



TEMA 8.

EVENTOS

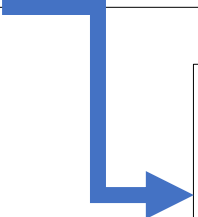
1. MÉTODO CLÁSICO Y OBSOLETO

- Hace años, en las primeras versiones de JavaScript, la captura de un evento se realizaba desde el propio código HTML.
- Todavía se puede hacer porque se ha mantenido esta forma de captura por cuestiones de compatibilidad, pero no es nada aconsejable, ya que complica enormemente el mantenimiento del código al estar mezclado HTML y JavaScript de forma confusa.

1. MÉTODO CLÁSICO Y OBSOLETO

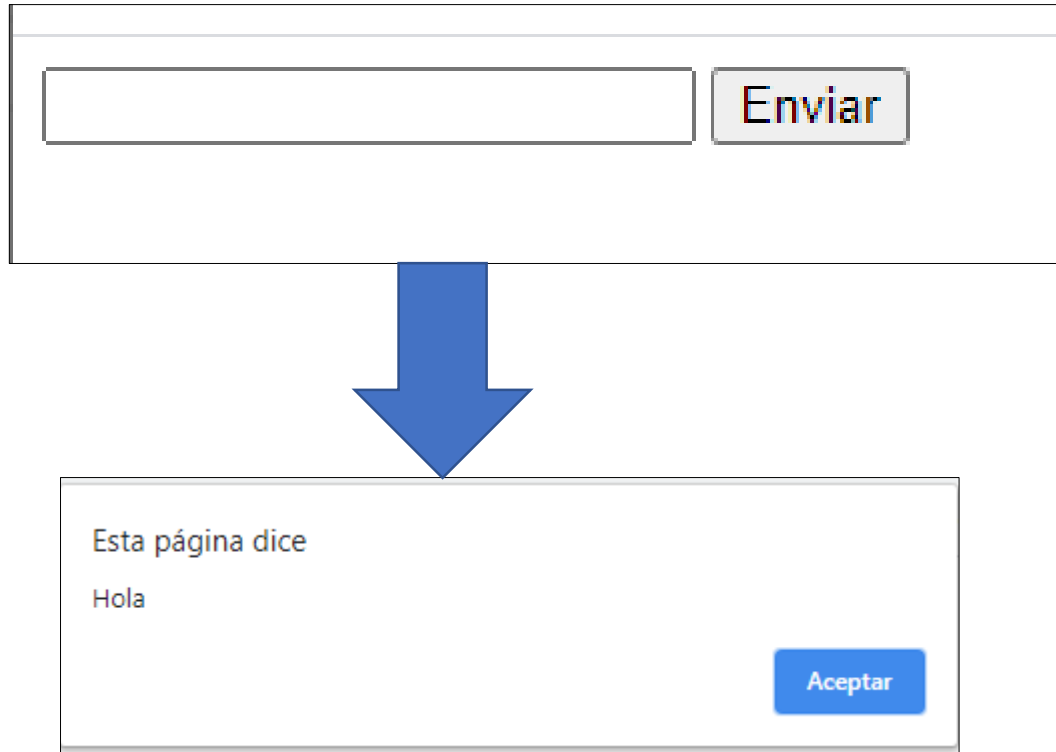
- Este primer método utiliza atributos en las etiquetas de los elementos de la página que comienzan con la palabra **on** seguida del nombre del evento.
- Por ejemplo: **onclick**. Como valor del atributo se indica el nombre de la función que contiene el código a ejecutar. Ejemplo:

```
<input type="text" name="nombre" id="nombre">  
<input type="submit" value="Enviar" onclick="javascript:funcion();">
```



```
function funcion() {  
    alert("Hola");  
}
```

1. METODOS CLÁSICOS Y OBSOLETOS



1. METODOS CLÁSICOS Y OBSOLETOS

Tipo de evento	Nombre con prefijo on (eliminar cuando proceda)	Descripción aprenderaprogramar.com
Relacionados con el ratón	onclick	Click sobre un elemento
	ondblclick	Doble click sobre un elemento
	onmousedown	Se pulsa un botón del ratón sobre un elemento
	onmouseenter	El puntero del ratón entra en el área de un elemento
	onmouseleave	El puntero del ratón sale del área de un elemento
	onmousemove	El puntero del ratón se está moviendo sobre el área de un elemento
	onmouseover	El puntero del ratón se sitúa encima del área de un elemento
	onmouseout	El puntero del ratón sale fuera del área del elemento o fuera de uno de sus hijos
	onmouseup	Un botón del ratón se libera estando sobre un elemento
	contextmenu	Se pulsa el botón derecho del ratón (antes de que aparezca el menú contextual)
Relacionados con el teclado	onkeydown	El usuario tiene pulsada una tecla (para elementos de formulario y body)
	onkeypress	El usuario pulsa una tecla (momento justo en que la pulsa) (para elementos de formulario y body)
	onkeyup	El usuario libera una tecla que tenía pulsada (para elementos de formulario y body)

