



Ministerio de  
**TRABAJO, EMPLEO  
Y SEGURIDAD SOCIAL**



*Paraguay  
de la gente*

# Eventos-Javascript

---

En el caso de la Web, los eventos se desencadenan dentro de la ventana del navegador y tienden a estar unidos a un elemento específico que reside en ella — podría ser un solo elemento, un conjunto de elementos, el documento HTML cargado en la pestaña actual o toda la ventana del navegador. Hay muchos tipos diferentes de eventos que pueden ocurrir, por ejemplo:

- ❖ El usuario hace clic con el mouse sobre un elemento determinado o coloca el cursor sobre un elemento determinado.
- ❖ El usuario presiona una tecla en el teclado.
- ❖ El usuario cambia el tamaño o cierra la ventana del navegador.
- ❖ Una página web termina de cargar.
- ❖ Un formulario se envía
- ❖ Un video se reproduce, pausa o finaliza la reproducción.
- ❖ Un error ocurre.

# Definición

---

Los eventos definen lo que hace el usuario en la página, o en el elemento de la página al que se le aplica. Cualquier acción que haga el usuario es un evento: mover el raton, hacer click o doble click, pulsar una tecla, entrar con el ratón en un elemento, seleccionarlo con el ratón o con la tecla TAB, etc. Incluso el hecho de acabar de cargarse la página es un evento.

En Javascript se puede detectar cuando se produce un determinado evento, y aplicarle un código para que al producirse el evento se ejecute ese código. De esta forma se interactúa con el usuario, el cual al provocar el evento desencadena la ejecución del código javascript asociado.

# Lista de Eventos

---

El nombre de cada evento se construye mediante el prefijo `on`, seguido del nombre en inglés de la acción asociada al evento. Así, el evento de pinchar un elemento con el ratón se denomina `onclick` y el evento asociado a la acción de mover el ratón se denomina `onmousemove`.

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.

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.

# Manejadores de Eventos

---

## MANEJADOR DE EVENTOS O EVENT HANDLER COMO ATRIBUTO DE ELEMENTO HTML. CONFIRM.

¿Cómo indicar en el código que una serie de instrucciones deben ejecutarse cuando tiene lugar un evento?

La forma más tradicional sería la consistente en añadir un atributo a la etiqueta HTML que recibe el evento, también llamado “captura de eventos en línea”:

`<etiqueta nombreEvento="acción a ejecutar"> ... </etiqueta>`

Donde acción a ejecutar es un fragmento de código JavaScript, que en caso de comprender varias sentencias deben ir separadas por punto y coma.

`<button onclick="document.getElementById('demo').innerHTML=Date()">La hora es?</button>`

Archivo codigo.js	Archivo index.html	Vista Navegador
	<pre>&lt;!DOCTYPE html&gt; &lt;html&gt; &lt;body&gt;  &lt;h1&gt;Evento-Forma Tradicional&lt;/h1&gt; &lt;p&gt;No es la forma mas adecuada&lt;/p&gt; &lt;p&gt;No se utiliza un archivo codigo.js aparte&lt;/p&gt; &lt;p&gt;Para saber la hora actual, pulsa e boton!!!&lt;/p&gt;  &lt;button onclick="document.getElementById('demo').innerHTML=Date()"&gt;La hora es?&lt;/button&gt;  &lt;p id="demo"&gt;&lt;/p&gt;  &lt;/body&gt; &lt;/html&gt;</pre>	<div><h2>Evento-Forma Tradicional</h2><p>No es la forma mas adecuada</p><p>No se utiliza un archivo codigo.js aparte</p><p>Para saber la hora actual, pulsa e boton!!!</p><div>La hora es?</div><p>Mon Jul 20 2020 14:40:40 GMT-0400 (hora estándar de Paraguay)</p></div>



# Manejadores de eventos como funciones externas

---

La definición de los manejadores de eventos en los atributos HTML es el método más sencillo pero menos aconsejable de tratar con los eventos en JavaScript. El principal inconveniente es que se complica en exceso en cuanto se añaden algunas pocas instrucciones, por lo que solamente es recomendable para los casos más sencillos.

Si se realizan aplicaciones complejas, como por ejemplo la validación de un formulario, es aconsejable agrupar todo el código JavaScript en una función externa y llamar a esta función desde el elemento HTML.

