Frontend Assignment

**Module 2:**

**HTML Basics Theory**:

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

**HTML** stands for **Hyper Text Markup Language**. It is the standard language used to create

and structure content on the web.

**Purpose of HTML in Web Development:**

* **Structure Content:** HTML provides the basic structure of web pages by using elements like headings, paragraphs, images, lists, and links.
* **Content Organization:** It organizes content so that web browsers can interpret and display it correctly.
* **Linking Documents:** HTML allows linking to other web pages or resources through hyperlinks, forming the foundation of web navigation.
* **Embedding Media:** It supports embedding images, videos, audio, and other multimedia elements.
* **Semantic Meaning:** HTML5 introduces semantic elements (like <article>, <section>, <nav>, etc.) to help describe the purpose of different parts of the content, improving accessibility and SEO.

In summary, **HTML is essential for defining the content and layout of a website**, serving as the foundation upon which CSS and JavaScript enhance appearance and interactivity.

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

|  |
| --- |
| <html> |
| <head> |
| <title>My First Web Page</title> |
| </head> |
| <body> |
| <h1>Welcome</h1> |
| <p>This is a simple HTML page</p> |
| </body> |
| </html> |

**Mandatory Tags and Their Purposes:**

* **<html>**
  + The root element of the HTML document.
  + Encloses all content on the page.
* **<head>**
  + Contains metadata about the document (not displayed on the page).
  + Includes tags like <title>, <meta>, <link>, and <style>.
* **<title>**
  + Sets the title of the web page (displayed on the browser tab).
  + Required inside the <head>.
* **<body>**
  + Contains the visible content of the web page (text, images, links, etc.).
  + Everything inside the <body> is rendered on the screen.

These tags form the **foundation of every HTML page**. Without them, a browser would struggle to correctly interpret or render the content.

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

In HTML, elements are categorized as either **block-level** or **inline** based on how they behave in the layout of a web page.

|  |  |
| --- | --- |
| **Block-Level Elements** | **Inline Elements** |
| * **Start on a new line** and take up the **full width** available. | * **Do not start on a new line**, and only take up as much width as necessary. |
| * Used to define the **structure and layout** of a document. | * Typically used to **style or modify small parts** of text/content. |
| * Can contain **other block-level and inline elements**. | * Cannot contain block-level elements (only text or other inline elements). |
| **Examples:**   * <div> – generic container * <p> – paragraph * <h1> to <h6> – headings * <ul>, <ol>, <li> – lists * <section>, <article>, <header>, <footer> | **Examples:**   * <span> – generic inline container * <a> – hyperlink * <strong> – bold text * <em> – italic text * <img> – image |

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

**Semantic HTML** refers to using HTML elements that **clearly describe their meaning or purpose** in the structure of a web page. These elements not only define how the content looks but also **convey its meaning** to browsers, developers, search engines, and assistive technologies (like screen readers).

**Why Semantic HTML Is Important:**

* **Accessibility**
* **Screen readers** and other assistive technologies rely on semantic elements to navigate content correctly.
* Semantic tags like <nav>, <header>, <article> help users with disabilities understand page structure and skip to relevant content.
* **SEO (Search Engine Optimization)**
* Search engines use semantic tags to better understand page structure and content importance.
* Improves **search indexing** and ranking, as content within semantic elements is more meaningful to crawlers.
* **Readability and Maintainability**
* Makes the code easier to read and maintain for developers.
* Helps teams collaborate more effectively by making the structure self-explanatory.

Using **semantic HTML** improves both the **usability** and **visibility** of your website, making it a best practice in modern web development.

**HTML Forms**

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

**HTML forms are used to collect user input and send it to a server for processing. They are essential for interactive features like login pages, search boxes, contact forms, surveys, and more.**

**A form is defined using the <form> element, which can contain various input controls.**

**Key Form Elements and Their Purposes:**

* **<input>**
* A versatile element used to create various types of user input fields.
* Controlled by the type attribute (e.g., text, email, password, checkbox, etc.).
* **<textarea>**
* Allows users to enter multi-line text (e.g., comments, messages).
* Unlike <input>, it doesn’t use the type attribute.
* **<select>**
* Creates a drop-down list of options.
* Works with nested <option> elements to define choices.
* **<button>**
* Used to submit the form, reset it, or trigger custom JavaScript.
* Can contain text or HTML (like icons).

