

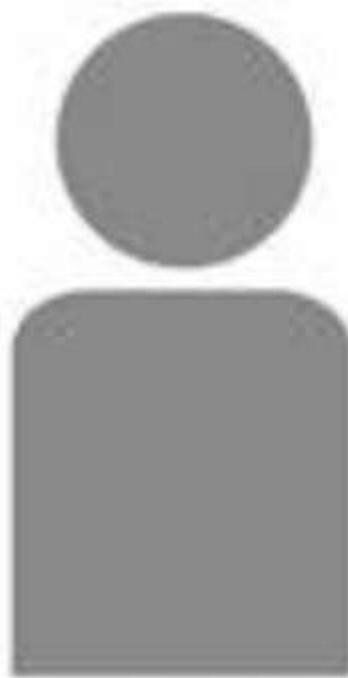
Programa académico **CAMPUS**

Trainer
Ing. Carlos H. Rueda C.



JAVASCRIPT

Fundamentos de Eventos



USUARIO



evento Javascript

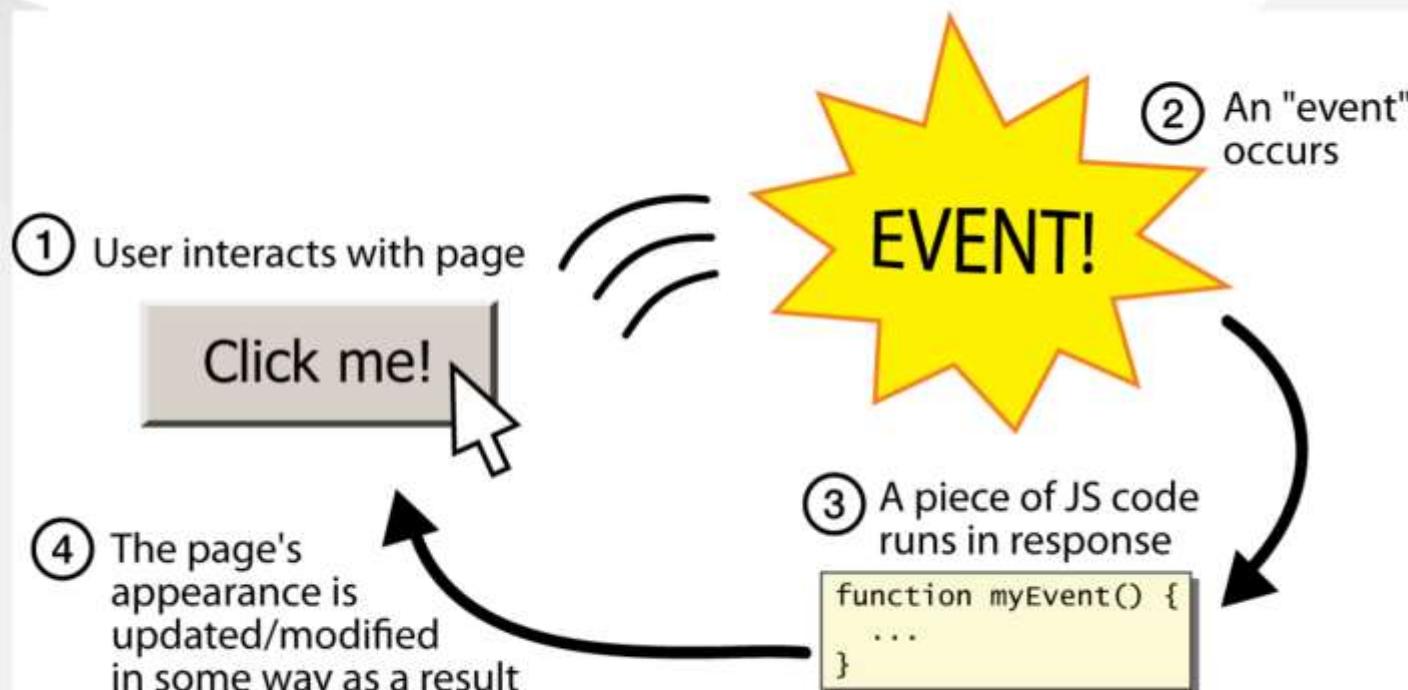


f(x)

función asociada

¿Qué son los eventos?

