

JScript

---

## **JScript**

[Jscript](#)

[Internet JScript](#)

[JScript](#)

[FlieSystemObject](#)

[Script](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

# **JScript**

[JScript](#)

[JScript](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

# **JScript**

[Jscript](#)

[JScript](#)

[JScript](#)

[JScript](#)

[JScript](#)

[JScript](#)

[JScript](#)

[JScript](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **JScript**

JScript Microsoft ECMA 262 ECMAScript 3  
JScript ECMA JScript

## **JScript**

JScript C++ JScript JScript

JScript Java JScript “”  
Active Server Pages ASP Internet Windows

JScript JScript JScriot JScript

JScript

Web

---

[© 2000 Microsoft Corporation](#)

JScript

---

# **JScript**

Microsoft JScript

“”

JScript Jscript Jscript

“;” (); JScript  
Jscript

aBird = "Robin"; //“ Robin” aBird  
var today = new Date(); // today

{} JScript JScript                           **for**   **while**

Jscript C++

```
function convert(inches) {  
    feet = inches / 12; //  
    miles = feet / 5280;  
    nauticalMiles = feet / 6080;  
    cm = inches * 2.54;  
    meters = inches / 39.37;  
}  
km = meters / 1000; //  
kradius = km;  
mradius = miles;
```

```
JScript ()  
aGoodIdea = "Comment your code thoroughly."; //  
(/*), */  
/*  
  
aGoodIdea  
  
*/  
JScript */ JScript  
  
//  
// aGoodIdea  
//  
// aGoodIdea  
  
var extendedIdea = aGoodIdea + " You never know when
```

JScript (=) = LvalueLvalue

- 
- 
- 

= RvalueRvalues JScript

anInteger = 3;

Jscript “ 3 anInteger”“anInteger 3”

= == (==)

JScript JScript JScript “” JScript —  
JScript

```
3.9          //
"Hello!"      //
false         //
null          //
{x:1, y:2}    //
[1,2,3]       //
function(x){return x*x;} //
```

```
+ //
- //
* //
/ //
```

JScript

```
var anExpression = 3 * (4 / 5) + 6;
var aSecondExpression = Math.PI * radius * radius;
var aThirdExpression = aSecondExpression + "%" + anEx
var aFourthExpression = "(" + aSecondExpression + ")" %
```

JScript

---

## **JScript**

How old am I?

Jscript

```
NumberOfDaysLeft = EndDate - TodaysDate;
```

**var**

```
var count; //  
var count, amount, level; //      var  
var count = 0, amount = 100; //  
  
var JScript      undefined      var JScript JScript  
          .           .           .  
          var
```

Jscript

- 
- 
- 

JScript        *myCounter*    *mYCounter*

- ASCII(\_)
- 
- .

\_pagecount

Part9

Number\_Items

99Balloons //

Smith&Wesson // ""       &

JScript        null

var bestAge = null;  
var muchTooOld = 3 \* bestAge; // muchTooOld       0

Jscript        undefined

var currentCount;  
var finalCount = 1 \* currentCount; // finalCount      NaN

JScript      **null**    **undefined**    **null**    0      **undefined** **NaN**  
**null**    **undefined**

**var**

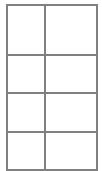
noStringAtAll = ""; //                    noStringAtAll

var volume = length \* width; //       — length width

JScript Jscript

JScript C++JScript

Jscript JScript JScript



```
var x = 2000;    //
var y = "Hello"; //
x = x + y;      //
document.write(x); // 2000Hello
```

[parseInt](#)      [parseFloat](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **JScript**

### Jscript

- 
- 
- 
- 
- 
- Null
- Undefined

Unicode JScript

"Happy am I; from care I'm free!"

"Avast, ye lubbers!" roared the technician.'

"42"

'C'

JScript C++

**char** Jscript ""

## Jscript JScript JS

0 10 8 16 Jscript “0” 8 0 7  
“0”“8”“9”

“0x”x|X 16 0 9 A F A F 10 15  
0xF 15 0x10 16

“e” 10 Jscript IEEE754  
 $\pm 1.7976931348623157 \times 10^{308} \pm 5 \times 10^{-324}$ “0”

“0x”“00” Jscript

.0001, 0.0001, 1e-4, 1.0e-4		0.0001
3.45e2		345
42		42
0378	08	378
0377	1	255
0.0001		0.0001
00.0001		N/A
0Xff		255
0x37CF		14287
0x3e7	‘e’	999
0x3.45e2		N/A

## JScript

- NaN
- JScript

- JScript
- 00Jscript00

## Boolean

boolean      **true**    **false** Boolean

Boolean Jscript

y = (x == 2000);

x 2000 Boolean      **true** yx2000boolean      **false**

Boolean boolean boolean JScript

```
if (x == 2000)
    z = z + 1;
else
    x = x + 1;
```

boolean      **true** Jscript      **if/else** z = z + 1 boolean  
**false** x = x + 1

0null      **false**    **true**

if (x = y + z) //      —

x y+z y+z x x y+z

if (x == y + z) //

Null

Jscript null null null

null “”“”boolean null

Jscript null 0 C C++ Jscript  
**Object** null

## Undefined

undefined

- 
- 

undefined “undefined” x

```
//  
if (x == undefined)  
  //
```

```
//           -  
//   "undefined"  
if (typeof(x) == undefined)  
  //
```

```
//  
if (typeof(x) == "undefined")  
  //
```

undefined null

someObject.prop == null;

**true**

- someObject.prop null
- someObject.prop
- **in**

if ("prop" in someObject)

```
// someObject      'prop'
```

---

[© 2000 Microsoft Corporation](#)

JScript

---

## JScript

JScript ,

-	!	~	=	delete
++	<	<<	<i>oP=</i>	<a href="#">typeof</a>
--	>	>>		<a href="#">void</a>
*	<=	>>>		<a href="#">instanceof</a>
/	>=	&	<a href="#">new</a>	new
%	==	^	<a href="#">in</a>	in
+	!=			
-	&&			
	<a href="#">?:</a>			
	,			
	====			
	!==			

== === "1" 1 true "1" 1

**ArrayFunctionStringNumberBooleanErrorDate**  
**RegExp** true

```
//  
var string1 = "Hello";  
var string2 = "Hello";  
  
//      String  
var StringObject1 = new String(string1);  
var StringObject2 = new String(string2);  
  
//      true  
if (string1 == string2)
```

```
//  
//      false  
if (StringObject1 == StringObject2)  
//  
//      String  
//  toString()  valueOf()  
if (StringObject1.valueOf() == StringObject2)  
//
```

---

[© 2000 Microsoft Corporation](#)

JScript

---

Jscript

“”“” Boolean

**true    false**

Jscript

- **if**
- **if/else**
- **?:**
- **switch**

Jscript

- **while**
- **do/while**
- **for/in**
- **for**

```
JScript      if  if...else      if JScript          if...else
           if      if   if...else

if  if...else          true    if

// smash()
//           newShip      true
if (newShip)
  smash(champagneBottle,bow);

//
if (rind.color == "deep yellow " && rind.texture == "large")
{
  theResponse = ("Is it a Crenshaw melon?");
}

//
var theReaction = "";
if ((dayOfWeek == "Saturday") || (dayOfWeek == "Sunday"))
{
  theReaction = ("I'm off to the beach!");
}
else
{
  theReaction = ("Hi ho, hi ho, it's off to work I go!");
}
```

```
JScript           if
var hours = "";
//           hours      theHour
//           theHour - 12
hours += (theHour >= 12) ? " PM" : " AM";
"" JScript
"" ((x == 123) && (y == 42))Jscript x 123 y
42 true y Jscript
false
|| OR
false           true AND
runfirst() 0   false   runsecond()
if ((runfirst() == 0) || (runsecond() == 0)) {
    //
}
```

JScript

[for](#)

[for...in](#)

[while](#)

[do...while](#)

```
for
for

/*
    "icount++"
*/

var howFar = 10; //          10
var sum = new Array(howFar); //      sum      10
var theSum = 0;
sum[0] = 0;

for(var icount = 0; icount < howFar; icount++) {      //
theSum += icount;
sum[icount] = theSum;
}

var newSum = 0;
for(var icount = 0; icount > howFar; icount++) {      //
newSum += icount;
}

var sum = 0;
for(var icount = 0; icount >= 0; icount++) {      //
sum += icount;
}
```

## **for..in**

JScript . **for..in**

Internet **alert** Jscript

//

```
var myObject = new Object();
myObject.name = "James";
myObject.age = "22";
myObject.phone = "555 1234";
```

//

```
for (prop in myObject)
{
```

// "The property 'name' is James"

window.alert("The property " + prop + " is " + myObj

```
}
```

**for..in** VBScript  
VBScript  
Internet VBScript

**For Each...Next** JScript  
**For Each...Next** JScript  
**For Each...Next** Jscript

**for..in** Jscript  
**Enumerator**  
**for..in**

## **while**

**while for**      **while " n "**      **while**  
Internet                  while

```
var x = 0;  
while ((x != 42) && (x != null))  
{  
    x = window.prompt("What is my favourite number?", x)  
}
```

```
if (x == null)  
    window.alert("You gave up!");  
else  
    window.alert("Yep - it's the Ultimate Answer!");
```

**while**      **while**      **while**

JScript      **do...while** while

```
var x = 0;  
do  
{  
    x = window.prompt("What is my favourite number?", x)  
} while ((x != 42) && (x != null));
```

```
if (x == null)  
    window.alert("You gave up!");  
else  
    window.alert("Yep - it's the Ultimate Answer!");
```

## **break continue**

Microsoft Jscript              break  
**for**      **for...in**              continue

**break**    **switch**

**break**    **continue**

```
var x = 0;
do
{
    x = window.prompt("What is my favourite number?", x

    //
    if (x == null)
        break;

    //
    //
    if (Number(x) == x)
        continue;

    //
    window.alert("Please only enter in numbers!");

} while (x != 42)

if (x == null)
    window.alert("You gave up!");
else
    window.alert("Yep - it's the Ultimate Answer!");
```

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **JScript**

Microsoft Jscript “”

Jscript

Jscript                    [eval\(\)](#) Jscript            **eval()**

```
var anExpression = "6 * 9 % 7";
var total = eval(anExpression); //      total      5
var yetAnotherExpression = "6 * (9 % 7)";
total = eval(yetAnotherExpression) //      total      12
//      totality
var totality = eval(...surrounded by acres of clams.');
```

Jscript

```
Checktriplet checkTriplet  
“epsilon”  
  
var epsilon = 0.0000000001; //  
  
//  
function integerCheck(a, b, c)  
{  
    //  
    if ( (a*a) == ((b*b) + (c*c)) )  
        return true;  
  
    return false;  
} //  
  
//  
function floatCheck(a, b, c)  
{  
    //  
    var delta = ((a*a) - ((b*b) + (c*c)))  
  
    //  
    delta = Math.abs(delta);  
  
    //      epsilon
```

```
if (delta < epsilon)
    return true;

return false;
} //



//  

function checkTriplet(a, b, c)
{
    //  

    var d = 0;  

    // "           a"  

    //           a  b  

    if (b > a)
    {
        d = a;
        a = b;
        b = d;
    }  

    //           a  c  

    if (c > a)
    {
        d = a;
        a = c;
        c = d;
```

```
}

//      3
if (((a % 1) == 0) && ((b % 1) == 0) && ((c % 1) == 0)
{
    //
    return integerCheck(a, b, c);
}
else
{
    //
    return floatCheck(a, b, c);
}
} //



//      'result'
var result = checkTriplet(sideA, sideB, sideC);
```

---

JScript

---

# JScript

Jscript Jscript  
**document** ActiveX

. Internet

**window**

Jscript “”  
"hello" — **length** 101 **length** **myArray[100]** =

Jscript “expando”<>>

var myObj = new Object();

// expando 'name' 'age'  
myObj.name = "Fred";  
myObj.age = 42;

Jscript expando

var myObj = new Object();

// object.property  
// expando  
//  
//

myObj["not a valid identifier"] = "This is the property value";

// expando

//

myObj[100] = "100";

0 **length** expando expando

**length**

//

```
var myArray = new Array(3);

//          3
myArray[0] = "Hello";
myArray[1] = 42;
myArray[2] = new Date(2000, 1, 1);

//          3
window.alert(myArray.length);

//      expando
myArray.expando = "JScript!";
myArray["another Expando"] = "Windows";

//          3      expando
//
window.alert(myArray.length);

Jscript — 5

//          0
var iMaxNum = 5;
//
var i, j;

//          1          iMaxNum + 1
var MultiplicationTable = new Array(iMaxNum + 1);

//
```

```
for (i = 1; i <= iMaxNum; i++)
{
    //
    MultiplicationTable[i] = new Array(iMaxNum + 1);

    //
    for (j = 1; j <= iMaxNum; j++)
    {
        MultiplicationTable[i][j] = i * j;
    }
}

window.alert(MultiplicationTable[3][4]); //      12
window.alert(MultiplicationTable[5][2]); //      10
window.alert(MultiplicationTable[1][4]); //      4
```

JScript

---

```
pasta          this  
// pasta  
function pasta(grain, width, shape, hasEgg)  
{  
    //  
    this.grain = grain;  
  
    //  
    this.width = width;  
  
    //  
    this.shape = shape;  
  
    //      boolean  
    this.hasEgg = hasEgg;  
}
```

```
var spaghetti = new pasta("wheat", 0.2, "circle", true);
var linguine = new pasta("wheat", 0.3, "oval", true);
```

```
// spaghetti  
spaghetti.color = "pale straw";  
spaghetti.drycook = 7;  
spaghetti.freshcook = 0.5;
```

```
var chowFun = new pasta("rice", 3, "flat", false);
// chowFun          pasta
//           spaghetti
//
// '    foodgroup'    pasta
//      pasta
//
pasta.prototype.foodgroup = "carbohydrates"
//
//     spaghetti.foodgroupchowFun.foodgroup
// “      carbohydrates”
```

```
pasta          toString  
  
// pasta  
//  
function pasta(grain, width, shape, hasEgg)  
{  
    //  
    this.grain = grain;  
  
    //  
    this.width = width;  
  
    //  
    this.shape = shape;  
  
    //      boolean  
    this.hasEgg = hasEgg;  
  
    //      toString  
    //  
    //  
    //  
    this.toString = pastaToString;  
}  
  
//      past  
function pastaToString()  
{
```

```
//  
  
return "Grain: " + this.grain + "\n" +  
    "Width: " + this.width + "\n" +  
    "Shape: " + this.shape + "\n" +  
    "Egg?: " + Boolean(this.hasEgg);  
}  
  
var spaghetti = new pasta("wheat", 0.2, "circle", true);  
//      toString()      spaghetti  
//      Internet  
window.alert(spaghetti);
```

---

JScript

---

Microsoft Jscript 11 “”

**Array Boolean Date Function Global Math Number Object RegExp Error String**

## Array

**new** **Array()**

```
var theMonths = new Array(12);
theMonths[0] = "Jan";
theMonths[1] = "Feb";
theMonths[2] = "Mar";
theMonths[3] = "Apr";
theMonths[4] = "May";
theMonths[5] = "Jun";
theMonths[6] = "Jul";
theMonths[7] = "Aug";
theMonths[8] = "Sep";
theMonths[9] = "Oct";
theMonths[10] = "Nov";
theMonths[11] = "Dec";
```

**Array** Jscript      **length** 0  
length

```
var theMonths = new Array("Jan", "Feb", "Mar", "Apr", "
"Jul", "Aug", "Sep", "Oct", "Nov", "Dec");
```

**Array** Jscript      **length** Jscript 0 1 length  
1

## **String**

Jscript                    [string](#)                    [substring](#)

```
aString = "0123456789";
var aChunk = aString.substring(4, 7); // aChunk      "456"
var aNotherChunk = aString.substring(7, 4); // aNotherC
//
firstLetter = theMonths[5].substring(0,1); //      firstLetter
```

**String length 0**

```
var howLong = "Hello World".length //      howLong 10
```

## **Math**

**Math** pi 3.14159...

**Math.PI**

//

var circleArea = Math.PI \* radius \* radius; // Math PI

**Math** pow pi

//

volume = (4/3)\*(Math.PI\*Math.pow(radius,3));

## Date

**Date**                   **Date** 1970 1 1  
00:00:00.000 GMT GMT UTC““““  
Jscript 250,000 B.C. 255,000 A.D

**new Date**

```
/*
```

```
“ Day Month Date 00:00:00 Year”
```

```
thisIsToday
```

```
*/
```

```
var thisIsToday = new Date();
```

```
var toDay = new Date(); //
```

```
//
```

```
var thisYear = toDay.getFullYear();
```

```
var thisMonth = theMonths[toDay.getMonth()];
```

```
var thisDay = thisMonth + " " + toDay.getDate() + ", " + 1
```

## Number

**Math**

**PI Microsoft Jscript**

**Number**

MAX_VALUE	1.79E+308
MIN_VALUE	2.22E-308
NaN	""
POSITIVE_INFINITY	Number.MAX_VALUE
NEGATIVE_INFINITY	Number.MAX_VALUE

**Number.NaN**    ""    0

NaN    Number.NaN    NaN

**Number.NaN**    **NaN**

**isNaN()**

---

[© 2000 Microsoft Corporation](#)

JScript

---

# Jscript

Jscript Jscript

Jscript Jscript

break	delete	function	return	typeof
case	do	if	switch	var
catch	else	in	this	void
continue	false	instanceof	throw	while
debugger	finally	new	true	with
default	for	null	try	

abstract	double	goto	native	static
boolean	enum	implements	package	super
byte	export	import	private	synchronized
char	extends	int	protected	throws
class	final	interface	public	transient
const	float	long	short	volatile

Jscript            **String**    **parseInt**

JScript

---

# **JScript**

---

[© 2000 Microsoft Corporation](#)

JScript

---

**new**

```
var myObject = new Object();          //  
var myBirthday = new Date(1961, 5, 10); //      Date  
var myCar = new Car();              //
```

**this**

```
new Object() Date() Function() this

function Circle (xPoint, yPoint, radius) {
    this.x = xPoint; //      x
    this.y = yPoint; //      y
    this.r = radius; //
}

Circle Circle Circle

var aCircle = new Circle(5, 11, 99);
```

```

Circle          this
Circle.prototype.pi = Math.PI;
function ACirclesArea () {
    return this.pi * this.r * this.r; // ?r2
}
Circle.prototype.area = ACirclesArea; // C
var a = ACircle.area();           // Circle

VBScript        Trim      String
//           trim
// String
String.prototype.trim = function()
{
    //
    //
    return this.replace(/(^s*)|(\s*$)/g, "");
}

//
var s = " leading and trailing spaces ";
//   " leading and trailing spaces (35)"
window.alert(s + " (" + s.length + ")");

//
s = s.trim();

```

```
//  "leading and trailing spaces (27)"  
window.alert(s + " (" + s.length + ")");
```

---

[© 2000 Microsoft Corporation](#)

JScript

---

0 1 1 \* 2 \* ... 1

“ 0 1 0”

0

—

“” -24 -25  
-26

JScript

```
//  
//  
//      -1  
//  
//  
function factorial(aNumber) {  
    aNumber = Math.floor(aNumber); //  
    if (aNumber < 0) { //      0  
        return -1;  
    }  
    if (aNumber == 0) { //      0      1  
        return 1;  
    }  
    else return (aNumber * factorial(aNumber - 1)); //
```

}

---

[© 2000 Microsoft Corporation](#)

JScript

---

JScript

C++ “”“{}”JScript

var aCentaur = "a horse with rider,"; // aCentaur

// JScript

function antiquities() // aCentaur  
{

// JScript

var aCentaur = "A centaur is probably a mounted Scythian

// JScript

aCentaur += ", misreported; that is, "; //

// JScript

} //

var nothinginparticular = antiquities();

aCentaur += " as seen from a distance by a naive innocent

/\*

          "A centaur is probably a mounted Scythian  
misreported; that is, "

"a horse with rider, as seen from a distance by a naive innocent

\*/

```
tweak();
var aNumber = 100;
function tweak() {
    var newThing = 0; //      newThing
                      //      newThing      aNumber
    newThing = aNumber;

    //      42      aNumber aNumber = 42;
    if (false) {
        var aNumber; //
        aNumber = 123; //
    } //

} //
```

JScript

var someVariable;

var someVariable = "something";

JScript JScript — var —

JScript

---

JScript

Numbers Boolean ( **true** **false**)

""

**Array** **toString()**

**String** **new String("something")**

[ASCII](#) ANSI "Zoo" "aardvark"

**toUpperCase()** **toLowerCase()**

IE

```
//  
//  
function Clobber(param)  
{  
    //  
    //  
    param = new Object();  
    param.message = "This will not work";  
}  
  
//  
//  
function Update(param)  
{  
    //  
    //  
    param.message = "I was changed";  
}  
  
//  
var obj = new Object();  
obj.message = "This is the original";  
  
//      Clobber      obj.message
```

```
Clobber(obj);
window.alert(obj.message); //      "This is the original"

// Update obj.message
Update(obj);
window.alert(obj.message); //      "I was changed"
```

---

[© 2000 Microsoft Corporation](#)

JScript

---

JScript 01 2 50 3 49 length  
length 51 4

JScript

“[]”        *entryNum*

```
theListing = addressBook[entryNum];
theFirstLine = theListing[1];
```

“.”

myObject.aProperty

“[]”

myObject["aProperty"] //

“.”		
“[]”		

**for ... in**

---

[© 2000 Microsoft Corporation](#)

JScript

---

## JScript JScript

\b	
\f	
\n	
\r	
\t	(Ctrl-I)
\'	
\"	
\\\	

(\\)

```
document.write('The image path is C:\\webstuff\\mypage\\'
document.write('The caption reads, "After the snow of '9'
```

JScript

---

[null](#) JScript C C++

**Null**

JScript

JScript Web HTML <HEAD> <BODY>  
<BODY> <HEAD>

IEASP WSH

JScript                   **true**

```
"100" == 100;  
false == 0;
```

""== false

```
"100" === 100;  
false === 0;
```

theRadius = aPerimeterPoint - theCenterpoint \* theCorrec

**for..in**

[for..in](#)

**with**

with

**this**

this

**this**

**this**

**IE**

</SCRIPT>"</SCRIPT>" "</SCR" "IPT>"

**IE**

---

[© 2000 Microsoft Corporation](#)

JScript

---

Jscript

**@cc\_on**      **@if**    **@set** Jscript

Netscape Navigator

```
/*@cc_on @*/
/*@if (@_jscript_version >= 4)
alert("JScript version 4 or better");
@else @*/
alert("You need a more recent script engine.");
/*@end @*/
```

**@cc\_on**

---

JScript

---

**true   NaN**

@_win32	Win32 true
@_win16	Win16 true
@_mac	Apple Macintosh true
@_alpha	DEC Alpha true
@_x86	Intel true
@_mc680x0	Motorola 680x0 true
@_PowerPC	Motorola PowerPC true
@_jscript	true
@_jscript_build	Jscript
@_jscript_version	majorminor Jscript

---

[© 2000 Microsoft Corporation](#)

JScript

---

Microsoft JScript    **write( )**    **writeln( )document**

—

## **document.write( ) document.writeln( )**

document      **write( )** HTML

```
document.write("Pi is approximately equal to " + Math.PI  
document.write( );
```

```
"document.write" "w();"
```

```
function w(m) { //  
m = "" + m + ""; //      m  
if ("undefined" != m) { //  
    document.write(m);  
}  
document.write("<br>");  
}
```

```
w('<IMG SRC="horse.gif">');  
w();  
w("This is an engraving of a horse.");  
w();
```

**writeln( ) write( )** HTML <PRE> <XMP>

**write( )**      **writeln( )**

```
<HTML>  
<HEAD>  
<SCRIPT LANGUAGE="JScript">
```

```
function singOut() {  
var theMoment = new Date();  
var theHour = theMoment.getHours();  
var theMinute = theMoment.getMinutes();  
var theDisplacement = (theMoment.getTimezoneOffset()) /  
theHour -= theDisplacement;  
if (theHour > 23) {  
theHour -= 24  
}  
document.write(theHour + " hours, " + theMinute + " min");  
window.setTimeout("singOut()", 60000);  
}  
</SCRIPT>  
</HEAD>  
<BODY>  
<SCRIPT>  
singOut();  
</SCRIPT>  
</BODY>  
</HTML>
```

window     **alert()**   **document.write()**

```
window.alert(theHour + " hours, " + theMinute + " minute");  
window.setTimeout("singOut()", 60000);  
}
```

**document clear()**

document.clear();

---

[© 2000 Microsoft Corporation](#)

JScript

---

**window**      **window** "window.alert()"

```
alert HTML ""
```

```
window.alert("");
```

“\_”“”“”

**confirm    true    false**

```
var truthBeTold = window.confirm("“”“”"
if (truthBeTold) {
window.alert("           Web      ");
} else window.alert("           ");
```

"""" "<undefined>"

**alert( )    confirm( )    prompt**

var theResponse = window.prompt("      ", "      ")  
                        '

---

[© 2000 Microsoft Corporation](#)

JScript

---

JScript	
JScript ECMA	<a href="#"><u>JScript ECMA</u></a>
JScript -ECMA	<a href="#"><u>JScript -</u></a> <a href="#"><u>ECMA</u></a>

---

[© 2000 Microsoft Corporation](#)

JScript

---

# Microsoft JScript - ECMA

	/
	<a href="#">Array</a> , <a href="#">concat</a> , <a href="#">join</a> , <a href="#">length</a> , <a href="#">reverse</a> , <a href="#">slice</a> <a href="#">sort</a>
	<a href="#">(=)</a> <a href="#">(OP=)</a>
Boolean	<a href="#">Boolean</a> <a href="#">/*...*/</a> <a href="#">//</a>
/	<a href="#">NaN</a> <a href="#">null</a> true, false <a href="#">Infinity</a> <a href="#">undefined</a>
	<a href="#">Break</a> <a href="#">continue</a> <a href="#">do...while</a> <a href="#">for</a> <a href="#">for...in</a> <a href="#">if...else</a> <a href="#">Labeled</a> <a href="#">return</a> <a href="#">switch</a> <a href="#">while</a>
	<a href="#">Date</a> <a href="#">getDate</a> , <a href="#">getDay</a> , <a href="#">getFullYear</a> , <a href="#">getHours</a> , <a href="#">getMilliseconds</a> , <a href="#">getMinutes</a> , <a href="#">getMonth</a> , <a href="#">getSeconds</a> , <a href="#">getTime</a> , <a href="#">getTimezoneOffset</a> , <a href="#">getYear</a> , <a href="#">getUTCDate</a> , <a href="#">getUTCDay</a> , <a href="#">getUTCFullYear</a> , <a href="#">getUTCHours</a> , <a href="#">getUTCMilliseconds</a> , <a href="#">getUTCMilliseconds</a> , <a href="#">getUTCMonth</a> , <a href="#">getUTCSeconds</a> , <a href="#">setDate</a> , <a href="#">setFullYear</a> , <a href="#">setHours</a> , <a href="#">setMilliseconds</a> , <a href="#">setMinutes</a> , <a href="#">setMonth</a> , <a href="#">setSeconds</a> , <a href="#">setTime</a> , <a href="#">setYear</a> , <a href="#">setUTCDate</a> , <a href="#">setUTCFullYear</a> , <a href="#">setUTCHours</a> , <a href="#">setUTCMilliseconds</a> , <a href="#">setUTCMilliseconds</a> , <a href="#">setUTCMonth</a> , <a href="#">setUTCSeconds</a> , <a href="#">toGMTString</a> , <a href="#">toLocaleString</a> , <a href="#">toUTCString</a> , <a href="#">parse</a> , <a href="#">UTC</a>
	<a href="#">Function</a> <a href="#">new</a> <a href="#">this</a> <a href="#">var</a> <a href="#">with</a>
	<a href="#">Error</a> , <a href="#">description</a> , <a href="#">number</a> , <a href="#">throw</a> , <a href="#">try...catch</a>
	<a href="#">Caller</a> , <a href="#">Function</a> <a href="#">arguments</a> , <a href="#">length</a>
Global	<a href="#">Global</a> <a href="#">escape</a> , <a href="#">unescape</a> <a href="#">eval</a> <a href="#">isFinite</a> , <a href="#">isNaN</a> <a href="#">parseInt</a> , <a href="#">parseFloat</a>
	<a href="#">Math</a> <a href="#">abs</a> , <a href="#">acos</a> , <a href="#">asin</a> , <a href="#">atan</a> , <a href="#">atan2</a> , <a href="#">ceil</a> , <a href="#">cos</a> , <a href="#">exp</a> , <a href="#">floor</a> , <a href="#">log</a> , <a href="#">max</a> , <a href="#">min</a> , <a href="#">pow</a> ,

	<a href="#">random</a> , <a href="#">round</a> , <a href="#">sin</a> , <a href="#">sqrt</a> , <a href="#">tan</a> , <a href="#">E</a> , <a href="#">LN2</a> , <a href="#">LN10</a> , <a href="#">LOG2E</a> , <a href="#">LOG10E</a> , <a href="#">PI</a> , <a href="#">SQRT1_2</a> , <a href="#">SQRT2</a>
	<a href="#">Number</a> <a href="#">MAX_VALUE</a> , <a href="#">MIN_VALUE</a> <a href="#">NaN</a> <a href="#">NEGATIVE_INFINITY</a> , <a href="#">POSITIVE_INFINITY</a>
	<a href="#">Object</a> <a href="#">new</a> <a href="#">constructor</a> , <a href="#">prototype</a> , <a href="#">toString</a> , <a href="#">valueOf</a>
	<a href="#">(+)</a> , <a href="#">(-)</a> <a href="#">(%)</a> <a href="#">(*)</a> , <a href="#">(/)</a> <a href="#">(-)</a> <a href="#">(==)</a> , <a href="#">(!=)</a> <a href;"="">(&lt;)</a> , <a href;"="">(&lt;=)</a> <a href;"="">(&gt;)</a> <a href;"="">(&gt;=)</a> <a href;"="">(&amp;&amp;)</a> , <a href;"="">(  )</a> , <a href;"="">(!)</a> <a href;"="">(&amp;)</a> , <a href;"="">( )</a> , <a href;"="">(~)</a> , <a href;"="">(^)</a> <a href;"="">(&lt;&lt;)</a> , <a href;"="">(&gt;&gt;)</a> <a href;"="">(&gt;&gt;&gt;)</a> <a href;"="">(?:)</a> <a href;"="">(.)</a> <a href="#">delete</a> , <a href="#">typeof</a> , <a href="#">void</a> <a href="#">(--)</a> , <a href="#">(++)</a> , <a href="#">(=====)</a> , <a href="#">(!==)</a>
	<a href="#">Array</a> <a href="#">Boolean</a> <a href="#">Date</a> <a href="#">Function</a> <a href="#">Global</a> <a href="#">Math</a> <a href="#">Number</a> <a href="#">Object</a> <a href="#">RegExp</a> <a href="#">String</a>
	<a href="#">RegExp</a> <a href="#">index</a> , <a href="#">input</a> , <a href="#">lastIndex</a> , <a href="#">\$1...\$9</a> , <a href="#">source</a> , <a href="#">compile</a> , <a href="#">exec</a> , <a href="#">test</a> <a href="#">Regular Expression Syntax</a>
	<a href="#">String</a> <a href="#">charAt</a> , <a href="#">charCodeAt</a> , <a href="#">fromCharCode</a> <a href="#">indexOf</a> , <a href="#">lastIndexOf</a> <a href="#">split</a> <a href="#">toLowerCase</a> , <a href="#">toUpperCase</a> <a href="#">length</a> <a href="#">concat</a> , <a href="#">slice</a> <a href="#">match</a> , <a href="#">replace</a> , <a href="#">search</a> <a href="#">anchor</a> , <a href="#">big</a> , <a href="#">blink</a> , <a href="#">bold</a> , <a href="#">fixed</a> , <a href="#">fontcolor</a> , <a href="#">fontsize</a> , <a href="#">italics</a> , <a href="#">link</a> , <a href="#">small</a> , <a href="#">strike</a> , <a href="#">sub</a> , <a href="#">sup</a>

JScript

---

## Microsoft JScript - ECMA

	/
	<a href="#">VBArray</a> <a href="#">dimensions</a> , <a href="#">getItem</a> , <a href="#">lbound</a> , <a href="#">toArray</a> , <a href="#">ubound</a>
	<a href="#">@cc_on</a> <a href="#">@if</a> <a href="#">@set</a>
	<a href="#">getVarDate</a>
	<a href="#">Enumerator</a> <a href="#">atEnd</a> , <a href="#">item</a> , <a href="#">moveFirst</a> , <a href="#">moveNext</a>
	<a href="#">Enumerator</a>  <a href="#">VBArray</a> <a href="#">ActiveXObject</a> <a href="#">GetObject</a>
	<a href="#">ScriptEngine</a> <a href="#">ScriptEngineBuildVersion</a> <a href="#">ScriptEngineMajorVersion</a> <a href="#">ScriptEngineMinorVersion</a>

JScript

---

# JScript

	<u>\$1...\$9</u>
	<u>abs</u>
	<u>acos</u>
Automation	<u>ActiveXObject</u>
	<u>+</u>
NAME HTML	<u>anchor</u>
	<u>arguments</u>
	<u>Array</u>
	<u>asin</u>
	<u>=</u>
	<u>atan</u>
X y, x	<u>atan2</u>
Boolean	<u>atEnd</u>
<b>String</b> HTML <BIG>	<u>big</u>
	<u>&amp;</u>
	<u>&lt;&lt;</u>
	<u>~</u>
	<u> </u>
	<u>&gt;&gt;</u>
	<u>^</u>
HTML <BLINK> <b>String</b>	<u>blink</u>
HTML <B> <b>String</b>	<u>bold</u>
Boolean	<u>Boolean</u>
<i>label</i>	<u>break</u>
	<u>caller</u>
<b>try</b>	<u>catch</u>
	<u>@cc_on</u>
	<u>ceil</u>
	<u>charAt</u>
Unicode	<u>charCodeAt</u>
	<u>,</u>
JScript	<u>//</u>
JScript	<u>/*..*/</u>
Boolean	
	<u>compile</u>
	<u>concat Array</u>
<b>String</b>	<u>concat String</u>
JScript	
	<u>?:</u>

	<a href="#">constructor</a>	
	<a href="#">continue</a>	
	<a href="#">cos</a>	
	<a href="#">Date</a>	
	<a href="#">--</a>	
	<a href="#">delete</a>	
	<a href="#">description</a>	
	<a href="#">Dictionary</a>	
VBArray	<a href="#">dimensions</a>	
	<a href="#">/_</a>	
<b>false</b>	<a href="#">do...while</a>	
Euler	<a href="#">E</a>	
	<a href="#">Enumerator</a>	
	<a href="#">==</a>	
JScript	<a href="#">Error</a>	
<b>String</b>	<a href="#">escape</a>	
JScript	<a href="#">eval</a>	
	<a href="#">exec</a>	
<i>e</i>	<a href="#">exp</a>	
	<a href="#">FileSystemObject</a>	
HTML <TT> <b>String</b>	<a href="#">fixed</a>	
	<a href="#">floor</a>	
HTML COLOR <FONT>	<b>String</b>	<a href="#">fontcolor</a>
HTML SIZE <FONT>	<b>String</b>	<a href="#">fontsize</a>
<b>true</b>	<a href="#">for</a>	
	<a href="#">for...in</a>	
Unicode	<a href="#">fromCharCode</a>	
	<a href="#">Function</a>	
	<a href="#">function</a>	
<b>Date</b>	<a href="#">getDate</a>	
<b>Date</b>	<a href="#">getDay</a>	
<b>Date</b>	<a href="#">getFullYear</a>	
<b>Date</b>	<a href="#">getHours</a>	
	<a href="#">getItem</a>	
<b>Date</b>	<a href="#">getMilliseconds</a>	
<b>Date</b>	<a href="#">getMinutes</a>	
<b>Date</b>	<a href="#">getMonth</a>	
Automation	<a href="#">GetObject</a>	
<b>Date</b>	<a href="#">getSeconds</a>	
<b>Date</b>	<a href="#">getTime</a>	
UTC	<a href="#">getTimezoneOffset</a>	
UTC <b>Date</b>	<a href="#">getUTCDate</a>	
UTC <b>Date</b>	<a href="#">getUTCDay</a>	

