

Below is a **rewritten, simplified, internally deep, and exam-ready version** of **JavaScript Events**, written in the **same format, depth, and clarity style** you've been asking for.

I have:

- Simplified language
- Explained internal working (**how + why**)
- Structured it for **college exams, viva, and interviews**
- Kept examples **short, meaningful, and clear**

🎯 JavaScript Events

1 What is an Event?

An **event** is an **action or occurrence** that happens in the browser and can be **detected by JavaScript**.

These actions can be triggered by:

- The **user**
- The **browser itself**

📌 JavaScript uses events to **interact with HTML pages**.

✓ Common Examples of Events

Action	Event
Page loads	load
Button click	click
Key pressed	keydown
Mouse moved	mousemove
Window resized	resize
Form submitted	submit
Input value changed	input

2 Why are Events Important?

Without events:

- Web pages would be **static**
- No user interaction possible

With events:

- Buttons respond
- Forms validate
- Pages update dynamically
- Apps become **interactive**

📌 Events are the backbone of dynamic web applications

3 Events and DOM (Internal Understanding)

Events are part of **DOM Level 3**.

🧠 Internal Flow

vbnet

User Action



Browser detects **event**



DOM creates **Event Object**



Event handler executes



JavaScript code runs

Every **HTML element** has:

- A predefined set of **events**
- Ability to trigger JavaScript code

4 What is an Event Handler?

An **event handler** is **JavaScript code that runs when an event occurs**.

It tells the browser:

“When THIS event happens, run THIS code.”

5 Using Event Handlers in HTML (Inline)

Syntax

html

```
<element onevent="JavaScript code">
```

✓ Example: Inline JavaScript

html

```
<button onclick="this.style.color='red'">  
Click Me  
</button>
```

🧠 Internal Explanation

- `onclick` → event handler
- `this` → refers to the `<button>`
- Style changes **only when clicked**

📌 Inline handlers execute **immediately when event fires**

6 Using Functions with Event Handlers (Recommended)

Instead of writing logic inline, we call functions.

✓ Example: Function-based Event Handling

html

```
<div id="box"></div>  
<button onclick="changeColor()">Change Color</button>  
  
<script>  
function changeColor() {  
  document.getElementById("box").style.backgroundColor = "blue";  
}  
</script>
```

🧠 Why this is better?

- Cleaner HTML
- Reusable logic
- Easy debugging
- Better readability

7 Multiple Functions on One Event

A single event can trigger **multiple functions**.

✓ Example

html

```
<div id="text" onmouseenter="increaseSize(); changeColor()">  
    Hello World  
</div>  
  
<script>  
function increaseSize() {  
    document.getElementById("text").style.fontSize = "25px";  
}  
  
function changeColor() {  
    document.getElementById("text").style.color = "red";  
}  
</script>
```

📌 One event → multiple responses

8 The JavaScript Event Object

Whenever an event occurs, JavaScript creates an **event object**.

This object contains:

- Event type
- Target element
- Mouse position
- Key pressed
- Timestamp

📌 Event Object Definition

The **event object** stores all information related to the triggered event.

Example

js

```
function handleClick(event) {  
    console.log(event.type); // click  
    console.log(event.target); // clicked element  
}
```

9 Event Object Hierarchy

All events inherit from a base object:

objectivec

Event

- |—— MouseEvent
- |—— KeyboardEvent
- |—— FocusEvent
- |—— InputEvent
- |—— TouchEvent
- └—— UIEvent

10 Types of JavaScript Event Objects

Event Object	Handles
MouseEvent	Click, hover, move
KeyboardEvent	Key press/release
FocusEvent	Focus/blur
InputEvent	Form input
DragEvent	Drag & drop
TouchEvent	Mobile touch
WheelEvent	Mouse wheel
ClipboardEvent	Copy/paste
StorageEvent	localStorage/sessionStorage
TransitionEvent	CSS transitions
AnimationEvent	CSS animations

11 Key Properties of Event Object (Exam Focus)

Property	Meaning
event.type	Event name
event.target	Element that triggered event

Property	Meaning
<code>event.currentTarget</code>	Element handling event
<code>event.timeStamp</code>	Time of event
<code>event.preventDefault()</code>	Stop default action
<code>event.stopPropagation()</code>	Stop event bubbling

1 | 2 Event Bubbling (Quick Concept)

By default, events **bubble upward**.

css

Button → Div → Body → Document

📌 This is why `event.target` is important.

1 | 3 One-Line Exam Answers

Q: What is an event?

👉 An action triggered by the user or browser that JavaScript can respond to.

Q: What is an event handler?

👉 JavaScript code that executes when an event occurs.

Q: What is the event object?

👉 An object containing details about the triggered event.

Q: Are events part of DOM?

👉 Yes, DOM Level 3.

1 | 4 Key Takeaways

- ✓ Events make web pages interactive
- ✓ Handled using event handlers
- ✓ Each event creates an event object
- ✓ Events belong to DOM
- ✓ One event can trigger multiple functions

▶ What's Next?

Perfect continuation topics in same depth:

1 | addEventListener() (Modern Event Handling)

- 2 Event Bubbling vs Capturing**
- 3 Default Events & preventDefault()**
- 4 Keyboard & Mouse Events in detail**

👉 Just type “**next**” and we continue 🚀