**Clase 45 Agregando interacción con eventos**

**Documentacion:**

**Mouse:**

[Opcion1](https://es.javascript.info/mouse-events-basics)

[Opcion2](https://developer.mozilla.org/es/docs/Web/API/MouseEvent)

**Teclado:**

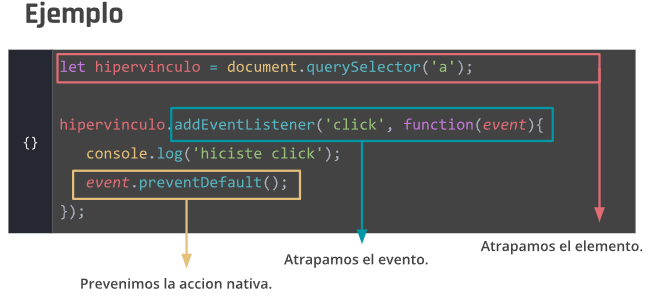
[Opcion1](https://naviscode.com/blog/eventos-de-teclado-en-javascript)

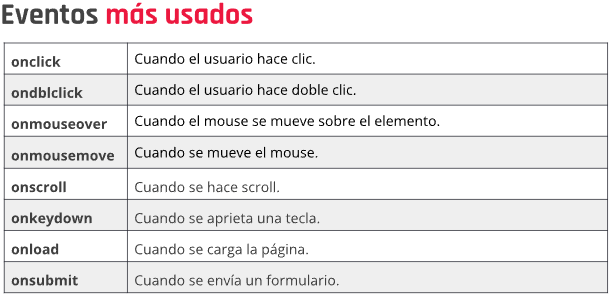
[Opcion2](https://developer.mozilla.org/es/docs/Web/API/KeyboardEvent)

[Opcion3](https://es.javascript.info/keyboard-events)

**Eventos:**

1. **Onload:** Este evento permite que todo el script se ejecute recien cuando haya cargado todo el html.
2. **Onclick:** Este evento nos permite ejecutar una acción cuando se haga click sobre el elemento que le indicamos.
3. **PreventDefault:** Nos permite evitar que se ejecute el evento predeterminado/nativo del elemento que seleccionamos.

j



La diferencia entre onALGO y addEvenListener es:  
onAlgo que si yo defino 2 onload por ejemplo la segunda va a pisar la primera, en addEvenListener no se pisan, van a suceder todos los eventos.

**addEventListener:**

* **Método:**
  + Utiliza el método **addEventListener** para agregar un evento a un elemento.
  + Puedes asociar varios manejadores de eventos al mismo tipo de evento.
  + Es el método preferido para la mayoría de las aplicaciones modernas.

**on:**

* **Propiedad:**
  + Utiliza la propiedad **on** del elemento para asignar un manejador de eventos directamente.
  + Solo puedes asignar un manejador de eventos para un tipo de evento específico.

**¿Cuándo usar cada uno?**

* **addEventListener:**
  + Es más flexible y preferido en aplicaciones modernas.
  + Permite agregar y quitar eventos dinámicamente.
  + Permite asociar varios manejadores de eventos al mismo tipo de evento.
* **on:**
  + Es más simple y puede ser útil en situaciones simples.
  + Puede ser más conveniente cuando solo necesitas un manejador de eventos para un tipo específico.

**Recomendación:**

* **addEventListener es la opción moderna y generalmente preferida.** Proporciona una mejor estructura de código y flexibilidad.
* **on se considera menos deseado en desarrollo moderno.** Su uso está más asociado con enfoques más antiguos y puede limitar la capacidad de gestionar múltiples manejadores de eventos o de quitar eventos de forma dinámica.

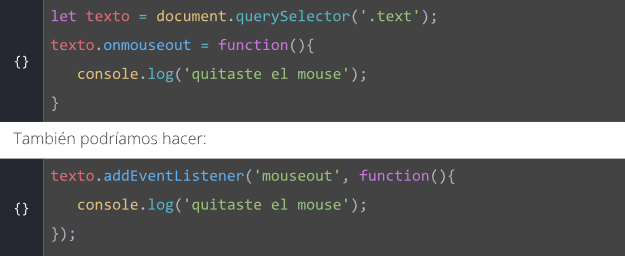
**Eventos de mouse:**

**MouseOver:**



**MouseOut:**

Este evento se ejecuta cada vez que se mueve el mouse



**Eventos de teclado:**

1. **keydown:** Se dispara al presionar una tecla
2. **keyup:** Se dispara al soltar una tecla
3. **keypress:** se dispara al tocar y finaliza al soltar la tecla, este evento esta deprecado asi que es mejor no usarlo

**Limitaciones y Consideraciones:**

**No Captura Teclas Modificadoras:**

El evento keypress no detecta teclas modificadoras como Shift, Ctrl o Alt. Si necesitas manejar estas teclas, podrías considerar usar keydown o keyup en su lugar.

**No Detecta Todas las Teclas:**

Algunas teclas especiales y no imprimibles (como las teclas de función o las teclas de dirección) no activan el evento keypress. Para estas situaciones, puedes usar keydown o keyup.

**Codificación de Caracteres:**

keypress trabaja directamente con el código ASCII de los caracteres, lo que puede tener implicaciones cuando se trata de caracteres especiales o de diferentes idiomas.

Propiedades de teclas:

Key: Con esta propiedad podemos saber que tecla se presiono

**Eventos de Mouse:  
  
Evento mouseenter**: Se dispara cuando el mouse entra al control que está escuchando. Este evento se ejecuta sólo una vez, para que se vuelva a ejecutar debemos salir del control y volver a entrar.  
  
**Evento mouseleave:** Se dispara cuando el mouse sale del control que está escuchando. Este evento se ejecuta sólo una vez, para que se vuelva a ejecutar debemos ingresar al control y volver a salir.  
  
Son simples, entro y salgo de un control y sus respectivos eventos se disparan al momento en que sucede la acción, pero no es lo mismo para los siguientes:  
  
**Evento mouseover:** Se dispara cuando el mouse se encuentra dentro del control que esta escuchando el evento, es decir, el puntero entra al control y se ejecuta, pero mientras el puntero este adentro del control y lo mueva se vuelve a ejecutar. Hay que destacar la diferencia con el evento mouseenter, ya que el ultimo solo se ejecuta una vez, en cambio el mouseover cuantas veces me mueva dentro del control.  
  
**Evento mouseout:** Se dispara cuando el mouse sale del control que esta escuchando el evento. Pero es diferente al mouseleave, en este caso el evento se dispara si algún elemento hijo del mismo dispara el mouseout.