Vincular JS y HTML

Archivo externo	`<script type="t ext /ja vas cri pt" src="Co dig o2.j s"> - </s cri pt>`
En el mismo archivo, dentro de body o head	`<script type="t ext /ja vas cri pt"> </s cri pt>`
El código javascript puede estar insertado en atributos de etiquetas HTML.	<p onclick="alert('Insertando javascript si haces click aquick')">
Inserción en código HTML	<script type="text/javascript"> // Código javascript </script>

Depuración de errores

Click der. en navegador, Inspeccionar elemento, Console o Sources	
Dentro de concidionales, bucles, otras instrucciones...	alert(variableConValor);
	console.log("instrucción", texto);

Sentencias de control

Switch Case switch(nota){ case 1: case 2: alert("Muy deficiente"); break; case 3: alert("Insuficiente"); break;}

Print

Ventana de alerta	alert(" ");

Funciones de cadena

Reemplazar carácteres	cadena = cadena.replace(/,/g, '');

Funciones de array

Dar la vuelta a los elementos del array	array = array.reverse();
Convertir elementos del array en una cadena	array = array.join();

Objetos

var objeto_mio = new Object();	Creaci'on de un objeto
var animal1 = Object.create(Animal);	
objeto_mio.propiedadtal; objeto_mio.metodocual([parámetros o no]);	Acceso a propiedad del objeto
tragaperras['saldo'] = 10;	
tragaperras = {'saldo': '10, 'puntos' : 1}	
var myObj = { myMethod: function(params) { // ...do something }	Métodos de objeto
myOtherMethod(params) { // ...do something else }	
object.methodname(params);	Llamar al método de objeto

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Working_with_Objects

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

Clase Date	
Las fechas se indexan a 0!	
var miFecha = new Date()	Creación de objeto Date con la fecha por defecto (la actual)
miFecha = new Date(año,mes,dia,hora,minutos,segundos, milisegundos)	Creación de fecha date con instancia por parámetro
	los parámetros son siempre numéricos
Métodos get	
f.getDate();	Devuelve el día del mes. Número entre 1 y 31.
getDay()	Devuelve el día de la semana. Entre 0 (domingo) y 6 (sábado)
getFullYear()	Retorna el año con 4 dígitos.
getHours()	Retorna la hora. Entre 0 y 23.
getMilliseconds()	Devuelve los milisegundos entre O y 9999
getMinutes()	Devuelve los minutos. Entre 0 y 59.
getMonth()	Devuelve el mes. Entre 0 (enero) y 11 (diciembre)
getSeconds()	Devuelve los segundos. Entre 0 y 59.
getTime()	Devuelve los milisegundos transcurridos entre el día 1 de enero de 1970 y la fecha correspondiente al objeto al que se le pasa el mensaje.
var unixTimeZero = Date.parse('01 Jan 1970 00:00:00 GMT');	Analiza una fecha y devuelve el número de milisegundos pasados desde el 1 de enero de 1970 hasta la fecha analizada
Métodos set	
var event = new Date('August 19, 1975 23:15:30'); event.setDate(24);	Actualiza el día del mes.
	`f.setDate( f.getD ate() + 1 ) ;`
setFullYear()	Cambia el año de la fecha al número que recibe por parámetro.
setHours()	Actualiza la hora 0-24
setMilliseconds()	Establece el valor de los milisegundos.
setMinutes()	Cambia los minutos.
setMonth()	Cambia el mes (el mes empieza por 0).
setSeconds()	Cambia los segundos.
var event1 = new Date('July 1, 1999'); var event2 = new Date(); event2.setTime(event1.getTime());	Actualiza la fecha completa. Recibe un número de milisegundos desde el 1 de enero de 1970.
Otros	

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

Clase Date (cont)	
toDateString()	Convierte la fecha del objeto Date en una cadena de caracteres.
toTimeString()	Convierte la parte de tiempo de un objeto Date en una cadena

Clase Math	
-Math no es un constructor. No se usa como método para objetos ya creados.	
Funciones	
PI	
E	constante de Euler
Métodos	
return Math.abs(a - b); console.log(difference(3, 5)); // expected output: 2	El valor absoluto de un número
console.log(Math.ceil(.95)); // expected output: 1	Devuelve el número entero superior.
console.log(Math.round(0.9)); // expected output: 1	Redondea al número más próximo
console.log(Math.exp(0)); // expected output: 1	Devuelve el exponencial
floor()	Devuelve el número entero inferior.
log()	Devuelve el logaritmo natural.
Math.max([value1[, value2[, ...]]])	Devuelve el número máximo entre los números pasados como argumento.
min()	Devuelve el número mínimo entre los números pasados como argumento.
Math.pow(base, exponent)	Devuelve el resultado de un número elevado a una potencia pasada como argumento
function getRandomInt(max) { return Math.floor(**Math.random()** * Math.floor(max)); }	Devuelve un número aleatorio entre O y 1. O <= x < 1
sqrt ()	Devuelve la raíz cuadrada.

