# Principales métodos/propiedades de cadena/string

* Length: (propiedad)

Calcula la longitud de una cadena de texto (el número de caracteres que la forman)

* concat():

Concatena a la cadena, la cadena pasada como parámetro

* toUpperCase():

Transforma todos los caracteres de la cadena a sus correspondientes caracteres en mayúsculas:

* toLowerCase()

Transforma todos los caracteres de la cadena a sus correspondientes caracteres en minúsculas:

* charAt(posicion)

Obtiene el carácter que se encuentra en la posición indicada:

* indexOf(caracter)

Calcula la posición en la que se encuentra el carácter indicado dentro de la cadena de texto. Si el carácter se incluye varias veces dentro de la cadena de texto, se devuelve su primera posición empezando a buscar desde la izquierda. Si la cadena no contiene el carácter, la función devuelve el valor -1:

* lastIndexOf(caracter):

Calcula la última posición en la que se encuentra el carácter indicado dentro de la cadena de texto. Si la cadena no contiene el carácter, la función devuelve el valor -1. La función lastIndexOf() comienza su búsqueda desde el final de la cadena hacia el principio, aunque la posición devuelta es la correcta empezando a contar desde el principio de la palabra.

* substring(inicio, final):

Extrae una porción de una cadena de texto. El segundo parámetro es opcional. Si sólo se indica el parámetro inicio, la función devuelve la parte de la cadena original correspondiente desde esa posición hasta el final. Si se indica un inicio negativo, se devuelve la misma cadena original:

* substr(inicio, cuantos)

Extrae una subcadena desde la posición especificada, tantos caracteres como se indique en el segundo parámetro

* split(separador)

Convierte una cadena de texto en un array de cadenas de texto. La función parte la cadena de texto determinando sus trozos a partir del carácter separador indicado

* endsWith(cadena)

Chequea (true-false) si el string finaliza con la cadena especificada

* startsWith(cadena)

Chequea (true-false) si el string empieza con la cadena especificada

* includes(cadena)

Chequea (true-false) si el string contiene la cadena especificada

* match()

Busca el string o expresión regular y retorna la/s coincidencia/s

* repeat()

Devuelve un nuevo string con un número especificado de copias de un string existente

* replace()

Sustituye la expresión regular especificada en el string

* search()

Retorna la posición del valor, o expresión regular especificada dentro del string

* slice()

Extrae parte de un string y devuelve uno nuevo

* toString()

Convierte el valor de un objeto a string

* valueOf()

Inverso a toString, convierte el valor primitivo a objeto

* trim()

Elimina los espacios en blanco, de ambos lados del string

# Principales métodos/propiedades numéricas

## Propiedades

* NaN

Representa un valor "Not-a-Number"

* MAX\_VALUE

Retorna el máximo valor posible en JavaScript

* MIN\_VALUE

Retorna el valor mas pequeño posible en JavaScript

* NEGATIVE\_INFINITY

Representa el valor infinito negativo

* POSITIVE\_INFINITY

Representa el valor infinito

* prototype

Permite añadir propiedades y métodos a un objeto

# Métodos

* isNaN()

Permite proteger a la aplicación de posibles valores numéricos no definidos

* Infinity

Hace referencia a un valor numérico infinito y positivo (también existe el valor -Infinity para los infinitos negativos)

* toFixed(dígitos)

Devuelve el número original con tantos decimales como los indicados por el parámetro dígitos y realiza los redondeos necesarios. Se trata de una función muy útil por ejemplo para mostrar precios

* isFinite()

Chequea si el valor es finito

* isInteger()

Chequea que el valor se a entero

* toExponential(x)

Convierte el numero a notación exponencial

* toPrecision(x)

Formatea el numero a x decimales

* toString()

Convierte a cadena

* valueOf()

Convierte el valor primitivo del numero

# Principales métodos/propiedades matemáticas (math.)

## Propiedades

* E Numero E (aprox. 2.718)
* LN2 Logaritmo natural de 2 (aprox. 0.693)
* LN10 Logaritmo natural de 10 (aprox. 2.302)
* LOG2E Logaritmo de E en base 2 (aprox. 1.442)
* LOG10E Logaritmo de E en base 10 (aprox. 0.434)
* PI PI (aprox. 3.14)
* SQRT1\_2 Raíz cuadrada de 1/2 (aprox. 0.707)
* SQRT2 Raíz cuadrada de 2 (aprox. 1.414)

