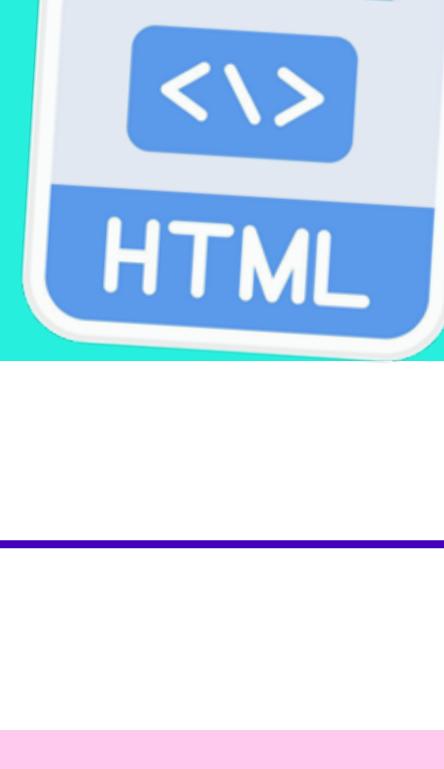


Eventos

- Los eventos son acciones que pasan dentro de un navegador web y que le ocurren a elementos HTML.
- Éstos son inherentes al navegador web en sí, el cual es el encargado de controlarlos y detonarlos cada que una acción por parte del usuario ocurra.
- Cuando se implementan con JavaScript, se tienen acceso a ese conjunto de eventos y se pueden "capturar", lo que significa que el código JavaScript "reacciona" a los mismos.
- Podemos manejar los eventos si se asigna un event handler a los elementos HTML a los cuales queremos capturarles los eventos cuando éstos ocurran.
- Un evento puede ser ocasionado por una acción interna del código, del navegador web en sí o puede ser ocasionado por una acción del usuario.

Los event handlers normalmente son asignados usando atributos de evento, esto es, atributos especiales de un elemento HTML.

Normalmente los atributos de eventos empiezan con el prefijo **on** seguido de una referencia al evento que se quiere escuchar, los cuales son prácticamente autoexplicativos.



Los más usados son

onchange Ocurre cuando un elemento cambió de valor.

onclick Ocurre cuando el usuario dio clic sobre la superficie de un elemento.

onmouseover Ocurre cuando el usuario mueve el puntero sobre la superficie de un elemento.

onmouseout Ocurre cuando el puntero salió de la superficie de un elemento.

onkeydown Ocurre cuando el usuario presiona una tecla sobre un elemento que tenga el foco.

onload Ocurre cuando el navegador termina de descargar la página completa.

Un event handler puede ser un pedazo de código JavaScript que directamente se asigne en el atributo de evento.

Ejemplo:

```
<button id="boton1" onclick="alert('Hola Mundo!')">Presióname</button>
```

o una función ya definida en JavaScript.

Ejemplo:

```
<button onclick="manejadorEvento()">Presióname</button>
```

También se pueden asignar eventos usando el DOM directamente.

Ejemplo:

```
document.getElementById("boton1").onclick = manejadorEvento();
```

Y también se pueden agregar varias funciones a un mismo evento, usando la propiedad **addEventListener** del **DOM**, al hacer esto se puede estar ejecutando 2 o más pedazos de código cada vez que un evento en particular ocurra, esto no sobrescribe asignaciones previas, si no que cada vez que se asigne, se crea una ejecución.

