

# PROGRAMACIÓN BÁSICA EN JAVASCRIPT



## Módulo 2





# CLASE 8

Introducción

Eventos

Tipos de Eventos

Varios elementos

Mouseover y Mouseleave

Argumentos de un Evento

Eventos disparados por el teclado

---

# Introducción

Hoy comenzaremos con los Eventos, algo sumamente necesario para darle vida a todos nuestros proyectos con JavaScript Vanilla.

Esta unidad no es extensa como las anteriores, de hecho, verán que cada vez son más cortas porque todo se concentra en la práctica que usted pueda tener.

Entonces es necesario que practique el doble de lo que lee, para poder asimilar la mayor cantidad de conocimiento posible.

Es así que arrancamos con la última parte del contenido de esta unidad.



- +
  - **Los Eventos son acciones que activan los cambios que realiza JavaScript en los sitios dinámicos. Gracias a esto, el DOM va a "escuchar" (listen) qué hace el usuario y activar cierta parte del código, en base a cierto estímulo.**
  -
- +
  - 
  -

# ¿Qué es un *evento*?

Por ejemplo, si un usuario hace click en un botón o presiona cierta tecla (estímulo), esto nos permitirá correr un código específico (como cambiar el contenido de un texto o cambiar el color del fondo).

Para agregar un Evento, usamos `addEventListener()`. Para eso:

1. Seleccionamos el elemento (por ejemplo, un botón).
2. Agregamos el **eventListener** que:
  - a. Responda frente a un estímulo (por ejemplo, cuando el usuario haga click).
  - b. Ejecute cierta parte del código (mostrar una alerta).

```
document.querySelector("button").addEventListener("click", function() {  
    console.log("¡Alguien hizo click en el botón!")  
})
```



## ¿Qué es un *evento*?

En este ejemplo, el primer argumento del método `addEventListener()` es un String ("click") que indica el tipo de Evento que se estará "escuchando". El segundo argumento es una Función denominada callback (en español, vendría a ser la "llamada de vuelta"). Es una Función que no corre inmediatamente, sino que depende de otra cosa (el Evento del click del usuario).

## Tipos de eventos

Al igual que con el DOM existen muchos métodos y propiedades para utilizar.

Por el momento veremos las más comunes.

Recuerda que siempre es necesario que vayas buscando el método o funcionalidad que necesite.

Evento	Descripción
<code>onchange</code>	Modificación de un elemento HTML
<code>onclick</code>	Usuario hace <i>click</i> sobre un elemento HTML
<code>onmouseover</code>	Usuario mueve el <i>mouse</i> sobre un elemento HTML
<code>onmouseout</code>	Usuario quita el <i>mouse</i> del elemento HTML
<code>onkeydown</code>	Usuario presiona una tecla
<code>onload</code>	<i>Browser</i> termina de cargar la página



## Varios elementos

En los Eventos, el `this` esta palabra clave hace referencia al elemento específico que dispara el Evento en un momento determinado.

Supongamos que tenemos una lista con los siguientes elementos:

Si quisiéramos que cada nombre cambiara de color ante el click del usuario, podríamos usar la keyword `this`.

Este código iterará sobre cada uno de los `<li>`. Cada uno de ellos responderá al estímulo del click, cambiando el color de texto a rojo ("red").



Por ejemplo, si el usuario hace click en "Matias", <sup>+</sup>this va a referirse al tag `<li>` que contiene a "Matias". En cambio, si el <sub>•</sub>usuario hace click en "Denis", <sub>o</sub>this referirá al tag `<li>` que contiene a "Denis".

```
<ul class="profesores">
  <li>Matias</li>
  <li>Rodrigo</li>
  <li>Denis</li>
</ul>;
let lis =
document.querySelectorAll(".profesores >
li");
for (let i = 0; i < lis.length; i++) {
  lis[i].addEventListener("click", function ()
{
  this.style.color = "red";
});
}
```

## RECUERDE

Podemos agregar Eventos a varios elementos al mismo tiempo a través de la palabra reservada **this**.

