

<code><script type="tipo-lenguaje" defer ></code>	<code>%=</code>
<code> código-script</code>	<code>**=</code>
<code></script></code>	<code>++</code>
<code><script src="fichero.js"></script></code>	<code>--</code>
<code><etiqueta-html evento="instrucción-javascript"></code>	<code>+</code>
<code>enlace</code>	<code>-</code>
<code>enlace</code>	<code>*</code>
<code>const nombre-constante-1=valor-1</code>	<code>/</code>
<code> [, nombre-constante-2=valor-2] ...</code>	<code>%</code>
<code>var nombre-variable;</code>	<code>**</code>
<code>let nombre-variable;</code>	<code>!</code>
<code>var nombre-variable=valor ;</code>	<code>&&</code>
<code>let nombre-variable=valor ;</code>	<code> </code>
<code>var nombre-variable-1 ,nombre-variable-2, ... ;</code>	<code>~</code>
<code>let nombre-variable-1 ,nombre-variable-2, ... ;</code>	<code><<</code>
<code>var nombre-variable-1 [=valor-1]</code>	<code>>></code>
<code> [,nombre-variable-2 [=valor-2]]... ;</code>	<code>>>></code>
<code>let nombre-variable-1 [=valor-1],</code>	<code>==</code>
<code> [nombre-variable-2 [=valor-2]]... ;</code>	<code>!=</code>
Number	<code>===</code>
♦ MAX_VALUE	<code>== =</code>
♦ MIN_VALUE	<code>!= =</code>
♦ NaN	<code><</code>
♦ NEGATIVE_INFINITY	<code><=</code>
♦ POSITIVE_INFINITY	<code>></code>
♦ EPSILON	<code>>=</code>
♦ MAX_SAFE_INTEGER	<code>Condición?valor1:valor2</code>
♦ MIN_SAFE_INTEGER	<code>typeof expresión</code>
♦ isNaN(número)	<code>//comentario</code>
♦ isFinite(número)	<code>/*</code>
♦ isInteger(número)	<code>comentario</code>
♦ isSafeInteger(número)	<code>*/</code>
♦ parseInt(cadena [,base])	<code>alert(cadena)</code>
♦ parseFloat(cadena)	<code>window.alert(cadena)</code>
♦ toExponential([número-decimales])	<code>confirm(cadena)</code>
♦ toFixed([número-decimales])	<code>window.confirm(cadena)</code>
♦ toString()	<code>prompt(cadena)</code>
String	<code>window.prompt(cadena)</code>
♦ length	<code>console.log(cadena)</code>
♦ prototype	<code>window.console.log(cadena)</code>
♦ charAt(posición)	<code>document.write(cadena)</code>
♦ charCodeAt(posición)	<code>document.writeln(cadena)</code>
♦ concat(lista-cadenas)	<code>try {</code>
♦ indexOf(cadena [, posición])	<code>instrucciones-1 }</code>
♦ lastIndexOf(cadena [, posición])	<code>catch (variable-excepción if condición) { ...</code>
♦ includes(cadena [, posición])	<code>instrucciones-2 }</code>
♦ search(cadena)	<code>catch (variable-excepción) { ...</code>
♦ substr(posición, número-caracteres)	<code>instrucciones-3 }</code>
♦ substring(posición-1 [, posición-2])	<code>finally {</code>
♦ slice(posición-1 [, posición-2])	<code>instrucciones-n }</code>
♦ toLowerCase()	<code>throw excepción</code>
♦ toUpperCase()	<code>if(condición) {</code>
♦ toString([base])	<code>instrucciones-1</code>
♦ fromCharCode(lista-códigos-unicode)	<code>} else {</code>
♦ codePointAt(posición)	<code>instrucciones-2</code>
♦ replace(cadena-reemplazar, cadena-nueva)	<code>}</code>
♦ repeat(número)	<code>if(condición-1) {</code>
♦ trim()	<code>instrucciones-1</code>
♦ trimLeft()	<code>} else if (condición-2) {</code>
♦ trimRight()	<code>instrucciones-2</code>
♦ valueOf()	<code>} else if (condición-3) {</code>
♦ padEnd(longitud)	<code>instrucciones-3 ...</code>
♦ padStart(longitud)	<code>}</code>
♦ endsWith(cadena[,longitud])	<code>else {</code>
♦ startsWith(cadena[,longitud])	<code>instrucciones-n</code>
<code>=</code>	<code>}</code>
<code>+=</code>	
<code>-=</code>	
<code>*=</code>	
<code>/=</code>	

```

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()
♦ getMilliseconds()
♦ getTime()
♦ getTimezoneOffset()
♦ getUTCDate()
♦ getUTCDay()
♦ getUTCMonth()
♦ getUTCFullYear()
♦ getUTCHours()
♦ getUTCMinutes()
♦ getUTCSeconds()
♦ getUTCMilliseconds()
♦ setDate(día-mes)
♦ setMonth(número-mes)
♦ setFullYear(año)
♦ setHours(hora)
♦ setMinutes(minutos)
♦ setSeconds(segundo)
♦ setMilliseconds(milisegundos)
♦ setTime(milisegundos)
♦ setUTCDate(día)
♦ setUTCMonth(número-mes)
♦ setUTCFullYear(año)
♦ setUTCHours(hora)
♦ setUTCMinutes(minutos)
♦ setUTCSeconds(segundos)
♦ setUTCMilliseconds(milisegundos)
♦ toString()
♦ 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
♦ appName
♦ appVersion
♦ language
♦ mimeType
♦ 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)
isFinite(valor)
escape(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[,última])**
 ♦ **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

var nombre-variable = new Object()
nombre-objeto.nombre-propiedad=valor
nombre-objeto[nombre-propiedad]= valor
nombre-objeto.nombre-método= function([parámetros]) {
    cuerpo-método
}
nombre-objeto[nombre-método]=function ([parámetros]) {
    cuerpo-método
}
nombre-objeto.nombre-propiedad

function nombre-pseudoclase( lista-parámetros ) {
    cuerpo
}
var nombre-objeto= new nombre-pseudoclase(
    valores-parámetros)
this.nombre-propiedad=valor
this.__defineGetter__(nombre-propiedad,
    function(parámetro) { cuerpo})
this.__defineSetter__(nombre-propiedad,
    function(parámetro) { cuerpo})
this.nombre-método=function ([parámetros]) {
    cuerpo-método
}
this.nombre-propiedad
nombre-pseudoclase.prototype.nombre-propiedad=valor
nombre-objeto.__defineGetter__(nombre-propiedad,
    function(parámetro) { cuerpo})
nombre-pseudoclase.prototype.__defineGetter__(
    nombre-propiedad, function(parámetro) { cuerpo})
nombre-objeto.__defineSetter__(nombre-propiedad,
    function(parámetro) { cuerpo})
nombre-pseudoclase.prototype.__defineSetter__(
    nombre-propiedad, function(parámetro) { cuerpo})
nombre-pseudoclase.prototype.nombre-método
    =function([parámetros]) {
        cuerpo}
nombre-clase-padre.call(this, parámetros)

