Module 1: Introduction to Web Development

1. What is Web Development?

   Web development refers to the process of building, creating, and maintaining websites or web applications. It encompasses a wide range of activities, from designing the website's layout and interface to writing code, managing server infrastructure, and optimizing the user experience.

---

2. What is a website? And explain the differences between static and dynamic websites.

   A **website** is a collection of interconnected web pages that are hosted on a server and accessible through the internet. Websites are designed to provide information, services, or functionality to users via a web browser, such as Google Chrome, Mozilla Firefox, Safari, or Microsoft Edge.

   Static Websites:

- Content is fixed and does not change dynamically.
- Examples: Portfolio sites, simple informational websites(Wikipedia).

   Dynamic Websites:

- Content changes based on user interactions or data from the server.
- Examples: Social media platforms, e-commerce sites.

---

3. Explain the differences of web design, front-end and back-end development.

    1. **Web Design:** Focuses on the aesthetics and usability of a website.

    2. **Frontend Development** (Client-side):
        - Focuses on the user interface (UI) and user experience (UX).

- Involves coding the visual elements of a website that users interact with directly.
- Common technologies:
    - **HTML (HyperText Markup Language):** Structures the content of a webpage.
    - **CSS (Cascading Style Sheets):** Styles the layout, colors, and fonts.
    - **JavaScript:** Adds interactivity and dynamic behavior to web pages.
- Example tools: React, Angular, Vue.js.

3. **Backend Development** (Server-side):
    - Handles the behind-the-scenes functionality, such as data processing, business logic, and database management.
    - Involves building and managing the server, application, and database.
    - Common technologies:
        - Server-side languages: Python, Ruby, PHP, Java, Node.js.
        - Databases: MySQL, PostgreSQL, MongoDB.
    - Example tools: Django, Flask, Express.js, Laravel.

Module 2: HTML

1. What does HTML stand for? And Explain what HTML is and why we use it?

HTML stands for **HyperText Markup Language**.

HTML is the standard markup language used to create and structure content on the web. It is the foundation of every webpage and serves as the skeleton for displaying text, images, links, videos, and other multimedia on the internet.

- **HyperText**: Refers to text that contains links (hyperlinks) to other documents or resources.
- **Markup Language**: A system of tags that define the structure and layout of web content.

HTML works by using a system of elements (tags) to organize and format information.

We use HTML for-

Structuring Web Pages: HTML organizes content into headings, paragraphs, lists, tables, etc., ensuring a logical layout and readability.

Embedding Multimedia: HTML allows the inclusion of images, videos, audio, and other media into web pages.

Hyperlinking: Hyperlinks enable navigation between different web pages or resources.

Browser Interpretation: Web browsers (like Chrome, Firefox, and Safari) interpret HTML to render the content of a website visually for users.

Foundation for Web Development: HTML serves as the base layer for websites, often combined with:

- **CSS**: To style the appearance of elements.
- **JavaScript**: To add interactivity and dynamic features.

Accessibility: HTML, when used correctly, ensures content is accessible to all users, including those using assistive technologies like screen readers.

---

2. Explain the HTML structure. E.g. what <!DOCTYPE html> is and how many parts in HTML structure and so on.

HTML documents follow a standardized structure that organizes content and metadata. This structure ensures that browsers can correctly interpret and render the webpage.

