

# JavaScript

## Índice

Variables, Expresiones y Estructuras de Control.....	2
Objetos (Object).....	4
Objeto Math.....	4
Objeto Number.....	4
Objeto Array.....	4
Objeto String.....	5
Objeto Date.....	6
Objeto JSON.....	6
Expresiones Regulares.....	8
Window Script Host.....	10
ActiveX WScript.Shell.....	11
ActiveX Shell.Application.....	12
ActiveX Scripting.FileSystemObject.....	14
ActiveX WScript.Network.....	16
Teoría de objetos.....	17
JavaScript Jargon.....	19

[http://www.w3schools.com/jsref/jsref\\_obj\\_date.asp](http://www.w3schools.com/jsref/jsref_obj_date.asp) JavaScript Tutorial

<http://msdn2.microsoft.com/en-us/library/ms950396.aspx>

# Variables, Expresiones y Estructuras de Control

## Variables

siempre implícitas, [A-Za-Z0-9\_] no pueden empezar por número, distinguen mayúsculas

números enteros: (NN)decimal (0NN)octal (0xNN)hexadecimal

numeros reales: (NN[.NN][eNN])

booleanos: (1)true, (0>false

cadenas: (\a\b\f\n\r\t\\\'")caracteres\_especiales .length .substring(i,j)

sin valor: null (se convierte a todos los tipos 0, "", false)

var variable=valor; // se puede redefinir a otro tipo, dentro de funciones con var locales, sin var globales

tipos de valores: array boolean date number object string function undefined

if (typeof variable === 'tipo') ...

## Expresiones

el primer valor define el resultado de la expresión, los demás se convierten

proceso por líneas, varios comandos en una misma línea separados por ';'

operadores de asignación: =, +=, -=, \*=, /=, %=, &=, |=, ^=, <<=, >>=, >>>=

operador condicional: ... ? ... : ...

operadores lógicos: ||, &&

operadores de comparación estricta: ===, !== (mismo tipo y mismo valor), ==, != (solo mismo valor)

operadores de comparación: ==, !=, <, <=, >, >= (se pueden emplear con cadenas)

operadores aritméticos: +, -, \*, /, %, ++, -- (+ se puede emplear con cadenas)

operadores bit a bit: &, |, ^, ~, <<, >>, >>>

parseInt(str[,radix]); // radix 2..36, si no radix considera: 0x.: hexa, 0.. octal, decimal; a entero, ignora decimales, si no empieza con número NaN, también de real a entero

parseFloat(str); // "abc" return NaN, "1.2abc" return 1.2

eval(expresion); cualquier expresión o grupo de expresiones JS

## Estructuras de Control

```
if (condicion) { ... } [ else { ... } ]
```

```
while (condicion) { ... }
```

```
for (...; condicion; ...) { ... }
```

```
for (variable in objeto) { ... } // recorre las propiedades del objeto (por índice)
```

```
break; continue;
```

```
switch (...) { case ... : .... break; }
```

```
try { ... } catch (e) { ... } // throw .... // excepciones
```

## Objetos (Object)

```
miobjeto= new Object ();    ~ miobjeto= {}  
miarray = new Array ();    ~ miarray = []  
micadena = new String (); ~ micadena = ""  
mifuncion = new Function (); ~ micadena = ""
```

[Number Date Math](#)

```
clase.prototype.<funcion> = function { ... }  
// propiedades (objeto Object) no se incluyen en length  
m["variable"]= m.variable= valor; // da de alta propiedades  
valor= m["variable"]= m.variable; // recupera propiedades  
for (var n in m) writeline (n, m[n]); // n variable string  
delete m.string; elimina la propiedad
```

## Objeto Math

```
abs(n)  
n=sin(a) n=cos(a) n=tan(a) a=acos(n),a=asin(n) a=atan(n) a=atan2(y,x)<-pi..pi> // a en radianes  
ceil(n)<entero mayor o igual> floor(n)<entero menor o igual> round(n)<entero más cercano>  
exp(n)=pow(E,n) log(n)<logaritmo natural base e> // Euler=2.7 1828 1828 45 90 23 53 60 287  
max(n1,n2) min(n1,n2)  
pow(base,exponent) // 1000=pow(10,3)  
sqrt(n) // raíz cuadrada, (n<0)=0  
random() // 0..1 inclusive  
n.toString([radix]) // 16.toString(16) == "10"
```

