

Day 26



Handling Browser Events:

What does “handling an event” mean?

Handling an event means:

1. Detecting when something happens (click, key press, mouse move, etc.)
2. Running code in response.

Example:

“User clicks a button → show a message”

Three Ways to Handle Events

In JavaScript, events can be handled in 3 ways:

Method 1 → HTML Event Attribute (old method)

You put the event directly in HTML.

Example:

```
<button onclick="alert('Button clicked!')>Click</button>
```

Cons: Not recommended now. Mixes HTML & JS.

Method 2 → Assign event to element property

Simple and clean.

Example:

```
<button id="btn">Click</button>
```

```
<script>
document.getElementById("btn").onclick = function () {
    alert("Clicked!");
};

</script>
```

Limitation:

You can assign **only ONE function** to .onclick.

Method 3 → addEventListener (best method)

Most modern and flexible method.

Example:

```
<button id="btn">Click</button>

<script>

document.getElementById("btn").addEventListener("click", () => {

    alert("Button clicked using addEventListener!");

});

</script>
```

Advantages:

- You can attach multiple event handlers
- Works with all event types
- Better for real projects

Also,

Removing Event Listeners

You can remove an event only if it is stored in a function.

```
5  remote_event.html > ...
1   <button id="btn">Click</button>
2
3   <script>
4     function sayHello() {
5       alert("Hello");
6     }
7
8     btn.addEventListener("click", sayHello);
9
10    // Remove the event
11    btn.removeEventListener("click", sayHello);
12  </script>
```

--The End--