<script type="*tipo-lenguaje***" defer >**
código-script
</script>
<script src="*fichero.js***"></script>**
<*etiqueta-htmlevento*=*"instrucción-javascript"*>
<ahref="javascript:*instrucción*" >*enlace*
<ahref="javascript:*nombre-función*" >*enlace*
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*, … ;
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**
- **MIN_VALUE**
- **NaN**
- **NEGATIVE_INFINITY**
- **POSITIVE_INTINITY**
- **EPSILON**
- **MAX_SAFE_INTEGER**
- **MIN_SAFE_INTEGER**
- **isNaN(***número***)**
- **isFinite(***número***)**
- **isInteger(***número***)**
- **isSafeInteger(***número***)**
- **parseInt(***cadena* **[,***base***])**
- **parseFloat(***cadena***)**
- **toExponential([***número-decimales***])**
- **toFixed([***número-decimales***])**
- **toString()**

String

- **length**
- **prototype**
- **charAt(***posición***)**
- **charCodeAt(***posición***)**
- **concat(***lista-cadenas***)**
- **indexOf(***cadena* **[,** *posición***])**
- **lastIndexOf(***cadena* **[,** *posición***])**
- **includes(***cadena* **[,** *posición***])**
- **search(***cadena***)**
- **substr(***posición, número-caracteres***)**
- **substring(***posición-1* **[,** *posición-2***])**
- **slice(***posición-1* **[,** *posicón-2***])**
- **toLowerCase()**
- **toUpperCase()**
- **toString([***base***])**
- **fromCharCode(***lista-códigos-unicode***)**
- **codePointAt(***posición***)**
- **replace(***cadena-remplazar***,** *cadena-nueva***)**
- **replaceAll(***cadena-remplazar***,** *cadena-nueva***)**
- **repeat(número)**
- **trim()**
- **trimLeft()**
- **trimRight()**
- **valueOf()**
- **padEnd(***longitud***)**
- **padStrart(***longitud***)**
- **endsWith(***cadena***[,***longitud***])**
- **startsWith(***cadena***[,***longitud***])**

=
+=
-=
***=**
/=
%=
****=**
++
--
+
-
*
/
%
**
!
&&
||
~
<<
>>
>>>
==
!=
===
== =
!==
<
<=
>
>=
*Condición***?***valor1***:***valor2*
typeof *expresión*
//*comentario*
/*
comentario
***/**
alert(*cadena***)**
window.alert(*cadena* **)**
confirm(*cadena*)
window.confirm(*cadena*)
prompt(*cadena*)
window.prompt(*cadena*)
console.log(*cadena*)
window.console.log(*cadena*)
document.write(*cadena*)
document.writeln(*cadena*)

```
try {
instrucciones-1 }
catch ( variable-excepción if condición) {    ...
instrucciones-2}
catch ( variable-excepción ) {    …
instrucciones-3 }
finally {
instrucciones-n }
throw excepción
if( condición) {
instrucciones-1
}  else {
instrucciones-2
        }
if( condición-1) {
instrucciones-1
}  elseif ( condición-2) {
instrucciones-2
}  elseif ( condición-3) {
instrucciones-3        ...
    }
else  {
instrucciones-n
    }
```

switch(*expresión*) {
case *valor-1*:
instrucciones-1
[break];
case *valor-2*:
***instrucciones-2* ...**
break;
default:
instrucciones-n
}

while(*condición***) {**
instrucciones
}
do {
instrucciones
}while(*condición***)**
for(*inicialización***;***condición-fin***;** *incremento***) {**
instrucciones
}
for(*variable* **in** *objeto***) {**
instrucciones
}
for(*variable* **of** *objeto***) {**
instrucciones
}

Date

- **new()**
- **new(***milisegundos***)**
- **new(***cadena-fecha***)**
- **new(***año***,** *mes***,** *día***[,** *hora***,***minuto***,***segundo***,***miliseg***])**
- **now()**
- **parse(***cadena***)**
- ***UTC(****año, mes, día***[,***hora***[,**minuto**[,***segundo***[,** *milisegundo***]]]])**
- **getDate()**
- **getDay()**
- **getMonth()**
- **getFullYear()**
- **getHours()**
- **getMinutes()**
- **getSeconds()**
- **getMiliseconds()**
- **getTime()**
- **getTimezoneOffset()**
- **getUTCDate()**
- **getUTCDay()**
- **getUTCMonth()**
- **getUTCFullYear()**
- **getUTCHours()**
- **getUTCMinutes()**
- **getUTCSeconds()**
- **getUTCMIliseconds()**
- **setDate(***día-mes***)**
- **setMonth(***número-mes***)**
- **setFullYear(***año***)**
- **setHours(***hora***)**
- **setMinutes(***minutos***)**
- **setSeconds(***segundo***)**
- **setMiliseconds(***milisegundos***)**
- **setTime(***milisegundos***)**
- **setUTCDate(***día***)**
- **setUTCMonth(***número-mes***)**
- **setUTCFullYear(***año***)**
- **setUTCHours(***hora***)**
- **setUTCMinutes(***minutos***)**
- **setUTCSeconds(***segundos***)**
- **setUTCMIliseconds(***milisegundos***)**
- **toDateString()**
- **toLocaleDateString([***cadena-código-país***])**
- **toTimeString()**
- **toLocaleTimeString([***cadena-código-país***])**
- **toString()**
- **toUTCString()**
- **toISOString()**
- **toJSON()**
- **valueOf()**