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

**In HTML forms, the method attribute defines how data is sent to the server. The two most common methods are GET and POST.**

|  |  |
| --- | --- |
| **GET Method** | **POST Method** |
| **Characteristics:**   * **Data is appended to the URL as query parameters.** * **Can be bookmarked or shared.** * **Has length limitations (varies by browser/server).** * **Less secure — data is visible in the URL.** | **Characteristics:**   * **Data is sent in the request body, not visible in the URL.** * **No size limitations for form data (subject to server limits).** * **More secure for sending sensitive information.** * **Cannot be bookmarked with data.** |
| **When to Use:**   * **For non-sensitive data.** * **When retrieving or filtering data (e.g., search forms).** * **When bookmarking or link sharing is useful.** | **When to Use:**   * **For sensitive or private data (e.g., login forms, contact forms).** * **When performing data modification (e.g., create, update, delete).** * **For larger data submissions (e.g., file uploads, long comments).** |

**Use GET for retrieving data and POST for sending or changing data, especially when security or data size matters.**

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

**The <label> element provides a textual description for a form control, such as an <input>, <textarea>, or <select>. Its main role is to tell users what each form field is for.**

**How <label> Improves Accessibility:**

* **Screen Reader Support:**
  + **When a screen reader user focuses on a form field, the label text is read aloud, making the form understandable to users with visual impairments.**
* **Clickable Area:**
  + **Clicking on the label also activates the associated form control (e.g., a checkbox), making the form easier to use for people with motor difficulties.**
* **Better User Experience:**
  + **Labels make forms more intuitive and prevent users from guessing what to input in each field.**

**In summary, the <label> element is essential for creating accessible, user-friendly forms, especially for users relying on assistive technologies.**

HTML Tables

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

An HTML table is used to **organize data into rows and columns**. It consists of several nested elements that define how the data is structured and displayed.

**📘 Key HTML Table Elements and Their Purposes:**

* **<table>**
* The **container** element for the entire table.
* All table-related tags go inside this element.
* **<tr> (Table Row)**
* Defines a **row** in the table.
* Each row can contain one or more cells (<th> or <td>).
* **<th> (Table Header Cell)**
* Defines a **header cell** in the table.
* Text is **bold** and **centered by default**.
* Usually used inside the first row to **label columns**.
* Can also be used for row headers.
* **<td> (Table Data Cell)**
* Defines a **standard data cell**.
* Used to display the actual content in the table.
* **<thead> (Table Head Section)**
* Groups **header rows** of a table.
* Helps **semantically** separate the table’s header from its body.
* Often used with <tbody> and <tfoot> for better structure and accessibility.

|  |  |  |
| --- | --- | --- |
|  |  |  |

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

Both colspan and rowspan are **attributes** used with <td> or <th> elements in HTML tables to **merge cells** either **horizontally** or **vertically**.

* **colspan (Column Span)**
* Merges a **cell across multiple columns**.
* Useful for creating a header or data cell that spans more than one column.
* **rowspan (Row Span)**
* Merges a **cell across multiple rows**.
* Useful when a cell (e.g., a category) applies to multiple rows of data.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Direction** | **Purpose** |
| colspan | Horizontal  (columns) | |  |  | | --- | --- | |  | Merge multiple columns into one | |
| |  |  | | --- | --- |   Rowspan | |  |  | | --- | --- | |  | Vertical (rows) | | |  |  | | --- | --- | |  | Merge multiple rows into one | |

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

In the early days of web development, HTML tables were commonly used to create page layouts (e.g., multi-column designs). However, **this approach is outdated and discouraged** for several important reasons:

* **Problems with Using Tables for Layout:**
* **Poor Accessibility**
  + Screen readers interpret tables as data tables, not as layout tools, which can confuse users relying on assistive technologies.
* **Not Responsive**
  + Tables are rigid and don't adapt well to different screen sizes, making mobile-friendly design difficult.
* **Hard to Maintain**
  + Table-based layouts are complex and fragile. Changing one part often affects the entire layout.
* **Mixes Content and Presentation**
  + Tables are meant for structuring data, not presentation. Using them for layout violates semantic HTML principles.
* **Slower Page Rendering**
  + Browsers must wait to parse the entire table before rendering, which can negatively impact performance.
* **Better Alternative: Use CSS for Layout**

Modern web design uses **CSS (Cascading Style Sheets)** for all layout and styling. Key layout techniques include:

| **Technique** | **Description** |
| --- | --- |
| **Flexbox** | 1D layout system — great for rows or columns |
| **Grid** | 2D layout system — ideal for complex layouts |
| **Media Queries** | Enables responsive design for all devices |
|  |  |

* **When to Use Tables:** Use tables **only for displaying tabular data**, such as financial reports, schedules, or data grids — **not** for layout.