06 GESTIÓN DE EVENTOS

Gestión de Eventos

La interacción con el usuario de una aplicación web viene marcada por el manejo de los distintos eventos que se puedan producir. Una vez que se ha cargado una página web comienza la actividad por parte del usuario, que realiza todo tipo de acciones sobre el conjunto de elementos del documento HTML. Estas acciones se materializan en **eventos** que pueden ser tratados o controlados mediante **manejadores de eventos**.

Por ejemplo, al hacer click en un componente de una página, podremos desencadenar una acción. En el caso del evento **click**, el manejador sería **onClick**.

Las aplicaciones web creadas con el lenguaje JavaScript utilizan un modelo de **programación basada en eventos**. En este tipo de programación, los scripts se dedican a esperar a que el usuario "haga algo" (que pulse una tecla, que mueva el ratón, que cierre la ventana del navegador) y el script responde a la acción del usuario normalmente procesando esa información y generando un resultado.

Los eventos hacen posible que los usuarios transmitan información a los programas. JavaScript define numerosos eventos que permiten una interacción completa entre el usuario y las páginas/aplicaciones web. La pulsación de una tecla constituye un evento, así como pinchar o mover el ratón, seleccionar un elemento de un formulario, redimensionar la ventana del navegador, etc. JavaScript permite asignar una función a cada uno de los eventos. De esta forma, cuando se produce cualquier evento, JavaScript ejecuta su función asociada. Este tipo de funciones se denominan "event handlers" en inglés y suelen traducirse por "manejadores de eventos".

Una vez más, en la implementación de JavaScript, hay que tener en cuenta el navegador en el que se ejecute la aplicación. La incompatibilidad más importante en JavaScript se da precisamente en el modelo de eventos del navegador.

Modelo básico de eventos

Este modelo simple de eventos se introdujo para la versión 4 del estándar HTML. Aunque sus características son limitadas, es el único modelo que es compatible en todos los navegadores y por tanto, el único que permite crear aplicaciones que funcionan de la misma manera en todos los navegadores.

En este modelo, cada elemento o etiqueta HTML define su propia lista de posibles eventos que se le pueden asignar. Un mismo tipo de evento (por ejemplo, pinchar el botón izquierdo del ratón) puede estar definido para varios elementos HTML diferentes y un mismo elemento HTML puede tener asociados varios eventos diferentes.

Cada evento tiene un manejador asociado cuyo nombre se construye mediante el prefijo on, seguido del nombre en inglés de la acción asociada al evento

En el manejador podemos incluir:

una instrucción javascript o varias separadas por ;

```
<input type="text"  size="10" onclick="alert('Has entrado en la caja');">
```

o la llamada a una función

```
<input type="text"  size="10" onclick="contarClicks()">
```

A la función podemos pasarle parámetros.

Variable this

JavaScript define una variable especial llamada this que se crea automáticamente en los eventos y se puede utilizar para referirse al elemento HTML que ha provocado el evento. Por ejemplo podemos pasárselo a la función que ponemos en el manejador de eventos.

Objeto event

JavaScript define un objeto llamado event que se crea automáticamente en los eventos y se puede utilizar para referirse al evento. Por ejemplo podemos pasárselo a la función que ponemos en el manejador de eventos y luego utilizar la propiedad type de este objeto.

Ejemplo 1.html

```
<html>
<HEAD>
<SCRIPT>
<!--
cont=0;
function contarClicks(){
cont++;
document.getElementById("idclick").value=cont;
}
function quienClica(elemento){
alert ("Clica "+elemento.name);
}
function quePasa (evento){
document.getElementById("id5").innerHTML=evento.type;
}
//-->
</SCRIPT>
</HEAD>
<body topmargin="40" leftmargin="40" onclick="contarClicks()">
<h4> EJERCICIO 1 </h4>
CAJA 1<input type="text" size="10" onclick="alert('Has entrado en la caja 1')"><br><br><br>
CAJA 2<input type="text" size="10" name="caja2" onclick="contarClicks()"><br><br><br>
CAJA 3<input type="text" size="10" name="caja3" onclick="quienClica(this)"><br><br><br>
CAJA 4<input type="text" size="10" name="caja4" onclick="quienClica(this)"><br><br><br>
CAJA 5<input type="text" size="10" onmouseover="quePasa(event)" onmouseout="quePasa(event)"><br><br>
CLICKS <input type="text" id="idclick" readonly="readonly"><br><br>
<div id="id5">
Lo que pasa en la caja 5
</div>
</body>
</html>
```