1. METODOS CLÁSICOS Y OBSOLETOS

Relacionados con formularios	onfocus	Un elemento del formulario toma el foco
	onblur	Un elemento del formulario pierde el foco
	onchange	Un elemento del formulario cambia
	onselect	El usuario selecciona el texto de un elemento input o textarea
	onsubmit	Se pulsa el botón de envío del formulario (antes del envío)
	onreset	Se pulsa el botón reset del formulario
Relacionados con ventanas o frames	onload	Se ha completado la carga de la ventana
	onunload	El usuario ha cerrado la ventana
	onresize	El usuario ha cambiado el tamaño de la ventana
Relacionados con animaciones y transiciones	animationend, animationiteration, animationstart, beginEvent, endEvent, repeatEvent, transitionend	
Relacionados con la batería y carga de la batería	chargingchange, chargingtimechange, dischargingtimechange, levelchange	

1. METODOS CLÁSICOS Y OBSOLETOS

Relacionados con llamadas tipo telefonía	alerting, busy, callschanged, connected, connecting, dialing, disconnected, disconnecting, error, held, holding, incoming, resuming, statechange
Relacionados con cambios en el DOM	DOMAttrModified, DOMCharacterDataModified, DOMContentLoaded, DOMElementNameChanged, DOMNodeInserted, DOMNodeInsertedIntoDocument, DOMNodeRemoved, DOMNodeRemovedFromDocument, DOMSubtreeModified
Relacionados con arrastre de elementos (drag and drop)	drag, dragend, dragenter, dragleave, dragover, dragstart, drop
Relacionados con video y audio	audioprocess, canplay, canplaythrough, durationchange, emptied, ended, ended, loadeddata, loadedmetadata, pause, play, playing, ratechange, seeked, seeking, stalled, suspend, timeupdate, volumechange, waiting, complete
Relacionados con la conexión a internet	disabled, enabled, offline, online, statuschange, connectionInfoUpdate
Otros tipos de eventos	Hay más tipos de eventos: relacionados con la pulsación sobre pantallas, uso de copy and paste (copiar y pegar), impresión con impresoras, etc.

2. METODO ACONSEJABLE

- El método que se aconseja actualmente es utilizar el método **addEventListener** que está disponible en todos los elementos.
- A este método hay que indicarle el nombre del evento y la función callback a ejecutar. Es el método que se debe utilizar según la norma actual.
- Todos los navegadores actuales lo reconocen (las versiones de Internet Explorer anteriores a la 9, no lo hacen) y tiene una ventaja técnica sobre los dos métodos anteriores: se puede asignar más de una función al mismo evento.

2. METODO ACONSEJABLE

- Partimos del ejemplo anterior HTML (eliminando obviamente onclick del código HTML).

```
<input type="text" name="nombre" id="nombre">  
<input type="submit" value="Enviar" id="enviar">
```

- La function función la conservamos dentro del fichero JS.

```
function funcion() {  
    alert("Hola");  
}
```

2. METODO ACONSEJABLE

- Hay que añadir la funcionalidad al botón enviar en JS, para eso nos servimos de `addEventListener` (y por supuesto, de `document.querySelector`).

```
let enviar=document.querySelector("#enviar");  
enviar.addEventListener("click",funcion);
```

