ESCUELA PÚBLICA:

DE TODOS

PARA TOBOS



```
<script type="tipo-lenguaje" defer >
                                                                    /=
        código-script
</script>
                                                                    %=
<script src="fichero.js"></script>
                                                                     **=
<etiqueta-htmlevento="instrucción-javascript">
                                                                     ++
<ahref="javascript:instrucción" >enlace</a>
<ahref="javascript:nombre-función" >enlace</a>
const nombre-constante-1=valor-1 [, nombre-constante-
2=valor-2] ...
var nombre-variable;
let nombre-variable;
                                                                    %
var nombre-variable=valor;
let nombre-variable=valor;
var nombre-variable-1, nombre-variable-2, ...;
                                                                    &&
let nombre-variable-1, nombre-variable-2, ...;
                                                                    Ш
var nombre-variable-1 [=valor-1],[nombre-variable-2
[=valor-2]]...;
                                                                     <<
let nombre-variable-1 [=valor-1],[nombre-variable-2
                                                                     >>
[=valor-2]]...;
                                                                    >>>
Number
♦ MAX_VALUE
                                                                    1=
♦ MIN VALUE
♦ NaN
                                                                     == =
♦ NEGATIVE INFINITY
                                                                     !==
♦ POSITIVE INTINITY
                                                                     <=
♦ EPSILON
♦ MAX SAFE INTEGER
♦ MIN_SAFE_INTEGER
                                                                    Condición?valor1:valor2
♦ isNaN(número)
                                                                     typeof expresión
♦ isFinite(número)
                                                                    //comentario
♦ isInteger(número)
♦ isSafeInteger(número)
                                                                    comentario
◆ parseInt(cadena [,base])
                                                                    */
♦ parseFloat(cadena)
                                                                    alert(cadena)
♦ toExponential([número-decimales])
                                                                    window.alert(cadena)
♦ toFixed([número-decimales])
                                                                    confirm( cadena)
♦ toString()
                                                                     window.confirm(cadena)
String
                                                                     prompt( cadena)
♦ length
                                                                    window.prompt(cadena)
◆ prototype
                                                                     console.log(cadena)
♦ charAt(posición)
                                                                     window.console.log(cadena)
♦ charCodeAt(posición)
                                                                     document.write(cadena)
◆ concat(lista-cadenas)
                                                                     document.writeln(cadena)
♦ indexOf(cadena [, posición])
                                                                    try {
♦ lastIndexOf(cadena [, posición])
                                                                    instrucciones-1 }
♦ includes(cadena [, posición])
                                                                     catch (variable-excepción if condición) { ...
                                                                    instrucciones-2}
◆ search(cadena)
                                                                     catch (variable-excepción) { ...
◆ substr(posición, número-caracteres)
                                                                    instrucciones-3 }
◆ substring(posición-1 [, posición-2])
                                                                    finally {
◆ slice(posición-1 [, posicón-2])
                                                                    instrucciones-n }
♦ toLowerCase()
                                                                     throw excepción
♦ toUpperCase()
                                                                    if( condición) {
♦ toString([base])
                                                                    instrucciones-1
◆ fromCharCode(lista-códigos-unicode)
                                                                    } else {
◆ codePointAt(posición)
                                                                     instrucciones-2
◆ replace(cadena-remplazar, cadena-nueva)
◆ replaceAll(cadena-remplazar, cadena-nueva)
                                                                    if( condición-1) {
♦ repeat(número)
                                                                    instrucciones-1
+ trim()
                                                                    } elseif (condición-2) {
trimLeft()
                                                                    instrucciones-2
trimRight()
                                                                    } elseif ( condición-3) {
valueOf()
                                                                     instrucciones-3
◆ padEnd(longitud)
◆ padStrart(longitud)
                                                                     else {
♦ endsWith(cadena[,longitud])
                                                                    instrucciones-n
◆ startsWith(cadena[,longitud])
                                                                         }
```