http://www.w3schools.com/js/js_htmldom_events.asp

Métodos addEventListener() y removeEventListener()

addEventListener permite asignar uno, dos o más manejadores de eventos a un evento

removeEventListener permite eliminar un manejador de evento de un elemento

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
</head>
<h2>JavaScript addEventListener()</h2>

<p>Podemos añadir varios manejadores de eventos mediante addEventListener</p>

<button id="boton" onclick="alert('Ahora verás')">Púlsame</button>
<button id="botonaux" onclick="alert('Ahora verás')">Soy un auxiliar</button>

<script>
var x = document.getElementById("boton");
x.addEventListener("click", funcionUna);
x.addEventListener("click", funcionOtra);
document.getElementById("botonaux").onclick=funcionCambio;

function funcionUna() {
    alert ("Hola, soy Una.\nMe ejecuto primero ");
    }
function funcionOtra() {
    alert ("Hola, soy Otra.\nMe ejecuto después");
    x.removeEventListener("click", funcionOtra);
}
function funcionCambio() {
    alert ("Hola, soy anulador.\nAnulo el onclick inicial del elemento");
}
</script>
</body>
</html>
```

Ejemplo 2.html

Eventos del ratón: (Definidos para todos los elementos)

Evento	Manejador	Descripción
click	**onclick**	Pulsar sobre el botón izquierdo del ratón
dblclick	ondblclick	Doble click sobre el botón izquierdo del ratón
mousedown	onmousedown	Pulsar (sin soltar) un botón del ratón
mousemove	onmousemove	Mover el puntero del ratón dentro de un elemento
mouseenter	**onmouseenter**	el puntero del ratón *"entra"* en el elemento (pasa por encima del elemento)
mouseleave	**onmouseleave**	El ratón está dentro de un elemento y *sale* del elemento
mouseout	onmouseout	El ratón está dentro de un elemento y *sale* del elemento o de alguno de sus hijos
mouseover	onmouseover	El puntero del ratón *"entra"* en el elemento (pasa por encima del elemento) o de alguno de sus hijos
mouseup	onmouseup	Soltar el botón que estaba pulsado en el ratón

Ejemplo 3.html

Uso particular de onclick

onclick puede recibir un valor true o false tras haber ejecutado una función y a su vez retornar este valor que recibe. En el caso de que el valor devuelto sea false, si onclick estaba asociado a un link no se efectuará el ir a ese link. Si estaba asociado a un botón submit, no se enviarán los datos del formulario.

 ir a google
No abrirá la página

Combinando el mensaje de confirmación de window.confirm con un evento onclick podemos ofrecer al usuario la posibilidad de cancelar ciertas acciones, como la visita a una página o enviar los datos del formulario

 ir a google

Ejemplo 4.html

Usos de onmouseover y onmouseout

Su principal uso es para dar información respecto a un enlace, una imagen, etc. cuando el usuario se coloca encima de él o modificar la apariencia del elemento sobre el que se mueve el ratón.

Ejemplo 5.html

```
<div style="width:150px; height:60px; border:thin solid silver; background:#b0c4de"
        onmouseover="this.style.background='silver'"
        onmouseout="this.style.background='#b0c4de'">
Pasa por encima...
</div>
```

