Theory

Module 2 – Mernstack – HTML

• HTML Basics

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

Ans:

HTML (HyperText Markup Language) is the standard markup language used to create and design the structure of web pages. It uses a system of tags and attributes to organize and format content such as text, images, links, and multimedia on the web. In simple terms, HTML is the language that tells web browsers what to display and how to display it.

The **purpose of HTML in web development** is to provide the **structure and layout** of a web page. It acts as the foundation for creating websites by organizing content into headings, paragraphs, images, links, and other elements.

Key purposes of HTML in web development:

1. **Structuring Content:**
   - HTML organizes text, images, and media into a readable and logical format.
2. **Creating Web Page Elements:**
   - It defines elements like headings, lists, tables, forms, and buttons.
3. **Linking Web Pages:**
   - HTML allows you to link multiple web pages together using hyperlinks (<a> tags).
4. **Embedding Media:**
   - You can add images, audio, videos, and more using HTML tags.
5. **Interacting with Other Technologies:**
   - HTML works with CSS for styling and JavaScript for interactivity, forming the core of front-end web development.

2. Explain the basic structure of an HTML document. Identify the mandatory tags and their purposes.1

Ans:

An HTML document follows a specific structure made up of mandatory tags that help the browser understand and render the content correctly. Here's a breakdown of the basic structure:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <h1>Main Heading</h1>
    <p>This is a paragraph.</p>
  </body>
</html>
```

- <!DOCTYPE html>
  - Declares the document type.
  - Tells the browser that the document is written in HTML5.

- <html>
  - The root element of the HTML document.
  - Wraps all other elements on the page.

- <head>
  - Contains metadata (data about the document), such as:
  - Title of the page (<title>)
  - Links to CSS files
  - Scripts

- <title>
  - Sets the title of the web page (shown in the browser tab).
  - Must be inside the <head> section.

- <body>
  - Holds everything that is visible to users:
  - Text
  - Images
  - Buttons
  - Links
  - Any other content you want to show on the page

3. What is the difference between block-level elements and inline elements in HTML? Provide examples of each.

Ans:

1. **Block-level Elements:**

- Start on a new line
- Take up the full width of the parent container (by default)
- Can contain other block-level or inline elements

- Examples:
  - – Generic container
  - <p> – Paragraph
  - <h1> to <h6> – Headings
  - , <ol>, <li> – Lists
  - <section>, <article>, <header>, <footer>

- Example Code:

```
<p>This is a paragraph.</p>
<div>This is a block-level container.</div>
<h1>This is a heading</h1>
```

2. Inline Elements

- Do not start on a new line
- Only take up as much width as needed
- Can only contain text or other inline elements (not block-level)

- Examples:
  - – Generic inline container
  - <a> – Link
  - , – Bold and italic emphasis
  - <img> – Image
  -
 – Line break

  - Example Code:

```
<p>This is <strong>bold</strong> and <em>italic</em> text inside a paragraph.</p>
<a href="#">Click here</a> to visit our page.
```

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

Ans:

Role of Semantic HTML:

Semantic HTML refers to using HTML tags that convey the meaning of the content inside them, rather than just how it looks. These elements clearly describe their purpose, which helps both developers and browsers (and assistive technologies) understand the structure and content of a web page.

Why Semantic HTML Is Important:

1. Accessibility:

- Semantic tags help screen readers and other assistive tools interpret content correctly.

- They provide clear roles for elements (e.g., headings, navigation, articles), making the web more usable for people with disabilities.

2. SEO (Search Engine Optimization):

- Search engines use the structure and meaning of semantic elements to better index and rank your content.
- Helps search engines determine which parts of your content are most important.

3. Readability and Maintainability:

- Makes your code easier to read and understand for other developers (or future you).
- Encourages clean, organized structure.

➢ **Example:**

```
<header>
  <h1>My Blog</h1>
  <nav>
    <a href="#">Home</a>
    <a href="#">Articles</a>
    <a href="#">Contact</a>
  </nav>
</header>

<main>
  <article>
    <h2>What is Semantic HTML?</h2>
    <p>Semantic HTML improves structure and accessibility...</p>
  </article>

  <aside>
    <h3>Related Posts</h3>
    <ul>
      <li><a href="#">Intro to HTML</a></li>
    </ul>
  </aside>
</main>

<footer>
  <p>© 2025 My Blog</p>
</footer>
```

• HTML Forms

1. What are HTML forms used for? Describe the purpose of the input, textarea, select, and button elements.

Ans:

What Are HTML Forms Used For:

HTML forms are used to collect user input and send that data to a server for processing. They are a fundamental part of interactive websites and allow users to communicate with a website or web application.

- Signing up or logging in
- Searching
- Submitting feedback or contact information
- Making purchases

Purpose of HTML Form Elements:

1. <input>

- Used to collect single-line input from the user.
- Very versatile: supports many types via the type attribute, such as:
  - text, email, password, number, checkbox, radio, date, etc.
  -

Example:

<input type="text" name="username" placeholder="Enter your name">

2. <textarea>

- Used to collect multi-line text input.
- Ideal for messages, comments, or descriptions.

Example:

<textarea name="message" rows="4" cols="30" placeholder="Write your message here..."></textarea>

3. <select>

- Creates a drop-down menu.

- Allows users to choose from a list of options.
- Can also be set to allow multiple selections with the multiple attribute.

Example:

```
<select name="country">
  <option value="us">United States</option>
  <option value="ca">Canada</option>
</select>
```

4. <button>

- Used to perform actions like submitting a form or triggering JavaScript.
- Can have different types: submit, reset, or just button.

Example:

```
<button type="submit">Submit</button>
<button type="reset">Clear</button>
```

2. Explain the difference between the GET and POST methods in form submission. When should each be used?

Ans:

When an HTML form is submitted, it sends data to a server using either the **GET** or **POST** method. These methods define how the data is sent and used.

➢ **GET Method**

- Appends form data to the URL as query parameters.
- Example: example.com/form?name=John&age=25
- Data is visible in the browser's address bar.
- Has size limits (since URLs can't be too long).
- Data can be bookmarked and cached.
- Less secure — not suitable for sensitive information.

➢ **Use GET when:**

- The form does not contain sensitive data (e.g., search bars).
- You want the URL to be shareable or bookmarkable.

➢ **POST Method**

- Sends form data in the request body, not in the URL.
- Data is not visible in the URL.
- No size limit for data (within practical server limits).
- More secure for sensitive information like passwords.
- Cannot be bookmarked or cached.

➢ **Use POST when:**

- Submitting sensitive or personal data (e.g., login forms).
- Performing actions that modify data on the server (e.g., creating an account, submitting a comment).

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

Ans:

Purpose of the <label> Element in a Form:

The <label> element is used to **describe or identify a form input**, such as a text field, checkbox, or dropdown. It provides a **text label** that tells the user what the input is for.

❖ **Key Purposes of <label>:**

1. **Describes Input Fields:**
   - Tells users what to enter in an input field.
   - Example: Name: [_____]
2. **Improves Accessibility:**
   - Helps **screen readers** identify form controls for users with visual impairments.
   - When a <label> is correctly linked to an input, screen readers will read the label when the input is focused.
3. **Increases Click Area (for checkboxes and radios):**
   - Clicking on the label will also activate the associated input, improving usability.

• HTML Tables

1. Explain the structure of an HTML table and the purpose of each of the following elements: <table>, <tr>, <th>, <td>, and <thead>.

Ans:

Structure of an HTML Table:

An HTML table is used to organize data in rows and columns. It is made up of several elements that each serve a specific purpose.

Basic Structure Example:

```
<table>
  <thead>
    <tr>
      <th>Name</th>
      <th>Age</th>
    </tr>
  </thead>
  <tr>
    <td>Alice</td>
    <td>25</td>
  </tr>
  <tr>
    <td>Bob</td>
    <td>30</td>
  </tr>
</table>
```

Explanation of Each Element:

- <table>:
  The container element for the entire table. It wraps all table rows, headers, and data.
- <tr>:
  Stands for table row. Groups a row of cells together (either headers or data).
- <th>:
  Stands for header. Defines a heading cell. Text is bold and centered by default.
- <td>:
  Stands for table data. Represents a regular cell that contains data.
- <thead>: table
  Groups one or more <tr> elements that make up the table's header section. Helps with structure and accessibility.

2. What is the difference between colspan and rowspan in tables? Provide examples.

Ans:

❖ **Colspan:**

- Definition: Allows a cell to extend horizontally across multiple columns.
- Use Case: When you want a single cell to take the space of several columns.

➢ Example of colspan:

```
<table border="1">
  <tr>
    <th colspan="2">Name</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>First</td>
    <td>Last</td>
    <td>25</td>
  </tr>
</table>
```

➢ Explanation:
The header cell "Name" spans two columns (First and Last).

❖ **Rowspan:**

- Definition: Allows a cell to extend vertically across multiple rows.
- Use Case: When a cell should cover multiple rows in a column.

➢ Example of rowspan:

```
<table border="1">
  <tr>
    <th rowspan="2">Name</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>25</td>
  </tr>
</table>
```

➢ Explanation:
The "Name" cell spans two rows, so it stays aligned while other data is listed in adjacent rows.

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

Ans:

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

1. Not Semantic:
   Tables are meant for tabular data, not page structure. Using them for layout confuses screen readers and other assistive technologies.
2. Poor Accessibility:
   Screen readers rely on meaningful HTML structure. Tables used for layout create a confusing experience for users with disabilities.
3. Difficult to Maintain:
   Table-based layouts are hard to edit, especially if the structure is complex or nested.
4. Not Responsive:
   Tables don't adapt well to different screen sizes, especially mobile. They're rigid and require extra work to make responsive.
5. Outdated Practice:
   This was common in the early days of web design, but CSS offers much better solutions today.

❖ **Better Alternative: CSS (Cascading Style Sheets)**

➢ Modern CSS allows flexible and responsive layouts using:

- Flexbox (display: flex)
- Grid (display: grid)
- Media Queries (for responsive design)
- Positioning and Spacing (like margin, padding, etc.)

Example using Flexbox:

```
<div style="display: flex; gap: 20px;">
  <div style="flex: 1;">Sidebar</div>
  <div style="flex: 3;">Main Content</div>
</div>
```

Example using Grid:

```
<div style="display: grid; grid-template-columns: 1fr 3fr; gap: 20px;">
  <div>Sidebar</div>
  <div>Main Content</div>
</div>
```