Félix Ángel Muñoz Bayón Pág. 1





switch(expresión) {	♦ toLocaleDateString([cadena-código-país])
case valor-1:	◆ toTimeString()
instrucciones-1	♦ toLocaleTimeString([cadena-código-país])
[break];	◆ toString()
case valor-2:	◆ toUTCString()
instrucciones-2	♦ toISOString()
break;	♦ toJSON()
default:	◆ valueOf()
instrucciones-n }	Math
I	♦ E: constante e valor aproximado 2.718.
while(condición) {	♦ LN2
instrucciones	♦ LN10
}	♦ LOG2E
, do {	♦ LOG10E
instrucciones	♦ PI
}while(condición)	♦ SQRT1_2
for(inicialización; condición-fin; incremento) {	♦ SQRT2
instrucciones	♦ abs(número)
}	◆ ceil(número)
for(variable in objeto) {	◆ floor(número)
instrucciones	◆ round(número)
}	◆ trunc(número)
for(variable of objeto) {	◆ exp(número)
instrucciones	◆ expm1(número)
}	◆ pow(base,exponente)
Date	◆ sqrt(número)
♦ new()	◆ cbrt(número)
◆ new(milisegundos)	♦ imul(número-1,número-2)
◆ new(cadena-fecha)	♦ log(número)
♦ new(año, mes, día[, hora,minuto,segundo,miliseg])	♦ log10(número)
♦ now()	♦ log2(número)
♦ parse(cadena)	♦ log1p(número)
♦ UTC(año, mes, día[,hora[,minuto[,segundo[,	♦ max(lista-números)
milisegundo]]]])	♦ min(lista-números)
♦ getDate()	
♦ getDay()	♦ sign(número)
♦ getMonth()	◆ cos(número)
♦ getFullYear()	♦ sin(número)
♦ getHours()	♦ tan(número)
♦ getMinutes()	♦ acos(número)
♦ getSeconds()	♦ asin(número)
♦ getMiliseconds()	♦ atan(número)
♦ getTime()	
♦ getTimezoneOffset()	◆ cosh(número)
♦ getUTCDate()	♦ sinh(número)
♦ getUTCDay()	♦ tanh(número)
♦ getUTCMonth()	♦ acosh(número)
♦ getUTCFullYear()	♦ asinh(número)
♦ getUTCHours()	♦ atanh(número)
♦ getUTCMinutes()	♦ clz32([número])
♦ getUTCSeconds()	♦ fround(número)
♦ getUTCMIliseconds()	♦ hypot(lista-números)
♦ setDate(día-mes)	navigator
◆ setMonth(número-mes)	
◆ setFullYear(año)	♦ appName
♦ setHours(hora)	♦ appVersion
◆ setMinutes(minutos)	♦ language
◆ setSeconds(segundo)	♦ mimeTypes
◆ setMiliseconds(milisegundos)	♦ platform
◆ setTime(milisegundos)	♦ plugins
◆ setUTCDate(día)	♦ userAgent
◆ setUTCMonth(número-mes)	◆ cookieEnabled
◆ setUTCFullYear(año)	
◆ setUTCHours(hora)	♦ onLine
◆ setUTCMinutes(minutos)	♦ javaEnabled()
◆ setUTCSeconds(segundos)	screen
◆ setUTCMIliseconds(milisegundos)	♦ availHeight
	♦ availWidth
♦ toDateString()	♦ height