```
function resalta(elemento) {
switch(elemento.style.borderColor) {
case 'silver':
elemento.style.borderColor = 'black';
break;
case 'black':
elemento.style.borderColor = 'silver';
break;
}
}
<div style="width:150px; height:60px; border:thin solid silver"
onmouseover="resalta(this)" onmouseout="resalta(this)">
Sección de contenidos...
</div>
```

Hemos visto que dado un elemento HTML identificado por un id podemos acceder después desde javascript

var e = document.getElementById ("id");

Dependiendo del tipo de elemento podemos acceder y ver o cambiar determinadas propiedades

e.value	**(ej. para un <input text>)**
e.innerHTML	**(ej. para un , <p>,...)**
e.style.propiedad	**propiedades de estilo**

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

CORRESPONDENCIA ENTRE LAS PROPIEDADES DE LOS ESTILOS EN CSS Y EN JAVASCRIPT

Propiedad CSS	Propiedad DOM en Javascript
background	background
background-attachment	backgroundAttachment
background-color	backgroundColor
background-image	backgroundImage
background-position	backgroundPosition
background-repeat	backgroundRepeat
border	border
border-color	borderColor
border-style	borderStyle
border-top	borderTop
border-right	borderRight
border-left	borderLeft
border-bottom	borderBottom
border-top-color	borderTopColor
border-right-color	borderRightColor
border-bottom-color	borderBottomColor
border-left-color	borderLeftColor
border-top-style	borderTopStyle
border-right-style	borderRightStyle
border-bottom-style	borderBottomStyle
border-left-style	borderLeftStyle
border-top-width	borderTopWidth
border-right-width	borderRightWidth
border-bottom-width	borderBottomWidth
border-left-width	borderLeftWidth

border-width	borderWidth
clear	clear
clip	clip
color	color
display	display
float	cssFloat
font	font
font-family	fontFamily
font-size	fontSize
font-style	fontStyle
font-variant	fontVariant
font-weight	fontWeight
height	height
left	left
letter-spacing	letterSpacing
line-height	lineHeight
list-style	listStyle
list-style-image	listStyleImage
list-style-position	listStylePosition
list-style-type	listStyleType
margin	margin
margin-top	marginTop
margin-right	marginRight
margin-bottom	marginBottom
margin-left	marginLeft
overflow	overflow
padding	padding

padding-top	paddingTop
padding-right	paddingRight
padding-bottom	paddingBottom
padding-left	paddingLeft
position	position
text-aling	textAling
text-decoration	textDecoration
text-indent	textIndent
text-transform	textTransform
top	top
vertical-align	verticalAlign
visibility	visibility
white-space	whiteSpace
width	width
word-spacing	wordSpacing
z-index	zIndex

Los valores que pueden tomar las propiedades son los ya vistos en CSS

Fondo

background-color: nombre de color, #ffffff, rgb(0-255,0-255,0-255), transparent
background-image: url(imagen.gif), none
background-repeat: repeat, repeat-x, repeat-y, no-repeat
background-position: xpos ypos, x% y%, left/right/center top/bottom/center,..
background-attachment: scroll, fixed
background: propiedad combinada que contiene a las anteriores con el siguiente orden →color image repeat attachment position)

Tipos de letra

font-family: serif, sans-serif, cursive, fantasy, monospace
font-style: italic, oblique, normal
font-variant: small-caps, normal
font-weight: normal, bold, bolder, lighter, valor entre 100 y 900
font-size: xx-small, x-small, small, médium, large, x-large, xx-large, smaller, larger, valor
font: agrupa todos los anteriores con el orden → style weight size family

Texto

color: cualquier color
text-indent: valor de sangría
text-decoration: underline, overline, line-through, blink (parpadea), none
text-transform: capitalize, uppercase, lowercase, none
letter-spacing: valor de espacio entre caracteres
word-spacing: valor de espacio entre palabras
white-space: pre, nowrap, normal
line-height: valor de interlineado
text-align: left, right, center, justify
vertical-align: baseline, sub, supper, top, middle, bottom (sólo para tablas o elementos inline)
direction: ltr (left to right), rtl (right to left)
width: valor de anchura
height: valor de altura

display: inline, block, none

inline: displaya el elemento en línea)
block: displaya el elemento en bloque)
none: oculta

position: static, relative, absolute,fixed

visibility:hidden, visible

Caja

Se utilizan con las etiquetas div , span, p, listas, tablas...
Una caja tiene contenido, margen interior (padding), borde, margen exterior (margin)

width: auto, valor de anchura o porcentaje (no la aceptan las inline)
height: auto, valor de altura o porcentaje (no la aceptan las inline)
border-color: nombre de color, notación hexadecimal, notación modo rgb, transparent
border-top-color
border-right-color
border-bottom-color
border-left-color
border-width: thin, medium, thick, valor
border-top-width
border-right-width
border-bottom-width
border-left-width
border-style: solid, dashed, dotted, double, groove, ridge, inset, outset, hidden, none
border-top-style
border-right-style
border-bottom-style