Los eventos son **notificaciones** que informan sobre acciones relevantes en una página web, como **clics**, desplazamientos, pulsaciones de teclas o reproducción de archivos multimedia.



Ejemplo práctico:

```
<button onClick="alert('¡Hola!')">Saludar</button>
```

```
const boton = document.querySelector("button");
boton.onclick = () => alert("¡Hola desde JavaScript!");
```

“ Los eventos permiten que una página reaccione dinámicamente a la interacción del usuario. ”

Eventos mediante HTML

2.1. Sintaxis General

```
<button onClick="alert('¡Hola!')">Saludar</button>
```

- Esta forma es básica pero mezcla **HTML** con **JavaScript**, lo que dificulta el mantenimiento.

Ejemplo con función:

```
<button onClick="saludar()">Saludar</button>

<script>
  function saludar() {
    alert("¡Hola desde una función!");
  }
</script>
```

2.2. Organizando la funcionalidad

La funcionalidad se puede separar para mejor mantenimiento.

Código en un archivo separado:

```
<button id="saludar">Saludar</button>
<script src="script.js"></script>

document.getElementById("saludar").addEventListener("click", () => {
  alert("¡Hola desde archivo separado!");
});
```

Uso de parámetros enfocado en eventos

Propiedad `.onclick`

```
<button id="boton">Saludar</button>
<script>
  const boton = document.getElementById("boton");
  boton.onclick = () => alert("¡Hola, mundo!");
</script>
```

Ventaja: Simple y directo.

.setAttribute()

Permite asignar dinámicamente atributos a los elementos.

```
<button id="boton">Saludar</button>
<script>
  const boton = document.getElementById("boton");
  boton.setAttribute("onclick", "alert('¡Evento dinámico!')");
</script>
```

“ Aunque funcional, **addEventListener** es más flexible y modular. ”

Otro Ejemplo

```
<style>
  body {
    transition: background-color 0.5s;
  }
</style>
<body>
  <button id="changeColorButton">Cambiar Color de Fondo</button>
</body>
```

Continuación

```
<script>
  const button = document.getElementById('changeColorButton');
  // Función para cambiar el color de fondo
  const changeBackgroundColor = () => {
    document.body.style.backgroundColor = getRandomColor();
  };
  // Asignación del evento utilizando setAttribute()
  button.setAttribute('onclick', 'changeBackgroundColor()');
  // Función para obtener un color aleatorio en formato hexadecimal
  function getRandomColor() {
    const letters = '0123456789ABCDEF';
    let color = '#';
    for (let i = 0; i < 6; i++) {
      color += letters[Math.floor(Math.random() * 16)];
    }
    return color;
  }
</script>
```

Ejemplo adicional - Uso de addEventListener

Asignar múltiples eventos al mismo elemento:

```
<button id="boton">Interactuar</button>
<script>
  const boton = document.getElementById("boton");

  boton.addEventListener("click", () => alert("Clic detectado!"));
  boton.addEventListener(
    "mouseover",
    () => (boton.style.backgroundColor = "lightblue")
  );
</script>
```

Doble clic en un elemento de texto

Detectar cuando el usuario realiza un doble clic en un texto.

```
<p id="texto">Haz doble clic aquí</p>
<script>
  const texto = document.getElementById("texto");
  texto.addEventListener("dblclick", () => {
    alert("¡Doble clic detectado!");
  });
</script>
```

Cambio en el contenido de un campo de texto

Detectar cambios en un campo de texto con el evento `input`.

```
<input type="text" id="campo" placeholder="Escribe aquí" />
<p id="resultado"></p>
<script>
  const campo = document.getElementById("campo");
  const resultado = document.getElementById("resultado");

  campo.addEventListener("input", () => {
    resultado.textContent = `Has escrito: ${campo.value}`;
  });
</script>
```