## Objeto Number

```
Number.MAX_VALUE .MIN_VALUE .NaN .NEGATIVE_INFINITY .POSITIVE_INFINITY  
NaN // no es un número  
isNaN(n) // true|false  
infinity
```

## Objeto Array

```
var m= new Array (40); = []; // m[i] con base 0, número de elementos
```

```

m.join ([ch=,]);           // a cadena separada por ch
m.split ([ch=,]);         // de cadena separada por ch
m.reverse ();             // invierte
m.sort ();                // ordena
m[2]= new Array (20);     // m[i][j]
m3= m1.concat (m2);       // retorna m1+m2, m2 puede ser una lista de valores
m2= m1.slice (n1,n2);     // retorna subarray desde n1 hasta n2 no incluido, si n2<0 final

m.push (e);               // agrega elemento
e= m.pop ();              // elimina último elemento, retorna el elemento eliminado
e= m.shift ();           // elimina primer elemento, retorna el elemento eliminado
length= m.unshift (e1,e2,...); // inserta elementos al inicio de un array
m2= m.splice (posición, número_elementos_a_borrar, [valor_a_agregar1,valor_a_agregar2,...]); // elimina e inserta en medio de una array
[3,4,5]= [1,2,3,4,5,6].splice(2 ,3, 11,12,13,14) > [1,2,11,12,13,14,6]

```

## Objeto String

```

s.length, s["length"]
s.charAt (index) index:0..length-1 return: char | undefined
s.charCodeAt (index) " return: unicode | NaN
s.concat (s2) return: s+s2
String.fromCharCode (code1,code2,...,coden) monta string desde unicodes
s.indexOf (substring [,startindex]) busca substring, starindexcomienzo de búsqueda, return 0..length-1 | -1
s.lastIndexOf (substring [,startindex])
s.match (rgExp) return array           "The rain in SPAIN stays mainly in the plain".match(/ain/gi) > ["ain","AIN","ain","ain"];
s.replace (rgExp, replaceText
s.search (rgexp) return position first substring match | -1           "The rain in SPAIN stays mainly in the plain".search("ain") > 5
s.slice (start [,end]) return section
s.split (str) return array of strings
s.substr (start [,length])
s.substring (start, end) return str the length end-start
s.toLowerCase ()
s.toUpperCase ()
o.toString ([radix])

```

```

function formato (val, num) {
    var str= ""+val;
    while (str.length < num) str= "0"+str;
    return str;
}
function recorta (val) { // quita espacios al principio y al final de val, val admite string y objetos

```

```

var a= ""+val;
while (a.charAt(0)==" ") a= a.substring (1);
while (a.charAt(a.length-1)==" ") a= a.substring (0,a.length-1);
return a;
}

```

## Objeto Date

```

var d = new Date() // current date and time
var d = new Date(milliseconds) //milliseconds since 1970/01/01 00:00:00 (UTC)
var d = new Date(dateString) // October 13, 1975 11:13:00
var d = new Date(year, month, day, hours, minutes, seconds, milliseconds)
local time or UTC (universal, or GMT) time.onds)
d.setDate d.getDate() // día del mes (1-31)
d.getDay() // día de la semana (0-6)
d.getTime() number of milliseconds between midnight, January 1, 1970 and the time stored in the Date object

```

```

function date2fec (date) {
    if (typeof(date)!="date") date= new Date(date);
    return date.getFullYear()*10000+ (date.getMonth()+1)*100+ date.getDate(); // getMonth 0-11
}
function date2hor (date) {
    if (typeof(date)!="date") date= new Date(date);
    returndate.getHours()*+ date.getMinutes()*+ date.getSeconds()+date.getMilliseconds()/;
}
function fec2date (fec) {
    return new Date(fecano (fec) , fecmes (fec) , fecdia (fec));
}

```

## Objeto JSON

object	nombre: { string: value, ... }
array	nombre: [ value, ... ]
value	string   number   object   array   true   false   null
string	"..."   '...' // especiales: \\ \v \b \f \n \r \t \uHHHH
number	[ - ]digit[ .digit][ eE+ -digit]

json.js

```

var myObject = myJSONtext.parseJSON(); // en vez de: var myObject = eval('(' + myJSONtext + ')');
var myJSONText = myObject.toJSONString();

```



# Expresiones Regulares

`\n \\ (\ \)`

`^` comienzo de entrada o línea

`$` final de entrada o línea

`*` carácter anterior 0 o más veces

`+` carácter anterior 1 o más veces

`?` carácter anterior 0 o 1

`.` 1 carácter cualquiera excepto `/n`

`(pattern)` y recuerda `[0]..[n]`

`x|y` x o y

`{n}` exactamente n apariciones del carácter anterior

`{n,m}` como mínimo n y como máximo m apariciones del carácter anterior

`[xyz]` `[a-z]` cualquiera de los caracteres entre corchetes, rangos con guión, `[^...]` ninguno de los caracteres

`\b` límite de palabra      `\B` no      espacio, retorno, ...

