is Important:

**Accessibility (for Screen Readers)**

* Semantic elements give **meaning** to the content.
* Screen readers and assistive technologies rely on these tags to **understand page structure**, like identifying headers, navigation, or articles.
* Helps users with disabilities **navigate the page** more easily.

**2. SEO (Search Engine Optimization)**

* Search engines like Google use semantic tags to better **understand the content** of a page.
* Improves **indexing** and increases chances of ranking higher in search results.
* Helps bots recognize key sections like headers, articles, or sidebars.

**3. Code Readability & Maintainability**

* Developers can quickly understand the layout and purpose of each section.
* Easier collaboration and maintenance.

**Examples of Semantic HTML Elements:**

| **Semantic Tag** | **Purpose** |
| --- | --- |
| <header> | Defines the header of a section or page. |
| <nav> | Defines a navigation menu. |
| <main> | Contains the main content of the page. |
| <section> | Defines a logical grouping of content. |
| <article> | Defines a standalone piece of content (e.g., blog post). |
| <aside> | Sidebar content, like related links or ads. |
| <footer> | Contains footer info like contact or copyright. |
| <figure> and <figcaption> | Used for images with captions. |

**Lab Assignment:**

**Task: Create a simple HTML webpage that includes: • A header (), footer (), main section (), and aside section (). • A paragraph with some basic text. • A list (both ordered and unordered). • A link that opens in a new tab.**

Answer:

<

<html >

<head>

  <title>Page Using &lt;sidebar&gt; </title>

  <style>

    body {

      font-family: Arial, sans-serif;

      margin: 0;

      padding: 0;

    }

    header, footer {

      background-color: #333;

      color: white;

      padding: 15px;

      text-align: center;

    }

    main {

      display: flex;

      padding: 20px;

    }

    section {

      flex: 3;

      padding-right: 20px;

    }

    sidebar {

      flex: 1;

      background-color: #f0f0f0;

      padding: 15px;

      /\* Make sure sidebar behaves like a block \*/

      display: block;

    }

    footer {

      margin-top: 20px;

    }

  </style>

</head>

<body>

  <header>

    <h1>My Simple HTML Page</h1>

  </header>

  <main>

    <section>

      <h2>Welcome!</h2>

      <p>This is a paragraph with some basic text to show how content appears on a webpage.</p>

      <h3>My To-Do List</h3>

      <ul>

        <li>Buy groceries</li>

        <li>Finish project</li>

        <li>Call mom</li>

      </ul>

      <h3>Steps to Complete</h3>

      <ol>

        <li>Plan the task</li>

        <li>Work on it</li>

        <li>Review and submit</li>

      </ol>

      <p>

        Visit <a href="https://www.google.com" target="\_blank">Google</a> in a new tab.

      </p>

    </section>

    <sidebar>

      <h3>Sidebar</h3>

      <p>This is a sidebar section using the &lt;sidebar&gt; tag (non-standard).</p>

    </sidebar>

  </main>

  <footer>

    <p>&copy; 2025 My Simple Webpage. All rights reserved.</p>

  </footer>

</body>

</html>

HTML Forms:

Theory Assignment:

Question 1: What are HTML forms used for? Describe the purpose of the input, textarea, select, and buttonelements.

Answer:

HTML forms are used to **collect user input** on a webpage and send it to a server or process it on the client side. Forms allow users to enter data such as text, choices, or files, which can then be submitted for tasks like signing up, logging in, searching, or sending messages.

| **Element** | **Purpose** |
| --- | --- |
| <input> | A versatile element to receive user data in many formats — text, password, email, number, checkbox, radio buttons, and more. Its behavior depends on the type attribute. For example, <input type="text"> is for single-line text input. |
| <textarea> | A multiline text input field used when users need to enter longer blocks of text, like comments or messages. Unlike <input>, it can expand over multiple lines. |
| <select> | A dropdown menu that allows users to select one (or multiple) options from a list. It contains <option> elements inside to define choices. |
| <button> | A clickable button that can submit a form (type="submit"), reset it (type="reset"), or be used for custom scripts (type="button"). Buttons can contain text or HTML content. |

• Question 2: Explain the difference between the GETand POSTmethods in form submission. When should each be used?

**GET**:

* When you want to **retrieve** or request data.
* When form data is **not sensitive**.
* When bookmarking or sharing the URL with parameters is useful (e.g., search pages).
* Example: Search forms, filters.