## Eventos: `mouseover` y `mouseleave`

Como ya vimos, existen muchos Eventos (además del "click").

Otros Eventos muy usados son:

- **mouseover:** Se activa cuando el mouse se pone sobre el elemento o alguno de sus descendientes.
- **mouseleave:** Se activa cuando el mouse sale del elemento (y sus descendientes).

Veamos un ejemplo. Supongamos que tuviéramos el siguiente texto:

```
<p id="target">¡Hola!</p>
```

Si quisiéramos que el fondo cambiara a amarillo cada vez que el cursor se pose sobre el elemento (y se desactive cuando ya no está posicionado), podríamos usar `mouseover` y `mouseleave`. En el siguiente [link](#), verás en acción estos eventos.

# Los *argumentos* de un Evento

Cada vez que se dispara un Evento, el browser genera un Objeto JavaScript con información muy valiosa que recaba un eventListener: qué tipo de Evento fue, a qué hora se generó, en qué posición estaba el mouse, si el usuario hizo click, etc.

Para poder acceder a él, debemos pasarle como argumento una Función Anónima que luego podamos loguear en la consola. Incluso, podemos acceder al valor de cualquiera de las Propiedades guardadas en él para, por ejemplo, reutilizarlo.

Resumiendo, para acceder a este Objeto JavaScript, debemos capturarlo como argumento dentro de la Función callback.

Veamos la sintaxis. En este caso, loguearemos en la consola la información del Objeto para que puedas explorarlo.

Podés seleccionar el `<body>` de una forma más rápida, usando `document.body` (en vez de `document.querySelector("body")`).

+

En el ejemplo, le pusimos el nombre `infoEvento` al argumento de la Función, pero podría ser cualquier otra palabra. Esto sirve para indicarle al browser que queremos usar ese Objeto JavaScript con información.

```
document.body.addEventListener("click",  
function (infoEvento) {  
    console.log(infoEvento);  
});
```

## Eventos disparados por el teclado

Podemos hacer que se activen ciertas acciones a partir de una tecla presionada. Por ejemplo, podés asociar una tecla a un sonido o imagen. Esto sirve para lograr una mayor interacción con el usuario.

### Tipo de Eventos del teclado

Evento	Descripción	Ejemplo
keydown	Se produce al presionar cualquier tecla (no hace falta mantenerla presionada)	Tecla "Shift" + tecla "a" (se producen dos Eventos)
keyup	Se produce al soltar la tecla presionada	Tecla "a" (si se mantiene presionada por mucho tiempo, va a captar solo la última "a" al levantar el dedo de la tecla)
keypress	Se produce en el instante de presionar una tecla	Se produce un solo Evento al presionar tecla "Shift" + tecla "a"

## Crear, agregar y remover elementos

En resumen, el Evento **keydown** se produce al presionar cualquier tecla. Por ejemplo, si presionamos la tecla "Shift" y la tecla "a" al mismo tiempo, se producen dos Eventos.

El Evento **keyup** se produce al soltar la tecla presionada. Es decir, cada tecla presionada (y soltada) disparará un nuevo Evento.

El Evento **keypress** también se produce al presionar una tecla. A diferencia del Evento **keydown**, si presionamos la tecla "Shift" y la tecla "a" al mismo tiempo, se producirá un solo Evento ("A").

*Importante:* Como vimos antes, cada Evento genera un Objeto JavaScript. En este caso, los Objetos **KeyboardEvent** describen la interacción del usuario con el teclado y ofrecen Propiedades distintas a las que vimos con el mouse. Para aprender más, mirá la documentación [Mozilla](#).

MÓDULO 2

+

o



OTEC PLATAFORMA 5 CHILE

# GRACIAS

Aquí finaliza la clase n°8 del módulo 2