`\d` `[0-9]`      `\D` no      dígito

`\s` `[\f\n\r\t\v]`      `\S` no      blanco

`\w` `[A-Za-z0-9_]`      `\W` no      letra o dígito

`\f \n \r \t` salto de página, salto de línea, retorno de carro, tabulador

`\num \n octal, \xn`

## regular expression object:

`var r= /pattern/[switch]`      switch: `i` (ignorecase), `g` (global), `gi`

`var r= new RegExp ("pattern", ["switch"])`

`r = s.replace (/fox/i, "pig");`

`r = s.replace (/(\S+)(\S+)(\S+)/g, "$3$2$1"); // Swap each pair of words`

```
String.prototype.trim = function() { return this.replace(/^\s+|\s+$/ , ''); };
function LTrim( value ) { var re = /\s*((\S+\s*)*)/g; return value.replace(re, "$1"); }
function RTrim( value ) { var re = /((\s*\S+)*)\s*/g; return value.replace(re, "$1"); }
function trim( value ) { return LTrim(RTrim(value)); }
String.prototype.swapcase = function() {
    return this.replace (/([a-z]+)|([A-Z]+)/g,
        function($0,$1,$2) { return ($1) ? $0.toUpperCase() : $0.toLowerCase(); })
}
Hora ^([0-9]|1\d|2[0-3]):([0-5]\d):([0-5]\d)$ (Por ejemplo 10:45:23)
Fecha ^\d{1,2}\.?\d{1,2}\.?\d{2,4}$ (Por ejemplo 01/01/2007)
function regexp_tester(texto,re) {
    var m = re.exec(texto);
    if (m == null) return "No match";
    var s = "Match at position " + m.index + " : "; for (i = 0; i < m.length; i++) s += " ("+(i+1)+")" + m[i];
```





## Window Script Host

[http://msdn.microsoft.com/en-us/library/fdee6589\(v=VS.94\).aspx](http://msdn.microsoft.com/en-us/library/fdee6589(v=VS.94).aspx) Windows Script Interfaces

### WScript Object

is the root object for the Windows Script Host object model

[http://msdn.microsoft.com/en-us/library/xazzc41b\(v=VS.85\).aspx](http://msdn.microsoft.com/en-us/library/xazzc41b(v=VS.85).aspx)

```
wscript /H:CScript // defecto WScript  
wscript /?
```

Properties: Arguments | BuildVersion | FullName (WScript Object) | Interactive | Name | Path | ScriptFullName | ScriptName | StdErr | StdIn | StdOut | Version

Method: ConnectObject | CreateObject | DisconnectObject | Echo | GetObject | Quit | Sleep

[http://msdn.microsoft.com/en-us/library/at5ydy31\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/at5ydy31(v=vs.85).aspx)

```
ActiveXObject ("WScript.Shell");           Environment, SpecialFolders, Shortcut, Popup  
ActiveXObject ("Shell.Application");       Folder, Registro,  
ActiveXObject ("Scripting.FileSystemObject");  
ActiveXObject ("WScript.Network");
```

```
WScript.echo ("Hola Mundo");
```

```
WScript.ScriptFullName;
```

```
WScript.quit (e?-1:0);
```

---

para C++

[http://msdn.microsoft.com/en-us/library/t9d4xf28\(v=VS.94\).aspx](http://msdn.microsoft.com/en-us/library/t9d4xf28(v=VS.94).aspx)

## ActiveX WScript.Shell

Permite: buscar carpetas especiales, crear accesos directos, ejecutar aplicaciones, manipular el registro, mostrar ventanas de confirmación, ...

```
var wss= new ActiveXObject ("WScript.Shell"); // WScript.CreateObject
```

