

If you use `onclick` to set multiple event handlers on the same element, only the last assigned handler will work. This is because `onclick` can only store a single event handler function at a time. Each time you assign a new function to `onclick`, it overwrites any previously assigned function.

- **Multiple Event Handlers:** Allows multiple event handlers for the same event type on a single element. Each call to `addEventListener` adds a new event handler.

Comparison

Feature	<code>addEventListener</code>	<code>onclick</code>
Multiple Handlers	Supports multiple handlers	Only one handler, overwrites if reassigned
Removing Handlers	Can use <code>removeEventListener</code>	Cannot remove specific handlers
Event Phases	Supports capture and bubble phase options	No capture or bubble phase control
Flexibility	More flexible and powerful	Simpler but limited
Use Case	Preferred for complex and modern applications	Good for simple, single-handler scenarios

Example: Multiple Event Handlers That AddEventListener

javascript

```
const button = document.getElementById("myButton");

// First click event handler
button.addEventListener("click", function() {
    console.log("Handler 1: Button clicked!");
});

// Second click event handler
button.addEventListener("click", function() {
    console.log("Handler 2: Button clicked!");
});
```

to have multiple event handlers work simultaneously, you should use `addEventListener`

javascript

```
const button = document.getElementById("myButton");

// First event handler
button.addEventListener("click", function() {
    console.log("First handler: Button clicked!");
});

// Second event handler
button.addEventListener("click", function() {
    console.log("Second handler: Button clicked!");
});
```