```

```

class nombre-clase () {
    [ constructor ([parámetros]) {
        instrucciones
    }]
    [ [static] nombre-método (parámetros) {
        instrucciones}]
}
get nombre-propiedad(){
    cuerpo
    return expresión;
}
set nombre-propiedad(parámetro){
    cuerpo
}
class nombre-clase () extends clase-padre {
    [ constructor ([parámetros]) {
        super([parámetros]);
        instrucciones
    }]
    [ [static] nombre-método (parámetros) {
        instrucciones}]
}
super.nombre-método

var nombre-objeto = {
    cuerpo
}
nombre-propiedad :valor,
get nombre-propiedad() { cuerpo} ,
set nombre-propiedad(parámetro) { cuerpo} ,
nombre-método :function ([parámetros]) {
    cuerpo
}
this.nombre-propiedad
for( variable in objeto ) {
    cuerpo
}
for (variable of objeto) {
    cuerpo
}
with (objeto) {
    instrucciones
}

Object.create(nombre-objeto, {definición-propiedades})
nombre-propiedad : {
    value:valor,
    writable:true|false ,
    enumerable:true|false,
    configurable:true|false
}
nombre-propiedad : {
    get :function([parametros]) { cuerpo-función } ,
    set :function(parámetro[, parametros]) {
        cuerpo-función } ,
    enumerable:true|false ,
    configurable:true|false
}
Object.defineProperty(nombre-objeto,
    nombre-propiedad, descriptor-propiedad)
Object.defineProperties(objeto, descriptores-propiedades)
Object.is(objeto-1,objeto-2)
Object.assign(destino, lista-objetos)
Object.keys(objeto)

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.onevento= nombre-metodo
window.onevento = function([parámetro]) {
  cuerpo de la función
}
document.nombre-formulario.nombre-
elemento.onevento=método
document.nombre-formulario.nombre-
elemento.onevento=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"]
referencia-formulario.elements["nombre-elemento"]
formulario.nombre-elemento.style.tipoestilo=valor
formulario.nombre-elemento.nombre-atributo=valor
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{hhh}
♦ \u{hhhhh}
♦ ^
♦ $
♦ *
♦ +
♦ ?
♦ {n}
♦ {n,m}
♦ {n,}
♦ (x)
♦ g
♦ i
expReg.test(cadena)
cadena1.replace(expReg,cadena2)
document.cookie
Document
♦ doctype
♦ documentElement
♦ children
♦ firstElementChild
♦ lastElementChild
♦ childElementCount
♦ documentUri
♦ URL
♦ implementation
♦ compactMode
♦ characterSet
♦ charset
♦ inputEncoding
♦ createElement(tipo)
♦ createTextNode(cadena)
♦ createAttribute(nombre-atributo)
♦ getElementById(identificador)
♦ getElementsByTagName(nombre-etiqueta)
♦ getElementsByClassName(nombre-clase)
♦ querySelector(selector)