border-left-style
border: propiedad combinada que resume las propiedades de width, style y color (en ese orden)
margin: auto, valor de longitud, porcentaje (top right bottom left)
margin-top
margin-right
margin-bottom
margin-left
padding: auto, valor de longitud, porcentaje (top right bottom left)
padding-top
padding-right
padding-bottom
padding-left

Eventos HTML:

Evento	Manejador	Descripción	Elementos para los que está definido
blur	onblur	Cuando un elemento pierde el foco	<button>, <input>, <label>, <select>,<textarea>, <body>
change	onchange	Cuando un elemento que se ha modificado pierde el foco	<input>, <select>, <textarea>
focus	onfocus	Cuando un elemento obtiene el foco	<button>, <input>, <label>, <select>,<textarea>, <body>
load	onload	La página (o imagen) se ha cargado completamente	<body> <img>
reset	onreset	Inicializar el formulario (borrar todos sus datos) al pulsar sobre un botón de tipo reset	<form>
resize	onresize	Se ha modificado el tamaño de la ventana del navegador	<body>
select	onselect	Seleccionar un texto	<input>, <textarea>
submit	onsubmit	Enviar el formulario	<form>
unload	onunload	Se abandona la página (por ejemplo al cerrar el navegador)	<body>

Una de las formas más sencillas de asegurar que cierto código se va a ejecutar después de que la página se cargue por completo es utilizar el evento **onload**

onunload no funciona en Opera ni en Chrome (si Firefox e IE)

onreset al igual que onclick puede retornar false en cuyo caso no se efectuará el inicializar el formulario.

onchange se produce cuando el usuario cambia el valor de un elemento de texto (<input type="text"> o <textarea>). También se produce cuando el usuario selecciona una opción en una lista desplegable (<select>). Sin embargo, el evento sólo se produce si después de realizar el cambio, el usuario *pasa* al siguiente campo del formulario, lo que técnicamente se conoce como que *"el campo de formulario ha perdido el foco"*.

Ejemplo 6.html

Eventos del teclado: (Definidos para elementos de formulario y <body>)

Evento	Manejador	Descripción
keydown	onkeydown	Pulsar una tecla del teclado (sin soltar)
keypress	onkeypress	Pulsar una tecla de carácter alfanumérico
keyup	onkeyup	Soltar una tecla pulsada

De todos los eventos disponibles en JavaScript, los eventos relacionados con el teclado son los más incompatibles entre diferentes navegadores y por tanto, los más difíciles de manejar. En primer lugar, existen muchas diferencias entre los navegadores, los teclados y los sistemas operativos de los usuarios, principalmente debido a las diferencias entre idiomas.
Además, existen tres eventos diferentes para las pulsaciones de las teclas (keydown, keypress y keyup). Por último, existen dos tipos de teclas: las teclas *normales* (como letras, números y símbolos normales) y las teclas *especiales* (como ENTER, Alt, Shift, etc.)
Cuando un usuario pulsa una tecla normal, se producen tres eventos seguidos y en este orden:

keydown,
keypress
y keyup.

El evento keydown se corresponde con el hecho de pulsar una tecla y no soltarla; el evento keypress es la propia pulsación de la tecla y el evento keyup hace referencia al hecho de soltar una tecla que estaba pulsada.
La forma más sencilla de obtener la información sobre la tecla que se ha pulsado es mediante el evento keypress.
La información que proporcionan los eventos keydown y keyup se puede considerar como más técnica, ya que devuelven el código interno de cada tecla y no el carácter que se ha pulsado.

Las propiedades de event para los eventos de teclado son:

keyCode	Código numérico de la tecla pulsada (13 para intro)
charCode	Código Unicode del carácter correspondiente a la tecla pulsada
key	Valor de la tecla pulsada (keypress)

Para ver a qué carácter corresponde podemos utilizar el método String.fromCharCode(evento.charCode)

Ejemplo 7.html

Para realizar una acción al salir de una caja (input text), ya sea porque han pulsado intro o porque han salido mediante el tabulador (la caja pierde el foco) podemos hacer lo siguiente:

<input type=”text” onblur=”realizarAcccion()” onkeypress=”if (event.keyCode==13) realizarAccion()”>

Información sobre los eventos de ratón

La información más relevante sobre los eventos relacionados con el ratón es la de las coordenadas de la posición del puntero del ratón.
El origen de las coordenadas siempre se encuentra en la esquina superior izquierda, pero se pueden obtener tres tipos de coordenadas:

- Respecto a la ventana del navegador
  Propiedades clientX y clientY del evento

- Respecto a la pantalla del ordenador
  Propiedades screenX y screenY

- Respecto a la propia página HTML (para cuando el usario ha hecho scroll sobre la página
  Propiedades pageX y pageY
  No definidas en Internet Explorer. Se pueden calcular sumando la posición respecto de la ventana del navegador (clientX, clientY) y el desplazamiento que ha sufrido la página (document.body.scrollLeft, document.body.scrollTop).

