

Event Delegation – JavaScript Interview Questions

1. What is Event Delegation?

Event Delegation is a technique where a single event listener is attached to a parent element to handle events for its child elements using event bubbling.

2. Why do we need Event Delegation?

It improves performance, reduces memory usage, handles dynamically added elements, and keeps the code clean and maintainable.

3. Which JavaScript event mechanism does Event Delegation use?

Event Delegation uses Event Bubbling, where events propagate from child elements to parent elements.

4. Can Event Delegation work with Event Capturing?

No. Event Delegation mainly relies on event bubbling, not capturing.

5. What is event.target in Event Delegation?

event.target refers to the actual element that triggered the event.

6. What is event.currentTarget?

event.currentTarget refers to the element on which the event listener is attached.

7. Difference between event.target and event.currentTarget?

event.target is the clicked element, while event.currentTarget is the element that handles the event listener.

8. How do you prevent Event Delegation from working?

By stopping event bubbling using event.stopPropagation().

9. When should you use Event Delegation?

When handling large lists, tables, menus, or dynamically generated elements.

10. When should you NOT use Event Delegation?

When events do not bubble, when only a few elements need listeners, or when strict control over child events is required.

11. Does Event Delegation work for all events?

No. Events like focus and blur do not bubble. Instead, focusin and focusout can be used.

12. How do you handle specific child elements in Event Delegation?

By checking class, tagName, id, or using methods like matches().

13. Write a simple Event Delegation example.

Attach one click listener to a parent element and use event.target to determine which child was clicked.

14. How does Event Delegation improve performance?

It reduces the number of event listeners, saving memory and improving execution speed.

15. Is Event Delegation related to addEventListener()?

Yes. Event Delegation is implemented using addEventListener() on a parent element.

16. What happens if you click on a nested child element?

The event bubbles up from the innermost element, and Event Delegation still works using `event.target`.

17. How do you handle nested elements in Event Delegation?

By using `closest()` to find the intended parent element.

18. Can Event Delegation cause bugs?

Yes, if `event.target` is not checked properly or clicks occur on unexpected nested elements.