Pasar el mouse sobre una imagen

Cambiar el fondo de una imagen cuando el mouse pasa sobre ella.

```

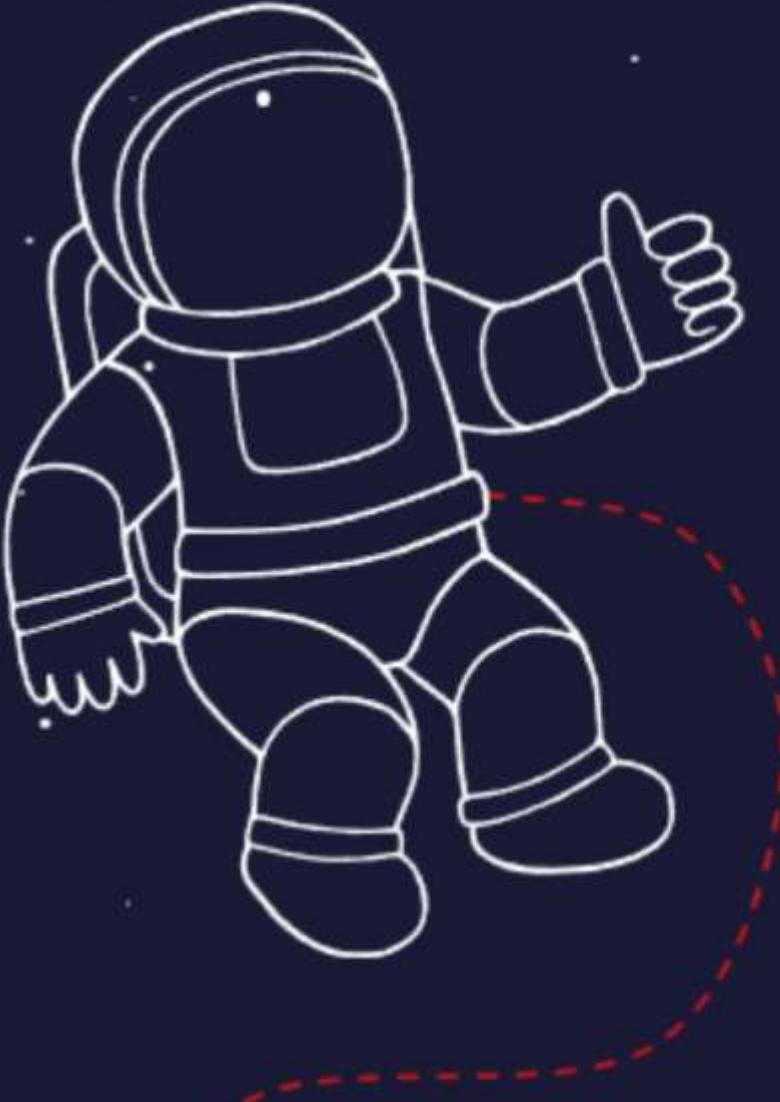
<script>
  const imagen = document.getElementById("imagen");

  imagen.addEventListener("mouseover", () => {
    imagen.style.border = "5px solid blue";
  });

  imagen.addEventListener("mouseout", () => {
    imagen.style.border = "none";
  });
</script>
```

Conclusión

- Evita usar eventos directamente en HTML.
- Prefiere `addEventListener` para una mayor modularidad.
- Separa la lógica de eventos en archivos **JavaScript** externos.

A white line-art illustration of an astronaut in a full spacesuit, including a helmet with a visor and a backpack. The astronaut is shown from the waist up, facing right and pointing their right index finger towards the text. They are set against a dark blue background with small white stars.

Programa académico **CAMPUS**

Trainer
Ing. Carlos H. Rueda C.

A white line-art illustration of a rocket ship. It has a pointed nose cone at the top, two fins on the side, and a base with a grid pattern. The rocket is angled upwards towards the top right of the frame, suggesting launch. It is positioned to the right of the 'Trainer' and 'Ing. Carlos H. Rueda C.' text.