```

- ◆ `querySelectorAll(selector)`
- ◆ `importNode(nodo, subárbol)`
- ◆ `adoptNode(nodo)`
- ◆ `append(nodo ...)`
- ◆ `prepend(nodo ...)`
- ◆ `getElementsByName(nombre)`
- ◆ `createEvent(nombre-evento)`
- Node**
 - ◆ `childNodes`
 - ◆ `parentNode`
 - ◆ `parentElement`
 - ◆ `firstChild`
 - ◆ `lastChild`
 - ◆ `nextSibling`
 - ◆ `previousSibling`
 - ◆ `nodeName`
 - ◆ `nodeType`
 - ◆ `nodeValue`
 - ◆ `ownerDocument`
 - ◆ `textContent`
 - ◆ `innerHTML`
 - ◆ `baseURI`
 - ◆ `hasChildNodes()`
 - ◆ `appendChild(nodo)`
 - ◆ `insertBefore(nuevo-nodo, nodo-posición)`
 - ◆ `removeChild(nodo)`
 - ◆ `replaceChild(nuevo-nodo, nodo-anterior)`
 - ◆ `cloneNode(subárbol)`
 - ◆ `isEqualNode(nodo)`
 - ◆ `isSameNode(nodo)`
 - ◆ `contains(nodo)`
 - ◆ `compareDocumentPosition(nodo)`
- Element**
 - ◆ `tagName`
 - ◆ `children`
 - ◆ `firstElementChild`
 - ◆ `lastElementChild`
 - ◆ `childElementCount`
 - ◆ `previousElementSibling`
 - ◆ `nextElementSibling`
 - ◆ `attributes`
 - ◆ `namespaceURI`
 - ◆ `prefix`
 - ◆ `localName`
 - ◆ `id`
 - ◆ `className`
 - ◆ `querySelector(selector)`
 - ◆ `querySelectorAll(selector)`
 - ◆ `getElementsByName(nombre-etiqueta)`
 - ◆ `getElementsByClassName(nombre-clase)`
 - ◆ `append(nodo ...)`
 - ◆ `prepend(nodo ...)`
 - ◆ `after(nodo ...)`
 - ◆ `before(nodo ...)`
 - ◆ `remove()`
 - ◆ `replaceWith(nodo ...)`
 - ◆ `getAttribute(nombre)`
 - ◆ `getAttributeNode(nombre)`
 - ◆ `hasAttribute(nombre)`
 - ◆ `removeAttribute(nombre)`
 - ◆ `removeAttributeNode(nodo-attr)`
 - ◆ `setAttribute(nombre,valor)`
 - ◆ `setAttributeNode(nodo-attr)`
 - ◆ `closest(selector)`
 - ◆ `matches(selector)`
 - ◆ `insertAdjacentElement(donde, elemento)`
 - ◆ `insertAdjacentText(donde,texto)`
- NodeList**
 - ◆ `length`
 - ◆ `item(posición)`