**POST**:

* When you want to **send or submit data** that changes the server state.
* When the data is **sensitive** (like passwords).
* When submitting large amounts of data.
* Example: Login forms, registration, posting comments.

Question 3: What is the purpose of the label element in a form, and how does it improve accessibility?

Answer:  
**Purpose of the <label> Element:**

The <label> element in HTML forms is used to **define a text description for form input fields** (like <input>, <textarea>, or <select>). It tells users what information is expected in a field.

**How It’s Used:**

There are two ways to link a <label> with a form element:

**Using the for attribute** (preferred method):

**Example:**

<label for="email">Email Address:</label>

<input type="email" id="email" name="email">

**Lab Assignment**

• Task: Create a contact form with the following fields: • Full name (text input) Email (email input) • Phone number (tel input) • Subject (dropdown menu) • Message (textarea) • Submit button Additional Requirements: • Use appropriate form validation using required, minlength, maxlength, and pattern. • Link form labels with their corresponding inputs using the forattribute.

Answer:

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>Contact Form</title>

  <style>

    body {

      font-family: Arial, sans-serif;

      padding: 20px;

      background-color: #f5f5f5;

    }

    form {

      background-color: #fff;

      padding: 25px;

      max-width: 500px;

      margin: auto;

      border-radius: 10px;

      box-shadow: 0 0 10px rgba(0,0,0,0.1);

    }

    label {

      display: block;

      margin-bottom: 5px;

      font-weight: bold;

    }

    input, select, textarea {

      width: 100%;

      padding: 8px;

      margin-bottom: 20px;

      border: 1px solid #ccc;

      border-radius: 5px;

      font-size: 16px;

    }

    button {

      padding: 10px 20px;

      font-size: 16px;

      background-color: #007bff;

      color: white;

      border: none;

      border-radius: 5px;

      cursor: pointer;

    }

    button:hover {

      background-color: #0056b3;

    }

  </style>

</head>

<body>

  <form action="#" method="post">

    <!-- Full Name -->

    <label for="fullname">Full Name:</label>

    <input type="text" id="fullname" name="fullname" required minlength="3" maxlength="50" pattern="[A-Za-z\s]+" title="Only letters and spaces allowed">

    <!-- Email -->

    <label for="email">Email:</label>

    <input type="email" id="email" name="email" required>

    <!-- Phone Number -->

    <label for="phone">Phone Number:</label>

    <input type="tel" id="phone" name="phone" required pattern="[0-9]{10}" title="Enter 10-digit phone number" maxlength="10">

    <!-- Subject Dropdown -->

    <label for="subject">Subject:</label>

    <select id="subject" name="subject" required>

      <option value="">-- Select a Subject --</option>

      <option value="general">General Inquiry</option>

      <option value="support">Support</option>

      <option value="feedback">Feedback</option>

      <option value="other">Other</option>

    </select>

    <!-- Message -->

    <label for="message">Message:</label>

    <textarea id="message" name="message" rows="5" required minlength="10" maxlength="500" placeholder="Type your message here..."></textarea>

    <!-- Submit Button -->

    <button type="submit">Submit</button>

  </form>

</body>

</html>

HTML Tables:

Theory Assignment:

Question 1: Explain the structure of an HTML table and the purpose of each of the

Following elements: <table>, <tr>, <th>, <td>, and <thead>.

**Answer:**

An HTML table is used to display data in a row-and-column format, similar to an Excel sheet. Tables are helpful for showing tabular data like schedules, product lists, or reports.

**Structure of an HTML Table**:

<table>

<thead>

<tr>

<th>Heading 1</th>

<th>Heading 2</th>

</tr>

</thead>

<tr>

<td>Row 1, Cell 1</td>

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

</tr>

</table>

| **Tag** | **Full Form** | **Purpose** |
| --- | --- | --- |
| <table> | — | The main container that defines the entire table. |
| <tr> | Table Row | Defines a single row in the table. Can be used in <thead>, <tbody>, or directly inside <table>. |
| <th> | Table Header | Defines a **header cell**. Text inside is **bold and centered by default**. Usually used inside <thead>. |
| <td> | Table Data | Defines a **standard cell** (table data). Used to hold actual content (text, images, links, etc.). |
| <thead> | Table Head | A section that groups all the **header rows** of the table. Helps with styling and accessibility. |

• Question 2: What is the difference between colspanand rowspanin tables? Provide examples.

Answer:

colspan and rowspan are **HTML attributes** used with <td> or <th> elements to **merge multiple columns or rows** in a table.

