**HTNL Basics**

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

**What is HTML?**

**HTML** stands for **HyperText Markup Language**. It is the **standard language used to create and structure content** on the web. HTML is not a programming language; rather, it is a **markup language** that tells a web browser how to display text, images, links, and other elements on a webpage.

**Purpose of HTML in Web Development**

1. **Structure Web Content**  
   HTML provides the basic structure of web pages using elements like headings (<h1> to <h6>), paragraphs (<p>), lists (<ul>, <ol>, <li>), and divisions (<div>).
2. **Embed Media and Elements**  
   It allows embedding images (<img>), videos (<video>), audio (<audio>), and interactive forms (<form>).
3. **Linking Web Pages**  
   HTML uses hyperlinks (<a>) to connect different pages and resources, enabling easy navigation across a website or to external content.
4. **Semantic Meaning**  
   Modern HTML includes **semantic elements** like <article>, <section>, <header>, <footer>, and <nav>, which provide meaning to the content and improve accessibility and SEO.
5. **Foundation for CSS and JavaScript**  
   HTML works together with:
   * **CSS (Cascading Style Sheets)** to style the appearance.
   * **JavaScript** to add interactivity and dynamic behavior.

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

**Basic Structure of an HTML Document**

An HTML document follows a specific structure made up of nested tags that define the page content and layout.

**📄 Example of a Basic HTML Document:**

<!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** |
| --- | --- |
| <!DOCTYPE html> | Declares the document type and HTML version (HTML5). It helps the browser render the page correctly. |
| <html> | Root element that wraps the entire HTML document. |
| <head> | Contains metadata (data about data) such as title, styles, and links to scripts or stylesheets. Not displayed on the page. |
| <title> | Sets the title of the webpage shown on the browser tab. |
| <body> | Contains the visible content of the webpage, such as text, images, links, and more. |

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

### ****Block-Level Elements:****

* **Behavior:** Always start on a new line.
* **Width:** Takes up the full width of the parent container by default.
* **Purpose:** Used to structure major sections of a page.

**Examples of block-level elements:**

<div>This is a block element</div>

<p>This is a paragraph</p>

<h1>This is a heading</h1>

<ul>

<li>List item</li>

</ul>

<section>This is a section</section>

### ****Inline Elements:****

* **Behavior:** Do **not** start on a new line.
* **Width:** Only takes up as much width as necessary.
* **Purpose:** Used for formatting content **within** block-level elements.

**Examples of inline elements:**

<span>This is a span</span>

<a href="#">This is a link</a>

<strong>Bold text</strong>

<em>Italic text</em>

<img src="image.jpg" alt="An image">

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

**What is Semantic HTML?**

Semantic HTML uses meaningful tags to clearly describe the structure and purpose of content on a web page.

Instead of using generic tags like <div> or <span> for everything, semantic elements use tags like <article>, <header>, and <footer> that convey the type of content they contain.

**Why is Semantic HTML Important?**

**1. Accessibility**

* Helps screen readers and assistive technologies better understand the layout and content.
* Makes web navigation easier for users with disabilities.

**2. SEO (Search Engine Optimization)**

* Search engines like Google use semantic structure to **index content more effectively**.
* Helps improve your page ranking because search engines better understand what your page is about.

**3. Code Readability and Maintainability**

* Easier for developers to read and understand the layout and purpose of the content.

### ****Examples of Semantic HTML Elements:****

| **Tag** | **Purpose** |
| --- | --- |
| <header> | Defines the top section (e.g., logo, nav) |
| <nav> | Contains navigation links |
| <main> | Main content of the document |
| <section> | Thematic grouping of content |
| <article> | Independent content like blog posts |
| <aside> | Sidebars or supplementary content |
| <footer> | Bottom of the page (e.g., contact info) |
| <figure> & <figcaption> | Images with captions |

**Example:**

<article>

<header>

<h2>Why Use Semantic HTML?</h2>

</header>

<p>Semantic HTML improves accessibility and SEO...</p>

<footer>

<p>Written by Khushi </p>

</footer>

</article>

**HTML Forms :**

**[5] 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 commonly found in:

* Login and registration pages
* Search bars
* Contact forms
* Surveys and quizzes

Forms use the <form> element, and inside it are **input controls** like text fields, checkboxes, dropdowns, etc.

**✅ Key Form Elements and Their Purposes:**

**🔸 1. <input>**

* Used to create **various types of fields** like text, password, email, checkbox, radio buttons, etc.
* Accepts the type attribute to specify what kind of input is expected.

**Examples:**

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

<input type="email" name="email" placeholder="Enter email">

<input type="checkbox" name="subscribe"> Subscribe

**🔸 2. <textarea>**

* Used to collect **multi-line** text input (e.g., comments or messages).
* Unlike <input>, it can contain multiple rows of text.

**Example:**

<textarea name="message" rows="5" cols="30">Type your message here...</textarea>

**🔸 3. <select> and <option>**

* Used to create a **dropdown list**.
* Users can select one (or more with multiple attribute) option(s) from a list.

**Example:**

<select name="country">

<option value="in">India</option>

<option value="us">USA</option>

<option value="uk">UK</option>

</select>

**🔸 4. <button>**

* Used to **submit** the form or trigger other actions via JavaScript.
* Can have type="submit", type="reset", or type="button".