UTC	<b>Date</b>	<a href="#">getUTCFullYear</a>	
UTC	<b>Date</b>	<a href="#">getUTCHours</a>	
UTC	<b>Date</b>	<a href="#">getUTCMilliseconds</a>	
UTC	<b>Date</b>	<a href="#">getUTCMinutes</a>	
UTC	<b>Date</b>	<a href="#">getUTCMonth</a>	
UTC	<b>Date</b>	<a href="#">getUTCSeconds</a>	
<b>Date</b>	VT_DATE	<a href="#">getVarDate</a>	
<b>Date</b>		<a href="#">getYear</a>	
		<a href="#">Global</a>	
		<a href="#">&gt;</a>	
		<a href="#">&gt;=</a>	
		<a href="#">====</a>	
		<a href="#">@if</a>	
		<a href="#">if..else</a>	
		<a href="#">++</a>	
		<a href="#">index</a>	
<b>String</b>		<a href="#">indexOf</a>	
		<a href="#">!=</a>	
<b>Number.POSITIVE_INFINITY</b>		<a href="#">Infinity</a>	
		<a href="#">input</a>	
Boolean		<a href="#">instanceof</a>	
Boolean		<a href="#">isFinite</a>	
Boolean	<b>NaN</b>	<a href="#">isNaN</a>	
HTML <I>	<b>String</b>	<a href="#">italics</a>	
		<a href="#">item</a>	
<b>String</b>		<a href="#">join</a>	
		<a href="#">Labeled</a>	
		<a href="#">lastIndex</a>	
<b>String</b>		<a href="#">lastIndexOf</a>	
VBArray		<a href="#">lbound</a>	
1		<a href="#">length Array</a>	
		<a href="#">length Function</a>	
<b>String</b>		<a href="#">length String</a>	
		<a href="#">&lt;</a>	
		<a href="#">&lt;=</a>	
HREF	HTML	<b>String</b>	<a href="#">link</a>
2			<a href="#">LN2</a>
10			<a href="#">LN10</a>
			<a href="#">log</a>
2	e Euler		<a href="#">LOG2E</a>
10 e Euler			<a href="#">LOG10E</a>
			<a href="#">&amp;&amp;</a>
			<a href="#">!</a>
			<a href="#">  </a>
			<a href="#">match</a>

	<a href="#">Math</a>
	<a href="#">max</a>
JScript	<a href="#">MAX_VALUE</a>
	<a href="#">min</a>
JScript	<a href="#">MIN_VALUE</a>
	<a href="#">%</a>
	<a href="#">moveFirst</a>
	<a href="#">moveNext</a>
	<a href="#">*</a>
<b>NaN</b>	<a href="#">NaN Global</a>
<b>NaN</b>	<a href="#">NaN Number</a>
JScript - <b>Number.MAX_VALUE</b>	<a href="#">NEGATIVE_INFINITY</a>
	<a href="#">new</a>
	<a href="#">!==</a>
	<a href="#">Number</a>
	<a href="#">number</a>
JScript	<a href="#">Object</a>
JScript	
197011	<a href="#">parse</a>
	<a href="#">parseFloat</a>
	<a href="#">parseInt</a>
3.141592653589793	<a href="#">PI</a>
JScript <b>Number.MAX_VALUE</b>	<a href="#">POSITIVE_INFINITY</a>
	<a href="#">pow</a>
	<a href="#">prototype</a>
0 1	<a href="#">random</a>
	<a href="#">RegExp</a>
	<a href="#">replace</a>
	<a href="#">return</a>
<b>Array</b>	<a href="#">reverse</a>
	<a href="#">round</a>
JScript	
	<a href="#">ScriptEngine</a>
	<a href="#">ScriptEngineBuildVersion</a>
	<a href="#">ScriptEngineMajorVersion</a>
	<a href="#">ScriptEngineMinorVersion</a>
	<a href="#">search</a>
	<a href="#">@set</a>
<b>Date</b>	<a href="#"> setDate</a>
<b>Date</b>	<a href="#"> setFullYear</a>
<b>Date</b>	<a href="#"> setHours</a>
<b>Date</b>	<a href="#"> setMilliseconds</a>
<b>Date</b>	<a href="#"> setMinutes</a>

<b>Date</b>	<a href="#">setMonth</a>
<b>Date</b>	<a href="#">setSeconds</a>
<b>Date</b>	<a href="#">setTime</a>
UTC <b>Date</b>	<a href="#">setUTCDate</a>
UTC <b>Date</b>	<a href="#">setUTCFullYear</a>
UTC <b>Date</b>	<a href="#">setUTCHours</a>
UTC <b>Date</b>	<a href="#">setUTCMilliseconds</a>
UTC <b>Date</b>	<a href="#">setUTCMinutes</a>
UTC <b>Date</b>	<a href="#">setUTCMonth</a>
UTC <b>Date</b>	<a href="#">setUTCSeconds</a>
<b>Date</b>	<a href="#">setYear</a>
	<a href="#">sin</a>
	<a href="#">slice Array</a>
	<a href="#">slice String</a>
HTML <SMALL> <b>String</b>	<a href="#">small</a>
<b>Array</b>	<a href="#">sort</a>
	<a href="#">source</a>
	<a href="#">split</a>
	<a href="#">sqrt</a>
0.5 1 2	<a href="#">SQRT1_2</a>
2	<a href="#">SQRT2</a>
HTML <STRIKE> <b>String</b>	<a href="#">strike</a>
	<a href="#">String</a>
HTML <SUB> <b>String</b>	<a href="#">sub</a>
	<a href="#">substr</a>
<b>String</b>	<a href="#">substring</a>
	<a href="#">-</a>
HTML <SUP> <b>String</b>	<a href="#">sup</a>
	<a href="#">switch</a>
JScript	
	<a href="#">tan</a>
Boolean	<a href="#">test</a>
	<a href="#">this</a>
<b>try...catch</b>	<a href="#">throw</a>
VBArray JScript	<a href="#">toArray</a>
GMT	<a href="#">toGMTString</a>
	<a href="#">toLocaleString</a>
	<a href="#">toLowerCase</a>
	<a href="#">toString</a>
	<a href="#">toUpperCase</a>
UTC	<a href="#">toUTCString</a>
JScript	<a href="#">try</a>
	<a href="#">typeof</a>
VBArray	<a href="#">ubound</a>
	<a href="#">-</a>

<b>escape</b>	<b>String</b>	<a href="#">unescape</a>
		<a href="#">&gt;&gt;&gt;</a>
197011 UTC GMT		<a href="#">UTC</a>
		<a href="#">valueOf</a>
		<a href="#">var</a>
Visual Basic		<a href="#">VBArray</a>
		<a href="#">void</a>
<b>false</b>		<a href="#">while</a>
		<a href="#">with</a>

---

[© 2000 Microsoft Corporation](#)

JScript

---

# **JScript**

JScript	
JScript	

---

[© 2000 Microsoft Corporation](#)

JScript

---

# JScript

JScript JScript JScript

5029	
5030	
5028	<a href="#">Array</a> <a href="#">arguments</a> _____
5010	<a href="#">Boolean</a>
5003	
5000	<a href="#">'this'</a>
5006	<a href="#">Date</a>
5015	<a href="#">Enumerator</a>
5022	_____
5020	<a href="#">")</a>
5019	<a href="#">“I”</a>
5023	<a href="#">Prototype</a>
5002	<a href="#">Function</a>
5008	
5021	
5014	<a href="#">JScript</a>
5001	<a href="#">Number</a>
5007	<a href="#">Object</a>
5012	
5016	
5005	<a href="#">String</a>
5017	
5026	
5027	
5025	<a href="#">URI</a>
5024	<a href="#">URI</a>
5009	
5018	
5013	<a href="#">VBArray</a>

[JScript](#)

JScript

---

# JScript

JScript JScript JScript JScript

1019	<u>“break”</u>
1020	<u>“continue”</u>
1030	
1027	<u>“switch”</u> <u>“default”</u>
1005	<u>“(”</u>
1006	<u>)”</u>
1012	<u>“/”</u>
1003	<u>“.”</u>
1004	<u>“;”</u>
1032	<u>“@”</u>
1029	<u>“@end”</u>
1007	<u>“ ”</u>
1008	<u>“{”</u>
1009	<u>“}”</u>
1011	<u>“=”</u>
1033	<u>“catch”</u>
1031	
1023	
1010	
1028	
1024	<u>“while”</u>
1014	
1026	
1025	
1018	<u>'return'</u>
1002	
1035	<u>“Throw”</u>
1016	
1015	

JScript

JScript

---

## JScript

Automation	<a href="#">GetObject</a>
	<a href="#">ScriptEngine</a>
	<a href="#">ScriptEngineBuildVersion</a>
	<a href="#">____</a>
	<a href="#">ScriptEngineMajorVersion</a>
	<a href="#">____</a>
	<a href="#">ScriptEngineMinorVersion</a>
	<a href="#">____</a>

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **GetObject**

[Automation](#)

**GetObject**([*pathname*] [, *class*])

*pathname*

*pathname*    *class*

*class*

*appname.objecttype*

*appname*

*objecttype*

**GetObject** Automation

**GetObject**

var CADObject;

CADObject = **GetObject**("C:\\CAD\\SCHEMA.CAD");

*pathname*                  *pathname* ("")        **GetObject**  
*pathname*        **GetObject**

(!)

SCHEMA.CAD

```
var LayerObject = GetObject("C:\\CAD\\SCHEMA.CAT
```

Automation

*class*

```
var MyObject;
```

```
MyObject = GetObject("C:\\DRAWINGS\\SAMPLE.DR
```

```
FIGMENT DRAWING MyObject
```

```
MyObject.Line(9, 90);
```

```
MyObject.InsertText(9, 100, "Hello, world.");
```

```
MyObject.SaveAs("C:\\DRAWINGS\\SAMPLE.DRW");
```

**GetObject**

**ActiveXObject**

**ActiveXObject**

**GetObject** ("")

*pathname*

[5](#)

[ActiveXObject](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **ScriptEngine**

ScriptEngine( )

## **ScriptEngine**

JScript	Microsoft JScript Scripting
VBA	Microsoft Visual Basic for Applications Scripting
VBScript	Microsoft Visual Basic Scripting Edition Scripting

## **ScriptEngine**

```
function GetScriptEngineInfo(){
    var s;
    s = "";
    s += ScriptEngine() + " Version ";
    s += ScriptEngineMajorVersion() + ".";
    s += ScriptEngineMinorVersion() + ".";
    s += ScriptEngineBuildVersion();
    return(s);
}
```

[ScriptEngineBuildVersion](#) | [ScriptEngineMajorVersion](#) |  
[ScriptEngineMinorVersion](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **ScriptEngineBuildVersion**

Scripting

ScriptEngineBuildVersion( )

Scripting (DLL)

## **ScriptEngineBuildVersion**

```
function GetScriptEngineInfo(){
    var s;
    s = "";
    s += ScriptEngine() + " Version ";
    s += ScriptEngineMajorVersion() + ".";
    s += ScriptEngineMinorVersion() + ".";
    s += ScriptEngineBuildVersion();
    return(s);
}
```

[5](#)

[ScriptEngine](#) | [ScriptEngineMajorVersion](#) | [ScriptEngineMinorVersion](#)

---

[© 2000 Microsoft Corporation](#)

---

JScript

---

## **ScriptEngineMajorVersion**

Scripting

ScriptEngineMajorVersion( )

Scripting (DLL)

## **ScriptEngineMajorVersion**

```
function GetScriptEngineInfo(){
    var s;
    s = "";
    s += ScriptEngine() + " Version ";
    s += ScriptEngineMajorVersion() + ".";
    s += ScriptEngineMinorVersion() + ".";
    s += ScriptEngineBuildVersion();
    return(s);
}
```

[5](#)

[ScriptEngine](#) | [ScriptEngineBuildVersion](#) | [ScriptEngineMinorVersion](#)

---

[© 2000 Microsoft Corporation](#)

---

JScript

---

## **ScriptEngineMinorVersion**

Scripting

ScriptEngineMinorVersion( )

Scripting (DLL)

## **ScriptEngineMinorVersion**

```
function GetScriptEngineInfo(){
    var s;
    s = "";
    s += ScriptEngine() + " Version ";
    s += ScriptEngineMajorVersion() + ".";
    s += ScriptEngineMinorVersion() + ".";
    s += ScriptEngineBuildVersion();
    return(s);
}
```

[5](#)

[ScriptEngine](#) | [ScriptEngineBuildVersion](#) | [ScriptEngineMajorVersion](#)

---

[© 2000 Microsoft Corporation](#)

---

JScript

---

# JScript

	<a href="#">abs</a>
	<a href="#">acos</a>
NAME HTML	<a href="#">anchor</a>
	<a href="#">asin</a>
	<a href="#">atan</a>
X y, x	<a href="#">atan2</a>
Boolean	<a href="#">atEnd</a>
<b>String</b> HTML <BIG>	<a href="#">big</a>
HTML <BLINK> <b>String</b>	<a href="#">blink</a>
HTML <B> <b>String</b>	<a href="#">bold</a>
	<a href="#">ceil</a>
	<a href="#">charAt</a>
Unicode	<a href="#">charCodeAt</a>
	<a href="#">compile</a>
	<a href="#">concat Array</a>
<b>String</b>	<a href="#">concat String</a>
	<a href="#">cos</a>
VBArray	<a href="#">dimensions</a>
<b>String</b>	<a href="#">escape</a>
JScript	<a href="#">eval</a>
	<a href="#">exec</a>
e	<a href="#">exp</a>
HTML <TT> <b>String</b>	<a href="#">fixed</a>
	<a href="#">floor</a>
HTML COLOR <FONT> <b>String</b>	<a href="#">fontcolor</a>
HTML SIZE <FONT> <b>String</b>	<a href="#">fontsize</a>
Unicode	<a href="#">fromCharCode</a>
<b>Date</b>	<a href="#">getDate</a>
<b>Date</b>	<a href="#">getDay</a>
<b>Date</b>	<a href="#">getFullYear</a>
<b>Date</b>	<a href="#">getHours</a>
	<a href="#">getItem</a>
<b>Date</b>	<a href="#">getMilliseconds</a>
<b>Date</b>	<a href="#">getMinutes</a>
<b>Date</b>	<a href="#">getMonth</a>
<b>Date</b>	<a href="#">getSeconds</a>
<b>Date</b>	<a href="#">getTime</a>
UTC	<a href="#">getTimezoneOffset</a>
UTC <b>Date</b>	<a href="#">getUTCDate</a>

UTC	<b>Date</b>	<a href="#">getUTCDay</a>
UTC	<b>Date</b>	<a href="#">getUTCFullYear</a>
UTC	<b>Date</b>	<a href="#">getUTCHours</a>
UTC	<b>Date</b>	<a href="#">getUTCMilliseconds</a>
UTC	<b>Date</b>	<a href="#">getUTCMinutes</a>
UTC	<b>Date</b>	<a href="#">getUTCMonth</a>
UTC	<b>Date</b>	<a href="#">getUTCSeconds</a>
<b>Date</b>	VT_DATE	<a href="#">getVarDate</a>
<b>Date</b>		<a href="#">getYear</a>
<b>String</b>		<a href="#">indexOf</a>
Boolean		<a href="#">isFinite</a>
Boolean	<b>NaN</b>	<a href="#">isNaN</a>
HTML <I>	<b>String</b>	<a href="#">italics</a>
		<a href="#">item</a>
<b>String</b>		<a href="#">join</a>
<b>String</b>		<a href="#">lastIndexOf</a>
VBArray		<a href="#">lbound</a>
HREF	HTML	<b>String</b>
		<a href="#">link</a>
		<a href="#">log</a>
		<a href="#">match</a>
		<a href="#">max</a>
		<a href="#">min</a>
		<a href="#">moveFirst</a>
		<a href="#">moveNext</a>
197011		<a href="#">parse</a>
		<a href="#">parseFloat</a>
		<a href="#">parseInt</a>
		<a href="#">pow</a>
0	1	<a href="#">random</a>
		<a href="#">replace</a>
<b>Array</b>		<a href="#">reverse</a>
		<a href="#">round</a>
		<a href="#">search</a>
<b>Date</b>		<a href="#"> setDate</a>
<b>Date</b>		<a href="#"> setFullYear</a>
<b>Date</b>		<a href="#"> setHours</a>
<b>Date</b>		<a href="#"> setMilliseconds</a>
<b>Date</b>		<a href="#"> setMinutes</a>
<b>Date</b>		<a href="#"> setMonth</a>
<b>Date</b>		<a href="#"> setSeconds</a>
<b>Date</b>		<a href="#"> setTime</a>
UTC	<b>Date</b>	<a href="#">setUTCDate</a>
UTC	<b>Date</b>	<a href="#">setUTCFullYear</a>
UTC	<b>Date</b>	<a href="#">setUTCHours</a>
UTC	<b>Date</b>	<a href="#">setUTCMilliseconds</a>

UTC	<b>Date</b>	<a href="#">setUTCMonth</a>
UTC	<b>Date</b>	<a href="#">setUTCSeconds</a>
	<b>Date</b>	<a href="#">setYear</a>
		<a href="#">sin</a>
		<a href="#">slice Array</a>
		<a href="#">slice String</a>
HTML <SMALL>	<b>String</b>	<a href="#">small</a>
	<b>Array</b>	<a href="#">sort</a>
		<a href="#">split</a>
		<a href="#">sqrt</a>
HTML <STRIKE>	<b>String</b>	<a href="#">strike</a>
HTML <SUB>	<b>String</b>	<a href="#">sub</a>
		<a href="#">substr</a>
	<b>String</b>	<a href="#">substring</a>
HTML <SUP>	<b>String</b>	<a href="#">sup</a>
		<a href="#">tan</a>
Boolean		<a href="#">test</a>
VBArray JScript		<a href="#">toArray</a>
GMT		<a href="#">toGMTString</a>
		<a href="#">toLocaleString</a>
		<a href="#">toLowerCase</a>
		<a href="#">toString</a>
		<a href="#">toUpperCase</a>
UTC		<a href="#">toUTCString</a>
VBArray		<a href="#">ubound</a>
<b>escape</b> String		<a href="#">unescape</a>
197011 UTC GMT		<a href="#">UTC</a>
		<a href="#">valueOf</a>

---

JScript

---

## **abs**

**Math.abs(*number*)**

*number*

*number*

## **abs**

```
function ComparePosNegVal(n)
{
    var s;
    var v1 = Math.abs(n);
    var v2 = Math.abs(-n);
    if (v1 == v2)
        s = n + " "
        s += -n + "";
    return(s);
}
```

[1](#)

Math

Math

---

© 2000 Microsoft Corporation

JScript

---

## **acos**

**Math.acos(*number*)**

*number*

*number*

[1](#)

[asin](#) | [atan](#) | [cos](#) | [sin](#) | [tan](#)

[Math](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **anchor**

NAME HTML

*strVariable*.**anchor**(*anchorString*)

*strVariable*

**String**

*anchorString*

HTML NAME

**anchor**   **String**        **anchor**

```
var strVariable = "This is an anchor" ;  
strVariable = strVariable.anchor("Anchor1");
```

*strVariable*

<A NAME="Anchor1">This is an anchor</A>

[1](#)

[link](#) | [String](#) | [String](#)

## String

---

[© 2000 Microsoft Corporation](#)

JScript

---

## apply

**apply([thisObj[,argArray]])**

thisObj

argArray

*argArray*    **arguments**    **TypeError**

*argArray*    *thisObj*            **Global**    *thisObj*

[5.5](#)

[Function](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **asin**

**Math.asin(*number*)**

*number*

[1](#)

[acos](#) | [atan](#) | [cos](#) | [Math](#) | [sin](#) | [tan](#)

[Math](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **atan**

**Math.atan(*number*)**

*number*

[1](#)

[acos](#) | [asin](#) | [atan2](#) | [cos](#) | [Math](#) | [sin](#) | [tan](#)

[Math](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **atan2**

X ( y,x)

Math.atan2(y, x)

x

X

y

y

-pi pi ( y,x)

1

[atan](#) | [tan](#)

[Math](#)

JScript

---

## **atEnd**

Boolean

*myEnum.atEnd()*

*myEnum Enumerator*

,           **atEnd**    **true**    **false**

## **atEnd**

function ShowDriveList(){

  var fso, s, n, e, x;

  fso = new ActiveXObject("Scripting.FileSystemObject")

  e = new Enumerator(fso.Drives);

  s = "";

  for (; !e.atEnd(); e.moveNext())

  {

    x = e.item();

    s = s + x.DriveLetter;

    s += " - ";

    if (x.DriveType == 3)

      n = x.ShareName;

    else if (x.IsReady)

      n = x.VolumeName;

    else

```
    n = "[      ]";
    s +=  n + "<br>";
}
return(s);
}
```

[2](#)

[item](#) | [moveFirst](#) | [moveNext](#)

[Enumerator](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **big**

HTML <BIG>      **String**

*strVariable*.**big( )**

*strVariable*    **String**

## **big**

```
var strVariable = "This is a string object";
strVariable = strVariable.big( );
```

*strVariable*

<BIG>This is a string object</BIG>

[1](#)

[small](#) | [String](#) | [String](#)

[String](#)

JScript

---

## **blink**

HTML <BLINK>      **String**

*strVariable*.**blink( )**

*strVariable*    **String**

## **blink**

```
var strVariable = "This is a string object";
strVariable = strVariable.blink( );
```

*strVariable*

<BLINK>This is a string object</BLINK>

Microsoft Internet Explorer <BLINK>

[1](#)

[String](#) | [String](#)

[String](#)

JScript

---

## **bold**

HTML <B>      **String**

*strVariable*.**bold()**

*strVariable*    **String**

## **bold**

```
var strVariable = "This is a string object";
strVariable = strVariable.bold( );
```

*strVariable*

<B>This is a string object</B>

[1](#)

[italics](#) | [String](#) | [String](#)

[String](#)

JScript

---

**call**

**call**([*thisObj*[,*arg1*[, *arg2*[, [.,*argN*]]]])

*thisObj*

*arg1*, *arg2*, , *argN*

**call**    **call**    *thisObj*

*thisObj*            **Global**    *thisObj*

5.5

[Function](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **ceil**

**Math.ceil(*number*)**

*number*

[1](#)

[floor](#) | [Math](#)

[Math](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **charAt**

*strObj.charAt(index)*

*strObj*

**String**

*index*

0 1

**charAt** 0 1

## **charAt**

```
function charAtTest(n){  
    var str = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"; /  
    var s; //  
    s = str.charAt(n - 1); // n - 1  
    //  
    return(s); //  
}
```

1

[String](#) | [String](#)

[String](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **charCodeAt**

Unicode

*strObj.charCodeAt(index)*

strObj

**String**

index

0 1

0 1

**NaN**

## **charCodeAt**

```
function charCodeAtTest(n){  
    var str = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"; //  
    var n;                                //  
    n = str.charCodeAt(n - 1);              //      n      Unicode  
    return(n);                            //  
}
```

5.5

[fromCharCode](#) | [String](#)

[String](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **compile**

*rgExp.compile(pattern, [flags])*

rgexp

pattern

*flags*

- g (     *pattern*)
- i ()
- m ()

**compile**    *pattern*

## **compile**

```
function CompileDemo(){  
    var rs;  
    var s = "AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPp"  
    //
```

```
var r = new RegExp("[A-Z]", "g");
var a1 = s.match(r)          //
//  

r.compile("[a-z]", "g");
var a2 = s.match(r)          //
return(a1 + "\n" + a2;
}
```

[3](#)

| . | .

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **concat (Array)**

*array1.concat([item1[, item2[, . . . [, itemN]]]])*

*array1*

**Array**

*item1, . . . , itemN*

*array1*

**concat    Array              array1**

*item1 ... itemN              array1*

•

•

**concat :**

```
function ConcatArrayDemo(){  
    var a, b, c, d;  
    a = new Array(1,2,3);  
    b = "JScript";  
    c = new Array(42, "VBScript");
```

```
d = a.concat(b, c);
//      [1, 2, 3, "JScript", 42, "VBScript"]
return(d);
}
```

[3](#)

[concat \(String\)](#) | [join](#) | [String](#)

[Array](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **concat String**

*string1.concat([string2[, string3[, . . . [, stringN]]]])*

*string1*

**String**

*string2, . . . , stringN*

*string1    String*

**concat**    *result = string1 + string2 + string3 + ... + stringN*                  *string1*

**concat**

function concatDemo()

{

    var str1 = "ABCDEFGHIJKLM"

    var str2 = "NOPQRSTUVWXYZ";

    var s = **str1.concat(str2);**

//

    return(s);

}

3

(+) | [Array](#) | [concat Array](#) | [String](#)

[String](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **COS**

**Math.cos(number)**

*number*

[1](#)

[acos](#) | [asin](#) | [atan](#) | [Math](#) | [sin](#) | [tan](#)

[Math](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **decodeURI**

(URI)

### **decodeURI(*URIdstring*)**

*URIdstring* URI

**decodeURI    unescape**

## **decodeURI**

*URIdString* URIError

[5.5](#)

[decodeURIComponent](#) | [encodeURIComponent](#)

[Global](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **decodeURIComponent**

(URI)

### **decodeURIComponent(*encodedURIStr*ing)**

*encodedURIStr*ing URI

URIComponent URI

*encodedURIStr*ing URIError

[5.5](#)

[decodeURI](#) | [encodeURI](#)

[Global](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **dimensions**

VBArray

*array*.**dimensions( )**

*array*    **VBArray**

## **dimensions** VBArray

Visual Basic VBScript JScript HTML

<HEAD> <BODY> JScript

```
<HEAD>
<SCRIPT LANGUAGE="VBScript">
<!--
Function CreateVBArray()
    Dim i, j, k
    Dim a(2, 2)
    k = 1
    For i = 0 To 2
        For j = 0 To 2
            a(j, i) = k
            k = k + 1
        Next
    Next
    CreateVBArray = a
End Function
-->
```

```
</SCRIPT>

<SCRIPT LANGUAGE="JScript">
<!--
function VBArrayTest(vba)
{
    var i, s;
    var a = new VBArray(vba);
    for (i = 1; i <= a.dimensions(); i++)
    {
        s = "The upper bound of dimension ";
        s += i + " is ";
        s += a.ubound(i)+ ".<BR>";
    }
    return(s);
}
-->
</SCRIPT>
</HEAD>
```

```
<BODY>
<SCRIPT language="jscript">
    document.write(VBArrayTest(CreateVBArray()));
</SCRIPT>
</BODY>
```

[getItem](#) | [lbound](#) | [toArray](#) | [ubound](#)

[VBArray](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **encodeURI**

(URI)

**encodeURI(URIStrong)**

*URIStrong* URI

**encodeURI** URI      **decodeURI**      **encodeURI** ":""/"";" "?"  
**encodeURIComponent**

[5.5](#)

[decodeURI](#) | [decodeURIComponent](#)

[Global](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **encodeURIComponent**

(URI)

**encodeURIComponent**(*encodedURIStr**t*)

*encodedURIStr**t* URI

**encodeURIComponent** URI

**encodeURIComponent**

web URI

**decodeURIComponent**

/folder1/folder2/default.html

**encodeURI**

[5.5](#)

[decodeURI](#) | [decodeURIComponent](#)

[Global](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

**escape**

**String**

**escape**(*charString*)

*charstring* **String**

**escape** *charstring* Unicode                    [ASCII](#)        %*xx*  
xx "%20"

255        %**u**xxxx

**escape** (URI)            **encodeURI** **encodeURIComponent**

[1](#)

[encodeURI](#) | [encodeURIComponent](#) | [String](#) | [unescape](#)

[Global](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **eval**

JScript .

**eval**(*codeString*)

*codeString* JScript JScript

**eval** JScript ,                   **Date**     *mydate*

**eval**("var mydate = new Date();");

**eval**   **eval** .

1

[String](#)

[Global](#)

JScript

---

**exec**

*rgExp.exec(str)*

*rgExp*

*str*

**String**

**exec**      **null**      **exec**      **RegExp** 01n (g)  
**match**

**exec**      **lastIndex**      **exec**      **lastIndex**  
**exec**      **inputindex**      **lastIndexInput**      **Index**      **LastIndex**

**exec**

```
function RegExpTest(){
    var ver = Number(ScriptEngineMajorVersion() + "." + ScriptEngineMinorVersion());
    if (ver >= 5.5){ // JScript
        var src = "The rain in Spain falls mainly in the plain.";
        var re = /\w+/g; //
        var arr;
```

```
while ((arr = re.exec(src)) != null)
    document.write(arr.index + "-" + arr.lastIndex + "\t" +
)
else{
    alert("      JScript      ");
}
}
```

[3](#)

[match](#) | [RegExp](#) | . | . | [search](#) | [test](#)

JScript

---

**exp**

e

**Math.exp(*number*)**

*number* . e

$e^{\text{number}}$     e 2.178            *number*

[1](#)

[E](#) | [Math](#)

[Math](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **fixed**

HTML <TT>      **String**

*strVariable.fixed( )*

**strVariable**    **String**

## **fixed**

```
var strVariable = "This is a string object";
strVariable = strVariable.fixed( );
```

*strVariable*

<TT>This is a string object</TT>

[1](#)

[String](#) | [String](#)

[String](#)

JScript

---

## **floor**

**Math.floor(*number*)**

*number* .

[1](#)

[ceil](#) | [Math](#)

[Math](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **fontcolor**

COLOR HTML <FONT>      **String**

*strVariable*.**fontcolor**(*colorVal*)

*strVariable*

**String**

*colorVal*

## **fontcolor**

```
var strVariable = "This is a string";
strVariable = strVariable.fontcolor("red");
```

*strVariable*

<FONT COLOR="RED">This is a string</FONT>

JScript

[1](#)

[fontsize](#) | [String](#) | [String](#)

[String](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **fontsize**

SIZE HTML <FONT>      **String**

*strVariable*.**fontsize**(*intSize*)

*strVariable*

**String**

*intSize*

## **fontsize**

```
var strVariable = "This is a string";
strVariable = strVariable.fontsize(-1);
```

*strVariable*

<FONT SIZE="-1">This is a string</FONT>

Microsoft JScript

[fontcolor](#) | [String](#) | [String](#)

[String](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **fromCharCode**

Unicode

**String.fromCharCode([code1[, code2[, ...[, codeN]]]])**

## **String**

### **String**

*code1, . . . , codeN*

Unicode

## **fromCharCode String**

*test* "plain"

var test = **String.fromCharCode(112, 108, 97, 105, 110);**

[3](#)

## [charCodeAt](#) | [String](#)

### [String](#)

---

JScript

---

## **getDate**

**Date**

*dateObj.getDate()*

*dateObj* **Date**

## (UTC) **getUTCDDate**

1 31 **Date**

## **getDate**

```
function DateDemo(){
    var d, s = "      : ";
    d = new Date();
    s += (d.getMonth() + 1) + "/";
    s += d.getDate() + "/";
    s += d.getYear();
    return(s);
}
```

1

[Date](#) | [getUTCDate](#) |  [setDate](#) | [setUTCDate](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **getDay**

**Date**

*dateObj.getDay()*

*dateObj* **Date**

(UTC) **getUTCDay**

**getDay** 0 6

0	
1	
2	
3	
4	
5	
6	

**getDay**

```
function DateDemo(){
    var d, day, x, s = "      : ";
    var x = new Array("      ", "      ", "      ");
    var x = x.concat("      ", "      ", "      ");
    var x = x.concat("      ");
    d = new Date();
    day = d.getDay();
    return(s += x[day]);
}
```

1

[Date](#) | [getUTCDay](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **getFullYear**

**Date**

*dateObj.getFullYear()*

*dateObj* **Date**

(UTC) **getUTCFullYear**

**getFullYear** 1976 1976 2000 2000 11 1900 11

**GetFullYear**

```
function DateDemo(){
    var d, s = "    UTC    : ";
    d = new Date();
    s += (d.getMonth() + 1) + "/";
    s += d.getDate() + "/";
    s += d.getFullYear();
    return(s);
}
```

3

[Date](#) | [getUTCFullYear](#) | [setFullYear](#) | [setUTCFullYear](#)

Date

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **getHours**

**Date**

*dateObj*.**getHours()**

*dateObj* **Date**

(UTC)

**getUTCHours**

**getHours** 0 23 0 1:00:00 am  
0 0

**Date**

**Date**

**getHours**

```
function TimeDemo(){  
    var d, s = "           : ";  
    var c = ":";  
    d = new Date();  
    s += d.getHours() + c;  
    s += d.getMinutes() + c;  
    s += d.getSeconds() + c;  
    s += d.getMilliseconds();  
    return(s);  
}
```

1

[Date](#) | [getUTCHours](#) | [setHours](#) | [setUTCHours](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **getItem**

*safeArray*.**getItem**(*dimension1*[, *dimension2*, ...], *dimensionN*)

*safeArray*

**VBArray**

*dimension1*, ..., *dimensionN*

VBArray        *n* VBArray

Visual Basic   VBScript   JScript   HTML  
<HEAD>   <BODY>   JScript

```
<HEAD>
<SCRIPT LANGUAGE="VBScript">
<!--
Function CreateVBArray()
    Dim i, j, k
    Dim a(2, 2)
    k = 1
    For i = 0 To 2
        For j = 0 To 2
            a(i, j) = k
            document.writeln(k)
            k = k + 1
    End For
End Function
```

```
Next
document.writeln("<BR>")
Next
CreateVBArry = a
End Function
-->
</SCRIPT>
<SCRIPT LANGUAGE="JScript">
<!--
function GetItemTest(vbarray)
{
    var i, j;
    var a = new VBArry(vbarray);
    for (i = 0; i <= 2; i++)
    {
        for (j =0; j <= 2; j++)
        {
            document.writeln(a.getItem(i, j));
        }
    }
}-->
</SCRIPT>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="JScript">
<!--
    GetItemTest(CreateVBArry());
-->
```

```
</SCRIPT>  
</BODY>
```

[1](#)

[dimensions](#) | [lbound](#) | [toArray](#) | [ubound](#)

[VBArry](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **getMilliseconds**

**Date**

*dateObj.getMilliseconds()*

*dateObj* **Date**

(UTC)

## **getUTCMilliseconds**

0-999

## **getMilliseconds**

```
function TimeDemo(){
    var d, s = "          : ";
    var c = ":";
    d = new Date();
    s += d.getHours() + c;
    s += d.getMinutes() + c;
    s += d.getSeconds() + c;
    s += d.getMilliseconds();
    return(s);
}
```

3

[Date](#) | [getUTCMilliseconds](#) | [setMilliseconds](#) | [setUTCMilliseconds](#)

[Date](#) | [getUTCMilliseconds](#) | [setMilliseconds](#) | [setUTCMilliseconds](#)

---

---

---

© 2000 Microsoft Corporation

JScript

---

## **getMinutes**

**Date**

*dateObj.getMinutes()*

*dateObj* **Date**

## (UTC) **getUTCMINUTES**

**getMinutes** 0 59  
0

**Date** 0

**Date**

## **getMinutes**

```
function TimeDemo(){
    var d, s = "          : ";
    var c = ":";
    d = new Date();
    s += d.getHours() + c;
    s += d.getMinutes() + c;
    s += d.getSeconds() + c;
    s += d.getMilliseconds();
    return(s);
}
```

3

[Date](#) | [getUTCMinutes](#) | [setMinutes](#) | [setUTCMinutes](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **getMonth**

**Date**

*dateObj.getMonth()*

*dateObj* **Date**

## (UTC) **getUTCMonth**

**getMonth** 0 11 **Date** 1 **Date** "Jan 5, 1996  
08:47:00" **getMonth** 0

## **getMonth**

```
function DateDemo(){
    var d, s = "      : ";
    d = new Date();
    s += (d.getMonth() + 1) + "/";
    s += d.getDate() + "/";
    s += d.getFullYear();
    return(s);
}
```

1

[Date](#) | [getUTCMonth](#) | [setMonth](#) | [setUTCMonth](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **getSeconds**

**Date**

*dateObj.getSeconds()*

*dateObj*      **Date**

## (UTC)      **getUTCSeconds**

**getSeconds** 0 59                  **Date** 0  
0 0

**Date**  
**Date**

## **getSeconds**

```
function TimeDemo(){
    var d, s = "          : ";
    var c = ":";
    d = new Date();
    s += d.getHours() + c;
    s += d.getMinutes() + c;
    s += d.getSeconds() + c;
    s += d.getMilliseconds();
    return(s);
}
```

1

[Date](#) | [getUTCSeconds](#) | [setSeconds](#) | [setUTCSeconds](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **getTime**

**Date**

*dateObj*.**getTime()**

*dateObj*    **Date**

**getTime** 1970 1 1  
1970

**Date** 1970 1 1 285,616

```
var MinMilli = 1000 * 60
var HrMilli = MinMilli * 60
var DyMilli = HrMilli * 24
```

## **getTime**

```
function GetTimeTest(){
    var d, s, t;
    var MinMilli = 1000 * 60;
    var HrMilli = MinMilli * 60;
    var DyMilli = HrMilli * 24;
    d = new Date();
    t = d.getTime();
    s = "It's been "
    s += Math.round(t / DyMilli) + " days since 1/1/70";
```

```
    return(s);  
}
```

1

[Date](#) | [setTime](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **getTimezoneOffset**

(UTC)

*dateObj*.**getTimezoneOffset()**

*dateObj*    **Date**

**getTimezoneOffset** UTC

UTC ( Pacific Daylight Time) UTC ( Japan)

12 1 Los Angeles New York City

**getTimezoneOffset** 480 300

## **getTimezoneOffset**

```
function TZDemo(){  
    var d, tz, s = " The current local time is ";  
    d = new Date();  
    tz = d.getTimezoneOffset();  
    if (tz < 0)  
        s += tz / 60 + " hours before GMT";  
    else if (tz == 0)  
        s += "GMT";  
    else  
        s += tz / 60 + " hours after GMT";  
    return(s);
```

}

1

Date

Date

---

© 2000 Microsoft Corporation

JScript

---

## **getUTCDate**

**Date** [\(UTC\)](#)

*dateObj.getUTCDate()*

*dateObj*    **Date**

## **getDate**

1 31                **Date**

## **getUTCDate**

```
function UTCDateDemo(){
    var d, s = "    UTC    : ";
    d = new Date();
    s += (d.getUTCMonth() + 1) + "/";
    s += d.getUTCDate() + "/";
    s += d.getUTCFullYear();
    return(s);
}
```

