Day 18

Understanding Events and Event Handling in JavaScript

Events and event handling are crucial aspects of creating interactive web applications. They allow you to respond to user actions and other occurrences on a webpage, such as clicks, key presses, and page loading.

1. Events:

- **Purpose:** Events represent actions or occurrences that happen in the browser, often triggered by user interactions or other browser actions. Examples include mouse clicks, keyboard presses, and page load events.
- **Types:** Events can be categorized into various types, such as user interactions (e.g., mouse clicks, form submissions), and system events (e.g., page load, network requests).

2. Event Listeners:

- **Purpose:** Event listeners are functions that are set up to execute when a specific event occurs on a particular element. They allow you to define what should happen when an event is triggered.
- **Usage**: You attach an event listener to an HTML element to monitor for a specific type of event. When the event occurs, the associated function (event handler) is executed, allowing you to respond to the event.

Common Events

1. Click Event:

- **Purpose**: Triggered when a user clicks on an element, such as a button or link. It is commonly used to handle user interactions like submitting forms or navigating between pages.

2. Mouseover Event:

- **Purpose:** Occurs when the mouse pointer hovers over an element. It is often used for creating interactive effects, such as showing tooltips or changing element styles when a user moves the cursor over an element.

3. Keydown Event:

- **Purpose:** Fired when a user presses a key on the keyboard. This event is useful for implementing keyboard shortcuts or responding to user input in form fields.

4. Load Event:

- **Purpose:** Triggered when the browser has fully loaded a webpage or specific elements like images and scripts. It is commonly used to execute code that depends on the complete loading of a page or resources.