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

**Ans** : **HTML (Hyper Text Markup Language)** is the standard markup language used to create and structure content on the web. It defines the structure of web pages by using a system of **tags** and **elements**.

**Purpose of HTML in Web Development:**

1. **Structure**:  
   HTML provides the basic structure of a webpage, such as headings, paragraphs, lists, links, images, tables, and more. It organizes content into a readable and logical format for browsers to render.
2. **Content Display** :

It tells the browser what content to display and how different elements relate to one another (e.g., defining a navigation bar, main content area, or footer).

1. **Hyperlinks:**HTML enables linking between pages via hyperlinks (<a> tags), forming the core of web navigation.
2. **Integration with Other technologies :**

HTML works alongside:

* + **CSS (Cascading Style Sheets)** to style the content (colors, layout, fonts, etc.).
  + **JavaScript** to add interactivity and dynamic behavior.

1. **Accessibility and Semantics:**Using semantic HTML elements (like <header>, <article>, <section>, <footer>) helps make content more accessible to screen readers and improves SEO (Search Engine Optimization).

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

**Ans :** An HTML document follows a standard structure composed of nested elements (tags) that define the content and layout of a web page. Here's an example of a basic HTML document and an explanation of the **mandatory tags** and their purposes:

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>My First Web Page</title>  </head>  <body>  <h1>Welcome to My Website</h1>  <p>This is a paragraph of text.</p>  </body>  </html> |

**Mandatory Tags and Their Purposes**

|  |  |
| --- | --- |
| **Tag** | **Purpose** |
| **<!DOCUMENT html>** | Declares the **HTML version** being used (HTML5 in this case). Helps the browser render the page correctly. |
| **<html>** | The **root element** of the HTML document. All other elements are contained within this tag. |
| **<head>** | Contains **metadata** about the document (not displayed on the page), such as the title, character encoding, links to CSS, and scripts. |
| **<title>** | Sets the **title of the web page** (displayed in the browser tab or window title). Must be inside the <head> tag. |
| **<body>** | Contains all the **visible content** of the web page, such as text, images, links, etc. |

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

### Ans : 3 Difference Between Block-Level and Inline Elements in HTML

HTML elements are broadly categorized into **block-level** and **inline** elements based on how they behave in the page layout.

**1. Block-Level Elements**

**Definition**:  
Block-level elements start on a **new line** and take up the **full width** available (by default), stretching from left to right across their container.

**Purpose**:  
They are used to **structure** the layout of a page (e.g., sections, containers, paragraphs).

**Examples**:

* <div> – generic container
* <p> – paragraph
* <h1> to <h6> – headings
* <ul>, <ol>, <li> – lists
* <section>, <article>, <header>, <footer> – semantic layout elements

**Example Usage**:

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

<div>This is a block-level container.</div>

**2. Inline Elements**

**Definition**:  
Inline elements do **not start on a new line**. They only take up as much **width** as necessary and flow **within** block-level elements.

**Purpose**:  
Used to **format** parts of content inside block elements without disrupting the flow.

**Examples**:

* <span> – generic inline container
* <a> – hyperlink
* <strong> – bold text
* <em> – italic text
* <img> – image
* <input> – form input field

**Example Usage**:

<p>This is a <strong>bold</strong> word inside a paragraph.</p>

<a href="#">Click here</a> to visit our site.

**Question 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 the use of HTML elements that **convey meaning** about the type of content they contain. These elements not only structure content visually but also provide **context** to web browsers, screen readers, and search engines.

### Why Semantic HTML Is Important .?

### Accessibility

### **Screen readers** and other assistive technologies rely on semantic tags to understand the content and its role.

### For example, using <nav> helps a screen reader identify navigation links, improving the user experience for visually impaired users.

### SEO (Search Engine Optimization)

* Search engines use semantic tags to **better understand page content**, which can improve search rankings.
* Proper use of headings (<h1>, <h2>, etc.) and sections (<article>, <section>) makes it easier for crawlers to **index and rank** pages appropriately.

### Maintainability and Clarity

* Semantic elements make the code more **readable** and **self-explanatory** for developers.
* This improves collaboration and reduces confusion when updating or maintaining code.