[3](#)

[Date](#) | [getDate](#) |  [setDate](#) | [setUTCDate](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **getUTCDay**

**Date** [\(UTC\)](#)

*dateObj.getUTCDay()*

*dateObj* **Date**

## **getDate**

**getUTCDay** 0 6

0
1
2
3
4
5
6

## **getUTCDay**

```
function DateDemo(){
    var d, day, x, s = "      ";
    var x = new Array("      ", "      ", "      ");
    x = x.concat("      ","      ","      ");
    x = x.concat("      ");
    d = new Date();
    day = d.getUTCDay();
    return(s += x[day]);}
```

}

3

[Date](#) | [getDay](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **getUTCFullYear**

**Date** [\(UTC\)](#)

*dateObj.getUTCFullYear()*

*dateObj*      **Date**

### **getFullYear**

**getUTCFullYear** 2000 2000 11 1900 11

## **getUTCFullYear**

```
function UTCDateDemo(){
    var d, s = "    UTC    : ";
    d = new Date();
    s += (d.getUTCMonth() + 1) + "/";
    s += d.getUTCDate() + "/";
    s += d.getUTCFullYear();
    return(s);
}
```

[3](#)

[Date](#) | [getFullYear](#) | [setFullYear](#) | [setUTCFullYear](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **getUTCHours**

**Date** [\(UTC\)](#)

*dateObj.getUTCHours()*

*dateObj* **Date**

## **getHours**

**getUTCHours** 0 23 0 1:00:00 A.M.  
0 0

**Date**  
**Date**

## **getUTCHours**

```
function UTCTimeDemo(){  
    var d, s = "           (UTC) : ";  
    var c = ":";  
    d = new Date();  
    s += d.getUTCHours() + c;  
    s += d.getUTCMilliseconds() + c;  
    s += d.getUTCSeconds() + c;  
    s += d.getUTCMinutes() + c;  
    return(s);  
}
```

3

[Date](#) | [getHours](#) | [setHours](#) | [setUTCHours](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **getUTCMilliseconds**

**Date** [\(UTC\)](#)

*dateObj.getUTCMilliseconds()*

*dateObj*    **Date**

## **getMilliseconds**

0-999

## **getUTCMilliseconds**

```
function UTCTimeDemo(){
    var d, s = "          (UTC) : ";
    var c = ":";
    d = new Date();
    s += d.getUTCHours() + c;
    s += d.getUTCMinutes() + c;
    s += d.getUTCSeconds() + c;
    s += d.getUTCMilliseconds();
    return(s);
}
```

[Date](#) | [getMilliseconds](#) | [setMilliseconds](#) | [setUTCMilliseconds](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **getUTCMMinutes**

**Date** [\(UTC\)](#)

*dateObj.getUTCMMinutes()*

*dateObj* **Date**

## **getMinutes**

**getUTCMMinutes** 0 59  
0 0

**Date** 0 1

**Date**

## **getUTCMMinutes**

```
function UTCTimeDemo()
{
    var d, s = "          (UTC) : ";
    var c = ":";
    d = new Date();
    s += d.getUTCHours() + c;
    s += d.getUTCMMinutes() + c;
    s += d.getUTCSeconds() + c;
    s += d.getUTCMilliseconds();
    return(s);
}
```

3

[Date](#) | [getMinutes](#) | [setMinutes](#) | [setUTCMinutes](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **getUTCMonth**

**Date** [\(UTC\)](#)

*dateObj.getUTCMonth()*

*dateObj* **Date**

## **getMonth**

**getUTCMonth** 0 11 Date 1  
1996 08:47:00.0" **getUTCMonth** 0 **Date** "Jan 5,

## **getUTCMonth**

```
function UTCDateDemo(){  
    var d, s = "    UTC      : ";  
    d = new Date();  
    s += (d.getUTCMonth() + 1) + "/";  
    s += d.getUTCDate() + "/";  
    s += d.getUTCFullYear();  
    return(s);  
}
```

[Date](#) | [getMonth](#) | [setMonth](#) | [setUTCMonth](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **getUTCSeconds**

**Date** [\(UTC\)](#)

*dateObj.getUTCSeconds()*

*dateObj* **Date**

## **getSeconds**

**getUTCSeconds** 0 59  
0 0

**Date** 0 1

**Date**

## **getUTCSeconds**

```
function UTCTimeDemo(){
    var d, s = "    UTC    : ";
    var c = ":";
    d = new Date();
    s += d.getUTCHours() + c;
    s += d.getUTCMinutes() + c;
    s += d.getUTCSeconds() + c;
    s += d.getUTCMilliseconds();
    return(s);
}
```

3

[Date](#) | [getSeconds](#) | [setSeconds](#) | [setUTCSeconds](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **getVarDate**

**Date** VT\_DATE

*dateObj*.**getVarDate()**

*dateObj*    **Date**

COM ActiveX(R) VT\_DATE Visual Basic VBScript

**getVarDate** JScript

[3](#)

[getDate](#) | [parse](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **getYear**

**Date**

*dateObj*.**getYear()**

*dateObj*    **Date**

## **getFullYear**

1900-1999 1900 1996 96 1825 2025  
1825 2025

JScript 1.0              **getYear**    **Date** 1900 1899 -1  
2000 100

## **getYear**

```
function DateDemo(){
    var d, s = "      : ";
    d = new Date();
    s += (d.getMonth() + 1) + "/";
    s += d.getDate() + "/";
    s += d.getYear();
    return(s);
}
```

1

[Date](#) | [getFullYear](#) | [getUTCFullYear](#) | [setFullYear](#) | [setUTCFullYear](#)  
| [setYear](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **indexOf**

**String**

*strObj.indexOf(subString[, startIndex])*

*strObj*

**String**

*subString*

**String**

*startIndex*

**String**

**indexOf**      **String** -1

*startindex*      *startindex*

**lastIndexOf**

## **indexOf**

```
function IndexDemo(str2){  
    var str1 = "BABEBIBOBUBABEBIBOBU"  
    var s = str1.indexOf(str2);  
    return(s);
```

}

1

[lastIndexOf](#) | [String](#) | [String](#)

[String](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **isFinite**

Boolean ,

**isFinite(*number*)**

*number*

<i>number</i>	<b>NaN</b>	<b>isFinite</b>	<b>true</b>	<b>false</b>
---------------	------------	-----------------	-------------	--------------

[3](#)

[isNaN](#)

[Global](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **isNaN**

Boolean      **NaN**

**isNaN**(*numValue*)

*numvalue*      **NAN**

**NaN**      **isNaN**    **true**    **false**      **parseInt**    **parseFloat**

**NaN**      **NaN**

[1](#)

[isFinite](#) | [NaN \(Global\)](#) | [parseFloat](#) | [parseInt](#)

[Global](#)

JScript

---

## **italics**

HTML <I>      **String**

*strVariable.italics( ) "String Literal".italics( )*

## **italics**

var strVariable = "This is a string";  
strVariable = strVariable.**italics( )**;

*strVariable*

<I>This is a string</I>

[1](#)

[\*\*bold\*\*](#) | [\*\*String\*\*](#) | [\*\*String\*\*](#)

[\*\*String\*\*](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

**item**

*enumObj.item()*

*myEnum*    **Enumerator**

**item**        **undefined**

**item**    **Drives**

```
function ShowDriveList(){
    var fso, s, n, e, x;
    fso = new ActiveXObject("Scripting.FileSystemObject"
    e = new Enumerator(fso.Drives);
    s = "";
    for (; !e.atEnd(); e.moveNext())
    {
        x = e.item();
        s = s + x.DriveLetter;
        s += " - ";
        if (x.DriveType == 3)
            n = x.ShareName;
        else if (x.IsReady)
            n = x.VolumeName;
        else
```

```
n = "[      ]";
s += n + "<br>";
}
return(s);
}
```

3

[atEnd](#) | [moveFirst](#) | [moveNext](#)

[Enumerator](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **join**

*arrayObj*.**join**(*separator*)

*arrayObj*

**Array**

*separator*

**String**

**String**

**null**

**join**

```
function JoinDemo(){  
    var a, b;  
    a = new Array(0,1,2,3,4);  
    b = a.join("-");  
    return(b);  
}
```

[2](#)

[Array](#) | [String](#)

[Array](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **lastIndexOf**

**String**

*strObj.lastIndexOf(substring[, startIndex])*

*strObj*

**String**

*substring*

**String**

*startIndex*

**String**

**lastIndexOf**      **String** -1

*startIndex*      *startIndex*

**indexOf**

**lastIndexOf**

function lastIndexDemo(str2)

{

  var str1 = "BABEBIBOBUBABEBIBOBU"

  var s = str1.lastIndexOf(str2);

  return(s);

}

1

[indexOf](#) | [String](#) | [String](#)

[String](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **lbound**

VBArray

safeArray.**lbound**(dimension)

safeArray

**VBArray**

dimension

VBArray

**lbound** 1

VBArray      **lbound**    undefined    *dimension*    VBArray “  
”

Visual Basic   VBScript   JScript   VBScript  
Visual Basic   0   HTML   <HEAD>   <BODY>   JScript

```
<HEAD>
<SCRIPT LANGUAGE="VBScript">
<!--
Function CreateVBArray()
Dim i, j, k
Dim a(2, 2)
k = 1
```

```
For i = 0 To 2
    For j = 0 To 2
        a(j, i) = k
        k = k + 1
    Next
Next
CreateVBArry = a
End Function
-->
</SCRIPT>
```

```
<SCRIPT LANGUAGE="JScript">

<!--
function VBArrayTest(vba)
{
    var i, s;
    var a = new VBArray(vba);
    for (i = 1; i <= a.dimensions(); i++)
    {
        s = "The lower bound of dimension ";
        s += i + " is ";
        s += a.lbound(i)+ ".<BR>";
        return(s);
    }
-->
```

```
</SCRIPT>

</HEAD>

<BODY>

<SCRIPT language="jscript">

    document.write(VBArrayTest(CreateVBArray()));

</SCRIPT>

</BODY>
```

[3](#)

[dimensions](#) | [getItem](#) | [toArray](#) | [ubound](#)

[VBArray](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **link**

HREF HTML      **String**

*strVariable.link(linkstring)* "String Literal".link(*linkstring*)

*linkstring* HTML HREF

**link**   **String**

```
var strVariable = "This is a hyperlink";
strVariable = strVariable.link("http://www.microsoft.com")
```

*strVariable*

<a href="http://www.microsoft.com">This is a hyperlink<

1

[anchor](#) | [String](#) | [String](#)

[String](#)

JScript

---

## **localeCompare**

*stringVar.localeCompare(stringExp)*

*stringVar*

**String**

*stringExp*

*stringVar*

**localeCompare**    *stringVar*    *stringExp* -10 +1

*stringVar*    *stringExp*         **localeCompare** -1         *stringVar*  
    *stringExp* +1 0

[5.5](#)

[toLocaleString](#)

**String**

JScript

---

**log**

**Math.log(*number*)**

*number*

*number*      *e*

1

[Math](#)

[Math](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

**match**

`stringObj.match(rgExp)`

*stringObj*

# String

*rgExp*

**match**      **null**    **RegExp**

**match      inputIndexlastIndexInput      Index      LastIndex**

(g)0 1 n **exec** 0n

**match**

```
function MatchDemo(){
```

```
var r, re; //
```

```
var s = "The rain in Spain falls mainly in the plain";
```

re = /ain/i; //

```
r = s.match(re); //
```

```
return(r);      // "ain"
```

}

```
g match

function MatchDemo(){
    var r, re;      //
    var s = "The rain in Spain falls mainly in the plain";
    re = /ain/ig;   //
    r = s.match(re); // 
    return(r);      //           "ain"
                    //
}
```

### **match**

```
var r, re = "Spain";
r = "The rain in Spain".replace(re, "Canada");
```

[3](#)

[exec](#) | [RegExp](#) | [replace](#) | [search](#) | [String](#) | [test](#)

[String](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **max**

**Math.max([number1[, number2[... [,numberN]]]])**

*number1, number2, . . . , numberN*

**NEGATIVE\_INFINITY    NaN    NaN**

[1](#)

[Math](#) | [min](#) | [NEGATIVE\\_INFINITY](#)

[Math](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **min**

**Math.min([number1[, number2[. . . [,numberN]]]])**

*number1, number2, . . . , numberN*

**POSITIVE\_INFINITY    NaN    NaN**

[1](#)

[Math](#) | [max](#) | [POSITIVE\\_INFINITY](#)

[Math](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **moveFirst**

*enumObj*.**moveFirst( )**

*enumObj*    **Enumerator**

[undefined](#)

### **moveFirst    Drives**

```
function ShowFirstAvailableDrive(){
    var fso, s, e, x;          //
    fso = new ActiveXObject("Scripting.FileSystemObject")
    e = new Enumerator(fso.Drives); //      Enumerator
    e.moveFirst();           //
    s = "";                     //      s
    do
    {
        x = e.item();           //
        if (x.IsReady)          //
        {
            s = x.DriveLetter + ":"; //      s
            break;
        }
    else
```

```
    if (e.atEnd())          //
    {
        s = "";
        break;
    }
    e.moveToNext();          //
}
while (!e.atEnd());          //
return(s);                  //
}
```

3

[atEnd](#) | [item](#) | [moveNext](#)

[Enumerator](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **moveNext**

*enumObj.moveNext( )*

*myEnum*    **Enumerator**

[undefined](#)

### **moveNext**   Drives

```
function ShowDriveList(){
    var fso, s, n, e, x;          //
    fso = new ActiveXObject("Scripting.FileSystemObject")
    e = new Enumerator(fso.Drives); //    Enumerator
    s = "";                      //    s
    for (; !e.atEnd(); e.moveNext())
    {
        x = e.item();
        s = s + x.DriveLetter;      //
        s += " - ";                // "-"
        if (x.DriveType == 3)
            n = x.ShareName;       //
        else if (x.IsReady)
            n = x.VolumeName;      //
        else
            n = "[           ]";   //
        s += n + "\n";
    }
}
```

```
    }  
    return(s); //  
}
```

3

[atEnd](#) | [item](#) | [moveFirst](#)

[Enumerator](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **parse**

1970 1 1

### **Date.parse(dateVal)**

*dateVal* "Jan 5, 1996 08:47:00" ActiveX(R) VT\_DATE

**parse**        *dateVal* 1970 1 1

**parse**    **Date**                      **Date**

var datestring = "November 1, 1997 10:15 AM";  
Date.parse(datestring)

### **parse**

- "/"“-”//”7/20/96”
- "July 10 1995" 2 4 2 70
- 
- 
- "Ju"
- 1996 11 9 "Tuesday November 9 1996"                      **Date** "Friday November 9 1996"
- JScript (UTC) (GMT)
- "10:”"10:11" "10:11:12"
- 24 12 "PM" "23:15 PM"
- 

## **parse**

```
function GetTimeTest(testdate){  
    var s, t;                //  
    var MinMilli = 1000 * 60;   //  
    var HrMilli = MinMilli * 60;  
    var DyMilli = HrMilli * 24;  
    t = Date.parse(testdate);    //      testdate  
    s = "There are "           //  
    s += Math.round(Math.abs(t / DyMilli)) + " days "  
    s += "between " + testdate + " and 1/1/70";  
    return(s);                 //  
}
```

[1](#)

[Date](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **parseFloat**

**parseFloat(*numString*)**

*numString*

**parseFloat**    *numString*            *numString*            **NaN**

**parseFloat("abc")**    //     NaN  
**parseFloat("1.2abc")** //     1.2

**isNaN**    **NaN**

[1](#)

[isNaN](#) | [parseInt](#) | [String](#)

[Global](#)

JScript

---

## **parseInt**

**parseInt(*numString*, [*radix*])**

*numString*

*radix*

2 36            *numString* '0x' '0'

**parseInt**    *numString*            *numString*            **NaN**

**parseInt("abc")**    //    **NaN**

**parseInt("12abc")** //    12

**isNaN**    **NaN**

[1](#)

[isNaN](#) | [parseFloat](#) | [String](#) | [valueOf](#)

[Global](#)

JScript

---

**pop**

*arrayObj.pop( )*

*arrayObj*    **Array**

**undefined**

[5.5](#)

[push](#)

[Array](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **pow**

**Math.pow(base, exponent)**

*base*

*exponent*

$base^{exponent}$  . 1000

**Math.pow(10,3);**

[1](#)

[Math](#)

[Math](#)

JScript

---

## **push**

*arrayObj.push([item1 [item2 [...] [itemN ]]])*

arrayObj

**Array**

item, item2,... itemN

**Array**

**push**

**concat**

[5.5](#)

[concat | pop](#)

[Array](#)

JScript

---

**random**

0 1

Math.random( )

0 1 0 101 JScript

1

Math

Math

---

© 2000 Microsoft Corporation

JScript

---

## **replace**

stringObj.replace(*rgExp*, *replaceText*)

stringObj

**String**      **replace**

*rgExp*

**String**      *rgExp*

*replaceText*

**String**      *stringObj*    *rgExp* Jscript 5.5      *replaceText*

**replace**    *stringObj*

<b>\$\$</b>	\$ JScript 5.5
<b>\$&amp;</b>	<i>stringObj</i> JScript 5.5
<b>\$`</b>	\$& <i>stringObj</i> JScript 5.5
<b>\$'</b>	\$& <i>stringObj</i> JScript 5.5
<b>\$n</b>	<i>n</i> <i>n</i> 19 JScript 5.5
<b>\$nn</b>	<i>nn</i> <i>nn</i> 0199 JScript 5.5

*replaceText* m+3 m                                      *rgExp*                      *m*  
m+2    *stringObj*        m+3    *stringObj*

## Replace    RegExp

```
replace "The" "A"

function ReplaceDemo(){
    var r, re;          //
    var ss = "The man hit the ball with the bat.\n";
    ss += "while the fielder caught the ball with the glove.";
    re = /The/g;        //
    r = ss.replace(re, "A"); // "A"      "The"
    return(r);          //
}

, replace

function ReplaceDemo(){
    var r, re;          //
    var ss = "The rain in Spain falls mainly in the plain.";
    re = /(\S+)(\s+)(\S+)/g;   //
    r = ss.replace(re, "$3$2$1"); // 
    return(r);          //
}

JSscript 5.5           replaceText "F" "Water boils at
212"                 

function f2c(s) {
    var test = /(\d+(\.\d*)?)F\b/g;  //
    return(s.replace
        (test,
```

```
function($0,$1,$2) {
    return((( $1 - 32) * 5/9) + "C");
}
);
}
document.write(f2c("Water freezes at 32F and boils at 212F"))
```

[1](#)

[exec](#) | [match](#) | [RegExp](#) | [search](#) | [String](#) | [test](#)

[String](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

**reverse**

**Array**

*arrayObj.reverse( )*

*arrayObj* **Array**

**reverse** **Array** **Array**

**reverse** [undefined](#)

**reverse**

```
function ReverseDemo(){  
    var a, l; //  
    a = new Array(0,1,2,3,4); //  
    l = a.reverse(); //  
    return(l); //  
}
```

[2](#)

[Array](#)

## Array

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **round**

`Math.round(number)`

*number*

*number* 0.5      *number*      **round**    *number*

[1](#)

[Math](#)

[Math](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **search**

*stringObj*.**search**(*rgExp*)

*stringObj*

**String**

*rgExp*

**search**           **search -1**

## **search**

```
function SearchDemo(){
    var r, re;          //
    var s = "The rain in Spain falls mainly in the plain.";
    re = /falls/i;      //
    r = s.search(re);    //
    return(r);           // Boolean
}
```

3

[exec](#) | [match](#) | . | [replace](#) | [String](#) | [test](#)

[String](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **setDate**

**Date**

*dateObj.setDate(numDate)*

*dateObj*

**Date**

*numDate*

## (UTC) **setUTCDate**

*numDate* **Date** *numDate* 1996 1 5  
**setUTCDate(32)** 1996 2 1

**setDate**

```
function SetDateDemo(newdate){  
    var d, s;          //  
    d = new Date();      //      date  
    d.setDate(newdate); //      date newdate  
    s = "Current setting is ";  
    s += d.toLocaleString();  
    return(s);          //  
}
```

3

[Date](#) | [getDate](#) | [getUTCDate](#) | [setUTCDate](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **setFullYear**

**Date**

*dateObj.setFullYear(numYear[, numMonth[, numDate]])*

*dateObj*

**Date**

*numYear*

*numMonth*

*numDate*

*numDate*

**set**    **get**

*numMonth* JScript

**getMonth**

[\(UTC\)](#)

**setUTCFullYear**

Date 1970 285,616

**setFullYear**

function SetFullYearDemo(newyear){

```
var d, s;          //  
d = new Date();      //    Date  
d.setFullYear(newyear); //  
s = "Current setting is ";  
s += d.toLocaleString();  
return(s);          //  
}
```

[3](#)

[Date](#) | [getFullYear](#) | [getUTCFullYear](#) | [setUTCFullYear](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **setHours**

**Date**

*dateObj.setHours(numHours[, numMin[, numSec[, numMilli]]])*

*dateObj*

**Date**

*numHours*

*numMin*

*numSec*

*numMilli*

**set    get**

*numMinutes* JScript

**getMinutes**

[\(UTC\)](#)

**setUTCHours**

"Jan 5, 1996 00:00:00"  
06:00:00."

**setHours(30)** "Jan 6, 1996

### **setHours**

```
function SetHoursDemo(nhr, nmin, nsec){  
    var d, s;                //  
    d = new Date();           //      Date  
    d.setHours(nhr, nmin, nsec); //  
    s = "Current setting is " + d.toLocaleString()  
    return(s);                //  
}
```

[3](#)

[Date](#) | [getHours](#) | [getUTCHours](#) | [setUTCHours](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **setMilliseconds**

**Date**

*dateObj.setMilliseconds(numMilli)*

*dateObj*

**Date**

*numMilli*

## (UTC)      **setUTCMilliseconds**

*numMilli 999 ()*

## **setMilliseconds**

```
function SetMSecDemo(nmsec){  
    var d, s;           //  
    var sep = ":";       //  
    d = new Date();      // Date  
    d.setMilliseconds(nmsec); //  
    s = "Current setting is ";  
    s += d.toLocaleString() + sep + d.getMilliseconds();  
    return(s);           //
```

}

3

[Date](#) | [getMilliseconds](#) | [getUTCMilliseconds](#) | [setUTCMilliseconds](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **setMinutes**

**Date**

*dateObj.setMinutes(numMinutes[, numSeconds[, numMilli]])*

*dateObj*

**Date**

*numMinutes*

*numSeconds*

*numMilli*

*numMilli*

**set**

**get**

*numSeconds* JScript

**getSeconds**

[\(UTC\)](#)

**setUTCMMinutes**

"Jan 5, 1996 00:00:00"  
01:30:00"

**setMinutes(90)** "Jan 5, 1996

**setMinutes**

function SetMinutesDemo(nmin, nsec){

```
var d, s;          //  
d = new Date();      //    Date  
d.setMinutes(nmin, nsec); //  
s = "Current setting is " + d.toLocaleString()  
return(s);          //  
}
```

[1](#)

[Date](#) | [getMinutes](#) | [getUTCMinutes](#) | [setUTCMinutes](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **setMonth**

**Date**

*dateObj.setMonth(numMonth[, dateVal])*

*dateObj*

**Date**

*numMonth*

*dateVal*

**getDate**

(UTC)

**setUTCMonth**

*numMonth* 11 ( 0 ) "Jan 5, 1996" **setMonth(14)**  
"Mar 5, 1997."

**setMonth**

```
function SetMonthDemo(newmonth){  
    var d, s;          //  
    d = new Date();      //  Date  
    d.setMonth(newmonth); //  
    s = "Current setting is ";
```

```
s += d.toLocaleString();
return(s);           //
}
```

[1](#)

[Date](#) | [getMonth](#) | [getUTCMonth](#) | [setUTCMonth](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **setSeconds**

**Date**

*dateObj.setSeconds(numSeconds[, numMilli])*

*dateObj*

**Date**

*numSeconds*

*numMilli*

<b>set</b>	<b>get</b>	<i>numMilli JScript</i>	<b>getMilliseconds</b>
------------	------------	-------------------------	------------------------

[\(UTC\)](#)      **setUTCSeconds**

"Jan 5, 1996 00:00:00"	<b>setSeconds(150)</b>	"Jan 5, 1996 00:02:30."
------------------------	------------------------	----------------------------

**setSeconds**

```
function SetSecondsDemo(nsec, nmsec){  
    var d, s; //  
    var sep = ":";
```

```
d = new Date();           //  Date
d.setSeconds(nsec, nmsec); //
s = "Current setting is ";
s += d.toLocaleString() + sep + d.getMilliseconds();
return(s);               //
}
```

[1](#)

[Date](#) | [getSeconds](#) | [getUTCSeconds](#) | [setUTCSeconds](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **setTime**

**Date**

*dateObj.setTime(milliseconds)*

*dateObj*

**Date**

*milliseconds*

GMT 1970 1 1

*milliseconds* 1970 1970 1 1 285,616

## **setTime**

### **setTime**

```
function SetTimeTest(newtime){  
    var d, s;          //  
    d = new Date();      //      Date  
    d.setTime(newtime);   //  
    s = "Current setting is ";  
    s += d.toUTCString();  
    return(s);          //  
}
```

1

[Date](#) | [getTime](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **setUTCDate**

**Date** ([UTC](#))

*dateObj.setUTCDate(numDate)*

*dateObj*

**Date**

*numDate*

## **setDate**

*numDate*    **Date**        *numDate* 1996 1 5  
**setUTCDate(32)** 1996 2 1

## **setUTCDate**

```
function SetUTCDateDemo(newdate){  
    var d, s; //  
    d = new Date(); // Date  
    d.setUTCDate(newdate); // UTC  
    s = "Current setting is ";  
    s += d.toUTCString();  
    return(s); //  
}
```

3

[Date](#) | [getDate](#) | [getUTCDate](#) |  [setDate](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **setUTCFullYear**

**Date** ([UTC](#))

*dateObj.setUTCFullYear(numYear[, numMonth[, numDate]])*

*dateObj*

**Date**

*numYear*

*numMonth*

*numDate*

*numDate*

**set**    **get**

*numMonth* JScript

**getUTCMonth**

## **setFullYear**

**Date** 1970 285,616

## **setUTCFullYear**

```
function SetUTCFullYearDemo(newyear){  
    var d, s;          //  
    d = new Date();      //      Date  
    d.setUTCFullYear(newyear); //      UTC  
    s = "Current setting is ";  
    s += d.toUTCString();  
    return(s);          //  
}
```

[3](#)

[Date](#) | [getFullYear](#) | [getUTCFullYear](#) | [setFullYear](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **setUTCHours**

**Date** ([UTC](#))

*dateObj.setUTCHours(numHours[, numMin[, numSec[, numMilli]]])*

*dateObj*

**Date**

*numHours*

*numMin*

*numSec numMilli*

*numSec*

*numMilli*

*numMilli*

**set**    **get**                      *numMin JScript*                      **getUTCMMinutes**

## **setHours**

"Jan 5, 1996 00:00:00"  
06:00:00."

**setHours(30)** "Jan 6, 1996

### **setUTCHours**

```
function SetUTCHoursDemo(nhr, nmin, nsec){  
    var d, s;                //  
    d = new Date();           //      Date  
    d.setUTCHours(nhr, nmin, nsec); //      UTC  
    s = "Current setting is " + d.toUTCString()  
    return(s);                //  
}
```

[3](#)

[Date](#) | [getHours](#) | [getUTCHours](#) | [setHours](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **setUTCMilliseconds**

**Date** [\(UTC\)](#)

*dateObj.setUTCMilliseconds(numMilli)*

*dateObj*

**Date**

*numMilli*

## **setMilliseconds**

*numMilli* 999

## **setUTCMilliseconds**

```
function SetUTCMSecDemo(nmsec){  
    var d, s;                      //  
    var sep = ":";                  //  
    d = new Date();                 //      Date  
    d.setUTCMilliseconds(nmsec);     //      UTC  
    s = "Current setting is ";  
    s += d.toUTCString() + sep + d.getUTCMilliseconds();  
    return(s);                      //
```

}

3

[Date](#) | [getMilliseconds](#) | [getUTCMilliseconds](#) | [setMilliseconds](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **setUTCMMinutes**

**Date** [\(UTC\)](#)

*dateObj.setUTCMMinutes(numMinutes[, numSeconds[, numMilli]])*

*dateObj*

**Date**

*numMinutes*

*numSeconds*

*numMilli*

*numMilli*

**getUTCSeconds**      *numSeconds* JScript

**setMinutes**

"Jan 5, 1996 00:00:00"  
01:10:00.00."

**setUTCMMinutes(70)** "Jan 5, 1996

**setUTCMMinutes**

function SetUTCMMinutesDemo(nmin, nsec){

```
var d, s;          //  
d = new Date();      //    Date  
d.setUTCMMinutes(nmin,nsec); //    UTC  
s = "Current setting is " + d.toUTCString()  
return(s);          //  
}
```

[3](#)

[Date](#) | [getMinutes](#) | [getUTCMinutes](#) | [setMinutes](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **setUTCMonth**

**Date** [\(UTC\)](#)

*dateObj.setUTCMonth(numMonth[, dateVal])*

*dateObj*

**Date**

*numMonth*

*dateVal*

**getUTCDate**

## **setMonth**

*numMonth* 11( 0 ) "Jan 5, 1996 00:00:00.00"  
**setUTCMonth(14)** "Mar 5, 1997 00:00:00.00."

## **setUTCMonth**

```
function SetUTCMonthDemo(newmonth){  
    var d, s; //  
    d = new Date(); // Date  
    d.setUTCMonth(newmonth); // UTC  
    s = "Current setting is ";
```

```
s += d.toUTCString();
return(s);           //
}
```

3

[Date](#) | [getMonth](#) | [getUTCMonth](#) | [setMonth](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **setUTCSeconds**

**Date** ([UTC](#))

*dateObj.setUTCSeconds(numSeconds[, numMilli])*

*dateObj*

**Date**

*numSeconds*

*numMilli*

**getUTCMilliseconds**      *numMilli* JScript

**setSeconds**

"Jan 5, 1996 00:00:00"  
00:02:30.00."

**setSeconds(150)** "Jan 5, 1996

**setSeconds**

```
function SetUTCSecondsDemo(nsec, nmsec){  
    var d, s; //  
    d = new Date(); // Date
```

```
d.setUTCSeconds(nsec, nmsec); // UTC
s = "Current UTC milliseconds setting is ";
s += d.getUTCMilliseconds(); //
return(s); //
}
```

3

[Date](#) | [getSeconds](#) | [getUTCSeconds](#) | [setSeconds](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

### **setYear**

Date

`dateObj.setYear(numYear)`

*dateObj*

Date

*numYear*

1900

## **setFullYear**

**Date** 1997      **setYear(97)** 2010      **setYear(2010)** 0-99  
**setFullYear**

JScript 1.0  
-1 2000

**setYear**    *numYear* 1900 1899  
                          *numYear* 100

*numYear*

1

[Date](#) | [getFullYear](#) | [getUTCFullYear](#) | [getYear](#) | [setFullYear](#) | [setUTCFullYear](#)

Date

[© 2000 Microsoft Corporation](#)

---

JScript

---

## **shift**

*arrayObj.shift( )*

*arrayObj*    **Array**

## **shift**

[5.5](#)

[unshift](#)

[Array](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **sin**

**Math.sin(*number*)**

*number*

[1](#)

[acos](#) | [asin](#) | [atan](#) | [cos](#) | [Math](#) | [tan](#)

[Math](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **slice (Array)**

arrayObj.**slice**(start, [end])

arrayObj

**Array**

start

*arrayObj*

end

*arrayObj*

**slice**   **Array**        *arrayObj*

**slice**   *end*              *start*              *length + start*      *length*              *end*  
                        *length + end*              *length*              *end*      **slice**   *arrayObj*  
*end*      *start*

*myArray*    *newArray*

*newArray* = *myArray.slice(0, -1)*

[slice \(String\) | String](#)

[Array](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **slice (String)**

*stringObj.slice(start, [end])*

stringObj

**String**

start

0           *stringObj*

end

0           *stringObj*

**slice    stringObj    String**

<b>slice</b>	<i>end</i>	<i>start</i>	<i>length + start</i>	<i>length</i>	<i>end</i>
		<i>length + end</i>	<i>length</i>	<i>end</i>	<b>slice</b>
<i>end</i>	<i>start</i>				<i>arrayObj</i>

**slice -1 str1**

**str1.slice(0)**

**str2.slice(0,-1)**

3

[Array](#) | [slice \(Array\)](#) | [String](#)

[String](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **small**

HTML <SMALL> **String**

*strVariable*.**small( )** "String Literal".**small( )**

## **small**

```
var strVariable = "This is a string";
strVariable = strVariable.small( );
```

*strVariable*

<SMALL>This is a string</SMALL>

[1](#)

[big](#) | [String](#) | [String](#)

[String](#)

JScript

---

**sort**

**Array**

arrayobj.**sort**(sortfunction)

*arrayObj*

**Array**

*sortFunction*

ASCII

**sort**    **Array**        **Array**

*sortfunction*

- 
- 
- 

**sort**

```
function SortDemo(){  
    var a, l; //  
    a = new Array("X" , "y" , "d", "Z", "v", "m", "r");  
    l = a.sort(); //  
    return(l); //
```

}

2

[Array](#)

[Array](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **splice**

*arrayObj.splice(start, deleteCount, [item1[, item2[, . . . [,itemN]]]])*

arrayObj

**Array**

start

0

deleteCount

item1, item2,. . .,itemN

**splice**    *start*        *arrayObj*    **Array**

[5.5](#)

[slice \(Array\)](#)

[Array](#)

---

[© 2000 Microsoft Corporation](#)

---

JScript

---

## **split**

*stringObj.split([separator[, limit]])*

*stringObj*

**String**      **split**

*separator*

*limit*

**split**      *stringObj*    *separator*    *separator*

**split**

function SplitDemo(){

  var s, ss;

  var s = "The rain in Spain falls mainly in the plain.";

  //

  ss = **s.split(" ")**;

  return(ss);

}

3

[concat](#) | [RegExp](#) | . | . | [String](#)

[String](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **sqrt**

**Math.sqrt(*number*)**

*number* .

*number*

[1](#)

[Math](#) | [SQRT1\\_2](#) | [SQRT2](#)

[Math](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **strike**

HTML <STRIKE>      **String**

*strVariable*.**strike( )** "String Literal".**strike( )**

## **strike**

```
var strVariable = "This is a string object";
strVariable = strVariable.strike( );
```

*strVariable*

<STRIKE>This is a string object</STRIKE>

[1](#)

[String](#) | [String](#)

[String](#)

JScript

---

## **sub**

HTML <SUB> **String**

*strVariable*.**sub( )** "String Literal".**sub( )**

## **sub**

```
var strVariable = "This is a string object";
strVariable = strVariable.sub( );
```

*strVariable*

<SUB>This is a string object</SUB>

[1](#)

[String](#) | [String](#) | [sup](#)

[String](#)

JScript

---

## **substr**

*stringvar.substr(start [, length ])*

stringvar

**String**

start

0

length

*length* 0

*stringvar*

## **substr**

```
function SubstrDemo(){
```

```
    var s, ss;          //
```

```
    var s = "The rain in Spain falls mainly in the plain.;"
```

```
    ss = s.substr(12, 5); //
```

```
    return(ss);          //    "Spain"
```

```
}
```

3

[String](#) | [String](#) | [substring](#)

[String](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **substring**

### **String**

*strVariable.substring(start, end)* "String Literal".**substring(start, end)**

start

0

end

0

**substring**    *start*        *end*

**substring**    *start*    *end*              *strvar.substring(0, 3)*  
**strvar.substring(3, 0)**

*start*    *end*    **NaN** 0

*start*    *end*              *strvar.substring(0, 3)*    *strvar.substring(3, 0)*  
3

## **substring**

function SubstringDemo(){

    var ss;                      //

    var s = "The rain in Spain falls mainly in the plain..";

    ss = s.**substring(12, 17);** //

```
    return(ss);          //  
}
```

1

[String](#) | [String](#) | [substr](#)

[String](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **sup**

HTML <SUP>      **String**

*strVariable.sup( )* "String Literal".**sup( )**

## **sup**

```
var strVariable = "This is a string object";
strVariable = strVariable.sup( );
```

*strVariable*

<SUP>This is a string object</SUP>

[1](#)

[String](#) | [String](#) | [sub](#)

[String](#)

JScript

---

## **tan**

**Math.tan(*number*)**

*number*.

*number*

[1](#)

[acos](#) | [asin](#) | [atan](#) | [atan2](#) | [cos](#) | [Math](#) | [sin](#)

[Math](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

**test**

Boolean

*rgexp*.**test(str)**

*rgexp*

str

**test**      **true**    **false**

**RegExp**   **test**

**test**

```
function TestDemo(re, s){  
    var s1; //  
    //  
    if (re.test(s)) //  
        s1 = " contains "; // s  
    else  
        s1 = " does not contain "; // s  
    return("''' + s + "'''" + s1 + "''' + re.source + "'''"); //
```

}

3

[RegExp](#) | . | . | .

