UD9-Eventos

Tabla de contenido

1 Introducción

Los eventos son manejadores que nos proporciona el navegador para que cuando detecte que se produzca una acción (evento), se ejecute un código asociado a esa acción. En esta unidad trataremos los eventos existentes en Javascript y las distintas formas de manejarlos.

2 Principales eventos

A continuación, mostramos un listado de los principales eventos existentes en Javascript:

- Eventos de ratón
  - onclick: al hacer un click en el elemento.
  - ondblclick: al hacer doble click en un elemento.
  - onmousedown: al hacer clic de ratón (sin soltarlo).
  - onmouseup: al soltar el botón del ratón previamente pulsado.
  - onmouseover: al entrar encima de un elemento con el ratón.
  - onmouseout: al salir de encima de un elemento con el ratón.
- Eventos de teclado
  - onkeydown: al pulsar una tecla (sin soltarla).
  - onkeyup: al soltar una tecla pulsada.
  - onkeypress: al pulsar una tecla.
- Eventos del documento
  - onchange: al hacer un cambio en un elemento.
  - onload: al cargarse una página.
  - onunload: al descargarse una página (salir de ella).
- Eventos de formulario
  - onsubmit: al enviar los datos de un formulario.
  - onreset: al resetear los datos de un formulario.

> El total de eventos disponibles está descrito en http://www.w3schools.com/jsref/dom_obj_event.asp

3 Eventos desde elementos HTML

La forma más sencilla (aunque menos práctica para tener un código limpio y ordenado) de indicar que hay un evento asociado a un elemento HTML es **indicándolo en el propio código.**

Ejemplo 1:

```
<input type="button" value="Boton Hola mundo" onclick="alert('Hola mundo');alert('Adios');" />
```

También en lugar de ejecutar una serie de instrucciones, es posible llamar a una función predefinida.

Ejemplo 2:

```
<input type="button" value="Boton miFuncion"
onclick="miFuncion('cadenaParam1');" />
```

4 Eventos desde JavaScript a objetos HTML

Podemos asignar/modificar mediante código javascript el manejador de un evento predefinido en un objeto HTML. Supongamos que tenemos un objeto con id="miObjeto" que posee el evento "onclick" sin asignar y la función "mostrarMensaje".

Podemos referenciar al elemento HTML con getElementByID y asignar la función como manejador del evento.

Por ejemplo:

```
function mostrarMensaje(){
      alert("Hola");
}
document.getElementById("miObjeto").onclick=mostrarMensaje;
```

↗ Atención: fijaos que pone "mostrarMensaje". Así asigna la función como manejadora del evento. Si ponemos "mostrarMensaje()" no funcionará, ya que ejecutará la función y asignará su resultado al manejador.

4.1 Obteniendo información del objeto event

Cuando se crea una función como manejador, al producirse el evento el navegador automáticamente manda como parámetro un objeto de tipo event.

```
function mostrarMensaje(evento){
      alert(evento.type);
}
document.getElementById("miObjeto").onclick=mostrarMensaje;
```

Este objeto posee cierta información útil del evento que se ha producido.

Entre otros atributos:

- type: dice el tipo de evento que es ("click", "mouseover", etc...). Devuelve el nombre del evento tal cual, sin el "on". Es útil para hacer una función que maneje varios eventos.
- key: en eventos de teclado, almacena el código de tecla de la tecla afectada por el evento.
- clientX / clientY: en eventos del ratón, devuelve las coordenadas X e Y donde se encontraba el ratón, tomando como referencia al navegador.
- screenX / screenY: en eventos del ratón, devuelve las coordenadas X e Y donde se encontraba el ratón, tomando como referencia la pantalla del ordenador.

Ejemplo:

```
function mostrarMensaje(evento){
      if(evento.type=="keyup"){
            alert(evento.key);
      }
      else if(evento.type=="click"){
            alert(evento.clientX+" "+evento.clientY);
      }
}
document.getElementById("miObjeto").onclick=mostrarMensaje;
document.onkeyup=mostrarMensaje;
```

4.2 Uso del objeto this en gestión de eventos

Cuando ejecutas código dentro de un evento, existe un objeto llamado “this”.

Este objeto es una referencia al elemento que ha producido el evento.

Ejemplo, si el evento se ha producido un evento al hacer clic a una imagen con id=”miImagen”, el objeto “this” será lo mismo que poner “document.getElementById(“miImagen”);

Ejemplo sin this:

```
<div id="cont" style="width:150px; height:60px; border:thin solid
silver"
onmouseover="document.getElementById('cont').style.borderColor='black'
;"
onmouseout="document.getElementById('cont').style.borderColor='red';">

  Contenidos

</div>
```

Equivalente con “this”:

```
<div id="cont" style="width:150px; height:60px; border:thin solid
silver" onmouseover=this.style.borderColor='black';"
onmouseout="this..style.borderColor='red';">

  Contenidos

</div>
```

5 Añadir “addEventListener” a objetos para gestionar eventos

Podemos declarar eventos mediante código usando **addEventListener(evento,manejador).**

Ojo, al asignar el evento, indicamos en nombre del evento, no del manejador,por ejemplo “click” no “onclick”

Ejemplo:

```
function mostrarMensaje(evento){

      if(evento.type==”keyup”){

            alert(evento.key);
      }

      else if(evento.type==”click”){

            alert(evento.clientX+” “+evento.clientY);

      }
}
```

```
document.getElementById("miObjeto").addEventListener("click",mostrarMensaje);

document.addEventListener("keyup",mostrarMensaje);

document.getElementById("miObjeto").addEventListener("dblclick",function (){

 alert("Codigo metido directamente");

});
```

5.1 Pasar parámetros en función asociada a un addEventListener

Esta es una forma de indicar qué valores queremos que tengan los parámetros de una función asociada a un addEventListener dentro del mismo addEventListener. Esta forma se llama función anónima, en lugar de poner el nombre de la función, definimos la función en la misma llamada:

elemento.addEventListener("evento",function() { funcionExterna(argumento) });

```
//Añado los listener a los botones. Pasando el campo función de esta forma, le indicamos que al producirse el evento tiene que ejecutar la función elegida con los parámetros que pasamos


Boton1_bt.addEventListener(MouseEvent.CLICK,function(){Imprimir(1)});

Boton2_bt.addEventListener(MouseEvent.CLICK,function(){Imprimir(2)});

function Imprimir(numero){
      alert("Se ha elegido el boton " + numero);
}
```