I.E.S. PV
PALOMERAS VALLECAS

EVENTOS

UT5.- INTERACCIÓN CON EL USUARIO EN JAVASCRIPT

Susana López Luengo

Modelo de gestión de eventos

- Los eventos son mecanismos que se accionan cuando el usuario realiza un cambio sobre una página web
- El encargado de crear la jerarquía de objetos que componen una página web es DOM (Document Object Model)
- DOM también es el encargado de gestionar los eventos.

Modelo de gestión de eventos

- Los eventos se sitúan en componentes de la página indicando cual es el que queremos que desencadene la acción.
- Para poder controlar un evento se necesita un **manejador** (handler).
  - Por ejemplo, en el caso del evento click, el manejador sería *onclick*.
    ```
    <img src="mundo.jpg" onclick="funcion()">
    ```
- Los eventos tienen un manejador asociado a un código JavaScript
- Este actúa sobre los objetos DOM sin que enviemos una nueva petición al servidor modificando las propiedades de los objetos de una página en el mismo navegador.
  https://www.w3schools.com/js/js_events.asp

Modelo de gestión de eventos

Grupos de eventos de la especificación DOM

- **Eventos del ratón**. Cuando el usuario hace uso del ratón para realizar una acción. (movimiento o pulsación del ratón), puede desencadenar un evento.
- **Eventos del teclado**. Se originan cuando el usuario pulsa alguna tecla del teclado.
- **Eventos HTML**. Se producen cuando hay algún cambio en la página del navegador. También pueden ocurrir cuando existe alguna interacción entre el cliente y el servidor.
- **Eventos DOM**, o eventos de mutación, son los que se originan cuando existe algún cambio en la estructura DOM de la página.

Modelo de gestión de eventos

Manejadores de eventos. Técnicas. ***Eventos en línea***

- Manejadores de eventos como atributos XHTML

  El código se incluye en el elemento XHTML

```
<div id="caja"
onclick="document.getElementById('caja').innerHTML=
'Texto cambiado...';">
  Pincha sobre este elemento
</div>
```

- Manejadores de eventos y variable this

```
<div id="caja" onclick="this.innerHTML= 'Texto
cambiado...';">
  Pincha sobre este elemento
</div>
```

Modelo de gestión de eventos

Manejadores de eventos. Técnicas. ***Eventos en línea***

- Manejadores de eventos con funciones externas

```
<script>
   function cambiartexto(elemento)
   {elemento.innerHTML= 'Texto cambiado...';"}
</script>

   <div id="caja" onclick="cambiartexto(this)”>
     Pincha sobre este elemento
   </div>
```

Modelo de gestión de eventos

Manejadores de eventos. Técnicas. ***Modelo de eventos tradicional***

- Manejadores de eventos semánticos. Pasos a seguir:

1. Asignar un identificador único al elemento XHTML

```
<div id="caja4" >
```

2. Asignar la función externa al evento

```
document.getElementById('caja4').onclick = cambiartexto2;
```

3. Crear una función de JavaScript encargada de manejar el evento.

```
function cambiartexto2(){ this.innerHTML = "Texto modificado.”;}
```

OJO!! Para poder utilizar este método es necesario que la página esté cargada totalmente. Solución (*Verás que creamos una función anónima*):

```
window.onload = function(){
    document.getElementById('caja4').onclick = cambiartexto2;}
```

Observarás que el código resultante es más limpio...

Modelo de gestión de eventos

Manejadores de eventos. Técnicas. ***Modelo tradicional***

- Manejadores de eventos semánticos. Observaciones:

Fíjate la diferencia que hay si ponemos ()

```
document.getElementById('caja4').onclick =
cambiartexto2;
```

Asigna la función externa cambiartexto2 al evento onclick de caja4

```
document.getElementById('caja4').onclick =
cambiartexto2();
```

Ejecuta la función cambiartexto2() y guarda su resultado en la propiedad onclick de caja4. No queremos hacer eso ¿verdad?

Ver ejemplo completo

Modelo de gestión de eventos

Manejadores de eventos del ratón

- onclick: Botón izquierdo del ratón.
- ondblclick: Doble clic
- onmousedown/onmouseup: Pulsar/soltar un botón del ratón
- onmouseover/onmouseout. El puntero del ratón entra/sale fuera de un elemento
- onmouseover. El puntero del ratón entra dentro de un elemento
- onmousemove Cuando se mueve el puntero dentro de un elemento.El evento se produce continuamente mientras el ratón permanezca dentro del elemento
- onmouseout. El puntero del ratón sale del elemento

Modelo de gestión de eventos

Manejadores de eventos del ratón

- El orden de ejecución de los eventos es mousedown, mouseup, click, mousedown, mouseup, click, dblclick
- Cuando se produce un evento el **objeto event** se crea automáticamente
- Para los eventos de ratón el objeto event tiene entre otras las siguientes propiedades
  - las coordenadas del ratón (screenX, screenY),
  - nombre del evento (type)
  - elemento que origina el evento (button)

