HTML THEORY

1. HTML Basics

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

Definition:

HTML stands for HyperText Markup Language.
It is the standard language used to create and structure content on web pages. Purpose in

Web Development:

- HTML is used to define the structure of a web page — such as headings, paragraphs, images, links, tables, and forms.
- It tells the web browser how to display text, multimedia, and layout.
- It works as the skeleton of a website, while CSS adds style and JavaScript adds interactivity.

Example:

```
<!DOCTYPE html>
<html>
<head>
  <title>My First Webpage</title>
</head>
<body>
  <h1>Welcome to My Website</h1>
  <p>This is a simple webpage created using HTML.</p>
</body>
</html>
```

---

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>Page Title</title>
</head>
<body>
  <h1>This is a Heading</h1>
  <p>This is a paragraph.</p>
</body>
</html>
```

Mandatory Tags and Their Purpose:

Tag	Purpose
<!DOCTYPE html>	Declares the document type and version of HTML (HTML5).
<html>	Root element that contains the entire HTML document.
<head>	Contains metadata (like title, styles, scripts) that is not shown on the webpage.
<title>	Sets the title of the webpage (appears in the browser tab).
<body>	Contains all visible content of the webpage (text, images, links, etc.).

---

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

Feature	Block-Level Elements	Inline Elements
Definition	Elements that start on a new line and take up the full width available.	Elements that stay in the same line and take up only the required width.
Display Behavior	Always start on a new line.	Do not start on a new line.
Examples	<div>, <p>, <h1>–<h6>, <ul>, <li>, <table>, <section>	<span>, <a>, <strong>, <em>, <img>, <label>

Example Code:

```
<!-- Block-Level Example -->
<p>This is a paragraph (block element).</p>
```

```
<div>This is a div (block element).</div>


<!-- Inline Example -->

<span>This is a span (inline element).</span>

<a href="#">This is a link (inline element).</a>
```

---

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 means using HTML tags that clearly describe their meaning and purpose to both browsers and developers.

Importance:

- Accessibility: Screen readers and assistive technologies can easily understand the structure of a page.
- SEO (Search Engine Optimization): Search engines can better understand page content and rank it accurately.
- Readability: Makes code easier to read and maintain.
- Standard Practice: Promotes a clean, structured, and meaningful webpage.

Examples of Semantic Elements:

Element	Purpose
<header>	Represents the top section or navigation area of a webpage.
<nav>	Defines a set of navigation links.
<main>	Represents the main content area.
<article>	Represents an independent piece of content (like a blog post).
<section>	Defines a section in a document.
<aside>	Represents sidebar or related content.
<footer>	Defines the footer or bottom section of a webpage.

Example Code:

```
<header>
```

```
  <h1>My Blog</h1>
  <nav>
    <a href="#">Home</a> |
    <a href="#">About</a> |
    <a href="#">Contact</a>
  </nav>
</header>

<main>
  <article>
    <h2>HTML Basics</h2>
    <p>HTML stands for HyperText Markup Language...</p>
  </article>
</main>

<footer>
  <p>© 2025 Jay Sompura</p>
</footer>
```

---

2. HTML Forms

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

Answer:

HTML forms are used to collect user input on a web page and send that data to a server for processing. For example – forms are used for login pages, registration, contact forms, feedback forms, search bars, etc.

HTML forms are created using the <form> tag, and inside it, we use various form elements like <input>, <textarea>, <select>, and <button>.

Main form elements:

1. <input> element:

- Used to take single-line input from the user.
- Example: text, email, password, number, date, etc.

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

2. <textarea> element:

   - Used to take multi-line input, like comments or messages.

```
<textarea name="message" rows="4" cols="30" placeholder="Enter your message"></textarea>
```

3. <select> element:

   - Used to create a dropdown list of options.

```
<select name="country">
   <option value="india">India</option>
   <option value="usa">USA</option>
   <option value="uk">UK</option>
</select>
```

4. <button> element:

   - Used to create a clickable button to submit or reset the form.

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

short:
Forms are used to collect user data, and these elements help in entering and submitting that data in different formats.

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

Answer:

In HTML, forms send data to the server using the method attribute of the <form> tag. There

are two main methods: GET and POST

Feature	GET Method	POST Method
Data visibility	Data is shown in the URL (e.g., ?name=Jay)	Data is hidden from the URL
Security	Less secure (data visible)	More secure (data hidden)
Feature	GET Method	POST Method

Data limit	Limited (around 2000 characters)	No size limit
Use case	Used for reading/fetching data (e.g., search queries)	Used for sending sensitive data (e.g., login, password)
Caching	Can be cached/bookmarked	Cannot be cached/bookmarked

Example:

```
<!-- GET Method Example -->
<form action="search.php" method="get">
  <input type="text" name="query" placeholder="Search...">
  <button type="submit">Search</button>
</form>


