



APRENDERAPROGRAMAR.COM

OBJETO EVENT  
JAVASCRIPT. TYPE,  
CURRENTTARGET,  
TARGET, TIMESTAMP,  
CLIENTX, SCREENX,  
PAGEX, BUTTON.  
(CU01160E)

Sección: Cursos

Categoría: Tutorial básico del programador web: JavaScript desde cero

Fecha revisión: 2029

**Resumen:** Entrega nº60 del Tutorial básico "JavaScript desde cero".

Autor: César Krall

## OBJETO EVENT. INFORMACIÓN DEL EVENTO.

Podemos añadir manejadores de eventos a los elementos del DOM y hacer que exista uno o varios manejadores de eventos para gestionar la respuesta ante dicho evento. Además en ocasiones necesitaremos información relacionada con el evento: por ejemplo en qué coordenadas de la pantalla se ha hecho click, o qué tecla ha sido pulsada. Para ello JavaScript dispone del objeto Event.



El objeto Event es un objeto predefinido de JavaScript que almacena información sobre un evento y es pasado, para cada evento que tiene lugar, como argumento a la función o funciones que gestionan el evento.

Las propiedades y métodos del objeto Event presentan la misma problemática que los eventos en relación a la estandarización y compatibilidad entre navegadores. No todos los navegadores reconocen la misma información en el objeto Event. También las propiedades del objeto Event se están ampliando continuamente, con lo cual es difícil decir qué cantidad de propiedades (información) está disponible en el objeto Event. Y también no existe una única especificación oficial de propiedades del objeto Event. Nosotros intentaremos abstraernos de esa situación y centrarnos en comprender los conceptos y conocer las propiedades y métodos del objeto event más usados y estandarizados. Comprendiendo esto, cuando necesitemos alguna información específica podremos buscar información para resolver un problema concreto. Esta es una mejor estrategia que tratar de conocer de antemano y tener un listado preciso de propiedades y métodos del objeto Event.

Las propiedades y métodos del objeto Event se resumen en la siguiente tabla. Veremos su aplicación práctica en el código de ejemplo que incluimos posteriormente:

Tipo	Nombre	Descripción aprenderaprogramar.com
Propiedades de control del evento	type	Devuelve el tipo de evento producido, sin el prefijo on (p.ej. click)
	target	Devuelve el elemento del DOM que disparó el evento (inicialmente)
	currentTarget	Devuelve el elemento del DOM que está disparando el evento actualmente (no necesariamente el elemento que disparó el evento, ya que puede ser un disparo debido a burbujeo)
Otras propiedades de control del evento	eventPhase (indica en qué fase de tratamiento de evento estamos, 1 captura, 2 en objetivo, 3 burbujeo), bubbles (booleana, indica si es un evento que burbujea o no), cancelable (booleana, devuelve si el evento viene seguido de una acción predeterminada que puede ser cancelada), cancelBubble (booleana, devuelve si el evento actual se propagará hacia arriba en la jerarquía del DOM o no).	
Propiedad temporal	timeStamp	Devuelve una medida de tiempo en milisegundos desde un origen temporal determinado.

Tipo	Nombre	Descripción aprenderaprogramar.com
Propiedades de localización del puntero del ratón	clientX, clientY	Devuelven las coordenadas en que se encontraba el puntero del ratón cuando se disparó el evento. Las coordenadas están referidas a la esquina superior izquierda de la ventana del navegador y se expresan en píxeles.
	screenX, screenY	Devuelven las coordenadas en que se encontraba el puntero del ratón cuando se disparó el evento. Las coordenadas están referidas a la esquina superior izquierda de la pantalla y se expresan en píxeles.
	pageX, pageY	Devuelven las coordenadas en que se encontraba el puntero del ratón cuando se disparó el evento. Las coordenadas están referidas a la esquina superior izquierda del documento, que pueden ser distintas a las de la ventana si el usuario ha hecho scroll sobre el documento.
Propiedad para detectar el botón del ratón pulsado	button	Normalmente empleado para el evento mouseup (liberación de botón del ratón) para detectar cuál ha sido el botón pulsado. Contiene un valor numérico: 0 para click normal (botón izquierdo), 1 para botón central (botón en el scroll), 2 para botón auxiliar (botón derecho).
Propiedades relacionadas con el teclado	Para determinar qué tecla ha sido pulsada	Lo estudiaremos por separado en la siguiente entrega del curso
Propiedades relacionadas con drag and drop	Algunas no estandarizadas	dataTransfer, dropEffect, effectAllowed, files, types
Otras propiedades varias	Otras (algunas no estandarizadas)	x, y, layerx, layery, offsetX, offsetY, wheelDelta, detail, relatedNode, relatedTarget, view, attrChange, attrName, newValue, prevValue, data, lastEventId, origin, source
Método	stopPropagation()	Detiene la propagación del evento
Método	preventDefault()	Cancela (si es posible) la acción de defecto que debería ocurrir después del evento (equivale a return false para cancelar la acción).
Otros métodos	initEvent(a, b, c)	se usa para definir eventos. No lo estudiaremos.