Modelo de gestión de eventos

Manejadores de eventos del teclado

- onkeydown Cuando se pulsa o mantiene pulsada una **tecla**. El evento se produce hasta que la soltemos
- onkeypress Cuando se pulsa o mantiene pulsada una **tecla alfanumérica**
- onkeyup Cuando se libera una tecla pulsada

Modelo de gestión de eventos

Manejadores de eventos del teclado

- Cuando pulsamos una tecla que corresponda a un carácter alfanumérico, la secuencia de eventos es la siguiente: keydown, keypress, keyup.
- En el caso de pulsar una tecla que *no corresponda a un carácter alfanumérico,* la secuencia de eventos es: keydown, keyup.
- Además existe la posibilidad de que dejemos una tecla pulsada.
  - Si es con carácter alfanumérico, se repiten de forma continua los eventos keydown, keypress.
  - En caso de que no sea alfanumérico se repite de forma continuada el evento keydown solamente.

Modelo de gestión de eventos

Manejadores de eventos del teclado

- Las propiedades de event para los eventos de teclado son:
  - El código numérico de la tecla pulsada (**keyCode**)

    `e.code`
  - El código unicode del carácter correspondiente a la tecla pulsada (**charCode**)

    `e.key`
  - El elemento que origina el evento (**target**)

    `e.target`
  - Otras para identificar si hemos pulsado shift (**shiftKey**),control (**ctrlKey**), alt (**altKey**) .

Modelo de gestión de eventos

Manejadores de eventos HTML

- onload La página/elemento termina de cargarse
- onabort Cuando el usuario detiene la descarga de un elemento antes de que haya terminado,
- onerror Cuando ocurre un error al cargar un elemento
- onunload Cuando el navegador cierra el documento
- onresize La ventana del navegador cambia de tamaño
- onblur Cuando se abandona un campo de formulario
- onchange Cambia el contenido un campo de formulario
- onfocus Se tiene el foco en un campo de formulario
- onselect Cuando un campo de texto es seleccionado
- onsubmit Cuando se pulsa el botón enviar
- onscroll Cuando varía la posición del scroll
- onreset Cuando se pulsa el botón reset

Modelo de gestión de eventos. DOM

Manejadores de eventos. Técnicas. ***Modelo W3C***

Elemento.addEventListener("Evento", funcion);

Observa que el evento NO lleva "on". Ej: onclick

- Ejemplo:

```
boton.addEventListener("click", function(){
    alert("Hola mundo"); });
```

- Se puede utilizar notación flecha

```
boton.addEventListener("click",() =>{
    alert("Hola mundo"); });
```

- También se pueden utilizar una expresión de la función

```
boton.addEventListener("click", saludo);
function saludo(){ alert ("Hola mundo");}
```

OJO!! Recuerda que hay que cargar la página antes de empezar a trabajar

```
window.addEventListener("load",iniciar);
```

Manejadores de eventos. Técnicas. ***Modelo W3C***

Con addEventListener podemos **asociar más de una función al mismo evento**. Ejemplo:

```
boton.addEventListener("click", saludo);
boton.addEventListener("click", pintarsaludo);
```

Elemento.removeEventListener("Evento", funcion);

- Elimina el manejador de eventos, hay que indicarle el evento y la función asociada. Ejemplo:
  `boton.removeEventListener("click",pintarsaludo);`

Después de ejecutar removeEventListener, cuando se haga click en el botón sólo se ejecutará la función saludo

Modelo de gestión de eventos. DOM

Manejadores de eventos. Técnicas. ***Modelo W3C***

Extraer el elemento que ha capturado el evento **--> evento.target**

Cuando se ejecuta

Elemento.addEventListener("Evento", prueba);

- Al definir la función se puede pasar el evento como parámetro (e)
- Para extraer el elemento al que se ha aplicado ese evento utilizamos target

```
function prueba(e){
elemento = e.target
console.log(elemento.innerHTML);  }
```

https://developer.mozilla.org/es/docs/Web/API/Event/target

https://www.w3schools.com/jsref/event_target.asp

Gestión de eventos. Bubbling

- El bubbling (burbujeo) es una fase de los eventos que significa que cuando un evento es capturado en un elemento del DOM, se va elevando hacia sus padres en orden de jerarquía hasta llegar al objeto global (window). Por lo tanto, cuando un evento ocurre en un elemento, también está ocurriendo en sus padres.

  Ejemplo

- Para evitar este efecto: e.stopPropagation();

  -- Detiene la propagación del evento en el DOM


ESCUELA PÚBLICA:
DE TOD@S
PARA TOD@S
Vallecas