1. HTML Basics

Theory Assignment

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

Ans: HTML (HyperText Markup Language) is the standard language used to create and structure content on the web. It consists of a series of elements or tags that define the structure of web pages, such as headings, paragraphs, links, images, and other content types.

The purpose of HTML in web development is to:

1. **Structure content**: It defines the basic layout of a web page, organizing text, images, links, and other media in a logical way.
2. **Make content accessible**: HTML helps browsers render the web page correctly, making it accessible to users and search engines alike.

3. **Interactivity foundation**: While HTML itself doesn’t add interactivity, it forms the backbone for other languages like CSS (Cascading Style Sheets) and JavaScript to enhance design and functionality.

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

Ans: **Basic Structure of an HTML Document:**

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Page Title</title>
  </head>
  <body>
    <!-- Content of the page goes here -->
  </body>
</html>
```

Mandatory Tags and Their Purposes:

1. **<!DOCTYPE html>**:
   - **Purpose**: Declares the document type and version of HTML (in this case, HTML5). It helps browsers render the page correctly according to HTML5 standards.
2. **<html>**:
   - **Purpose**: This is the root element of the document that wraps all the HTML content.

   - It may include an attribute like lang="en" to specify the language of the document (in this case, English).

3. **<head>**:
   - **Purpose**: Contains metadata (data about data) for the document, such as the character encoding, title of the page, links to stylesheets, and scripts.
   - **Mandatory tags inside <head>**:
     - <meta charset="UTF-8">: Specifies the character encoding for the page (UTF-8 is a widely used character encoding).
     - <meta name="viewport" content="width=device-width, initial-scale=1.0">: Ensures the page is responsive on different screen sizes, particularly for mobile devices.
     - <title>: Defines the title of the webpage, shown in the browser tab.
4. **<body>**:

    - **Purpose**: Contains the actual content of the webpage that users interact with, such as text, images, links, and other elements.
    -

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

Ans: Block-Level Elements:

- **Definition**: Block-level elements take up the full width available (by default) and start on a new line, effectively creating a "block" of content. These elements can contain other block-level elements as well as inline elements.
- **Behavior**: They cause a line break before and after the element.
- **Examples**:
    - <div>: A container used to group other elements.
    - <p>: Represents a paragraph of text.

  - <h1>, <h2>, ..., <h6>: Header elements, which define headings of different levels.
  - <ul>, <ol>, <li>: Lists (unordered and ordered).

Inline Elements:

- **Definition**: Inline elements only take up as much width as necessary and do not cause a line break. They flow within the content and are usually used for styling or adding functionality within block-level elements.
- **Behavior**: They do not force a new line before or after the element, and they do not expand to take up the full width of the container.
- **Examples**:
  - <span>: A generic container used to style or group inline content.
  - <a>: Represents a hyperlink.
  - <strong>: Represents important text (typically bold).
  - <em>: Represents emphasized text (typically italicized).

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

Ans: **Role of Semantic HTML:**

Semantic HTML refers to the use of HTML elements that clearly describe their meaning both in terms of content and structure. These elements convey more than just presentation, helping browsers, developers, and search engines understand the purpose and context of the content on a webpage.

Why is Semantic HTML Important for Accessibility and SEO?

1. **Accessibility**:
   - **Screen Readers**: Semantic elements help assistive technologies (like screen readers) interpret and present content in a meaningful way to users with disabilities. For example, a screen reader can easily recognize a <header> or <nav> element and provide appropriate context to the user.

- **Clear Structure**: Proper use of semantic tags ensures that users with disabilities can navigate the content logically and easily.

2. **SEO (Search Engine Optimization)**:
    - **Content Relevance**: Search engines use semantic HTML to better understand the context of the page's content. Tags like <article>, <section>, and <header> help search engines assess the structure of the page and its relevance to search queries.
    - **Improved Rankings**: Pages that use semantic elements are often ranked higher by search engines because they are easier to index and provide a clearer, more organized structure.

Examples of Semantic Elements:

1. **<header>**: Represents introductory content or navigational links.
    - **Example**: <header><h1>Welcome to My Website</h1></header>
2. **<nav>**: Defines a block of navigation links.

   - **Example**: <nav><ul><li><a href="#">Home</a></li><li><a href="#">About</a></li></ul></nav>

3. **<article>**: Represents a self-contained piece of content, like a blog post or news article.
   - **Example**: <article><h2>Article Title</h2><p>Content of the article.</p></article>
4. **<section>**: Defines a section of content, typically with its own heading.
   - **Example**: <section><h2>About Us</h2><p>Some information about the company.</p></section>
5. **<footer>**: Represents the footer of a document or section, typically containing copyright information, contact details, etc.
   - **Example**: <footer><p>© 2025 My Website</p></footer>
6. **<aside>**: Represents content that is tangentially related to the content around it, such as sidebars or related links.

- **Example**: <aside><h3>Related Links</h3><ul><li><a href="#">Link 1</a></li></ul></aside>

7. **<main>**: Defines the main content of the document, excluding headers, footers, and sidebars.
    - **Example**: <main><h1>Main Article</h1><p>Content here...</p></main>

3.HTML Forms

Theory Assignment .

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

Ans: HTML Forms:

HTML forms are used to collect user input and submit it to a server for processing. They allow users to interact with a web application by entering data, which can then be used for various purposes such as account creation, login, feedback submission, or making transactions.

Purpose of Key Form Elements:

1. **<input>:**

- **Purpose: The <input> element is the most commonly used element in forms. It allows users to enter data in various types of fields, such as text, numbers, dates, checkboxes, radio buttons, and more.**
- **Example:**

html

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

  - **Common type attributes: text, password, checkbox, radio, submit, email, etc.**

2. **<textarea>:**
    - **Purpose: The <textarea> element is used for multi-line text input, allowing users to enter longer pieces of text (like comments, descriptions, or messages).**
    - **Example:**

html