## EJEMPLO CON TRATAMIENTO DE MANEJADORES DE EVENTOS COMO PROPIEDADES

Escribe este código y comprueba sus resultados:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html><head><title>Ejemplo aprenderaprogramar.com</title><meta charset="utf-8">
<script type="text/javascript">
var tiempoAnterior;
```

```
function ejemploJS1 (evobject) {
alert ('La función de gestión ha recibido objeto Event con tipo: ' + evobject.type);
evobject.target.style.color = 'red'; //Cambiamos el color si se ha producido el evento
if (tiempoAnterior) {
alert('Han transcurrido ' + (evobject.timeStamp-tiempoAnterior)/1000 + ' segundos desde el click anterior');}
tiempoAnterior = evobject.timeStamp;
}
function ejemploJS2 (evobject) {alert ('¡Haz hecho doble click!'); evobject.target.style.color = 'red';}
window.onload = function() {
document.getElementById('pulsador1').onclick = ejemploJS1;
document.getElementById('pulsador2').ondblclick = ejemploJS2;}
</script>
</head>
<body><div id="cabecera"><h2>Cursos aprenderaprogramar.com</h2><h3>Ejemplo JavaScript</h3></div>
<div style="color:blue; margin:20px;" id="pulsador1" > Haz click aquí </div>
<div style="color:blue;" id="pulsador2" > Haz doble click aquí </div>
</body>
</html>
```

El resultado esperado es que cuando se haga click sobre el texto "Haz click aquí" se muestre el mensaje: La función de gestión ha recibido objeto Event con tipo: click. Además si es la segunda o posterior vez que se hace click aparecerá el tiempo que ha pasado desde el anterior click, por ejemplo: Han transcurrido 5.117 segundos desde el click anterior.

En caso de hacerse doble click sobre el texto "Haz doble click aquí", se espera que aparezca el mensaje ¡Haz hecho doble click!

Del código anterior podemos destacar lo siguiente:

evobject.target nos devuelve el elemento HTML que disparó el evento. Valiéndonos de esto, cambiamos el color del texto justo después de mostrar los mensajes correspondientes.

Hemos escrito function ejemploJS1 (evobject) donde evobject representa el objeto Event que se recibe y que es un argumento que se pasa automáticamente cuando se invoca la función al dispararse el evento. El nombre de este argumento es el que nosotros decidamos, es decir, podemos llamarlo evobject, eventoDatos, miEvento u objetoRepresentaEvento, etc., según prefiramos. Incluso podríamos escribir function ejemploJS1 (), en este caso no podríamos recuperar información del evento ya que para hacerlo necesitamos hacerlo incluyendo un parámetro en la función.

La función que detecta el evento click es distinta de la que gestiona el evento dblclick ¿Podría ser la misma? Esta pregunta tiene dos respuestas: si la pregunta se refiere a si una función puede gestionar varios eventos distintos la respuesta es sí. Si la pregunta es si una función puede distinguir si se ha hecho click o se ha hecho doble click la respuesta es, inicialmente, que no es fácil. Decimos que inicialmente que no es fácil porque entre los eventos click y dblclick hay una dificultad especial que no existe entre otro tipo de eventos. La dificultad radica en que el evento click es "parte" o "comienzo" del evento dblclick. Por ello no es evidente cómo diferenciar entre dos clicks y un doble click, o cómo detener la ejecución de la respuesta a click cuando se hace doble click. En realidad, es posible que una función diferencie entre click y doble click basándonos en el intervalo de tiempo que transcurre, pero esto requiere un poco más de código y de momento no vamos a detenernos a estudiarlo.

## EJEMPLO CON TRATAMIENTO DE MANEJADORES DE EVENTOS COMO ATRIBUTOS

En este caso la diferencia radica en que al indicar el atributo del elemento HTML debemos pasar como argumento event. La sintaxis es:

```
onNombreDelEvento = "nombreFuncion(event)"
```

Donde event representa el objeto Event que se pasa a la función. Por ejemplo podríamos escribir `onmousedown="showCoords(event)"` que indica que cuando se pulse un botón del ratón sobre el elemento que recibe el evento, debe ejecutarse la función `showCoords`, que recibirá el objeto Event con la información correspondiente gracias a que indicamos `<<event>>` como parámetro que se le pasa a la función.

## INFORMACIÓN SOBRE EVENTOS DEL TECLADO

La información sobre eventos del teclado la abordaremos en la próxima entrega del curso.

## EJERCICIO

Crea un documento HTML que conste de un título h1 con el texto `<<Posición x: - Posición y>>`, y un div con ancho 500 px y alto 400 px. Utiliza el evento `onmouseover` para mostrar dinámicamente en el título h1 las coordenadas del puntero del ratón a medida que se desplace por el elemento div.

Por ejemplo, cuando el usuario coloca el ratón sobre el div se deberá mostrar: Posición x: 244 Posición y: 188, al mover el ratón las coordenadas indicadas cambiarán a otro valor, por ejemplo Posición x: 322 Posición y: 228, y así sucesivamente.

Para comprobar si tus respuestas y código son correctos puedes consultar en los foros [aprenderaprogramar.com](http://aprenderaprogramar.com).

**Próxima entrega:** CU01161E

**Acceso al curso completo** en [aprenderaprogramar.com](http://aprenderaprogramar.com) -- > Cursos, o en la dirección siguiente:  
[http://aprenderaprogramar.com/index.php?option=com\\_content&view=category&id=78&Itemid=206](http://aprenderaprogramar.com/index.php?option=com_content&view=category&id=78&Itemid=206)