Module – 1 HTML

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

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

- **Purpose of HTML in Web Development:**

Structure of Web Pages:
HTML provides the basic structure or *skeleton* of a webpage.

Content Display:
It helps display text, images, videos, and links in a browser.

Linking Web Pages:
HTML allows you to connect multiple pages using hyperlinks (<a> tag).

Foundation for CSS and JavaScript:
HTML works together with **CSS (for styling)** and **JavaScript (for interactivity)** to build complete and dynamic websites.

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

<!DOCTYPE html>

<html>

<head>

<title>My Web Page</title>

</head>

<body>

<h1>Welcome to My Website</h1>

This is a sample paragraph.

</body>

</html>

Mandatory Tags in an HTML Document

Tag	Purpose
<!DOCTYPE html>	Defines document type (HTML5).
<html>	Root tag; contains all HTML content.
<head>	Holds metadata and the page title.
<title>	Displays title in browser tab.
<body>	Displays visible webpage content.

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

1. Block-Level Elements

Definition:

Block-level elements **start on a new line** and take up the **full width** available (from left to right of the page).

Characteristics:

- Always start on a new line.
- Occupy the full width of their container.
- Can contain **other block-level or inline elements**.

Examples:

```
<div>Content inside a div</div>

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

<h1>This is a heading</h1>

<ul>

  <li>List item</li>

</ul>
```

2. Inline Elements

Definition:

Inline elements **do not start on a new line** and take up **only as much width as necessary** (just the space around their content).

Characteristics:

Stay **within the same line** as other elements.

Only take up space equal to their content.

Can contain **text or other inline elements** (but not block elements).

Examples:

```
<span>This is inline text</span>

<a href="#">Click here</a>

<strong>Bold text</strong>

<img src="image.jpg" alt="Image">
```

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

Semantic HTML makes webpages **meaningful, accessible, and SEO-friendly** by using elements that describe their true purpose.

Why It's Important for Accessibility

- Screen readers (used by visually impaired users) rely on semantic tags to **navigate** a page.
- Example: A screen reader knows <nav> means *navigation area*, <article> means *independent content*, and <footer> means *page end*.
- Without semantic HTML, assistive tools might not know what each section represents.

Why It's Important for SEO

- Search engines analyze semantic tags to understand **what your content is about**.
- Proper use of tags (like <article>, <header>, <section>) helps **index your site more effectively**.
- Using <h1> to <h6> correctly improves your page's **ranking and visibility**.

Examples of Semantic Elements

Tag	Meaning / Purpose
<header>	Defines the header of a page or section (like a title or logo).
<nav>	Defines a navigation area with links.
<section>	Defines a section or group of related content.
<article>	Defines independent, self-contained content (like a blog post or news).

Tag	Meaning / Purpose
<aside>	Defines side content such as a sidebar or ads.
<footer>	Defines the footer of a page or section.
<main>	Defines the main or central content area of a page.
<figure> and <figcaption>	Used to group images with captions.

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

An **HTML form** is used to **collect user input** on a webpage.
It allows users to **enter data** (like name, email, password, feedback, etc.) and **send it to a server** for processing (e.g., login, sign-up, search, feedback submission).

- **More on Each Element**

1. <input>

- Most versatile form element.
- Common types:
  - type="text" – for text
  - type="email" – for emails
  - type="password" – for passwords
  - type="checkbox" – for multiple selections
  - type="radio" – for single choice

  type="submit" – for form submission

2. <textarea>

- Allows users to type **multiple lines of text** (e.g., feedback or comments).

3. <select> and <option>

- Used for **drop-down menus**.

4. <button>

- Creates a **clickable button** for various actions.
- Types include:
  - type="submit" – submits form data
  - type="reset" – clears form fields
  - type="button" – used for custom actions (like JavaScript)

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

When you submit an HTML form, the browser sends the data to the server using either the **GET** or **POST** method (defined in the <form> tag's method attribute).

1. GET Method

Definition:

- The **GET** method sends form data **through the URL** (as query parameters).

2. POST Method

Definition:

- The **POST** method sends form data **inside the HTTP request body**, not visible in the URL.

When to Use Each

- Use **GET** → When you're just **retrieving data** (like a search or filter).
- Use **POST** → When you're **sending or modifying data** (like login, form submission, uploads).

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

The <label> element is used to **define a caption or description** for an input field in a form.

- **Purpose of <label>**

1. **Provides clarity** – It clearly tells users what to enter in each field.
2. **Improves usability** – Clicking on the label automatically focuses the input box (makes form easier to use).
3. **Improves accessibility** – Screen readers use labels to read out loud what each input field is for, helping visually impaired users.

- **Accessibility :**

- Screen readers can announce:
    - "Email Address — edit text" instead of just "edit text"
- Makes forms usable by people with disabilities or using voice navigation.
- Ensures compliance with **web accessibility standards** (like WCAG and ARIA).

8. 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 used to **display data in rows and columns**

- **Explanation of Each Tag**

Tag	Full Form / Meaning	Purpose / Description
<table>	Table	It is the **main container** for all table content. All rows, headers, and data cells go inside this tag.
<tr>	Table Row	Defines a **row** in the table. Each row can contain header cells (<th>) or data cells

Tag	Full Form / Meaning	Purpose / Description
		(<td>).
<th>	Table Header	Defines a **header cell**. Text inside <th> is **bold and centered** by default. It represents a heading for a column or row.
<td>	Table Data	Defines a **data cell** that holds actual data in the table.
<thead>	Table Head	Groups all the **header rows** (<th> elements). It separates the table's heading section from the body (<tbody>) and footer (<tfoot>).

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

colspan and rowspan are **HTML table attributes** used with the <th> or <td> elements.

They are used to **merge (combine)** multiple cells — either **horizontally** or **vertically**.

1. colspan **(Column Span)**

Definition:

colspan is used to **merge cells horizontally** — that is, across **multiple columns** in a single row.

Example :

<table border="1">

```
 <tr>
  <th colspan="3">Student Information</th>
 </tr>
 <tr>
  <th>Name</th>
  <th>Age</th>
  <th>City</th>
 </tr>
 <tr>
  <td>Vrushti</td>
  <td>22</td>
  <td>Ahmedabad</td>
 </tr>
</table>
```

2. rowspan (Row Span)

Definition:

rowspan is used to **merge cells vertically** — that is, across **multiple rows** in the same column.

Example:

```
<table border="1">
 <tr>
  <th rowspan="2">Name</th>
```

```
    <th>Subject</th>
    <th>Marks</th>
  </tr>
  <tr>
    <th>Science</th>
    <th>90</th>
  </tr>
</table>
```

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

In early web design, developers often used **HTML tables** to control the **layout** of webpages (like positioning text, images, etc.).

However, this practice is now **outdated** and **not recommended** for several reasons.

Better Alternative — Use CSS for Layout

Modern web design uses **CSS (Cascading Style Sheets)** for page layout and styling. CSS allows more **flexible**, **responsive**, and **accessible** design structures.