Day 6: Quick Review Of HTML

Date: 12-06-24

---

2. Quick Review of Concepts:

- **Day 1: Introduction to HTML**
    - Definition and purpose of HTML.
    - Setting up the VS Code editor.
    - Basic HTML structure and tags (e.g., <!DOCTYPE html>, <html>, <head>, <body>).
- **Day 2: HTML Elements and Attributes**
    - Basic tags like <p>, <h1>, <ol>, <ul>, <a>, src, alt.
    - Creating our first HTML page using learned elements.
- **Day 3: Forms and Media**
    - Tags like <form>, <input>, <label>, <button>.
    - Adding images and links to the webpage.
- **Day 4: HTML Tables**
    - Table elements (<table>, <tr>, <th>, <td>, <caption>).
    - Table attributes (border, cellpadding, cellspacing).
- **Day 5: Semantic HTML**
    - Importance of semantic HTML for accessibility, SEO, and maintainability.
    - Common semantic tags like <header>, <nav>, <main>, <section>, <article>, <aside>, <footer>.
- **Day 6: Comprehensive Web Page Creation**
    - Tasked with building a complete webpage using all learned elements and semantic tags.
    - Integration of a form and a table.

3. Doubt Clearing Session:

- Addressed common questions and challenges faced during the week.
- Clarified misunderstandings and reviewed complex concepts.
- Discussed practical applications and best practices.

4. Feedback on HTML Webpages:

- **General Feedback:**

    - Overall structure and organization.
    - Use of semantic tags and HTML elements.
    - Clarity and readability of code.
- **Specific Feedback:**
    - Suggestions for improving layout and design.
    - Tips for better form and table integration.
    - Advice on optimizing images and links.

5. Actionable Feedback:

- **Improving Code Structure:**
    - Ensure consistent indentation and spacing for readability.
    - Use comments to explain sections of the code.
- **Enhancing Accessibility:**
    - Use descriptive alt attributes for images.
    - Ensure forms are labeled properly for screen readers.
- **Optimizing Tables:**
    - Use appropriate table attributes for better presentation.
    - Ensure tables are responsive and readable on all devices.

Day 7: Introduction to CSS

Date:13-08-24

---

1. What is CSS?

- **Definition:**
    - CSS stands for Cascading Style Sheets.
    - It is used to control the layout and appearance of HTML elements on a web page.
- **Purpose:**
    - Separates content (HTML) from presentation (CSS).
    - Enhances the visual appeal and user experience of web pages.
    - Enables consistent styling across multiple web pages.

2. Types of CSS:

- **Inline CSS:**
    - Applied directly to an HTML element using the style attribute.
    - Example:
-
- This is a blue paragraph.
- **Pros:**
    - Quick and easy to apply for single elements.
- **Cons:**
    - Not suitable for applying styles across multiple elements or pages.
    - Can make HTML code cluttered and harder to maintain.

Internal CSS:

- Defined within a <style> tag inside the <head> section of an HTML document.
- Example:
-
- <head>
- <style>
- p {
- color: blue;
- font-size: 14px;
- }
- </style>
- </head>
- <body>
- This is a blue paragraph.
- </body>
- **Pros:**
    - Useful for applying styles to a single page.
- **Cons:**
    - Styles are not shared across multiple pages.
    - Can increase the size of the HTML document.

External CSS:

- Defined in a separate .css file, which is linked to the HTML document.
- Example:
-
    - <!-- HTML file -->
    - <head>
    - <link rel="stylesheet" type="text/css" href="styles.css">
    - </head>
    - <body>
    - This is a blue paragraph.
    - </body>
    -
    - <!-- styles.css file -->
    - p {
    - color: blue;
    - font-size: 14px;
    - }
    -
    - **Pros:**
        - Allows for separation of content and style.
        - Styles can be applied across multiple pages.
        - Easier to maintain and update.
    - **Cons:**
        - Requires an additional HTTP request to load the CSS file.

3. Basic CSS Syntax:

- **Selectors:**
    - Target HTML elements to apply styles.
    - Examples:
        - element (e.g., p, h1)
        - class (e.g., .my-class)
        - id (e.g., #my-id)
- **Properties and Values:**

- Define the style to be applied to the selected elements.
- Syntax:

 selector {

 property: value;

}

 Example:

-
  - p {
  - color: blue;
  - font-size: 14px;
  - }
  -

4. Example of Using Different Types of CSS:

- **HTML File:**

 <!DOCTYPE html>

<html lang="en">

<head>

 <meta charset="UTF-8">

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

 <title>CSS Example</title>

 <style>

  /* Internal CSS */

  h1 {

   color: green;

   text-align: center;

  }

 </style>

 <link rel="stylesheet" type="text/css" href="styles.css">

</head>

<body>

 <h1>This is a heading</h1>

```
  <p style="color: blue; font-size: 14px;">This is an inline styled paragraph.</p>
  <p>This is an externally styled paragraph.</p>
</body>
</html>
```

 External CSS File (styles.css):

```
/* External CSS */
p {
  color: red;
  font-size: 16px;
}
```