Environment

```
var wsse= wss.Environment(["Process"]); // System, User, Volatile, Process
```

```
wsse("PATH"); // "WINDIR","TMP",...
```

```
wsse.Remove(...)
```

Special Folders

```
var wssf= wss.SpecialFolders;
```

```
wssf("Desktop"); // "StarMenu","Favorites"
```

### Shortcut

```
var sc = wss.CreateShortcut (<camino.Ink>); // WshShortcut
```

```
sc.TargetPath = <path o url>;
```

```
sc.Save
```

```
sc.WorkingDirectory, sc.IconLocation, sc.Description, sc.WindowStyle, sc.HotKey, sc.Arguments
```

### Popup

```
intButton = wss.Popup (strText, [natSecondsToWait], [strTitle], [natType]) = MessageBox
```

botones: (0)Aceptar (1)AceptarCancelar (2)AnularReintentarOmitir (3)SiNoCancelar (4)SiNo (5)ReintentarCancelar

iconos: (16)error (32)pregunta (48)exclamación (64)información

retorno: (-1)time (1)OK (2)Cancel (3)Abort (4)Retry (5)Ignore (6)yes (7)No.

## ActiveX Shell.Application

Permite: cerrar windows, abrir explorador, buscar archivos, ...

```
var shell= new ActiveXObject ("Shell.Application");
shell.SetTime (); // abrimos el reloj
var archivo= shell.BrowseForFolder (0, rótulo, &H4000); // Folder H4000:muestra también archivos
archivo.Title; // nombre con extensión sin camino
camino= archivo.ParentFolder.ParseName(archivo.Title).Path; // camino completo
```

### Folder

MoveHere CopyHere  
NewFolder  
Items  
Run

### Crear un icono en el escritorio

```
var DesktopPath = shell.SpecialFolders ('Desktop')
var link = shell.CreateShortcut (DesktopPath+'\test.lnk')
link.Arguments = '1 2 3'
link.Description = 'test shortcut'
link.HotKey = 'CTRL+ALT+SHIFT+X'
link.IconLocation = 'app.exe,1'
link.TargetPath = 'c:\blah\app.exe'
link.WindowStyle = 3
link.WorkingDirectory = 'c:\blah'
link.Save()
```

### Registro

RegRead, RegWrite, RegDelete

Abreviaturas aceptadas:

HKEY\_CURRENT\_USER > HKCU

HKEY\_LOCAL\_MACHINE > HKLM

HKEY\_CLASSES\_ROOT > HKCR

### Crear una entrada en el registro de windows

```
shell.RegWrite ('HKCU\WSHTest\testkeydefault', 'testkeydefault')
shell.RegWrite ('HKCU\WSHTest\string1', 'testkeystring1')
shell.RegWrite ('HKCU\WSHTest\string2', 'testkeystring2', 'REG_SZ')
shell.RegWrite ('HKCU\WSHTest\string3', 'testkeystring3', 'REG_EXPAND_SZ')
shell.RegWrite ('HKCU\WSHTest\int', 123, 'REG_DWORD')
```

## Eliminar una entrada

```
shell.RegDelete ('HKCU\\Software\\ACME\\FortuneTeller\\MindReader')
```

```
shell.RegDelete ('HKCU\\Software\\ACME\\FortuneTeller\\')
```

```
shell.RegDelete ('HKCU\\Software\\ACME\\')
```

## ActiveX Scripting.FileSystemObject

[http://msdn.microsoft.com/en-us/library/6kxy1a51\(v=VS.85\).aspx](http://msdn.microsoft.com/en-us/library/6kxy1a51(v=VS.85).aspx)