Objeto Number	
b = new Number(valor);	
-Si no pasamos algún valor al constructor, la variable se inicializará con el valor 0.	
Propiedades	
MAX_VALUE Devuelve el mayor número posible en JavaScript. MIN_VALUE Devuelve el menor número posible en JavaScript. NaN Representa el valor especial Not a Number NEGATIVE_INFINITY Representa el infinito negativo POSITIVE INFINITY Representa el infinito positivo.	
Métodos	
n = toExponential(num)	Notación científica (1.3)

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

Objeto Number (cont)

n = num.toFixed(2)	destinar una cantidad fija de dígitos para la parte entera y otra para la parte fraccionaria, el número representa la fraccionaria
n = num.toPrecision(3)	marca la longitud fija del número (12.354 => 12.3)

String

Propiedades	
length	la longitud de la cadena
Métodos	
charAt(num)	Permite acceder a un carácter en concreto de una cadena
paragraph.indexOf(searchTerm);	Devuelve la posición de la primera ocurrencia del carácter pasado como parámetro.
lastIndexOf(string)	Devuelve la posición de la última ocurrencia del carácter (string) pasado como parámetro
match(cadena_o_e xpr)	Busca una coincidencia en una cadena y devuelve todas las coincidencias encontradas
replace(cadena_o_e xpr)	Busca una coincidencia en una cadena y si existe, la remplaza por otra cadena pasada como parámetro.
search(cadena_o_e xpr)	Busca una coincidencia en una cadena y devuelve la posición de la coincidencia.
str.slice(beginIndex[, endIndex])	extrae una sección de una cadena y devuelve una cadena nueva.`resultado = result ado.sl ice(0, -1);`
cadena.split([separador][,limite])	Convierte un objeto String en un array de cadenas mediante un separador
str.substr(inicio, longitud)	Devuelve una subcadena en base a un índice y longitud pasados como parámetros.
str.substring(inicio, fin)	returns the part of the string between the start and end indexes, or to the end of the string.
str.toLowerCase( )	Devuelve la cadena en minúsculas. No la cambia.
toUpperCase()	

Objeto Window

```
La interfaz Document representa cualquier página web cargada en el navegador y sirve como punto de entrada al
 Se crea un objeto de este tipo para cada ventana que pueda ser lanzada desde una página Web. Se considera u
```


Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

Objeto Window (cont)	
propiedades	
closed	Indica si una ventana está cerrada o no. Booleano
defaultStatus	Cadena que contiene el texto por defecto que aparece en la barra de estado (status bar) del navegador.
document	Documento actual de la ventana.
history	array que representa las URLS visitadas por la ventana
innerHeight	la altura utilizable de la ventana.
innerWidth	al ancho utilizable de la ventana
length	cuántos frames tiene la ventana actual.
location = 'http://www.exampl-e.com'	retorna un objeto Location con información acerca de la ubicación actual del documento.
locationbar	Objeto barra de direcciones de la ventana
menubar	Objeto barra de menús de la ventana.
name	
opener	Hace referencia a la ventana de navegador que abrió la ventana donde estamos trabajando.
outerHeight	Tamaño en pixels del espacio de toda la ventana, en vertical. Esto incluye las barras de desplazamiento, botones, etc.
outerWidth	Tamaño en pixels del espacio de toda la ventana, en horizontal. Esto incluye las barras de desplazamiento.
pageXoffset	Corresponde a la posición horizontal de la ventana
pageYoffset	Corresponde a la posición vertical de la ventana
parent	referencia al parent de al ventana actual
self	Corresponde a la ventana actual.
status	String con el mensaje que tiene la barra de estado.
top	Corresponde a la ventana de nivel superio
Métodos	
alert ()	Cuadro de diálogo
back()	Regresa a la página anterior según el historial.

