EVENTOS AVANZADOS JAVASCRIPT

1. Compartir manejador de eventos

Cuando se ejecuta el código de un manejador de eventos que está asignado a un solo control (por ejemplo hacer click en un botón) sabremos perfectamente quién ha desencadenado el evento (el botón) y de qué tipo de evento se trata (onclick).

Pero JavaScript permite compartir un mismo manejador de eventos. En dicho caso, cuando se ejecute el manejador de eventos podremos rescatar el control y el tipo de evento del objeto evento (argumento del manejador de eventos):

- ev.target ⇒ Control que ha desencadenado el evento
- ev.type ⇒ Tipo de evento

Veamos un ejemplo en que disponemos de un solo manejador de eventos llamado manejador_eventos que es compartido por los siguientes controles:

- Click en boton1
- Click en boton2
- Seleccionar elemento en el desplegable ciudades

```
1.  <body>
2.      <script>
3.          function inicio(){
4.              // El evento hacer click sobre cualquiera de los botones 1,2 y 3 será
    gestionado por el mismo manejador de eventos
5.
    document.getElementById('boton1').addEventListener('click',manejador_eventos);
6.
    document.getElementById('boton2').addEventListener('click',manejador_eventos);
7.
    document.getElementById('ciudades').addEventListener('change',manejador_eventos);
8.          }
9.
10.           function manejador_eventos(ev){
11.             let control= ev.target;
12.             let tipoevento = ev.type;
13.             alert(`Evento desencadenado por control ${control.value} y es de tipo
    ${tipoevento}`);
14.         }
15.         addEventListener('load',inicio);
16.     </script>
17.     <h1>Asignar varios eventos a un único manejador de eventos</h1>
18.     <input type="button" id="boton1" value="Botón 1">
19.     <input type="button" id="boton2" value="Botón 2">
20.     <select name="ciudades" id="ciudades">
```

```
21.         <option value="Utrera">Utrera</option>
22.         <option value="Los Palacios">Los Palacios</option>
23.     </select>
24. </body>
```

(Ver código en **asignar_varios_eventos_a_un_unico_eventhandler.html**)

2. Obtener información de los eventos

Existen algunos eventos que incorporan cierta información que nos será útil rescatar cuando se desencadenan.

2.1. Ejemplo evento onmousemove

Un ejemplo es el evento onmousemove, que se dispara cuando nos movemos con el ratón por la pantalla. En dicho caso podemos rescatar las coordenadas X e Y de la posición del ratón. Para ello tendremos que añadir un argumento en el manejador de eventos (será un objeto que contiene información del evento) , y dentro de la función rescataremos las propiedades clientX y clientY a partir del objeto evento:

```
  // Manejador de eventos
function myFunction(evt) { // evt contiene información relativa al evento
    var x = evt.clientX;
    var y = evt.clientY;
    var coor = "Coordenadas: (" + x + "," + y + ")";
    document.getElementById("demo").innerHTML = coor;
}
```

(Ver código **ej_obtener_info_onmousemove.html**)

2.2. Ejemplo eventos onkeydown , onkeyup y onkeypress

Cuando pulsamos las teclas se desencadena varios eventos:

- **onkeydown**: cuando presionamos una tecla
- **onkeyup**: cuando dejamos de presionar la tecla
- **onkeypress**: cuando presionamos y soltamos un tecla (solo se puede usar con las teclas que representan datos de tipo alfanumérico, como “a”, “1”, etc.)

Asociados a estos eventos disponemos de las siguientes propiedades:

- **keyCode**: es un código numérico que se corresponde con cada una de las teclas (incluso las teclas especiales como F1, F2, Alt, Ctrl, tienen un código asociado)
- **charCode**: cuando la tecla pulsada se corresponde con un valor alfanumérido (“a”, “b”, “3”, “.”, etc.) podremos rescatar su valor con dicha propiedad.