# Métodos

* abs(x) Retorna el valor absoluto de x
* acos(x) Retorna el arcoseno de x en radianes
* acosh(x) Retorna el arcoseno hyperbólico de x
* asin(x) Retorna el arcoseno de x
* asinh(x) Retorna el arcoseno hyperbólico de x
* …más trigonométricas…
* round(x) Redondea x al número entero más próximo
* ceil(x) Retorna x, redondeado hacia arriba
* floor(x) Retorna x, redondeado hacia abajo
* log(x) Retorna el logaritmo natural de x (en base E)
* max(x, y, z, ..., n) Retorna el número más alto de la lista
* min(x, y, z, ..., n) Retorna el número mas bajo de la lista
* pow(x, y) Retorna el valor de x a la potencia de y
* sqrt(x) Retorna raíz cuadrada de x
* cbrt(x) Retorna raíz cúbica de x
* trunc(x) Retorna la parte entera de x
* random() Retorna un número aleatorio entre 0 y 1

# Principales métodos/propiedades arrays

## Propiedades

* constructor Retorna la función que creo le objeto Array
* length Pone o retorna el número de elementos del array
* prototype permite añadir propiedades/métodos al objeto array

# Métodos

* concat(lista arrays)

Une 2 o más arrays en una sola, retorna el array resultante

* copyWithin(índice, inicio, fin])

Copia elementos de un array, empezando en índice, los que se encuentran desde inicio a fin, por defecto inicio=0 y fin=length

* entries()

Crea un nuevo objeto de key/valor, en el que los keys son los índices y el valor, cada uno de los valores del array

* keys()

Igual que entries, pero solo con los keys

* every(function(valorActual, indiceActual, array))

Verifica si cada uno de los elementos del array cumple con la condición de la función dada. Si es así, devuelve true si no false

* filter(function(valorActual, indiceActual, array))

Crea un nuevo array con los valores que cumplan las condiciones de la función

* map(function(valorActual, indiceActual, array))

Crea un nuevo array con los valores retornados por la función, por cada elemento del array

* fill(valor, índice, final)

Llena un array con determinado valor desde el índice a final, Si no se especifica índice y final, es todo el array

* find(function(valorActual, indiceActual, array))

Retorna el valor del primer elemento que cumple la condición de la función

* findIndex(function(valorActual, indiceActual, array))

Igual que find, pero retorna el índice

* some(function(valorActual, indiceActual, array))

Igual al find en valores numéricos, con >, >=, <, <=, ¡= …

* forEach(function(valorActual, indiceActual, array))

Llama a la función, por cada elemento del array

* includes(cadenaString,inicioIndice)

Chequea (true-false) si un array contiene un elemento especificado

* indexOf(elemento,inicioIndice)

Busca un elemento y retorna la posición

* lastIndexOf(elemento,inicioIndice)

Busca un elemento y retorna la posición, desde el final del array

* Array.isArray(array)

Chequea (true-false) si el objeto es un array

* join(separador)

Pone los elementos de un array en un string usando el separador especificado

* reduce(function(total, valorActual, indiceActual, array))

ejecuta una función **reductora** sobre cada elemento de un array, devolviendo como resultado un único valor

* reduceRight()

Similara a reduce(va de derch a izda)

* reverse()

Invierte el orden de los elementos del array

* pop()

Elimina el último elemento del array y lo retorna

* push(lista elementos)

Añade la lista de elementos al final del array, retornando la nueva longitud

* shift()

Elimina el primer elemento del array y lo retorna

* unshift(lista elementos)

Añade nuevo elemento al principio del array, retornando la nueva longitud

* slice(inicio,fin)

Selecciona una parte del array y lo retorna como nuevo array

* sort()

Ordena los elementos del array .sort((a,b)=>{(a-b)});

* splice(IndiceInicio,cuantosBorra,listaAAñadir)

Añade/Elimina elementos desde un array

cuantosBorra y ListaAAñadir, son opcionales

Si el índice es negativo, se empieza del final del array

* toString()

