**HTML Theory Assignment**

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

* HTML (HyperText Markup Language) is the standard markup language used to create and structure web pages. It provides the basic framework for websites by defin
* ing elements like headings, paragraphs, images, links, tables, and forms.
* Purpose in Web Development:  
   Defines the structure and content of a webpage.  
   Provides a way to insert text, images, audio, video, and links.  
   Works as the skeleton of a webpage, while CSS adds styling and JavaScript adds interactivity.  
   Ensures web content is properly displayed across browsers.

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

Basic Structure of an HTML Document:

<!DOCTYPE html>  
<html>  
 <head>  
 <title>My First Webpage</title>  
 </head>  
 <body>  
 <h1>Hello, World!</h1>  
 <p>This is my first webpage.</p>  
 </body>  
</html>

* Mandatory Tags and Their Purposes:  
  <!DOCTYPE html>: Declares the document type and version of HTML.  
   <html>: Root element that contains all the HTML content.  
   <head>: Contains metadata, title, links to CSS, scripts, etc.  
   <title>: Defines the title of the webpage (shown on the browser tab).  
   <body>: Contains the main visible content of the webpage (text, images, links, etc.).

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

* **Block-Level Elements:** Always start on a new line.  
   Occupy the full width available (stretch across the page).  
   Can contain other block-level or inline elements.  
  Examples: <div>, <p>, <h1> to <h6>, <section>, <article>, <ul>, <li>, <table>.  
    
  **Inline Elements:** Do not start on a new line.  
   Only take up as much width as needed.  
   Usually format text within block elements.  
  **Examples: <span>, <a>, <img>, <strong>, <em>, <br>.**

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

* **Role of Semantic HTML:**  
  Semantic HTML uses meaningful tags that describe the content’s role on a webpage, making the structure clear to both browsers and developers.  
    
  **Importance for Accessibility and SEO:**  
   Accessibility: Screen readers and assistive technologies can interpret semantic tags better, improving the experience for visually impaired users.  
   SEO: Search engines can understand the content hierarchy and relevance more effectively, improving page ranking.  
    
  **Examples of Semantic Elements**:  
   <header>: Represents the header section.  
   <nav>: Defines navigation menus.  
   <main>: Indicates the main content of the page.  
   <article>: Represents independent content (e.g., blog post).  
   <section>: Groups related content.  
   <footer>: Defines footer section of the page.

**Question 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 contain form elements that allow users to enter data.  
  
**Key Elements:** <input>: Used for single-line input such as text, numbers, passwords, etc.  
<textarea>: Used for multi-line input such as comments or messages.  
 <select>: Creates a dropdown list for selecting one or more options.  
 <button>: Used to submit forms, reset them, or trigger actions.

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

* **GET Method:** Sends form data appended to the URL (visible in the address bar).  
   Limited data capacity (~2000 characters).  
   Less secure (data is exposed in URL).  
   Useful for search queries or bookmarking.  
    
  **POST Method:**  
   Sends form data inside the HTTP request body (not visible in URL).  
   No size limitation for data.  
   More secure than GET.  
   Used for sensitive data (passwords, logins, file uploads).  
    
  **When to Use:**  
   GET: For retrieving or displaying non-sensitive, small data.  
   POST: For submitting or updating sensitive or large data.

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

Purpose of <label>:

Defines a label for form controls (input, select, textarea).  
 Helps users understand what each input field is for.  
  
How it Improves Accessibility:  
 Screen readers announce the label with the input, aiding visually impaired users.  
 Clicking the label focuses the input field, improving usability.  
 Makes forms more user-friendly and keyboard-accessible.

Question 8: Explain the structure of an HTML table and the purpose of <table>, <tr>, <th>, <td>, and <thead>.

Structure of an HTML Table:  
 <table>: Container for the table.  
 <tr>: Defines a row.  
 <th>: D efines a header cell (bold & centered).  
 <td>: Defines a data cell.  
 <thead>: Groups header content of the table.

* **Example**

**<table border="1">**

**<thead>**

**<tr>**

**<th>Name</th>**

**<th>Age</th>**

**</tr>**

**</thead>**

**<tr>**

**<td>Amit</td>**

**<td>22</td>**

**</tr>**

**</table>**

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

**Difference:** **colspan**: Merges columns (expands a cell across multiple columns).  
 **rowspan**: Merges rows (expands a cell across multiple rows).  
  
**Example Colspan:**

**<table border="1">**

**<tr>**

**<th colspan="2">Name</th>**

**<th>Age</th>**

**</tr>**

**<tr>**

**<td>First</td>**

**<td>Last</td>**

**<td>20</td>**

**</tr>**

**</table>**

**Example Rowspan:**

**<table border="1">**

**<tr>**

**<th rowspan="2">Name</th>**

**<td>Amit</td>**

**</tr>**

**<tr>**

**<td>Rahul</td>**

**</tr>**

**</table>**

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

Tables **should be used sparingly for layout purposes** because:

1. **Not Semantic** – Tables are meant for displaying tabular data (rows and columns of related information). Using them for layout confuses the structure of the page.
2. **Accessibility Issues** – Screen readers interpret tables as data, so using them for layout makes it hard for visually impaired users to navigate.
3. **Poor Maintainability** – Table-based layouts are rigid and hard to modify. Making even small design changes requires editing nested <table>, <tr>, and <td> elements.
4. **Slower Rendering** – Browsers need to load the entire table before displaying it, which can slow down page rendering.
5. **Not Responsive** – Tables don’t adapt well to different screen sizes (like mobile devices), leading to broken or unreadable layouts.

* **Better Alternative:**  
  Use **CSS** for page layout:
* **CSS Flexbox** : Great for one-dimensional layouts (rows or columns).
* **CSS Grid** : Best for two-dimensional layouts (rows + columns).
* **CSS Positioning & Float** : Can be used for specific placement, though modern flexbox/grid are preferred.