```
<html>
<head>
<script type="text/javascript">
function coordenadas(evento){
if (isNaN(evento.pageX))          {
          coordenadaX = evento.clientX + document.body.scrollLeft;
          coordenadaY = evento.clientY + document.body.scrollTop;
          }
else {
          coordenadaX = evento.pageX;
          coordenadaY = evento.pageY;
          }

alert ('Coordenadas respecto Navegador ['+evento.clientX+','+evento.clientY+']'+'\n'+
            'Coordenadas respecto Pantalla ['+evento.screenX +','+evento.screenY+']' +
'\n'+ 'Coordenadas respecto Página    ['+coordenadaX +','+coordenadaY+']');
}
</script>
</head>
<body onclick="coordenadas(event)" >
```

```
a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>
a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>
a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>
a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>
a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>
a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>a<br>v
</body>
</html>
```

La siguiente tabla resume los eventos más importantes definidos por JavaScript:

Evento	Descripción	Elementos para los que está definido
`onblur`	Deseleccionar el elemento	`<button>`, `<input>`, `<label>`, `<select>`, `<textarea>`, `<body>`
`onchange`	Deseleccionar un elemento que se ha modificado	`<input>`, `<select>`, `<textarea>`
`onclick`	Pinchar y soltar el ratón	Todos los elementos
`ondblclick`	Pinchar dos veces seguidas con el ratón	Todos los elementos
`onfocus`	Seleccionar un elemento	`<button>`, `<input>`, `<label>`, `<select>`, `<textarea>`, `<body>`
`onkeydown`	Pulsar una tecla (sin soltar)	Elementos de formulario y `<body>`
`onkeypress`	Pulsar una tecla	Elementos de formulario y `<body>`
`onkeyup`	Soltar una tecla pulsada	Elementos de formulario y `<body>`
`onload`	La página se ha cargado completamente	`<body>`
`onmousedown`	Pulsar (sin soltar) un botón del ratón	Todos los elementos
`onmousemove`	Mover el ratón	Todos los elementos
`onmouseout`	El ratón *"sale"* del elemento (pasa por encima de otro elemento)	Todos los elementos
`onmouseover`	El ratón *"entra"* en el elemento (pasa por encima del elemento)	Todos los elementos
`onmouseup`	Soltar el botón que estaba pulsado en el ratón	Todos los elementos
`onreset`	Inicializar el formulario (borrar todos sus datos)	`<form>`
`onresize`	Se ha modificado el tamaño de la ventana del navegador	`<body>`
`onselect`	Seleccionar un texto	`<input>`, `<textarea>`
`onsubmit`	Enviar el formulario	`<form>`
`onunload`	Se abandona la página (por ejemplo al cerrar el navegador)	`<body>`

Los eventos más utilizados en las aplicaciones web tradicionales son `onload` para esperar a que se cargue la página por completo, los eventos `onclick`, `onmouseover`, `onmouseout` para controlar el ratón y `onsubmit` para controlar el envío de los formularios.