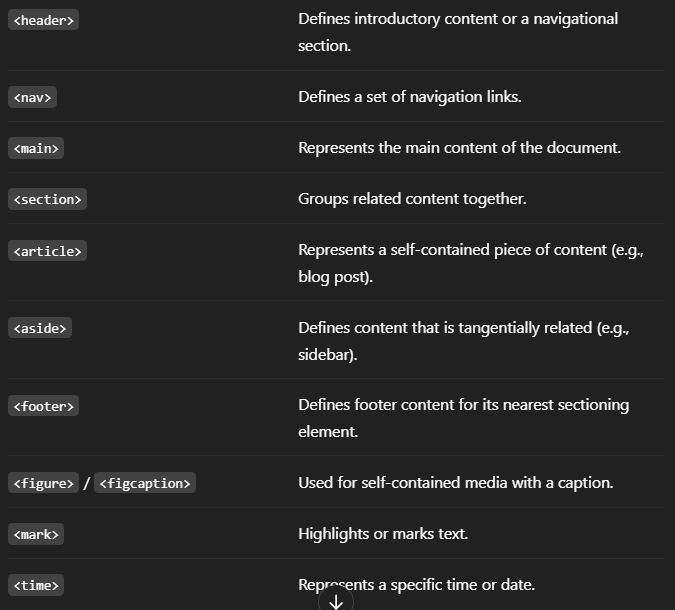
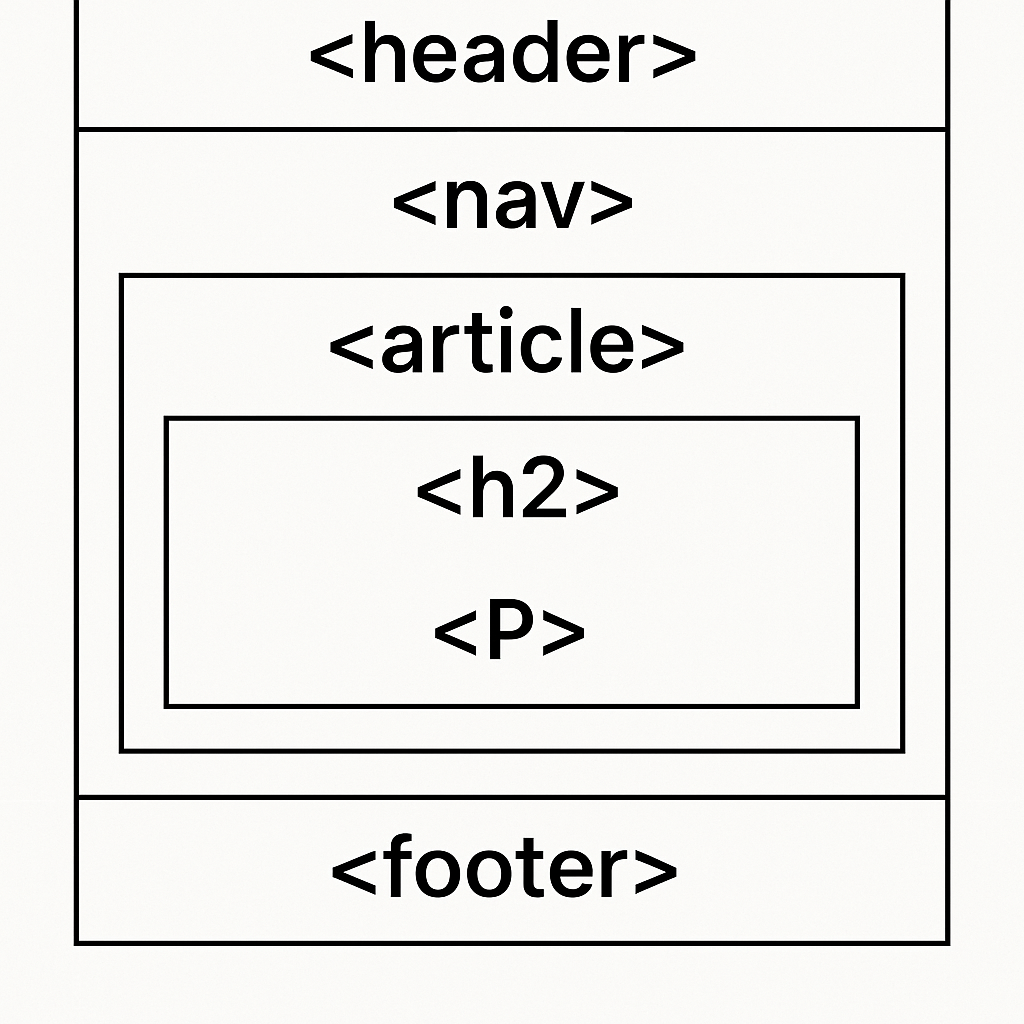
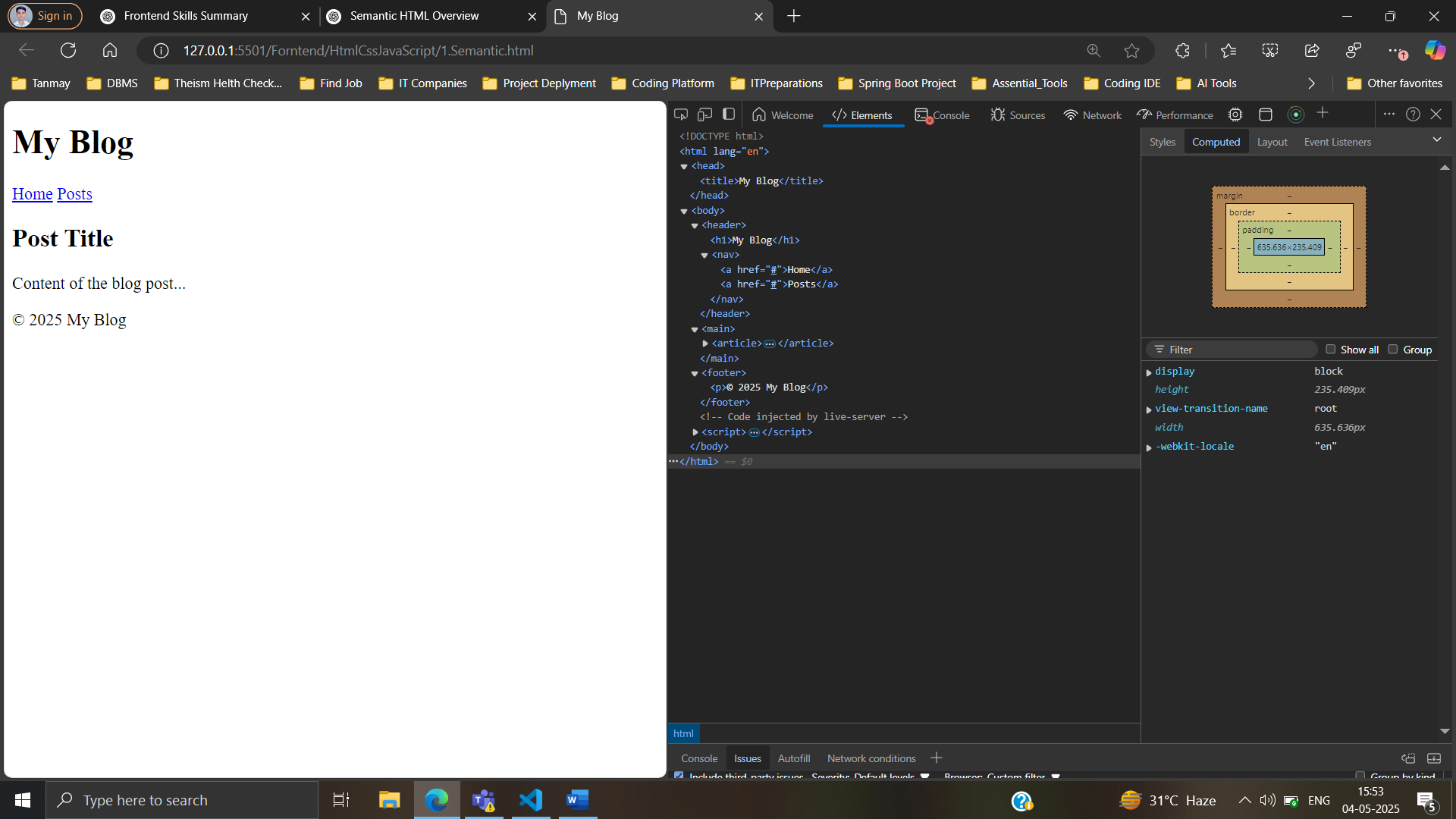
Semantic HTML: **Semantic HTML** refers to the use of HTML tags making the structure of a web page more understandable both to browsers and developers (as well as assistive technologies like screen readers).

