

Eventos en JS

EVENTOS

- Definen las posibles interacciones con los elementos de la web.

MOUSE EVENTS

onclick, ondblclick,
onmousedown, onmousemove,
onmouseover, onmouseout,
onmouseup

KEYBOARD EVENTS

- onkeydown, onkeypress, onkeyup

OBJECT EVENTS

- onload, onresize, onscroll, onunload

FORM EVENTS

- onblur, onfocus, onselect, onchange, onsubmit, onreset

Capturar un evento en el HTML

ESTABLECER CAPTURADOR EN CÓDIGO

```
<div id="divAzul" onclick="ejecutaEsto();" > </div>  
<body onload="ejecutaEsto()">
```

AÑADIR FUNCIONES A UN EVENTO EN CÓDIGO

```
<body onload="ejecutaEsto(); estoTambien();" >
```

Capturar un Evento por JS

- Utilizando una función sin variable

```
window.onload = function(){ };
```

- Definiendo la función en una variable

```
const miFuncion = function ejecutaEsto(){ //código a ejecutar };
```

```
const miFuncion2 = function estoTambien(){ //código a ejecutar2 };
```

```
window.onload = miFuncion ;
```

```
window.onload = miFuncion2 ; //sustituye a miFuncion
```

AÑADIR CAPTURADOR POR JAVASCRIPT

```
window.addEventListener("load", miFuncion);
```

```
window.addEventListener("load", miFuncion2 );
```

Formularios : Elementos básicos

`<input type="text">`

`<form type="password" action="codigo.php" method="get" name="formularioEjemplo">`

`<input type="password">`

`<input type="radio" name="sex" value="male">`
`<input type="radio" name="sex" value="female">`

`<input type="radio" name="edat" value=">18">`
`<input type="radio" name="edat" value="<18">`

`<input type="checkbox" name="vehicle" value="bike"> I have a bike
`
`<input type="checkbox" name="vehicle" value="car"> I have a car
`

`<select>`
`<option value="1">uno</option>`

`<input type="reset" value="Submit"/>`
`<button type="reset" value="Reset">`

First name:

Password:

☐ Male ☐ Female

☐ Mayor 18 años ☐ Menor 18 años

☐ I have a bike

☐ I have a car

¿Cuántos días a la semana sueles encender el ordenador?

Acceder a los Inputs con JS

`<form name="miFormulario" >` → `let formulario = document.forms["miFormulario"];`

INPUT TEXT / TEXTAREA / PASSWORD / EMAIL ...

`<input type="text" name="texto" />` → `let text = formulario["texto"].value;`

Acceder a Checkbox y Radio input

```
Hombre<input type="radio" name="sex" value="male">  
Mujer<input type="radio" name="sex" value="female">
```

```
const radios = formulario["sex"]; //array con los radio de name="sex"  
const radio="";  
for(let k =0; k< radios.length; k++){  
    if(radios[k].checked == true){  
        radio= radios[k].value;  
        break;  
    }  
}
```

```
<input type="checkbox" name="vehicle" value="bike"> I have a bike <br />  
<input type="checkbox" name="vehicle" value="car"> I have a car <br />
```

```
const checkboxes = formulario["vehicle"]; //array con los checkbox  
de name="vehicle"  
for(let k =0; k < checkboxes.length; k++){  
    if(checkboxes[k].checked== true){  
        alert("selected: "+checkboxes[k].value);  
    }  
}
```

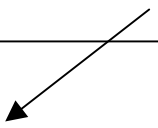
Validar un Formulario con JS

- Para validar un formulario antes de ser enviado, se debe realizar `onsubmit`. Si se retorna un `false`, no se enviará.

```
<form action="codigo.php" onsubmit="return validaForm();">  
  Nombre <input type="text" name="name" onblur="validaNom()" >  
  <input type="submit" >  
</form>
```

```
function validaForm(){  
  let formulario =  
    document.forms["miFormulario"]; let text =  
    formulario["name"].value;  
  if(text=="pedro"){  
    return true;  
  }  
  return false;  
}
```

Este formulario solo se enviará cuando
el nombre sea "pedro"



}

Cookies

- Variables vinculadas a un sitio web almacenadas en el navegador y el servidor.
- Se envían del navegador al servidor y viceversa.
- Se puede acceder siempre desde el servidor, y en ocasiones desde el navegador con JS.