3. EVENTOS DE RATÓN

EVENTO	EXPLICACIÓN
click	El usuario hace clic sobre el botón principal del dispositivo apuntador. Esta acción exactamente consiste en bajar y subir rápidamente el botón principal del dispositivo apuntador (o dar un golpe con el dedo en un dispositivo táctil)
dblclick	El usuario golpea dos veces rápidas sobre el botón principal del dispositivo apuntador (o con el dedo en un dispositivo táctil)
mousedown	El usuario pulsa un botón del dispositivo apuntador. Se produce este evento justo cuando el usuario pulsa el botón, antes de que le suelte.
mouseup	Se produce cuando el usuario suelta el botón del dispositivo apuntador que había pulsado previamente.
mousenter	Se produce cuando el usuario mueve el cursor del dispositivo apuntador dentro del elemento que captura el evento. Solo se produce si el cursor estaba fuera del elemento y el usuario le acaba de mover dentro.
mouseleave	Se produce cuando el usuario mueve el cursor del dispositivo apuntador fuera del elemento que captura el evento. Solo se produce si el cursor estaba dentro del elemento y el usuario le acaba de mover fuera.
mousemove	Se produce cada vez que el usuario mueve el cursor dentro del elemento (estando previamente dentro).
mouseover	Se produce cuando el cursor apuntador entra dentro del elemento o de cualquiera de los hijos del elemento.
mouseout	Se produce cuando el cursor apuntador abandona el elemento o cualquiera de los hijos del elemento.
contextmenu	Se produce cuando el usuario acaba de pedir el menú de contexto. Normalmente ese menú aparece al pulsar el botón secundario del dispositivo apuntador. En dispositivos táctiles suele ocurrir cuando el usuario deja el dedo apretado sobre el elemento 2 o 3 segundos al menos.

4. EVENTOS DE TECLADO

EVENTO	EXPLICACIÓN
keypress	Se produce cuando un usuario pulsa y suelta una tecla.
keydown	Se produce justo cuando el usuario ha pulsado la tecla (antes de que la suelte).
keyup	Se produce cuando el usuario suelta la tecla.

5. EVENTOS DE VENTANA

EVENTO	EXPLICACIÓN
scroll	Ocurre cuando se ha desplazado la ventana a través de las barras de desplazamiento o usando el dispositivo táctil.
resize	Evento de window que se produce cuando se cambia el tamaño de la ventana.

6. EVENTOS SOBRE CARGA Y DESCARGA DE ELEMENTOS

EVENTO	EXPLICACIÓN
load	<p>Se concluyó la carga del elemento. Es uno de los elementos más importantes a capturar para asegurar que el código siguiente funciona con la seguridad de que está cargado lo que necesitamos.</p> <p>Cuando se aplica al elemento window, se produce cuando todos los elementos del documento se han cargado.</p>
DOMContentLoaded	<p>Similar al evento <i>load</i>. Se produce cuando el documento HTML ha sido cargado. A diferencia de <i>load</i>, se dispara sin esperar a que se terminen de cargar las hojas de estilos, imágenes y elementos en segundo plano.</p> <p>En general, para JavaScript, es más conveniente este evento.</p>
abort	Se produce cuando se anula la carga de un elemento.
error	Sucede si hubo un error en la carga.
progress	Se produce si la carga está en proceso.
readystatechange	Ocurre cuando se ha modificado el estado del atributo readystate, lo cual ocurre cuando se ha modificado el estado de carga y descarga.

7. EVENTOS DE FORMULARIO

EVENTO SUBMIT

- Se trata del evento que produce el envío de los datos del formulario a su destino. Es muy habitual validar los datos previamente a su envío para que el usuario detecte los fallos antes de realizar el envío. Este evento se produce justo antes del envío y se asocia al propio formulario (también se puede capturar en el botón de tipo submit del formulario).
- Tras su captura podemos validar los datos, pero teniendo la precaución de que si no cancelamos la acción por defecto de este evento (mediante el **preventDefault** del objeto de evento), la acción se llevará a cabo.

7. EVENTOS DE FORMULARIO

- Como ejemplo, supongamos que tenemos un formulario en el que pedimos al usuario su nombre de usuario con un mínimo de 6 caracteres. No enviaremos datos si no se cumple esa premisa y cuando detectemos el fallo, avisaremos del error y no permitiremos el envío.
- El código podría ser éste:

```
<form action="#" id="formulario">  
  <input type="text" name="usuario" id="usuario">  
  <input type="submit" value="Enviar" id="enviar">  
</form>  
<div id="mensaje"></div>
```


7. EVENTOS DE FORMULARIO

```
let enviar=document.querySelector("#formulario");

enviar.addEventListener("submit",function (ev) {
    // la funcion va dentro del addEventListener

    let texto=document.querySelector("#usuario").value;
    if (texto.length<=6) {
        ev.preventDefault();
        document.querySelector("#usuario").style.backgroundColor="red";
        document.querySelector("#mensaje").innerHTML="Minimio 6 caracteres, tron";
    }
    else {
        alert("Formulario enviado");
    }
});
```

7. EVENTOS DE FORMULARIO

Minimio 6 caracteres, tron