HTML NOTES

Unit1:

Basic principles involved in developing a web site:

HTML stands for (Hyper Text Markup Language), it is the foundation of web development, it is used to structure of content on the web.

HTML is made up of **elements**, defined by **tags** (e.g., <html>, <head>, <body>, <p>, <a>, etc.).

Tags usually come in **pairs**: an opening tag (<p>) and a closing tag (</p>).

It is not programming a language it's a markup language.

It defines the structure of a website.

a) Proper Structure and Semantics

- **HTML5 semantic tags** ka use karo:
  - <header> → page heading
  - <nav> → navigation menu
  - <main> → main content
  - <section> → use for different topics
  - <article> → use for articles/blog posts
  - <footer> → use for info at the bottom of page

b) Clean and Readable Code

- Indentation (proper spacing).
- Comments (<!-- comment -->) use comments for code explanation.
- Avoid extra spaces and unused codes.

c) Document Structure

A basic HTML document has a defined structure:

```
<!DOCTYPE html>
<html>
 <head>
  <title>Page Title</title>
 </head>
```

```
  <body>
    <h1>Heading</h1>
    <p>This is a paragraph</p>
  </body>
</html>
```

- <!DOCTYPE html>: Declares the document type (HTML5).
- <html>: Root element.
- <head>: Metadata (title, styles, links).
- <body>: Visible page content.

SEO stands for – Search Engine Optimization.

It using the structure and semantic elements of HTML to make a webpage's content more understandable for search engine and it ultimately improve its ranking in search results.

---

d) Text Formatting and Media

- Use tags like <h1> to <h6> for headings.
- <p> tags for paragraphs.
- <a href="URL"> use for links.
- <img src="image.jpg"> use for images.
- <ul> (unordered), <ol> (ordered), <li> (list item) use for lists.

  <ul> (unordered): It defines an unordered list. Items within an unordered list are displayed with bullet by default, indicating the order of items is not significant.

  ```
  Ex. <ul>
        <li>Milk</li>
        <li>Eggs</li>
        <li>Bread</li>
      </ul>
  ```

  <ol> (Ordered List): It defines an ordered list. Items within an ordered list are displayed with numbers, letters, or Roman numerals by default, indicating the order of items is significant.

  ```
  Ex. <ol>
        <li>First step</li>
        <li>Second step</li>
  ```

```
  <li>Third step</li>
</ol>
```

<li> (List Item): It defines a single item within either an <ul> or an <ol> list. Every item in a list, whether ordered or unordered, must be enclosed within <li> tags.

```
Ex. <ul>
      <li>Item 1</li>
      <li>Item 2</li>
    </ul>

    <ol>
      <li>Task A</li>
      <li>Task B</li>
    </ol>
```

e) Attributes

- HTML tags can have **attributes** to provide extra information.

  Ex. <img src="image.jpg" alt="Description" width="300">

f) Separation of Concerns

- Use **CSS** for styling (<style> or linked **.css** files).
- Use **JavaScript** for behaviour and interactivity (<script> or linked **.js** files).
- HTML should focus on **structure only**.

g) Responsive Design (Basic Setup)

- Use the <meta name="viewport" content="width=device-width, initial-scale=1.0"> tag in <head> to ensure the site displays well on all devices.

---

6. Forms and Input

HTML supports user input through **forms**:

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

  <label>Name:</label>

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

  <input type="submit" value="Submit">

</form>
```

Planning process:

The **planning process of HTML** involves steps that help ensure a website or web page is well-structured, user-friendly, and meets its goals.

1. Define the Purpose and Goals

- What is the website for? (e.g., blog, portfolio, business)
- Who is the target audience?
- What actions should users take? (e.g., sign up, read articles, buy products)

2. Content Planning

- What content will you include? (e.g., text, images, videos)
- Create a content outline or sitemap:
    - Home
    - About
    - Services
    - Contact

3. Layout of the Page

- Draw a basic layout of your pages.
- Identify areas like:
    - Header (logo, navigation)
    - Main content
    - Sidebar (optional)
    - Footer (contact info, links)

4. Choose the Structure (HTML Elements)

Plan what HTML tags you'll use:

- <header>, <nav>, <main>, <section>, <article>, <aside>, <footer>
- Text elements: <h1>–<h6>, <p>, <a>, <ul>, <li>, <img>, <form>, etc.

6. Plan for Responsiveness

- Think about how your layout will adapt to mobile, tablet, and desktop.
- Consider using CSS frameworks like Bootstrap or writing media queries.

7. Tool & Resource Planning

Decide on:

- Code editor (e.g., VS Code, Sublime)
- Libraries or frameworks (e.g., Bootstrap, Tailwind, jQuery)
- Version control (e.g., Git/GitHub)

◆ 8. Plan for Future Maintenance

- Use clean, commented, and modular code.
- Make sure others (or you in the future) can understand and update it.

Five Golden rules of web designing:

Five Golden Rules of Web Designing specifically tailored for websites built using **HTML (along with CSS and basic JS)**:

1. Keep It Simple and Consistent

- **Use clean and semantic HTML** to structure content.
- Avoid clutter: limit fonts, colours, and unnecessary elements.
- Maintain **consistent layout** and navigation across all pages.

📌 *Example:*

```
<header>
 <nav>
  <ul>
   <li><a href="index.html">Home</a></li>
   <li><a href="about.html">About</a></li>
  </ul>
 </nav>
</header>
```

2. Design for Accessibility

- Use semantic tags (<header>, <main>, <article>, etc.).
- Add **alt text** for images and **labels** for forms.
- Ensure the website can be navigated via keyboard.

📌 *Example:*

```
<img src="logo.png" alt="Company Logo">
<label for="email">Email:</label>
<input type="email" id="email" name="email">
```

3. Responsive Design

- Use HTML with **CSS media queries** or **responsive frameworks**.
- Design for mobile-first, then scale up to larger screens.

📌 *Tip:* Use the <meta> viewport tag:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

4. Fast Loading and Optimized Code

- Minimize HTML/CSS code.
- Optimize image sizes and avoid inline CSS when possible.
- Use proper file structure for better performance.

📌 *Example:*

```
<link rel="stylesheet" href="styles/main.css">
```

5. Clear Navigation and User Friendly

- Make links and buttons easy to find and intuitive.
- Use descriptive text in links and consistent navigation bars.

📌 *Example:*

```
<a href="services.html">View Our Services</a>
```

Designing navigation bar:

Designing a navigation bar (navbar) in HTML involves using HTML for structure and CSS for styling.

☑ **Simple Responsive Navigation Bar**

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Navigation Bar</title>
  <link rel="stylesheet" href="styles.css">
</head>
```

```
<body>
  <nav class="navbar">
    <div class="logo">MySite</div>
    <ul class="nav-links">
      <li><a href="#">Home</a></li>
      <li><a href="#">About</a></li>
      <li><a href="#">Services</a></li>
      <li><a href="#">Contact</a></li>
    </ul>
  </nav>
</body>
</html>
```

CSS (styles.css)

```
/* General Reset */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

/* Navbar Styling */
.navbar {
  display: flex;
  justify-content: space-between;
  align-items: center;
  background-color: #333;
  padding: 1rem 2rem;
}

.logo {
  color: white;
  font-size: 1.5rem;
  font-weight: bold;
}

.nav-links {
  list-style: none;
  display: flex;
  gap: 1.5rem;
}

.nav-links a {
  color: white;
  text-decoration: none;
  font-size: 1rem;
  transition: color 0.3s ease;
}

.nav-links a:hover {
  color: #00bcd4;
}
```

📦 **Key Features:**

- **Flexbox layout** for horizontal alignment.
- **Responsive basics** (mobile-friendly with media queries can be added).
- **Hover effect** for better interactivity.

