DOM y formularios

Laura Isabel López Naves

Eventos

- En una página se pueden producir diferentes eventos. Ejemplos:
- **load** : cuando se carga una página (<body>)

  <body onLoad="fun();alert('Carga la página')">
- **mouseover** : cuando se pasa en ratón por encima (por ejemplo sobre un elemento <a>). Ej: <a href="pagina1.htm" onMouseOver="fun()">
- **submit** : cuando se envía el formulario (<form>)

  <form name="f" action="p.html" onSubmit="validar()">
- **click** : cuando se pulsa el elemento cualquiera (puede ser un "div", "button", "checkbox", "radio", etc.) Ejemplos:

<input type="button" value="Botón" onClick="fun()">

<button type="button" onmouseover="alert('hola');"> Bot.</>

- Se puede responder a un evento con un "**Event Handler**" (función manejadora).

Eventos

- Existen tres formas de suscribir funciones manejadoras de eventos:
  - En línea: es la forma menos utilizada actualmente. Ejemplo:

    ** Prueba **
  - Como una propiedad del DOM. Ejemplo:

```
<input id="elem" type="button" value="Pulsar">
<script>
    elem.onclick = function() {
        alert(this.value);
    };
```

En este caso el valor de “this” dentro de la función manejadora es el propio elemento (“input”). Sería (“window”) en el caso de una función flecha.

  - Utilizando **addEventListener**. Es el método más utilizado y moderno.

```
<input id="elem" type="button" value="Pulsar">
<script>
    elem.addEventListener("click", () => alert("hola"));
    elem.addEventListener("click", () => console.log("hola"));
</script>
```

addEventListener permite tener varios métodos manejadores. Se puede eliminar un manejador utilizando **removeEventListener**

Eventos

- Se puede escuchar un evento “global” como el uso del teclado en el objeto window o un evento específico de un elemento como el click en un botón.
- En el manejador de evento se puede incluir un objeto **Event** como primer parámetro que proporciona información adicional. Ejemplo:
  - **target**: elemento DOM que produjo el evento.
  - **currentTarget**: elemento DOM donde se captura el evento.
  - **type**: tipo de vento.
  - **clientX, clientY**: coordenda X e Y
  - **key**: tecla pulsada

Los datos que devuelve el objeto Event dependen del tipo de evento que se produce: MouseEvent, KeyboardEvent, etc.
Ver: https://developer.mozilla.org/en-US/docs/Web/Events

```
window.addEventListener('keydown', event => {
    console.log(event.key);
});
const d = document.querySelector('#d1');
d.addEventListener('mousedown', event => {
    console.log(event.button) //0=left, 2=right
});
```

En los manejadores de eventos podemos utilizar funciones convencionales o funciones flecha
Si utilizamos funciones fecha no podemos utilizar “this” para obtener el objeto que captura el evento.

Event bubbling

- Por defecto los eventos se propagan siguiendo la estrategia
- “Event bubbling” que consiste en que el evento se propaga desde el ele mento que lo produce al elemento más exterior.
- Ejemplo:

```
<ul>
    <li>Pulsa aquí</li>
</ul>
<script>
    document.querySelector('ul').addEventListener('click', function(event) {
        console.log(this.nodeName); // UL
    });
    document.querySelector('li').addEventListener('click', function(event) {
        console.log(this.nodeName); // LI
        // event.stopPropagation();
    });
</script>
```

- En primer lugar visualiza “LI” y luego “UL”.
- Para evitar la propagación del eventos basta con descomentar última línea.

Eventos populares

- **load:** se dispara en el obteto window y en body cuando se termina de cargar la página.
- Eventos de ratón: **click**, **dblclick**, **mousedown**, **mousemove**, **mouseup**.
- Eventos de teclado: **keydown** (cuando se pulsa y se mantiene pulsada la tecla) y **keyup** (cuando se deja de pulsar la tecla).
- **scroll**: es lanzado por window cuando se produce scroll en la página.
- **submit**: cuando se envía un formulario (FORM).
- **focus**: cuando se pierde el foco en un elemento (por ejemplo en un INPUT).

```
<ul>
    <li>Pulsa aquí</li>
</ul>
<script>
    document.querySelector("li").addEventListener('mousemove', function(event) {
        console.log(event.clientX+" "+event.clientY);
    });
</script>
```

Hace un log de las coordenadas del ratón.

Ejercicio 4 (ej4.htm)

- A partir de una imagen (por ejemplo: reloj.gif) crear una página web que utilice el objeto “event” y las propiedades clientX y clientY.
- Cuando el usuario mueve el ratón por la pantalla, automáticamente se moverá el reloj junto al cursor.
- Como resultado, la imagen parece “perseguir” el cursor del ratón.
- Añadir el código necesario para que cada vez que se haga click se visualice un asterisco en la posición del ratón.