

[Todos mis módulos](#) / [Mis módulos](#) / [2023-2024](#) / [Desarrollo de aplicaciones web](#) / [Desarrollo web en entorno cliente](#)

/ [Gestión de Eventos y Formularios en JavaScript \(15%\)](#) / [Videos de la UD](#)

Videos de la UD

En este apartado encontraréis los videos referentes a la UD DWEC03



2. Gestión de Eventos

En la programación tradicional, las aplicaciones se ejecutan secuencialmente de principio a fin para producir sus resultados. Sin embargo, en la actualidad el modelo predominante es el de la programación basada en eventos. Los scripts y programas esperan sin realizar ninguna tarea hasta que se produzca un evento. El evento como tal se define como un suceso que ocurre cuando el usuario interactúa con la página, por ejemplo, si pulsa un botón. Una vez producido el evento, desde el código JS se ejecutará alguna tarea asociada a la aparición de ese evento y cuando concluye, finalmente el script vuelve al estado de espera.

JavaScript permite realizar scripts con ambos métodos de programación: secuencial y basada en eventos. Los eventos de JavaScript permiten la interacción entre las aplicaciones JavaScript y los usuarios. Cada vez que se pulsa un botón, se produce un evento. Cada vez que se pulsa una tecla, también se produce un evento. No obstante, para que se produzca un evento no es obligatorio que intervenga el usuario, ya que por ejemplo, cada vez que se carga una página, también se produce un evento.

El nivel 1 de DOM no incluye especificaciones relativas a los eventos JavaScript. El nivel 2 de DOM incluye ciertos aspectos relacionados con los eventos y el nivel 3 de DOM incluye la especificación completa de los eventos de JavaScript. Desafortunadamente, la especificación de nivel 3 de DOM se publicó en el año 2004, más de diez años después de que los primeros navegadores incluyeran los eventos.

Por este motivo, muchas de las propiedades y métodos actuales relacionados con los eventos son incompatibles con los de DOM. De hecho, navegadores como Internet Explorer tratan los eventos siguiendo su propio modelo incompatible con el estándar.

El modelo simple de eventos se introdujo en la versión 4 del estándar HTML y se considera parte del nivel más básico de DOM. Aunque sus características son limitadas, es el único modelo que es compatible con todos los navegadores y por tanto, el único que permite crear aplicaciones que funcionan de la misma manera en todos los navegadores.

EVENTOS MÁS COMUNES EN JAVASCRIPT

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.

Algunos eventos de la tabla anterior (`onclick`, `onkeydown`, `onkeypress`, `onreset`, `onsubmit`) permiten evitar la "acción por defecto" de ese evento. Más adelante se muestra en detalle este comportamiento, que puede resultar muy útil en algunas técnicas de programación.

Las acciones típicas que realiza un usuario en una página web pueden dar lugar a una sucesión de eventos. Al pulsar por ejemplo sobre un botón de tipo `<input type="submit">` se desencadenan los eventos `onmousedown`, `onclick`, `onmouseup` y `onsubmit` de forma consecutiva.

Evento	Descripción	Elementos para los que está definido
<code>onblur</code>	Un elemento pierde el foco	<code><button></code> , <code><input></code> , <code><label></code> , <code><select></code> , <code><textarea></code> , <code><body></code>
<code>onchange</code>	Un elemento ha sido modificado	<code><input></code> , <code><select></code> , <code><textarea></code>
<code>onclick</code>	Pulsar y soltar el ratón	Todos los elementos
<code>ondblclick</code>	Pulsar dos veces seguidas con el ratón	Todos los elementos
<code>onfocus</code>	Un elemento obtiene el foco	<code><button></code> , <code><input></code> , <code><label></code> , <code><select></code> , <code><textarea></code> , <code><body></code>
<code>onkeydown</code>	Pulsar una tecla y no soltarla	Elementos de formulario y <code><body></code>
<code>onkeypress</code>	Pulsar una tecla	Elementos de formulario y <code><body></code>
<code>onkeyup</code>	Soltar una tecla pulsada	Elementos de formulario y <code><body></code>
<code>onload</code>	Página cargada completamente	<code><body></code>
<code>onmousedown</code>	Pulsar un botón del ratón y no soltarlo	Todos los elementos
<code>onmousemove</code>	Mover el ratón	Todos los elementos

<code>onmouseout</code>	El ratón "sale" del elemento	Todos los elementos
<code>onmouseover</code>	El ratón "entra" en el elemento	Todos los elementos
<code>onmouseup</code>	Soltar el botón del ratón	Todos los elementos
<code>onreset</code>	Inicializar el formulario	<code><form></code>
<code>onresize</code>	Modificar el tamaño de la ventana	<code><body></code>
<code>onselect</code>	Seleccionar un texto	<code><input></code> , <code><textarea></code>
<code>onsubmit</code>	Enviar el formulario	<code><form></code>
<code>onunload</code>	Se abandona la página, por ejemplo al cerrar el navegador	<code><body></code>

MANEJADORES DE EVENTOS - EVENT HANDLERS - NATIVOS EN JAVASCRIPT

Un evento de JavaScript por sí mismo carece de utilidad. Para que los eventos resulten útiles, se deben asociar funciones o código JavaScript a cada evento. De esta forma, cuando se produce un evento se ejecuta el código indicado, por lo que la aplicación puede responder ante cualquier evento que se produzca durante su ejecución.

Las funciones o código JavaScript que se definen para cada evento se denominan *manejador de eventos* (*event handlers* en inglés) y como JavaScript es un lenguaje muy flexible, existen varias formas diferentes de indicar los manejadores:

- Manejadores como atributos de los elementos XHTML.
- Manejadores como funciones JavaScript externas.
- Manejadores "semánticos".

Gestión de Eventos desde JavaScript.

1. Gestión Básica de Eventos desde JavaScript

2. Gestión Avanzada de Eventos - Utilización del "addEventListener"

3. Otros Eventos Interesantes - Ratón y Teclas

MANEJADORES DE EVENTOS - EVENT HANDLERS - USANDO LIBRERÍAS EXTERNAS - JQUERY

Otra forma muy común de gestionar eventos en JavaScript es a través del uso de librerías externas. En este caso, utilizaremos JQuery, ya que es la biblioteca más utilizada en JavaScript. Para ello, os dejo una serie de tres videos donde veréis cómo podréis cargar la librería, seleccionar los elementos del DOM y agregarles eventos de manera muy sencilla a través de funciones de tipo "callback"

Gestión de Eventos utilizando JQuery.

1. JQuery - Cargar librería desde JS

2. JQuery - Uso de Selectores

3. JQuery - Gestión de Eventos



Navega por la unidad

◀ UD03 Encuesta valoración

Tu profesor/a ►

Contacta con nosotros:

Dirección: Paseo de Ubarburu 39, Edificio EnerTIC of. 206 · Donostia San Sebastián

Telefono : 945 567 953

E-mail: info@birt.eus

Twitter: @Birt_LH