



4 . Interacción con el usuario

- Eventos -

Eventos

Evento

- Mecanismo que se acciona cuando el usuario realiza un cambio sobre la página web.
- Capturar un evento es programar una acción para que se realice una tarea.
- El encargado de gestionar los eventos es el DOM (Document Object Model).

Manejador (handler)

- Palabra reservada que indica la acción que se va a manejar. Por ejemplo, en el evento click el manejador es onClick.



Eventos del ratón

Evento	Manejador	Descripción
Click	onclick	Cuando pulsamos el botón derecho del ratón.
Dblclick	ondblclick	Cuando hacemos doble click el botón derecho del ratón.
Mousedown	onmousedown	Cuando pulsamos un botón del ratón.
Mouseup	onmouseup	Cuando soltamos un botón del ratón que previamente teníamos pulsado.
Mouseout	onmouseout	Cuando el puntero del ratón esta dentro de un elemento y este puntero es desplazado fuera del elemento.
Mouseover	onmouseover	Cuando el puntero del ratón se encuentra fuera de un elemento, y este se desplaza hacia el interior.
Mousemove	onmousemove	Cuando el puntero del ratón se encuentra dentro de un elemento. Es importante señalar que este evento se producirá continuamente una vez tras otra mientras el puntero del ratón permanezca dentro del elemento.



Eventos del teclado

Evento	Manejador	Descripción
Keydown	onkeydown	Cuando pulsamos una tecla del teclado. Si mantenemos pulsada una tecla de forma continua, el evento se produce una y otra vez hasta que soltemos la misma
Keyup	onkeyup	Cuando soltamos una tecla.



Eventos HTML (I)

Evento	Manejador	Descripción
Load	onload	El evento load hace referencia a la carga de distintas partes de la página. Este se produce en el objeto Window cuando la página se ha cargado por completo. En el elemento actúa cuando la imagen se ha cargado. En el elemento <object> se acciona al cargar el objeto completo.
Unload	onunload	El evento unload actúa sobre el objeto Window cuando la página ha desaparecido por completo (por ejemplo, si pulsamos el aspa cerrando la ventana del navegador). También se acciona en el elemento <object> cuando desaparece el objeto.
Abort	onabort	Este evento se produce cuando el usuario detiene la descarga de un elemento antes de que haya terminado, actúa sobre un elemento <object> .
Error	onerror	El evento error se produce en el objeto Window cuando se ha producido un error en JavaScript . En el elemento cuando la imagen no se ha podido cargar por completo y en el elemento <object> en el caso de que un elemento no se haya cargado correctamente.



Eventos HTML (II)

Evento	Manejador	Descripción
Change	onchange	Cuando los cuadros de texto <code><input></code> y <code><textarea></code> pierden el foco y el contenido que tenían ha variado. Cuando un elemento <code><select></code> cambia de valor.
Submit	onsubmit	Cuando pulsamos sobre un botón de tipo <code>submit</code> en un formulario
Reset	onreset	Cuando pulsamos sobre un botón de tipo <code>reset</code> en un formulario.
Error	onerror	El evento error se produce en el objeto <code>Window</code> cuando se ha producido un error en <code>JavaScript</code> . En el elemento <code></code> cuando la imagen no se ha podido cargar por completo y en el elemento <code><object></code> en el caso de que un elemento no se haya cargado correctamente.



Eventos HTML (III)

Evento	Manejador	Descripción
Resize	onresize	Cuando redimensionamos el navegador, actúa sobre el objeto Window .
Scroll	onscroll	Cuando varía la posición de la barra de scroll en cualquier elemento que la tenga.
Focus	onfocus	Cuando un elemento obtiene el foco. En elementos button , input , label , select , textarea , body .
Blur	onblur	Cuando un elemento pierde el foco. En elementos button , input , label , select , textarea , body .



Eventos DOM

Evento	Descripción
DOMSubtreeModified	Cuando añadimos o eliminamos nodos en el subárbol de un elemento o documento.
DOMNodeInserted	Cuando añadimos un nodo hijo a un nodo padre.
DOMNodeRemoved	Cuando eliminamos un nodo que tiene nodo padre.
DOMNodeRemovedFromDocument	Cuando eliminamos un nodo del documento.
DOMNodeInsertedIntoDocument	Cuando añadimos un nodo al documento.



Modelos de gestión de eventos

Los eventos se ejecutan cuando el usuario interactúa con la página (haciendo clic, pulsando una tecla, moviendo el ratón, etc.) o bien se tratan de eventos no relacionados con acciones del usuario (como **onload**, que sucede cuando la página se carga).

Un poco de historia:

Incompatibilidades de navegadores: había que comprobar si los métodos eran soportados.