Félix Ángel Muñoz Bayón Pág. 2





colorepth()	♦ width	♦ href
pixel beginth		
document		•
a applets		
* applets * eplace[uri]	♦ alinkColor	♦ search
* coolie	♦ anchors	♦ assign(url)
Cookie Window Appen Mills, nombre, parámetros, reemplaza_historial) Appen	♦ applets	◆ replace(url)
domain	◆ bgColor	◆ reload([del-servidor])
embeds	♦ cookie	window
Factor	♦ domain	• • • • • • • • • • • • • • • • • • • •
Forms islaM(v/olor) islams islaM(v/olor) eval(expressión Number (cadena) String(v/olor) islinity/v/olor) isli		
inages	-	•
lastModified		
linkColor	•	
links		
Ecastion		
eferrer		escape(cadena)
* title		unescape(cadena
wirtle(rectos) surfucciones su		
writel(extos) writel(extos) writel(extos) writel(extos) writel(extos) mombre-función([valores-parámetros]) winden(extos) var nombre-variable=function() { cuerpo-función var nombre-función]=new Function window wind		
virtlen(rextos) nombre-función([valores-parámetros]) var nombre-variable=function() { var nombre-función] = name function nombre-función(parámetro1 = valor1,) { var nombre-función(parámetro1 = valor1,) { var nombre-función(parámetro1 = valor1,) { var nombre-función(parámetro1, parámetro2, var nombre-función = (parámetro3)) => { var nombre-función = (parámetro3) var nombre-función = (parámetro3) var nombre-función = (parámetro3) var nombre-función; var nombre-función; var nombre-función; var nombre-función; var nombre-función; var nombre-función; var nombre-función = parámetro => var nombre-función; var nombre-función([ista-valore3) var nombre-función([ista-valore3) var nombre-función([ista-valore3) var nombre-función([ista-valore3) var nombre-función([ista-valore3) var nombre-variable=nombre-función([ista-valore3) var nombre-variable=nombre-función([ista-valore3) vield vield var nombre-variable=nombre-función([ista-valore3) vield var nombre-variable=nombre-función([ista-valore3) var nombre-variable=nombre-función([ista-valore3) var nombre-array = new Array() var nombre-array		_
clear()	◆ writeIn(textos)	-
Open()	◆ clear()	
close window length	◆ open()	
length	♦ close()	
name		window[nombre-función]=new Function
defaultStatus Status Sta	_	(lista-argumento,cuerpo-función)
* status * outerHeight * outerHeight * outerWidth * innerWidth * innerWidth * opener * closed * opener * closed * opener * closed * opener * tool too too too too too too too too to		
 outerHeight outerWidth outerWidth innerHeight cuerpo function (parámetro3) {		
outerWidth		•
innerHeight	_	
innerWidth		
closed	<u> </u>	}
parent	♦ opener	function(parámetros) instrucción-del-return
top cuerpo-función; return valor;	♦ closed	
self menubar tolobar tolobar statusbar statusbar statusbar scrollbars location history document alert(mensaje) prompt(mensaje, valor_por_defecto) hiblur() close() setTimeout(expresión, milisegundos) setTimeout(expresión, milisegundos) clearInterout(identificador) print() scrollBy(x,y) moveBy(x, y) resizeBy(x, y) host host var nombre-dunción ([parámetro => expresión function* nombre-función = parámetro => expresión function* nombre-función([lista-parámetros]) { cuerpo-función function* nombre-función([lista-parámetros]) { cuerpo-función function* nombre-función([lista-valores]) nombre-variable=nombre-función([lista-valores]) nombre-variable=nombre-función([lista-valores]) nombre-variable.next().value yield expresión yield var nombre-array = new Array() nombre-array = new Array(número-elementos) var nombre-array [lista-valores] nombre-array [lista-valores] nombre-array [lista-valores] nombre-array [posición] Array resizeBy(x, y) resizeBy(x, y) resizeBy(x, y) resizeTo(x,y) location hash pop() push(lista-valores)	♦ parent	
* menubar	•	