Math

- **E: constante e valor aproximado 2.718.**
- **LN2**
- **LN10**
- **LOG2E**
- **LOG10E**
- **PI**
- **SQRT1_2**
- **SQRT2**
- **abs(***número***)**
- **ceil(***número***)**
- **floor(***número***)**
- **round(***número***)**
- **trunc(***número***)**
- **exp(***número***)**
- **expm1(***número***)**
- **pow(***base***,***exponente***)**
- **sqrt(***número***)**
- **cbrt(***número***)**
- **imul***(número-1***,***número-2***)**
- **log(***número***)**
- **log10(***número***)**
- **log2(***número***)**
- **log1p(***número***)**
- **max(***lista-números***)**
- **min(***lista-números***)**
- **random()**
- **sign(***número***)**
- **cos(***número***)**
- **sin(***número***)**
- **tan(***número***)**
- **acos(***número***)**
- **asin(***número***)**
- **atan(***número***)**
- **atan2(*y*,*x*)**
- **cosh(***número***)**
- **sinh(***número***)**
- **tanh(***número***)**
- **acosh(***número***)**
- **asinh(***número***)**
- **atanh(***número***)**
- **clz32([***número***])**
- **fround(***número***)**
- **hypot(***lista-números***)**

navigator

- **appCodeName**
- **appName**
- **appVersion**
- **language**
- **mimeTypes**
- **platform**
- **plugins**
- **userAgent**
- **cookieEnabled**
- **onLine**
- **javaEnabled()**

screen

- **availHeight**
- **availWidth**
- **height**

♦ **width**
♦ **colorDepth()**
♦ **pixelDepth()**

document

♦ **alinkColor**
♦ **anchors**
♦ **applets**
♦ **bgColor**
♦ **cookie**
♦ **domain**
♦ **embeds**
♦ **fgColor**
♦ **forms**
♦ **images**
♦ **lastModified**
♦ **linkColor**
♦ **links**
♦ **location**
♦ **referrer**
♦ **title**
♦ **vlinkColor**
♦ **write(***textos***)**
♦ **writeln(***textos***)**
♦ **clear()**
♦ **open()**
♦ **close()**

window

♦ **length**
♦ **name**
♦ **defaultStatus**
♦ **status**
♦ **outerHeight**
♦ **outerWidth**
♦ **innerHeight**
♦ **innerWidth**
♦ **opener**
♦ **closed**
♦ **parent**
♦ **top**
♦ **self**
♦ **menubar**
♦ **toolbar**
♦ **statusbar**
♦ **scrollbars**
♦ **location**
♦ **history**
♦ **document**
♦ **alert(***mensaje***)**
♦ **confirm(***mensaje***)**
♦ **prompt(***mensaje***,** *valor_por_defecto***)**
♦ **focus()**
♦ **blur()**
♦ **close()**
♦ **setInterval(***expresión***,** *milisegundos***)**
♦ **setTimeout(***expresión***,***milisegundos***):**
♦ **clearInterval(***identificador***)**
♦ **clearTimeout(***identificador***)**
♦ **print()**
♦ **scrollBy(***x***,***y***)**
♦ **scrollTo(***x***,** *y***)**
♦ **moveBy(***x***,** *y***)**
♦ **moveTo(***x***,***y***)**
♦ **resizeBy(***x***,** *y***)**
♦ **resizeTo(***x***,***y***)**

location

♦ **hash**
♦ **host**
♦ **hostname**
♦ **href**
♦ **pathname**
♦ **port**
♦ **protocol**
♦ **search**
♦ **assign(***url***)**
♦ **replace(***url***)**
♦ **reload([***del-servidor***])**

window

♦ **open(***URL***,** *nombre***,** *parámetros***,** *reemplaza_historial***)**

parseInt(*cadena* **[,** *base***])**
parseFloat(*cadena***)**
isNaN(*valor***)**
eval(*expresión* **)**
Number(*cadena***)**
String(*valor***)**
isFinity(*valor***)**
escape(*cadena***)**
unescape(cadena

function ***nombre-función*****(** **[*****lista-parámetros*****]** **) {**
instrucciones
}
nombre-función**([*****valores-parámetros*****]** **)**
var ***nombre-variable*****=function() {**
cuerpo-función
}
window[***nombre-función*****]=new Function**
(***lista-argumento*****,*****cuerpo-función*****)**
function ***nombre-función*****(*****parámetro1=valor1*****, ..) {**
cuerpo
}
function ***nombre-función*****(*****parámetro1*****,** ***parámetro2*****,**
…parámetro3**) {**
cuerpo
}
function(***parámetros*****)** ***instrucción-del-return***
return [***lista-valores*****]**
var ***nombre-función*** **= ([*****parámetros*****]) => {**
cuerpo-función;
return ***valor;***
}
var ***nombre-función*** **= ([*****parámetros*****]) =>*****expresión***
var ***nombre-función*** **=** ***parámetro*** **=> {**
cuerpo-función;
return ***valor;***
}
var ***nombre-función*** **=** ***parámetro*** **=>*****expresión***
function* ***nombre-función*****([*****lista-parámetros*****]) {**
cuerpo-función
}
nombre-variable**=*****nombre-función*****([*****lista-valores*****])**
nombre-variable**.next().value**
yield ***expresión***
yield