```
<textarea name="message" rows="4" cols="50" placeholder="Enter your message"></textarea>
```

        - Attributes like rows and cols define the size of the textarea.

3. <select>:
    - Purpose: The <select> element creates a drop-down menu that allows users to choose one or more options from a list.
    - Example:

html

```
<select name="country">
  <option value="USA">United States</option>
  <option value="CAN">Canada</option>
  <option value="UK">United Kingdom</option>
</select>
```

4. **<button>:**
   - **Purpose: The <button> element is used to create clickable buttons within forms. It can be used for submitting the form, resetting the form, or triggering JavaScript functions.**
   - **Example:**

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

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

Ans: GET vs POST Method:

- **GET:**
  - **Data: Sent in the URL.**
  - **Use: For retrieving data (e.g., search).**
  - **Visibility: Data is visible in the URL.**

    - **Limitations: Limited data size, less secure.**
    - **Example: <form method="GET" action="/search">**
- **POST:**
    - **Data: Sent in the request body.**
    - **Use: For submitting sensitive data or large amounts (e.g., login).**
    - **Visibility: Data is not visible in the URL.**
    - **Limitations: Cannot be bookmarked, more secure.**
    - **Example: <form method="POST" action="/login">**

When to Use:

- **GET: For non-sensitive, small data.**
- **POST: For sensitive data or large submissions.**

Question 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 in a form is used to define a label for an <input>, <textarea>, or other form control. It associates the text description with the form element, making the form easier to understand for users.

How It Improves Accessibility:

1. **Screen Reader Support: When a <label> is associated with a form control, screen readers can read out the label text to visually impaired users, helping them understand what information is required in each field.**
2. **Clickable Labels: By associating a label with a form control (using the for attribute, which matches the input's id), clicking on the label will focus on the corresponding input field. This is**

especially useful for users with motor impairments, as it increases the clickable area.

3. Clearer Context: Labels provide clear instructions about what each form field is for, improving the overall usability of the form.

4.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>and <thead>.

Ans: Structure of an HTML Table:

An HTML table is used to display tabular data (rows and columns) on a webpage. It consists of several elements that help organize and structure the table. Here's a breakdown of the most common elements:

