



# Manejo de Eventos en JavaScript

[Añadir Eventos](#)

[Tipos Comunes de Eventos](#)

[Propiedades del Objeto `Event`](#)

[Delegación de Eventos](#)

[Eventos Personalizados](#)

[Fases de Propagación del Evento](#)

[Ejemplo de Modo Captura:](#)

[Ejemplo de Modo Burbuja \(predeterminado\):](#)

[Prevención de Comportamientos Predeterminados](#)

[Ejemplo de uso:](#)

[Eventos de Teclado](#)

[Eventos de Ratón](#)

## Añadir Eventos

Método/Propiedad	Descripción	Ejemplo
<code>element.addEventListener(evento, función)</code>	Añade un manejador de eventos al elemento.	<code>button.addEventListener("click", manejarClick)</code>
<code>element.removeEventListener(evento, función)</code>	Elimina un manejador de eventos previamente añadido.	<code>button.removeEventListener("click", manejarClick)</code>

## Tipos Comunes de Eventos

Evento	Descripción	Ejemplo
<code>click</code>	Disparado al hacer clic en un elemento.	<code>element.addEventListener("click", handler)</code>
<code>dblclick</code>	Disparado al hacer doble clic en un elemento.	<code>element.addEventListener("dblclick", handler)</code>
<code>mouseover</code>	Disparado cuando el puntero pasa sobre un elemento.	<code>element.addEventListener("mouseover", handler)</code>
<code>mouseout</code>	Disparado cuando el puntero deja un elemento.	<code>element.addEventListener("mouseout", handler)</code>

<code>keydown</code>	Disparado cuando una tecla es presionada.	<code>element.addEventListener("keydown", handler)</code>
<code>keyup</code>	Disparado cuando una tecla es soltada.	<code>element.addEventListener("keyup", handler)</code>
<code>submit</code>	Disparado cuando un formulario es enviado.	<code>form.addEventListener("submit", handler)</code>
<code>change</code>	Disparado cuando el valor de un elemento <code>&lt;input&gt;</code> , <code>&lt;textarea&gt;</code> o <code>&lt;select&gt;</code> cambia.	<code>input.addEventListener("change", handler)</code>
<code>focus</code>	Disparado cuando un elemento recibe foco.	<code>input.addEventListener("focus", handler)</code>
<code>blur</code>	Disparado cuando un elemento pierde el foco.	<code>input.addEventListener("blur", handler)</code>
<code>load</code>	Disparado cuando una página o recurso ha cargado completamente.	<code>window.addEventListener("load", handler)</code>
<code>resize</code>	Disparado cuando la ventana cambia de tamaño.	<code>window.addEventListener("resize", handler)</code>
<code>scroll</code>	Disparado cuando ocurre un desplazamiento (scroll) en un elemento.	<code>window.addEventListener("scroll", handler)</code>

## Propiedades del Objeto `Event`

Propiedad	Descripción	Ejemplo
<code>event.type</code>	El tipo de evento que fue disparado.	<code>console.log(event.type)</code>
<code>event.target</code>	El elemento que disparó el evento.	<code>console.log(event.target)</code>
<code>event.currentTarget</code>	El elemento al que se añadió el listener (puede ser diferente de <code>target</code> ).	<code>console.log(event.currentTarget)</code>
<code>event.preventDefault()</code>	Previene el comportamiento predeterminado del evento (ej. evitar que un formulario se envíe).	<code>event.preventDefault()</code>
<code>event.stopPropagation()</code>	Detiene la propagación del	<code>event.stopPropagation()</code>

	evento hacia otros elementos padre.	
<code>event.stopImmediatePropagation()</code>	Detiene la propagación y evita que otros listeners en el mismo elemento sean llamados.	<code>event.stopImmediatePropagation()</code>
<code>event.key</code>	Obtiene la tecla que fue presionada en un evento de teclado.	<code>console.log(event.key)</code>
<code>event.clientX</code> , <code>event.clientY</code>	Coordenadas X e Y del puntero dentro de la ventana cuando se dispara el evento de ratón.	<code>console.log(event.clientX, event.clientY)</code>
<code>event.pageX</code> , <code>event.pageY</code>	Coordenadas X e Y del puntero respecto al documento completo.	<code>console.log(event.pageX, event.pageY)</code>

## Delegación de Eventos

Técnica	Descripción	Ejemplo
<b>Delegación de eventos</b>	En lugar de añadir listeners a cada elemento, se añade a un contenedor común y se verifica el <code>event.target</code> .	<pre>document.body.addEventListener("click", function(event) { if (event.target.matches(".boton")) { manejarClick(); } });</pre>

## Eventos Personalizados

Método/Propiedad	Descripción	Ejemplo
<code>new CustomEvent(evento, { detalle })</code>	Crea un evento personalizado con datos extra.	<pre>const miEvento = new CustomEvent("miEvento", { detail: { nombre: "ejemplo" } });</pre>
<code>element.dispatchEvent(evento)</code>	Dispara (lanza) un evento personalizado.	<code>element.dispatchEvent(miEvento)</code>

## Fases de Propagación del Evento

Los eventos en JavaScript tienen **tres fases** principales de propagación:

1. **Captura** ( `capturing phase` ): Desde el nodo más alto del DOM hacia el nodo objetivo.
2. **Objetivo** ( `target phase` ): Cuando el evento llega al nodo objetivo.
3. **Burbuja** ( `bubbling phase` ): Desde el nodo objetivo hacia los nodos padres.

### Ejemplo de Modo Captura:

```
document.addEventListener("click", function(event) {  
    console.log("Captura");  
}, true); // El tercer parámetro en true indica captura
```

### Ejemplo de Modo Burbuja (predeterminado):

```
document.addEventListener("click", function(event) {  
    console.log("Burbuja");  
}); // Por defecto, los eventos utilizan bubbling
```

### Prevención de Comportamientos Predeterminados

- `event.preventDefault()` : Previene el comportamiento predeterminado (ej. no enviar formulario, no seguir un enlace).

### Ejemplo de uso:

```
document.querySelector("a").addEventListener("click", function(event) {  
    event.preventDefault(); // Evita que el enlace se abra  
    console.log("Enlace bloqueado");  
});
```

### Eventos de Teclado

Propiedad	Descripción	Ejemplo
<code>event.key</code>	La tecla presionada.	<code>console.log(event.key)</code>
<code>event.code</code>	El código físico de la tecla (independiente de la disposición del teclado).	<code>console.log(event.code)</code>
<code>event.altKey</code> , <code>event.ctrlKey</code> , <code>event.shiftKey</code>	Verifica si se presionaron las teclas modificadoras.	<pre>if (event.ctrlKey) {     console.log("Ctrl presionado"); }</pre>

## Eventos de Ratón

Propiedad	Descripción	Ejemplo
<code>event.clientX</code> , <code>event.clientY</code>	Coordenadas del puntero dentro de la ventana.	<pre>console.log(event.clientX, event.clientY)</pre>
<code>event.button</code>	Botón del ratón que fue presionado (0: izquierdo, 1: central, 2: derecho).	<pre>if (event.button === 2) {   console.log("Botón derecho"); } }</pre>