Training Day 1 Report:
Date: 5 June 2024

Location: Science & Technology Entrepreneurs' Park

The training began with an introduction to **Web Development** and its foundational language – **HTML (HyperText Markup Language)**. The session emphasized how HTML structures content for websites, using elements, attributes, and tags. The instructor provided a comprehensive overview of how HTML acts as the building block for every website on the internet. We understood that without HTML, web browsers would have no way to interpret or display web content in a structured and meaningful way.

We were also made aware that HTML is a markup language and not a programming language. This is a crucial distinction because HTML doesn't perform any logic or computations; instead, it focuses solely on structuring the content. This includes headings, paragraphs, links, images, and multimedia elements that make a webpage functional and visually organized.

Overview of Topics Covered:

2.1.1 What is HTML

- HTML defines the structure and layout of a web document.
- It's not a programming language but a markup language using tags.
- Tags are used to mark up the beginning and end of an HTML element. For example, `<p>` denotes a paragraph and `</p>` marks the end.
- HTML forms the backbone of all web development, allowing developers to structure content in a way that web browsers can display accurately.
- We were shown a live example of a website without HTML structure, which appeared as plain, unformatted text, highlighting the importance of HTML.
- Real-world uses of HTML were discussed, including creating portfolios, e-commerce websites, educational blogs, and more.

2.1.2 HTML Editors

- Hands-on with editors like Notepad, VS Code, and online HTML playgrounds.
- We created our first basic HTML file.
- We learned the pros and cons of each editor. For example, Notepad is simple and helps understand HTML syntax manually, while VS Code is feature-rich with auto-completion, error-highlighting, and extensions.
- Online editors such as CodePen and JSFiddle allow us to test and share HTML/CSS/JS in real-time, making collaboration easier.
- We explored the default structure of an HTML document (`<!DOCTYPE html>`, `<html>`, `<head>`, `<body>`), and wrote our first "Hello World" HTML page.

2.1.3 HTML Elements

- Differentiating between block-level and inline elements.
- Block-level elements include `<div>`, `<p>`, `<h1>` – `<h6>`, `<ul>`, `<ol>`, and `<li>`.
- Inline elements include `<span>`, `<a>`, `<img>`, `<strong>`, and `<em>`.
- Practiced creating paragraphs, headings, lists, and div sections.
- We experimented with nesting elements within each other and discussed best practices for indentation and code readability.
- The session included exercises where we had to build a simple article page using both block and inline elements.

2.1.4 HTML Attributes

- Used `id`, `class`, `title`, `style`, `lang`, and `href` attributes.
- Understood their role in enhancing tags and interaction.
- The `id` and `class` attributes were demonstrated using CSS styling examples, showing how powerful these can be when combined with stylesheets or JavaScript.
- We explored how `href` works with anchor tags to link to external and internal pages.
- The `lang` attribute was shown in the context of accessibility and SEO, helping search engines and screen readers determine the language of the content.
- Exercises involved linking to a favorite website and applying styles via attributes.

2.1.5 HTML Headings

- Practiced all 6 levels: `<h1>` to `<h6>`.
- Discussed SEO importance of heading hierarchy.
- Learned that `<h1>` should be reserved for the main title and used only once per page, while `<h2>` to `<h6>` organize sub-content hierarchically.
- We were taught how search engines like Google prioritize properly structured heading tags for indexing content.
- We created a mock article with proper heading tags to mimic a blog structure, ensuring each section followed a logical and semantic hierarchy.

2.1.6 HTML Styles

- Introduced inline styling with the `style` attribute.
- Set color, font-size, background, and text alignment.
- Discussed why inline styles are generally avoided in larger projects in favor of CSS.
- Learned to style individual tags by adding multiple style properties separated by semicolons.
- Examples included setting a blue background for a heading, changing paragraph font sizes, and centering text using `text-align`.
- We were given an assignment to style a small poem using various inline styles to make it visually appealing.

2.1.7 HTML Formatting

- Learned about `<b>`, `<strong>`, `<i>`, `<em>`, `<mark>`, `<small>`, `<del>`, `<ins>`, and `<sub>/<sup>` tags.
- Understood the difference between semantic and non-semantic formatting tags.
- For example, `<strong>` and `<em>` convey meaning and are better for accessibility compared to `<b>` and `<i>`, which are purely visual.
- Real-world applications were demonstrated such as marking text as updated or deleted in blog posts using `<ins>` and `<del>`.
- Used `<sub>` and `<sup>` to format scientific and mathematical notations like H_2O and $E = mc^2$.
- The class participated in a collaborative editing session where each student formatted a sample biography using at least 6 formatting tags.

> *Day 1 wrapped up with small exercises that included building a basic profile page using all formatting and style tags learned.*

The profile included:

- A personal introduction using heading and paragraph tags.

- Basic styling using the `style` attribute to colorize and format the text.
- Emphasis tags to highlight important words like skills or achievements.
- Link to a favorite website using the `<a>` tag and `href` attribute.
- A list of hobbies using `<ul>` and `<li>` elements.

The exercises helped reinforce each concept by putting theory into practice. Students received individual feedback from the trainer, which helped clarify doubts and fine-tune their HTML structure. This hands-on approach made the session engaging and helped solidify the foundational concepts of web development.

By the end of the session, everyone felt confident in creating simple static web pages and understood the importance of clean and semantic HTML. This day set the tone for upcoming sessions which would cover CSS, JavaScript, and more advanced web development frameworks.

BY: Ekamjot Kaur

URN 2302867

CRN 2315264