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **toArray**

VBArray JScript

*safeArray*.**toArray( )**

*safeArray* **VBArray**

VBArray JScript VBArray JScript

VBArray (1, 2, 3), (4, 5, 6), (7, 8, 9) JScript 1, 2, 3, 4, 5, 6, 7,  
8, 9

JScript VBArray

Visual Basic VBScript JScript VB JScript  
HTML <HEAD> <BODY> JScript

```
<HEAD>
<SCRIPT LANGUAGE="VBScript">
<!--
Function CreateVBArray()
    Dim i, j, k
    Dim a(2, 2)
    k = 1
    For i = 0 To 2
        For j = 0 To 2
            a(j, i) = k
            document.writeln(k)
```

```
k = k + 1
Next
document.writeln("<BR>")
Next
CreateVBArry = a
End Function
-->
</SCRIPT>

<SCRIPT LANGUAGE="JScript">
<!--
function VBArrayTest(vbarray)
{
    var a = new VBArray(vbarray);
    var b = a.toArray();
    var i;
    for (i = 0; i < 9; i++)
    {
        document.writeln(b[i]);
    }
}
-->
</SCRIPT>
</HEAD>

<BODY>
<SCRIPT LANGUAGE="JScript">
<!--
```

```
VBArryTest(CreateVBArry());  
-->  
</SCRIPT>  
</BODY>
```

[3](#)

[dimensions](#) | [getItem](#) | [lbound](#) | [ubound](#)

[VBArry](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **toDateString**

*objDate*.**toDateString( )**

*objDate*    **Date**

## **toDateString**

[5.5](#)

[toTimeString](#) | [toLocaleDateString](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **toExponential**

*numObj.toExponential([fractionDigits])*

numObj

**Number**

fractionDigits

0 – 20 0 20

**toExponential**

*fractionDigits*

*fractionDigits*

**toExponential**

5.5

[toFixed](#) | [toPrecision](#)

[Number](#)

JScript

---

## **toFixed**

*numObj.toFixed([fractionDigits])*

*numObj*

**Number**

*fractionDigits*

0 – 20 0 20

**toFixed**                   *fractionDigits*

*fractionDigits*           **undefinedtoFixed** 0

[5.5](#)

[toExponential](#) | [toPrecision](#)

[Number](#)

JScript

---

## toGMTString

(GMT)

## `dateObj.toGMTString()`

## toGMTString

## toUTCString

**toGMTString**    **String** GMT "05 Jan 1996 00:00:00  
                  GMT"

1

## Date | toUTCString

Date

© 2000 Microsoft Corporation

JScript

---

## **toLocaleDateString**

*objDate.toLocaleDateString( )*

*objDate*    **Date**

**toLocaleDateString**

**toLocalDateString** –

[5.5](#)

[toDateString](#) | [toLocaleTimeString](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **toLocaleLowerCase**

*stringVar.toLocaleLowerCase( )*

*stringVar*    **String**

**toLocaleLowerCase**

**toLowerCase** Unicode

[5.5](#)

[toLocaleUpperCase](#) | [toLowerCase](#)

[String](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **toLocaleString**

*dateObj.toLocaleString()*

*dateObj*    **Date**

## **toLocaleString**    **String**

- 1601 1999 “”“”
- F        **toString**

1 5                      **toLocaleString** "01/05/96 00:00:00" "05/01/96  
00:00:00"

## **toLocaleString**

## **toLocaleString**

```
var d, s;                      //  
d = new Date();                //     Date  
s = "Current setting is ";  
s += d.toLocaleString();      //  
return(s);                    //  
}
```

[Date](#)

[Array](#) | [Date](#) | [Number](#) | [Object](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **toLocaleTimeString**

*objDate.toLocaleTimeString( )*

*objDate*    **Date**

## **toLocaleTimeString**

**toLocalTimeString** –

[5.5](#)

[ToString](#) | [toLocaleDateString](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **toLocaleUpperCase**

*stringVar.toLocaleUpperCase( )*

*stringVar*    **String**

**toLocaleUpperCase**

**toUpperCase** Unicode

[5.5](#)

[toLocaleLowerCase](#) | [toUpperCase](#)

[String](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **toLowerCase**

*strVariable.toLowerCase()* "String Literal".toLowerCase()

## **toLowerCase**

of the **toLowerCase**

```
var strVariable = "This is a STRING object";
strVariable = strVariable.toLowerCase();
```

*strVariable*

this is a string object

[1](#)

[String](#) | [String](#) | [toUpperCase](#)

[String](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **toPrecision**

*numObj.toPrecision ([precision])*

**numObj**

**Number**

**precision**

1 – 21 1 21

*precision - 1*

*precision*

**precision undefined toString**

[5.5](#)

[toFixed](#) | [toExponential](#)

[Number](#)

JScript

---

## **toString**

objectname.**toString**([radix])

objectname

radix

## **toString** JScript

Array	<b>Array</b>
Boolean	Boolean <b>true</b> “true” “false”
Date	
Error	
Function	<i>functionname</i> <b>toString</b> <b>function functionname( ) { [native code] }</b>
Number	
String	<b>String</b> “ [object objectname] ”    objectname

radix **toString** Radix

**function CreateRadixTable (){**

```
var s, s1, s2, s3, x;          //
s = "Hex  Dec  Bin \n";        //
for (x = 0; x < 16; x++)      //
{
    switch(x)                //
    {
        case 0 :             //
            s1 = "   ";       //
            s2 = "   ";       //
            s3 = "   ";       //
            break;             //
        case 1 :             //
            s1 = "   ";       //
            s2 = "   ";       //
            s3 = "   ";       //
            break;             //
        case 2 :             //
            s3 = "   ";       //
            break;             //
        case 3 :             //
            s3 = "   ";       //
            break;             //
        case 4 :             //
            s3 = "  ";        //
            break;             //
        case 5 :             //
            s3 = " ";         //
            break;             //
        case 6 :             //
            s3 = "  ";        //
            break;             //
        case 7 :             //
            s3 = " ";         //
            break;             //
        case 8 :             //
            s3 = " 0";        //
            break;             //
        case 9 :             //
            s3 = " 1";        //
            break;             //
        case 10 :            //
            s3 = " 2";        //
            break;             //
        case 11 :            //
            s3 = " 3";        //
            break;             //
        case 12 :            //
            s3 = " 4";        //
            break;             //
        case 13 :            //
            s3 = " 5";        //
            break;             //
        case 14 :            //
            s3 = " 6";        //
            break;             //
        case 15 :            //
            s3 = " 7";        //
            break;             //
    }
    document.write(s1 + s2 + s3);
}
```

```
        case 6 :  
            s3 = " ";  
            break;  
        case 7 :  
            s3 = " ";  
            break;  
        case 8 :  
            s3 = ""; ;  
            break;  
        case 9 :  
            s3 = ""; ;  
            break;  
        default:  
            s1 = "    ";  
            s2 = ""; ;  
            s3 = "    ";  
        } //  
        s += " " + x.toString(16) + s1 + x.toString(10)  
        s += s2 + s3 + x.toString(2)+ "\n";  
  
    }  
    return(s); // radix  
}
```

2

## function

[Array](#) | [Boolean](#) | [Date](#) | [Error](#) | [Function](#) | [Number](#) | [Object](#)  
| [String](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **toTimeString**

*objDate*.**toTimeString( )**

*objDate*    **Date**

## **toTimeString**

[5.5](#)

[toDateString](#) | [toLocaleTimeString](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **toUpperCase**

*strVariable.toUpperCase( )* "String Literal".toUpperCase( )

## **toUpperCase**

### **toUpperCase**

```
var strVariable = "This is a STRING object";  
strVariable = strVariable.toUpperCase();
```

*strVariable*

THIS IS A STRING OBJECT

[1](#)

[String](#) | [String](#) | [toLowerCase](#)

[String](#)

JScript

---

## **toUTCString**

[\(UTC\)](#)

*dateObj.toUTCString()*

*dateObj*    **Date**

**toUTCString**    **String**    UTC

**toUTCString**

```
function toUTCStrDemo(){
    var d, s;                //
    d = new Date();           //      Date
    s = "Current setting is ";
    s += d.toUTCString();     //      UTC
    return(s);               //      UTC
}
```

[3](#)

[Date](#) | [toGMTString](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **ubound**

VBArray

*safeArray.ubound(dimension)*

safeArray

**VBArray**

dimension

VBArray

**ubound 1**

VBArray      **ubound**    undefined    *dim*   VBArray “”

Visual Basic   VBScript   JScript   HTML  
<HEAD>   <BODY>   JScript

```
<HEAD>
<SCRIPT LANGUAGE="VBScript">
<!--
Function CreateVBArray()
    Dim i, j, k
    Dim a(2, 2)
    k = 1
    For i = 0 To 2
```

```
For j = 0 To 2
    a(j, i) = k
    k = k + 1
Next
Next
CreateVBArray = a
End Function
-->
</SCRIPT>
```

```
<SCRIPT LANGUAGE="JScript">
<!--
function VBArrayTest(vba)
{
    var i, s;
    var a = new VBArray(vba);
    for (i = 1; i <= a.dimensions(); i++)
    {
        s = "The upper bound of dimension ";
        s += i + " is ";
        s += a.ubound(i)+ ".<BR>";
        return(s);
    }
}
-->
</SCRIPT>
</HEAD>
```

```
<BODY>
<SCRIPT language="jscript">
  document.write(VBArrayTest(CreateVBArray()));
</SCRIPT>
</BODY>
```

[3](#)

[dimensions](#) | [getItem](#) | [lbound](#) | [toArray](#)

[VBArray](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

**unescape**

**escape**    **String**

**unescape**(*charstring*)

*charstring*    **String**

**unescape**    *charstring* %        xx    [ASCII](#)

%uxxxx Unicode        xxxx Unicode .

**unescape** (URI)        **decodeURI**    **decodeURIComponent**

[1](#)

[DecodeURI](#) | [decodeURIComponent](#) | [escape](#) | [String](#)

[Global](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **unshift**

*arrayObj.unshift([item1[, item2 [, . . . [, itemN]]]])*

arrayObj

**Array**

item1, item2, . . . , itemN

**Array**

## **unshift**

[5.5](#)

[shift](#)

[Array](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **UTC**

(UTC) ( GMT) 1970 1 1

**Date.UTC**(*year, month, day[, hours[, minutes[, seconds[,ms]]]]*)

year

*year* 0 99                   *year* 1900 + *year*

month

    0 11 ()

day

    1 31

hours

*minutes*0 23 ( 11pm)

minutes

*seconds*0 59

seconds

*milliseconds*0 59

ms

    0 999

**UTC** UTC 1970 1 1

150 JScript 2 30

**setTime**   **Date**

**UTC**   **Date**    **UTC** UTC       **Date**

<b>UTC</b>	<b>Date</b>
<i>year</i> 0 99	<i>1900 + year</i>

**UTC**

```
function DaysBetweenDateAndNow(yr, mo, dy){  
    var d, r, t1, t2, t3;          //  
    var MinMilli = 1000 * 60        //  
    var HrMilli = MinMilli * 60  
    var DyMilli = HrMilli * 24  
    t1 = Date.UTC(yr, mo - 1, dy) //      1/1/1970  
    d = new Date();                //      Date  
    t2 = d.getTime();              //  
    if (t2 >= t1)  
        t3 = t2 - t1;  
    else  
        t3 = t1 - t2;  
    r = Math.round(t3 / DyMilli);  
    return(r);                    //  
}
```

[Date](#) | [setTime](#)

[Date](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **valueOf**

*object.valueOf( )*

*object* JScript

JScript      **valueOf**

Array	<b>Array.toString</b> <b>Array.join</b>
Boolean	Boolean
Date	1970 1 1 <a href="#">UTC</a>
Function	
Number	
Object	
String	

**Math**   **Error**   **valueOf**

[2](#)

[toString](#)

[Array](#) | [Boolean](#) | [Date](#) | [Function](#) | [Number](#) | [Object](#) | [String](#)

JScript

---

# JScript

Automation	<a href="#">ActiveXObject</a>
	<a href="#">Array</a>
Boolean	<a href="#">Boolean</a>
	<a href="#">Date</a>
	<a href="#">Dictionary</a>
	<a href="#">Enumerator</a>
JScript	<a href="#">Error</a>
	<a href="#">FileSystemObject</a>
	<a href="#">__</a>
	<a href="#">Function</a>
	<a href="#">Global</a>
	<a href="#">Math</a>
	<a href="#">Number</a>
JScript	<a href="#">Object</a>
	<a href="#">RegExp</a>
	<a href="#">String</a>
Visual Basic	<a href="#">VBArray</a>

JScript

---

## **ActiveXObject**

[Automation](#)

*newObj = new ActiveXObject(servername.typename[, location])*

ActiveXObject

*newObj*

**ActiveXObject**

*servername*

*typename*

*location*

Automation

Automation

**ActiveXObject**

var ExcelSheet;

ExcelApp = new ActiveXObject("Excel.Application");

ExcelSheet = new ActiveXObject("Excel.Sheet");

Microsoft Excel ExcelSheet Excel  
Application ActiveSheet.Cells

```
// Excel Application
ExcelSheet.Application.Visible = true;
//
ExcelSheet.ActiveSheet.Cells(1,1).Value = "This is column
//
ExcelSheet.SaveAs("C:\\TEST.XLS");
// Application Quit Excel
ExcelSheet.Application.Quit();
```

Internet           **ActiveXObject**   *servername*  
"\myserver\public"       *servername* "myserver" DNS IP  
*servername*  
"myserver" Excel

```
function GetAppVersion() {
    var XLApp = new ActiveXObject("Excel.Application",
        return(XLApp.Version);
}
```

[1](#)

[GetObject](#)

JScript

---

## **Array**

*arrayObj = new Array()*  
*arrayObj = new Array([size])*  
*arrayObj = new Array([element0[, element1[, ...[, elementN]]]])*

*arrayObj*

**Array**

*size*

*size -1*

*element0,...,elementN*

*n + 1 n + 1*

[ ]

```
var my_array = new Array();
for (i = 0; i < 10; i++)
{
    my_array[i] = i;
}
x = my_array[4];
```

Microsoft JScript 4

**Array** 32400

**Array**      **length** 1

JScript

[constructor](#) | [length](#) | [prototype](#)

[concat](#) | [join](#) | [pop](#) | [push](#) | [reverse](#) | [shift](#) | [slice](#) | [sort](#) | [splice](#)  
[\\_\\_](#) | [toLocaleString](#) | [toString](#) | [unshift](#) | [valueOf](#)

[2](#)

[new](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **Boolean**

Boolean

*boolObj* = **new Boolean**([*boolValue*])

*boolObj*

## **Boolean**

*boolValue*

Boolean	<i>Boolvalue</i>	false	0	null	NaN	<b>Boolean</b>
<b>false</b>	<b>true</b>					

**Boolean** Boolean Boolean **Boolean** JScript **Boolean**

## **Boolean**

[constructor](#) | [prototype](#)

[toString](#) | [valueOf](#)

2

[new](#) | [var](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

Date

```
dateObj = new Date() dateObj = new Date(dateVal)  
dateObj = new Date(year, month, date[, hours[, minutes[, seconds[,ms]]]])
```

*dateObj*

Date

dateVal

*dateVal* 1970 1 1                        *dateVal*        **parse**  
*dateVal* ActiveX(R) VT\_DATE

year

1976 76

month

0 11 1 12

date

1 31

hours

*minutes* 0 23 11pm

minutes

*seconds* 0 59

seconds

*milliseconds* 0 59

ms

0 999

**Date** 150 JScript 2 30

**NaN**      **Date** (UTC)

**Date** 1970 1 1 285,616

**Date**    **Date**        **parse**    **UTC**

## **Date**

```
function DateDemo(){
    var d, s = "Today's date is: ";          //
    d = new Date();                      //  Date
    s += (d.getMonth() + 1) + "/";           //
    s += d.getDate() + "/";                 //
    s += d.getYear();                      //
    return(s);                            //
}
```

[constructor](#) | [prototype](#)

[getDate](#) | [getDay](#) | [getFullYear](#) | [getHours](#) | [getMilliseconds](#) |  
[getMinutes](#) | [getMonth](#) | [getSeconds](#) | [getTime](#) | [getTimezoneOffset](#) |

[getUTCDate](#) | [getUTCDay](#) | [getUTCFullYear](#) | [getUTCHours](#) |  
[getUTCMilliSeconds](#) | [getUTCMilliseconds](#) | [getUTCMonth](#) |  
[getUTCSeconds](#) | [getVarDate](#) | [getYear](#) | [setDate](#) | [setFullYear](#) |  
[setHours](#) | [setMilliSeconds](#) | [setMinutes](#) | [setMonth](#) | [setSeconds](#) |  
 [setTime](#) | [setUTCDate](#) | [setUTCFullYear](#) | [setUTCHours](#) |  
[setUTCMilliseconds](#) | [setUTCMilliseconds](#) | [setUTCMonth](#) | [setUTCSeconds](#)  
| | [setYear](#) | [toGMTString](#) | [toLocaleString](#) | [toUTCString](#) | [toString](#) |  
[valueOf](#) | [parse](#) | [UTC](#)

[1](#)

[new](#) | [var](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **Enumerator**

*enumObj* = **new Enumerator**([*collection*])

*enumObj*

### **Enumerator**

*collection*

### **Collection**

**Enumerator** VBScript

**For...Each**

**Enumerator :**

```
function ShowDriveList(){
    var fso, s, n, e, x;                //
    fso = new ActiveXObject("Scripting.FileSystemObject")
    e = new Enumerator(fso.Drives);      //      Eni
    s = "";
    for (; !e.atEnd(); e.moveNext())      //
    {
        x = e.item();
```

```
s = s + x.DriveLetter;
s += " - ";
if (x.DriveType == 3)          //
    n = x.ShareName;           //
else if (x.IsReady)           //
    n = x.VolumeName;          //
else
    n = "[      ]";
s += n + "<br>";
}
return(s);                     //
}
```

## Enumerator

[atEnd](#) | [item](#) | [moveFirst](#) | [moveNext](#)

[3](#)

[Drives](#) | [Files](#) | [Folders](#)

JScript

---

## Error

```
var newErrorObj = new Error(  
)  
var newErrorObj = new Error(  
    number  
)  
var newErrorObj = new Error(  
    number,  
    description  
)
```

### Error

number

description

Error      description      number

32 16

Error      throw                  Error

try...catch    Error

Error

```
try
{ x = y          // 
}catch(e){        //      e
    response.write(e)      //      "[object Error]".
    response.write(e.number & 0xFFFF) //      5009
    response.write(e.description) //      "'y' is undefined".
}
```

## Error

[description](#) | [number](#)

[5](#)

[new](#) | [throw](#) | [try...catch](#) | [var](#)

JScript

---

## **Function**

**1**

```
function functionName([argname1 [, ...[, argnameN]]]) {  
    body  
}
```

**2**

```
functionName = new Function( [argname1, [... argnameN,]] body );
```

*functionName*

argname1...argnameN

body

JScript

JScript 1 JScript

**Function** JScript 2

Fnction

1 JScript 2

**1**

```
function add(x, y)
{
    return(x + y);          //
}
```

**2**

```
var add = new Function("x", "y", "return(x+y);");
```

```
add(2, 3);
```

[arguments](#) | [caller](#) | [constructor](#) | [prototype](#)

[toString](#) | [valueOf](#)

[2](#)

[function](#) | [new](#) | [var](#)

---

[© 2000 Microsoft Corporation](#)

---

JScript

---

## **Global**

### **Global**

**Global**      **new** Scripting

[Infinity](#) | [NaN](#)

[escape](#) | [eval](#) | [isFinite](#) | [isNaN](#) | [parseFloat](#) | [parseInt](#) | [unescape](#)

[5](#)

[Object](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

# **Math**

**Math.**[{*property* | *method*}]

*property*

**Math**

*method*

**Math.**

**Math new**

[E](#) | [LN2](#) | [LN10](#) | [LOG2E](#) | [LOG10E](#) | [PI](#) | [SQRT1\\_2](#) | [SQRT2](#)

---

[abs](#) | [acos](#) | [asin](#) | [atan](#) | [atan2](#) | [ceil](#) | [cos](#) | [exp](#) | [floor](#) | [log](#)  
[max](#) | [min](#) | [pow](#) | [random](#) | [round](#) | [sin](#) | [sqrt](#) | [tan](#)

[1](#)

Number

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **Number**

*numObj* = **new Number**(*value*)

*numobj*

**Number**

*value*

Required. The numeric value of the **Number** object being created.

*value*

JScript    **Number**              **Number**

**Number**              **toString**

[MAX\\_VALUE](#) | [MIN\\_VALUE](#) | [NaN](#) | [NEGATIVE\\_INFINITY](#) |  
[POSITIVE\\_INFINITY](#) | [constructor](#) | [prototype](#)

[toLocaleString](#) | [toString](#) | [valueOf](#)

[1](#)

[Math](#) | [new](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **Object**

JScript

*obj* = **new Object**([*value*])

*obj*

**Object**

*value*

JScript      Number Boolean String value  
*value*    null undefined

**Object** JScript JScript

**toString**    Object

**Object** [JScript](#)

[prototype](#) | [constructor](#)

[toLocaleString](#) | [toString](#) | [valueOf](#)

[3](#)

[Function](#) | [Global](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **RegExp**

**RegExp.property**

*property*    **RegExp**

**RegExp**                      **RegExp**

index		-1
lastIndex		-1
lastMatch	\$&	
lastParen	\$+	
leftContext		
rightContext		
\$1 - \$9	\$1 - \$9	

[undefined](#)

**RegExp**                      **RegExp**

**RegExp**

```
function matchDemo(){
    var s;
    var re = new RegExp("d(b+)(d)","ig");
    var str = "cdbBdbsbdbdz";
    var arr = re.exec(str);
    s = "$1 contains: " + RegExp.$1 + "\n";
```

```
s += "$2 contains: " + RegExp.$2 + "\n";
s += "$3 contains: " + RegExp.$3;
return(s);
}
```

[\\$1...\\$9](#) | [index](#) | [input](#) | [lastIndex](#) | [lastMatch](#) | [lastParen](#) |  
[leftContext](#) | [rightContext](#)

**RegExp**

[3](#)

| . | [String](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

**1**

`re = /pattern/[flags]`

**2**

`re = new RegExp("pattern", ["flags"])`

*re*

*Pattern*

`1 "/" 2`

*flags*

`2 flag`

- g        *pattern*
- i
- m

**RegExp**

**RegExp**

**RegExp**

`1 2`

*pattern 1*      *pattern 2*      *pattern*      **compile**

(re)      **match**

```
function MatchDemo(){
    var r, re;          //
    var s = "The rain in Spain falls mainly in the plain";
    re = new RegExp("Spain","i"); //
    r = s.match(re);      //      s
    return(r);           //
}
```

[lastIndex](#) | [source](#)

[compile](#) | [exec](#) | [test](#)

[3](#)

[RegExp](#) | . | [String](#)

JScript

---

## **String**

*newString* = **new String**(["*stringLiteral*"])

*newString*

**String**

*stringLiteral*

Unicode

**String**      **String**      **new**    **String**

```
var alpha, beta;  
alpha = " ";  
beta = " ";
```

```
alpha.test = 10;
```

*beta test*

```
var gamma, delta;  
gamma = new String(" ");  
delta = new String(" ");
```

gamma.test = 10;

*delta*    *test*    **new String**    **String**    **String**

[constructor](#) | [length](#) | [prototype](#)

[anchor](#) | [big](#) | [blink](#) | [bold](#) | [charAt](#) | [charCodeAt](#) | [concat](#) | [fixed](#)  
\_\_\_\_\_| [fontcolor](#) | [fontsize](#) | [fromCharCode](#) | [indexOf](#) | [italics](#) |  
[lastIndexOf](#) | [link](#) | [match](#) | [replace](#) | [search](#) | [slice](#) | [small](#) | [split](#)  
\_\_\_\_\_| [strike](#) | [sub](#) | [substr](#) | [substring](#) | [sup](#) | [toLowerCase](#) |  
[toUpperCase](#) | [toString](#) | [valueOf](#)

[1](#)

[new](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **VBArray**

Visual Basic

*varName* = **new VBArray**(*safeArray*)

*varName*

VBArray

*safeArray*

**VBArray**

**VBArrays**

**VBArray**    *safeArray*    **VBArray** ActiveX

**VBArrays**

**dimensions**    **lbound**    **ubound**

Visual Basic VBScript VB JScript JScript  
HTML <HEAD> <BODY> JScript

<HEAD>

<SCRIPT LANGUAGE="VBScript">

<!--

Function CreateVBArray()

    Dim i, j, k

    Dim a(2, 2)

    k = 1

```
For i = 0 To 2
    For j = 0 To 2
        a(j, i) = k
        document.writeln(k)
        k = k + 1
    Next
    document.writeln("vbCRLF")
Next
CreateVBArray = a
End Function
-->
</SCRIPT>
```

```
<SCRIPT LANGUAGE="JScript">
<!--
function VBArrayTest(vbarray){
    var a = new VBArray(vbarray);
    var b = a.toArray();
    var i;
    for (i = 0; i < 9; i++)
    {
        document.writeln(b[i]);
    }
}
-->
</SCRIPT>
</HEAD>
```

```
<BODY>
<SCRIPT LANGUAGE="JScript">
<!--
    VBArrayTest(CreateVBArray());
-->
</SCRIPT>
</BODY>
```

## **VBArray**

[dimensions](#) | [getItem](#) | [lbound](#) | [toArray](#) | [ubound](#)

[3](#)

## [Array](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

# JScript

	<u>+</u>
	<u>=</u>
	<u>&amp;</u>
	<u>&lt;&lt;</u>
	<u>~</u>
	<u> </u>
	<u>&gt;&gt;</u>
	<u>^</u>
	<u>.</u>
Boolean	
	<u>?:</u>
	<u>--</u>
.	<u>delete</u>
	<u>/</u>
	<u>==</u>
	<u>&gt;</u>
	<u>&gt;=</u>
	<u>====</u>
	<u>++</u>
	<u>!=</u>
Boolean	<u>instanceof</u>
	<u>&lt;</u>
	<u>&lt;=</u>
	<u>&amp;&amp;</u>
	<u>!</u>
	<u>  </u>
	<u>%</u>
	<u>*</u>
	<u>new</u>
	<u>!==</u>
JScript	
	<u>-</u>
	<u>typeof</u>
	<u>_</u>
	<u>&gt;&gt;&gt;</u>
	<u>void</u>

JScript

---

## JScript

### JScript

.[]()	
++ -- - ~ ! delete new typeof void	
* / %	
+ - +	
<< >> >>>	
<<= > >= instanceof	instanceof
== != === !==	
&	
^	
&&	
?:	
= oP=	
,	

$$z = 78 * (96 + 3 + 45)$$

=, \*, (), +, +(), +, +, \*, =

1. 96 3 45 144

2. 78 144 11232

A 11232 z

JScript

---

	+
	--
	/
	++
	%
	*
	-
	--

	,
	?:
	==
	>
	>=
	====
	!=
	<
	<=
	&&
	!
	!==

	&
	<<
	~

	>>
	^
	>>>

[delete](#)

[instanceof](#)

[new](#)

[typeof](#)

[void](#)

[1](#)

JScript

---

( $+=$ )

result  $+=$  expression

result

expression

result = result + expression

$+=$

Boolean	

1

+ | . | .

---

[© 2000 Microsoft Corporation](#)

JScript

---

(+)

result = expression1 + expression2

result

expression1

expression2

+

Boolean	

1

+= | . | .

---

[© 2000 Microsoft Corporation](#)

---

JScript

---

(=)

result = expression

result

expression

=

j = k = l = 0;

j k l

1

| .

JScript

---

**“” (&=)**

“”

result &= expression

result

expression

:

result = result & expression

&= *result expression , “”*

0101 (result)

1100 (expression)

----

0100

1 1 0

1

& | .| .

---

[© 2000 Microsoft Corporation](#)

JScript

---

“” (&)

“”

result = expression1 & expression2

result

expression1

expression2

& “”

0101 (expression1)  
1100 (expression2)

----

0100 (result)

1 1 0

1

&= | .| .

---

[© 2000 Microsoft Corporation](#)

JScript

---

(<<)

result = expression1 << expression2

result

expression1

expression2

<< expression1 expression2

var temp

temp = 14 << 2

temp 56 14 00001110 56 00111000

1

<<= | >> | >>> | . | .

---

[© 2000 Microsoft Corporation](#)

---

JScript

---

“” (~)

“”

result = ~ expression

result

expression

~

- undefined null
- 
- 
- Boolean false 0 true 1

~

0101 (expression)

----

1010 (result)

1 0 0 1

1

! | . | .

---

[© 2000 Microsoft Corporation](#)

JScript

---

“” (|=)

“”

result |= expression

result

expression

result = result | expression

|= *result* *expression* “”

0101 (result)

1100 (expression)

----

1101 ( )

1 1 0

1

|||.

---

[© 2000 Microsoft Corporation](#)

JScript

---

“” ()

“”

result = expression1 | expression2

result

expression1

expression2

| “”

0101 (expression1)

1100 (expression2)

----

1101 ( )

1 1 0

1

| . | .

---

[© 2000 Microsoft Corporation](#)

JScript

---

(>>)

result = expression1 >> expression2

result

expression1

expression2

>> expression1 expression2 expression1 temp  
-4-14 11110010 -4 11111100

var temp  
temp = -14 >> 2

1

<< | >>= | >>> | . | .

---

[© 2000 Microsoft Corporation](#)

---

JScript

---

**“” ( $\wedge=$ )**

result  $\wedge=$  expression

result

expression

$\wedge=$

result = result  $\wedge$  expression

$\wedge=$

0101 (result)  
1100 (expression)

----

1001

1 1 0

1

Λ | . | .

---

[© 2000 Microsoft Corporation](#)

JScript

---

“” (^)

result = expression1 ^ expression2

result

expression1

expression2

^

0101 (expression1)

1100 (expression2)

----

1001 ( )

1 1 0

1

^= | . | .

---

[© 2000 Microsoft Corporation](#)

JScript

---

(,)

expression1, expression2

expression1

expression2

, , **for**

```
for (i = 0; i < 10; i++, j++)
{
    k = i + j;
}
```

**for** ,

1

for | . | .

---

[© 2000 Microsoft Corporation](#)

---

JScript

---

Boolean

expression1 **comparisonoperator** expression2

expression1

comparisonoperator

expression2

JScript Unicode

*expression1    expression2*

<><=>=

- expression1 expression2
- 
- **NaN    false**
- 
- 
- 

==!=

- Boolean
- **NaN**

- 
- null null undefined
- Boolean
- 

====!==

1

| .

---

[© 2000 Microsoft Corporation](#)

JScript

---

	<code>+=</code>
	<code>&amp;=</code>
	<code> =</code>
	<code>^=</code>
	<code>/=</code>
	<code>&lt;&lt;=</code>
	<code>%=</code>
	<code>*=</code>
	<code>&gt;&gt;=</code>
	<code>-=</code>
	<code>&gt;&gt;&gt;=</code>

JScript

---

(?:)

test ? 1 : 2

test

Boolean

1

*test*    **true**

2

*test*    **false**

?:    **if...else**                **if...else**

var now = new Date();

var greeting = "Good" + ((now.getHours() > 17) ? " eveni

6pm "Good evening."                **if...else**

var now = new Date();

var greeting = "Good";

if (now.getHours() > 17)

    greeting += " evening.;"

else

    greeting += " day.;"

1

if...else | .| .

---

© 2000 Microsoft Corporation

JScript

---

## **delete**

**delete** expression

*expression* JScript

*expression*      *expression*      **false**

**true**

[3](#)

JScript

---

(/=)

result /= expression

result

expression

/=

result = result / expression

1

L | . | .

JScript

---

(/)

result = number1 / number2

result

number1

number2

1

/ = | . | .

---

[© 2000 Microsoft Corporation](#)

JScript

---

**in**

*result = property in object*

*result*

*property*

*object*

**in** property

1

JScript

---

(++) (--)

**1**

```
result = ++variable  
result = --variable  
result = variable++  
result = variable--
```

**2**

```
++variable  
--variable  
variable++  
variable--
```

result

variable

```
var j, k;  
k = 2;  
j = ++k;
```

3            *j*

```
var j, k;  
k = 2;  
j = k++;
```

2 *j*

1

| .

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **instanceof**

Boolean

result = object **instanceof** class

result

object

class

*object    class              instanceof    true    object              object    null*  
**false**

## **instanceof**

```
function objTest(obj){  
    var i, t, s = ""; //  
    t = new Array(); //  
    t["Date"] = Date; //  
    t["Object"] = Object;  
    t["Array"] = Array;
```

```
for (i in t)
{
    if (obj instanceof t[i]) //      obj
    {
        s += "obj is an instance of " + i + "\n";
    }
    else
    {
        s += "obj is not an instance of " + i + "\n";
    }
}
return(s); //
```

```
var obj = new Date();
response.write(objTest(obj));
```

5

| .

JScript

---

(<<=)

result <<= expression

result

expression

<<=

result = result << expression

<<= result expression

var temp

temp = 14

temp <<= 2

temp 56 14 00001110 56 00111000

1

[<<](#) | [>>](#) | [>>>](#) | [.|.](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## “” (&&)

result = expression1 && expression2

result

expression1

expression2

True	<i>result</i>	True	False	<i>result</i>	False
------	---------------	------	-------	---------------	-------

JScript Boolean Boolean

- true
- false
- **null undefined** false
- false

[1](#)

| .

---

[© 2000 Microsoft Corporation](#)

JScript

---

“” (!)

result = !expression

result

expression

*result*

expression	result
True	False
False	True

!

- undefined   null
- 
- 
- Boolean false 0 true 1

!        *expression*        *result*        *expression*        *result* 1

1

z | . | .

---

[© 2000 Microsoft Corporation](#)

JScript

---

“” (||)

“”

result = expression1 || expression2

result

expression1

expression2

**True**   *result*   **True**   *result*

expression1	expression2	result
True	True	True
True	False	True
False	True	True
False	False	False

JScript Boolean Boolean

- true
- false
- **null**   undefined   false
- 0   false

1

| .

---

[© 2000 Microsoft Corporation](#)

JScript

---

(%=)

result % = expression

result

expression

%=

result = result % expression

1

% | . | .

JScript

---

(%)

result = number1 % number2

result

number1

number2

*number1      number2*

*resultA*

*result 5*

A = 19 % 6.7

1

%= | . | .

JScript

---

(\*=)

result \*= expression

result

expression

\*=

result = result \* expression

1

\* | ..| ..

JScript

---

(\*)

result = number1\*number2

result

number1

number2

1

\*= | . | .

---

[© 2000 Microsoft Corporation](#)

JScript

---

**new**

**new constructor[(arguments)]**

*constructor*

*arguments*

**new**

- 
- **this**
- 

**new**

```
my_object = new Object;  
my_array = new Array();  
my_date = new Date("Jan 5 1996");
```

1

function

---

[© 2000 Microsoft Corporation](#)

JScript

---

( $>>=$ )

result  $>>=$  expression

result

expression

$>>=$

result = result  $>>$  expression

$>>= \quad result \quad expression \quad result \quad temp \ -4-14$   
11110010 -4 11111100

var temp  
temp = -14  
temp  $>>=$  2

1

[<<](#) | [>>](#) | [>>>](#) | [.](#) | [.](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

(-=)

result -= expression

result

expression

-=

result = result - expression

1

- | . | .

JScript

---

(-)

1

result = number1 - number2

2

-number

result

number

number1

number2

1            - 2                    -

2

- undefined    **null**
-

- Boolean false 0 true 1

2                  *result*                  *result*

1

--|..|..

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **typeof**

**typeof[()expression[]] ;**

*expression* .

**typeof**    **typeof** "number," "string," "boolean," "object," "function,"  
"undefined."

**typeof**

[1](#)

| .

---

[© 2000 Microsoft Corporation](#)

JScript

---

(>>>)

result = expression1 >>> expression2

result

expression1

expression2

>>> *expression1 expression2*

var temp

temp = -14 >>> 2

*temp* -14 11111111 11111111 11111111 11110010  
1073741820 00111111 11111111 11111111 11111100

1

>>>= | << | >> | . | .

---

[© 2000 Microsoft Corporation](#)

JScript

---

(>>>=)

result >>>= expression

result

expression

>>>=

result = result >>> expression

>>>= *result* *expression*

var temp

temp = -14

temp >>>= 2

*temp* -14 11111111 11111111 11111111 11110010  
1073741820 00111111 11111111 11111111 11111100

1

[>>>](#) | [<<](#) | [>>](#) | [.](#) | [.](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

**void**

**void** expression

*expression* JScript .

**void**      [undefined](#)

[2](#)

| .

---

[© 2000 Microsoft Corporation](#)

JScript

---

# JScript

	<a href="#">\$1...\$9 Properties</a>
	<a href="#">arguments</a>
	<a href="#">caller</a>
	<a href="#">constructor</a>
	<a href="#">description</a>
Euler	<a href="#">E</a>
	<a href="#">index</a>
<b>Number.POSITIVE_INFINITY</b>	<a href="#">Infinity</a>
	<a href="#">input</a>
	<a href="#">lastIndex</a>
1	<a href="#">length Array</a>
	<a href="#">length Function</a>
<b>String</b>	<a href="#">length String</a>
2	<a href="#">LN2</a>
10	<a href="#">LN10</a>
2 e Euler	<a href="#">LOG2E</a>
10 e Euler	<a href="#">LOG10E</a>
JScript	<a href="#">MAX_VALUE</a>
JScript	<a href="#">MIN_VALUE</a>
<b>NaN</b>	<a href="#">NaN Global</a>
<b>NaN</b>	<a href="#">NaN Number</a>
JScript -Number.MAX_VALUE	<a href="#">NEGATIVE_INFINITY</a>
	<a href="#">number</a>
3.141592653589793	<a href="#">PI</a>
JScript Number.MAX_VALUE	<a href="#">POSITIVE_INFINITY</a>
	<a href="#">prototype</a>
	<a href="#">source</a>
0.5 1 2	<a href="#">SQRT1_2</a>
2	<a href="#">SQRT2</a>

JScript

---

**\$1...\$9**

**RegExp.\$n**

**RegExp**

**RegExp**

n

1 9.

