EVENTOS AVANZADOS JAVASCRIPT

[**Evento onLoad (página web cargada completamente)**](#_igr24td9auxs) **2**

[**Compartir manejador de eventos**](#_5t7ou7hueuti) **3**

[**Obtener información de los eventos**](#_wbzagpp626if) **5**

[Ejemplo evento onmousemove](#_r41xtea5pycf) 5

[Ejemplo eventos onkeydown , onkeyup y onkeypress](#_8qsigihb6jh0) 5

[**Crear nuevos elementos dinámicamente y asignar eventos**](#_xesg5krelaec) **7**

# Evento onLoad (página web cargada completamente)

El evento onLoad se desencadena cuando se ha cargado completamente la página, por lo que estará creado la estructura del DOM y podremos ejecutar código JS de forma segura para manipularlo, pues el navegador habrá creado todos los objetos correspondientes a los elementos HTML de nuestra página.

Podemos utilizar tres maneras diferentes para usar el evento onLoad:

1. **Mediante el atributo onload de la etiqueta <body> (Ver código ej\_onload\_v1.html**)

<body onload="iniciar()">

<script>

// Evento onload usando atributo onload en etiqueta body

function iniciar(){

let dato=document.getElementById("nombre").value;

alert(`Página cargada completamente \n Dato en cuadro de texto: ${dato} `);

}

</script>

<input type="text" name="nombre" id="nombre" value="Javier">

</body>

1. **Mediante la propiedad onload del objeto window (Ver código ej\_onload\_v2.html)**

<body>

<script>

// Evento onload usando propiedad onload en objeto window

window.onload = iniciar;

function iniciar(){

let dato=document.getElementById("nombre").value;

alert(`Página cargada completamente \n Dato en cuadro de texto: ${dato} `);

}

</script>

<input type="text" name="nombre" id="nombre" value="Javier">

</body>

1. **Añadiendo el evento mediante addEventListener (Ver código ej\_onload\_v3.html)**

<body>

<script>

// Evento onload usando addEventListener

function iniciar(){

let dato=document.getElementById("nombre").value;

alert(`Página cargada completamente \n Dato en cuadro de texto: ${dato} `);

}

addEventListener('load', iniciar);

</script>

<input type="text" name="nombre" id="nombre" value="Javier">

</body>

# Compartir manejador de eventos

Cuando se ejecuta el código de un manejador de eventos que está asignado a un solo control (por ejemplo hacer click en un botón) sabremos perfectamente quién ha desencadenado el evento (el botón) y de qué tipo de evento se trata (onclick).

Pero JavaScript permite compartir un mismo manejador de eventos. En dicho caso, cuando se ejecute el manejador de eventos podremos rescatar el control y el tipo de evento del objeto evento (argumento del manejador de eventos):

* ev.target ⇒ Control que ha desencadenado el evento
* ev.type ⇒ Tipo de evento

Veamos un ejemplo en que disponemos de un solo manejador de eventos llamado manejador\_eventos que es compartido por los siguientes controles:

* Click en boton1
* Click en boton2
* Seleccionar elemento en el desplegable ciudades

<body>

<script>

function inicio(){

// El evento hacer click sobre cualquiera de los botones 1,2 y 3 será gestionado por el mismo manejador de eventos

document.getElementById('boton1').addEventListener('click',manejador\_eventos);

document.getElementById('boton2').addEventListener('click',manejador\_eventos);

document.getElementById('ciudades').addEventListener('change',manejador\_eventos);

}

function manejador\_eventos(ev){

let control= ev.target;

let tipoevento = ev.type;

alert(`Evento desencadenado por control ${control.value} y es de tipo ${tipoevento}`);

}

addEventListener('load',inicio);

</script>

<h1>Asignar varios eventos a un único manejador de eventos</h1>

<input type="button" id="boton1" value="Botón 1">

<input type="button" id="boton2" value="Botón 2">

<select name="ciudades" id="ciudades">

<option value="Utrera">Utrera</option>

<option value="Los Palacios">Los Palacios</option>

</select>

</body>

(Ver código en **asignar\_varios\_eventos\_a\_un\_unico\_eventhandler.html** )

# Obtener información de los eventos

Existen algunos eventos que incorporan cierta información que nos será útil rescatar cuando se desencadenan.