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

Objeto Window (cont)	
blur()	Elimina el foco del objeto window actual.
close()	Cierra una página
confirm()	Cuadro diálogo con botones aceptar y cancelar.
find()	Busca texto en una página.
focus()	Pide traer la ventana al frente
forward()	Avanza una página.
home()	Mueve la ventana activa.
var ventanaNueva = window.open(); ventanaNueva.document.write	Abre una nueva ventana.
print()	
prompt()	Genera un cuadro de diálogo con un cuadro de texto para que el usuario introduzca valores.
window.resizeTo(aWidth, aHeight)	Modifica el tamaño de una ventana.
setInterval()	Evalúa una expresión cada cierto tiempo.
setTimeOut(aviso() , 5000)	Evalúa una expresión después de un tiempo especificado.
window.scrollBy(x-coord, y-coord);	Mueve el contenido de una ventana en función de una cantidad especificada en píxeles.
window.scrollTo(x-coord, y-coord)	Mueve el contenido de una ventana especificando una nueva posición de la esquina superior izquierda.
stop()	Detiene una página.

Document	
Con este objeto es posible manipular las propiedades y contenido de los elementos principales de las páginas web. Este objeto cuenta con una serie de subobjetos como son puntos de anclaje, imágenes, vínculos o formularios	
Propiedades	
alinkColor	Devuelve o define el color que tendrán los vínculos activos en el cuerpo
anchors	Devuelve una lista de todas las anclas (anchors) del documento.
document.bgColor = "darkblue";	da/define el color de fondo (bgColor) del documento actual.
cookie	cadena con los valores de las cookies del documento actual
domain	Guarda el nombre del servidor que ha servido el documento
fgColor	da/define el color del documento actual o de su texto

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

Document (cont)	
forms	un array con todos los formularios del documento
images	Array con todas las imágenes del documento (etiqueta <images>)
lastModified	una cadena con la fecha de la última modificación del documento
links	array con los enlaces externos (etiqueta <a ref>)
referrer	Cadena con la URL del documento que llamó al actual, en caso de usar un enlace.
URL	Cadena con la dirección del documento
vlinkColor	Propiedad en la que se guarda el color en formato RGB de los enlaces visitados.
Métodos	
window.captureEvents(eventType)	capturar los eventos que ocurran en la página web. Recibe como parámetro el evento que se desea capturar.
document.close()	cierra el flujo de escritura de un documento
getSelection ()	Devuelve un string que contiene el texto que se ha seleccionado
handleEvent()	Activa el manejador del evento especificado.
home()	Carga la página de inicio.
open()	Abre el flujo del documento
releaseEvents()	Libera los eventos que han sido interceptados
routeEvents ()	Intercepta un evento y lo pasa a lo largo de la jerarquía del objeto que lo lanzó.
write()	Escribe datos en el documento.
writeln ()	Escribe datos además de un salto de línea en el documento.
document.title	

History	
current	la cadena que contiene la URL de la entrada actual del historial
length	número de páginas que han sido visitadas.
next	la cadena que contiene la siguiente entrada del historial.
previous	la cadena que contiene la anterior entrada del historial
back( )	Carga la URL del documento anterior del historial.
forward()	Carga la URL del documento siguiente del historial
go()	Carga la URL del documento especificado por el índice que pasamos como parámetro dentro del historial.

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

Objeto Location

Su finalidad principal es, por una parte, modificar el objeto location para cambiar a una nueva URL, y extraer los componentes de dicha URL de forma separada para poder trabajar con ellos de forma individual si es el caso.

Propiedades

hash	la cadena que representa el anclaje de la URL. Es decir, la parte de la URL que va después de la etiqueta #.
host	Cadena que contiene el nombre del servidor y el número del puerto, dentro de la URL.
hostname	el nombre de dominio del servidor (o la dirección IP), dentro de la URL.
href	la URL completa.
pathname	Cadena que contiene el camino al recurso, dentro de la URL.
port	Cadena que contiene el número de puerto del servidor, dentro de la URL.
protocol	Cadena que contiene el protocolo utilizado (incluyendo los dos puntos), dentro de la URL.
search	Cadena que contiene la información pasada en una llamada a un script, dentro de la URL.

Métodos

location.assign("http://www.ejemplo.com"); // Equivalente a location.href = "http://www.ejemplo.com"	Carga un nuevo documento.
// Método reload() location.reload(true); / *Recarga la página. Si el argumento es true, se carga la página desde el servidor. Si es false, se carga desde la cache del navegador* /	Carga de nuevo el documento actual.
location.replace("http://www.ejemplo.com"); // Similar a assign(), salvo que se borra la página actual del array history del navegador	Sustituye la URL del documento actual por otra URL.

