🌟 5. Event Handling – Complete Notes

5.1 Event Basics

- **Event:** কোনো action যা browser বা user trigger করে (click, submit, hover ইত্যাদি)
- **Add Event:**
    1. Inline → `<button onclick="alert('hi')">Click</button>`
    2. JS → `element.addEventListener("click", function(){})`
- **Best practice:** `addEventListener()` ব্যবহার করা, multiple listener সহজে manage করা যায়

5.2 Event Object

- `event.target` → যেই element trigger করেছে
- `event.currentTarget` → যেই element listener আছে
- `event.type` → event type (click, keydown…)
- Mouse events → `clientX`, `clientY`
- Keyboard events → `key`, `code`

5.3 Different Event Types

- **Mouse:** click, dblclick, mouseover, mouseout, mousemove, contextmenu
- **Keyboard:** keydown, keyup, keypress
- **Form:** submit, change, input, focus, blur
- **Window/Document:** load, resize, scroll, DOMContentLoaded

---

5.4 Multiple Listeners & Removing Listeners

- এক element-এ multiple listener attach possible
- Remove listener:

```
function toggle(){ console.log("hi"); }
btn.addEventListener("click", toggle);
btn.removeEventListener("click", toggle);
```

5.5 Event Flow (Bubbling & Capturing)

- **Bubbling:** target → parent → grandparent
- **Capturing:** grandparent → parent → target
- Control:
  - `element.addEventListener("click", fn, true)` → capturing
  - `event.stopPropagation()` → event stop

5.6 Event Delegation

- Parent element-এ listener বসানো
- `event.target` দিয়ে child handle করা
- Dynamic elements handle করা সহজ
- Example:

```
ul.addEventListener("click", function(event){
  if(event.target.tagName === "LI"){
    event.target.classList.toggle("active");
  }
});
```

5.7 Prevent Default Behavior

- Default browser behavior বন্ধ করা `event.preventDefault()` দিয়ে
- Examples:
  - Form submit → reload বন্ধ
  - Link click → navigation block
  - Context menu block

5.8 Once, Passive, Options

- `once: true` → listener শুধু একবার run হবে
- `passive: true` → preventDefault use হবে না, scroll performance better

- `capture: true` → capturing phase এ listener fire হবে

5.9 Custom Events

- নিজস্ব event তৈরি: `new Event()` / `new CustomEvent()`
- `dispatchEvent()` → fire event
- `detail` → extra data পাঠানোর জন্য
- Example:

```
let greetEvent = new CustomEvent("greet", { detail: { name: "Touhid" } });
btn.addEventListener("greet", e => console.log(e.detail.name));
btn.dispatchEvent(greetEvent);
```

5.10 Advanced Use Cases

- **Keyboard shortcuts:** Ctrl+S, Shift+A → event.ctrlKey + event.key
- **Drag & Drop:** dragstart, dragover, drop
- **Debouncing / Throttling:** rapid event fire কমানো (scroll, resize)
- **Infinite Scroll:** scroll bottom → load more content
- **Form Validation:** preventDefault + JS validation

✅ Key Takeaways

1. Always prefer `addEventListener()` over inline events
2. Event delegation → efficient handling of many/dynamic elements
3. PreventDefault → control default browser actions
4. Once, passive, capture → event behavior control
5. Custom Events → communicate between JS modules
6. Advanced use → keyboard shortcuts, drag-drop, infinite scroll, form validation