Convierte un array a string, y retorna el resultado

* valueOf()

Retorna el valor primitivo de un array

# Principales métodos globales

* eval(cadena)

Evalua el string y lo ejecuta como si fuera código javascript

* isFinite(valor)

Determinar si es un valor numérico legal

* isNaN(valor)

Determinar si son solo dígitos

* Number(objeto)

Convierte el valor del objeto en un numero

* parseFloat(string)

Convierte el string, en coma flotante.Si no, NaN

* parseInt(string)

Convierte el string en entero. Si no, NaN

* String() Convierte el valor del objeto en cadena

# Principales métodos/propiedades DOM

* [activeElement](https://www.w3schools.com/jsref/prop_document_activeelement.asp)

retorna el actual elemento con el foco

var le = linea.activeElement.TagName

* [documentElement](https://www.w3schools.com/jsref/prop_document_documentelement.asp)

Retorna sobre el documento en general del html

* [addEventListener(evento,funcion)](https://www.w3schools.com/jsref/met_document_addeventlistener.asp)

Añadir eventos al element

* [removeEventListener(evento,funcion)](https://www.w3schools.com/jsref/met_document_removeeventlistener.asp)

elimina el evento añadido por [addEventListener()](https://www.w3schools.com/jsref/met_document_addeventlistener.asp). Los parámetros son los mismo que en addEventListener

* [referrer](https://www.w3schools.com/jsref/prop_doc_referrer.asp)

Retorna la URL del documento que a cargado el actual

* [head](https://www.w3schools.com/jsref/prop_doc_head.asp)

Retorna el <head> del documento

* [title](https://www.w3schools.com/jsref/prop_doc_title.asp)

Pone o retorna el <title> del documento

* [URL](https://www.w3schools.com/jsref/prop_doc_url.asp)

Retorna la URL del documento

* [body](https://www.w3schools.com/jsref/prop_doc_body.asp)

Pone o retorna el cuerpo del documento(<body>)

* [open()](https://www.w3schools.com/jsref/met_doc_open.asp)

Abre el documento para la salida con con document.write()

* write()

Escribe código html en el documento

* writeln()

Igual que write, pero añade un salto de línea

* [close()](https://www.w3schools.com/jsref/met_doc_close.asp)

Cierra el documento abierto con document.open()

* [createAttribute(string)](https://www.w3schools.com/jsref/met_document_createattribute.asp)

Crea un atributo nuevo en una variable

* setAttributeNode(atributo)

Asignamos el nuevo atributo a la línea

* setAttribute(string,valor)

Simplifica a las 2 anteriores

* getAttribute(string)

Retorna el valor del atributo que hemos creado nosotros

* [createElement()](https://www.w3schools.com/jsref/met_document_createelement.asp)

Crea un Elemento (tag) nuevo

VER LOS METODOS Y PROPIEDADES DEL LOS ELEMENTOS

* [forms](https://www.w3schools.com/jsref/coll_doc_forms.asp)[].

Array con la lista de formularios del documento

* [getElementById()](https://www.w3schools.com/jsref/met_document_getelementbyid.asp)

Retorna el elemento por si Id

* [getElementsByClassName()](https://www.w3schools.com/jsref/met_document_getelementsbyclassname.asp)

Retorna array de elementos seleccionados por si class

* [getElementsByName()](https://www.w3schools.com/jsref/met_doc_getelementsbyname.asp)

Retorna array de elementos seleccionados por si name

* [getElementsByTagName()](https://www.w3schools.com/jsref/met_document_getelementsbytagname.asp)

Retorna array de elementos seleccionados por el tag

* [querySelector()](https://www.w3schools.com/jsref/met_document_queryselector.asp)

Retorna el elemento seleccionado por el selector css

* [querySelectorAll()](https://www.w3schools.com/jsref/met_document_queryselectorall.asp)

Retorna array de elementos seleccionados por el selector de ccs

# Principales métodos/propiedades ELEMENT

* appendChild()

Añade el nuevo hijo al nodo como el último elemento

* childElementCount

Retorna el numero de hijos

* childNodes

Retorna el array de hijos de elementos

* children

Retorna un array de elementos child (sin nodos de texto ni comentarios)

