

Day-8

Task-1 : Building a Simple Web Page with HTML, CSS, and JavaScript

1. HTML (HyperText Markup Language)

HTML forms the structure of a web page. It defines the layout and the content, like headings, paragraphs, images, links, and more.

- **Example Elements:**
 - `<h1>`, `<p>`: Headings and paragraphs.
 - `<div>`: General container for content.
 - `<a>`: Links.
 - ``: Images.

HTML Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>My Web Page</title>
</head>
<body>
  <h1>Welcome to My Page</h1>
  <p>This is a simple webpage with HTML, CSS, and JavaScript.</p>
  <button id="myButton">Click Me</button>
</body>
</html>
```

2. CSS (Cascading Style Sheets)

CSS styles the HTML content. It adds color, spacing, and layouts, making the web page visually appealing.

- **Usage:**
 - Inline: Inside the `style` attribute of an HTML element.
 - Internal: Inside a `<style>` tag in the `<head>`.
 - External: In a separate `.css` file.

CSS Example (in an external file `styles.css`):

```
body {
  font-family: Arial, sans-serif;
  margin: 20px;
  background-color: #f5f5f5;
}

h1 {
  color: #4CAF50;
}

button {
  background-color: #4CAF50;
  color: white;
  padding: 10px 15px;
  border: none;
  cursor: pointer;
}

button:hover {
  background-color: #45a049;
}
```

Link the CSS file to the HTML:

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

3. JavaScript (JS)

JavaScript adds interactivity and logic to the web page. It allows you to handle user interactions like clicks and create dynamic content.

JavaScript Example (in an external file `script.js`):

```
// JavaScript to change button text on click
document.getElementById("myButton").addEventListener("click", function() {
  this.textContent = "You clicked me!";
  alert("Button was clicked!");
});
```

Link the JS file to the HTML:

```
<script src="script.js"></script>
```