- HTMLCollection**
 - ◆ `length`
 - ◆ `item(posición)`
 - ◆ `namedItem(nombre)`
- NamedNodeMap**
 - ◆ `length`
 - ◆ `getNamedItem(nombre)`
 - ◆ `removeNamedItem(nombre)`
 - ◆ `setNamedItem(nodo)`
 - ◆ `item(posición)`
- Attr**
 - ◆ `name`
 - ◆ `ownerElement`
 - ◆ `specified`
 - ◆ `value`
 - ◆ `namespaceURI`
 - ◆ `prefix`
 - ◆ `localName`
 - ◆ `nodeName`
- HtmlTableElement**
 - ◆ `caption`
 - ◆ `createCaption()`
 - ◆ `deleteCaption()`
 - ◆ `tHead`
 - ◆ `createTHead()`
 - ◆ `deleteTHead()`
 - ◆ `tFoot`
 - ◆ `createTFoot()`
 - ◆ `deleteTFoot()`
 - ◆ `tBodies`
 - ◆ `createTBody()`
 - ◆ `rows`
 - ◆ `insertRow([índice])`
 - ◆ `deleteRow(índice)`
- HTMLTableSectionElement**
 - ◆ `rows`
 - ◆ `insertRow([índice])`
 - ◆ `deleteRow(índice)`
- HTMLTableRowElement**
 - ◆ `rowIndex`
 - ◆ `sectionRowIndex`
 - ◆ `cells`
 - ◆ `insertCell([índice])`
 - ◆ `deleteCell(índice)`
- document.body**
- document.head**
- addEventListener(nombreDeEvento,función,soloCaptura)**
- attachEvent(nombre-evento, función-ejecutar)**
- Evento**
 - ◆ `currentTarget`
 - ◆ `eventPhase`
 - ◆ `target`
 - ◆ `timeStamp`
 - ◆ `type`
 - ◆ `cancelBubble`
 - ◆ `bubbles`
 - ◆ `cancelable`
 - ◆ `defaultPrevented`
 - ◆ `composed`
 - ◆ `isTrusted`
 - ◆ `preventDefault()`
 - ◆ `stopPropagation()`
 - ◆ `stopImmediatePropagation()`
 - ◆ `composedPath()`
 - ◆ `removeEventListener(nombre-evento, nombre-función, fase-captura)`
 - ◆ `detachEvent(nombre-evento , nombre-función)`
- <http://code.jquery.com/jquery-3.7.1.js>
- <http://code.jquery.com/jquery-3.7.1.min.js>

