Day 3 of training

Organizing Content with Tables and Containers

1. **Tables in HTML**
    - Tables are used to display data in a structured, grid-like format of rows and columns.
    - **Core Table Tags:**
        - **`<table>`:** The main container for the entire table.
        - **`<tr>` (Table Row):** Defines a row in the table.
        - **`<td>` (Table Data):** Defines a standard data cell within a row.
        - **`<th>` (Table Header):** Defines a header cell. Text in a `<th>` is typically bold and centered by default. It's semantically important to use this for column or row headers.
    - **Semantic Table Structure:** For more complex tables, it's good practice to use these structural tags to define the header, body, and footer sections of the table:
        - **`<thead>`:** Groups the header content (usually the row with the `<th>` elements).
        - **`<tbody>`:** Groups the main body content of the table.
        - **`<tfoot>`:** Groups the footer content.
2. **Spanning Rows and Columns**
    - We learned how to make cells span across multiple rows or columns, which is essential for creating more complex table layouts.
    - **`colspan` Attribute:** Used on a `<td>` or `<th>` to make a cell span across multiple **columns**. For example, `colspan="2"` makes a cell take up the width of two columns.
    - **`rowspan` Attribute:** Used to make a cell span across multiple **rows**. For example, `rowspan="3"` makes a cell take up the height of three rows.
3. **The `<div>` Tag - The Generic Container**
    - The `<div>` tag is one of the most important and commonly used tags in HTML. It's a **generic block-level container** that has no semantic meaning on its own.
    - Its main purpose is to **group related elements together** so they can be styled with CSS or manipulated with JavaScript as a single unit. It's used to create divisions or sections on a webpage. For example, you might wrap a heading and a few paragraphs in a `<div>` to create a "card" component.
4. **Block vs. In-line Elements**
    - This was a very important concept. HTML elements generally fall into one of two categories:
    - **Block-level Elements:** These elements always start on a new line and take up the full width available to them. Examples include `<div>`, `<p>`, `<h1>`-`<h6>`, `<ul>`, and `<li>`.
    - **In-line Elements:** These elements do not start on a new line and only take up as much width as necessary. They can sit side-by-side with other in-line elements. Examples include `<a>`, `<img>`, and `<span>`. The `<span>` tag is the

in-line equivalent of a `<div>`; it's a generic container used to group in-line elements for styling.