* clientHeight

Retorna la altura del elemento incluyendo el padding

* [clientLeft](https://www.w3schools.com/jsref/prop_element_clientleft.asp)

Retorna la anchura del borde izquierdo

* [clientTop](https://www.w3schools.com/jsref/prop_element_clienttop.asp)

Retorna la altura (top) del un elemento

* [clientWidth](https://www.w3schools.com/jsref/prop_element_clientwidth.asp)

Retorna la anclura de un elemento, incluyendo el padding

* [cloneNode()](https://www.w3schools.com/jsref/met_node_clonenode.asp)

Clona un elemento

* [contentEditable](https://www.w3schools.com/jsref/prop_html_contenteditable.asp)

Pone o retorna si el contenido de un elemento es editable

* [firstChild](https://www.w3schools.com/jsref/prop_node_firstchild.asp)

Retorna el primer hijo del elemento

* [firstElementChild](https://www.w3schools.com/jsref/prop_element_firstelementchild.asp)

Retorna el primer hijo elemento de un elemento

* [getAttribute(stringAtributo)](https://www.w3schools.com/jsref/met_element_getattribute.asp)

Retorna el valor del atributo especificado

* [getAttributeNode(stringAtributo)](https://www.w3schools.com/jsref/met_element_getattributenode.asp)

Retorna los atributos especificados del nodo

* [getAttributeNames()](https://www.w3schools.com/jsref/met_element_getattributenode.asp)

Retorna array con los atributos del nodo

* [hasAttribute(stringAtributo)](https://www.w3schools.com/jsref/met_element_hasattribute.asp)

Retorna true si dispone del atributo especificado

* [hasAttributes()](https://www.w3schools.com/jsref/met_node_hasattributes.asp)

Retorna true si un elemento tiene atributos

* [hasChildNodes()](https://www.w3schools.com/jsref/met_node_haschildnodes.asp)

Retorna true si un elemento tiene hijos

* [innerHTML](https://www.w3schools.com/jsref/prop_html_innerhtml.asp)

Pone o retorna el valor del contenido de un elemento

* [innerText](https://www.w3schools.com/jsref/prop_node_innertext.asp)

Pone o retorna el texto del contenido de un nodo y sus descendientes

* [insertAdjacentElement()](https://www.w3schools.com/jsref/met_node_insertadjacentelement.asp)

Inserta un elemento HTML en la posición relativa del elemento actual

* [insertAdjacentHTML()](https://www.w3schools.com/jsref/met_node_insertadjacenthtml.asp)

Inserta un texto formateado en HTML en la especificada posicion relativa del elemento actual

* [insertAdjacentText()](https://www.w3schools.com/jsref/met_node_insertadjacenttext.asp)

Inserta un texto en la posición relativa del elemento actual

* [insertBefore()](https://www.w3schools.com/jsref/met_node_insertbefore.asp)

Inserta un nuevo nodo hijo antes del especificado

* [lastChild](https://www.w3schools.com/jsref/prop_node_lastchild.asp)

Retorna el ultimo nodo hijo

* [lastElementChild](https://www.w3schools.com/jsref/prop_element_lastelementchild.asp)

Retorna el ultimo nodo hijo de un elemento

* [nodeName](https://www.w3schools.com/jsref/prop_node_nodename.asp)

Retorna el name de un nodo

* [nodeType](https://www.w3schools.com/jsref/prop_node_nodetype.asp)

Retorna el tipo de nodo

* [nodeValue](https://www.w3schools.com/jsref/prop_node_nodevalue.asp)

Pone o retorna el valor de un nodo

* [parentNode](https://www.w3schools.com/jsref/prop_node_parentnode.asp)

Retorna el nodo padre de un elemento

* [parentElement](https://www.w3schools.com/jsref/prop_node_parentelement.asp)

Retorna el nodo padre de un elemento

* [removeChild()](https://www.w3schools.com/jsref/met_node_removechild.asp)

Elimina el nodo hijo de un elemento

* [replaceChild()](https://www.w3schools.com/jsref/met_node_replacechild.asp)

Sustituye el nodo hijo de un elemento