Crear / Modificar /Borrar / Leer una Cookie de forma nativa

```
document.cookie='nom=James';
```

```
document.cookie='apellido=Bond';
```

```
document.cookie='nom=Jaime';
```

```
document.cookie='nom=James;expires=Thu, 2 Aug 2020 20:47:11 UTC;' ;
```

```
textoGuardado=hola; nom=Jaime; apellido=Bond
```


Crear/Modificar una Cookie

- Podemos utilizar el siguiente código para crear/modificar el valor de una cookie

```
function setCookie(cname,cvalue,exdays) {  
    let d = new Date();  
    d.setTime(d.getTime()+(exdays*24*60*60*1000));  
    let expires = "expires="+d.toGMTString();  
    document.cookie = cname + "=" + cvalue + "; " + expires;  
}
```

- ***cname*** es el nombre de la cookie a guardar y ***cvalue*** es el nombre del valor.
- ***exdays*** es el número de días que va a vivir la cookie. Con un valor negativo la cookie se borrará

Ejemplo de utilización:

```
function guardarValor(){  
    let texto = window.prompt("indica un valor");  
  
    setCookie("textoGuardado",texto,5);  
}
```

Leer una Cookie

- Podemos utilizar el siguiente código para crear/modificar el valor de una cookie

```
function getCookie(cname) {  
  let name = cname + "=";  
  let ca = document.cookie.split(';');  
  for(let i=0; i<ca.length; i++) {  
    let c = ca[i].trim();  
    if (c.indexOf(name)==0) return c.substring(name.length,c.length);  
  }  
  return "";  
}
```

Leer una Cookie (w3schools)

```
function obtineValor(){  
  let texto =  
    getCookie("textoGuardado");  
} console.log(texto)
```

EXTRA

Más sobre eventos y formularios

Añadir un Evento por JS

```
addEvent(window, "load" , miFuncion1);  
addEvent(window, "load" , miFuncion2);
```

- Utilizando la siguiente función. (para utilizarla debe estar en nuestro .js)

```
/**Cross-browser remove Event  
 * @param {Object} objeto del que quitar el evento  
 * @param {Object} tipo de evento (click, mouseover,..)  
 * @param {Object} funcion a quitar  
 */  
function removeEvent(obj, evType, fn){  
    if(obj.removeEventListener){  
        obj.removeEventListener(evType, fn);  
    }else if(obj.detachEvent){  
        obj.detachEvent('on'+evType, obj[fn.toString()+evType]);  
        obj[fn.toString()+evType]=null;  
    }else{  
        obj['on'+evType]=function(){};  
    }  
}
```

Quitar un Evento por JS

```
removeEvent(window, "load" , miFuncion1);
```

- Utilizando la siguiente función. (para utilizarla debe estar en nuestro .js)

```
/**Cross-browser remove Event
 * @param {Object} objeto del que quitar el evento
 * @param {Object} tipo de evento (click, mouseover,..)
 * @param {Object} funcion a quitar
 */
function removeEvent(obj, evType, fn){
    if(obj.removeEventListener){
        obj.removeEventListener(evType, fn);
    }else if(obj.detachEvent){
        obj.detachEvent('on'+evType, obj[fn.toString()+evType]);
        obj[fn.toString()+evType]=null;
    }else{
        obj['on'+evType]=function(){};
    }
}
```

Acceder a Select input

```
<select name="listado" value="-1" multiple>
  <option value="1"> uno</option>
  <option value="2"> dos</option>
  <option value="3"> tres</option>
</select>
```

```
//comprobamos que se han seleccionado dos elementos
const selector = formulario["listado"];
const indexSeleccionado=selector.selectedIndex; //primer elemento
seleccionado const optionsArr = selector.options;//array con las
opciones

if( indexSeleccionado ==-1) {/*no se ha seleccionado option.*/  }

for(let k =0; k < optionsArr.length; k++){ //recorremos el array de
opcionesif(optionsArr[k].selected == true){
    }      alert("selected: "+optionsArr[k].text);
}
```

Crear Expresiones Regulares

pqo @gmail.com

Guille@hotmail.com

Empieze por :

FIN

$^ [A-z0-9]^+$

Cualquier texto
sin espacios

@

Una @

$[A-z0-9]^+$

Cualquier texto
sin espacios

.

Un punto

$[A-Za-z0-9]\{2,3\} \$$

De dos a tres
caracteres sin
espacios

$^$ inicio , $\$$ final

$[A-z]$ rango permitido , $^[^A-z]$ rango prohibido , $(a|b|c)$ alternativas

$\{1,5\}$ de 1 a 5 repeticiones , $*$ de 0 a infinitas repeticiones , $+$ de 1 a infinitas repeticiones

- Utiliza los elementos para crear la expresión regular que concuerde con un email:

$^ + @ .$

$[A-Za-z0-9] \{2,3\}$

Validar una Expresión Regular

Objeto RegExp.test(texto)

```
let texto = 'nom@gmail.com';

//creamos un objeto de expresión regular
let patt1=new RegExp("^[A-z]{1,5}$");
if ( patt1.test(texto) ){ //comprobamos si se hace.matching con test
    alert ('Nombre valido');
} else {          alert('Nombre  invalido');    }
```

texto.match(Expresion Regular)

```
let texto = 'cadena de texto a validar';

//creamos la expresión regular como una string
let RegExPattern = /^[^0-9][A-z]{1,5}$/;

if ( texto.match(RegExPattern) ) { //comprobamos el matching con match
    alert('Password valido');
} else {          alert('Password invalido');    }
```