

eventListeners

.data

.preventDefault()

.target

EventListener

Parámetro del eventListener

Hay un parámetro que se le puede pasar a la función que está dentro del eventListener y contiene información del evento. **No importa el nombre del parámetro** dado que siempre va a cumplir esta función.

```
button.addEventListener('click', function (event) {  
    console.log(event);  
});
```

¿Qué información ves?

target (el elemento clickeado)

clientX, clientY (Coordenadas donde estaba el mouse)

altKey, ctrlKey (Si se clickeó la tecla Ctrl o Alt)

Target

Una de las propiedades más importantes, *o al menos que más vamos a usar*, que contiene el event, es el **target**. Este contiene información del elemento en el cuál se lanzó el evento.

```
button.addEventListener('click', function (event) {  
    console.log(event.target);  
});
```

Una vez que tenemos el target, básicamente tenemos el elemento HTML, ese Nodo que ya conocemos. Por lo tanto, podemos usar las propiedades que conocemos:

classList
style
innerHTML
textContent
Etc

Prevent Default

Muchas veces tenemos elementos HTML que tienen un comportamiento por defecto al dispararse un evento, por ejemplo un link que redirige a otra página, o un botón dentro de un formulario, que envía la información de un formulario.

Cuando utilizamos un evento sobre estos elementos, podemos disparar el evento y cancelarlo sin detener el resto del funcionamiento del evento. Esto se logra mediante el método **preventDefault()**

```
link.addEventListener('click', function (event) {  
    event.preventDefault();  
});
```

Cuando le hago click al link, cancelo su comportamiento y por eso no me redirige a ningún lado.

Eventos de teclado

e.keyCode

Eventos de teclado

Existen eventos de teclado que permiten escuchar todo lo que el usuario teclea. Se pueden aplicar a elementos particulares como un input así como al window y escuchar todo lo que el usuario teclea.

Aplicado en inputs

```
const input = document.querySelector("input");

input.addEventListener('keydown', function (e) {
  console.log(e.keyCode);
})

const input = document.querySelector("input");

input.addEventListener('keyup', function (e) {
  console.log(e.keyCode);
})
```

Aplicado en el window

```
window.addEventListener('keydown', function (e) {
  console.log(e.keyCode);
})

window.addEventListener('keyup', function (e) {
  console.log(e.keyCode);
})
```


Data Attribute

.data

.preventDefault()

.target

data

Parámetro del eventListener

Los atributos data-* permiten almacenar información adicional sobre un elemento HTML

Un atributo cualquiera cuyo nombre comience con **data-** es un atributo de datos. Supongamos que tenemos un artículo y deseamos almacenar información adicional que no tiene ninguna representación visual. En ese caso, alcanza con que hagamos uso de los atributos data

```
<div  
  class= "card"  
  id="bariloche"  
  data-index="123"  
  data-promo="20">  
  ...  
</div>
```

```
data-index = "123"  
data-promo = "20"
```

Información Extra

Acceder a data

El nombre de un atributo de datos comienza en HTML con data-. Sólo puede estar formado por letras minúsculas, números y los caracteres: guión (-), punto (.), dos puntos (:) y guión bajo (_) -- NUNCA una letra mayúscula (A a Z).

El nombre del atributo en JavaScript será el del correspondiente atributo HTML pero en camelCase, sin guiones, puntos, etc.

1. `event.target.getAttribute("data-index");`

2. `event.target.dataset.index`

```
button.addEventListener('click', function (event) {  
    console.log(event.target.dataset.index);  
});
```

```
<button data-index="1"></button>
```