### Examples of Semantic HTML Elements

| **Element** | **Purpose** |
| --- | --- |
| <header> | Represents the top section of a page or section (often contains logo, title, nav). |
| <nav> | Contains the navigation links. |
| <main> | Denotes the main content of the page, unique from header/footer. |
| <section> | Groups related content or themes within a page. |
| <article> | Represents self-contained content (e.g., blog post, news article). |
| <aside> | Holds content tangentially related to the main content (e.g., sidebars). |
| <footer> | Defines the bottom section of a page or section (e.g., copyright, links). |
| <figure> and <figcaption> | Used for images or illustrations with captions. |

### Example in Context

|  |
| --- |
| <main><article><header><h1>How to Bake a Cake</h1><p>By Jane Doe</p></header><section><h2>Ingredients</h2><ul><li>Flour</li><li>Eggs</li><li>Sugar</li></ul></section><section><h2>Instructions</h2><p>Mix ingredients and bake at 350°F for 30 minutes.</p></section><footer><p>Published on May 17, 2025</p></footer></article></main> |

**HTML Forms**

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

**HTML forms** are essential for collecting user data on websites. They enable interactions like user registrations, feedback submissions, and search functionalities. Here's a breakdown of key form elements:

* **<input>:** This versatile element handles various data types, including text, passwords, emails, and more. It can be configured with different type attributes to suit specific needs.
* **<textarea>:** Allows users to enter multi-line text, ideal for comments, messages, or descriptions.
* **<select>:** Creates a dropdown menu from which users can choose one or more options. It often includes <option> tags to define the available choices.
* **<button>:** Defines a clickable button, commonly used to submit forms or trigger specific actions.

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

* **GET**: Appends form data to the URL, making it visible in the browser's address bar. It's suitable for retrieving data without causing side effects, such as search queries.
* **POST**: Sends form data within the request body, keeping it hidden from the URL. This method is preferred for submitting sensitive information, like passwords, or when creating or updating records.

**When to use each:**

* Use **GET** for:
  + Search forms
  + Filtering data
  + Navigating between pages
* Use **POST** for:
  + User registrations
  + Login forms
  + Submitting feedback or comments
  + Uploading files

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

The **<label>** element associates a text description with a form control, enhancing both usability and accessibility:

* **Improved Screen Reader Support**: Screen readers announce the label text when the associated form control gains focus, aiding users with visual impairments.
* **Enhanced Clickable Area**: Clicking the label text focuses on the corresponding form control, which is particularly beneficial for users with motor impairments.

To correctly associate a label with a form control, ensure the for attribute of the <label> matches the id of the form control:

**<label for="username">Username</label>**

**<input type="text" id="username" name="username">**

**HTML Tables**

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

An HTML table is structured using several elements, each serving a specific purpose:

* **<table>**: Defines the entire table structure.
* **<tr>**: Represents a table row, containing one or more cells.
* **<th>**: Specifies a header cell in a table row. Text within <th> elements is typically bold and centered by default.
* **<td>**: Denotes a standard data cell in a table row.
* **<thead>**: Groups the header content in a table, typically containing one or more <tr> elements with <th> cells. This semantic grouping aids in accessibility and styling.

|  |
| --- |
| <table>  <thead>  <tr>  <th>Month</th>  <th>Savings</th>  </tr>  </thead>  <tbody>  <tr>  <td>January</td>  <td>$100</td>  </tr>  <tr>  <td>February</td>  <td>$80</td>  </tr>  </tbody>  </table> |

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

**colspan**: Allows a cell to span across multiple columns.

**rowspan**: Enables a cell to span across multiple rows.

|  |
| --- |
| colspan : Example  <table>  <thead>  <tr>  <th colspan="2">Student</th>  <th>Major</th>  </tr>  </thead>  <tbody>  <tr>  <td>John</td>  <td>Doe</td>  <td>Physics</td>  </tr>  </tbody>  </table> |

|  |
| --- |
| **Row span :**  **<table>**  **<thead>**  **<tr>**  **<th rowspan="2">Student</th>**  **<th>Major</th>**  **</tr>**  **<tr>**  **<th>Credits</th>**  **</tr>**  **</thead>**  **<tbody>**  **<tr>**  **<td>John</td>**  **<td>Physics</td>**  **<td>240</td>**  **</tr>**  **</tbody>**  **</table>** |

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

Using tables for layout purposes is discouraged because:

* **Accessibility Issues**: Screen readers may misinterpret the layout, leading to a poor user experience for visually impaired users.
* **Maintenance Challenges**: Table-based layouts can be cumbersome to modify and maintain, especially as design requirements evolve.
* **SEO Concerns**: Search engines may misinterpret the content structure, potentially affecting search rankings.

**Better Alternative:**

**CSS (Cascading Style Sheets)** is the recommended approach for layout design. CSS offers greater flexibility, responsiveness, and separation of content from presentation. Techniques like Flexbox and Grid Layout provide powerful tools for creating complex layouts without relying on tables.