**Example:**

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

**🧩 Simple Form Example:**

<form action="/submit" method="post">

<label>Name: <input type="text" name="name"></label><br>

<label>Message: <textarea name="message"></textarea></label><br>

<label>Country:

<select name="country">

<option value="in">India</option>

<option value="us">USA</option>

</select>

</label><br>

<button type="submit">Send</button>

</form>

**[6] Explain the difference between the GET and POST methods in form submission. When should each be used?**

| **Feature/Aspect** | **GET Method** | **POST Method** |
| --- | --- | --- |
| **Data Location** | Sent in the **URL** (?key=value) | Sent in the **request body** (invisible in URL) |
| **Visibility** | Visible in the browser address bar | Hidden from the URL |
| **Security** | ❌ Less secure – data exposed in URL | ✅ More secure – not exposed in URL |
| **Data Length Limit** | Limited (URL length restrictions) | No significant limit (larger data allowed) |
| **Browser Caching** | ✅ Can be cached | ❌ Not cached |
| **Bookmarking** | ✅ Can be bookmarked with form data | ❌ Cannot be bookmarked |
| **Use Cases** | Viewing data, search, filters | Submitting sensitive or large data, uploading files |
| **Back Button Behavior** | Parameters remain and can be resubmitted | Form usually resubmitted on back navigation |
| **Example Use** | Search bar, filter forms | Login, registration, contact forms |

**When to Use Each:**

* **Use GET when:**
  + You are retrieving or viewing data (no side effects)
  + You want users to bookmark or share URLs with parameters
* **Use POST when:**
  + You are submitting, modifying, or uploading data
  + You need to send sensitive or large amounts of data

**[7] What is the purpose of the label element in a form, and how does it improve accessibility?**

The purpose of the <label> element in a form is to provide a clear, descriptive name for a form control like an input, textarea, or select element.

**How it improves accessibility:**

1. Associates text with form controls  
   The <label> tells users what each form field is for, making forms easier to understand.
2. Supports screen readers  
   For users with visual impairments, screen readers read the label so they know what the form field is asking for.
3. Increases clickable area  
   Clicking on the label sets focus to the input field, helping users with limited motor control.

**Example:**

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

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

**In this example:**

* The label is linked to the input by the for and id attributes.
* Screen readers can announce “Username” when the input is selected.
* Clicking “Username” focuses the input field.

HTML Table :

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

### ****1.**** <table> ****– Table Container****

* The <table> element wraps all the table content.
* It defines the beginning and end of the table.

### ****2.**** <tr> ****– Table Row****

* Stands for **Table Row**.
* Each <tr> represents a **single row** of the table.
* It can contain header cells (<th>) or data cells (<td>).

### ****3.**** <td> ****– Table Data Cell****

* Stands for **Table Data**.
* Used to define **individual data cells** within a row.
* Must be placed inside a <tr>.

### ****4.**** <thead> ****– Table Head****

* Groups the **header section** of a table.
* Usually contains one or more <tr> rows with <th> cells (not <td>).
* Helps screen readers and browsers understand table structure.

**Example :**

<table>

<thead>

<tr>

<th>Name</th>

<th>Age</th>

</tr>

</thead>

<tr>

<td>John</td>

<td>25</td>

</tr>

<tr>

<td>Emily</td>

<td>30</td>

</tr>

</table>

**[9] What is the difference between colspan and rowspan in tables? Provide examples.**

| **Attribute** | **Description** |
| --- | --- |
| colspan | Merges **columns** (i.e., spans across multiple columns in a single row) |
| rowspan | Merges **rows** (i.e., spans across multiple rows in a single column) |

**colspan Example :**

<table border="1">

<tr>

<th colspan="2">Name & Age</th>

</tr>

<tr>

<td>John</td>

<td>25</td>

</tr>

</table>

**Rowspan Example :**

<table border="1">

<tr>

<td rowspan="2">John</td>

<td>Math</td>

</tr>

<tr>

<td>Science</td>

</tr>

</table>

**[10]** **Why should tables be used sparingly for layout purposes? What is a better alternative?**

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

Using HTML tables to control **page layout** (positioning elements like headers, sidebars, etc.) is discouraged because:

1. **Not Semantically Correct**
   * Tables are meant for displaying **tabular data**, not structuring a webpage.
   * Misusing them reduces the clarity of your HTML code.
2. **Poor Accessibility**
   * Screen readers interpret tables as data, which confuses users with disabilities when used for layout.
3. **Hard to Maintain and Update**
   * Table-based layouts are rigid and difficult to modify, especially on large or complex pages.
4. **Not Responsive**
   * Tables do not adapt well to smaller screens like phones or tablets, making mobile design challenging.

### Better Alternative: ****CSS with Semantic HTML****

Instead of using tables for layout, modern web design uses:

| **Technique** | **Description** |
| --- | --- |
| **Flexbox** | Aligns elements in rows or columns |
| **Grid Layout** | Creates complex, responsive layouts easily |
| **Media Queries** | Adjusts layout based on screen size |

**Example :**

<style>

.container {

display: flex;

justify-content: space-between;

}

</style>

<div class="container">

<div>Left Section</div>

<div>Right Section</div>

</div>