

JavaScript Events - Interview Notes

1. What are Events?

Events in JavaScript are actions or occurrences that happen in the browser, such as a user clicking a button, pressing a key, or loading a page. JavaScript can 'listen' for these events and run code in response.

2. Event Flow (Capturing & Bubbling)

There are two main phases when an event happens:

- **Capturing phase:** Event travels from the root () down to the target element.
- **Bubbling phase:** Event travels back up from the target element to the root.

By default, event listeners listen in the bubbling phase. To listen in capturing, use `addEventListener(type, listener, true)`.

3. Important Event Methods

- **preventDefault():** Prevents the browser's default behavior (e.g., link navigation, form submit).
- **stopPropagation():** Stops the event from propagating further (bubbling/capturing).
- **stopImmediatePropagation():** Stops propagation AND prevents other listeners on the same element from running.

4. Commonly Asked Interview Questions

- Difference between bubbling and capturing?
- What is event delegation and why is it used?
- Difference between `preventDefault()` and `stopPropagation()`?
- What is the difference between `target` and `currentTarget`?
- What are synthetic events in React?

5. Event Delegation

Event delegation is a technique where you attach a single event listener to a parent element to handle events from its children. This works because of event bubbling.

Example:

```
document.getElementById("list").addEventListener("click", function(e) { if
(e.target.tagName === "LI") { console.log("Item clicked:", e.target.textContent); }
});
```

6. target vs currentTarget

• `e.target` → The actual element that triggered the event. • `e.currentTarget` → The element the event listener is attached to.

7. preventDefault() Examples

1. Prevent link navigation

```
document.querySelector("a").addEventListener("click", function(e) {
e.preventDefault(); console.log("Navigation blocked!"); });
```

2. Prevent form submission

```
document.getElementById("form").addEventListener("submit", function(e) {
e.preventDefault(); console.log("Form submit prevented!"); });
```

8. Removing Elements with parentNode

```
button.addEventListener("click", function(e) { let removeElement =  
  e.target.parentNode; removeElement.remove(); });
```

This removes the parent element of the clicked button.