contained in the scripting type library (scrrun.dll). suport text file

### FileSystemObject

```
var fso= new ActiveXObject ("Scripting.FileSystemObject"); // CreateObject en VBS
fso.DeleteFolder (folder [,force]); // f:path,mask(error si ninguno) force:dafault:true force delete read-only, borra incluso con contenido
fso.MoveFolder (source, destination); // s:path,mask(error si ninguno) d:path(acabado en \ si padre)(error si existe)
fso.CopyFolder (source, destination[, overwrite]); // recursivo s:path,mask(error si ninguno) d:path(acabado en \ si padre) o:default:true
fso.GetDriveName (path);
fso.GetParentFolderName (path);
fso.GetFileName (path);
fso.GetBaseName (path); // sin extensión
fso.GetExtensionName (path);
fso.GetAbsolutePathName (path); // completa el camino con el directorio actual y expandiendo "." y ".."
fso.GetTempName ();
var fo= fso.CreateFolder("c:\\test");
var fo= fso.GetFolder ("c:");
var fo= fso.GetSpecialFolder (que); // (0)Windows (1)System (2)Temporary
ts= fso.GetSpecialFolder(2).CreateTextFile (fso.GetTempName());
fso.MoveFile (source,destination); // s: path, file or mask(error si ninguno), d:path(terminado en\),file(error si existe)
fso.CopyFile (source, destination[, ¿no? overwrite]); // recursivo s:file,mask(error si ninguno) d:path(acabado en \ si padre) o:default:true permite comodines en
varios niveles
fso.DeleteFile (file[,force]); // f:file,mask f:true delete read only
```

### Drive

```
var dr= fso.GetDrive (fso.GetDriveName(camino));
TotalSize (bytes)
AvailableSpace = FreeSpace (bytes)
DriveLetter. DriveType, SerialNumber, FileSystem (FAT,FAT32,NTFS), IsReady, ShareName(type3), VolumeName(itype3)
Path, RootFolder
DriveType: (1)Removable (2)Fixed (3)Network (4)CDROM (5)RAMDisk
Drives
fso.Drives
```

### Folder

```
var f = fso.GetFolder("c:\\tmp");
var d= f.DateCreated;
f.Copy, f.Delete, f.Move, f.CreateTextFile
MoveFolder,FolderExists,
Drive,IsRootFolder
Folders
```

## File

```
var f= fso.GetFile ("c:\\test.txt");  
f.Move ("c:\\file.txt"); f.Copy ("c:\\file.txt"); f.Delete (); OpenAsTextStream([iomode][,format])  
Name, Path, Size, ShortName, ShortPath, DateCreated,DateLastModified,DateLastAcceded, ParentFolder, Drive, Size(bytes), Type("Text Document", ...)  
Attributes: (0)Normal (1)ReadOnly (2)Hidden (4)System (8)Volume (16)Directory (32)Archive (64)Alias (128)Compressed  
Files
```

## TextStream

```
var ForReading=1; ForWriting=2; ForAppending=8 // iomode  
var FormatDefault=-2; FormatUnicode=-1; FormatAscii=0; // format  
var ts= fso.CreateTextFile ("c:\\test.txt", true); // empty text file  
var ts= fso.OpenTextFile ("c:\\test.txt", ForWriting, true);  
var ts= fi.OpenAsTextStream (ForWriting, true);  
Write ("..."); WriteLine ("..."); WriteBlankLines (n);  
Read (n); ReadLine (); ReadAll ();  
ts.Skip (n°caracteres), ts.SkipLine(); // Left, Right, Mid  
bool ts.AtEndOfLine, bool ts.AtEndOfStream, enbase1 ts.Column, enbase1 ts.Line  
Close  
while (its.AtEndOfStream) { var s= ts.ReadLine (); }
```

## Dictionary

Add, Exists, Items, keys, Remove, RemoveAll, Count, Item, Key

## Enumerator

```
var en= new Enumerator (fso.Drives)  
for (; ien.atEnd(); en.moveNext()) { var x= en.item(); }  
var fc = new Enumerator(f.folders); if (fc<=0) return n;  
for (; !fc.atEnd(); fc.moveNext())
```

## ActiveX WScript.Network

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/438d1705-fb98-4a63-86e5-a8cc01f4ce16.asp>

```
var wsn = new ActiveXObject ("WScript.Network");  
wsn.UserName  
wsn.UserDomain  
wsn.ComputerName  
wsn.MapNetworkDrive ("S:", "\\server\share");  
wsn.RemoveNetworkDrive('E')  
wsn.SetDefaultPrinter "HP LaserJet 5"
```

```
function InGroup (Domain, Group, User)  
{  
    obj = GetObject("WinNT://" + Domain + "/" + Group);  
    return obj.IsMember("WinNT://" + Domain + "/" + User);  
}
```