Gestión de ventanas

Abrir y cerrar nuevas ventanas	<a href="mipagina.html" target="_blank"> window.open(URL, nombre_ventana, parametros_configuracion , historia)
window.close()	se cierra la ventana.

```
`var ventana = window.op en( " " ,"","w idt h=300, height =20 0");
```


Sponsored by Readable.com
Measure your website readability!
https://readable.com

Gestión de ventanas (cont)

Apariencia	directories=yes|no|1|0 Obsoleto. IE only fullscreen=yes|no|1|0 Modo pantalla completa. Valor por defecto es no. IE solo. height-=pixels Altura de la ventana. Min. valor es 100 left=pixels Posición izquierda de la ventana. Números negativos no permitidos. location=yes|no|1|0 Campo de direcciones si o no. Opera solo. menubar=yes|no|1|0 Barra de menú si o no. resizable=yes|no|1|0 Si se puede modificar tamaño de la ventana. IE solo. scrollbars=yes|no|1|0 Barras de desplazamiento si o no. IE, Firefox & Opera solo status=yes|no|1|0 Barra de estado. titlebar=yes|no|1|0 Barra de título. toolbar=yes|no|1|0 Barra de herramientas. top=pixels Posicion superior de la ventana.
Comuni-cación	De la ventana secundaria a la principal. La propiedad opener del objeto window hace referencia a la ventana principal. opener.document.bgColor = “red” ;

Funciones predefinidas del lenguaje

if (isFinite(1000 / x))	
eval('2 + 2')	Convierte una cadena que pasamos como argumento en código JavaScript ejecutable.
IsNaN ()	
Number ()	Convierte el objeto pasado como argumento en un número que represente el valor de dicho objeto
String()	Convierte el objeto pasado como argumento en una cadena que represente el valor de dicho objeto.
parseInt()	Convierte la cadena que pasamos como argumento en un valor numérico de tipo entero.
parseFloat()	Convierte la cadena que pasamos como argumento en un valor numérico de tipo flotante.
if (typeof this[i] === 'string') { this[i] = this[i].toUpper-Case(); }	Comprobar que variable sea de algún valor

Funciones definidas por el usuario

function nombre([parametros]) { grupo_de_instrucciones; [return valor;] }
Los valores se pueden definir a la hora de crear la función
function numeroAleatorio(min = 1, max = 100){ var valorAleatorio = Math.floor(Math.random() * (max - min) + min); return valorAleatorio; }

- Los argumentos son un array : arguments.length, arguments[1]

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

Funciones definidas por el usuario (cont)

- Los parámetros se pueden pasar a las funciones de dos formas : por valor o por referencia. No se peude modificar el argumento dentro de la función, pero sí los objetos.	
- El array se considera un objeto	
typeof	devuelve el tipo
-Una función puede ser anónima	var cuadrado = function(number) {return number * number};
-Es mejor definir un grupo de funciones en el head para poder asegurar su uso en el body	
-Poner var para hacer variables locales	
-No usar var para usar/modificar valores de las globales	

Arrays

Declaración	var nombre_del_array = new Array(); var nombreArr = [];
	var nombre_array = new Array(10);
	var lista_productos = new Array('Pan', 'Agua', 'Lentejas');
Inicialización	nombre_del_array[i] = “valor”;
	lista_productos[0] = “pan”;
Propiedades	
length	Longitud, se puede modificar
prototype	Sirve para agregar nuevas propiedades y métodos al objeto array. Las instancias Array heredan de Array.prototype.
	Array.prototype.Mayus = function() { ... } a.Mayus();
	Array.prototype.dia = "lunes";
	Array.prototype.Susti = function(vi, vf) { for (var i = 0; i< this.length; i++) { if (this[i] == vi) { this[i] = vf ; } } }
Métodos	
concat()	Une dos o más arrays
indexOf(string)	Busca un elemento en el array y devuelve su posición. **Devuelve -1 si no lo encuentra**
join()	Junta los elementos de un array en una cadena
lastIndexOf(string)	Busca un elemento en un array desde el final y devuelve su posición.
pop()	Borra el último elemento del array y devuelve su contenido