**\$1...\$9**

**\$1...\$9**

```
function matchDemo(){
    var s;
    var re = new RegExp("d(b+)(d)","ig");
    var str = "cdbBdbbdbdz";
    var arr = re.exec(str);
    s = "$1 contains: " + RegExp.$1 + "\n";
    s += "$2 contains: " + RegExp.$2 + "\n";
    s += "$3 contains: " + RegExp.$3;
    return(s);
}
```

1

[RegExp](#) | .

[RegExp](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **arguments**

### **function arguments**

*function.arguments*

function

### **arguments**

### **arguments length arguments**

### **arguments**

```
function ArgTest(){
    var i, s, numargs = arguments.length;
    s = numargs;
    if (numargs < 2)
        s += " argument was passed to ArgTest. It was ";
    else
        s += " arguments were passed to ArgTest. They were "
    for (i = 0; i < numargs; i++)
    {
        s += arguments[i] + " ";
    }
    return(s);
}
```

2

[function](#) | [length](#) ([Array](#))

[Function](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **caller**

*functionName*.**caller**

*functionName*

**caller** JScript                   **caller** null

**caller**       *functionName*.**toString**

**caller**

```
function CallLevel(){
    if (CallLevel.caller == null)
        return("CallLevel was called from the top level.");
    else
        return("CallLevel was called by another function.");
}
```

2

[function](#)

[Function](#)

JScript

---

## **constructor**

*object*.**constructor**

*object*

**constructor** prototype

Global   Math   [JScript](#) constructor

```
x = new String("Hi");
if (x.constructor == String)
    //
```

```
function MyFunc {
    //
}
```

```
y = new MyFunc;
if (y.constructor == MyFunc)
    //
```

[prototype](#)

[Array](#) | [Boolean](#) | [Date](#) | [Function](#) | [Math](#) | [Number](#) | [Object](#)  
| [String](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **description**

*object.description [= stringExpression]*

### **description**

object

Error

stringExpression

## **description**

### **description**

try

  x = y //

catch(var e){ //               e

  document.write(e) //    "[object Error]".

  document.write((e.number & 0xFFFF)) //    5009.

  document.write(**e.description**) //    "'y' is undefined".

}

number

Error

---

© 2000 Microsoft Corporation

JScript

---

**E**

Euler              E 2.718

*numVar = Math.E*

1

[exp](#) | [Math](#)

[Math](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **global**

Boolean global (g)                   **false**

### *rgExp.global*

*rgExp*

global     **global**    true    false

global

**global** "g" "the" "a" "The"  
"the" "t"

**gi    m**

```
function RegExpPropDemo(flag){  
    if (flag.match(/[^gim]/))        //  
        return("Flag specified is not valid");  
    var r, re, s                    //  
    var ss = "The man hit the ball with the bat.\n";  
    ss += "while the fielder caught the ball with the glove.";  
    re = new RegExp("the",flag);    //  
    r = ss.replace(re, "a");        // "a"      "the"  
    s = "Regular Expression property values:\n\n"  
    s += "global ignoreCase multiline\n"  
    if (re.global)                //    global
```

```
s += " True ";
else
    s += "False ";
if (re.ignoreCase)          // ignoreCase
    s += " True ";
else
    s += "False ";
if (re.multiline)           // multiline
    s += "  True ";
else
    s += "  False ";
s += "\n\nThe resulting string is:\n\n" + r;
return(s);                  //
}
```

## 5.5

[ignoreCase](#) | [multiline](#) | .

[RegExp](#)

JScript

---

## **hasOwnProperty**

*object*.**hasOwnProperty**(*proName*)

*object*

*proName*

<i>object</i>	<b>hasOwnProperty</b>	true	false
---------------	-----------------------	------	-------

<b>String</b>	split	false	true
---------------	-------	-------	------

```
var s = new String("JScript");
print(s.hasOwnProperty("split"));
print(String.prototype.hasOwnProperty("split"));
```

[5.5](#)

in

## Object

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **ignoreCase**

Boolean ignoreCase (        i)        **false**

*rgExp.ignoreCase*

*rgExp*    **RegExp**

ignoreCase        **ignoreCase**    true    false

ignoreCase

**ignoreCase** "i" "the" "a" "The" ignoreCase  
"T" "t"

**gi    m**

```
function RegExpPropDemo(flag){  
    if (flag.match(/[^gim]/))        //  
        return("Flag specified is not valid");  
    var r, re, s                      //  
    var ss = "The man hit the ball with the bat.\n";  
    ss += "while the fielder caught the ball with the glove.";  
    re = new RegExp("the",flag);     //  
    r = ss.replace(re, "a");        //    "a"    "the"  
    s = "Regular Expression property values:\n\n"  
    s += "global ignoreCase multiline\n"  
    if (re.global)                  //    global
```

```
s += " True ";
else
    s += "False ";
if (re.ignoreCase)          // ignoreCase
    s += " True ";
else
    s += "False ";
if (re.multiline)           // multiline
    s += "  True ";
else
    s += "  False ";
s += "\n\nThe resulting string is:\n\n" + r;
return(s);                  //
}
```

## 5.5

[global](#) | [multiline](#) | .

[RegExp](#)

JScript

---

**index**

**RegExp.index**

**RegExp**

**index -1**

**index            index    lastIndex**

```
function RegExpTest(){
    var ver = Number(EngineMajorVersion() + "." + ScriptEngineMinorVersion());
    if (ver >= 5.5){
        var src = "The rain in Spain falls mainly in the plain.";
        var re = /\w+/g;
        var arr;
        while ((arr = re.exec(src)) != null)
            print(arr.index + "-" + arr.lastIndex + "\t" + arr);
    }
    else{
        alert("You need a newer version of JScript for this to work.");
    }
}
```

3

RegExp | .

RegExp

---

© 2000 Microsoft Corporation

JScript

---

## **Infinity**

Number.POSITIVE\_INFINITY

Infinity

**Infinity    Global** Scripting

[3](#)

[POSITIVE\\_INFINITY](#) | [NEGATIVE\\_INFINITY](#)

[Global](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

**input (\$\_)**

**RegExp.input**

**RegExp**

**input**

**input**

```
function inputDemo(){
    var s;
    var re = new RegExp("d(b+)(d)","ig");
    var str = "cdbBdbbdbdz";
    var arr = re.exec(str);
    s = "The string used for the match was " + RegExp.inp
    return(s);
}
```

3

[RegExp](#) | .

[RegExp](#)

---

[© 2000 Microsoft Corporation](#)

---

JScript

---

## **isPrototypeOf**

*object1.isPrototypeOf(object2)*

object1

object2

<i>object2</i>	<i>object1</i>	<b>isPrototypeOf</b>	<b>true</b>	<i>object2</i>	<i>object1</i>
<i>object2</i>		<b>isPrototypeOf</b>	<b>false</b>		

## **isPrototypeOf**

```
function test(){
    var re = new RegExp(); // 
    return (RegExp.prototype.isPrototypeOf(re)); // true
}
```

## [Object](#)

JScript

---

## **lastIndex**

RegExp.lastIndex

### **RegExp**

#### **lastIndex -1**

lastIndex RegExp exec test String matchreplace  
split

#### **lastIndex**

- lastIndex -1.
- **lastIndex**      **test**    **exec**      **lastIndex -1**
- **lastIndex**                 **lastIndex -1**
- **lastIndex**

#### **lastIndex                 index    lastIndex**

```
function RegExpTest(){
    var ver = Number(ScriptEngineMajorVersion() + "." + ScriptEngineMinorVersion());
    if (ver >= 5.5){
        var src = "The rain in Spain falls mainly in the plain.";
        var re = /\w+/g;
        var arr;
        while ((arr = re.exec(src)) != null)
            print(arr.index + "-" + arr.lastIndex + "\t" + arr);
```

```
}
```

```
else{
```

```
    alert("You need a newer version of JScript for this to work")
```

```
}
```

```
}
```

3

[RegExp](#) | .

[RegExp](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **length (Array)**

1

*numVar = arrayObj.length*

numVar

arrayObj

**Array**

**length** my\_array.length 7 2

```
var my_array = new Array( );
my_array[0] = "Test";
my_array[6] = "Another Test";
```

**length**              **length**

**length**              [undefined](#)

**length :**

```
function LengthDemo(){
    var a;
    a = new Array(0,1,2,3,4);
    return(a.length);
}
```

2

[length \(Function\)](#) | [length \(String\)](#)

[Array](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **length (Function)**

*functionName.length*

*functionName*

**length** Scripting

**length**

```
function ArgTest(a, b){  
    var i, s = "The ArgTest function expected ";  
    var numargs = ArgTest.arguments.length;  
    var expargs = ArgTest.length;  
    if (expargs < 2)  
        s += expargs + " argument. ";  
    else  
        s += expargs + " arguments. ";  
    if (numargs < 2)  
        s += numargs + " was passed.";  
    else  
        s += numargs + " were passed.";  
    return(s);  
}
```

2

[arguments](#) | [length \(Array\)](#) | [length \(String\)](#)

[Function](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **length (String)**

**String**

*strVariable.length "String Literal".length*

**length**      **String**      **String**      **length - 1**

[1](#)

[length Array](#) | [length Function](#) | [String](#) | [String](#)

[String](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

**LN10**

10

*numVar = Math.LN10*

**LN10** 2.302

1

Math

Math

---

© 2000 Microsoft Corporation

JScript

---

**LN2**

2

*numVar = Math.LN2*

**LN2** 0.693

1

Math

Math

---

© 2000 Microsoft Corporation

JScript

---

## **LOG10E**

$10 \quad e$

*varName* = **Math.LOG10E**

**LOG10E** 0.434

[1](#)

[Math](#)

[Math](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **LOG2E**

2         $e$

*varName* = **Math.LOG2E**

**LOG2E** 1.442

[1](#)

[Math](#)

[Math](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **MAX\_VALUE**

JScript 1.79E+308

number.MAX\_VALUE

*number* **Number** .

## **MAX\_VALUE Number**

[2](#)

[MIN\\_VALUE](#) | [NaN](#) | [NEGATIVE\\_INFINITY](#) | [POSITIVE\\_INFINITY](#)  
[\\_\\_](#) | [toString](#)

[Number](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

**message**

*errorObj.message*

*errorObj*

**Error**

<b>message</b>	<b>description</b>
----------------	--------------------

TypeError

```
try {  
    // 'null'  
    null.doSomething();  
}  
catch(e){  
    print(e.name + ": " + e.message);  
    print(e.number + ": " + e.description);  
}
```

[description](#) | [name](#)

[Error](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **MIN\_VALUE**

JScript 2.22E-308

*number*.**MIN\_VALUE**

*number* **Number** .

**MIN\_VALUE** **Number**

[2](#)

[MAX\\_VALUE](#) | [NaN](#) | [NEGATIVE\\_INFINITY](#) | [POSITIVE\\_INFINITY](#)  
[\\_\\_](#) | [toString](#)

[Number](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **multiline**

Boolean **multiline** (        **m**)        **false**

*rgExp*.**multiline**

*rgExp*    **RegExp**

**multiline**        **multiline**    **true**    **false**    **m**        **multiline**    **true**

**multiline**    **false** "^\\$"  
    "\\r"    "\$"    "\\n"    "\\r"

**multiline** "m"    "while"    "and"    **multiline** "while"  
**multiline**

**gi**    **m**

```
function RegExpPropDemo(flag){  
    if (flag.match(/[^gim]/))        //  
        return("Flag specified is not valid");  
    var r, re, s                    //  
    var ss = "The man hit the ball with the bat.";  
    ss += "\\nwhile the fielder caught the ball with the glove.  
    re = new RegExp("^while",flag);    //  
    r = ss.replace(re, "and");        // "a"    "the"  
    s = "Regular Expression property values:\\n\\n"  
    s += "global ignoreCase multiline\\n"  
    if (re.global)                  //    global
```

```
s += " True ";
else
    s += "False ";
if (re.ignoreCase)           // ignoreCase
    s += " True ";
else
    s += "False ";
if (re.multiline)           // multiline
    s += "  True ";
else
    s += "  False ";
s += "\n\nThe resulting string is:\n\n" + r;
return(s);                  //
}
```

## 5.5

[global](#) | [ignoreCase](#) | .

[RegExp](#)

JScript

---

## **name**

*errorObj.name*

*errorObj*

## **Error**

### **name**

ConversionError	
RangeError	<b>Array</b>
ReferenceError	<b>null</b>
RegExpError	<b>igm</b>
SyntaxError	<b>eval</b>
TypeError	
URIError	(URI)

TypeError

```
try {
  // 'null'
  null.doSomething();
}
catch(e){
  print(e.name + ":" + e.message);
```

```
    print(e.number + ":" + e.description);  
}
```

## 5.5

[description](#) | [message](#) | [number](#)

[Error](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

**NaN**

number.**NaN**

*number* **Number** .

**NaN    Number**

**NaN**              **NaN    isNaN**

[2](#)

[isNaN](#) | [MAX\\_VALUE](#) | [MIN\\_VALUE](#) | [NEGATIVE\\_INFINITY](#) |  
[POSITIVE\\_INFINITY](#) | [toString](#)

[Number](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **NaN (Global)**

**NaN**

NaN

**NaN**

**Global** Scripting

[3](#)

[isNaN](#)

[Global](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **NEGATIVE\_INFINITY**

JScript    **-Number.MAX\_VALUE**

*number.NEGATIVE\_INFINITY*

*number*    **Number** .

NEGATIVE\_INFINITY Number

JScript    **NEGATIVE\_INFINITY** -infinity

[2](#)

[MAX\\_VALUE](#) | [MIN\\_VALUE](#) | [NaN](#) | [POSITIVE\\_INFINITY](#) |  
[toString](#)

[Number](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

**number**

**Error number**

*object.number [= errorNumber]*

object

**Error**

errorNumber

32 16

**number**

try

  x = y //

  catch(var e){ //              e

    document.write(e) //      "[object Error]".

    document.write(**e.number>>16 & 0xFFFF**) //

    document.write(**e.number & 0xFFFF**) //              5009

    document.write(e.description) //      '"y' is undefined".

}

[description](#) | [message](#) | [name](#)

[Error](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

**PI**

3.141592653589793

*numVar = Math.PI*

**PI** 3.14159.

[1](#)

[Math](#)

[Math](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **POSITIVE\_INFINITY**

JScript (**Number.MAX\_VALUE**)

### **Number.POSITIVE\_INFINITY**

*number* **Number** .

POSITIVE\_INFINITY Number

JScript **POSITIVE\_INFINITY** infinity

[2](#)

[MAX\\_VALUE](#) | [MIN\\_VALUE](#) | [NaN](#) | [NEGATIVE\\_INFINITY](#) |  
[toString](#)

[Number](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **propertyIsEnumerable**

Boolean

*object.propertyIsEnumerable(proName)*

*object*

*proName*

<i>proName</i>	<i>object</i>	<b>For...In</b>	<b>propertyIsEnumerable</b>	true
	<i>object</i>		<b>propertyIsEnumerable</b>	false

## **propertyIsEnumerable**

```
function testIsEnumerable(){
    var a = new Array("apple", "banana", "cactus");
    return(a.propertyIsEnumerable(1));
}
```

## Object

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **prototype**

*objectName.prototype*

*objectName*

**prototype** “”

**Array**

**Array.prototype**

```
function array_max( ){
    var i, max = this[0];
    for (i = 1; i < this.length; i++)
    {
        if (max < this[i])
            max = this[i];
    }
    return max;
}
Array.prototype.max = array_max;
var x = new Array(1, 2, 3, 4, 5, 6);
var y = x.max( );
```

*y*    *x* 6

[JScript](#) **prototype**

2

[constructor](#)

[Array](#) | [Boolean](#) | [Date](#) | [Function](#) | [Number](#) | [Object](#) | [String](#)

---

---

[© 2000 Microsoft Corporation](#)

JScript

---

**source**

*rgExp*.**source**

*rgExp*

**source**

```
function SourceDemo(re, s){  
    var s1;  
    //  
    if (re.test(s))  
        s1 = " contains ";  
    else  
        s1 = " does not contain ";  
    //  
    return(s + s1 + re.source);  
}
```

3

| | .

JScript

---

**SQRT1\_2**

0.5 2

*numVar = Math.SQRT1\_2*

**SQRT1\_2** 0.707.

[1](#)

[Math](#) | [sqrt](#) | [SQRT2](#)

[Math](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

**SQRT2**

2

*numVar = Math.SQRT2*

**SQRT2** 1.414.

[1](#)

[Math](#) | [sqrt](#) | [SQRT1\\_2](#)

[Math](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

**undefined**

**undefined**

undefined

**undefined Global**

**undefined**

**undefined "undefined"**

undefined      **undefined**

```
var declared; //  
if (declared == undefined) //  
    document.write("declared has not been given a value.");  
  
if (typeOf(notDeclared) == "undefined")  
    document.write("notDeclared has not been defined.");
```

[5.5](#)

[Global Object](#)

JScript

---

## JScript

<i>label</i>	<a href="#">break</a>
<b>try</b>	<a href="#">catch</a>
	<a href="#">@cc_on</a>
JScript	<a href="#">//</a>
JScript	<a href="#">/*..*/</a>
	<a href="#">continue</a>
<b>false</b>	<a href="#">do...while</a>
<b>true</b>	<a href="#">for</a>
	<a href="#">for...in</a>
	<a href="#">function</a>
	<a href="#">@if</a>
	<a href="#">if...else</a>
	<a href="#">Labeled</a>
	<a href="#">return</a>
	<a href="#">@set</a>
	<a href="#">switch</a>
	<a href="#">this</a>
<b>try...catch</b>	<a href="#">throw</a>
JScript	<a href="#">try</a>
	<a href="#">var</a>
<b>false</b>	<a href="#">while</a>
	<a href="#">with</a>

JScript

---

**@cc\_on**

**@cc\_on**

**@cc\_on** scripting

**@cc\_on**

**/\*@cc\_on\*/**

...

**@if**    **@set**

3

| . | [@if](#) | [@set](#)

JScript

---

**@if**

```
@if (
    condition1
)
    text1
[@elif (
    condition2
)
    text2]
[@else
    text3]
@end
```

condition1, condition2

[Boolean](#)

text1

*condition1*   **true**

text2

*condition1*   **false**    *condition2*   **true**

text3

*condition1*   *condition2*    **false**

**@if**               **@elif**               **@elif**   **@else**  
**@if**

alert(@if (@\_win32) "using Windows NT or Windows 95

3

| . | [@cc\\_on](#) | [@set](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

**@set**

**@set** @varname = *term*

varname

JScript "@"

term

Boolean                   **@set** JScript

**@set** @myvar1 = 12

**@set** @myvar2 = (@myvar1 \* 20)

**@set** @myvar3 = @\_jscript\_version

- ! ~
- \* / %
- + -
- << >> >>>
- < <= > >=
- == != === !==

- & ^ |
- && ||

**NaNNaN @if**

@if (@newVar != @newVar)

...

**NaN**

3

| . | @cc\_on | @if

---

© 2000 Microsoft Corporation

JScript

---

## break

*label*

**break** [*label*];

*label*

**break**

**break**

```
function BreakTest(breakpoint){  
    var i = 0;  
    while (i < 100)  
    {  
        if (i == breakpoint)  
            break;  
        i++;  
    }  
    return(i);  
}
```

[1](#)

[continue](#) | [do...while](#) | [for](#) | [for...in](#) | [Labeled](#) | [while](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **Comment**

JScript

**1**

// *comment*

**2**

```
/*
comment
*/
```

*comment*

**3**

//@CondStatement

**4**

```
/*@
condStatement
@*/
```

*condStatement* 3 //" "@"

JScript

1 2

3 4 1 2

**comment**

```
function myfunction(arg1, arg2){  
    /*  
     *  
     var r;  
     //  
     r = arg1 + arg2; //  
     return(r);  
}
```

[1](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **continue**

**continue** [*label*];

*label*    **continue**

**whiledo...whilefor    for..in    continue                      continue**

- **while    do...while true**
- **for true**
- **for...in**

## **continue**

```
function skip5(){  
    var s = "", i=0;  
    while (i < 10)  
    {  
        i++;  
        //      5  
        if (i==5)  
        {  
            continue;  
        }  
        s += i;  
    }
```

```
    return(s);  
}
```

[1](#)

[break](#) | [do...while](#) | [for](#) | [for..in](#) | [Labeled](#) | [while](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **do...while**

**false**

do statement  
**while** (*expression*) ;

statement

*expression*    **true**

*expression*

Boolean        **true**    **false**        .    *expression*    **true**        *expression*    **false**

*expression*

## **do...while Drives**

```
function GetDriveList(){
    var fso, s, n, e, x;
    fso = new ActiveXObject("Scripting.FileSystemObject"
    e = new Enumerator(fso.Drives);
    s = "";
    do
    {
        x = e.item();
```

```
s = s + x.DriveLetter;
s += " - ";
if (x.DriveType == 3)
    n = x.ShareName;
else if (x.IsReady)
    n = x.VolumeName;
else
    n = "[Drive not ready]";
    s += n + "<br>";
e.moveToNext();
}
while (!e.atEnd());
return(s);
}
```

[3](#)

[break](#) | [continue](#) | [for](#) | [for...in](#) | [while](#) | [Labeled](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

**for**

true

**for** (*initialization; test; increment*)  
*statements*

initialization

test

Boolean    *test*    **true**    *statement*              *test*    **false**

increment

statements

*test*    **true**

**for**

**for**

```
/*           i       0       1
             i       10      */
var myarray = new Array();
```

```
for (i = 0; i < 10; i++) {  
    myarray[i] = i;  
}
```

[1](#)

[for...in](#) | [while](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **for...in**

**for** (*variable* **in** [*object* | *array*])  
*statements*

variable

*object*    *array*

*object*, *array*

statement

*object*    *array*

*variable*    *object*    *array*                    *object*    *array*

*variable* 012.....

**for ... in**

```
function ForInDemo(){  
  //  
  var a, key, s = "";  
  //  
  a = {"a" : "Athens" , "b" : "Belgrade", "c" : "Cairo"}
```

```
//  
for (key in a) {  
    s += a[key] + "<BR>";  
}  
return(s);  
}
```

[5](#)

## **enumerator**

[for | while](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

**function**

```
function functionname([arg1 [, arg2 [,...[, argN]]]]) {  
    statements  
}
```

*functionname*

*arg1...argN*

*statements*

JScript

**function**            *statements*

**function**

```
function myfunction(arg1, arg2) {  
    var r;  
    r = arg1 * arg2;  
    return(r);  
}
```

1

new

---

© 2000 Microsoft Corporation

JScript

---

## **if...else**

```
if (condition)
    statement1
[else
    statement2]
```

condition

[Boolean](#)    *condition*    [null undefined](#)    *condition*    **false**

statement1

*condition*    **true**

statement2

*condition*    **false**

*statement1*    *statement2* ({})

**else**    **if**

```
if (x == 5)
    if (y == 6)
        z = 17;
else
    z = 20;
```

```
if (x == 5)
{
    if (y == 6)
        z = 17;
}
else
    z = 20;
```

*statement1*

```
if (x == 5)
    z = 7;
    q = 42;
else
    z = 19;
```

if    else       if    else

1

(?:)

JScript

---

## **Labeled**

*label* : *statements*

label

statement

*label*

**break**   **continue**        **break**   **continue**

**continue**   **labeled**

```
function labelDemo(){
    var a = new Array();
    var i, j, s = "", s1 = "";
    Outer:
    for (i = 0; i < 5; i++)
    {
        Inner:
        for (j = 0; j < 5; j++)
        {
```

```
if (j == 2)
    continue Inner;
else
    a[i,j] = j + 1;
}
}
for (i = 0;i < 5; i++)
{
    s = ""
    for (j = 0; j < 5; j++)
    {
        s += a[i,j];
    }
    s1 += s + "\n";
}
return(s1)
}
```

3

[break](#) | [continue](#)

JScript

---

**return**

**return**[*()*[*expression*][*]*];

*expression*

**return**      *expression*      *expression*      **return**  
undefined.

**return**

```
function myfunction(arg1, arg2){  
    var r;  
    r = arg1 * arg2;  
    return(r);  
}
```

1

function

JScript

---

## **switch**

```
switch (expression) {   case label :  
    statementlist  
case label :  
    statementlist  
...  
default :  
    statementlist  
}
```

*expression*

*label*

*expression*      *label* === *expression*    *statementlist*      **break**  
**switch**

*statementlist*

**default**      *expression*      **switch**

*label*      *label*    *expression*      **default**

**switch**

- *expression*    *label*
- *label*    *expression*      *statementlist*  
            **break**      **switch**      **break**      *label*

- *label*      *expression*      **default**      **default**
- **switch**

```
function MyObject() {  
...}
```

```
switch (object.constructor){  
case Date:  
...  
case Number:  
...  
case String:  
...  
case MyObject:  
...  
default:  
...  
}
```

[3](#)

[break](#) | [if...else](#)

---

[© 2000 Microsoft Corporation](#)

---

JScript

---

**this**

**this.property**

property

**this**

**this** Car

```
function Car(color, make, model){  
    this.color = color;  
    this.make = make;  
    this.model = model;  
}
```

JScript            **this**    **window**

[1](#)

[new](#)

JScript

---

**throw**

**try...catch...finally**

**throw** exception

*exception*

**try...catch...finally**

```
function TryCatchDemo(x){  
    try {  
        try {  
            if (x == 0) //  
                throw "x equals zero"; //  
            else  
                throw "x does not equal zero"; //  
        }  
        catch(e) { // "x = 0"  
            if (e == "x equals zero") //  
                return(e + " handled locally."); //  
            else //  
                throw e; //  
        } //  
    }  
    catch(e) { //  
        return(e + " handled higher up."); //  
    }  
}
```

```
document.write(TryCatchDemo(0));  
document.write(TryCatchDemo(1));
```

[5](#)

[try...catch](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **try...catch...finally**

JScript

```
try { tryStatements}  
catch(exception) {  
    catchStatements}  
finally {  
    finallyStatements}
```

tryStatement

exception

*exception*

catchStatement

*tryStatement*

finallyStatements

## **try...catch...finally JScript**

```
tryStatements      catchStatement      tryStatements  
catchStatements    exception        tryStatements      catchStatements  
  
tryStatements    catchStatements    throw  
  
tryStatements    catchStatements    finallyStatements  
  
try    catch        catch        finallyStatements
```

*finallyStatements*

**catch**

JScript

```
try {
    print("Outer try running..");
    try {
        print("Nested try running...");
        throw "an error";
    }
    catch(e) {
        print("Nested catch caught " + e);
        throw e + " re-thrown";
    }
    finally {
        print("Nested finally is running...");
    }
}
catch(e) {
    print("Outer catch caught " + e);
}
finally {
    print("Outer finally running");
}

// Windows Script Host          WScript.Echo(s)
function print(s){
    document.write(s);
```

}

Outer try running..

Nested try running...

Nested catch caught an error

Nested finally is running...

Outer catch caught an error re-thrown

Outer finally running

[5](#)

[throw](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

**var**

**var** *variable1* [ = *value1* ] [, *variable2* [ = *value2* ], ...]

*variable*, *variable2*

*value*, *value2*

**var**

**var**

**var** index;

**var** name = "Thomas Jefferson";

**var** answer = 42, counter, numpages = 10;

[1](#)

[function](#) | [new](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

**while**

**false**

**while** (*expression*)  
*statements*

*expression*

Boolean      *expression*    **true**      *expression*    **false**

*statements*

*expression*    **true**

**while**    *expression*    *expression*    **false**

**while**

```
function BreakTest(breakpoint){  
    var i = 0;  
    while (i < 100)  
    {  
        if (i == breakpoint)  
            break;  
        i++;  
    }  
    return(i);
```

}

[1](#)

[break](#) | [continue](#) | [do...while](#) | [for](#) | [for...in](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

**with**

**with** (*object*)  
*statements*

*object*

*statements*

*object*

**with**                  **Math**

x = Math.cos(3 \* Math.PI) + Math.sin(Math.LN10)

y = Math.tan(14 \* Math.E)

**with**

**with** (Math){  
    x = cos(3 \* PI) + sin (LN10)  
    y = tan(14 \* E)  
}

[1](#)

this

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **FileSystemObject**

[FileSystemObject](#)

[FileSystemObject Scripting](#)

[FileSystemObject](#)

[FileSystemObject](#)

[FileSystemObject](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

# **FileSystemObject**

Active Server Pages Windows Scripting Host scripting Web  
Web

Scripting    **FileSystemObject (FSO)**

- [FileSystemObject Scripting](#)
- [FileSystemObject](#)
- [FileSystemObject](#)
- .
- .
- [FileSystemObject](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **FileSystemObject Scripting**

**FileSystemObject (FSO)**

*object.method*

- HTML Web
- Windows Scripting Host Microsoft Windows
- Script Control

FSO FSO Internet Web Internet Explorer

**FileSystemObject**

FSO

FSO Access SQL

Scripting (Scrrun.dll) FSO

**TextStream**

---

[© 2000 Microsoft Corporation](#)

JScript

---

# FileSystemObject

## FileSystemObject(FO)

/	
FileSystemObject	FSO
Drive	"drive" CD-ROM RAM
Drives	<b>Drives</b>
File	
Files	
Folder	
Folders	<b>Folder</b>
TextStream	

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **FileSystemObject**

### **FileSystemObject (FSO)**

- CreateObject FileSystemObject
- 
- 

FSO Scripting Scrrun.dll FSO Scrrun.dll  
Web

## **FileSystemObject**

**CreateObject**    **FileSystemObject**    VBScript  
**FileSystemObject**

Dim fso

Set fso = CreateObject("Scripting.FileSystemObject")

**FileSystemObject**

JScript

```
var fso;  
fso = new ActiveXObject("Scripting.FileSystemObject");
```

**Scripting**        **FileSystemObject**        **FileSystemObject**

**FileSystemObject**      **CreateTextFile**    **CreateFolder**  
FSO

**FileSystemObject**    **DeleteFile**    **DeleteFolder**      **File**  
**Folder**    **Delete**

**FileSystemObject**      **CopyFile**  
**File**    **Copy**

**FileSystemObject** "get"

- GetDrive
- GetFolder
- GetFile

VBScript

```
Dim fso, f1
Set fso = CreateObject("Scripting.FileSystemObject")
Set f1 = fso.GetFile("c:\test.txt")
```

JScript

```
var fso, f1;
fso = new ActiveXObject("Scripting.FileSystemObject");
f1 = fso.GetFile("c:\\test.txt");
```

"get" "create"	<b>CreateFolder</b>	<b>GetFolder</b>
<b>NamePathSize</b>	<b>CreateFolder</b>	VBScript

```
Sub CreateFolder
    Dim fso, fldr
    Set fso = CreateObject("Scripting.FileSystemObject")
    Set fldr = fso.CreateFolder("C:\MyTest")
    Response.Write "Created folder: " & fldr.Name
End Sub
```

JScript      **CreateFolder**

```
function CreateFolder()
```

```
{  
    var fso, fldr;  
    fso = new ActiveXObject("Scripting.FileSystemObject");  
    fldr = fso.CreateFolder("C:\\MyTest");  
    Response.Write("Created folder: " + fldr.Name);  
}
```

## **GetFolder**

VBScript

### **GetFolder**

```
Set fldr = fso.GetFolder("c:\")
```

JScript

```
var fldr = fso.GetFolder("c:\\");
```

**Folder**      **Name**    VBScript

```
Response.Write "Folder name is: " & fldr.Name
```

JScript      **Name**

```
Response.Write("Folder name is: " + fldr.Name);
```

VBScript

```
Dim fso, f1
```

```
Set fso = CreateObject("Scripting.FileSystemObject")
```

```
'
```

```
Set f1 = fso.GetFile("c:\detlog.txt")
```

```
'
```

```
Response.Write "File last modified: " & f1.DateLastModified
```

JScript

```
var fso, f1;
```

```
fso = new ActiveXObject("Scripting.FileSystemObject");
```

```
//
```

```
f1 = fso.GetFile("c:\\detlog.txt");
//  
Response.Write("File last modified: " + f1.DateLastModif
```

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **FileSystemObject (FSO) Windows**

## **Drive**

- **TotalSize**
- **AvailableSpace**   **FreeSpace**
- **DriveLetter**
- CD-ROM   RAM                            **DriveType**
- SerialNumber
- FATFAT32NTFS                            **FileSystem**
- **IsReady**
- /ShareName   VolumeName
- **Path**   **RootFolder**

### **FileSystemObject**

## **Drive**

**Drive**

**Drive**

**GetDrive**

**Drive**   **drv**

VBScript

**Drive**

```
Sub ShowDriveInfo(drvPath)
```

```
Dim fso, drv, s
```

```
Set fso = CreateObject("Scripting.FileSystemObject")
```

```
Set drv = fso.GetDrive(fso.GetDriveName(drvPath))
```

```
s = "Drive " & UCase(drvPath) & " - "
```

```
s = s & drv.VolumeName & "<br>"
```

```
s = s & "Total Space: " & FormatNumber(drv.TotalSize)
```

```
s = s & " Kb" & "<br>"
```

```
s = s & "Free Space: " & FormatNumber(drv.FreeSpace)
```

```
s = s & " Kb" & "<br>"  
Response.Write s  
End Sub
```

JScript

```
function ShowDriveInfo1(drvPath)  
{  
    var fso, drv, s ="";  
    fso = new ActiveXObject("Scripting.FileSystemObject"  
    drv = fso.GetDrive(fso.GetDriveName(drvPath));  
    s += "Drive " + drvPath.toUpperCase() + " - ";  
    s += drv.VolumeName + "<br>";  
    s += "Total Space: " + drv.TotalSize / 1024;  
    s += " Kb" + "<br>";  
    s += "Free Space: " + drv.FreeSpace / 1024;  
    s += " Kb" + "<br>";  
    Response.Write(s);  
}
```

	FileSystemObject.CreateFolder
	Folder.Delete FileSystemObject.DeleteFolder
	Folder.Move FileSystemObject.MoveFolder
	Folder.Copy FileSystemObject.CopyFolder
	Folder.Name
	FileSystemObject.FolderExists
<b>Folder</b>	FileSystemObject.GetFolder
	FileSystemObject.GetParentFolderName
	FileSystemObject.GetSpecialFolder

## FileSystemObject

VBScript      **Folder**    **FileSystemObject**

```
Sub ShowFolderInfo()
    Dim fso, fldr, s
    '   FileSystemObject
    Set fso = CreateObject("Scripting.FileSystemObject")
    '   Drive
    Set fldr = fso.GetFolder("c:")
    '
    Response.Write "Parent folder name is: " & fldr & "<br>
    "
    Response.Write "Contained on drive " & fldr.Drive & "<br>
    "
    If fldr.IsRootFolder = True Then
        Response.Write "This is the root folder." & ""<br>"<br>
```

```

Else
    Response.Write "This folder isn't a root folder." & "<br>"
End If
' FileSystemObject
fso.CreateFolder ("C:\Bogus")
Response.Write "Created folder C:\Bogus" & "<br>"
'
Response.Write "Basename = " & fso.GetBaseName("c:")
'
fso.DeleteFolder ("C:\Bogus")
Response.Write "Deleted folder C:\Bogus" & "<br>"
End Sub

```

JScript    **Folder**    **FileSystemObject**

```

function ShowFolderInfo()
{
    var fso, fldr, s = "";
    // FileSystemObject
    fso = new ActiveXObject("Scripting.FileSystemObject")
    // Drive
    fldr = fso.GetFolder("c:");
    //
    Response.Write("Parent folder name is: " + fldr + "<br>'"
    //
    Response.Write("Contained on drive " + fldr.Drive + "<t
    //
    if (fldr.IsRootFolder)
        Response.Write("This is the root folder.");

```

```
else
    Response.Write("This folder isn't a root folder.");
    Response.Write("<br><br>");
    // FileSystemObject
    fso.CreateFolder ("C:\\Bogus");
    Response.Write("Created folder C:\\Bogus" + "<br>");
    //
    Response.Write("Basename = " + fso.GetBaseName("c:\\"
    //
    fso.DeleteFolder ("C:\\Bogus");
    Response.Write("Deleted folder C:\\Bogus" + "<br>");
}
```

---

JScript

---

•  
•

""

### **CreateTextFile** VBScript

```
Dim fso, f1  
Set fso = CreateObject("Scripting.FileSystemObject")  
Set f1 = fso.CreateTextFile("c:\testfile.txt", True)
```

### JScript

```
var fso, f1;  
fso = new ActiveXObject("Scripting.FileSystemObject");  
f1 = fso.CreateTextFile("c:\\testfile.txt", true);
```

