

Eventos

Cordero Hernández, Marco R



En sesiones pasadas...



- Atributos y propiedades de elementos
- Modificación del documento
- Operaciones extendidas del DOM
- Introducción a **eventos**

01

Event Listeners

02

Objeto event

03

Event Bubbling

The background is a dark blue space-themed illustration. It features two large spiral galaxies, one in the top-left and one in the bottom-right, composed of concentric blue and yellow dots. Numerous small white stars and blue 'x' marks are scattered across the background. Three yellow comets with long tails are depicted: one in the top-right, one in the bottom-left, and one in the bottom-center.

01

EVENT LISTENERS

addEventListener

Dado que solo se puede tener un **handler** por elemento con las estrategias revisadas previamente, se debe emplear un mecanismo alternativo para agregar más de un evento. A esto se le conocerá como adición de **listeners**

- elemento.***addEventListener(evento, handler)***
- elemento.***removeEventListener(evento, handler)***

Idealmente se pasa un **callback** como argumento en el parámetro **handler**



addEventListener

```
<button id="btn5">Botón 5</button>
<script>
  // Event handler
  btn5.onclick = () => alert('Hola mundo 1');

  // Event listeners
  btn5.addEventListener('click', () => alert('Hola mundo 2'));
  btn5.addEventListener('click', funcionAdicional);
  // Observa que ya no se usa "on"

  function funcionAdicional() {
    alert('Hola mundo 3');
  }
</script>
```

Ejercicio

En el mismo HTML del ejercicio anterior (incrementar/decrementar) realiza lo siguiente:

- Crea una función que cuente la cantidad de clicks totales (la función solo incrementa un contador y lo muestra en otro párrafo)
- Añade esta función como ***eventListener*** a los botones de incrementar y decrementar (ambos)



02

**OBJETO
EVENT**

Objeto *event*

El objeto **event** tiene muchas propiedades sobre el evento ocurrido sobre un elemento, de manera que es posible conocer detalles de lo sucedido tales como:

- **event.type**: Indica qué tipo de evento (“click”, “change”, “keyup”, etc.)
- **event.target**: Regresa el elemento (de HTML) que primero ejecutó el evento
- **event.currentTarget**: Elemento que controla el evento (similar al *this* de un objeto)
- **event.clientX/clientY**: Coordenadas del cursor



03

**EVENT
BUBBLING**



Elementos anidados

- Si varios elementos anidados tienen un **handler** al mismo evento (ej. “click”), *todos los elementos padres (ancestros) también ejecutarán el evento.*

Caso de estudio

Supón que dentro de un *div* existe un *span* y dentro de este un *párrafo*. Si los 3 elementos tienen un **handler** al evento “click”, al dar click en el *párrafo* también se ejecutarán los eventos del *span* y posteriormente del *div*.



Elementos anidados

```
<div onclick="alert('hola 3')">  
  <span onclick="alert('hola 2')">  
    <p onclick="alert('hola 1')">  
      Texto  
    </p>  
  </span>  
</div>
```



Most deeply
nested element





Delegación de eventos

Para evitar la *propagación de eventos hacia arriba* es posible “atrapar” el evento en un contenedor padre y desde este mismo determinar qué hacer dependiendo de dónde ocurrió el evento. Para este propósito es posible usar **`event.target`**.

También se puede usar **`event.target.closest(selector de css)`** para encontrar el padre más cercano *que cumpla con el selector indicado*.

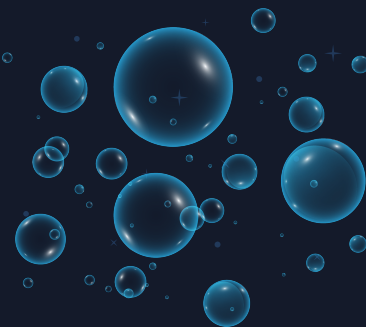


Delegación de eventos

```
<table id="tab1" border="1">
  <tr>
    <td>cell1</td>
    <td>cell2</td>
  </tr>
  <tr>
    <td>cell2</td>
    <td>cell4</td>
  </tr>
</table>
```

```
<script>
  tab1.onclick = function(event) {
    if (event.target.nodeName !== 'TD') return;

    // tr padre más cercano
    let row = event.target.closest('tr');
    alert(`Fila número ${row.rowIndex}`);
  }
</script>
```





Demo mousemove



X: 916

Y: 256





Ejercicio

Replica la siguiente vista junto con su funcionalidad

Tabla dinámica

Introduce el elemento a agregar aquí

Número

Nombre

Agregar

☐ Bloquear numeración

Número	Nombre
1	Marco
2	Ricardo
3	Cordero
4	Hernández