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

Arrays (cont)	
push()	Añade nuevos elementos al array y devuelve su nueva longitud.
reverse()	Invierte el orden de los elementos en el array.
a = frutas.shift()	Elimina el primer elemento del array y devuelve el elemento
var citrus = fruits.slice(1, 3); fruits.slice(-1);	Selecciona parte de un array y devuelve su contenido en un array nuevo. Útil para mostrar elemtnos del array sin accederlos directam
arr.sort()	Ordena los elementos del array localmente y devuelve el array.La ordenación no es necesariamente estable. El por defecto es de Uni

	arr.sort([compareFunction]) Especifica una función que define el modo de ordenamiento. los elementos del array son ordenados de ac
	`if (a es menor que b según criterio de ordena miento) { return -1; } if (a es mayor que b s`
	`var arr = [ 40, 1, 5, 200 ]; function comparar ( a, b ){ return a - b; } arr.sort( comparar ); //`
	`arr.so rt( fun cti on( a,b ){r eturn a - b;});`

splice()	cambia el contenido de un array eliminando elementos existentes y/o agregando nuevos elementos.
	`months.sp lice(4, 1, 'May'); // replaces 1 element at 4th index`
	`months.sp lice(1, 0, 'Feb'); // inserts at 1st index position`

arr.toString();	Convierte un array en string y devuelve el resultado (entre comas)
arr.unshift("-Uno","Dos");	Añade elementos al inicio del array y devuelve la nueva longitud
split(","/g)	Convierte un string en un array
includes(variable) || !csPais.includes(cPais)	comprueba si existe el argumento en el array, devuelve true o false

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

Recuerda

Los prompt guardan cadenas	
arguments es un array	
if(num == undefined || num == null || num == 0 || isNaN(num)){	
En math.random() el rango es de 0 a 1. *(max - min) + min	
Para definir una variable por defecto se puede hacer en argumentos con =	
Hasta que el usuario pulse Cancelar while(cadena !== null){}	
indexOf() actúa como booleano	
sort ordena en ASCII por defecto, hay que utilizar a,b para ordenar de otra manera	
las fechas empeizan por 0	
var objeto_mio = new Object(); objeto_mio.propiedadtal; objeto_mio.metodocual([parámetros o no]);	
En notaciones array, la convención es que '-1' devuelva el último elemento. En el caso de JS: `array[:-1]` o `array.s li ce(-1)`	
Para ver el tipo de una variable	console.log(typeof variable);

Arrays multidimensionales

En js los arrays guardan información dinámicamente (se le pueden añadir elementos)	var dias = new Array(7);
	dias[i] = new Array(4); // Cada día contiene un array de 4 elementos
	dias[i][j] = Math.floor(Math.random() * 40);
	for(i = 0; i< dias.length; i++) { for(j = 0; j < dias[i].length ; j++) { document.write("Temperatura dia "+i + " hora "+j + " = "+dias[i][j] + "
	"); }

Pueden almacenar tipos diferentes	
Guardar array dentro de otro array	arr1.push(arr2);
--	
Inicializar array	var cualificaciones = [];
con declaración	var items = [[1, 2], [3, 4], [5, 6]]; console.log(items[0][0]); // 1 console.log(items);
	var arr = [1, 2, [3, 4], 5]; alert (arr[2][1]); //alerts "4"
Introducir valores	items.push(['a','b','c']) //[[1, 2], [3, 4], [5, 6], ['a','b','c']]
Recorrer valores	for(var i = 0; i < cubes.length; i++) { var cube = cubes[i]; for(var j = 0; j < cube.length; j++) { display("cube[" + i + "][" + j + "] = " + cube[j]); } }

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

Arrays multidimensionales (cont)

```
for( var i = 0; i < items.length; i++){ alert(items[i][0]); }
```

Eventos

<element onload = function> || element.onmouseout = function; || window.addEventListener('load', () => { || window.onload = () => { //window loaded }

Al dejar los parámetros vacíos, el objeto que llama la función es el parámetro gracias al keyword this	function a(){ this.style.backgroundColor='aq-ua';}
Para que no cargue todas las funciones hasta que termine de cargarse la página	window.onload = function(){

el evento tiene varios atributos y métodos como target (el originador del evento), type (el tipo de evento) y stopPropagation() para detener la propagación del evento.