**FileSystemObject**    **OpenTextFile**        **ForWriting**  
VBScript

```
Dim fso, ts  
Const ForWriting = 2  
Set fso = CreateObject("Scripting. FileSystemObject")  
Set ts = fso.OpenTextFile("c:\test.txt", ForWriting, True)
```

### JScript

```
var fso, ts;  
var ForWriting= 2;  
fso = new ActiveXObject("Scripting.FileSystemObject");  
ts = fso.OpenTextFile("c:\\test.txt", ForWriting, true);
```

**OpenAsTextStream**        **ForWriting** VBScript

```
Dim fso, f1, ts
Const ForWriting = 2
Set fso = CreateObject("Scripting.FileSystemObject")
fso.CreateTextFile ("c:\test1.txt")
Set f1 = fso.GetFile("c:\test1.txt")
Set ts = f1.OpenAsTextStream(ForWriting, True)
```

JScript

```
var fso, f1, ts;
var ForWriting = 2;
fso = new ActiveXObject("Scripting.FileSystemObject");
fso.CreateTextFile ("c:\\test1.txt");
f1 = fso.GetFile("c:\\test1.txt");
ts = f1.OpenAsTextStream(ForWriting, true);
```

FileSystemObject OpenTextFile File OpenAsTextStream

TextStream WriteWriteLine WriteBlankLines

	Write
	WriteLine
	WriteBlankLines

**TextStream Close**

/

VBScript

```
Sub CreateFile()
    Dim fso, tf
    Set fso = CreateObject("Scripting.FileSystemObject")
    Set tf = fso.CreateTextFile("c:\testfile.txt", True)
    '
    tf.WriteLine("Testing 1, 2, 3.")
    '
    tf.WriteBlankLines(3)
    '
```

```
    tf.Write ("This is a test.")
    tf.Close
End Sub
```

JScript

```
function CreateFile()
{
    var fso, tf;
    fso = new ActiveXObject("Scripting.FileSystemObject")
    tf = fso.CreateTextFile("c:\\testfile.txt", true);
    //
    tf.WriteLine("Testing 1, 2, 3.");
    //
    tf.WriteBlankLines(3);
    //
    tf.Write ("This is a test.");
    tf.Close();
}
```

**TextStream**   **Read**  
**ReadLine**   **ReadAll**

	Read
	ReadLine
	ReadAll

**Read**   **ReadLine**              **Skip**   **SkipLine** **read**  
**LeftRight**   **Mid**

VBScript

```
Sub ReadFiles
    Dim fso, f1, ts, s
    Const ForReading = 1
    Set fso = CreateObject("Scripting.FileSystemObject")
    Set f1 = fso.CreateTextFile("c:\testfile.txt", True)
    '
    Response.Write "Writing file <br>"
    f1.WriteLine "Hello World"
    f1.WriteBlankLines(1)
    f1.Close
    '
    Response.Write "Reading file <br>"
    Set ts = fso.OpenTextFile("c:\testfile.txt", ForReading)
    s = ts.ReadLine
    Response.Write "File contents = '" & s & "'"
    ts.Close
```

End Sub

JScript

```
function ReadFiles()
{
    var fso, f1, ts, s;
    var ForReading = 1;
    fso = new ActiveXObject("Scripting.FileSystemObject");
    f1 = fso.CreateTextFile("c:\\testfile.txt", true);
    //
    Response.Write("Writing file <br>");
    f1.WriteLine("Hello World");
    f1.WriteBlankLines(1);
    f1.Close();
    //
    Response.Write("Reading file <br>");
    ts = fso.OpenTextFile("c:\\testfile.txt", ForReading);
    s = ts.ReadLine();
    Response.Write("File contents = '" + s + "'");
    ts.Close();
}
```

## FSO

	File.Move FileSystemObject.MoveFile
	File.Copy FileSystemObject.CopyFile
	File.Delete
	FileSystemObject.DeleteFile

VBScript C \tmp \temp

C \tmp \temp

## Sub ManipFiles

```
Dim fso, f1, f2, s
Set fso = CreateObject("Scripting.FileSystemObject")
Set f1 = fso.CreateTextFile("c:\testfile.txt", True)
Response.Write "Writing file <br>
'
f1.Write ("This is a test.")
'
f1.Close
Response.Write "Moving file to c:\tmp <br>
'
C      (C:\\)
Set f2 = fso.GetFile("c:\testfile.txt")
'
\temp
f2.Move ("c:\tmp\testfile.txt")
Response.Write "Copying file to c:\temp <br>
'
\temp
```

```
f2.Copy ("c:\temp\testfile.txt")
Response.Write "Deleting files <br>
'
Set f2 = fso.GetFile("c:\tmp\testfile.txt")
Set f3 = fso.GetFile("c:\temp\testfile.txt")
'
f2.Delete
f3.Delete
Response.Write "All done!"
End Sub
```

JScript

```
function ManipFiles()
{
    var fso, f1, f2, s;
    fso = new ActiveXObject("Scripting.FileSystemObject")
    f1 = fso.CreateTextFile("c:\\testfile.txt", true);
    Response.Write("Writing file <br>");
    //
    f1.Write("This is a test.");
    //
    f1.Close();
    Response.Write("Moving file to c:\\tmp <br>");
    //      C      (C:\\)
    f2 = fso.GetFile("c:\\testfile.txt");
    //          \\tmp
    f2.Move ("c:\\tmp\\testfile.txt");
    Response.Write("Copying file to c:\\temp <br>");
```

```
//      \temp
f2.Copy ("c:\\temp\\testfile.txt");
Response.Write("Deleting files <br>");
//
f2 = fso.GetFile("c:\\tmp\\testfile.txt");
f3 = fso.GetFile("c:\\temp\\testfile.txt");
//
f2.Delete();
f3.Delete();
Response.Write("All done!");
}
```

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **FileSystemObject**

### **FileSystemObject**

Active Server Pages Windows Scripting Host

Active Server Pages

Web .asp

&ltBODY>...</BODY>

<%...%>

**Option Explicit** HTML <HTML>

<%...%>      **Option Explicit**

Sub Print(x)

    Response.Write "<PRE>&ltFONT FACE=""      "" SIZE="

        Response.Write x

    Response.Write "</FONT></PRE>"

End Sub

Main

Windows Scripting Host

Sub Print(x)

    WScript.Echo x

End Sub

Main

' FileSystemObject  
'Copyright 1998 Microsoft Corporation

# Option Explicit

```
' 1)      "&"  
'  
'  
'  
'  
' 2)      "Option Explicit"  
'          DriveTypeCDROM  
'          DriveTypeCDORM  
'  
' 3)  
'  
'      On Error Resume Next Err
```

## Dim TabStop

```
Dim NewLine

Const TestDrive = "C"
Const TestFilePath = "C:\Test"

'#####
' Drive.DriveType
'#####

Const DriveTypeRemovable = 1
Const DriveTypeFixed = 2
Const DriveTypeNetwork = 3
Const DriveTypeCDROM = 4
Const DriveTypeRAMDisk = 5

'#####
' File.Attributes
'#####

Const FileAttrNormal = 0
Const FileAttrReadOnly = 1
Const FileAttrHidden = 2
Const FileAttrSystem = 4
Const FileAttrVolume = 8
Const FileAttrDirectory = 16
Const FileAttrArchive = 32
Const FileAttrAlias = 64
Const FileAttrCompressed = 128

'#####
```

```
''''''''''''''''''''''''''''''''''
Const OpenFileForReading = 1
Const OpenFileForWriting = 2
Const OpenFileForAppending = 8
''''''''''''''''''''''''''''''''''
```

```
' ShowDriveType
```

```
'           Drive
```

```
' - Drive.DriveType
''''''''''''''''''''''''''''''
```

```
Function ShowDriveType(Drive)
```

```
Dim S
```

```
Select Case Drive.DriveType
```

```
Case DriveTypeRemovable
```

```
    S = "Removable"
```

```
Case DriveTypeFixed
```

```
    S = "Fixed"
```

```
Case DriveTypeNetwork
```

```
    S = "Network"
```

```
Case DriveTypeCDROM
```

```
    S = "CD-ROM"
```

```
Case DriveTypeRAMDisk
```

```
    S = "RAM Disk"
```

```
Case Else
```

```
S = "Unknown"  
End Select  
  
ShowDriveType = S  
  
End Function
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''  
' ShowFileAttr  
'  
'  
'  
' - File.Attributes  
' - Folder.Attributes  
''''''''''''''''''''''''''''''''''''''''''''''''''''
```

```
Function ShowFileAttr(File) ' File  
  
Dim S  
Dim Attr  
  
Attr = File.Attributes  
  
If Attr = 0 Then  
    ShowFileAttr = "Normal"  
    Exit Function  
End If  
  
If Attr And FileAttrDirectory Then S = S & "Directory"  
If Attr And FileAttrReadOnly Then S = S & "Read-Onl
```

```
If Attr And FileAttrHidden Then S = S & "Hidden "
If Attr And FileAttrSystem Then S = S & "System "
If Attr And FileAttrVolume Then S = S & "Volume "
If Attr And FileAttrArchive Then S = S & "Archive "
If Attr And FileAttrAlias Then S = S & "Alias "
If Attr And FileAttrCompressed Then S = S & "Compre
```

```
ShowFileAttr = S
```

```
End Function
```

---

### ' GenerateDriveInformation

```
' - FileSystemObject.Drives
' - Iterating the Drives collection
' - Drives.Count
' - Drive.AvailableSpace
' - Drive.DriveLetter
' - Drive.DriveType
' - Drive.FileSystem
' - Drive.FreeSpace
' - Drive.IsReady
' - Drive.Path
' - Drive.SerialNumber
' - Drive.ShareName
```

```
' - Drive.TotalSize  
' - Drive.VolumeName  
::::::::::::::::::
```

## Function GenerateDriveInformation(FSO)

Dim Drives

Dim Drive

Dim S

Set Drives = FSO.Drives

S = "Number of drives:" & TabStop & Drives.Count & N

'

S = S & String(2, TabStop) & "Drive"

S = S & String(3, TabStop) & "File"

S = S & TabStop & "Total"

S = S & TabStop & "Free"

S = S & TabStop & "Available"

S = S & TabStop & "Serial" & NewLine

'

S = S & "Letter"

S = S & TabStop & "Path"

S = S & TabStop & "Type"

S = S & TabStop & "Ready?"

S = S & TabStop & "Name"

S = S & TabStop & "System"

S = S & TabStop & "Space"

```
S = S & TabStop & "Space"
S = S & TabStop & "Space"
S = S & TabStop & "Number" & NewLine

'
S = S & String(105, "-") & NewLine

For Each Drive In Drives
    S = S & Drive.DriveLetter
    S = S & TabStop & Drive.Path
    S = S & TabStop & ShowDriveType(Drive)
    S = S & TabStop & Drive.IsReady

If Drive.IsReady Then
    If DriveTypeNetwork = Drive.DriveType Then
        S = S & TabStop & Drive.ShareName
    Else
        S = S & TabStop & Drive.VolumeName
    End If
    S = S & TabStop & Drive.FileSystem
    S = S & TabStop & Drive.TotalSize
    S = S & TabStop & Drive.FreeSpace
    S = S & TabStop & Drive.AvailableSpace
    S = S & TabStop & Hex(Drive.SerialNumber)
End If

S = S & NewLine

Next
```

```
GenerateDriveInformation = S  
End Function
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''
```

### ' **GenerateFileInfo**

```
'  
'  
'  
' - File.Path  
' - File.Name  
' - File.Type  
' - File.DateCreated  
' - File.DateLastAccessed  
' - File.DateLastModified  
' - File.Size
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''
```

```
Function GenerateFileInfo(File)
```

```
Dim S
```

```
S = NewLine & "Path:" & TabStop & File.Path  
S = S & NewLine & "Name:" & TabStop & File.Name  
S = S & NewLine & "Type:" & TabStop & File.Type  
S = S & NewLine & "Attribs:" & TabStop & ShowFileA  
S = S & NewLine & "Created:" & TabStop & File.DateC  
S = S & NewLine & "Accessed:" & TabStop & File.DateA
```

```
S = S & NewLine & "Modified:" & TabStop & File.Date  
S = S & NewLine & "Size" & TabStop & File.Size & Ne
```

```
GenerateFileInfo = S
```

```
End Function
```

```
"'-----'
```

### **' GenerateFolderInformation**

```
'
```

```
'
```

```
'
```

```
' - Folder.Path  
' - Folder.Name  
' - Folder.DateCreated  
' - Folder.DateLastAccessed  
' - Folder.DateLastModified  
' - Folder.Size
```

```
"'-----'
```

```
Function GenerateFolderInformation(Folder)
```

```
Dim S
```

```
S = "Path:" & TabStop & Folder.Path
```

```
S = S & NewLine & "Name:" & TabStop & Folder.Name
```

```
S = S & NewLine & "Attribs:" & TabStop & ShowFileA
```

```
S = S & NewLine & "Created:" & TabStop & Folder.Dat
```

```
S = S & NewLine & "Accessed:" & TabStop & Folder.D
```

```
S = S & NewLine & "Modified:" & TabStop & Folder.D  
S = S & NewLine & "Size:" & TabStop & Folder.Size &  
  
GenerateFolderInformation = S  
  
End Function
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''  
' GenerateAllFolderInformation  
'  
'  
'  
' - Folder.Path  
' - Folder.SubFolders  
' - Folders.Count  
''''''''''''''''''''''''''''''''''''''''''''''''
```

```
Function GenerateAllFolderInformation(Folder)
```

```
Dim S  
Dim SubFolders  
Dim SubFolder  
Dim Files  
Dim File
```

```
S = "Folder:" & TabStop & Folder.Path & NewLine & N  
Set Files = Folder.Files
```

```
If 1 = Files.Count Then  
S = S & "There is 1 file" & NewLine
```

```
Else
    S = S & "There are " & Files.Count & " files" & NewLine
End If

If Files.Count <> 0 Then
    For Each File In Files
        S = S & GenerateFileInfo(File)
    Next
End If

Set SubFolders = Folder.SubFolders

If 1 = SubFolders.Count Then
    S = S & NewLine & "There is 1 sub folder" & NewLine
Else
    S = S & NewLine & "There are " & SubFolders.Count & " sub folders"
End If

If SubFolders.Count <> 0 Then
    For Each SubFolder In SubFolders
        S = S & GenerateFolderInfo(SubFolder)
    Next
    S = S & NewLine
    For Each SubFolder In SubFolders
        S = S & GenerateAllFolderInfo(SubFolder)
    Next
End If

GenerateAllFolderInformation = S
```

End Function

A decorative horizontal bar consisting of a series of black downward-pointing triangles, evenly spaced along a thin horizontal line.

## ' GenerateTestInformation

1

C:\Test

1

- - FileSystemObject.DriveExists
- - FileSystemObject.FolderExists
- - FileSystemObject.GetFolder

Function GenerateTestInformation(FSO)

# Dim TestFolder

Dim S

```
If Not FSO.DriveExists(TestDrive) Then Exit Function  
If Not FSO.FolderExists(TestFilePath) Then Exit Function
```

```
Set TestFolder = FSO.GetFolder(TestFilePath)
```

`GenerateTestInformation` = `GenerateAllFolderInformation`

End Function

Digitized by srujanika@gmail.com

## ' DeleteTestDirectory

1

```
' test  
'  
' - FileSystemObject.GetFolder  
' - FileSystemObject.DeleteFile  
' - FileSystemObject.DeleteFolder  
' - Folder.Delete  
' - File.Delete  
*****
```

## Sub DeleteTestDirectory(FSO)

```
Dim TestFolder  
Dim SubFolder  
Dim File  
  
'  
FSO.DeleteFile(TestFilePath & "\Beatles\OctopusGardei  
Set File = FSO.GetFile(TestFilePath & "\Beatles\Bathroc  
File.Delete  
  
'  
FSO.DeleteFolder(TestFilePath & "\Beatles")  
FSO.DeleteFile(TestFilePath & "\ReadMe.txt")  
Set TestFolder = FSO.GetFolder(TestFilePath)  
TestFolder.Delete  
  
End Sub
```

```
' CreateLyrics  
'  
'  
'  
' - FileSystemObject.CreateTextFile  
' - TextStream.WriteLine  
' - TextStream.Write  
' - TextStream.WriteBlankLines  
' - TextStream.Close
```

## Sub CreateLyrics(Folder)

# Dim TextStream

```
Set TextStream = Folder.CreateTextFile("OctopusGarder
```

```
TextStream.Write("Octopus' Garden ") '
```

```
TextStream.WriteLine("(by Ringo Starr)")
```

**TextStream.WriteBlankLines(1)**

```
TextStream.WriteLine("I'd like to be under the sea in an
```

```
TextStream.WriteLine("He'd let us in, knows where we've
```

**TextStream.WriteBlankLines(2)**

# TextStream.Close

```
Set TextStream = Folder.CreateTextFile("BathroomWinc
TextStream.WriteLine("She Came In Through The Bathr
TextStream.WriteLine("")"
TextStream.WriteLine("She came in through the bathroo
TextStream.WriteLine("But now she sucks her thumb an
TextStream.WriteBlankLines(2)
TextStream.Close

End Sub
```

```
#####
' GetLyrics
'
'
'     lyrics
'
'
' - FileSystemObject.OpenTextFile
' - FileSystemObject.GetFile
' - TextStream.ReadAll
' - TextStream.Close
' - File.OpenAsTextStream
' - TextStream.AtEndOfStream
' - TextStream.ReadLine
#####
```

```
Function GetLyrics(FSO)

Dim TextStream
Dim S
Dim File
```

```
'  
'
```

```
Set TextStream = FSO.OpenTextFile(TestFilePath & "\B
```

```
S = TextStream.ReadAll & NewLine & NewLine  
TextStream.Close
```

```
Set File = FSO.GetFile(TestFilePath & "\Beatles\Bathroo  
Set TextStream = File.OpenAsTextStream(OpenFileForF  
Do While Not TextStream.AtEndOfStream  
    S = S & TextStream.ReadLine & NewLine  
Loop  
TextStream.Close
```

```
GetLyrics = S
```

```
End Function
```

```
"'....."
```

```
' BuildTestDirectory
```

```
'
```

```
    FileSystemObject
```

```
'
```

```
' C:\Test
```

```
' C:\Test\ReadMe.txt
```

```
' C:\Test\Beatles
```

```
' C:\Test\Beatles\OctopusGarden.txt
```

```
' C:\Test\Beatles\BathroomWindow.txt  
'  
' - FileSystemObject.DriveExists  
' - FileSystemObject.FolderExists  
' - FileSystemObject.CreateFolder  
' - FileSystemObject.CreateTextFile  
' - Folders.Add  
' - Folder.CreateTextFile  
' - TextStream.WriteLine  
' - TextStream.Close  
::::::::::::::::::
```

## Function BuildTestDirectory(FSO)

Dim TestFolder

Dim SubFolders

Dim SubFolder

Dim TextStream

' (a) (b)

```
If Not FSO.DriveExists(TestDrive) Then  
    BuildTestDirectory = False  
    Exit Function  
End If
```

```
If FSO.FolderExists(TestFilePath) Then  
    BuildTestDirectory = False  
    Exit Function  
End If
```

```
Set TestFolder = FSO.CreateFolder(TestFilePath)

Set TextStream = FSO.CreateTextFile(TestFilePath & "\\")

TextStream.WriteLine("My song lyrics collection")
TextStream.Close

Set SubFolders = TestFolder.SubFolders
Set SubFolder = SubFolders.Add("Beatles")
CreateLyrics SubFolder
BuildTestDirectory = True

End Function

' test
' test
' test
```

```
Sub Main

Dim FSO

TabStop = Chr(9)
NewLine = Chr(10)

Set FSO = CreateObject("Scripting.FileSystemObject")
```

```
If Not BuildTestDirectory(FSO) Then  
    Print "Test directory already exists or cannot be created  
    Exit Sub  
End If  
  
Print GenerateDriveInformation(FSO) & NewLine & Ne  
Print GenerateTestInformation(FSO) & NewLine & New  
Print GetLyrics(FSO) & NewLine & NewLine  
DeleteTestDirectory(FSO)  
  
End Sub
```

---

[© 2000 Microsoft Corporation](#)

JScript

---

# Script

Dictionary	<a href="#">Add Dictionary</a>
Folders Folder	<a href="#">Add Folders</a>
	<a href="#">BuildPath</a>
TextStream	<a href="#">Close</a>
	<a href="#">Copy</a>
	<a href="#">CopyFile</a>
	<a href="#">CopyFolder</a>
	<a href="#">CreateFolder</a>
TextStream	<a href="#">CreateTextFile</a>
	<a href="#">Delete</a>
	<a href="#">DeleteFile</a>
	<a href="#">DeleteFolder</a>
True False	<a href="#">DriveExists</a>
Dictionary True False	<a href="#">Exists</a>
True False	<a href="#">FileExists</a>
True False	<a href="#">FolderExists</a>
	<a href="#">GetAbsolutePathName</a>
	<a href="#">GetBaseName</a>
Drive	<a href="#">GetDrive</a>
	<a href="#">GetDriveName</a>
	<a href="#">GetExtensionName</a>
File	<a href="#">GetFile</a>
	<a href="#">GetFileName</a>
	<a href="#">GetFileVersion</a>
Folder	<a href="#">GetFolder</a>
	<a href="#">GetParentFolderName</a>
	<a href="#">GetSpecialFolder</a>
	<a href="#">GetTempName</a>
Dictionary	<a href="#">Items</a>
Dictionary	<a href="#">Keys</a>
	<a href="#">Move</a>
	<a href="#">MoveFile</a>
	<a href="#">MoveFolder</a>
TextStream	<a href="#">OpenAsTextStream</a>
TextStream	<a href="#">OpenTextFile</a>
TextStream	<a href="#">Read</a>
TextStream	<a href="#">ReadAll</a>
TextStream	<a href="#">ReadLine</a>
Dictionary	<a href="#">Remove</a>
Dictionary	<a href="#">RemoveAll</a>

TextStream	<a href="#">Skip</a>
TextStream	<a href="#">SkipLine</a>
TextStream	<a href="#">Write</a>
TextStream	<a href="#">WriteBlankLines</a>
TextStream	<a href="#">WriteLine</a>

---

[© 2000 Microsoft Corporation](#)

JScript

---

## Add (Dictionary)

### Dictionary

object.**Add** (key, item)

object

### Dictionary

key

*item*    *key*

item

*key*    *item*

*key*

### Add

```
var d;  
d = new ActiveXObject("Scripting.Dictionary");  
d.Add("a", "Athens");  
d.Add("b", "Belgrade");  
d.Add("c", "Cairo");
```

[Add \(Folders\)](#) | [Exists](#) | [Items](#) | [Keys](#) | [Remove](#) | [RemoveAll](#)  
[Dictionary](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## Add (Folders)

**Folders**    **Folder**

object.**Add** (folderName)

object

**Folders**

folderName

**Folder**

**Add**

```
function AddNewFolder(path,folderName)
{
    var fso, f, fc, nf;
    fso = new ActiveXObject("Scripting.FileSystemObject"
    f = fso.GetFolder(path);
    fc = f.SubFolders;
    if (folderName != "") )
        nf = fc.Add(folderName);
    else
        nf = fc.Add("New Folder");
}
```

*folderName*

[Add \(Dictionary\)](#) [Folders](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **BuildPath**

*object.***BuildPath**(*path, name*)

*object*

*FileSystemObject*

*path*

*name*

*name*

*path*

## **BuildPath**

### **BuildPath**

```
function GetBuildPath(path)
```

```
{
```

```
    var fso, newpath;
```

```
    fso = new ActiveXObject("Scripting.FileSystemObject")
```

```
    newpath = fso.BuildPath(path, "New Folder");
```

```
    return(newpath);
```

```
}
```

[GetAbsoluteName](#) | [GetBaseName](#) | [GetDriveName](#) |  
[GetExtensionName](#) | [GetFileName](#) | [GetParentFolderName](#) |  
[GetTempName](#)      [FileSystemObject](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **Close**

### **TextStream**

*object.Close( );*

*object* **TextStream**

## **Close**

```
var fso;  
fso = new ActiveXObject("Scripting.FileSystemObject");  
a = fso.CreateTextFile("c:\\testfile.txt", true);  
a.WriteLine("This is a test.");  
a.Close();
```

---

JScript

---

## **Copy**

object.**Copy( destination[, overwrite] );**

object

**File    Folder**

destination

overwrite

Boolean

**True**

**False**

**Copy   File   Folder   FileSystemObject.CopyFile**  
**FileSystemObject.CopyFolder              object**

## **Copy**

```
var fso, f;  
fso = new ActiveXObject("Scripting.FileSystemObject");  
f = fso.CreateTextFile("c:\\testfile.txt", true);  
f.WriteLine("This is a test.");  
f.Close();  
f = fso.GetFile("c:\\testfile.txt");
```

```
f.Copy("c:\\windows\\desktop\\test2.txt");
```

[CopyFile](#) | [CopyFolder](#) | [Delete](#) | [Move](#) | [OpenAsTextStream](#)  
[File](#) | [Folder](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **CopyFile**

object.**CopyFile** ( source, destination[, overwrite] )

object

*object*      **FileSystemObject**

source

destination

*source*

overwrite

Boolean	<b>true</b>	<b>false</b>	<b>true</b>
<i>destination</i>	<i>overwrite</i>	<b>CopyFile</b>	

*source*

```
fso = new ActiveXObject("Scripting.FileSystemObject");
fso.CopyFile ("c:\\mydocuments\\letters\\*.doc", "c:\\temp\\
```

```
fso = new ActiveXObject("Scripting.FileSystemObject");
fso.CopyFile ("c:\\mydocuments\\*\\R1??97.xls", "c:\\ter
```

*source*    *destination* ()        *destination*

*destination*

- *destination*        *source*
- *destination*        *overwrite*      **false**        *source*
- *destination*

*source*           **CopyFile**

[Copy](#) | [CopyFolder](#) | [CreateTextFile](#) | [DeleteFile](#) | [MoveFile](#)  
[FileSystemObject](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **CopyFolder**

object.**CopyFolder** ( source, destination[, overwrite] );

object

    FileSystemObject

source

destination

*source*

overwrite

    Boolean

**true**

**false**

**true**

*source*

```
fso = new ActiveXObject("Scripting.FileSystemObject");
fso.CopyFolder ("c:\\mydocuments\\letters\\*", "c:\\tempf
```

```
fso = new ActiveXObject("Scripting.FileSystemObject");
fso.CopyFolder ("c:\\mydocuments\\*\\*", "c:\\tempfolder
```

*source*   *destination* ()           *destination*  
    *destination*

- *destination*      *source*
- *destination*
- *destination*      *source*    *destination*      *overwrite*    **false**
- *destination*      *overwrite*    **false**

*source*

## **CopyFolder**

[CopyFile](#) | [Copy](#) | [CreateFolder](#) | [DeleteFolder](#) | [MoveFolder](#)  
[FileSystemObject](#)

---

© 2000 Microsoft Corporation

JScript

---

## **CreateFolder**

object.**CreateFolder**(foldername)

object

    FileSystemObject

    foldername

## **CreateFolder**

```
var fso = new ActiveXObject("Scripting.FileSystemObject");
var a = fso.CreateFolder("c:\\new folder");
```

[CopyFolder](#) | [DeleteFolder](#) | [MoveFolder](#)      [FileSystemObject](#)

---

© 2000 Microsoft Corporation

JScript

---

## CreateTextFile

### TextStream

object.**CreateTextFile**(filename[, overwrite[, unicode]])

object

FileSystemObject Folder

filename

overwrite

Boolean                   **true**    **false**

unicode

Boolean	Unicode	ASCII	Unicode	<b>true</b>
ASCII		<b>false</b>	ASCII	

## CreateTextFile

```
var fso = new ActiveXObject("Scripting.FileSystemObject");
var a = fso.CreateTextFile("c:\\testfile.txt", true);
a.WriteLine("This is a test.");
a.Close();
```

*overwrite*   **false**       *filename*

[CreateFolder](#) | [OpenAsTextStream](#) | [OpenTextFile](#)      [FileSystemObject](#)  
[\\_](#) | [Folder](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **Delete**

*object.***Delete( force );**

object

**File    Folder**

force

Boolean              **True    False**

Delete File Folder FileSystemObject.DeleteFile  
FileSystemObject.DeleteFolder

## **Delete**

### **Delete**

```
var fso, f;  
fso = new ActiveXObject("Scripting.FileSystemObject");  
f = fso.CreateTextFile("c:\\testfile.txt", true);  
f.WriteLine("This is a test.");  
f.Close();  
f = fso.GetFile("c:\\testfile.txt");  
f.Delete();
```

[Copy](#) | [DeleteFile](#) | [DeleteFolder](#) | [Move](#) | [OpenAsTextStream](#)  
[File](#) | [Folder](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **DeleteFile**

*object.***DeleteFile** ( *filespec*[, *force*] );

object

    FileSystemObject

filespec

*filespec*

force

    Boolean

**true**    **false**

## **DeleteFile**

### **DeleteFile**

```
function DeleteFile(filespec)
```

```
{
```

```
    var fso;
```

```
    fso = new ActiveXObject("Scripting.FileSystemObject")
```

```
    fso.DeleteFile(filespec);
```

```
}
```

[CopyFile](#) | [CreateTextFile](#) | [Delete](#) | [DeleteFolder](#) | [MoveFile](#)

## FileSystemObject

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **DeleteFolder**

*object.***DeleteFolder** ( *folderspec*[, *force*] );

object

    FileSystemObject

folderspec

*folderspec*

force

    Boolean

**true**    **false**

## **DeleteFolder**

**DeleteFolder**

**DeleteFolder**

function DeleteFolder(folderspec)

{

    var fso;

    fso = new ActiveXObject("Scripting.FileSystemObject")

    fso.DeleteFolder(folderspec);

}

[CopyFolder](#) | [CreateFolder](#) | [Delete](#) | [DeleteFile](#) | [MoveFolder](#)  
[FileSystemObject](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **DriveExists**

**True    False**

*object*.**DriveExists**(*drivespec*)

*object*

    FileSystemObject

*drivespec*

**DriveExists**    **true**        **Drive**    **IsReady**

## **DriveExists**

function ReportDriveStatus(drv)

{

    var fso, s = "";

    fso = new ActiveXObject("Scripting.FileSystemObject")

    if (fso.DriveExists(drv))

        s += "Drive " + drv + " exists.";

    else

        s += "Drive " + drv + " doesn't exist.";

    return(s);

}

[Drive Object](#) | [Drives Collection](#) | [FileExists](#) | [FolderExists](#) | [GetDrive](#) |  
[GetDirectoryName](#) | [IsReady](#) | [FileSystemObject](#)

---

© 2000 Microsoft Corporation

JScript

---

## Exists

**Dictionary true false**

*object.Exists(key)*

object

**Dictionary**

key

**Dictionary key**

## Exists

function keyExists(k)

{

var fso, s = "";

d = new ActiveXObject("Scripting.Dictionary");

d.Add("a", "Athens");

d.Add("b", "Belgrade");

d.Add("c", "Cairo");

if (d.Exists(k))

    s += "Specified key exists.;"

else

    s += "Specified key doesn't exist.;"

return(s);

}

[Add \(Dictionary\)](#) | [Items](#) | [Keys](#) | [Remove](#) | [RemoveAll](#)  
[Dictionary](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **FileExists**

**True ; False**

*object.FileExists(filespec)*

object

FileSystemObject

filespec

..

## **FileExists**

function ReportFileStatus(filespec)

{

```
var fso, s = filespec;
fso = new ActiveXObject("Scripting.FileSystemObject"
if (fso.FileExists(filespec))
    s += " exists.";
else
    s += " doesn't exist.";
return(s);
}
```

[DriveExists](#) | [FolderExists](#) | [GetFile](#) | [GetFileName](#)  
[FileSystemObject](#)

JScript

---

## **FolderExists**

**True    False**

*object.FolderExists(folderspec)*

object

    FileSystemObject

folderspec

## **FileExists**

```
function ReportFolderStatus(fldr)
{
    var fso, s = fldr;
    fso = new ActiveXObject("Scripting.FileSystemObject"
    if (fso.FolderExists(fldr))
        s += " exists.";
    else
        s += " doesn't exist.";
    return(s);
}
```

[DriveExists](#) | [FileExists](#) | [GetFolder](#) | [GetParentFolderName](#)  
[FileSystemObject](#)

JScript

---

## **GetAbsolutePathName**

*object*.GetAbsolutePathName(*pathspec*)

*object*

FileSystemObject

*pathspec*

(\)

c:\mydocuments\reports      **GetAbsolutePathName**

pathspec	
"c:"	"c:\mydocuments\reports"
"c.."	"c:\mydocuments"
"c:\\"	"c:\\"
"c:.*\may97"	"c:\mydocuments\reports\*.*\may97"
"region1"	"c:\mydocuments\reports\region1"
"c:\\..\\..\\mydocuments"	"c:\mydocuments"

GetAbsolutePathName

function ShowAbsolutePath(path)

{

  var fso, s= "";

  fso = new ActiveXObject("Scripting.FileSystemObject")  
  s += fso.GetAbsolutePathName(path);

```
    return(s);  
}
```

[GetBaseName](#) | [GetDrive](#) | [GetDriveName](#) | [GetExtensionName](#) |  
[GetFile](#) | [GetFileName](#) | [GetFileVersion](#) | [GetFolder](#) |  
[GetParentFolderName](#) | [GetSpecialFolder](#) | [GetTempName](#)  
[FileSystemObject](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **GetBaseName**

*object*.GetBaseName(*path*)

*object*

    FileSystemObject

*path*

*path*       **GetBaseName** ("")

    GetBaseName    *path*

**GetBaseName**

function ShowBaseName(filespec)

{

    var fso, s = "";

    fso = new ActiveXObject("Scripting.FileSystemObject")

    s += fso.GetBaseName(filespec);

    return(s);

}

[GetAbsolutePathName](#) | [GetDrive](#) | [GetDirectoryName](#) | [GetExtensionName](#)

[\\_\\_](#) | [GetFile](#) | [GetFileName](#) | [GetFileVersion](#) | [GetFolder](#) |  
[GetParentFolderName](#) | [GetSpecialFolder](#) | [GetTempName](#)  
[FileSystemObject](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **GetDrive**

### **Drive**

*object.***GetDrive** (*drivespec*);

*object*

FileSystemObject

*drivespec*

*drivespec* (c) (c:) (c:\) (\computer2\share1)

*drivespec*

**GetDrive**      *drivespec*

DriveSpec = GetDriveName(GetAbsolutePathName(Path))

### **GetDrive**

function ShowFreeSpace(drvPath)

{

var fso, d, s ="";

fso = new ActiveXObject("Scripting.FileSystemObject")

d = **fso.GetDrive**(fso.GetDriveName(drvPath));

s = "Drive " + drvPath.toUpperCase( ) + " - ";

s += d.VolumeName + "<br>";

```
s += "Free Space: " + d.FreeSpace/1024 + " Kbytes";
return(s);
}
```

[GetAbsoluteName](#) | [GetBaseName](#) | [GetDriveName](#) |  
[GetExtensionName](#) | [GetFile](#) | [GetFileName](#) | [GetFileVersion](#) |  
[GetFolder](#) | [GetParentFolderName](#) | [GetSpecialFolder](#) | [GetTempName](#)  
[FileSystemObject](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **GetDriveName**

*object*.GetDriveName(*path*)

*object*

    FileSystemObject

*path*

**GetDriveName** ("")

**GetDriveName**   *path*

**GetDriveName**

function GetDriveLetter(*path*)

{

    var fso, s ="";

    fso = new ActiveXObject("Scripting.FileSystemObject")

    s += fso.GetDrive(**fso.GetDriveName(fso.GetAbsolute]**)

    return(s);

}

[GetAbsolutePathName](#) | [GetBaseName](#) | [GetDrive](#) | [GetExtensionName](#)

[\\_\\_](#) | [GetFile](#) | [GetFileName](#) | [GetFileVersion](#) | [GetFolder](#) |  
[GetParentFolderName](#) | [GetSpecialFolder](#) | [GetTempName](#)  
[FileSystemObject](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **GetExtensionName**

*object*.GetExtensionName(*path*)

*object*

    FileSystemObject

*path*

(\)

*path*       **GetExtensionName** ("")

### **GetExtensionName**

function ShowExtensionName(filespec)

{

    var fso, s = "";

    fso = new ActiveXObject("Scripting.FileSystemObject")

    s += fso.GetExtensionName(filespec);

    return(s);

}

[GetAbsolutePathName](#) | [GetBaseName](#) | [GetDrive](#) | [GetDirectoryName](#) |

[GetFile](#) | [GetFileName](#) | [GetFileVersion](#) | [GetFolder](#) |  
[GetParentFolderName](#) | [GetSpecialFolder](#) | [GetTempName](#)  
[FileSystemObject](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **GetFile**

### **File**

object.**GetFile**(filespec)

object

FileSystemObject

filespec

*filespec*

## **GetFile**

function Show FileAccessInfo(filespec)

{

var fso, f, s;

fso = new ActiveXObject("Scripting.FileSystemObject")

f = **fso.GetFile**(filespec);

s = f.Path.toUpperCase() + "<br>";

s += "Created: " + f.DateCreated + "<br>";

s += "Last Accessed: " + f.DateLastAccessed + "<br>";

s += "Last Modified: " + f.DateLastModified

return(s);