Modelo de registros de eventos en línea. Se añade como atributo a la etiqueta html:
``

Modelo de registro de eventos tradicional: los eventos son una propiedad de los elementos:
`elemento.onclick=accion;`

Modelo avanzado de registro de eventos del W3C: Utiliza
`addEventListener();`

Modelo avanzado de registro de eventos de Microsoft: Utiliza
`attachEvent();`



Eventos en línea

Modelo de registro de eventos en línea

El evento se añade como atributo de una etiqueta **html**. Se ejecuta primero el script y luego la acción por defecto.

Ejemplos:

1) `Pulsa aquí`

2) `Pulsa aquí`

```
function alertar() {  
    alert("Has pulsado el enlace");  
}
```

Para evitar que el navegador ejecute acción por defecto:

`Pulsa aquí`

De este modo no ejecutaría la acción por defecto: redirigir a la página siguiente.html.

Para preguntar si ejecutar o no la acción por defecto:

`Pulsa aquí`

```
function preguntar() {  
    return confirm("¿Desea realmente ir a esa dirección?");  
}
```



Eventos en línea

¿Por qué no utilizar el modelo de registro de eventos en línea?

**No separa la programación
de la estructura.**

**Si la función redirige a
una página no se
ejecutan las demás
tareas.**

**Al ejecutar la acción por
defecto se pueden perder los
objetos que hay en memoria
(ej. Al cambiar de página).**



Modelo de registro de eventos tradicional

El evento se convierte en una propiedad del elemento:

Ejemplos:

```
elemento.onclick = accion;  
function accion() {...}
```

Para eliminar un gestor de eventos de un elemento:

```
elemento.onclick = null;
```

Para llamar al gestor de eventos cuando queramos:

```
elemento.onclick();
```



Modelo de registro de eventos avanzado del W3C

Utiliza un evento `addEventListener()` con tres argumentos:

- Tipo de evento.
- Función a ejecutar.
- Valor booleano para elegir la fase de captura (`true`) o burbujeo (`false`) que es por defecto (si no se pone nada).

```
elemento.addEventListener('evento', función, false|true);
```

Ejemplo:

```
document.getElementById("mienlace").addEventListener('click', alertar, false);
```

¡¡Ojo!! No lleva “on” delante del nombre del evento.



Modelo de registro de eventos avanzado del W3C

Eliminar un evento de un elemento:

```
elemento.removeEventListener();
```

Crear más de un evento para el mismo elemento:

```
document.getElementById("mienlace").addEventListener("click", alertar, false);  
document.getElementById("mienlace").addEventListener("click", avisar, false);  
document.getElementById("mienlace").addEventListener("click", chequear, false);
```

Crear un evento anónimo (no está asociado a ninguna función):

```
element.addEventListener(  
    "click",  
    function () {  
        this.style.backgroundColor = "#cc0000";  
    },  
    false  
);
```



Modelo de registro de eventos avanzado MS

Utiliza un evento `attachEvent()` con dos argumentos:

- Tipo de evento.
- Función a ejecutar.

¡¡Ojo!! En este modelo de registro de eventos se utiliza “on” delante del nombre.

```
elemento.attachEvent('evento', función);
```

Ej:

```
document.getElementById("mienlace").attachEvent('onclick', alertar);
```

Eliminar un evento de un elemento:

```
elemento.detachEvent("onclick", accion);
```

Crear más de un evento para el mismo elemento:

```
elemento.attachEvent("onclick", hacerUnaCosa);  
elemento.attachEvent("onclick", hacerOtraCosa);
```



EJERCICIO

EJERCICIO - U4T2:

- Vas a crear un juego que consiste en encontrar parejas en 12 cartas con 6 parejas de los personajes de los Simpson. El juego consistirá en lo siguiente:
 - La aplicación deberá tener una tabla con 3 filas y cuatro columnas de un color. Además habrá un cuadro de texto con el valor 0 pero no modificable.
 - Cuando el usuario haga clic sobre una celda, se mostrará una imagen.
 - Cuando el usuario haga clic sobre otra celda, se mostrará otra imagen.
 - Si las dos imágenes son iguales, se cambiará el color de la celda y se añadirá 1 al cuadro de texto.
 - Si las dos imágenes son diferentes, se ocultarán mostrando nuevamente el color inicial.
- En primer lugar, crea un esquema en una hoja con el árbol DOM del documento html y en otra indica los métodos que vas a crear asociados a qué evento.
- Trata de utilizar el menor número de variables posibles: utiliza las propiedades de los elementos para cambiar su comportamiento.



END



prof.jduran@iesalixar.org