Module 15: HTML in Full Stack

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

Answer: **HTML (Hypertext Markup Language)** is the standard markup language used to create and design documents that are displayed in a web browser. It provides the **basic structure** of a webpage by using tags to define elements like headings, paragraphs, links, images, lists, and more.

Purpose:

- Structure the Content
- Display Information in Web Browsers
- Embed Multimedia
- Create Links and Navigation
- Forms and User Input
- Semantic Meaning

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

Answer:

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

</body>

</html>

Basic Structure of HTML:

- <!DOCTYPE html> – Declares HTML5.
- <html> – Root of the document.
- <head> – Metadata and title.
- <title> – Title in browser tab.
- <body> – All visible page content.

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

Answer:

Block-Level Elements:

Block-level elements start on a new line and take up the full width available (by default).

Example:

This is paragraph

<h2>Html Tag</h2>

Inline Elements:

Inline elements **do not start on a new line**, and only take up as much width as necessary.

Example:

<u>Underline</u>

Home

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. These elements help both browsers and developers understand the purpose of different parts of a web page.

```
<!DOCTYPE html>

<html>

<head>

        <title>Semantic HTML Example</title>

</head>

<body>

<header>

         <h1>My Blog</h1>

</header>

<nav>

        <a href="#">Home</a>

        <a href="#">Articles</a>

        <a href="#">Contact</a>

</nav>

<main>

        <article>

                <h2>Understanding Semantic HTML</h2>

                 <p>Semantic HTML makes websites more accessible and SEO-friendly.</p>

        </article>
```

```
</main>

<footer>

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

</footer>

</body>

</html>
```

Question 5) What are HTML forms used for? Describe the purpose of the input, text area, select, and button elements.

Answer: **HTML forms** are used to **collect user input** and send it to a **server** for processing. They are commonly used in login pages, contact forms, search boxes, surveys, and registration pages.

<input>:

- Used to **collect single-line input** such as text, numbers, passwords, emails, etc.
- Has various type attributes: text, password, email, checkbox, radio, submit, etc.

<text area>:

- Used to collect **multi-line text input**, such as comments or messages.

<select>:

- Creates a **drop-down list** with multiple <option> items.
- Used when the user needs to **choose from predefined options**.

<button>:

- Represents a clickable **button**.
- Can be used to **submit**, **reset**, or trigger JavaScript functions.

Question 6) Explain the difference between the GET and POST methods in form submission. When should each be used?

Answer:

Feature	GET	POST
Visibility	Data is **visible** in browser's address bar	Data is **hidden** from the URL
Length Limit	Limited in size (usually ~2048 characters)	No size limit (can send large data)
Caching	Can be cached and bookmarked	Cannot be cached or bookmarked
Security	Less secure (data exposed in URL)	More secure (data hidden from URL)

Question 7) What is the purpose of the label element in a form, and how does it improve accessibility?

Answer: The <label> element in HTML is used to **define a caption or description** for a form control like an <input>, <select>, <text area>, etc.

Main Purposes of <label>:

1. **Describes the Input Field**
   It tells users what information should be entered into a specific form field.
2. **Improves Clickability**
   When associated correctly, clicking the <label> also focuses or activates the corresponding form control (like a checkbox or text box).
3. **Enhances Accessibility**
   Screen readers use the <label> to inform visually impaired users what each field is for, making the form more **navigable and understandable**.

   Example:

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

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

Question 8) Explain the structure of an HTML table and the purpose of each of the following elements: <table>, <tr>, <td> and <thead>.

Answer:

- **<table>**: The container for the entire table.
- **<tr>**: Defines a row within the table.
- **<td>**: Defines a single cell within a row.

- **<thead>**: Groups the header content.

```
<table border="1">
  <thead>
    <tr>
      <th>Fruit</th>
      <th>Price</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Apple</td>
      <td>$2</td>
    </tr>
    <tr>
      <td>Banana</td>
      <td>$1</td>
    </tr>
  </tbody>
</table>
```

Question 9) What is the difference between colspan and rowspan in tables? Provide examples.

Answer: Colspan: Used for to merge the column in a row.

Example:

```
<table border="1" cellpadding="10">
    <tr>
        <th colspan="2">Header Spanning 2 Columns</th>
    </tr>
    <tr>
        <td>Column 1</td>
        <td>Column 2</td>
    </tr>
</table>
```

Rowspan: Used to merge the row in the table.

Example:

```
<table border="1" cellpadding="10">
    <tr>
        <td rowspan="2">Row Spanning 2 Rows</td>
        <td>Row 1, Column 2</td>
    </tr>
    <tr>
        <td>Row 2, Column 2</td>
    </tr>
</table>
```

Question 10) Why should tables be used sparingly for layout purposes? What is a better alternative?

Answer: Tables should be used sparingly for layout because they were designed for displaying tabular data, not structuring web pages. Using tables for layout creates accessibility issues, makes code harder to maintain, and is not responsive for different screen sizes. It also negatively

impacts SEO and page loading speed. A better alternative is using semantic HTML elements combined with CSS techniques like Flexbox or Grid, which provide cleaner, more responsive, and accessible layouts.

Submitted By: Ansari Amanhusain Kalimuddin

Course: Full Stack Developer

Batch: 07th July 2025.