Otras propiedades dependen del tipo de evento, ya que hay eventos de ratón, teclado, de drag, de fetch...

https://flaviocopes.com/javascript-events/

Expresiones regulares

^	Empieza por
$	Acaba por
[abcd]	Deben preceder unos a otros a, b, c, d, abcd
[a-z]	De un caracetr a otro
[0-9]{4}	aparece exactamente 4 veces
[^abcd]	cualquier ocurrencia excepto abcd
[a-c0-2]*	zero or more consecutive occurrences of a, b, c, 0, 1, 2.
(|||)	Cualquiera de estas combinaciones
\(ab\)	La expresión hará match en grupo

Regex caracteres

Metacharacter Sequence	Meaning	Example Expression	Example Match
.	Matches any single character except the newline character.	**ab.def**	**abcdef**, **ab9def**, **ab=def**
\s	Matches a whitespace character (such as a space, a tab, a form feed, etc.)	**abcd\se**	**abcd e**, **abcd**(tab)**e**
\S	NOT whitespace	**\S\S\s\S**	**AB D**, **99**(tab)**9**
\w	A word character. A word character is a letter, a number, or an underscore. This set of characters may also be represented by the regex character set **[a-zA-Z0-9_]**	**\w\{1,\}-\w\{1,\}** (see quantifiers, below)	**well-wishes**, **far-fetched**
\W	NOT a word character	**\w\W\{1,\}\w**	**a,!-{?8;b**, **9-5**

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

Regex \

Metacharacter Sequence	Matches
\n	a newline
\t	a tab
\r	a carriage return
\v	a vertical tab
\f	a form feed

Regex 3

^	Start of string or line	^abc	**abc** (appearing at start of string or line)
$	End of string, or end of line	xyz$	**xyz** (appearing at end of string or line)
\b	Word boundary	ing\b	match**ing** (matches **ing** if it is at the end of a word)
\B	NOT word-boundary	\Bing\B	st**ing**er (matches **ing** if it is not at the beginning or end of the word)
\<	Start of word	\<when	**when**ever (matches **when** only if it is at the beginning of a word)
\>	End of word	ever\>	when**ever** (matches **ever** only if it is at the end of a word)

eventos 3

evitar acción por defecto	<A href="pagina.html" onClick="return preguntar()">Pulsa aqui function preguntar(){ return confirm("¿Desea realmente ir a esa dirección?"); }
Si se añaden los paréntesis después del nombre de la función, en realidad se está ejecutando la función y guardando el valor devuelto por la función en la propiedad onclick de elemento.	// Asignar una función externa a un evento de un elemento document.getElementById("pinchable").onclick = muestraMensaje; // Ejecutar una función y guardar su resultado en una propiedad de un elemento document.getElementById("pinchable").onclick = muestraMensaje();
al cargar la página...	window.onload = function () {}
obtener elementos por tagname	let casillas = document.getElementsByTagName('td'); casillas[i].addEventListener("mouseover", pintar);
cada vez que se llama a función, resetea	function seleccionarColor(e){ resetSeleccion(); pincel = this.getAttribute("class"); this.style.borderWidth = "thick"; this.style.borderColor = "white"; var selec = document.getElementById("-seleccionado"); selec.style.backgroundColor = pincel; } function resetSeleccion(){ for(let k = 0; k < paleta.length; k++){ paleta[k].style.borderWidth = "thin"; paleta[k].style.borderColor = "black"; } }
	function pintar(evento){ //this es el elemento que lo llamó //console.log(this, evento); this.style.backgroundColor = pincel; }
eventos DOM	boton1.onclick=valida';
	var tipo = evento.type;

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

eventos 3 (cont)	
	function resalta(elEvento) { var evento = elEvento || window.event; switch(evento.type) { case 'mouseover': this.style.borderColor = 'black'; break; case 'mouseout': this.style.borderColor = 'silver'; break; } }
	window.onload = function() { document.getElementById("seccion").onmouseover = resalta; document.getElementById("seccion").onmouseout = resalta; }
Teclado	

	window.addEventListener("keypress", desactivar); var pulsado = true;
	function desactivar(){ if(pulsado == true){ pulsado = false; } else if(pulsado == false){ pulsado = true; } }
	function pintar(){ if(pulsado !== false){ console.log("pintando"); this.style.backgroundColor = pincel; } }
	for(let i = 0; i < casillas.length; i++){ casillas[i].addEventListener("mouseover", pintar); }

keydown	keyCode
keypress	keyCode, charCode
keyup	keyCode
	window.onload = function(){ document.onkeypress = enviar; } function enviar(evento){ var tecla = evento.keyCode; if(tecla == 13){ return false; }else{ return true; } }
Ratón	var coordenadaX = evento.clientX; var coordenadaY = evento.clientY;
Método addEventListener().	Este método tiene tres argumentos: el tipo de evento, la función a ejecutar y un valor booleano (true o false), que se utiliza para indicar cuándo se debe capturar el evento: en la fase de captura (true) o de burbujeo (false). elemento.addEventListener('evento', función, false|true)

