



Carmelo José Jaén Díaz



Obtención de información de un evento

-
- C.F.G.S. DAW
 - 6 horas semanales
 - Mes aprox. de impartición: Ene
 - iPasen - cjaedia071@g.educaand.es

Índice



Objetivo

Glosario

Interacción persona - ordenador

Objetivos

Características. Usable.

Características. Visual.

Características. Educativo y actualizado.

OBJETIVO



- Descubrir elementos propios de JavaScript como los eventos
- Registrar en el front-end información de navegación, usuario, etc., mediante cookies o mediante el almacenamiento local que ofrece el estándar HTML5.

GLOSARIO



Expresión regular. Secuencia de caracteres que forman un patrón determinado, generalmente un patrón de búsqueda.

Listener. Código responsable de controlar eventos. Están a la escucha (de ahí su nombre) y, cuando ocurre un evento, se ejecuta el código que el programador haya implementado.

INTRODUCCIÓN



En las lecciones anteriores aprendimos cómo asociar *escuchadores* o *listeners* a elementos para que, cuando se produzca un evento (`click`, `mouseover`, `blur`, etc.) se ejecute una función.

Pero la mayor parte de las veces esto no es suficiente porque necesitamos saber información del evento que se ha producido: ¿qué tipo de evento ha sido? ¿sobre qué elemento se ha producido?

La función que ejecutamos cuando se produce el evento, tiene un parámetro que podríamos decir que se genera *automáticamente* y contiene toda la información del evento que necesitamos.

INTRODUCCIÓN



En esta lección aprenderás cómo **capturar toda esa información** utilizando tanto **navegadores modernos** como el ya obsoleto **Internet Explorer**.

Recuerda que, si no quieres hacer la comprobación con el modelo de eventos de Microsoft, únicamente debes eliminar la siguiente sentencia:

```
if (!e) e = window.event;
```

OBTENER LA INFORMACIÓN DE UN EVENTO

Modelo de eventos del W3C



Para estudiar cómo podemos obtener la información de un evento anteriormente mencionada, emplearemos el modelo de eventos del W3C.

Para ello, implementamos una etiqueta de título en un documento HTML.

```
<h1 id="eventos">Obtener información de un evento</h1>
```

Y a continuación, le asociaremos dos funcionalidades accionadas mediante eventos a esta etiqueta.

Recuerda implementar este código en un fichero JS externo.

OBTENER LA INFORMACIÓN DE UN EVENTO

Modelo de eventos del W3C



En primer lugar asociaremos la funcionalidad descrita en la función `manejador()`, a la etiqueta con `id="eventos"`, accionada por los eventos `mouseover` y `mouseout`:

```
document.getElementById("eventos").addEventListener("mouseover", manejador);  
document.getElementById("eventos").addEventListener("mouseout", manejador);
```

Recuerda que cuando asociamos una función a un evento, lo hacemos sin incluir paréntesis, por lo que no le pasamos ningún argumento.

La función `manejador()` sin embargo va a tener un parámetro (`e`, `event` o `evento`), ya que *automáticamente* cuando asociamos la función al evento, el navegador crea un objeto de tipo `Evento`. Este objeto almacena el evento que se ha producido (recuerda que podemos asociar varios eventos a la interacción con un elemento, como en este caso: `mouseover` y `mouseout`).

OBTENER LA INFORMACIÓN DE UN EVENTO

Compatibilidad modelo de eventos Microsoft



Esta creación *automática* del objeto **Evento** es propio de los navegadores modernos, los navegadores de Microsoft anteriores a la versión IE9 no lo hacen por lo que tenemos que incluir la sentencia:

```
if (!e) e = window.event; //Asociamos manualmente el evento que se ha  
producido al objeto Evento.
```

De esta manera, aseguramos la compatibilidad con este tipo de navegadores, aunque podríamos obviarla si no queremos contemplar esta casuística.

OBTENER LA INFORMACIÓN DE UN EVENTO

Modelo de eventos del W3C



```
switch (e.type) {  
  case "mouseover":  
    this.style.color = "purple";  
    break;  
  case "mouseout":  
    this.style.color = "yellow";  
    break;  
}
```

OBTENER LA INFORMACIÓN DE UN EVENTO

Modelo de eventos del W3C



Finalmente, la función `manejador()` resulta:

```
function manejador(e) {  
    //Valoramos la posibilidad de que se utilice un navegador de Microsoft  
    if (!e) e = window.event;  
  
    switch (e.type) {  
        case "mouseover":  
            this.style.color = "purple";  
            break;  
        case "mouseout":  
            this.style.color = "yellow";  
            break;  
    }  
}
```

OBTENER LA INFORMACIÓN DE UN EVENTO

Modelo de eventos del W3C



A continuación, vamos a estudiar la propiedad `target` del objeto `Evento`.

Para ello, vamos a definir dos etiquetas de título en nuestro documento HTML:

```
<h2 id="parrafo1">Párrafo 1</h2>
<h2 id="parrafo2">Párrafo 2</h2>
```

A las cuales, les vamos a asociar una funcionalidad, `saludo()`, accionada mediante el mismo tipo de evento, `click`.

Recuerda implementar este código en un fichero JS externo.

```
document.getElementById("parrafo1").addEventListener("click", saludo);
document.getElementById("parrafo2").addEventListener("click", saludo);
```

OBTENER LA INFORMACIÓN DE UN EVENTO

Modelo de eventos del W3C



La función `saludo()` mostrará un `alert()` cuando se haga *click* sobre la etiqueta con `id="parrafo1"` e `id="parrafo2"` de manera *personalizada*. Para ello, empleamos la propiedad `target`, que nos permite distinguir que elemento ha sido el que ha lanzado el evento.

`e.target` contiene el elemento que ha lanzado el evento, en este caso una etiqueta `<h2>`, como este elemento contiene el atributo `id` que nos permite identificar inequívocamente el elemento lo concatenamos con nuestro objeto `Evento`, resultando:

```
if (e.target.id == "parrafo1") alert("Has pulsado el primer párrafo");
```

OBTENER LA INFORMACIÓN DE UN EVENTO

Modelo de eventos del W3C



Finalmente, la función `saludo()` resulta:

```
function saludo(e) {  
    //Valoramos la posibilidad de que se utilice un navegador de Microsoft  
    if (!e) e = window.event;  
  
    if (e.target.id == "parrafo1") alert("Has pulsado el primer párrafo");  
    else if (e.target.id == "parrafo2") alert("Has pulsado el segundo  
párrafo");  
  
    alert("Has pulsado el " + e.target.id);  
    //Otra manera de obtener el elemento que ha generado el evento.  
}
```