 toolbar statusbar scrollbars location history document alert(mensaje) prompt(mensaje, valor_por_defecto) hour() blur() close() setinterval(expresión, milisegundos) setinterval(identificador) clearTimeout(identificador) print() setinterval(identificador) clearTimeout(identificador) print() serollby(x,y) serollby(x,y) mombre-array = new Array(lista-valores) scrollby(x,y) mombre-array=[lista-valores] nombre-array=[lista-valores] pop() hash pop() push(lista-valores) 		return <i>valor;</i>
Statushar) var nombre-función = ([narámetros]) =>evnresión
scrollbars scrollbars location history document alert(mensaje) confirm(mensaje, valor_por_defecto) hombre-variable=nombre-función([lista-valores]) close() setTimeout(expresión, milisegundos) clearTimeout(identificador) clearTimeout(identificador) print() scrollBy(x,y) scrollBy(x,y) moweBy(x, y) moweBy(x, y) resizeBy(x, y) resizeBy(x, y) host clartimeout(identificador) setSitus (cuerpo-función ([lista-parámetros]) { cuerpo-función { function* nombre-función([lista-valores]) { nombre-variable=nombre-función([lista-valores]) { nombre-variable.next().value { yield expresión { yield expresión { yield* nombre-función-generadora(lista-valores) { var nombre-array = new Array() { var nombre-array = new Array() { var nombre-array = new Array() { var nombre-array = new Array(lista-valores) { var nombre-array = new Array(lista-		
location		
<pre>history document document function* nombre-función = parámetro =>expresión function* nombre-función([lista-parámetros]) { cuerpo-función } cuerpo-función } prompt(mensaje) focus() focus() hur() close() setInterval(expresión, milisegundos) setTimeout(expresión, milisegundos): clearInterval(identificador) clearTimeout(identificador) print() print() scrollBy(x,y) scrollBy(x,y) scrollTo(x, y) movePo(x, y) movePo(x,y) resizeBy(x, y) resizeBy(x, y) resizeBy(x, y) location hash host</pre>		return valor;
<pre>document</pre>		· ·
 alert(mensaje) confirm(mensaje) prompt(mensaje, valor_por_defecto) focus() blur() close() setInterval(expresión, milisegundos) clearInterval(identificador) clearInterval(identificador) print() print() print() print() scrollBy(x,y) scrollTo(x, y) moweBy(x, y) moweBy(x, y) resizeBy(x, y) resizeBy(x, y) resizeTo(x,y) host alert(mensaje) cuerpo-función prambre-función([lista-valores]) value <	•	
<pre>confirm(mensaje) prompt(mensaje, valor_por_defecto) focus() hous() hous() vield expresión vield expresión vield setInterval(expresión, milisegundos) setTimeout(expresión, milisegundos): clearInterval(identificador) vield* nombre-función-generadora(lista-valores) clearInterval(identificador) var nombre-array = new Array() print() var nombre-array = new Array(lista-valores) scrollBy(x,y) var nombre-array = new Array(número-elementos) scrollTo(x, y) var nombre-array=[lista-valores] moveBy(x, y) nombre-array[posición] resizeBy(x, y) resizeBy(x, y) inombre-array = new Array(lista-valores) var nombre-array=[lista-valores] nombre-array[posición] resizeBy(x, y) inombre-array[posición] resizeBo(x,y) inombre-array[posición] resizeBo(x,y) inombre-array[posición] inombre-array[posición]</pre>		
<pre>prompt(mensaje, valor_por_defecto) focus() focus() blur() vield expresión vield expresión vield expresión value setInterval(expresión, milisegundos) setTimeout(expresión, milisegundos): clearInterval(identificador) vield* nombre-función-generadora(lista-valores) clearTimeout(identificador) var nombre-array = new Array() print() var nombre-array = new Array(lista-valores) scrollBy(x,y) var nombre-array = new Array(número-elementos) scrollTo(x, y) var nombre-array=[] moveBy(x, y) moveTo(x,y) resizeBy(x, y) resizeBy(x, y) resizeTo(x,y) location hash pop() host</pre>		
 focus() blur() close() setInterval(expresión, milisegundos) setTimeout(expresión, milisegundos): clearInterval(identificador) clearTimeout(identificador) print() serollBy(x,y) scrollBy(x,y) moveBy(x, y) moveTo(x,y) resizeBy(x,y) resizeBy(x,y) resizeBy(x,y) hash host nombre-variable.next().value yield value value<td></td><td>•</td>		•