♦ **value**
♦ **done**

yield* ***nombre-función-generadora*****(*****lista-valores*****)**
var ***nombre-array*** **= new Array()**
var ***nombre-array*** **= new Array(*****lista-valores*****)**
var ***nombre-array*** **= new Array(*****número-elementos*****)**
var ***nombre-array*****=[]**
var ***nombre-array*****=[*****lista-valores*****]**
nombre-array**[*****posición*****]**

Array

♦ **length**
♦ **shift()**
♦ **pop()**
♦ **push(lista-valores)**
♦ **unshift(lista-valores)**

♦ **splice(inicio, nºelemento[, lista-valores])**
♦ **reverse()**
♦ **sort()**
♦ **slice(inicio[,último])**
♦ **indexOf(valor[,inicio])**
♦ **lastIndexOf(valor[,inicio])**
♦ **includes(valor[,inicio])**
♦ **concat(array)**
♦ **join(caracter)**
♦ **forEach(función)**
♦ **fill(valor [, inicio [, final]])**
♦ **find(nombre-función)**
♦ **findIndex(nombre-función)**
♦ **entries()**
♦ **keys()**
♦ **copyWithin(posición [. inicio [, final]])**
♦ **some(nombre-función|function ([parámetros]){cuerpo})**
♦ **of(lista-valores)**
♦ **from(objeto-map|objeto-set)**

String
♦ **split(caracter)**

for(*nombre* of *nombre-array*) *instrucción*
for(*nombre* in *nombre-array*) *instrucción*

Map
♦ **new()**
♦ **size**
♦ **get(clave)**
♦ **set(clave , valor)**
♦ **has(clave)**
♦ **delete(clave)**
♦ **clear()**
♦ **entries()**
♦ **keys()**
♦ **values()**
♦ **toString()**
♦ **valueOf()**
♦ **forEach(función (valor, clave, objeto) { cuerpo-función})**

for(*nombre*of*objeto-map*) *instrucción*
for ([clave, valor] of*objeto-map*) *instrucción*

onblur
onchange
onclick
ondblclick
onfocus
onkeydown
onkeypress
onkeyup
onload
onmousedown
onmousemove
onmouseout
onmouseover
onmouseup
onreset
onselect
onsubmit
onunload
onafterprint
onbeforeprint
onbeforeunload
onerror
onhaschange
onload
onmessage
onoffline
ononline
onpagehide
onpageshow
onpopstate
onredo
onresize
onstorage
onundo
onunload
onblur
onchange
oncontextmenu
onfocus
onformchange
onforminput
oninput
oninvalid
onreset
onselect
onsubmit

Event
♦ **type**
♦ **target**
♦ **cancelBubble**
♦ **offsetX**
♦ **offsetY**
♦ **clientX**
♦ **clientY**
♦ **screenX**
♦ **screenY**
♦ **button**
♦ **fromElement**
♦ **toElement**
♦ **shiftKey**
♦ **ctrlKey**
♦ **altKey**
♦ **keyCode**
♦ **charCode**
♦ **key**
♦ **char**

window.on*evento*= *nombre-metodo*
window.on*evento* = **function**([*parámetro*]) {
cuerpo de la función
}
document.*nombre-formulario.nombre-elemento*.on*evento=método*
document.nombre-formulario.*nombre-elemento*.on*evento*=**function**([parámetro]) {
cuerpo de la función
}
document.forms.*nombre-formulario*
document.*nombre-formulario*
nombre-formulario
document.forms[*posición*]
document.forms["*nombre-formulario"*]
document.forms.item(*posición*)
document.forms.namedItem(“*nombre-formulario***”)**
nombre-formulario.nombre-elemento
referencia-formulario.nombre-elemento
nombre-formulario.**elements**[“*nombre-elemento*”]
nombre-formulario.**elements**[*índice*]
referencia-formulario.**elements**[“*nombre-elemento*”]
referencia-formulario.**elements**[*índice*]
nombre-formulario.nombre-contenedor.**elements**[*índice*]
referencia-formulario. nombre-contenedor.**elements**[*índice*]
formulario.nombre-elemento.**style**.*tipoestilo=valor*
*formulario.nombre-elemento.nombre-atributo=va*lor
var nombre= new RegExp(expresión-regular[,claves])
var nomExpresion= /expresión-regular/[claves];

♦ **x|y**
♦ **.**
♦ **[abc]**
♦ **[^abc]**
♦ **\b**

- **\B**
- **\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