**Why Semantic HTML Is Important:**

* Helps screen readers interpret the page correctly.
* **Search** engines understand your content better.
* Easier for developers to read and maintain.
* Standard structure across web projects.







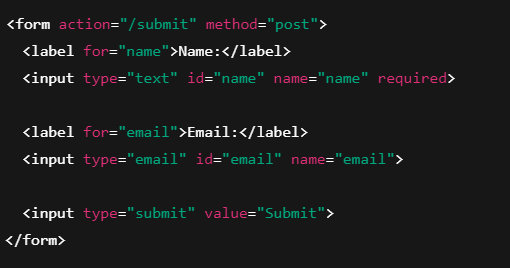
**🚫 Non-Semantic Tags (Less Meaningful):**

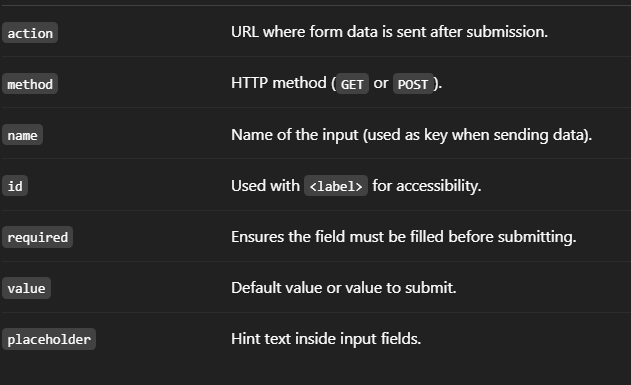
* <div>
* <span>

These are used for generic containers when no semantic meaning is required.

**Form:**

In HTML, **forms** are used to collect user input. They are essential for interactions like signing up, logging in, submitting feedback, etc.





**📊 HTML Tables**

HTML **tables** are used to display data in rows and columns, similar to spreadsheets.

**🧱 CSS Layout: Overview & Techniques**

CSS **layout** is about arranging elements on a page. Modern layout techniques help build responsive, flexible, and visually appealing websites.

**📚 Main CSS Layout Techniques:**

| **Layout Method** | **Description** |
| --- | --- |
| **Normal Flow** | Default document layout (block and inline elements). |
| **Float** | Old method for side-by-side content. |
| **Flexbox** | 1D layout – best for rows or columns. |
| **Grid** | 2D layout – best for complex layouts. |
| **Positioning** | Manual placement of elements (absolute, fixed, etc.). |
| **Display** | block, inline, inline-block, none, etc. |

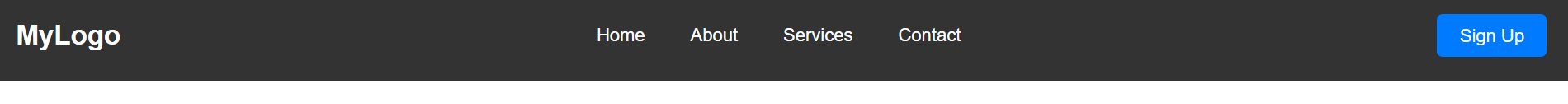
**🎯 CSS Positioning Explained**

CSS **positioning** allows you to control the layout of elements on a web page precisely.

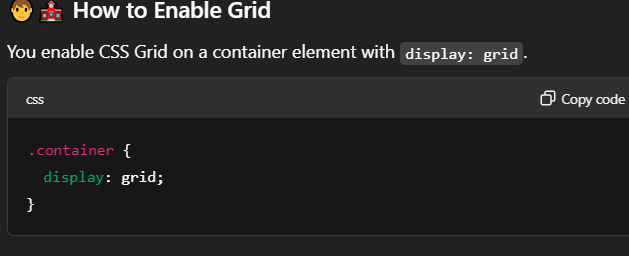
|  |
| --- |
| Manual placement of elements (absolute, fixed, etc.). |

**Flexbox in CSS (Flexible Box Layout)**

**Flexbox is a CSS layout model that allows you to design responsive layout structures without using floats or positioning.**

****

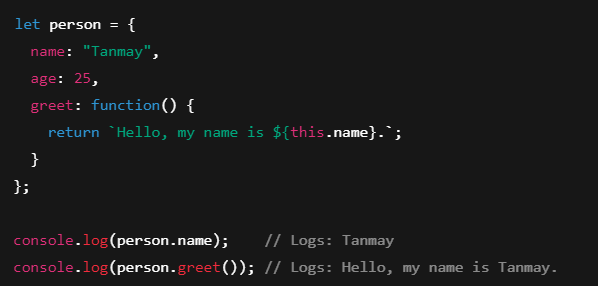
**CSS Grid** is another powerful layout system in CSS that allows you to create complex two-dimensional layouts with rows and columns. It’s more flexible than Flexbox when you need to work with both axes (horizontal and vertical) at the same time.



