JavaScript

Module 4 - Functions and Event Handling

Chapter 8 - Event Handling

Aitrich

Events

- Events are actions or occurrences that happen in a web browser or web page, triggered by user interactions, browser actions, or system changes.

Examples:

- User clicks a button (click event)
- A page finishes loading (load event)
- The mouse moves over an element (mouseover event)
- A form is submitted (submit event)
- A key is pressed (keydown event)
- The browser window is resized (resize event)

Event Handling

- ➢ Event handling is the process of responding to events in JavaScript.
- ➢ It involves attaching event handlers (functions) to specific events.
- ➢ When those events occur, the designated code is executed.

Steps:

1. Identifying the Event: Determine which event you want to respond to (e.g., click, load, submit).
2. Attaching an Event Handler: Use one of these methods to assign a function to handle the event.
3. Writing the Event Handler Function: Create the function that will contain the code to be executed when the event happens.

Event Handlers

- Attaching event handlers: You can attach event handlers to HTML elements using two primary methods:

1. Inline event attributes

2. addEventListener() method

Event Handlers

Inline event attributes:

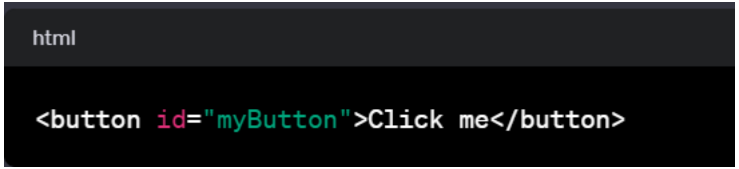
- These are attributes directly added to HTML elements, such as onclick, onmouseover, onload, etc.
- Syntax: <element onevent="handlerFunction()">
- Example: <button onclick="alert('Button clicked!')">Click me</button>

addEventListener() method:

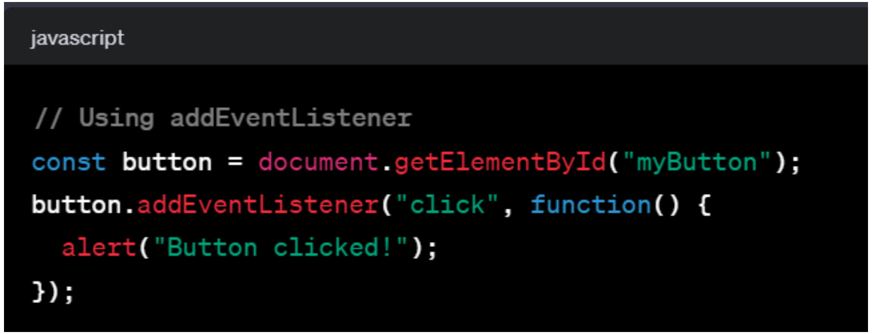
- This method is considered more flexible and modern.
- It allows you to attach multiple handlers to the same event, control event bubbling, and remove handlers later.
- Syntax: element.addEventListener(event, handlerFunction);
- Example: button.addEventListener('click', () => alert('Button clicked!'));

Responding to user interactions (clicks, mouse events).

Click Events:



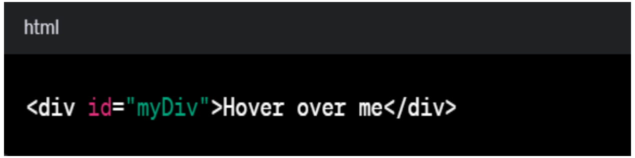
```html
<button id="myButton">Click me</button>
```



```javascript
// Using addEventListener
const button = document.getElementById("myButton");
button.addEventListener("click", function() {
  alert("Button clicked!");
});
```

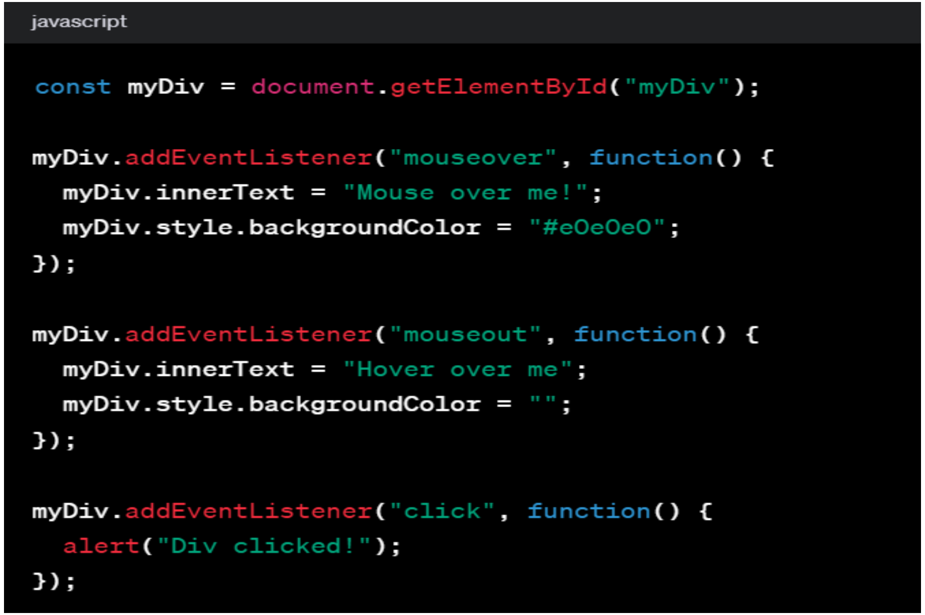
Responding to user interactions (clicks, mouse events).

Mouse Events:



html

```html
<div id="myDiv">Hover over me</div>
```



javascript

```javascript
const myDiv = document.getElementById("myDiv");

myDiv.addEventListener("mouseover", function() {
  myDiv.innerText = "Mouse over me!";
  myDiv.style.backgroundColor = "#e0e0e0";
});

myDiv.addEventListener("mouseout", function() {
  myDiv.innerText = "Hover over me";
  myDiv.style.backgroundColor = "";
});

myDiv.addEventListener("click", function() {
  alert("Div clicked!");
});
```

Thank You