Veamos un ejemplo de uso de estos tres eventos y las propiedades. En el siguiente ejemplo además se está utilizando un solo manejador de eventos, que es compartido por los tres eventos onkeydown, onkeyup y onkeypress. Dentro del manejador de eventos podremos saber cuál de ellos lo ha desencadenado examinado la propiedad type del objeto evento (el cual lo tenemos disponible a través del argumento de la función).

```
1.  <script>
2.          window.onload = function() {
3.              document.onkeydown = muestraInformacion;
4.              document.onkeyup = muestraInformacion;
5.              document.onkeypress = muestraInformacion;
6.            }
7.
8.          // Manejador de eventos compartido por varios eventos (onkeydown, onkeyup,
    onkeypress)
9.          function muestraInformacion(ev) {
10.             let mensaje = "Tipo de evento: " + ev.type + "<br>" +  // Con ev.type
    podemos rescatar qué evento se ha producido (recordemos que es un manejador de
    eventos compartido)
11.                             "Propiedad keyCode: " + ev.keyCode + "<br>" +
12.                             "Propiedad charCode: " + ev.charCode + "<br>" +
13.                             "Carácter pulsado: " + String.fromCharCode(ev.charCode);
14.
15.             document.getElementById("info").innerHTML +=
    "<br>---------------------------------------<br>" + mensaje
16.         }
17.     </script>
18.     <div id="info"></div>
```

(Ver código en **ej_obtener_info_onkeyup-onkeydown-onkeypress.html**)

3. Crear nuevos elementos dinámicamente y asignar eventos

JavaScript permite crear nuevos elementos de manera dinámica (tal y como hemos visto en el tema anterior del DOM). Además de crear los nuevos elementos (párrafos, botones, etc.) podremos asignarle de manera dinámica el manejador de eventos.

Veamos un ejemplo de cómo crear varios párrafos de manera dinámica y añadirles un manejador de eventos compartido. Dentro del manejador de eventos podremos obtener información de qué control desencadenó el evento con ev.target y el tipo de evento con ev.type (siendo ev el objeto de tipo evento que tenemos disponible en el argumento del manejador de eventos).

```
1.  <body>
2.     <script>
3.         function iniciar(){
4.             let contenedor= document.getElementById("contenedor");
5.             // Creamos cuatro párrafos de forma dinámica y le asignamos un
    manejador de eventos compartido
6.             for (let i=0 ; i < 4; i++){
7.                 let parrafo=document.createElement("p");
8.                 parrafo.appendChild(document.createTextNode(`Párrafo ${i} creado
    dinámicamente`));
9.                 parrafo.id=`parrafo${i}`;
10.                contenedor.appendChild(parrafo);
11.                parrafo.addEventListener('click',manejador); // Añadimos el
    manejador de eventos al nuevo párrafo creado
12.            }
13.        }
14.
15.  //Manejador eventos compartido por todos los párrafos creado de manera dinámica
16.        function manejador(ev){
17.            alert(`Control:${ev.target.id} \n Tipo de evento:${ev.type}`);
18.        }
19.        addEventListener('load',iniciar);
20.    </script>
21.    <h1>Haz clic sobre los párrafos que han sido creados dinámicamente</h1>
22.    <div id="contenedor"></div>
23. </body>
```

(Ver código en **ej_crear_eltos_y_asinar_eventos_dinamicamente.html**)

4. Cómo pasar información al manejador de eventos usando addEventListener

Al usar el método addEventListener para añadir un manejador de eventos podemos pasar información al mismo usando una función anónima tal y como podemos ver en el siguiente ejemplo:

```
<!DOCTYPE html>
<html>
<head>
    <title>DOM Event Example</title>
    <style>
        #t {
            border: 1px solid red
        }
        #t1 {
            background-color: pink;
        }
    </style>
    <script>
        // Function to change the content of t2
        function modifyText(evt, new_text) {
            alert(evt.target + " " + evt.type);
            var t2 = document.getElementById("t2");
            t2.firstChild.nodeValue = new_text;
        }
        // Function to add event listener to t
        function load() {
            var el = document.getElementById("t");
            el.addEventListener("click", function (evt) { modifyText(evt, "four") },
false);
        }
    </script>
</head>

<body onload="load();">
    <table id="t">
        <tr>
            <td id="t1">one</td>
        </tr>
        <tr>
            <td id="t2">two</td>
        </tr>
    </table>
</body>
</html>
```

(Ver código en **ej_pasar_info_al_manejador.html**)