**Difference Between colspan and rowspan**

| **Attribute** | **Purpose** | **Works On** | **Example Use** |
| --- | --- | --- | --- |
| colspan | Merges **multiple columns** into one cell | <td> / <th> | Creating a title row across 2+ columns |
| rowspan | Merges **multiple rows** into one cell | <td> / <th> | Creating a cell that covers multiple rows |

**Example of colspan:**

html

CopyEdit

<table border="1">

<tr>

<th colspan="3">Student Details</th>

</tr>

<tr>

<th>Name</th>

<th>Age</th>

<th>Grade</th>

</tr>

<tr>

<td>Rahul</td>

<td>18</td>

<td>A</td>

</tr>

</table>

**Example of rowspan:**

html

CopyEdit

<table border="1">

<tr>

<th rowspan="2">Name</th>

<td>Rahul</td>

</tr>

<tr>

<td>Amit</td>

</tr>

</table>

Question 3: Why should tables be used sparingly for layout purposes? What is a better alternative?

**Answer:**

**Why Tables Should Be Used Sparingly for Layout:**

In the early days of web design, developers used <table> tags to create entire page layouts. However, this approach is now **discouraged** for many reasons:

| **Problem** | **Explanation** |
| --- | --- |
| **Poor accessibility** | Screen readers and assistive technologies have trouble interpreting table-based layouts, which are meant for data, not structure. |
| **Not responsive** | Tables do not adapt well to different screen sizes, especially on mobile devices. |
| **Hard to maintain** | Table-based layouts are rigid and make editing or updating the design difficult. |
| **Mixes content with design** | Tables combine structure with styling, violating the **separation of concerns** principle. |
| **Slower page load** | Tables require more code for layout than modern CSS-based techniques. |

**Better Alternative: Use CSS for Layout**

Modern websites use **CSS layout techniques** to create flexible, responsive designs.

**Recommended Alternatives:**

| **CSS Method** | **Description** |
| --- | --- |
| Flexbox | Great for 1D layouts (rows or columns). Simple and powerful. |
| CSS Grid | Ideal for 2D layouts (both rows and columns). Offers full control over layout. |
| Media Queries | Help build **responsive layouts** for all screen sizes. |

**Example Using CSS Flex Instead of Table:**

html

CopyEdit

<div style="display: flex; justify-content: space-between;">

<div>Left</div>

<div>Right</div>

</div>

Instead of using:

html

<table>

<tr>

<td>Left</td>

<td>Right</td>

</tr>

</table>

**Lab Assignment**

• Task: Create a product catalog table that includes the following columns: • Product Name • Product Image (use placeholder image URLs) • Price • Description • Availability (in stock, out ofstock) Additional Requirements: • Use theadfor the table header. • Add a border and some basic styling using inline CSS. • Use colspanor rowspanto merge cells where applicable.

Answer:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Product Catalog</title>

</head>

<body>

<h2 style="text-align: center;">Product Catalog</h2>

<table style="width: 90%; margin: auto; border-collapse: collapse;" border="1">

<!-- Table Header -->

<thead style="background-color: #f2f2f2;">

<tr>

<th style="padding: 10px;">Product Name</th>

<th style="padding: 10px;">Image</th>

<th style="padding: 10px;">Price</th>

<th style="padding: 10px;">Description</th>

<th style="padding: 10px;">Availability</th>

</tr>

</thead>

<!-- Product 1 -->

<tr>

<td style="padding: 10px;">Sneakers</td>

<td style="padding: 10px;">

<img src="https://via.placeholder.com/100" alt="Sneakers">

</td>

<td style="padding: 10px;">₹1,999</td>

<td style="padding: 10px;" rowspan="2">

Comfortable and stylish shoes perfect for daily wear.

</td>

<td style="padding: 10px;">In Stock</td>

</tr>

<!-- Product 2 shares the description using rowspan -->

<tr>

<td style="padding: 10px;">Running Shoes</td>

<td style="padding: 10px;">

<img src="https://via.placeholder.com/100" alt="Running Shoes">

</td>

<td style="padding: 10px;">₹2,499</td>

<td style="padding: 10px;">Out of Stock</td>

</tr>

<!-- Product 3 uses colspan for unavailable image -->

<tr>

<td style="padding: 10px;">Backpack</td>

<td style="padding: 10px;" colspan="2" align="center">Image Unavailable</td>

<td style="padding: 10px;">Durable backpack for school and travel use.</td>

<td style="padding: 10px;">In Stock</td>

</tr>

</table>

</body>

</html>

Bottom of Form