<http://ajax.aspnetcdn.com/ajax/jquery/jquery-3.7.1.min.js>
<http://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js>

```
$(document).ready(function() { cuerpo-función});
$(function() {cuerpo-función});
$( nombre-función);
$("")
$("etiqueta")
$("etiqueta.clase")
$(".clase")
$("etiqueta#identificador")
$("#identificador")
$("selector-1,selector-2, selector-3,..")
$("selector-1 selector-2")
$("selector-1>selector-2")
$("selector-1 ~selector-2")
$("selector-1+selector-2")
$("etiqueta[atributo]")
$("[atributo]")
$("etiqueta[atributo='valor']")
$("[atributo='valor']")
$("etiqueta[atributo!= 'valor']")
$("[atributo!= 'valor']")
$("etiqueta[atributo|= 'valor']")
$("[atributo|= 'valor']")
$("etiqueta[atributo^= 'valor']")
$("[atributo^= 'valor']")
$("etiqueta[atributo$= 'valor']")
$("[atributo$= 'valor']")
$("etiqueta[atributo*= 'valor']")
$("[atributo*= 'valor']")
$("etiqueta[atributo~= 'valor']")
$("[atributo~= 'valor']")
$(":text")
$(":button")
$(":checkbox")
$(":file")
$(":header")
$(":image")
$(":input")
$(":password")
$(":radio")
$(":reset")
$(":selected")
$(":submit")
$(":checked")
$(":enabled")
$(":disabled")
$(":hidden")
$(":visible")
$(":animated")
$(":focus")
$(":root")
$("selector:parent")
$("selector:empty")
$("selector-1 selector-2:first-child")
$("selector-1 selector-2:last-child")
$("selector-1 selector-2:nth-child(posición)")
$("selector-1 selector-2:nth-last-child(posición)")
$("selector-1 selector-2:only-child")
$("selector:eq(posición)")
$("selector:first")
$("selector:last")
$("selector:gt(índice)")
$("selector:lt(índice)")
$("selector:even")
$("selector:odd")
$("selector:contains(texto)")
$("selector-1:has(selector-2)")
$("selector-1:not(selector-2)")
$("selector:first-of-type")
$("selector:last-of-type")
```

```
$("selector:nth-of-type(posición)")
$("selector:nth-last-of-type(posición)")
$("selector:only-of-type")
$("lang(código-lenguaje)")
.length
.add("selector")
.not("selector")
.filter("selector|función")
.has(elemento)
.eq(posición)
.first()
.last()
.slice(inicio[,fin])
.children()
.contents()
.find(selector)
.next()
.prev()
.nextAll()
.nextUntil(selector)
.prevAll()
.siblings()
.parent()
.parents()
.clone()
.index(elemento)
.attr(atributo, valor)
.attr({formato-estilos-css})
.css("nombre-propiedad")
.css(array-propiedades)
.css(propiedad, valor)
.css({estilos-css})
.css(nombre-propiedad,function(indice, valor-anterior){cuerpo})
.height()
.height(valor)
.height(function( indice, valor-anterior ){ cuerpo })
.innerHeight()
.innerHeight(valor)
.innerHeight(function( indice, valor-anterior ){ cuerpo })
.outerHeight()
.outerHeight(valor)
.outerHeight(function( indice, valor-anterior ){ cuerpo })
.width()
.width(valor)
.width(function( indice, valor-anterior ){ cuerpo })
.innerWidth()
.innerWidth(valor)
.innerWidth(function( indice, valor-anterior ){ cuerpo })
.outerWidth()
.outerWidth(valor)
.outerWidth(function( indice, valor-anterior ){ cuerpo })
.position()
.offset()
.offset(coordenadas)
.offset(function( indice, valor-anterior ){ cuerpo })
.scrollLeft()
.scrollLeft(valor)
.scrollTop()
.scrollTop(valor)
.addClass(nueva-clase-elemento)
.removeClass(clase)
.toggleClass(clase)
.before(contenido)
.before(function(){cuerpo-función})
.before(nombre-función)
.prepend(contenido)
.prepend(nombre-función)
.prepend(function(){cuerpo-función})
.after(contenido)
.after(nombre-función)
.after(function(){cuerpo-función})
```

```

.append(contenido)
.append(nombre-función)
.append(function(){cuerpo-función})
.html()
.html([contenido])
.text()
.text([contenido])
.val()
.val(valor)
.remove()
.detach()
.hide()
.show([tiempo-milisegundo])
.fadeIn([tiempo-milisegundos])
.fadeOut([tiempo-milisegundos])
.slideDown()
.slideUp()
.slideToggle()
.evento(function() { cuerpo-función})
.toggle(function() {cuerpo-función1}, function()
{cuerpo-función-2})
.delegate(selector, evento, nombre-función)
.delegate(selector, evento, function(){ cuerpo-función })
.delegate(selector, evento, datos, function()
{ cuerpo-función })
.delegate(selector, { evento-1: function(){ cuerpo-función-1
} [, evento-2: function(){ cuerpo-función-2 }..})
.on(selector, evento, nombre-función)
.on(selector, evento, function(){ cuerpo-función })
.on(selector, evento, datos, function(){ cuerpo-función })
.on(selector, { evento-1: function(){ cuerpo-función-1 } [,
evento-2: function(){ cuerpo-función-2 }..})
.undelegate(selector)
.undelegate(evento)
.undelegate(selector, evento)
.undelegate(selector, evento, nombre-función)
.undelegate()
.off(evento)
.off(evento, selector)
.off(evento, selector, nombre-función)
.off()
Eventos
♦ load
♦ ready
♦ unload
♦ click
♦ dblclick
♦ mousedown
♦ mouseup
♦ mouseleave
♦ mousemove
♦ mouseenter
♦ mouseout
♦ mouseover
♦ submit
♦ hover
♦ change
♦ contentmenu
♦ error
♦ focus
♦ focusin
♦ focusout
♦ keydown
♦ keyup
♦ keypress
♦ resize
♦ scroll
♦ select
$(selector).nombreevento()
Event
♦ target