La llamada a la función se realiza de la forma habitual, indicando su nombre seguido de los paréntesis y de forma opcional, incluyendo todos los argumentos y parámetros que se necesiten.

```
<button onclick="mostrarFecha()">La hora es?</button>
```

Archivo codigo.js	Archivo index.html	Vista Navegador
<pre>function mostrarFecha() {     document.getElementById("demo").innerHTML =     Date(); }</pre>	<pre>&lt;!DOCTYPE html&gt; &lt;html&gt; &lt;body&gt;  &lt;h1&gt;Evento- Forma Fichero Externo&lt;/h1&gt;  &lt;p&gt;La mas utilizada&lt;/p&gt;  &lt;p&gt;Para saber la hora actual, pulsa e boton!!!&lt;p&gt;  &lt;button onclick="mostrarFecha()"&gt;La hora es?&lt;/button&gt;  &lt;script src=codigo.js&gt;&lt;/script&gt;  &lt;p id="demo"&gt;&lt;/p&gt;  &lt;/body&gt; &lt;/html&gt;</pre>	<div><h2>Evento- Forma Fichero Externo</h2><p>La mas utilizada</p><p>Para saber la hora actual, pulsa e boton!!!</p><div>La hora es?</div></div>

# Manejadores de eventos con el `addEventListener()`

---

El `addEventListener()` es una función incorporada en JavaScript que tiene el evento para escuchar, y un segundo argumento que se llamará cada vez que el acontecimiento descrito es despedido. Se puede agregar cualquier cantidad de controladores de eventos a un solo elemento sin sobrescribir los controladores de eventos existentes.

Sintaxis:

```
element.addEventListener (evento, escucha);
```

**Parámetros:**

**evento:** el evento puede ser cualquier evento válido de JavaScript. Los eventos se usan sin el prefijo "on", como "clic" en lugar de "onclick" o "mousedown" en lugar de "onmousedown".

**oyente (función de controlador):** puede ser una función de JavaScript que responda al evento.

Archivo codigo.js	Archivo index.html	Vista Navegador
<pre>document.addEventListener("click", function(){  document.getElementById("demo").inne rHTML = "Hola Mundo, prueba del addEventListener!"; });</pre>	<pre>&lt;!DOCTYPE html&gt; &lt;html&gt; &lt;body&gt;  &lt;h1&gt;Forma- addEventListener&lt;/h1&gt;  &lt;p&gt;Este ejemplo utiliza el método addEventListener() para adjuntar un evento de clic la pagina.&lt;/p&gt;  &lt;p&gt;Haga clic en cualquier parte de la pagina.&lt;/p&gt;  &lt;p id="demo"&gt;&lt;/p&gt;  &lt;script src="codigo.js"&gt; &lt;/script&gt;  &lt;/body&gt; &lt;/html&gt;</pre>	<div> <b>Forma- addEventListener</b> <p>Este ejemplo utiliza el método addEventListener() para adjuntar un evento de clic la pagina.</p> <p>Haga clic en cualquier parte de la pagina.</p> <p>Hola Mundo, prueba del addEventListener!</p> </div>

# Ejemplo con addEventListener()

---

❖ Click

❖ Mouseover

❖ Mouseout

## Archivo codigo.js

```
const x = document.getElementById("clickIt");
const y = document.getElementById("hoverPara");

x.addEventListener("click", RespondClick);
y.addEventListener("mouseover", RespondMouseOver);
y.addEventListener("mouseout", RespondMouseOut);

function RespondMouseOver() {
    document.getElementById("effect").innerHTML +=
        "MouseOver Event-Encima del texto" + "<br>";
}

function RespondMouseOut() {
    document.getElementById("effect").innerHTML +=
        "MouseOut Event-Fuera del Texto" + "<br>";
}

function RespondClick() {
    document.getElementById("effect").innerHTML +=
        "Evento Click" + "<br>";
}
```

## Archivo index.html

```
<!DOCTYPE html>
<html>

<body>
    <button id="clickIt">Haga Click
    aqui!!!</button>

    <p id="hoverPara">Pasar el cursor
    encima de este texto !</p>

    <b id="effect"></b>

    <script> codigo.js</script>
</body>

</html>
```

## Vista Navegador

Haga Click aqui!!!

Pasar el cursor encima de este texto !

**Evento Click**

**MouseOver Event-Encima del texto**

**MouseOut Event-Fuera del Texto**

**MouseOver Event-Encima del texto**

**MouseOut Event-Fuera del Texto**

**MouseOver Event-Encima del texto**

**MouseOut Event-Fuera del Texto**

# Fuentes Consultadas

<https://www.aprenderaprogramar.com/>

<https://uniwebsidad.com/libros/javascript>

[https://www.w3schools.com/js/js\\_events.asp](https://www.w3schools.com/js/js_events.asp)

<https://www.geeksforgeeks.org/javascript-addeventlistener-with-examples/>