}

[GetAbsoluteName](#) | [GetBaseName](#) | [GetDrive](#) | [GetDriveName](#) |  
[GetExtensionName](#) | [GetFileName](#) | [GetFileVersion](#) | [GetFolder](#) |  
[GetParentFolderName](#) | [GetSpecialFolder](#) | [GetTempName](#)  
[FileSystemObject](#)

---

© 2000 Microsoft Corporation

JScript

---

## **GetFileName**

*object.***GetFileName**(*pathspec*)

*object*

    FileSystemObject

*pathspec*

*pathspec*       **GetFileName** ("")

**GetFileName**

**GetFileName**

function ShowFileName(filespec)

{

    var fso, s = "";

    fso = new ActiveXObject("Scripting.FileSystemObject")

    s += fso.**GetFileName**(filespec);

    return(s);

}

[GetAbsolutePathName](#) | [GetBaseName](#) | [GetDrive](#) | [GetDirectoryName](#) |

[GetExtensionName](#) | [GetFile](#) | [GetFileVersion](#) | [GetFolder](#) |  
[GetParentFolderName](#) | [GetSpecialFolder](#) | [GetTempName](#)  
[FileSystemObject](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **getFileVersion**

*Object*.**GetFileVersion**(*pathspec*)

*object*

**FileSystemObject**

*pathspec*

*pathspec*           **GetFileVersion** 0("")

**GetFileVersion**

**GetFileVersion**

function ShowFileVersion(*pathspec*)

{

  var fso, s = "";

  fso = new ActiveXObject("Scripting.FileSystemObject")

  s += fso.GetFileVersion(*pathspec*);

  if (s == "")

    s = "              ";

  return(s);

}

1

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **GetFolder**

### **Folder**

object.**GetFolder**(folderspec)

object

FileSystemObject

folderspec

*folderspec*

## **GetFolder**

function ShowFolderList(folderspec)

{

var fso, f, fc, s;

fso = new ActiveXObject("Scripting.FileSystemObject"

f = **fso.GetFolder**(folderspec);

fc = new Enumerator(f.SubFolders);

s = "";

for (; !fc.atEnd(); fc.moveNext())

{

s += fc.item();

s += "<br>";

```
    }  
    return(s);  
}
```

[GetAbsoluteName](#) | [GetBaseName](#) | [GetDrive](#) | [GetDirectoryName](#) |  
[GetExtensionName](#) | [GetFile](#) | [GetFileName](#) | [GetFileVersion](#) |  
[GetParentFolderName](#) | [GetSpecialFolder](#) | [GetTempName](#)  
[FileSystemObject](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **GetParentFolderName**

*object*.GetParentFolderName(*path*)

*object*

    FileSystemObject

*path*

*path*       **GetParentFolderName** ("")

**GetParentFolderName**   *path*

    GetParentFolderName

function ShowParentFolderName(filespec)

{

    var fso, s = "";

    fso = new ActiveXObject("Scripting.FileSystemObject")

    s += fso.GetParentFolderName(filespec);

    return(s);

}

[GetAbsolutePathName](#) | [GetBaseName](#) | [GetDrive](#) | [GetDirectoryName](#) |

[GetExtensionName](#) | [GetFile](#) | [GetFileName](#) | [GetFileVersion](#) |  
[GetFolder](#) | [GetSpecialFolder](#) | [GetTempName](#)      [FileSystemObject](#)

---

© 2000 Microsoft Corporation

JScript

---

## **GetSpecialFolder**

*object.***GetSpecialFolder**(*folderspec*)

*object*

FileSystemObject

*folderspec*

*folderspec*

WindowsFolder	0	Windows Windows
SystemFolder	1	System
TemporaryFolder	2	Temp TMP

## **GetSpecialFolder**

```
var fso, tempfile;  
fso = new ActiveXObject("Scripting.FileSystemObject");  
  
function CreateTempFile()  
{  
    var tfolder, tfile, tname, fname, TemporaryFolder = 2;  
    tfolder = fso.GetSpecialFolder(TemporaryFolder);  
    tname = fso.GetTempName();
```

```
tfile = tfolder.CreateTextFile(tname);
return(tfile);
}
tempfile = CreateTempFile();
tempfile.writeline("Hello World");
tempfile.close();
```

[GetAbsoluteName](#) | [GetBaseName](#) | [GetDrive](#) | [GetDirectoryName](#) |
[GetExtensionName](#) | [GetFile](#) | [GetFileName](#) | [GetFileVersion](#) |
[GetFolder](#) | [GetParentFolderName](#) | [GetTempName](#)  
[FileSystemObject](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **GetTempName**

*object*.GetTempName ( );

*object*      **FileSystemObject**

**GetTempName**                  **CreateTextFile**

**GetTempName**

```
var fso, tempfile;  
fso = new ActiveXObject("Scripting.FileSystemObject");  
  
function CreateTempFile()  
{  
    var tfolder, tfile, tname, fname, TemporaryFolder = 2;  
    tfolder = fso.GetSpecialFolder(TemporaryFolder);  
    tname = fso.GetTempName();  
    tfile = tfolder.CreateTextFile(tname);  
    return(tfile);  
}  
tempfile = CreateTempFile();  
tempfile.writeline("Hello World");  
tempfile.close();
```

[GetExtensionName](#) | [GetFile](#) | [GetFileName](#) | [GetFileVersion](#) |  
[GetFolder](#) | [GetParentFolderName](#) | [GetSpecialFolder](#)  
[FileSystemObject](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **Items**

### **Dictionary**

*object*.Items( )

*object*   **Dictionary**

## **Items**

```
function ItemsDemo()
{
    var a, d, i, s;                  //
    d = new ActiveXObject("Scripting.Dictionary");
    d.Add ("a", "Athens");           //
    d.Add ("b", "Belgrade");
    d.Add ("c", "Cairo");
    a = (new VBArray(d.Items())).toArray();  //
    s = "";
    for (i in a)                    //      dictionary
    {
        s += a[i] + "<br>";
    }
    return(s);                      //
}
```

[Add \(Dictionary\)](#) | [Exists](#) | [Keys](#) | [Remove](#) | [RemoveAll](#)  
[Dictionary](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## Keys

### Dictionary

*object*.Keys( )

*object*    **Dictionary**

## Keys

```
function KeysDemo()
{
    var a, d, i, s;                  //
    d = new ActiveXObject("Scripting.Dictionary");
    d.Add ("a", "Athens");           //
    d.Add ("b", "Belgrade");
    d.Add ("c", "Cairo");
    a = (new VBArray(d.Keys())).toArray(); //
    s = "";
    for (i in a)                    //    dictionary
    {
        s += a[i] + " - " + d(a[i]) + "<br>";
    }
    return(s);                      //
}
```

[Add \(Dictionary\)](#) | [Exists](#) | [Items](#) | [Remove](#) | [RemoveAll](#)  
[Dictionary](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **Move**

object.**Move( destination );**

object

**File    Folder**

destination

Move File Folder FileSystemObject.MoveFile  
FileSystemObject.MoveFolder

[Copy](#) | [Delete](#) | [MoveFile](#) | [MoveFolder](#) | [OpenAsTextStream](#)  
[File](#) | [Folder](#)

JScript

---

## **MoveFile**

object.**MoveFile** ( source, destination );

object

    FileSystemObject

source

*source*

destination

*destination*

*source*           *destination* (\)

*destination*

*destination*

- *destination*
- *destination*
- *destination*

*source*           **MoveFile**

## **MoveFile**

function MoveFile2Desktop(filespec)

{

    var fso;

```
fso = new ActiveXObject("Scripting.FileSystemObject"  
fso.MoveFile(fileSpec, "c:\\windows\\desktop\\");  
}
```

[CopyFile](#) | [DeleteFile](#) | [GetFile](#) | [GetFileName](#) | [Move](#) | [MoveFolder](#)  
\_\_\_\_\_| [OpenTextFile](#)      [FileSystemObject](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **MoveFolder**

object.**MoveFolder** ( source, destination );

object

    FileSystemObject

source

*source*

destination

*destination*

*source*           *destination* (\)

*destination*           *destination*

- *destination*
- *destination*
- *destination*

*source*           **MoveFolder**

## **MoveFolder**

function MoveFldr2Desktop(fldrspec)

{

    var fso;

```
fso = new ActiveXObject("Scripting.FileSystemObject"  
fso.MoveFolder(fldrspec, "c:\\windows\\desktop\\");  
}
```

[CopyFile](#) | [DeleteFile](#) | [GetFile](#) | [GetFileName](#) | [Move](#) | [MoveFile](#)  
| [OpenTextFile](#) | [FileSystemObject](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **OpenAsTextStream**

### **TextStream**

*object*.**OpenAsTextStream**([*iomode*, [*format*]])

*object*

### **File**

*iomode*

/

**ForReading    ForWriting    ForAppending**

*format*

ASCII

*iomode*

ForReading	1	
ForWriting	2	
ForAppending	8	

*format*

TristateUseDefault	-2	
TristateTrue	-1	Unicode
TristateFalse	0	ASCII

OpenAsTextStream FileSystemObject OpenTextFile  
OpenAsTextStream

### **OpenAsTextStream**

```
function TextStreamTest( )
{
    var fso, f, ts, s;
    var ForReading = 1, ForWriting = 2, ForAppending = 8;
    var TristateUseDefault = -2, TristateTrue = -1, TristateFalse = 0;
    fso = new ActiveXObject("Scripting.FileSystemObject");
    fso.CreateTextFile( "test1.txt" );           // Create a file
    f = fso.GetFile("test1.txt");
    ts = f.OpenAsTextStream(ForWriting, TristateUseDefault);
    ts.Write( "Hello World" );
    ts.Close();
    ts = f.OpenAsTextStream(ForReading, TristateUseDefault);
    s = ts.ReadLine();
    ts.Close();
    return(s);
}
```

[Copy](#) | [CreateTextFile](#) | [Delete](#) | [Move](#) | [OpenTextFile](#)      [File](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **OpenTextFile**

### **TextStream**

object.**OpenTextFile**(filename[, iomode[, create[, format]]])

object

*object*      **FileSystemObject**

filename

iomode

ForReading ForWriting ForAppending

create

Boolean                  *filename*                  **True**    **False**

format

ASCII

*iomode*

ForReading	1	
ForWriting	2	
ForAppending	8	

*format*

TristateTrue	Unicode

TristateFalse	ASCII
TristateUseDefault	

## **OpenTextFile**

```
var fs, a, ForAppending;  
ForAppending = 8;  
fs = new ActiveXObject("Scripting.FileSystemObject");  
a = fs.OpenTextFile("c:\\testfile.txt", ForAppending, false);  
...  
a.Close();
```

[CreateTextFile](#) | [OpenAsTextStream](#)      [FileSystemObject](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **Read**

### **TextStream**

object.**Read**(characters)

object

### **TextStream**

characters

**Read** Header

function GetHeader()

{

```
var fso, f;
var ForReading = 1, ForWriting = 2;
fso = new ActiveXObject("Scripting.FileSystemObject");
f = fso.OpenTextFile("c:\\testfile.txt", ForWriting, true);
f.Write("Header");
f.Write("1234567890987654321");
f.Close();
f = fso.OpenTextFile("c:\\testfile.txt", ForReading);
return(f.Read(6));
```

}

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **ReadAll**

### **TextStream**

```
object.ReadAll( );
```

```
object      TextStream
```

## **ReadAll**

### **ReadAll**

```
function GetEverything()
{
    var fso, f;
    var ForReading = 1, ForWriting = 2;
    fso = new ActiveXObject("Scripting.FileSystemObject");
    f = fso.OpenTextFile("c:\\testfile.txt", ForWriting, true);
    f.Write("Header");
    f.Write("1234567890987654321");
    f.Close();
    f = fso.OpenTextFile("c:\\testfile.txt", ForReading);
    return(f.ReadAll());
}
```

[Read](#) | [ReadLine](#) | [Skip](#) | [SkipLine](#)

[TextStream](#)

---

[© 2000 Microsoft Corporation](#)

---

JScript

---

## **ReadLine**

### **TextStream**

*object*..ReadLine( )

*object* **TextStream**

## **Line**

```
function GetLine()
{
    var fso, f, r;
    var ForReading = 1, ForWriting = 2;
    fso = new ActiveXObject("Scripting.FileSystemObject");
    f = fso.OpenTextFile("c:\\testfile.txt", ForWriting, true);
    f.WriteLine("Hello world!");
    f.WriteLine("JScript is fun");
    f.Close();
    f = fso.OpenTextFile("c:\\testfile.txt", ForReading);
    r = f.ReadLine();
    return(r);
}
```

[Read](#) | [ReadAll](#) | [Skip](#) | [SkipLine](#)      [TextStream](#)

JScript

---

## **Remove**

### **Dictionary**

*object.***Remove(key)**

object

### **Dictionary**

key

*key*    **Dictionary**

## **Remove**

```
var a, d, i, s;          //  
d = new ActiveXObject("Scripting.Dictionary");  
d.Add ("a", "Athens");    //  
d.Add ("b", "Belgrade");  
d.Add ("c", "Cairo");  
...  
d.Remove("b");         //
```

---

[Add \(Dictionary\)](#) | [Exists](#) | [Items](#) | [Keys](#) | [RemoveAll](#)  
[Dictionary](#)

[© 2000 Microsoft Corporation](#)

---

JScript

---

## **RemoveAll**

### **RemoveAll    Dictionary**

*object*.RemoveAll( )

*object*    **Dictionary**

## **RemoveAll**

```
var a, d, i;          //  
d = new ActiveXObject("Scripting.Dictionary");  
d.Add ("a", "Athens");    //  
d.Add ("b", "Belgrade");  
d.Add ("c", "Cairo");  
...  
d.RemoveAll( );    //    dictionary
```

[Add \(Dictionary\)](#) | [Exists](#) | [Items](#) | [Keys](#) | [Remove](#)    [Dictionary](#)

---

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **Skip**

### **TextStream**

object.**Skip**(characters)

object

### **TextStream**

characters

## **Skip**

function SkipDemo()

{

var fso, f, r;

var ForReading = 1, ForWriting = 2;

fso = new ActiveXObject("Scripting.FileSystemObject")

f = fso.OpenTextFile("c:\\testfile.txt", ForWriting, true);

f.WriteLine("Hello world!");

f.WriteLine("JScript is fun");

f.Close();

f = fso.OpenTextFile("c:\\testfile.txt", ForReading);

f.Skip(6);

```
r = f.ReadLine();
return(r);
}
```

---

[Close](#) | [Read](#) | [ReadAll](#) | [ReadLine](#) | [SkipLine](#) | [Write](#) | [WriteLine](#)  
\_\_\_\_\_| [WriteBlankLines](#)      [TextStream](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **SkipLine**

### **TextStream**

*object*.SkipLine( )

*object* **TextStream**

## **SkipLine**

```
function SkipLineDemo()
{
    var fso, f, r
    var ForReading = 1, ForWriting = 2;
    fso = new ActiveXObject("Scripting.FileSystemObject"
    f = fso.OpenTextFile("c:\\testfile.txt", ForWriting, true)
    f.WriteLine("Hello world!");
    f.WriteLine("JScript is fun");
    f.Close();
    f = fso.OpenTextFile("c:\\testfile.txt", ForReading);
    f.SkipLine();
    r = f.ReadLine();
    return(r);
}
```

[Read](#) | [ReadAll](#) | [ReadLine](#) | [Skip](#)      [TextStream](#)

JScript

---

## **Write**

### **TextStream**

```
object.Write(string)
```

object

### **TextStream**

string

## **WriteLine**

### **Write**

```
function WriteDemo()
{
    var fso, f, r
    var ForReading = 1, ForWriting = 2;
    fso = new ActiveXObject("Scripting.FileSystemObject")
    f = fso.OpenTextFile("c:\\testfile.txt", ForWriting, true)
    f.Write("Hello world!");
    f.Close();
    f = fso.OpenTextFile("c:\\testfile.txt", ForReading);
    r = f.ReadLine();
    return(r);
}
```

[WriteBlankLines](#) | [WriteLine](#)      [TextStream](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **WriteBlankLines**

### **TextStream**

*object*.WriteBlankLines(*lines*)

*object*

### **TextStream**

*lines*

## **WriteBlankLines**

```
function WriteBlanksDemo()
```

```
{
```

```
    var fso, f, r;
```

```
    var ForReading = 1, ForWriting = 2;
```

```
    fso = new ActiveXObject("Scripting.FileSystemObject")
```

```
    f = fso.OpenTextFile("c:\\testfile.txt", ForWriting, true);
```

```
    f.Write("Hello world!");
```

```
    f.WriteBlankLines(2);
```

```
    f.Write("JScript is fun!");
```

```
    f.Close();
```

```
    f = fso.OpenTextFile("c:\\testfile.txt", ForReading);
```

```
    r = f.ReadAll();
```

```
    return(r);
```

}

[Write](#) | [WriteLine](#)     [TextStream](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **WriteLine**

### **TextStream**

*object*.**WriteLine**([*string*])

*object*

### **TextStream**

*string*

## **WriteLine**

```
var fso, f;  
fso = new ActiveXObject("Scripting.FileSystemObject");  
f = fso.CreateTextFile("c:\\testfile.txt", true);  
f.WriteLine("This is a test.");  
f.Close();
```

[Write](#) | [WriteBlankLines](#)      [TextStream](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## Script

	<a href="#">Dictionary</a>
	<a href="#">Drive</a>
	<a href="#">Drives</a>
	<a href="#">File</a>
File	<a href="#">Files</a>
	<a href="#">FileSystemObject</a>
	<a href="#">Folder</a>
Folder Folder	<a href="#">Folders</a>
	<a href="#">TextStream</a>

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **Dictionary**

```
y = new ActiveXObject("Scripting.Dictionary")
```

### **Dictionary PERL**

#### **Dictionary :**

```
var y = new ActiveXObject("Scripting.Dictionary");
y.add ("a", "test");
if (y.Exists("a"))
    document.write("true");
...
...
```

[Add \(Dictionary\)](#) | [Exists](#) | [Items](#) | [Keys](#) | [Remove](#) | [RemoveAll](#)

[Count](#) | [Item](#) | [Key](#)

[FileSystemObject](#) | [TextStream](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **Drive**

### **Drive**

```
function ShowFreeSpace(drvPath)
{
    var fso, d, s;
    fso = new ActiveXObject("Scripting.FileSystemObject"
    d = fso.GetDrive(fso.GetDriveName(drvPath));
    s = "Drive " + drvPath + " - ";
    s += d.VolumeName + "<br>";
    s += "Free Space: " + d.FreeSpace/1024 + " Kbytes";
    return(s);
}
```

## **Drive**

[AvailableSpace](#) | [DriveLetter](#) | [DriveType](#) | [FileSystem](#) | [FreeSpace](#) |  
[IsReady](#) | [Path](#) | [RootFolder](#) | [SerialNumber](#) | [ShareName](#) | [TotalSize](#)  
| [VolumeName](#)

[Drives](#) | [File](#) | [Files](#) | [Folder](#) | [Folders](#) | [GetDrive](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## Drives

### Drives

Drives   Drives   Enumerator

```
function ShowDriveList()
{
    var fso, s, n, e, x;
    fso = new ActiveXObject("Scripting.FileSystemObject");
    e = new Enumerator(fso.Drives);
    s = "";
    for (; !e.atEnd(); e.moveNext())
    {
        x = e.item();
        s = s + x.DriveLetter;
        s += " - ";
        if (x.DriveType == 3)
            n = x.ShareName;
        else if (x.IsReady)
            n = x.VolumeName;
        else
            n = "[Drive not ready]";
        s += n + "<br>";
    }
    return(s);
```

}

[Count](#) | [Item](#)

[Drive](#) | [Drives](#) | [File](#) | [Files](#) | [Folder](#) | [Folders](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **File**

### **File**

```
function ShowFileInfo(filespec)
{
    var fso, f, s;
    fso = new ActiveXObject("Scripting.FileSystemObject"
    f = fso.GetFile(filespec);
    s = f.DateCreated;
    return(s);
}
```

[Copy](#) | [Delete](#) | [Move](#) | [OpenAsTextStream](#)

[Attributes](#) | [DateCreated](#) | [DateLastAccessed](#) | [DateLastModified](#) |  
[Drive](#) | [Name](#) | [ParentFolder](#) | [Path](#) | [ShortName](#) | [ShortPath](#) | [Size](#)  
| [Type](#)

[Drive](#) | [Drives](#) | [Files](#) | [Folder](#) | [Folders](#)

JScript

---

## **Files**

### **File**

#### **Files Enumerator for**

```
function ShowFolderFileList(folderSpec)
{
    var fso, f, f1, fc, s;
    fso = new ActiveXObject("Scripting.FileSystemObject");
    f = fso.GetFolder(folderSpec);
    fc = new Enumerator(f.files);
    s = "";
    for (; !fc.atEnd(); fc.moveNext())
    {
        s += fc.item();
        s += "<br>";
    }
    return(s);
}
```

## **Files**

[Count](#) | [Item](#)

[Drive](#) | [Drives](#) | [File](#) | [Folder](#) | [Folders](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **FileSystemObject**

```
y = new ActiveXObject("Scripting.FileSystemObject")
```

**FileSystemObjext** **TextStream**

```
var fso = new ActiveXObject("Scripting.FileSystemObjext");
var a = fso.CreateTextFile("c:\\testfile.txt", true);
a.WriteLine("This is a test.");
a.Close();
```

<b>ActiveXObject</b>	<b>FileSystemObject</b>	<b>CreateTextFile</b>
<b>TextStream</b> (a)	<b>WriteLine</b>	<b>Close</b>

[BuildPath](#) | [CopyFile](#) | [CopyFolder](#) | [CreateFolder](#) | [CreateTextFile](#) |  
[DeleteFile](#) | [DeleteFolder](#) | [DriveExists](#) | [FileExists](#) | [FolderExists](#) |  
[GetAbsolutePathName](#) | [GetBaseName](#) | [GetDrive](#) | [GetDirectoryName](#) |  
[GetExtensionName](#) | [GetFile](#) | [GetFileName](#) | [GetFolder](#) |  
[GetParentFolderName](#) | [GetSpecialFolder](#) | [GetTempName](#) | [MoveFile](#)  
| [MoveFolder](#) | [OpenTextFile](#)

[Drives](#)

[Dictionary](#) | [Drive](#) | [Drives](#) | [File](#) | [Files](#) | [Folder](#) | [Folders](#) |

## TextStream

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **Folder**

### **Folder**

```
function ShowFolderInfo(folderSpec)
{
    var fso, folder, s;
    fso = new ActiveXObject("Scripting.FileSystemObject");
    folder = fso.GetFolder(folderSpec);
    s = folder.DateCreated;
    return(s);
}
```

[Copy](#) | [Delete](#) | [Move](#) | [OpenAsTextStream](#)

[Attributes](#) | [DateCreated](#) | [DateLastAccessed](#) | [DateLastModified](#) |  
[Drive](#) | [Files](#) | [IsRootFolder](#) | [Name](#) | [ParentFolder](#) | [Path](#) |  
[ShortName](#) | [ShortPath](#) | [Size](#) | [SubFolders](#) | [Type](#)

[Drive](#) | [Drives](#) | [File](#) | [Files](#) | [Folders](#)

JScript

---

## **Folders**

**Folder**   **Folder**

**Folders**   **Enumerator**   **for**

```
function ShowFolderList(folderSpec)
{
    var fso, f, fc, s;
    fso = new ActiveXObject("Scripting.FileSystemObject");
    f = fso.GetFolder(folderSpec);
    fc = new Enumerator(f.SubFolders);
    s = "";
    for (; !fc.atEnd(); fc.moveNext())
    {
        s += fc.item();
        s += "<br>";
    }
    return(s);
}
```

[Add \(Folders\)](#)

[Count | Item](#)

[Drive](#) | [Drives](#) | [File](#) | [Files](#) | [Folder](#) | [SubFolders](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **TextStream**

**TextStream.**{*property* | *method( )*}

<i>property</i>	<i>method</i>	<b>TextStream</b>	<b>TextStream</b>
<b>FileSystemObject</b>		<b>TextStream</b>	

a FileSystemObject CreateTextFile TextStream

```
var fso = new ActiveXObject("Scripting.FileSystemObjec
var a = fso.CreateTextFile("c:\\testfile.txt", true);
a.WriteLine("           ");
a.Close();
```

WriteLine Close TextStream

[Close](#) | [Read](#) | [ReadAll](#) | [ReadLine](#) | [Skip](#) | [SkipLine](#) | [Write](#) |  
[WriteBlankLines](#) | [WriteLine](#)

[AtEndOfLine](#) | [AtEndOfStream](#) | [Column](#) | [Line](#)

[Dictionary](#) | [FileSystemObject](#)

JScript

---

## Script

<b>TextStream</b>	<b>True</b>	<a href="#">AtEndOfLine</a>
<b>False</b>		
<b>TextStream</b>	<b>True False</b>	<a href="#">AtEndOfStream</a>
		<a href="#">Attributes</a>
		<a href="#">AvailableSpace</a>
<b>TextStream</b>		<a href="#">Column</a>
<b>Dictionary</b>		<a href="#">CompareMode</a>
<b>Dictionary</b>		<a href="#">Count</a>
		<a href="#">DateCreated</a>
		<a href="#">DateLastAccessed</a>
		<a href="#">DateLastModified</a>
		<a href="#">Drive</a>
		<a href="#">DriveLetter</a>
<b>Drives</b>	<b>Drive</b>	<a href="#">Drives</a>
		<a href="#">DriveType</a>
<b>Files</b>	<b>File</b>	<a href="#">Files</a>
		<a href="#">FileSystem</a>
		<a href="#">FreeSpace</a>
<b>True False</b>		<a href="#">IsReady</a>
<b>True False</b>		<a href="#">IsRootFolder</a>
<b>Dictionary</b>		<a href="#">Item</a>
<b>Dictionary</b>		<a href="#">Key</a>
<b>TextStream</b>		<a href="#">Line</a>
		<a href="#">Name</a>
folder		<a href="#">ParentFolder</a>
		<a href="#">Path</a>
<b>Folder</b>		<a href="#">RootFolder</a>
		<a href="#">SerialNumber</a>
		<a href="#">ShareName</a>
8.3		<a href="#">ShortName</a>
8.3		<a href="#">ShortPath</a>
		<a href="#">Size</a>
<b>Folders</b>		<a href="#">SubFolders</a>
		<a href="#">TotalSize</a>
		<a href="#">Type</a>
		<a href="#">VolumeName</a>

JScript

---

## **AtEndOfLine**

**TextStream**    **true**    **false**

*object*.AtEndOfLine

*object*    **TextStream**

## **AtEndOfLine TextStream**

### **AtEndOfLine**

```
function GetALine(filespec)
{
    var fso, a, s, ForReading;
    ForReading = 1, s = "";
    fso = new ActiveXObject("Scripting.FileSystemObject"
    a = fso.OpenTextFile(filespec, ForReading, false);
    while (!a.AtEndOfLine)
    {
        s += a.Read(1);
    }
    a.Close();
    return(s);
}
```

[AtEndOfStream](#)    [TextStream](#)

---

[© 2000 Microsoft Corporation](#)

---

JScript

---

## **AtEndOfStream**

**TextStream**    **true**    **false**

*object*.AtEndOfStream

*object*    **TextStream**

## **AtEndOfStream TextStream**

### **AtEndOfStream**

```
function GetALine(filespec)
{
    var fso, f, s, ForReading;
    ForReading = 1, s = "";
    fso = new ActiveXObject("Scripting.FileSystemObject"
    f = fso.OpenTextFile(filespec, ForReading, false);
    while (!f.AtEndOfStream)
        s += f.ReadLine();
    f.Close();
    return(s);
}
```

[AtEndOfLine](#)    [TextStream](#)

JScript

---

## Attributes

/

object.**Attributes** [= newattributes]

object

**File**    **Folder**

newattributes

*newattributes*    *object*

*newattributes*

Normal	0	
ReadOnly	1	/
Hidden	2	/
System	4	/
Volume	8	
Directory	16	
Archive	32	/
Alias	64	
Compressed	128	

## Attributes

function ToggleArchiveBit(filespec)  
{

```
var fso, f, r, s;
fso = new ActiveXObject("Scripting.FileSystemObject"
f = fso.GetFile(filespec)
if (f.attributes && 32)
{
    f.attributes = f.attributes - 32;
    s = "Archive bit is cleared.";
}
else
{
    f.attributes = f.attributes + 32;
    s = "Archive bit is set.";
}
return(s);
}
```

[DateCreated](#) | [DateLastAccessed](#) | [DateLastModified](#) | [Drive](#) | [Files](#) |  
[IsRootFolder](#) | [Name](#) | [ParentFolder](#) | [Path](#) | [ShortName](#) | [ShortPath](#)  
| [Size](#) | [SubFolders](#) | [Type](#)  
[File](#) | [Folder](#)

---

JScript

---

## **AvailableSpace**

*object*.AvailableSpace

*object*    **Drive**

**AvailableSpace**    **FreeSpace**    quotas

### **AvailableSpace**

function ShowAvailableSpace(drvPath)

{

```
var fso, d, s;
fso = new ActiveXObject("Scripting.FileSystemObject"
d = fso.GetDrive(fso.GetDriveName(drvPath));
s = "Drive " + drvPath.toUpperCase() + " - ";
s += d.VolumeName + "<br>";
s += "Available Space: " + d.AvailableSpace/1024 + " F
return(s);
}
```

[DriveLetter](#) | [DriveType](#) | [FileSystem](#) | [FreeSpace](#) | [IsReady](#) | [Path](#)  
[RootFolder](#) | [SerialNumber](#) | [ShareName](#) | [TotalSize](#) | [VolumeName](#)  
[Drive](#)

JScript

---

## **Column**

### **TextStream**

*object.Column*

*object*    **TextStream**

### **Column 1**

## **Column**

```
function GetColumn()
{
    var fso, f, m;
    var ForReading = 1, ForWriting = 2;
    fso = new ActiveXObject("Scripting.FileSystemObject");
    f = fso.OpenTextFile("c:\\testfile.txt", ForWriting, true);
    f.Write("Hello World!");
    f.Close();
    f = fso.OpenTextFile("c:\\testfile.txt", ForReading);
    m = f.ReadLine();
    return(f.Column);
}
```

[Line](#)    [TextStream](#)

JScript

---

## **CompareMode**

### **Dictionary**

*object*.**CompareMode**[ = *compare* ]

*object*

### **Dictionary**

*compare*

*compare* 0 ()1 (), 2 ()2 ID (LCID)

### **Dictionary**

### **CompareMode**

```
function TestCompareMode(key)
{
    //
    var a, d;
    var BinaryCompare = 0, TextCompare = 1;
    d = new ActiveXObject("Scripting.Dictionary");
    //
    d.CompareMode = TextCompare;
    //
    d.Add("a", "Athens");
    d.Add("b", "Belgrade");
```

```
d.Add("c", "Cairo");
return(d.Item(key));
}
```

[Key](#)    [Dictionary](#)

---

© 2000 Microsoft Corporation

JScript

---

## **Count**

### **Dictionary**

*object*.**Count**

*object* “”

**Count** :

```
function CountDemo()
{
    var a, d, i, s;                //
    d = new ActiveXObject("Scripting.Dictionary");
    d.Add ("a", "Athens");         //
    d.Add ("b", "Belgrade");
    d.Add ("c", "Cairo");
    a = (new VBArray(d.Keys()));   //
    s = "";
    for (i = 0; i < d.Count; i++) //      dictionary
    {
        s += a.getItem(i) + " - " + d(a.getItem(i)) + "<br>";
    }
    return(s);                    //
}
```

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **DateCreated**

*object*.DateCreated

*object*    **File**    **Folder**

### **DateCreated**

```
function ShowFileInfo(filespec)
{
    var fso, f, s;
    fso = new ActiveXObject("Scripting.FileSystemObject");
    f = fso.GetFile(filespec);
    s = "Created: " + f.DateCreated;
    return(s);
}
```

[Attributes](#) | [DateLastAccessed](#) | [DateLastModified](#) | [Drive](#) | [Files](#) |  
[IsRootFolder](#) | [Name](#) | [ParentFolder](#) | [Path](#) | [ShortName](#) | [ShortPath](#)  
| [Size](#) | [SubFolders](#) | [Type](#)  
[File](#) | [Folder](#)

---

JScript

---

## **DateLastAccessed**

*object*.DateLastAccessed

*object*    **File**    **Folder**

### **DateLastAccessed**

```
function ShowFileInfo(filespec)
{
    var fso, f, s;
    fso = new ActiveXObject("Scripting.FileSystemObject");
    f = fso.GetFile(filespec);
    s = filespec.toUpperCase() + "<br>";
    s += "          " + f.DateCreated + "<br>";
    s += "          " + f.DateLastAccessed + "<br>";
    s += "          " + f.DateLastModified;
    return(s);
}
```

---

[Attributes](#) | [DateCreated](#) | [DateLastModified](#) | [Drive](#) | [Files](#) |  
[IsRootFolder](#) | [Name](#) | [ParentFolder](#) | [Path](#) | [ShortName](#) | [ShortPath](#)  
| [Size](#) | [SubFolders](#) | [Type](#)  
[File](#) | [Folder](#)

[© 2000 Microsoft Corporation](#)

---

JScript

---

## **DateLastModified**

*object*.DateLastModified

*object*    **File**    **Folder**

### **DateLastModified**

```
function ShowFileAccessInfo(filespec)
{
    var fso, f, s;
    fso = new ActiveXObject("Scripting.FileSystemObject");
    f = fso.GetFile(filespec);
    s = filespec.toUpperCase() + "<br>";
    s += "          " + f.DateCreated + "<br>";
    s += "          " + f.DateLastAccessed + "<br>";
    s += "          " + f.DateLastModified;
    return(s);
}
```

[Attributes](#) | [DateCreated](#) | [DateLastAccessed](#) | [Drive](#) | [Files](#) |  
[IsRootFolder](#) | [Name](#) | [ParentFolder](#) | [Path](#) | [ShortName](#) | [ShortPath](#)  
| [Size](#) | [SubFolders](#) | [Type](#)  
[File](#) | [Folder](#)

---

JScript

---

## **Drive**

*object.***Drive**

*object*    **File**    **Folder**

### **Drive**

```
function ShowFileInfo(filespec)
{
    var fso, f, s;
    fso = new ActiveXObject("Scripting.FileSystemObject");
    f = fso.GetFile(filespec);
    s = f.Name + " on Drive " + f.Drive + "<br>";
    s += "           " + f.DateCreated + "<br>";
    s += "           " + f.DateLastAccessed + "<br>";
    s += "           " + f.DateLastModified;
    return(s);
}
```

[Attributes](#) | [DateCreated](#) | [DateLastAccessed](#) | [DateLastModified](#) |  
[Files](#) | [IsRootFolder](#) | [Name](#) | [ParentFolder](#) | [Path](#) | [ShortName](#) |  
[ShortPath](#) | [Size](#) | [SubFolders](#) | [Type](#)  
[File](#) | [Folder](#)

---

JScript

---

## **DriveLetter**

*object*.DriveLetter

*object* **Drive**

**DriveLetter** 0 ("")

### **DriveLetter**

```
function ShowDriveLetter(drvPath)
{
    var fso, d, s;
    fso = new ActiveXObject("Scripting.FileSystemObject");
    d = fso.GetDrive(fso.GetDriveName(drvPath));
    s = "Drive " + d.DriveLetter.toUpperCase() + ": - ";
    s += d.VolumeName + "<br>";
    s += "Available Space: " + d.AvailableSpace/1024 + " K
    return(s);
}
```

[AvailableSpace](#) | [DriveType](#) | [FileSystem](#) | [FreeSpace](#) | [IsReady](#) |  
[Path](#) | [RootFolder](#) | [SerialNumber](#) | [ShareName](#) | [TotalSize](#) |  
[VolumeName](#)  
[Drive](#)

---

JScript

---

## **Drives**

**Drives**      **Drive**

*object*.**Drives**

*object* FileSystemObject

**Drives**

**Enumerator**    **for**    **Drives**

function ShowDriveList()

{

var fso, s, n, e, x;

fso = new ActiveXObject("Scripting.FileSystemObject")

e = new Enumerator(**fso.Drives**);

s = "";

for (; !e.atEnd(); e.moveNext())

{

x = e.item();

s = s + x.DriveLetter;

s += " - ";

if (x.DriveType == 3)

    n = x.ShareName;

else if (x.IsReady)

    n = x.VolumeName;

else

    n = "[                ]";

```
    s += n + "<br>";
}
return(s);
}
```

[Drives](#) | [Files](#) | [SubFolders](#)  
[FileSystemObject](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **DriveType**

*object*.DriveType

*object*    **Drive**

### **DriveType**

```
function ShowDriveType(drvpath)
{
    var fso, d, s, t;
    fso = new ActiveXObject("Scripting.FileSystemObject");
    d = fso.GetDrive(drvpath);
    switch (d.DriveType)
    {
        case 0: t = "Unknown"; break;
        case 1: t = "Removable"; break;
        case 2: t = "Fixed"; break;
        case 3: t = "Network"; break;
        case 4: t = "CD-ROM"; break;
        case 5: t = "RAM Disk"; break;
    }
    s = "Drive " + d.DriveLetter + ": - " + t;
    return(s);
}
```