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

eventos 3 (cont)	
Formularios	function enviar(form){ if (formulario.alta.checked == true){ formulario.action = "paginas/alta.html"; }
check	var edad = document.getElementById("edad").value;
	function validaCheck() { var elemento = document.getElementById("campoCondiciones") ; if (!elemento.checked) { return false; } return true; }
regex	var patronE = new RegExp("^(1[8-9]|[2-9][0-9]|100)$"); if(!patronE.test(edad)){ alert("-Debes ser mayor de 18 y menor de 100"); }
select	valoresGenero=document.getElementsByName("genero"); for(var i=0; i<valoresGenero.length; i++) { if(valoresGenero[i].checked){ seleccionado=true; }
validación	<form action="URL" method="post" name="formValidado" onsubmit="return validar()"> </form>
	function validación() { valor = document.getElementById("campo").value; if(valor == null || valor.length == O){ alert("El campo no puede esta vacío"); return false; } return true; }
onblur	
onchange	
onclick	
ondblclick	
onfocus	
onkeydown	
onkeypress	
onkeyup	
onload	
onmousedown	
onmousemove	
onmouseout	
onmouseover	
onmouseup	
Los objetos de nodo al ponerse en un bucle for modifican el length del mismo objeto, por eso conviene usar while	var lista = cajaB.getElementsByTagName("option"); while(lista.length > 0){

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

regex 4

var patronE = new RegExp("^(1[8-9]|[2-9][0-9]|100)$");	i: no distingue entre mayúsculas y minúsculas. g: búsqueda global en toda la cadena; no para en la primera coincidencia. m : multiline
	cadena.match cadena.match(er)
	cadena.search cadena.search(er)
	cadena.replace(er, texto)
	if(!patronE.test(edad)){
( ) Agrupar caracteres	
[ ] Conjunto de caracteres opcionales	
\O Carácter nulo	
\t Caracter ASCII 9 (tabulador)	

Cookies

crear	document.cookie = 'nombreCookie=valorCookie; expires=fechaDeExpiración; path=rutaParaLaCookie';
	max-age=" + 60*60*24*30
	document.cookie = "name=oeschger";
	var todas = document.cookie.split(';'); var c = todas[i]; while (c.charAt(0)==' ') c = c.substring(1); if (c.indexOf(name) == 0) return c.substring(name.length,c.length)

Eventos 2

Añadirle un listener al target (en este caso 'casillas') que llama a la función cuando se produce el evento	casillas[i].addEventListener("mouseover", pintar);
la función cambia la propiedad del elemento que lo llamó, en este caso el target	function pintar(){ this.style.backgroundColor = pincel; }
	<form name="registro" method="get" onsubmit="-comprobar()">
llamar función con DOM	boton.onclick=validar;

Métodos y atributos DOM

window.onload = function () { console.log("cargada"); }	
let lienzo = document.getElementById('lienzo');	
let casillas = document.getElementsByTagName('td');	
var attribute = element.getAttribute(attributeName);	this.getAttribute("class");
conseguir valor de campo	var valor = document.getElementById("usuario").value;
Dar valor a elemento	document.getElementById("demo").innerHTML = 5 + 6;

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

Métodos y atributos DOM (cont)

Insertar valor conjunto a elemento	var d1 = document.getElementById('one'); d1.insertAdjacentHTML('afterend', ' two ');
index seleccionado de elemento	var cliente = lista.options[lista.selectedIndex];
	var idCliente = lista.options[lista.selectedIndex].value;
coger propiedad de elemento que llama	function cargarContenido(e) { let id = e.target.value;

Propiedades DOM

document.body.	
childNodes	lista de nodos hijo del nodo padre, devuelve array
nodeName	cadena, define el nombre del nodo, devuelve una cadena
nodeType	número, muestra el tipo, devuelve un número
nodeValue	devuelve el valor del nodo, devuelve una cadena
firstChild	El primer hijo del nodo. nodo
lastChild	El último hijo del nodo, nodo
previousSibling	El hermano anterior al nodo, nodo o null
nextSibling	
hasChildNodes	tiene hijos o no, booleano
Attributes	matriz con los atributos

Iteración

i += 3	cada 3 veces...