<!-- POST Method Example -->
<form action="login.php" method="post">
  <input type="text" name="username" placeholder="Username">
  <input type="password" name="password" placeholder="Password">
  <button type="submit">Login</button>
</form> When
```

to use:

- Use GET when sending non-sensitive data (like search filters).
- Use POST when sending sensitive or large data (like forms, passwords, or files).

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

The <label> element defines a caption or title for a form input (like a textbox, checkbox, etc.). It tells users what kind of information is expected in the input field.

Syntax:

```
<label for="username">Username:</label>
<input type="text" id="username" name="username">
```

Here, the for attribute in the label is linked to the input’s id. Purpose:

- Makes forms easier to understand.
- Helps users know which input field is for what data.
- Improves form accessibility for screen readers (used by visually impaired users).

Accessibility Benefit:
When a label is properly connected to an input, screen readers can read out the label text — helping blind users understand what to type.

Example:

```
<label for="email">Email Address:</label>

<input type="email" id="email" name="email" placeholder="Enter your email"> In
```

short:
The <label> element improves usability and accessibility, making forms more user-friendly and readable by assistive technologies.

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>. Answer:

An HTML table is used to display data in rows and columns (like an Excel sheet). It helps organize related information in a structured format.

The basic structure of a table looks like this:

```
<table>

  <thead>

    <tr>

      <th>Roll No</th>

      <th>Name</th>

      <th>Marks</th>

    </tr>

  </thead>
  <tr>
    <td>1</td>
    <td>Jay</td>
    <td>90</td>
  </tr>
  <tr>
    <td>2</td>
    <td>Ravi</td>
    <td>85</td>
  </tr>
</table>
```

Explanation of elements:

1. <table> (Table tag):
    - It defines the start and end of a table.
    - All table-related elements (<tr>, <th>, <td>, <thead>, etc.) are written inside it.
    - Example:
    - <table> ... </table>
2. <tr> (Table Row):
    - Defines a row in a table.
    - Each <tr> contains multiple cells (<th> or <td>).
    - Example:
    - <tr><td>1</td><td>Jay</td><td>90</td></tr>
3. <th> (Table Header Cell):
    - Defines a header cell (column heading).
    - Text in <th> is bold and centered by default.
    - Example:
    - <th>Student Name</th>
4. <td> (Table Data Cell):
    - Defines a data cell in a table (actual content).

- Example:
- <td>Jay</td>

5. <thead> (Table Head Section):
    - Used to group the header rows of a table.
    - Usually contains <tr> and <th> elements.
    - Helps in styling or fixing header rows when scrolling large tables.
    - Example:
    - <thead>
    -   <tr><th>ID</th><th>Name</th></tr>
    - </thead>

In short:
<table> defines the whole table,
<tr> defines a row,
<th> defines a header cell,
<td> defines data,
and <thead> groups all table headers together.

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

colspan and rowspan are table cell attributes used to merge multiple cells in a table.

Attribute	Meaning	Used To	Example Output
colspan	Combine columns (side-by-side)	Merge cells horizontally	One cell covers multiple columns
rowspan	Combine rows (top-to-bottom)	Merge cells vertically	One cell covers multiple rows

---

Example 1: colspan

```
<table border="1">

  <tr>

    <th colspan="2">Student Info</th>

  </tr>

  <tr>
```

```
    <td>Name</td>

    <td>Jay</td>

  </tr>

</table>
```

Here, the first cell spans 2 columns — “Student Info” is centered across both columns.

Example 2: rowspan

```
<table border="1">

  <tr>

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

    <td>Jay</td>

  </tr>

  <tr>

    <td>Ravi</td>

  </tr>

</table>
```

Here, the “Name” header spans 2 rows vertically.

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

In older websites, tables were often used for page layouts (like dividing sections, columns, and banners). But this is not recommended today because:

- Disadvantages of using tables for layout:

1. Not responsive: Tables do not adjust well on small screens like mobiles.
2. Hard to maintain: Table-based layout code is large and complex.
3. Slow to load: More HTML code means slower performance.
4. Poor accessibility: Screen readers may find it difficult to interpret layout tables.
5. Not SEO-friendly: Search engines may not properly understand table-based layouts.

- **Better Alternative:**

Use CSS layout techniques like:

- **Flexbox (display: flex)**
- **CSS Grid (display: grid)**
- **Bootstrap framework (for responsive layouts)**

Example using CSS Flexbox:

```
<div style="display: flex; gap: 10px;">
  <div style="flex: 1; background: lightblue;">Column 1</div>
  <div style="flex: 1; background: lightgreen;">Column 2</div>
</div>
```

In short:
Use tables only for tabular data, not for webpage design.
For page structure and layout — CSS Flexbox or Grid is the modern, responsive, and cleaner solution.