## **Syntax**

element.addEventListener(eventType, listenerFunction, options);

#### **Mouse Events**

"click" - when mouse clicks an element

"dblclick" - double-click

"mousedown" - mouse button pressed

"mouseup" - mouse button released

"mouseenter" - mouse enters element (no bubbling)

"mouseleave" - mouse leaves element (no bubbling)

"mousemove" - mouse moves within the element

"mouseover" - mouse moves onto the element (bubbles)

"mouseout" - mouse moves out of the element (bubbles)

"contextmenu" - right-click

# **Keyboard Events**

"keydown" - key is pressed

"keyup" - key is released

"keypress" - key is pressed (deprecated)

#### **Form Events**

"submit" - form submission

"change" - input changes

"input" - input value modified

"focus" - input is focused (no bubble)

"blur" - input loses focus (no bubble)

"focusin" - input is focused (bubbles)

"focusout" - input loses focus (bubbles)

"reset" - form reset

# Window Events

"load" - page finishes loading
"resize" - window resized

"scroll" - window or element scrolls

"unload" - page is unloaded

"beforeunload" - warning before leaving the page

# **Clipboard Events**

"copy"

"cut"

"paste"

## **Drag and Drop Events**

"drag"

"dragstart"

"dragend"

"dragenter"

"dragover"

"dragleave"

"drop"

### **Touch Events**

"touchstart"

"touchend"

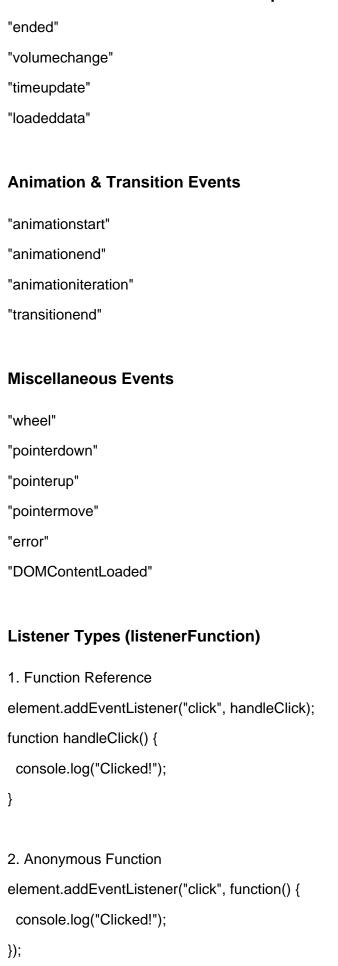
"touchmove"

"touchcancel"

### **Media Events**

"play"

"pause"



```
3. Arrow Function
element.addEventListener("click", () => {
  console.log("Clicked!");
});

4. Object with handleEvent Method
const listenerObj = {
  handleEvent(event) {
    console.log("Event type:", event.type);
  }
};
element.addEventListener("click", listenerObj);
```

# **Options (Third Parameter)**

The third argument can be a boolean or an object:

```
element.addEventListener("click", listener, {
  once: true,  // runs only once
  capture: false,  // use capture phase (true) or bubbling (false)
  passive: true  // tells browser the listener won't call preventDefault()
});
```