## Ejemplo evento onmousemove

Un ejemplo es el evento onmousemove, que se dispara cuando nos movemos con el ratón por la pantalla. En dicho caso podemos rescatar las coordenadas X e Y de la posición del ratón. Para ello tendremos que añadir un argumento en el manejador de eventos (será un objeto que contiene información del evento) , y dentro de la función rescataremos las propiedades clientX y clientY a partir del objeto evento:

// Manejador de eventos

function myFunction(evt) { // evt contiene información relativa al evento

var x = evt.clientX;

var y = evt.clientY;

var coor = "Coordenadas: (" + x + "," + y + ")";

document.getElementById("demo").innerHTML = coor;

}

(Ver código **ej\_obtener\_info\_onmousemove.html** )

## Ejemplo eventos onkeydown , onkeyup y onkeypress

Cuando pulsamos las teclas se desencadena varios eventos:

* **onkeydown**: cuando presionamos una tecla
* **onkeyup**: cuando dejamos de presionar la tecla
* **onkeypress**: cuando presionamos y soltamos un tecla (solo se puede usar con las teclas que representan datos de tipo alfanumérico, como “a”, “1”, etc.)

Asociados a estos eventos disponemos de las siguientes propiedades:

* **keyCode**: es un código numérico que se corresponde con cada una de las teclas (incluso las teclas especiales como F1, F2, Alt, Ctrl, tienen un código asociado)
* **charCode**: cuando la tecla pulsada se corresponde con un valor alfanumérido (“a”, “b”, “3”, “.”, etc.) podremos rescatar su valor con dicha propiedad.

Veamos un ejemplo de uso de estos tres eventos y las propiedades. En el siguiente ejemplo además se está utilizando un solo manejador de eventos, que es compartido por los tres eventos onkeydown, onkeyup y onkeypress. Dentro del manejador de eventos podremos saber cuál de ellos lo ha desencadenado examinado la propiedad type del objeto evento (el cual lo tenemos disponible a través del argumento de la función).

<script>

window.onload = function() {

document.onkeydown = muestraInformacion;

document.onkeyup = muestraInformacion;

document.onkeypress = muestraInformacion;

}

// Manejador de eventos compartido por varios eventos (onkeydown, onkeyup, onkeypress)

function muestraInformacion(ev) {

let mensaje = "Tipo de evento: " + ev.type + "<br>" + // Con ev.type podemos rescatar qué evento se ha producido (recordemos que es un manejador de eventos compartido)

"Propiedad keyCode: " + ev.keyCode + "<br>" +

"Propiedad charCode: " + ev.charCode + "<br>" +

"Carácter pulsado: " + String.fromCharCode(ev.charCode);

document.getElementById("info").innerHTML += "<br>--------------------------------------<br>" + mensaje

}

</script>

<div id="info"></div>

(Ver código en **ej\_obtener\_info\_onkeyup-onkeydown-onkeypress.html**)

# Crear nuevos elementos dinámicamente y asignar eventos

JavaScript permite crear nuevos elementos de manera dinámica (tal y como hemos visto en el tema anterior del DOM). Además de crear los nuevos elementos (párrafos, botones, etc.) podremos asignarle de manera dinámica el manejador de eventos.

Veamos un ejemplo de cómo crear varios párrafos de manera dinámica y añadirles un manejador de eventos compartido. Dentro del manejador de eventos podremos obtener información de qué control desencadenó el evento con ev.target y el tipo de evento con ev.type (siendo ev el objeto de tipo evento que tenemos disponible en el argumento del manejador de eventos).

<body>

<script>

function iniciar(){

let contenedor= document.getElementById("contenedor");

// Creamos cuatro párrafos de forma dinámica y le asignamos un manejador de eventos compartido

for (let i=0 ; i < 4; i++){

let parrafo=document.createElement("p");

parrafo.appendChild(document.createTextNode(`Párrafo ${i} creado dinámicamente`));

parrafo.id=`parrafo${i}`;

contenedor.appendChild(parrafo);

parrafo.addEventListener('click',manejador); // Añadimos el manejador de eventos al nuevo párrafo creado

}

}

//Manejador de eventos compartido por todos los párrafos creado de manera dinámica

function manejador(ev){

alert(`Control:${ev.target.id} \n Tipo de evento:${ev.type}`);

}

addEventListener('load',iniciar);

</script>

<h1>Haz clic sobre los párrafos que han sido creados dinámicamente</h1>

<div id="contenedor"></div>

</body>

(Ver código en **ej\_crear\_eltos\_y\_asinar\_eventos\_dinamicamente.html** )