 blur() close() setInterval(expresión, milisegundos) setTimeout(expresión, milisegundos): clearInterval(identificador) clearTimeout(identificador) print() serollBy(x,y) scrollBy(x,y) moveBy(x, y) moveTo(x,y) resizeBy(x,y) resizeTo(x,y) hash host yield value var nombre-función-generadora(lista-valores) var nombre-array = new Array(lista-valores) var nombre-array=[lista-valores] nombre-array=[lista-valores] nombre-array[posición] Array length shift() pop() push(lista-valores) 	♦ focus()	
 setInterval(expresión, milisegundos) setTimeout(expresión,milisegundos): clearInterval(identificador) print() scrollBy(x,y) scrollTo(x, y) moveBy(x, y) moveTo(x,y) resizeBy(x,y) resizeBy(x,y) resizeTo(x,y) hash host value done yield* nombre-función-generadora(lista-valores) var nombre-función-generadora(lista-valores) var nombre-array = new Array(lista-valores) var nombre-array = new Array(número-elementos) var nombre-array=[lista-valores] nombre-array[lista-valores] nombre-array[posición] Array length pop() push(lista-valores) 		
 setTimeout(expresión,milisegundos): clearInterval(identificador) vield* nombre-función-generadora(lista-valores) clearTimeout(identificador) var nombre-array = new Array() print() var nombre-array = new Array(lista-valores) scrollBy(x,y) var nombre-array = new Array(número-elementos) scrollTo(x, y) var nombre-array=[lista-valores] moveBy(x, y) moveTo(x,y) resizeBy(x, y) resizeBy(x, y) resizeTo(x,y) length location hash pop() push(lista-valores) 		
 clearInterval(identificador) clearTimeout(identificador) print() scrollBy(x,y) scrollTo(x, y) moveBy(x, y) moveTo(x,y) resizeBy(x, y) resizeTo(x,y) hash host yield* nombre-función-generadora(lista-valores) var nombre-array = new Array(lista-valores) var nombre-array = new Array(número-elementos) var nombre-array=[lista-valores] nombre-array[lista-valores] nombre-array[posición] Array length shift() pop() push(lista-valores) 		♦ value
 clearTimeout(identificador) print() scrollBy(x,y) scrollTo(x, y) moveBy(x, y) moveTo(x,y) resizeBy(x, y) resizeTo(x,y) hash host var nombre-array = new Array(lista-valores) var nombre-array=[lista-valores] nombre-array[lista-valores] nombre-array[posición] Array plength pop() push(lista-valores) 		
 print() var nombre-array = new Array(lista-valores) scrollBy(x,y) var nombre-array = new Array(número-elementos) scrollTo(x, y) moveBy(x, y) moveTo(x,y) resizeBy(x, y) resizeTo(x,y) length location hash pop() host var nombre-array = new Array(lista-valores) var nombre-array=[lista-valores] nombre-array[posición] Array length pop() push(lista-valores) 	· · · · · · · · · · · · · · · · · · ·	
♦ scrollTo(x, y) var nombre-array=[] ♦ moveBy(x, y) var nombre-array=[lista-valores] ♦ moveTo(x,y) nombre-array[posición] ♦ resizeBy(x, y) Array ♦ resizeTo(x,y) ♦ length location ♦ shift() ♦ hash ♦ pop() ♦ host ♦ push(lista-valores)		
 ♦ moveBy(x, y) ♦ moveTo(x,y) ♦ resizeBy(x, y) ♦ resizeTo(x,y) ♦ length location ♦ hash ♦ pop() ♦ push(lista-valores) 		
$\begin{array}{lll} \bullet \ \operatorname{moveTo}(x,y) & nombre-array[posición] \\ \bullet \ \operatorname{resizeBy}(x,y) & \operatorname{Array} \\ \bullet \ \operatorname{resizeTo}(x,y) & \bullet \ \operatorname{length} \\ \operatorname{location} & \bullet \ \operatorname{shift}() \\ \bullet \ \operatorname{hash} & \bullet \ \operatorname{pop}() \\ \bullet \ \operatorname{host} & \bullet \ \operatorname{push(lista-valores)} \end{array}$		
♦ resizeBy(x, y) Array ♦ resizeTo(x,y) ♦ length location ♦ shift() ♦ hash ♦ pop() ♦ host ♦ push(lista-valores)		
♦ resizeTo(x,y) ♦ length location ♦ shift() ♦ hash ₱ pop() ♦ host ₱ push(lista-valores)		
location		♦ length
♦ host ♦ push(lista-valores)		♦ shift()
1.000	♦ hash	
♦ hostname ♦ unshift(lista-valores)	♦ host	
	♦ hostname	unshitt(lista-valores)