```
<table>
  <thead>
    <tr>
      <th>Header 1</th>
      <th>Header 2</th>
      <th>Header 3</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Data 1</td>
      <td>Data 2</td>
      <td>Data 3</td>
    </tr>
    <tr>
      <td>Data 4</td>
      <td>Data 5</td>
      <td>Data 6</td>
    </tr>
  </tbody>
</table>
```

Purpose of Each Element:

1. **<table>:**
   - **Purpose: This is the container element that defines the table. All other table elements (such as rows and cells) are nested inside the <table> element.**
   - **Example: <table>...</table>**
2. **<tr> (Table Row):**
   - **Purpose: The <tr> element defines a row in the table. It groups together <th> (header cells) or <td> (data cells) elements for each row.**
   - **Example: <tr>...</tr>**
3. **<th> (Table Header):**
   - **Purpose: The <th> element defines a header cell in the table, typically displayed in bold and centered by default. It is used to label the columns (or rows) of data.**
   - **Example: <th>Header 1</th>**
4. **<td> (Table Data):**

- **Purpose: The <td> element defines a standard data cell in the table. It holds the actual data or content for each cell within a row.**
- **Example: <td>Data 1</td>**

5. **<thead> (Table Header Group):**
    - **Purpose: The <thead> element is used to group the header content in a table. It typically contains the <tr> with <th> elements, and it is useful for applying styling or handling table data dynamically.**
    - **Example: <thead><tr><th>Header</th></tr></thead>**

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

Ans: Difference Between colspan and rowspan in Tables:

- **colspan:**

    - Purpose: The colspan attribute is used to specify how many columns a table cell should span (or stretch across) horizontally.
    - Use Case: It allows a single cell to span multiple columns, making the table layout more flexible and organized.
- rowspan:
    - Purpose: The rowspan attribute is used to specify how many rows a table cell should span vertically.
    - Use Case: It allows a single cell to extend over multiple rows.

Examples:

1. Using colspan:
    - This example shows a cell that spans two columns.

html

Copy

```
<table border="1">
```

```
  <tr>
    <th colspan="2">Header Spanning 2 Columns</th>
  </tr>
  <tr>
    <td>Data 1</td>
    <td>Data 2</td>
  </tr>
</table>
```

In this example, the header cell <th> spans across two columns because of the colspan="2" attribute.

2. **Using rowspan:**
    - **This example shows a cell that spans two rows.**

html

```
<table border="1">
  <tr>
    <td rowspan="2">Rowspan 2 Rows</td>
    <td>Data 1</td>
  </tr>
  <tr>
    <td>Data 2</td>
  </tr>
</table>
```

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

Ans: Why Tables Should Be Used Sparingly for Layout Purposes:

1. **Accessibility Issues:**
    - **Tables were originally designed for displaying tabular data, not for layout. Using them for layout purposes can confuse screen readers and assistive technologies, making the content harder to understand for users with disabilities.**
2. **Responsiveness Problems:**
    - **Tables can be difficult to make responsive (i.e., adapt to different screen sizes) when used for layout. Complex table structures may break or appear misaligned on mobile devices, making the page harder to navigate.**
3. **SEO Impact:**
    - **Search engines may not interpret content in a table used for layout as effectively as content within divs and other semantic elements. This can negatively impact SEO because search engines prefer well-structured content.**
4. **Maintainability:**

   - **Tables for layout can make the code harder to read and maintain. A complex table structure with nested elements can be cumbersome to edit and debug compared to using simpler, more modern layout techniques.**

Better Alternatives for Layout:

1. **CSS Flexbox:**
   - **Flexbox is a powerful CSS layout module that allows for responsive, flexible layouts. It's ideal for creating layouts where elements can be arranged in rows, columns, or grids with great control over their alignment, spacing, and positioning.**
   - **Example:**

css

```
.container {
  display: flex;
  justify-content: space-between;
}
.item {
  flex: 1;
}
```

2. CSS Grid:
    - CSS Grid is another powerful layout system that allows for two-dimensional layout (both rows and columns). It's perfect for complex layouts with items that need to be positioned both horizontally and vertically.
    - Example:

css

```
.container {
  display: grid;
  grid-template-columns: repeat(3, 1fr);
}
.item {
  grid-column: span 1;
}
```

3. **CSS Floats :**
   - **While floats were historically used for layout (e.g., for creating multi-column layouts), they are generally less reliable and harder to control compared to Flexbox and Grid. Floats are still useful for floating elements like images, but they aren’t recommended for full layouts anymore.**