**Objects**

Objects are collections of key-value pairs and are used to represent real-world entities with properties and methods.

**Creating an Object:**

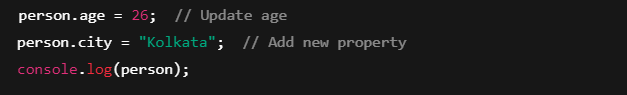


**Accessing Object Properties:**

You can access object properties using dot notation or bracket notation.



**Adding/Updating Object Properties:**



**🖱️ Event Handling in JavaScript**

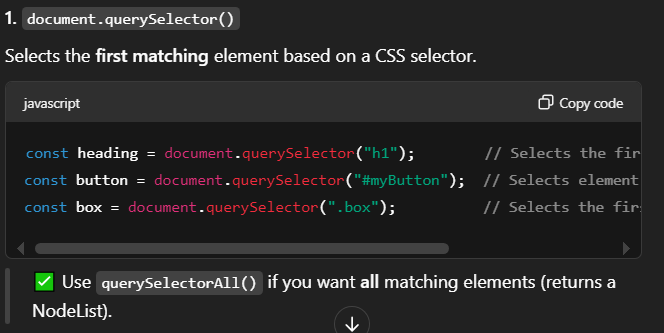
Event handling in JavaScript allows you to execute code when a user interacts with elements on a webpage, such as clicking a button, typing in an input field, or submitting a form.

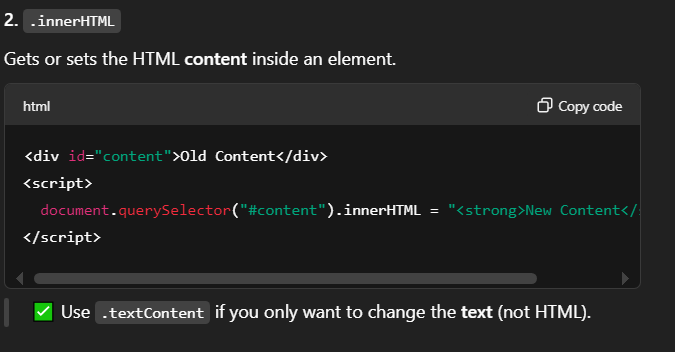
**1. Click Event**

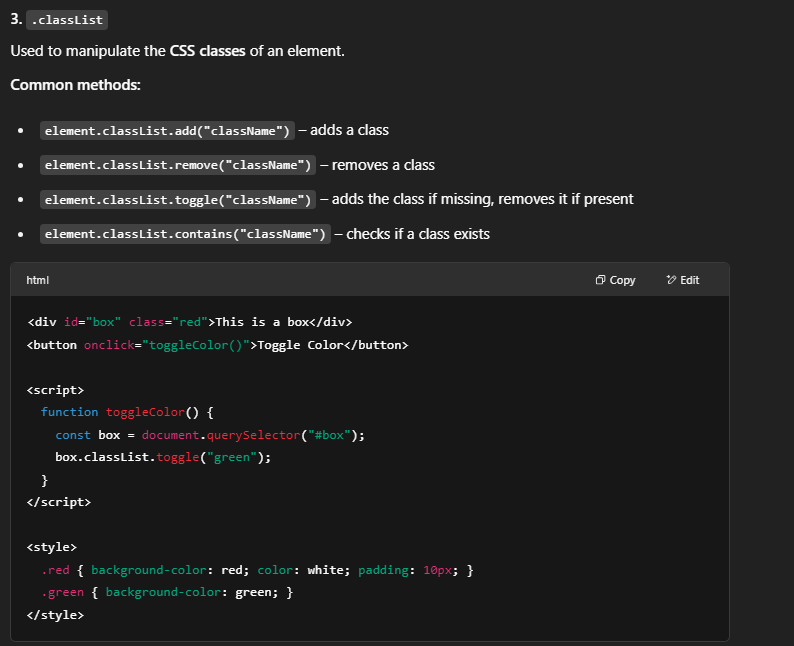
The **click event** is triggered when a user clicks on an element like a button or a link.

**DOM Manipulation:**

DOM (Document Object Model) manipulation allows you to dynamically access and change the content, structure, and style of a web page using JavaScript.







**3. React.js (Components, Props/State, Hooks, Lifecycle)**

**Topics:**

* Functional components
* useState, useEffect
* Props passing and lifting state
* Component lifecycle using useEffect
* Form handling and conditional rendering

**Project:** *Weather App (React)*

* Use useState for city input and weather data.
* Fetch data using useEffect.
* Props for reusable components (e.g., WeatherCard).