

## Keyboard Events

**keydown:** Triggered when a key is pressed down.

Example:

```
document.addEventListener("keydown", e => console.log("Key down:", e.key));
```

**keyup:** Triggered when a key is released.

Example:

```
document.addEventListener("keyup", e => console.log("Key up:", e.key));
```

**keypress:** Deprecated. Similar to keydown but only for character keys.

Example:

```
document.addEventListener("keypress", e => console.log("Key press:", e.key));
```

## Form Events

**submit:** Fired when a form is submitted.

Example:

```
form.addEventListener("submit", e => { e.preventDefault(); console.log("Form submitted"); });
```

**change:** Fired when input value changes and loses focus.

Example:

```
input.addEventListener("change", e => console.log("Changed to:", e.target.value));
```

**input:** Fired every time the value of an input changes.

Example:

```
input.addEventListener("input", e => console.log("Typing:", e.target.value));
```

**focus:** Triggered when the input gets focus (does not bubble).

Example:

```
input.addEventListener("focus", () => console.log("Focused"));
```

**blur:** Triggered when the input loses focus (does not bubble).

Example:

```
input.addEventListener("blur", () => console.log("Blurred"));
```

**focusin:** Like focus but bubbles.

Example:

```
form.addEventListener("focusin", () => console.log("Focus in"));
```

**focusout:** Like blur but bubbles.

Example:

```
form.addEventListener("focusout", () => console.log("Focus out"));
```

**reset: Fired when a form is reset.**

Example:

```
form.addEventListener("reset", () => console.log("Form reset"));
```

## Window Events

**load: Triggered when the page fully loads.**

Example:

```
window.addEventListener("load", () => console.log("Page loaded"));
```

**resize: Fired when the window is resized.**

Example:

```
window.addEventListener("resize", () => console.log("Window resized"));
```

**scroll: Fired when scrolling occurs.**

Example:

```
window.addEventListener("scroll", () => console.log("Scrolling"));
```

**unload: Fired when the page is unloading.**

Example:

```
window.addEventListener("unload", () => console.log("Unloading"));
```

**beforeunload: Fired before the user leaves the page.**

Example:

```
window.addEventListener("beforeunload", (e) => { e.preventDefault(); e.returnValue = ""; });
```

## Clipboard Events

**copy: Triggered when content is copied.**

Example:

```
document.addEventListener("copy", () => console.log("Copied"));
```

**cut: Triggered when content is cut.**

Example:

```
document.addEventListener("cut", () => console.log("Cut"));
```

**paste: Triggered when content is pasted.**

Example:

```
document.addEventListener("paste", () => console.log("Pasted"));
```

## Drag and Drop Events

**drag: Fired continuously while the element is being dragged.**

Example:

```
element.addEventListener("drag", () => console.log("Dragging"));
```

**dragstart: Fired when drag starts.**

Example:

```
element.addEventListener("dragstart", () => console.log("Drag started"));
```

**dragend: Fired when drag ends.**

Example:

```
element.addEventListener("dragend", () => console.log("Drag ended"));
```

**dragenter: Fired when dragged item enters drop target.**

Example:

```
dropZone.addEventListener("dragenter", () => console.log("Drag entered"));
```

**dragover: Fired when dragged item is over a drop target.**

Example:

```
dropZone.addEventListener("dragover", e => e.preventDefault());
```

**dragleave: Fired when dragged item leaves drop target.**

Example:

```
dropZone.addEventListener("dragleave", () => console.log("Drag left"));
```

**drop: Fired when dragged item is dropped.**

Example:

```
dropZone.addEventListener("drop", e => { e.preventDefault(); console.log("Dropped"); });
```

## Touch Events

**touchstart: Fired when touch starts.**

Example:

```
element.addEventListener("touchstart", () => console.log("Touch start"));
```

**touchend: Fired when touch ends.**

Example:

```
element.addEventListener("touchend", () => console.log("Touch end"));
```

**touchmove: Fired when finger moves.**

Example:

```
element.addEventListener("touchmove", () => console.log("Touch move"));
```

**touchcancel: Fired when touch is interrupted.**

Example:

```
element.addEventListener("touchcancel", () => console.log("Touch canceled"));
```

## Media Events

**play:** Triggered when media starts playing.

Example:

```
video.addEventListener("play", () => console.log("Playing"));
```

**pause:** Triggered when media is paused.

Example:

```
video.addEventListener("pause", () => console.log("Paused"));
```

**ended:** Triggered when media ends.

Example:

```
video.addEventListener("ended", () => console.log("Ended"));
```

**volumechange:** Triggered when volume changes.

Example:

```
video.addEventListener("volumechange", () => console.log("Volume changed"));
```

**timeupdate:** Fired periodically as playback position changes.

Example:

```
video.addEventListener("timeupdate", () => console.log("Time updated"));
```

**loadeddata:** Fired when media data is loaded.

Example:

```
video.addEventListener("loadeddata", () => console.log("Data loaded"));
```

## Animation & Transition Events

**animationstart:** Fired when an animation starts.

Example:

```
element.addEventListener("animationstart", () => console.log("Animation started"));
```

**animationend:** Fired when animation ends.

Example:

```
element.addEventListener("animationend", () => console.log("Animation ended"));
```

**animationiteration:** Fired each time animation repeats.

Example:

```
element.addEventListener("animationiteration", () => console.log("Animation repeated"));
```

**transitionend:** Fired when CSS transition ends.

Example:

```
element.addEventListener("transitionend", () => console.log("Transition ended"));
```

## Miscellaneous Events

**wheel: Fired when mouse wheel is used.**

Example:

```
window.addEventListener("wheel", () => console.log("Wheel event"));
```

**pointerdown: Pointer (mouse, touch) pressed down.**

Example:

```
element.addEventListener("pointerdown", () => console.log("Pointer down"));
```

**pointerup: Pointer released.**

Example:

```
element.addEventListener("pointerup", () => console.log("Pointer up"));
```

**pointermove: Pointer moves.**

Example:

```
element.addEventListener("pointermove", () => console.log("Pointer move"));
```

**error: Fired when a script or media file fails to load.**

Example:

```
window.addEventListener("error", () => console.log("Error occurred"));
```

**DOMContentLoaded: Fired when initial HTML is loaded and parsed.**

Example:

```
document.addEventListener("DOMContentLoaded", () => console.log("DOM loaded"));
```

## Mouse Events

**mouseenter: Fired when the mouse enters the element (does not bubble).**

Example:

```
element.addEventListener("mouseenter", () => console.log("Mouse entered"));
```

**mouseleave: Fired when the mouse leaves the element (does not bubble).**

Example:

```
element.addEventListener("mouseleave", () => console.log("Mouse left"));
```

**mouseover: Fired when the mouse is over the element or its child (bubbles).**

Example:

```
element.addEventListener("mouseover", () => console.log("Mouse over"));
```

**mouseout: Fired when the mouse leaves the element or its child (bubbles).**

Example:

```
element.addEventListener("mouseout", () => console.log("Mouse out"));
```

**mousemove:** Fired repeatedly as the mouse moves within the element.

Example:

```
element.addEventListener("mousemove", () => console.log("Mouse moving"));
```

**contextmenu:** Fired on right-click.

Example:

```
element.addEventListener("contextmenu", e => { e.preventDefault(); console.log("Right-clicked"); });
```