UD 04. EVENTOS

Desarrollo web en entorno cliente
CFGS DAW

Sergio García Barea
sergio.garcia@ceedcv.es
2019/2020

Versión:191011.0922

Licencia

Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

 Importante

 Atención

 Interesante

ÍNDICE DE CONTENIDO

UD04. EVENTOS

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. EVENTOS

2.1 Principales eventos

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

- onfocus: al obtener un foco.
- onblur: al salir del foco de un elemento.
- onchange: al hacer un cambio en un elemento.
- onclick: al hacer un click en el elemento.
- ondblclick: al hacer doble click en un elemento.
- onkeydown: al pulsar una tecla (sin soltarla).
- onkeyup: al soltar una tecla pulsada.
- onkeypress: al pulsar una tecla.
- onload: al cargarse una página.
- onunload: al descargarse una página (salir de ella).
- 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.
- onsubmit: al enviar los datos de un formulario.
- onreset: al resetear los datos de un formulario.
- onselect: al seleccionar un texto.
- onresize: al modificar el tamaño de la página del navegador.

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

2.2 Manejo de eventos desde elementos XHTML

La forma más sencilla (aunque menos práctica para tener un código limpio y ordenado) de indicar que hay un evento asociado a un elemento XHTML 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');" />
```

2.3 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 se 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>
```

2.4 Asignación de eventos desde código a objetos XHTML

Podemos asignar/modificar mediante código el manejador de un evento predefinido en un objeto XHTML de una forma similar a esta. Supongamos que tenemos un objeto con id=”miObjeto” que posee el evento “onclick” sin asignar y la función “mostrarMensaje”.

Podemos referenciar al elemento XHTML como vimos en la UD 02 y asignar la función como manejador del evento como vemos en este código

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

Atención: fijaros 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.

2.5 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.
- keyCode: 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.keyCode);
      }
      else if(evento.type==”click”){
            alert(evento.clientX+” “+evento.clientY);
      }
}
document.getElementById(“miObjeto”).onclick=mostrarMensaje;
document.onkeyup=mostrarMensaje;
```

2.6 Declarando eventos desde código

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

Ejemplo:

```
function mostrarMensaje(evento){
      if(evento.type==”keyup”){
            alert(evento.keyCode);
      }
      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”);
});
```

3. DRAG AND DROP

Además de los eventos típicos tratados, existe un proceso (en el que entran en juego varios eventos) que es útil para el desarrollo de aplicaciones Web, el "Drag and Drop" (Arrastrar y soltar).

En W3Schools podéis obtener teoría y ejemplos de esta técnica https://www.w3schools.com/html/html5_draganddrop.asp

Aquí un ejemplo completo:

```
<!DOCTYPE HTML>
<html>
<head>
<script>
function allowDrop(ev) {
  ev.preventDefault();
}

function drag(ev) {
  ev.dataTransfer.setData("text", ev.target.id);
}

function drop(ev) {
  ev.preventDefault();
  var data = ev.dataTransfer.getData("text");
  ev.target.appendChild(document.getElementById(data));
}
</script>
</head>
<body>

<div id="div1" ondrop="drop(event)" ondragover="allowDrop(event)"></div>
<img id="drag1" src="img_logo.gif" draggable="true" ondragstart="drag(event)" width="336" height="69">

</body>
</html>
```

4. MATERIAL ADICIONAL

[1] Curso de Javascript en Udacity https://www.udacity.com/course/javascript-basics--ud804

5. BIBLIOGRAFÍA

[1] Referencia Javascript

http://www.w3schools.com/jsref/

[2] Referencia eventos Javascript

http://www.w3schools.com/jsref/dom_obj_event.asp

[3] Libros Web: manejo de eventos

http://librosweb.es/libro/javascript/capitulo_6.html

[4] Libros Web: objeto event

http://librosweb.es/libro/javascript/capitulo_6/obteniendo_informacion_del_evento_objeto_event.html