### OBTENER INFORMACION SOBRE LAS UNIDADES DE CD

```
var strComputer='.', a= '';  
objWMIService = Getobject ('winmgmts:'+ 'impersonationLevel=impersonate')!\\ + strComputer + '\root\cimv2')  
colItems = objWMIService.ExecQuery ('Select * from Win32_CDROMDrive')
```

```
for (objItem In colItems)  
{  
    a += 'Description: ' + objItem.Description + "\n"  
    a += 'Name: ' + objItem.Name + "\n"  
    a += 'Manufacturer: ' + objItem.manufacturer + "\n"  
    a += 'Media type: ' + objItem.mediaType + "\n"  
    a += 'PNP Device ID: ' + objItem.PNPDeviceID + "\n"  
}  
alert(a)
```



# Teoría de objetos

[http://msdn.microsoft.com/en-us/library/d1et7k7c\(v=VS.94\).aspx](http://msdn.microsoft.com/en-us/library/d1et7k7c(v=VS.94).aspx) JavaScript for IE 9 ECMAScript-5 (ES5)

<http://web.ontuts.com/tutoriales/namespacing-en-javascript/>

## Asignación de métodos por prefijo

```
var myApp= {}  
myApp.num= 14;  
myApp.sum= function(a,b) { ... } // no utilizar this con objetos utilizados como namespace (contexto)
```

## Notacion literal

```
var myApp= {  
    num: 14,  
    sum: function(a,b) { ... }  
}
```

## Módulos

```
var myApp= (function(){  
    var num= 14; // privados  
    var sum= function(a,b) { ... }  
    return {  
        num2:num; // públicos  
        sum2:function() { return num+num; }  
    }  
})();
```

## Contexto dinámico (jQuery)

```
var myApp= {}  
(function(app){  
    var num= 14; // privados  
    var sum= function(a,b) { ... }  
    app.num2= num; // públicos  
    app.sum2= function() { return num+num; }  
})(myApp);
```

## Contexto proxy

```
var ClaseA= function (num1) {  
    var num= num1 || 0;  
    this.incrementa = function() { return num++; }  
}  
var A1={}; ClaseA.call (A1);
```



## JavaScript Jargon

<http://www.codeproject.com/KB/scripting/javascript-jargon.aspx>

```
var func= function (uni,dos) { ... } // función anónima asignada a una variable, ventaja única, la brevedad
function func (uni,dos) { ... } // window.func
func ('one', 'two');
window.func ('one', 'two');
```

```
func.apply (objeto, [ 'one', 'two' ] );
func.call (objeto, 'one', 'two' );
```

### Closure

ámbito funcional (conjunto de variables) que se mantiene viva después del retorno de la función

```
function externa() {
    var a="Mundo";
    return function interna () { alert ("Hola "+a); };
}
var f= externa(); f();
function coche(marca,modelo) {
    var n=0; // variable privada
    function tieneBateria() { return true; } // función privada
    return { marca:marca, modelo:modelo, inicia: function() { if (tieneBateria()) n++; } };
}
var miCoche= coche("Honda","Civic"); miCoche.inicia();
```

pega: se crean las funciones con cada instancia (no se reusan)

### Currying

consiste en envolver el attr función de una nueva función con menos argumentos, y nos permite fijar los valores de dos de los argumentos

```
función attr (clave, valor) { ... } // Establece un atributo de un elemento
$("#bob"). attr ("titulo","es el título");

función getAttrFn (elem, clave) { return function (valor) { elem.attr (clave, valor); }; }
var setTitle= getAttrFn ($("#bob"),"title"); setTitle("es el título");
```

### Hoisting

elevación, las variables defcnidas (var) en una función se elevan al comienzo de la función

```
var a=1;
function hola() {
    alert(a); // indefinido, no 1
    var a=2;
```

```
    alert(a); // 2  
}
```

las expresiones de función igual que variable (se eleva la declaración no la asignación)  
las declaraciones de función se elevan coompletas

### función auto-ejecutable

```
( function(p) { ... } (v) );
```

### this

objeto que invoca la función , no el propietario (this == window)  
funcion.call(obj) // obj > this

### prototype

```
var Vehicle= function() { ... };  
Vehicle.prototype= { iniciado:false, iniciar:function() { ...} };  
var v= new Vehicle(); // new crea un objeto nuevo y vacio, el objeto apunta al prototipo, el this de la función apunta al objeto  
var Coche= function() { ... };  
Coche.prototype= new Vehicle();  
Coche.prototype.abreTecho= function() {}  
var v= new Coche();
```

## Depuración Javascript

```
HKEY_CURRENT_USER  
Software\Microsoft\Windows Script\Settings\JITDebug
```

Luego en el código javascript poner la palabra reservadadebugger.