



# Tecnología

JAVASCRIPT 02 | Módulo: JAVASCRIPT  
FRONT

# ◀TEMA DE LA CLASE▶

EVENTOS

# OBJETIVOS |

Hay situaciones en las que necesitamos realizar una determinada acción **cuando ocurra un determinado caso**. En estas situaciones, no sabemos exactamente cuando tenemos que activar nuestra funcionalidad, ya que no podemos predecir cuando el usuario de nuestra página realizará la acción necesaria (*y podrá ser diferente en cada situación*).

En estas situaciones es cuando entran en juego los **eventos**.

## ¿Qué es un evento?

En Javascript existe un concepto llamado **evento**, que no es más que una notificación de que alguna **característica interesante** acaba de ocurrir, generalmente relacionada con el **usuario** que navega por la página.

Dichas características pueden ser muy variadas:

- Click de ratón del usuario sobre un elemento de la página
- Pulsación de una tecla específica del teclado
- Reproducción de un archivo de audio/video
- Scroll de ratón sobre un elemento de la página
- El usuario ha activado la opción «Imprimir página»

# EVENTO |

Un evento es algo que pasa en el browser o que es ejecutado por el usuario. Por ejemplo:

- La página terminó de cargar.
- El usuario hizo click en un botón.
- El input del formulario cambió.

# EVENTOS |

## EVENTOS MÁS USADOS

- onclick
- ondblclick
- onmouseover
- onmouseout
- onmousemove
- onscroll
- onkeydown
- onload
- onfocus
- onblur
- onchange
- onsubmit

# EVENTO ONLOAD |

## ONLOAD

El evento onLoad es un evento que permite que todo el script se ejecute cuando se haya cargado por completo el objeto document dentro del objeto window.

```
{} window.onload = function(){
    console.log("el documento está listo");
}
```



Se suele escribir el código js dentro de esta función para **prevenir errores** que pueden ocurrir si el documento no está totalmente cargado al momento de la ejecución del script.

# EVENTO ONCLICK |

## ONCLICK

El evento onclick nos permite ejecutar una acción cuando se haga click sobre el elemento al cual le estamos aplicando la propiedad.

```
{  
  btn.onclick = function(){  
    console.log("¡hiciste click!");  
  }  
}
```

# PREVENT DEFAULT() |

El **preventDefault()** nos permite evitar que se ejecute el evento predeterminado o nativo del elemento al que se lo estamos aplicando.

Podemos usarlo, por ejemplo, para prevenir que una etiqueta "a" se comporte de manera nativa y que en vez de eso haga otra acción.



Siempre tenemos que tener seleccionado el elemento al que le queremos aplicar el **preventDefault()** mediante los selectores.



# PREVENT DEFAULT() |

{ }

```
let hipervinculo = document.querySelector("a");

hipervinculo.addEventListener("click", function(event){
    console.log("hiciste click");
    event.preventDefault();
});
```

# PREVENT DEFAULT() |