HTML

formulario select	<form action="/action_page.php"> <select name="cars"> <option value="volvo">Volvo XC90</option> <option value="saab">Saab 95</option> <option value="mercedes">Mercedes SLK</option> <option value="audi">Audi TT</option> </select> <input type="s-ubmit" value="Submit"> </form>
formulario checkbox	<input type="checkbox" name="vehicle1" value="Bike"> I have a bike <input type="checkbox" name="vehicle2" value="Car"> I have a car <input type="checkbox" name="vehicle3" value="Boat" checked> I have a boat
espacio	

Modificar DOM

crear elemento	var parrafo1 = document.createElement("p");
crear nodo de texto	var texto1 = document.createTextNode("Memento mori")
adjuntar hijo	parrafo1.appendChild(texto1);
	caja.appendChild(parrafo1);
	al adjuntar uno existente en otro lugar, desaparece del primero
	no se puede adjuntar en el documento directamente: document.body.appendChild(nodoCheck);

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

Modificar DOM (cont)	
crear atributo	entrada.setAttribute("type", "number");
eventlistener en elemento adjuntado	
Coger valores de atributos de nodo	alert(enlace.href); muestra http://www.com
getElements devuelve un array de objetos al que podemos acceder	var input = document.getElementsByTagName("input");
	var valor1 = input[0].value;
borrar el mismo nodo a que referenciamos	lista[0].parentNode.removeChild(lista[0]);
Coger el valor interno del atributo valor (value="")	var valor1 = input[0].value;

AJAX	

Inicializar ajax	if(window.XMLHttpRequest) { {
	peticion_http = new XMLHttpRequest();}
	else if(window.ActiveXObject) {
	peticion_http = new ActiveXObject("Microsoft.XMLHTTP");

	if(window.XMLHttpRequest) {
	return new XMLHttpRequest();
	else if(window.ActiveXObject) {
	return new ActiveXObject("Microsoft.XMLHTTP");
Abrir petición AJAX GET en dirección	ajax.open("GET", "U73.php?idCliente="+id);

Comprobar que AJAX está cargado	ajax.onreadystatechange = function () {
	if(ajax.readyState == 4 && ajax.status == 200){
	var datos = ajax.responseText;
	}
	}ajax.send(null);

Abrir petición AJAX POST	if(window.XMLHttpRequest) {
	return new XMLHttpRequest();
	else if(window.ActiveXObject) {

AJAX y JSON	
	if($idcliente == "c3"){ $resultado = ["nombre"=>"Sara", "direccion"=>"Pontevedra", "telefono"=>111222333, "email"=>"Sara@mail.com"];
Respuesta JSON	$resul = json_encode($resultado);

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

JQuery	
Enlazar	<script src="jquery-3.4.1.min.js"></script>
window onload	$(document).ready(function() { // });
añadir comportamiento a nodo	$("a").click(function(event){
prevenir comportamiento por defecto	event.preventDefault();
añadir clase a nodo	$("a").addClass("test");
quitar clase a nodo	$("a").removeClass("test");
referirse a nodo que llama a evento	$(this).hide("slow");
callback	función que se pasa como argumento a otra función y se ejecuta después de que su función padre se haya completado
callback sin argumentos	$.get("myhtmlpage.html", myCallBack);
	When $.get() finishes getting the page myhtmlpage.html, it executes the myCall-Back() function.
callback con argumentos	$.get("myhtmlpage.html", function() { myCallBack(param1, param2); });
añadir texto	$("body").append("Se fue")
guardar elemento jquery	var firstHeadingElem = $("h1").get(0);
	var firstHeadingElem = $("h1")[0];
Although logo1 and logo2 are created in the same way (and wrap the same DOM element), they are not the same object.	var logo1 = $("#logo"); var logo2 = $("#logo");
	alert($("#logo") === $("#logo")); // alerts "false"
	var logo1 = $("#logo"); var logo1Elem = logo1.get(0); var logo2 = $("#logo"); var logo2Elem = logo2.get(0); alert(logo1Elem === logo2Elem); // alerts "true"
A los elementos p`con clase test les añade un evento click	$("p.test").on("click", function(){ });
Varios eventos	$("p").on({ "click": function() { console.log("clicked!"); }, "mouseover": function() { console.log("hovered!"); } });
Sçolo ocurre una vez	$("p").one("click", firstClick);

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

JQuery (cont)

function firstClick() { console.log("You just clicked this for the first time!"); // Now set up the new handler for subsequent clicks; // omit this step if no further click responses are needed $(this).click(function() { console.log("You have clicked this before!"); }); }

Every event handling function receives an event object, which contains many properties and methods. target

The DOM element that initiated the event

By **Sara** (lasago)
cheatography.com/lasago/

Not published yet.
Last updated 21st May, 2019.

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com