

# Eventos de JavaScript

Diseño Web II  
Undécimo Grado de Informática

## Introducción a los eventos de JavaScript

En el diseño y desarrollo web, los **eventos** son acciones que ocurren en el navegador o en la página web, y que el programador puede detectar para ejecutar un código. Piénsalo como una forma de hacer que una página web sea interactiva. Cuando un usuario interactúa con un sitio, como al hacer clic en un botón, pasar el mouse sobre un enlace o escribir en un campo de texto, está desencadenando eventos. JavaScript nos permite responder a estas interacciones para crear una experiencia de usuario dinámica.

## Conceptos clave

1. **Evento:** Es una acción que sucede en el navegador. Por ejemplo, un clic del ratón, una pulsación de tecla, la carga de una página o el envío de un formulario.
2. **Manejador de evento (Event Handler):** Es una función de JavaScript que se ejecuta cuando ocurre un evento específico. Su propósito es definir qué debe hacer la página en respuesta a la acción del usuario.
3. **Objeto de evento (Event Object):** Cuando un evento ocurre, el navegador crea un objeto que contiene información sobre dicho evento, como el tipo de evento, el elemento que lo desencadenó y la posición del cursor. Este objeto se pasa automáticamente como un argumento a la función del manejador de evento.

## Formas de manejar eventos

Existen varias maneras de asignar un manejador de evento a un elemento HTML:

### 1. Atributo HTML (Método desactualizado)

Este método consiste en añadir el código JavaScript directamente como un atributo en la etiqueta HTML. Aunque es sencillo, se considera una práctica obsoleta porque mezcla la estructura del HTML con la lógica del JavaScript, lo que dificulta el mantenimiento del código.

### Ejemplo:

```
<button onclick="alert('¡Hola desde un evento!')">Haz clic aquí</button>
```

## 2. Propiedades de los elementos (DOM Level 0)

En este enfoque, se asigna una función a una propiedad del elemento HTML. Cada evento tiene su propiedad correspondiente, como `onclick`, `onmouseover`, etc.

**Ejemplo:**

```
<button id="miBoton">Haz clic aquí</button>

<script>
  const boton = document.getElementById('miBoton');
  boton.onclick = function() {
    alert('¡Se ha presionado el botón!');
  };
</script>
```

## 3. `addEventListener()` (Método recomendado)

Este es el método más moderno y recomendado. Permite adjuntar múltiples manejadores de evento a un solo elemento para el mismo tipo de evento, sin sobrescribir los anteriores. Su sintaxis es:

`elemento.addEventListener(evento, funcion, [opciones])`

- `evento`: El nombre del evento (sin el prefijo `on`, por ejemplo, `'click'` en lugar de `'onclick'`).
- `funcion`: La función que se ejecutará cuando ocurra el evento.
- `opciones`: Un objeto opcional que puede tener propiedades como `once` (el evento solo se ejecutará una vez) o `capture` (para el flujo del evento).

**Ejemplo:**

```
<button id="miBoton">Haz clic aquí</button>

<script>
  const boton = document.getElementById('miBoton');

  // Primer manejador de evento
  boton.addEventListener('click', function() {
    alert('¡Primer mensaje!');
  });

  // Segundo manejador de evento para el mismo evento
  boton.addEventListener('click', function() {
    console.log('¡Segundo mensaje en la consola!');
  });
</script>
```

## Eventos comunes en JavaScript

Tipo de Evento	Descripción	Ejemplo de Uso
Mouse	click, dblclick (doble clic), mouseover, mouseout	Clic en un botón, cambio de estilo al pasar el mouse.
Teclado	keydown (presionar tecla), keyup (soltar tecla)	Validación de formularios mientras se escribe.
Formulario	submit (enviar), focus (elemento activo), change	Enviar datos a un servidor, validar campos.
Ventana/Documento	load (página cargada), resize (cambio de tamaño)	Ejecutar código cuando la página se carga por completo.

### Ejemplo práctico: Un contador de clics

Intenta hacer este ejercicio en un editor de código, puedes hacerlo en tu teléfono celular en la aplicación “Code Studio” o en tu computadora.  
Este ejemplo ilustra cómo usar `addEventListener` para crear un simple contador de clics en un botón.

#### Código:

Archivo con nombre: Contador.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Contador de Clics</title>
</head>
<body>

  <h1>Contador de Clics</h1>
  <p>Clics realizados: <span id="contador">0</span></p>
  <button id="botonIncrementar">Incrementar</button>

  <script src="contador.js"></script>

</body>
</html>
```

archivo con nombre “contador.js”

```
// Elementos del HTML
const boton =
document.getElementById('botonIncrementar');
const spanContador = document.getElementById('contador');

let clics = 0; // Variable para almacenar el conteo

// manejador de evento para el evento 'click'
boton.addEventListener('click', function() {
  // Incrementamos la variable
  clics++;

  // nuevo valor
  spanContador.textContent = clics;

  // en la consola para depuración
  console.log(`¡Se ha hecho clic! El contador es:
  ${clics}`);
});
```

En este ejemplo, cuando el usuario hace clic en el botón, la función anónima dentro de `addEventListener` se ejecuta, aumentando el valor de la variable `clics` y actualizando el texto del `<span>`. Este proceso demuestra la interactividad que se puede lograr utilizando eventos.