```

```

♦ currentTarget
♦ delegateTarget
♦ relatedTarget
♦ type
♦ data
♦ result
♦ pageX
♦ pageY
♦ which
♦ metaKey
♦ namespace
♦ timeStamp
♦ preventDefault()
♦ isDefaultPrevented()
♦ stopPropagation()
♦ stopImmediatePropagation()
♦ isImmediatePropagationStopped()

```

```

nombre-objeto = new XMLHttpRequest()
nombre-objeto=new ActiveXObject("Microsoft.XMLHTTP")
nombre-objeto.onreadystatechange = nombre_funcion
nombre-objeto.onload = nombre_funcion
nombre-objeto.onprogress = nombre_funcion
nombre-objeto.onloadend = nombre_funcion
nombre-objeto.open("GET"|"POST","fichero"
[,asincrona])
nombre-objeto.setRequestHeader("nombre","valor");
"Content-Type","application/x-www-form-urlencoded"
"Content-Type","application/json"
nombre-objeto.send([parámetros | null])
nombre-objeto.readyState
nombre-objeto.status
nombre-objeto.responseText
nombre-objeto.responseXML
JSON.stringify(objeto)
JSON.parse(objeto)

```

```

fetch(url, objeto).then(función-correcto).catch(función-
error)
objeto={
method:GET|POST,
headers: {cabeceras},
body:datos,
cache: default |reload | no-cache | no-store | forcé-cache
| only-if-cached
}
let objeto = new DOMParser();
let variable = objeto.parseFromString(datos,
"application/xml");

```

```

$(selector).load(fichero)
$(selector).load(fichero?nombre-1=valor-1&nombre-
2=valor2..., función)
param({ nombre-parámetro-1 : valor-1 , nombre-
parametro-2 : valor-2 ... })
$(selector).load(fichero,{ nombre-parámetro-1 : valor-1 ,
nombre-parametro-2 : valor-2 ... }, función);
$.get(fichero [, parámetros] [, function-vuelve] [, tipo-dato])
$.get(opciones)
$.post(fichero [, parámetros] [, function-vuelve] [, tipo-
dato])
$.post(opciones)
$.ajax(url [, opciones ])
$.ajax([opciones])
$.ajaxSetup(opciones)
♦ url:dirección
♦ data:objeto
♦ dataType:xml|json|html|text
♦ method:GET|POST
♦ headers:cabecera
♦ success:función

```


- ◆ **complete:***función*
- ◆ **error:***función*
- ◆ **contentType:***false*,
- ◆ **processData:***false*
- \$.ajaxComplete**(*función*)
- \$.ajaxSuccess**(*función*)
- \$.ajaxError**(*función*)
- \$.ajaxSend**(*función*)
- \$.ajaxStart**(*función*)
- \$.ajaxStop**(*función*)
- \$.getJSON**(*url* [, *datos*] [, *función*])
- fetch**(*url* [, *objeto-configuración*]).**then**(*nombre-función*)
- .catch**(*nombre-función*)
- ◆ **method:**"GET"|"POST"
- ◆ **headers:** *cabecera*
- ◆ **body:** *dato*
- ◆ **cache:**no-cache