Félix Ángel Muñoz Bayón Pág. 3



onpopstate



```
◆ splice(inicio, nºelemento[, lista-valores])
                                                                    onredo
                                                                   onresize
♦ reverse()
                                                                    onstorage
♦ sort()
                                                                    onundo
♦ slice(inicio[,último])
                                                                    onunload
♦ indexOf(valor[,inicio])
                                                                   onblur
♦ lastIndexOf(valor[,inicio])
                                                                    onchange
includes(valor[,inicio])
                                                                    oncontextmenu
♦ concat(array)
                                                                   onfocus
♦ join(caracter)
                                                                   onformchange
♦ forEach(función)
                                                                    onforminput
♦ fill(valor [, inicio [, final]])
                                                                    oninput
♦ find(nombre-función)
                                                                   oninvalid
♦ findIndex(nombre-función)
                                                                   onreset
entries()
                                                                   onselect
♦ keys()
                                                                    onsubmit
◆ copyWithin(posición [. inicio [, final]])
                                                                   Event
◆ some(nombre-función|function ([parámetros]){cuerpo})
                                                                    ♦ type
♦ of(lista-valores)
                                                                    ♦ target
♦ from(objeto-map|objeto-set)
                                                                    ♦ cancelBubble
String
                                                                    ♦ offsetX
split(caracter)
                                                                    ♦ offsetY
for( nombre of nombre-array) instrucción

♦ clientX
for( nombre in nombre-array) instrucción
                                                                    ♦ clientY
Map
                                                                    ♦ screenX

    new()

                                                                    ♦ screenY
♦ size
                                                                    ♦ button
• get(clave)
                                                                    ♦ fromElement

♦ set(clave , valor )

                                                                    ♦ toElement
♦ has(clave)
                                                                    ♦ shiftKey
♦ delete(clave)
                                                                    ♦ ctrlKey
♦ clear()

♦ altKey
• entries()
                                                                    ♦ keyCode
♦ keys()
                                                                    ♦ charCode
♦ values()
                                                                    ♦ key
♦ toString()
                                                                    ♦ char
valueOf()
                                                                    window.onevento= nombre-metodo
♦ forEach(función (valor, clave, objeto) { cuerpo-función} )
                                                                    window.onevento = function([parámetro]) {
for( nombreofobjeto-map) instrucción
                                                                    cuerpo de la función
for ([clave, valor] ofobjeto-map) instrucción
                                                                    document.nombre-formulario.nombre-
onblur
                                                                   elemento.onevento=método
onchange
                                                                    document.nombre-formulario.nombre-
onclick
                                                                    elemento.onevento=function([parámetro]) {
ondblclick
                                                                    cuerpo de la función
onfocus
onkeydown
                                                                    document.forms.nombre-formulario
onkeypress
                                                                    document.nombre-formulario
onkeyup
                                                                    nombre-formulario
onload
                                                                    document.forms[posición]
onmousedown
                                                                    document.forms["nombre-formulario"]
onmousemove
                                                                    document.forms.item(posición)
onmouseout
                                                                    document.forms.namedItem("nombre-formulario")
onmouseover
                                                                   nombre-formulario.nombre-elemento
onmouseup
                                                                   referencia-formulario.nombre-elemento
onreset
                                                                   nombre-formulario.elements["nombre-elemento"]
onselect
                                                                   nombre-formulario.elements[índice]
onsubmit
                                                                   referencia-formulario.elements["nombre-elemento"]
onunload
                                                                   referencia-formulario.elements[índice]
onafterprint
                                                                   nombre-formulario.nombre-contenedor.elements[índice]
onbeforeprint
                                                                   referencia-formulario. nombre-contenedor.elements[índice]
onbeforeunload
                                                                   formulario.nombre-elemento.style.tipoestilo=valor
onerror
                                                                   formulario.nombre-elemento.nombre-atributo=valor
onhaschange
                                                                    var nombre= new RegExp(expresión-regular[,claves])
onload
                                                                   var nomExpresion= /expresión-regular/[claves];
onmessage
                                                                    ♦ x | y
onoffline
ononline
                                                                    ♦ [abc]
onpagehide
                                                                    ♦ [^abc]
onpageshow
                                                                    ♦ \b
```

Félix Ángel Muñoz Bayón Pág. 4



♦ \B

ESCUELA PUBLICA:

DE TODOS

DARA TODOS

- **♦** \d
- ♦ \D
- **♦** \f
- **♦** \n **♦** \r
- **♦ **s
- **♦**S
- **♦** \t
- **♦** \w
- **♦** \W
- **♦** \0
- **♦ \cX**
- ♦ \xhh
- ♦ \uxxxx ♦ \u{hhhh}
- ♦ \u{hhhhh}
 ♦ ^
- **♦** \$
- **+**
- **†?**
- ♦ {n}
- ♦ {n,m}
- ♦ {n,}
- **♦** (x)
- ♦ g

♦ i

expReg. test (cadena)cadena1.replace(expReg,cadena2)

document.cookie

Félix Ángel Muñoz Bayón Pág. 5