Page design:

Common structure elements like header, navigation, main content, sidebar, and footer. This can serve as a template for most simple websites.

🌐 Example: Basic HTML Page Design

```
<!DOCTYPE html>
```

```
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1">
 <title>Simple Page Layout</title>
 <style>
  body {
   font-family: Arial, sans-serif;
   margin: 0;
  }
  header, footer {
   background-color: #333;
   color: white;
   text-align: center;
   padding: 1rem;
  }
  nav {
   background-color: #444;
   padding: 0.5rem;
  }
  nav a {
   color: white;
   margin: 0 1rem;
   text-decoration: none;
  }
  .container {
   display: flex;
   flex-wrap: wrap;
  }
  main {
      flex: 3;
      padding: 1rem;
      background-color: #f4f4f4;
    }
    aside {
      flex: 1;
      padding: 1rem;
      background-color: #ddd;
    }
    footer {
      margin-top: auto;
    }
  </style>
</head>
<body>

  <header>
    <h1>My Website</h1>
    <p>Welcome to my homepage</p>
  </header>

  <nav>
    <a href="#">Home</a>
    <a href="#">About</a>
    <a href="#">Services</a>
    <a href="#">Contact</a>
  </nav>

  <div class="container">
    <main>
```

```
      <h2>Main Content Area</h2>

      <p>This is where your main page content goes. You can add text, images, videos, etc.</p>

    </main>


    <aside>

      <h3>Sidebar</h3>

      <p>Additional content, ads, or links can go here.</p>

    </aside>

  </div>


  <footer>

    <p>&copy; 2025 My Website</p>

  </footer>


</body>

</html>
```

🔧 **You Can Customize:**

- Add sections for a **form**, **gallery**, or **contact info**
- Use frameworks like **Bootstrap** or **Tailwind CSS** for responsive design
- Enhance with **JavaScript** for interactivity

Home Page Layout:

Modern HTML5 structure along with some minimal CSS for styling. You can expand on this depending on your needs (e.g., adding navigation, hero section, footer, etc.).

✅ **Example: Simple Home Page Layout (HTML + CSS)**

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
<title>Home Page</title>

<style>

  * {

    margin: 0;

    padding: 0;

    box-sizing: border-box;

  }


  body {

    font-family: Arial, sans-serif;

    line-height: 1.6;

  }


  header {

    background: #333;

    color: #fff;

    padding: 1rem 2rem;

    text-align: center;

  }


  nav {

    background: #444;

    display: flex;

    justify-content: center;

    gap: 1.5rem;

    padding: 0.8rem 0;

  }


  nav a {

    color: #fff;
```

```
      text-decoration: none;
    }

    main {
      padding: 2rem;
      text-align: center;
    }

    .hero {
      background: #f4f4f4;
      padding: 2rem;
      margin: 2rem 0;
    }

    footer {
      background: #333;
      color: #fff;
      text-align: center;
      padding: 1rem 2rem;
      position: fixed;
      width: 100%;
      bottom: 0;
    }
  </style>
</head>
<body>

  <header>
    <h1>Welcome to Our Website</h1>
  </header>
  <nav>
    <a href="#">Home</a>
    <a href="#">About</a>
    <a href="#">Services</a>
    <a href="#">Contact</a>
  </nav>

  <main>
    <section class="hero">
      <h2>Your Dream Website Starts Here</h2>
      <p>We help you build responsive and modern web experiences.</p>
    </section>

    <section>
      <h3>Our Services</h3>
      <p>Web Design, Development, SEO Optimization, and more.</p>
    </section>
  </main>

  <footer>
    <p>&copy; 2025 YourCompany. All rights reserved.</p>
  </footer>

</body>
</html>
```

🔧 You Can Customize:

- **Colors & Fonts** in the <style> section
- **Add Images** using <img src="path" alt="description">

- **Add more sections** for testimonials, team, etc.
- **Use Bootstrap or Tailwind CSS** for faster styling