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home</title>
</head>
<body>
    <p>Hello DCS Students. Have a nice day!</p>
    <p>Hello DCS Students. Have a nice day!</p>
    <p>Hello DCS Students. Have a nice day!</p>
    <p>Hello DCS Students. Have a nice day!</p>
    <p id="p">Hello</p>
</body>
</html>
```

Breakdown:

- `<!DOCTYPE html>`: Declares the document type and version of HTML being used. It tells the web browser that the document adheres to the HTML5 standard. Always at the very beginning of the HTML document.

- `<html>`: The root element of the page and
- attribute `lang`: Specifies the language of the document (e.g., `lang="en"` for English).

- `<head>`: Contains metadata like the page title.
- `<body>`: Contains the content visible to users.
- `<h1>`: A large heading.
- `<p>`: A paragraph.
- `<a>`: A hyperlink.

---

3. What are the differences between HTML tags and HTML elements?

A tag is the **syntax** used in HTML to mark the beginning and/or end of an HTML element.

Tags are enclosed in angle brackets (`< >`).

Types of Tags:

- **Opening Tag**: Marks the start of an element. Example: `<p>`
- **Closing Tag**: Marks the end of an element, preceded by a slash (`/`). Example: `</p>`

An element consists of an **opening tag**, **content**, and a **closing tag** (if required).

An element is the complete structure that defines the functionality or content on the webpage.

Example: <h1>Welcome to My Website</h1>

Though the terms **HTML tags** and **HTML elements** are often used interchangeably, they have distinct meanings in the context of HTML.

---

4. Describe at least 10 HTML tags and explain their definition.

1. **`<html>`**
    - **Definition**: The root element of an HTML document. It contains all the other elements and represents the entire webpage.

---

2. **`<head>`**

- **Definition**: Contains metadata and links to resources, such as CSS stylesheets and JavaScript files. This section is not displayed directly on the webpage.

---

3. `<body>`

- **Definition**: Contains the content of the webpage that is visible to users, such as text, images, and interactive elements.

---

4. `<h1>` to `<h6>`

- **Definition**: Represent headings, with `<h1>` being the largest and most important, and `<h6>` being the smallest.

---

5. `<p>`

- **Definition**: Defines a paragraph of text.

---

6. `<a>`

- **Definition**: Defines a hyperlink, used to navigate to another page or resource.
- **Attributes**:
  - `href`: Specifies the URL of the link.

---

7. `<img>`

- **Definition**: Embeds an image in the webpage.
- **Attributes**:
  - `src`: Specifies the image source (URL or file path).

- `alt`: Provides alternative text for accessibility.

---

8. `<ul>` and `<li>`

- **Definition**: `<ul>` creates an unordered (bulleted) list, and `<li>` represents an individual list item.

---

9. `<table>`, `<tr>`, `<td>`

- **Definition**: Used to create tables.
  - `<table>`: Defines the table.
  - `<tr>`: Defines a row in the table.
  - `<td>`: Defines a cell in a row.

---

10. `<form>`

- **Definition**: Defines a form for user input.
- **Attributes**:
  - `action`: Specifies the URL to send form data.
  - `method`: Specifies the HTTP method (`GET` or `POST`)

<u>10 types of HTML tags</u>

1. **Structural Tags**:

- Define the basic structure of an HTML document.
- Eg. <html> , <head>, <body>

2. **Headings and Text Formatting Tags**:

- Used to structure and format text.
- Eg. <h1> to <h6>, <p>, <b>, <i>

3. **Linking and Navigation Tags**:
- Used to create links and navigation menus.
- Eg. <nav>, <a>

4. **Media Tags**:
- Used to embed images, audio, and video.
- Eg. <img>, <audio>, <video>

5. **List Tags**:
- Used to create ordered, unordered, or description lists.
- Eg. , <ol>, <li>, <dl>

6. **Table Tags**:
- Used to create tables.
- Eg. <table>, <tr>, <td>, <th>

7. **Form Tags**:
- Used to collect input from users.
- Eg. <form>, <input>, <textarea>, <option>, <button>, <label>, <select>

8. **Semantic Tags** (HTML5):
- Introduce meaning to the structure of a webpage.
- Eg. <header>, <footer>, <article>, <section>, <aside>, <main>

9. **Scripting Tags:**
- **Enable dynamic behavior on a webpage.**
- Eg. <script>, <noscript>

10. **Self-Closing Tags**:
- Do not require a closing tag.
- Eg. <img>,
, <hr>, <meta>, <input>

---

5. What is **HTML Attribute**?

An **HTML attribute** *provides additional information* about an HTML element, enhancing its functionality and controlling its behavior or appearance. Attributes are always included *within the opening tag of an element*.

Common HTML attributes are:

1. *Id* : Uniquely identifies an element on the page.
2. *class* : Groups elements for styling or scripting.
3. *style* : Adds inline CSS to an element.
4. *href* : Specifies the URL for a hyperlink.
5. *src* : Specifies the source file for an embedded resource, like an image or video.
6. *alt* : Provides alternative text for an image, improving accessibility.
7. *title* : Adds a tooltip that appears when the user hovers over the element.
8. *disabled* : Disables an input element or button.
9. *name* : Identifies form elements for backend processing.
10. *value* : Specifies the initial value of form elements like inputs.