[AvailableSpace](#) | [DriveLetter](#) | [FileSystem](#) | [FreeSpace](#) | [IsReady](#) |  
[Path](#) | [RootFolder](#) | [SerialNumber](#) | [ShareName](#) | [TotalSize](#) |  
[VolumeName](#)  
[Drive](#)

---

© 2000 Microsoft Corporation

JScript

---

## Files

**Files**      **File**

*object.***Files**

*object*      **Folder**

**Files**

```
function ShowFolderFileList(folderspec)
{
    var fso, f, fc, s;
    fso = new ActiveXObject("Scripting.FileSystemObject");
    f = fso.GetFolder(folderspec);
    fc = new Enumerator(f.files);
    s = "";
    for (; !fc.atEnd(); fc.moveNext())
    {
        s += fc.item();
        s += "<br>";
    }
    return(s);
}
```

[Attributes](#) | [DateCreated](#) | [DateLastAccessed](#) | [DateLastModified](#) |  
[Drive](#) | [IsRootFolder](#) | [Name](#) | [ParentFolder](#) | [Path](#) | [ShortName](#) |  
[ShortPath](#) | [Size](#) | [SubFolders](#) | [Type](#)

## Folder

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **FileSystem**

*object*.FileSystem

*object*    **Drive**

FATNTFS CDFS

### **FileSystem**

```
function ShowFileSystemType(drvPath)
{
    var fso,d, s;
    fso = new ActiveXObject("Scripting.FileSystemObject"
    d = fso.GetDrive(drvPath);
    s = d.FileSystem;
    return(s);
}
```

[AvailableSpace](#) | [DriveLetter](#) | [DriveType](#) | [FreeSpace](#) | [IsReady](#) |  
[Path](#) | [RootFolder](#) | [SerialNumber](#) | [ShareName](#) | [TotalSize](#) |  
[VolumeName](#)  
    [Drive](#)

JScript

---

## **FreeSpace**

*object*.FreeSpace

*object*      **Drive**

**FreeSpace**    **AvailableSpace**    quotas

### **FreeSpace**

```
function ShowFreeSpace(drvPath)
{
    var fso, d, s;
    fso = new ActiveXObject("Scripting.FileSystemObject");
    d = fso.GetDrive(fso.GetDriveName(drvPath));
    s = "Drive " + drvPath.toUpperCase() + " - ";
    s += d.VolumeName + "<br>";
    s += "Free Space: " + d.FreeSpace/1024 + " Kbytes";
    return(s);
}
```

[AvailableSpace](#) | [DriveLetter](#) | [DriveType](#) | [FileSystem](#) | [IsReady](#) |  
[Path](#) | [RootFolder](#) | [SerialNumber](#) | [ShareName](#) | [TotalSize](#) |  
[VolumeName](#)  
[Drive](#)

JScript

---

## **IsReady**

**True    False**

*object*.**IsReady**

*object*    **Drive**

CD-ROM    **IsReady**    **True**

**IsReady**

```
function ShowDriveInfo(drivpath)
{
    var fso, d, s, t;
    fso = new ActiveXObject("Scripting.FileSystemObject")
    d = fso.GetDrive(drivpath)
    switch (d.DriveType)
    {
        case 0: t = "Unknown"; break;
        case 1: t = "Removable"; break;
        case 2: t = "Fixed"; break;
        case 3: t = "Network"; break;
        case 4: t = "CD-ROM"; break;
        case 5: t = "RAM Disk"; break;
    }
    s = "Drive " + d.DriveLetter + ": - " + t;
    if (d.IsReady)
        s += "<br>" + ";
}
```

```
else
    s += "<br>" + "
";
return(s);
}
```

[AvailableSpace](#) | [DriveLetter](#) | [DriveType](#) | [FileSystem](#) | [FreeSpace](#) |  
[Path](#) | [RootFolder](#) | [SerialNumber](#) | [ShareName](#) | [TotalSize](#) |  
[VolumeName](#)

[Drive](#)

---

© 2000 Microsoft Corporation

JScript

---

## **IsRootFolder**

**True    False**

*object*.IsRootFolder

*object*    **Folder**

## **IsRootFolder**

```
function DisplayLevelDepth(pathspec)
{
    var fso, f, n, s = "";
    fso = new ActiveXObject("Scripting.FileSystemObject");
    f = fso.GetFolder(pathspec);
    n = 0;
    if (f.IsRootFolder)
        s = "The specified folder is the root folder."
    else
    {
        do
        {
            f = f.ParentFolder;
            n++;
        }
        while (!f.IsRootFolder)
        s = "The specified folder is nested " + n + " levels deep"
    }
}
```

```
    return(s);  
}
```

[Attributes](#) | [DateCreated](#) | [DateLastAccessed](#) | [DateLastModified](#) |  
[Drive](#) | [Files](#) | [Name](#) | [ParentFolder](#) | [Path](#) | [ShortName](#) | [ShortPath](#)  
| [Size](#) | [SubFolders](#) | [Type](#)  
[Folder](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **Item**

**Dictionary**    *key*    *item*    *key*    *item*/

object.**Item**(*key*)[ = *newitem*]

## **Item**

object	<b>Dictionary</b>	
key	<i>item</i> <i>key</i>	
<i>newitem</i>	<b>Dictionary</b>	<i>newitem</i> <i>key</i>

*key*    *item*    *newitem*    *key*    *key*    *key*

## **Item**

```
function DicTest(keyword)
{
    var a, d;
    d = new ActiveXObject("Scripting.Dictionary");
    d.Add("a", "Athens");
    d.Add("b", "Belgrade");
    d.Add("c", "Cairo");
    a = d.Item(keyword);
    return(a);
}
```

[CompareMode](#) | [Count](#) | [Key](#)

[Dictionary](#) | [Drives](#) | [Files](#) | [Folders](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## Key

**Dictionary** *key*

object.Key(*key*) = newkey

## Key

object	<b>Dictionary</b>
key	<i>key</i>
newkey	<i>key</i>

*key*   *key*   *key*   *item*

## Key

```
var d;  
d = new ActiveXObject("Scripting.Dictionary");
```

```
function AddStuff()  
{  
    var a;  
    d.Add("a", "Athens");  
    d.Add("b", "Belgrade");  
    d.Add("c", "Cairo");  
}
```

```
function ChangeKey(oldkey, newkey)  
{  
    var s;
```

```
d.Key("c") = "Ca";
s = "Key " + oldkey + " changed to " + newkey;
return(s);
}
```

[CompareMode](#) | [Count](#) | [Item](#)  
[Dictionary](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **Line**

### **TextStream**

object.**Line**

*object*      **TextStream**

**Line** 1

## **Line**

function GetLine()

{

  var fso, f, r

  var ForReading = 1, ForWriting = 2;

  fso = new ActiveXObject("Scripting.FileSystemObject")

  f = fso.OpenTextFile("c:\\textfile.txt", ForWriting, true)

  f.WriteLine("Hello world!");

  f.WriteLine("JScript is fun");

  f.Close();

  f = fso.OpenTextFile("c:\\textfile.txt", ForReading);

  r = f.ReadAll();

  return(**f.Line**);

}

[Column](#)      [TextStream](#)

---

[© 2000 Microsoft Corporation](#)

---

JScript

---

## Name

/

object.**Name** [= newname]

## Name

	File	Folder
object		
newname	<i>newname</i>	<i>object</i>

## Name

```
function ShowFileAccessInfo(filespec)
{
    var fso, f, s;
    fso = new ActiveXObject("Scripting.FileSystemObject"
    f = fso.GetFile(filespec);
    s = f.Name + " on Drive " + f.Drive + "<br>";
    s += "           " + f.DateCreated + "<br>";
    s += "           " + f.DateLastAccessed + "<br>";
    s += "           " + f.DateLastModified;
    return(s);
}
```

[Attributes](#) | [DateCreated](#) | [DateLastAccessed](#) | [DateLastModified](#) |  
[Drive](#) | [Files](#) | [IsRootFolder](#) | [ParentFolder](#) | [Path](#) | [ShortName](#) |

[ShortPath](#) | [Size](#) | [SubFolders](#) | [Type](#)  
[File](#) | [Folder](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **ParentFolder**

*object*.ParentFolder

*object*    **File**    **Folder**

### **ParentFolder**

```
function ShowFileInfo(filespec)
{
    var fso, f, s;
    fso = new ActiveXObject("Scripting.FileSystemObject");
    f = fso.GetFile(filespec);
    s = f.Name + " in " + f.ParentFolder + "<br>";
    s += "      " + f.DateCreated + "<br>";
    s += "      " + f.DateLastAccessed + "<br>";
    s += "      " + f.DateLastModified;
    return(s);
}
```

[Attributes](#) | [DateCreated](#) | [DateLastAccessed](#) | [DateLastModified](#) |  
[Drive](#) | [Files](#) | [IsRootFolder](#) | [Name](#) | [Path](#) | [ShortName](#) |  
[ShortPath](#) | [Size](#) | [SubFolders](#) | [Type](#)  
[File](#) | [Folder](#)

JScript

---

## Path

*object*.**Path**

*object*   **FileFolder**    **Drive**

C C: C:\

**File**    **Path**

```
function ShowFileAccessInfo(filespec)
{
    var fso, d, f, s;
    fso = new ActiveXObject("Scripting.FileSystemObject"
    f = fso.GetFile(filespec);
    s = f.Path.toUpperCase() + "<br>";
    s += "      " + f.DateCreated + "<br>";
    s += "      " + f.DateLastAccessed + "<br>";
    s += "      " + f.DateLastModified
    return(s);
}
```

[Attributes](#) | [AvailableSpace](#) | [DateCreated](#) | [DateLastAccessed](#) |  
[DateLastModified](#) | [Drive](#) | [DriveLetter](#) | [DriveType](#) | [Files](#) |  
[FileSystem](#) | [FreeSpace](#) | [IsReady](#) | [IsRootFolder](#) | [Name](#) |  
[ParentFolder](#) | [RootFolder](#) | [SerialNumber](#) | [ShareName](#) | [ShortName](#)  
| [ShortPath](#) | [Size](#) | [SubFolders](#) | [TotalSize](#) | [Type](#) | [VolumeName](#)

[Drive](#) | [File](#) | [Folder](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **RootFolder**

### **Folder**

*object*.RootFolder

*object* **Drive**

### **Folder**

#### **RootFolder**

```
function GetRootFolder(drv)
{
    var fso,d;
    fso = new ActiveXObject("Scripting.FileSystemObject"
    if (fso.DriveExists(drv))
    {
        d = fso.GetDrive(drv);
        return(d.RootFolder);
    }
    else
        return(false);
}
```

[AvailableSpace](#) | [DriveLetter](#) | [DriveType](#) | [FileSystem](#) | [FreeSpace](#) |  
[IsReady](#) | [Path](#) | [SerialNumber](#) | [ShareName](#) | [TotalSize](#) |  
[VolumeName](#)

[Drive](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **SerialNumber**

*object*.**SerialNumber**

*object*   **Drive**

### **SerialNumber**

#### **SerialNumber**

```
function ShowDriveInfo(drivpath){  
    var fso, d, s, t;  
    fso = new ActiveXObject("Scripting.FileSystemObject")  
    d = fso.GetDrive(fso.GetDriveName(fso.GetAbsolutePathName(drivpath)))  
    switch (d.DriveType)  
    {  
        case 0: t = "Unknown"; break;  
        case 1: t = "Removable"; break;  
        case 2: t = "Fixed"; break;  
        case 3: t = "Network"; break;  
        case 4: t = "CD-ROM"; break;  
        case 5: t = "RAM Disk"; break;  
    }  
    s = "Drive " + d.DriveLetter + ": - " + t;  
    s += "<br>" + "SN: " + d.SerialNumber;  
    return(s);  
}
```

1

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **ShareName**

*object*.ShareName

*object*    **Drive**

*object*        **ShareName** ("")

### **ShareName**

function ShowDriveInfo(drvpath)

{

  var fso, d, s;

  fso = new ActiveXObject("Scripting.FileSystemObject")

  d = fso.GetDrive(fso.GetDriveName(fso.GetAbsolutePathName(drvpath)))

  s = "Drive " + d.DriveLetter + ": - " + **d.ShareName**;

  return(s);

}

[AvailableSpace](#) | [DriveLetter](#) | [DriveType](#) | [FileSystem](#) | [FreeSpace](#) |

[IsReady](#) | [Path](#) | [RootFolder](#) | [SerialNumber](#) | [TotalSize](#) |

[VolumeName](#)

[Drive](#)

JScript

---

## **ShortName**

8.3

*object*.**ShortName**

*object*    **File**    **Folder**

**File**    **ShortName**

```
function ShowShortName(filespec)
{
    var fso, f, s;
    fso = new ActiveXObject("Scripting.FileSystemObject");
    f = fso.GetFile(filespec);
    s = "The short name for " + "" + f.Name;
    s += "" + "<br>";
    s += "is: " + "" + f.ShortName + "";
    return(s);
}
```

[Attributes](#) | [DateCreated](#) | [DateLastAccessed](#) | [DateLastModified](#) |  
[Drive](#) | [Files](#) | [IsRootFolder](#) | [Name](#) | [ParentFolder](#) | [Path](#) |  
[ShortPath](#) | [Size](#) | [SubFolders](#) | [Type](#)  
[File](#) | [Folder](#)

JScript

---

## **ShortPath**

8.3

*object*.ShortPath

*object*    **File**    **Folder**

**File**    **ShortName**

```
function ShowShortPath(filespec)
{
    var fso, f, s;
    fso = new ActiveXObject("Scripting.FileSystemObject");
    f = fso.GetFile(filespec);
    s = "The short path for " + "" + f.Name;
    s += "" + "<br>";
    s += "is: " + "" + f.ShortPath + "";
    return(s);
}
```

[Attributes](#) | [DateCreated](#) | [DateLastAccessed](#) | [DateLastModified](#) |  
[Drive](#) | [Files](#) | [IsRootFolder](#) | [Name](#) | [ParentFolder](#) | [Path](#) |  
[ShortName](#) | [Size](#) | [SubFolders](#) | [Type](#)  
[File](#) | [Folder](#)

---

JScript

---

## **Size**

*object.***Size**

*object*    **File**    **Folder**

**Folder**    **Size**

```
function ShowFolderSize(filespec)
{
    var fso, f, s;
    fso = new ActiveXObject("Scripting.FileSystemObject");
    f = fso.GetFolder(filespec);
    s = f.Name + " uses " + f.size + " bytes.";
    return(s);
}
```

[Attributes](#) | [DateCreated](#) | [DateLastAccessed](#) | [DateLastModified](#) |  
[Drive](#) | [Files](#) | [IsRootFolder](#) | [Name](#) | [ParentFolder](#) | [Path](#) |  
[ShortName](#) | [ShortPath](#) | [SubFolders](#) | [Type](#)  
[File](#) | [Folder](#)

---

JScript

---

## **SubFolders**

### **Folders**

*object*.SubFolders

*object*      **Folder**

### **SubFolders**

```
function ShowFolderList(folderspec)
{
    var fso, f, fc, s;
    fso = new ActiveXObject("Scripting.FileSystemObject");
    f = fso.GetFolder(folderspec);
    fc = new Enumerator(f.SubFolders);
    s = "";
    for (; !fc.atEnd(); fc.moveNext())
    {
        s += fc.item();
        s += "<br>";
    }
    return(s);
}
```

[Attributes](#) | [DateCreated](#) | [DateLastAccessed](#) | [DateLastModified](#) |  
[Drive](#) | [Files](#) | [IsRootFolder](#) | [Name](#) | [ParentFolder](#) | [Path](#) |  
[ShortName](#) | [ShortPath](#) | [Size](#) | [Type](#)

## Folder

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **TotalSize**

*object*.TotalSize

*object*    **Drive**

### **TotalSize**

```
function SpaceReport(drvPath)
{
    var fso, d, s;
    fso = new ActiveXObject("Scripting.FileSystemObject");
    d = fso.GetDrive(fso.GetDriveName(drvPath));
    s = "Drive " + drvPath + " - ";
    s += d.VolumeName + "<br>";
    s += "Total Space: " + d.TotalSize/1024 + " Kbytes <br>";
    s += "Free Space: " + d.FreeSpace/1024 + " Kbytes";
    return(s);
}
```

[AvailableSpace](#) | [DriveLetter](#) | [DriveType](#) | [FileSystem](#) | [FreeSpace](#) |  
[IsReady](#) | [Path](#) | [RootFolder](#) | [SerialNumber](#) | [ShareName](#) |  
[VolumeName](#)  
[Drive](#)

---

JScript

---

## Type

.TXT ""

*object*.**Type**

*object*    **File**    **Folder**

## Type

```
function ShowFileType(filespec)
{
    var fso, f, s;
    fso = new ActiveXObject("Scripting.FileSystemObject"
    if (fso.FolderExists(filespec))
        f = fso.GetFolder(filespec);
    else if (fso.FileExists(filespec))
        f = fso.GetFile(filespec);
    else
        s = "File or Folder does not exist.";
    s = f.Name + " is a " + f.Type;
    return(s);
}
```

[Attributes](#) | [DateCreated](#) | [DateLastAccessed](#) | [DateLastModified](#) |  
[Drive](#) | [Files](#) | [IsRootFolder](#) | [Name](#) | [ParentFolder](#) | [Path](#) |  
[ShortName](#) | [ShortPath](#) | [Size](#) | [SubFolders](#)  
[File](#) | [Folder](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **VolumeName**

/

*object.***VolumeName** [= *newname*]

*object*

## **Drive**

*newname*

,        *newname*    *object*

## **VolumeName**

function SpaceReport(drvPath)

{

var fso, d, s;

fso = new ActiveXObject("Scripting.FileSystemObject")

d = fso.GetDrive(fso.GetDriveName(drvPath));

s = "Drive " + drvPath + " - ";

s += **d.VolumeName** + "<br>";

s += "Total Space: " + d.TotalSize/1024 + " Kbytes <br>"

s += "Free Space: " + d.FreeSpace/1024 + " Kbytes";

return(s);

}

[AvailableSpace](#) | [DriveLetter](#) | [DriveType](#) | [FileSystem](#) | [FreeSpace](#) |  
[IsReady](#) | [Path](#) | [RootFolder](#) | [SerialNumber](#) | [ShareName](#) | [TotalSize](#)

---

## [Drive](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

“”

JScript

---

? \* ? \* 'data?.dat'

data1.dat

data2.dat

datax.dat

dataN.dat

\* ? 'data\*.dat'

data.dat

data1.dat

data2.dat

data12.dat

datax.dat

dataXYZ.dat

? \*

JScript

---

“”Warren McCulloch Walter Pitts

1956 , Stephen Kleene McCulloch Pitts “”  
“”“”

Ken Thompson Ken Thompson Unix Unix *qed*

---

[© 2000 Microsoft Corporation](#)

JScript

---

- 
- 
- 

web HTML HTML

VBScript Visual Basic C Jscript JScript  
VBScript

---

[© 2000 Microsoft Corporation](#)

JScript

---

a z

JScript	VBScript	
/^\[ \t]*\$/	"^\[ \t]*\$"	
/\d{2}-\d{5}/	"\d{2}-\d{5}"	ID 25
	"<(.*)>.*<\1>"	HTML

\	'n' "n""\n' "\\" "\\" "\\" "(" "("		
^	<b>RegExp Multiline</b> ^ '\n' '\r'		
\$	<b>RegExp Multiline</b> \$ '\n' '\r'		
*	zo* "z" "zoo" * {0,}		
+	'zo+' "zo" "zoo" "z"+ {1,}		
?	"do(es)??" "do" "does" "do" ? {0,1}		
{n}	n n 'o{2}' "Bob" 'o' "food" o		
{n,}	n n 'o{2,}' "Bob" 'o' "fooooood" o'o{1,}' 'o+"o{0,}' 'o*' "		
{n,m}	m n n <= m n m "o{1,3}" "foooooood" o'o{0,1}' 'o?' "		
?	(* , + , ?, { n } , { n, } , { n,m } ) "ooooo" "o+?" "o" 'o+' 'o'		
.	"\n" "\n" '[.\n]' "		
(pattern)	pattern Matches VBScript	<b>SubMatches</b> JScript	<b>\$0...\$9</b>
	'(' ')'		
(?:pattern)	pattern "" () 'industr(?:y ies) 'industry industries'		
(?=pattern)	pattern 'Windows (?=95 98 NT 2000)' "Windows 2000" "Windows" "Windows 3.1" "Windows"		
(?!pattern)	Negative lookahead matches the search string at any point where a string not matching pattern 'Windows (?!=95 98 NT 2000)' "Windows 3.1" "Windows" "Windows 2000" "Windows"		
x y	x y'z food' "z" "food"(z f)oood' "zood" "food"		
[xyz]	'[abc]' "plain" 'a'		
[^xyz]	'[^abc]' "plain" 'p'		
[a-z]	'[a-z]' 'a' 'z'		
[^a-z]	'[^a-z]' 'a' 'z'		
\b	'er\b' "never" 'er' "verb" 'er'		

\B	'er\B' "verb" 'er' "never" 'er'
\cx	x \cM Control-M x A-Z a-z c 'c'
\d	[0-9]
\D	[^0-9]
\f	\x0c \cL
\n	\x0a \cJ
\r	\x0d \cM
\s	[ \f\n\r\t\v]
\S	[^ \f\n\r\t\v]
\t	\x09 \cI
\v	\x0b \cK
\w	'[A-Za-z0-9_]'
\W	'[^A-Za-z0-9_]'
\xn	n n '\x41' "A"\x041' '\x04' & "1" ASCII .
\num	num num '(.)\1'
\n	\ n n n n (0-7) n
\nm	\ nm is preceded by at least nm nm \ nm n n m n m (0-7) \ nm nm
\nml	n (0-3) m l (0-7) nml
\un	n n Unicode \u00A9 (?)

JScript

---

JScript (/)

*/expression/*

VBScript ("")

*"expression"*

( *expression*) **RegExp Pattern**

JScript

---

\	
((), (?=), (?:), []	
*, +, ?, {n}, {n,}, {n,m}	
^, \$, \anymetacharacter	
	“”

[© 2000 Microsoft Corporation](#)

JScript

---

'A' 'A'

/a/

/7/

/M/

VBScript

"a"

"7"

"M"

JScript 'a"7' 'M'

/a7M/

VBScript

"a7M"

JScript

---

(\)

\$	<b>RegExp Multiline</b> \$ '\n' '\r' \$ \\$
( )	\( \)
*	* \*
+	+ \+
.	\n . \
[ ]	[ \[
?	? \?
\	'n' 'n"\n' "\\" "\\" '\C' "("
^	^ \^
{ }	{ \{
	\

---

[© 2000 Microsoft Corporation](#)

JScript

---

\cx	x \cM Control-M 'c'
\f	\x0c \cL
\n	\x0a \cJ
\r	\x0d \cM
\s	[ \f\n\r\t\v]
\S	[^ \f\n\r\t\v]
\t	\x09 \cI
\v	\x0b \cK

---

[© 2000 Microsoft Corporation](#)

JScript

---

(.) (\n) JScript 'aac"abc"acc"adc'  
'a1c"a2c'a-c' a#c'

/a.c/

VBScript

"a.c"

(.) () JScript 'filename.ext'

/filename\.ext/

VBScript

"filename\.ext"

Chapter 1, Chapter 2

([ ])

- T T T
- V V W

JScript 'Chapter 1"Chapter 2"Chapter 3"Chapter 4' 'Chapter  
5'

/Chapter [12345]/

VBScript

"Chapter [12345]"

'Chapter' 'Chapter'

JScript

/Chapter [1-5]/

VBScript

"Chapter [1-5]"

Unicode

•

[\\-]

•

[-a-z]  
[a-z-]

•

[!--]  
[!-~]

(^) JScript 5

/Chapter [^12345]/

VBScript

"Chapter [^12345]"

1, 2, 3, 4, or 5 'Chapter 7' 'Chapter 9'

(-) JScript

/Chapter [^1-5]/

VBScript

"Chapter [^1-5]"

JScript

/[A-Za-z0-9]/

VBScript

"[A-Za-z0-9]"

---

[© 2000 Microsoft Corporation](#)

JScript

---

*	zo* "z" "zoo" * {0,}
+	'zo+' "zo" "zoo" "z"+ {1,}
?	"do(es)??" "do" "does" "do" ? {0,1}
{n}	n n 'o{2}' "Bob" 'o' "food" o
{n,}	n n 'o{2,}' "Bob" 'o' "fooooood" o'o{1,}' 'o+"o{0,}' 'o*''
{n,m}	m n n <= m n m "o{1,3}" "fooooood" o'o{0,1}' 'o?'

JScript

/Chapter [1-9][0-9]\*/

VBScript

"Chapter [1-9][0-9]\*"

0 9

'+' '?' 'Chapter'

99 JScript

/Chapter [0-9]{1,2}/

VBScript

"Chapter [0-9]{1,2}"

99 Chapter 0 JScript

/Chapter [1-9][0-9]?>/

/Chapter [1-9][0-9]{0,1}/

VBScript

"Chapter [1-9][0-9]?"

"Chapter [1-9][0-9]{0,1}"

'\*' '+' '?'

HTML H1

<H1>Chapter 1 – Introduction to Regular Expressions</H

(<) H1

/<.\*>/

VBScript

"<.\*>"

H1 <H1>

/<.\*?>/

"<.\*?>"

'\*' '+' '?' '?'

---

---

[© 2000 Microsoft Corporation](#)

JScript

---

'Chapter'

<code>^</code>	<b>RegExp Multiline</b> <code>^ '\n'</code> <code>'\r'</code>
<code>\$</code>	<b>RegExp Multiline</b> <code>\$ '\n' '\r'</code>
<code>\b</code>	
<code>\B</code>	

`'^*'`

`'^' '^'`

`'$'`

JScript

`/^Chapter [1-9][0-9]{0,1}/`

VBScript

`"^Chapter [1-9][0-9]{0,1}"`

`/^Chapter [1-9][0-9]{0,1}\$/`

VBScript

`"^Chapter [1-9][0-9]{0,1}\$"`

JScript 'Chapter'

**\bCha/**

VBScript

"\bCha"

'\b' 'Chapter' 'ter'

/ter\b/

"ter\b"

'apt' 'Chapter' 'aptitude' 'apt'

**\Bapt/**

"\Bapt"

'Chapter' 'apt' 'aptitude'

JScript

---

'|' '|' JScript VBScript 'Chapter'  
'Section'

/^Chapter|Section [1-9][0-9]{0,1}\$/  
"^Chapter|Section [1-9][0-9]{0,1}\$"

'Chapter' 'Section' 'Chapter 22' 'Chapter' 'Section 22'  
'Section 22'

'Chapter' 'Section' 'Chapter 1' 'Section 3'

'Chapter' 'Section' JScript

/^(Chapter|Section) [1-9][0-9]{0,1}\$/

VBScript

"^(Chapter|Section) [1-9][0-9]{0,1}\$"

'Chapter Section'	<i>submatch</i>	VBScript	<b>Submatches</b>
JScript	<b>RegExp</b>	\$1-\$9	

'Chapter' 'Section'

'?:' JScript

/^(?:Chapter|Section) [1-9][0-9]{0,1}\$/

VBScript

"^(?:Chapter|Section) [1-9][0-9]{0,1}\$"

'?:'        ?= '?!'

Windows 3.1 Windows 95 Windows 98 Windows NT

Windows 95 Windows 98 Windows NT Windows 2000  
JScript Windows 95 Windows 98 Windows NT

/Windows(?=95 |98 |NT )/

VBScript

"Windows(?=95 |98 |NT )"

'Windows 98' 'Windows' '98'

---

[© 2000 Microsoft Corporation](#)

JScript

---

'?:', '?=', or '?!'

1 99 '\n' n' n

Is is the cost of of gasoline going up up?

JScript

`\b([a-z]+)\1\b/gi`

VBScript

`"\b([a-z]+)\1\b"`

`'[a-z]+'\1' "is issued" "this is"`

JScript ('g') ('i') VBScript

**RegExp**

JScript

```
var ss = "Is is the cost of of gasoline going up up?.\n";
var re = \b([a-z]+)\1\b/gim;      // .
var rv = ss.replace(re,"$1");    // .
```

VBScript

```
Dim ss, re, rv
ss = "Is is the cost of of gasoline going up up?." & vbNewLine
Set re = New RegExp
```

```
re.Pattern = "\b([a-zA-Z]+)\1\b"
re.Global = True
re.IgnoreCase = True
re.Multiline = True
rv = re.Replace(ss,"$1")
```

VBScript                  **RegExp**

**replace**    \$1                  \$2, \$3

(URI) URI (ftp, http, etc)/

<http://msdn.microsoft.com:80/scripting/default.htm>

JScript

```
/(\w+):\//([^\:/]+)(:\d*)?([^\# ]*)/
```

VBScript

```
"(\w+):\//([^\:/]+)(:\d*)?([^\# ]*)"
```

web '^' '/' ':' web '\#'

URI

**RegExp.\$1** "http"

**RegExp.\$2** "msdn.microsoft.com"

**RegExp.\$3** ":80"

**RegExp.\$4** "/scripting/default.htm"

JScript

---

# **JScript**

---

[© 2000 Microsoft Corporation](#)

JScript

---

Microsoft® JScript®

URL Internet Web  
Microsoft Corporation

Microsoft Microsoft

© 1991-2000 Microsoft Corporation

MicrosoftMSMS-DOSActiveXJScriptMicrosoft PressMSDN  
Visual BasicWindowsWindows NTWin32 Win32s Microsoft  
Corporation

---

[© 2000 Microsoft Corporation](#)

JScript

---

## JScript

### ASCII

(ASCII) 7 ASCII ANSI 128 (0–127)

### Automation

Automation

### Boolean

**true**      **false** Boolean Boolean

- true
- false
- null    **undefined** false
- false

ASCII

JScript // /\* \*/

(<) (<=) (>) (>=) (!=) (==)

({})

JScript

- **new**
- **this**

JScript JScript Array Boolean Date Function Global  
Math Number Object RegExp Regular Expression String

/

- 
- “”“”

**null**

null null

- **null**
- **null**

JScript JScript Boolean primitive

**String**

(UTC)

GMT

JScript

## **Number    Boolean**

---

[© 2000 Microsoft Corporation](#)

JScript

---

## Microsoft JScript

	1.0	2.0	3.0	4.0	5.0	5.1	5.5
Microsoft Internet Explorer 3.0	x						
Microsoft Internet Information Server 1.0		x					
Microsoft Internet Explorer 4.0			x				
Microsoft Internet Information Server 4.0			x				
Microsoft Windows Scripting Host 1.0			x				
Microsoft Visual Studio 6.0				x			
Microsoft Internet Explorer 5.0					x		
Microsoft Internet Information Services 5.0						x	
Microsoft Windows 2000						x	
Microsoft Internet Explorer 5.5							x

## JScript

	1.0	2.0	3.0	4.0	5.0	5.5
0...n					x	
<u>\$1...\$9</u>		x				
abs	x					
acos	x					
ActiveXObject		x				
+	x					
anchor	x					
apply				x		
arguments		x				
Array		x				
asin	x					
=	x					
atan	x					
atan2	x					
atEnd		x				
big	x					
&	x					
<<	x					
~	x					
	x					
>>	x					
^	x					
blink	x					

bold	x				
Boolean		x			
break	x				
call				x	
callee					x
caller		x			
catch			x		
@cc_on		x			
ceil	x				
charAt	x				
charCodeAt				x	
,	x				
//	x				
/*..*/	x				
	x				
compile		x			
concat Array		x			
concat String		x			
		x			
		x			
?:	x				
constructor		x			
continue	x				
cos	x				
		x			
Date	x				
decodeURI				x	
decodeURIComponent				x	
--	x				
delete		x			
description			x		
dimensions		x			
/	x				
do...while		x			
E	x				
encodeURI				x	
encodeURIComponent				x	
Enumerator		x			
==	x				
Error			x		
escape	x				
eval	x				
exec		x			
exp	x				
fixed	x				

floor	x				
fontcolor	x				
fontsize	x				
for	x				
for...in			x		
fromCharCode		x			
Function		x			
function	x				
getDate	x				
getDay	x				
getFullYear		x			
getHours	x				
getItem		x			
getMilliseconds		x			
getMinutes	x				
getMonth	x				
GetObject		x			
getSeconds	x				
getTime	x				
getTimezoneOffset	x				
getUTCDate		x			
getUTCDay		x			
getUTCFullYear		x			
getUTCHours		x			
getUTCMilliseconds		x			
getUTCMinutes		x			
getUTCMonth		x			
getUTCSeconds		x			
getVarDate		x			
getYear	x				
Global		x			
global				x	
>	x				
>=	x				
hasOwnProperty			x		
====	x				
@if		x			
if...else	x				
ignoreCase			x		
++	x				
index		x			
indexOf	x				
!=	x				
Infinity		x			
input		x			

instanceof			x	
isFinite		x		
isNaN	x			
isPrototypeOf				x
italics	x			
item		x		
join		x		
Labeled		x		
lastIndex		x		
lastIndexOf	x			
lastMatch				x
lastParen				x
lbound		x		
leftContext				x
length Arguments				x
length Array	x			
length Function	x			
length String	x			
<	x			
<=	x			
link	x			
LN2	x			
LN10	x			
localeCompare				x
log	x			
LOG2E	x			
LOG10E	x			
&&	x			
!	x			
	x			
match		x		
Math	x			
max	x			
MAX_VALUE	x			
message				x
min	x			
MIN_VALUE	x			
%	x			
moveFirst		x		
moveNext		x		
multiline			x	
*	x			
name				x
NaN Global		x		
NaN Number	x			

NEGATIVE_INFINITY		x			
new	x				
!==	x				
Number		x			
number			x		
Object		x			
	x				
parse	x				
parseFloat	x				
parseInt	x				
PI	x				
pop			x		
POSITIVE_INFINITY		x			
pow	x				
prototype		x			
propertyIsEnumerable			x		
push			x		
random	x				
RegExp		x			
		x			
		x			
replace	x				
return	x				
reverse		x			
rightContext			x		
round	x				
ScriptEngine		x			
ScriptEngineBuildVersion		x			
ScriptEngineMajorVersion		x			
ScriptEngineMinorVersion		x			
search		x			
@set		x			
setDate	x				
setFullYear		x			
setHours	x				
setMilliseconds		x			
setMinutes	x				
setMonth	x				
setSeconds	x				
setTime	x				
setUTCDate		x			
setUTCFullYear		x			

setUTCHours		x		
setUTCMilliseconds		x		
setUTCMinutes		x		
setUTCMonth		x		
setUTCSeconds		x		
setYear	x			
shift				x
sin	x			
slice Array		x		
slice String		x		
small	x			
sort		x		
source		x		
splice			x	
split		x		
sqrt	x			
SQRT1_2	x			
SQRT2	x			
strike	x			
String	x			
sub	x			
substr		x		
substring	x			
-	x			
sup	x			
switch		x		
tan	x			
test		x		
this	x			
throw			x	
toArray		x		
toDateString			x	
toExponential			x	
toFixed			x	
toGMTString	x			
toLocaleDateString			x	
toLocaleLowercase			x	
toLocaleString	x			
toLocaleTimeString			x	
toLocaleUppercase			x	
toLowerCase	x			
toPrecision			x	
toString	x			
toTimeString			x	
toUpperCase	x			

toUTCString		x		
try			x	
typeof	x			
ubound		x		
-	x			
undefined			x	
unescape	x			
unshift			x	
>>>	x			
UTC	x			
valueOf		x		
var	x			
VBArray		x		
void		x		
while	x			
with	x			

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **Array**

**Array** (array[0])

```
var piArray = new Array(3.14159);
```

```
var piArray = new Array(1);
piArray [0] = 3.14159;
```

40

| [JScript](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

**Array**   **length**  
JScript

**Array**   **length** ( **Nan**)

length 40

**Array**   **length**

```
var my_array = new Array();  
my_array.length = 99;
```

| [JScript](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **Array arguments**

**Function.prototype.apply**

**Array**

**Arguments**

- Array Arguments

| [apply](#) | [JScript](#) | [JScript](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **Boolean**

**Boolean**    **Boolean.prototype.toString**    **Boolean.prototype.valueOf**  
                    **Boolean**

```
var o = new Object;  
o.f = Boolean.prototype.toString;  
o.f();
```

- Boolean Boolean .prototype.toString Boolean.prototype.valueOf

[Boolean](#) | [JScript](#) | . | [\\_\\_](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

```
myFunction() = 42; //      42
```

•

```
myVar = myFunction(42);
```

•

```
myFunction = new Function("return 42;");
```

[Function](#) | [JScript](#) | [JScript](#) | [JScript](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

**'this'**

**this**      **this** JScript :

- 
- 

JScript                **this**                **new**

**this**

- **this** circle                .radius

**this** JScript

[This](#) | [.| JScript](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## Date

**Date    Date.prototype.toString    Date.prototype.valueOf    Date**

```
var o = new Object;  
o.f = Date.prototype.toString;  
o.f();
```

- **Date    Date.prototype.toString    Date.prototype.valueOf**

[Date](#) | [getDate](#) | .

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **Enumerator**

**Enumerator**    **Enumerator.prototype.atEnd**  
**Enumerator.prototype.item**    **Enumerator.prototype.moveFirst**  
**Enumerator.prototype.moveNext**        **Enumerator**

```
var o = new Object;  
o.f = Enumerator.prototype.atEnd;  
o.f();
```

- **Enumerator**    **Enumerator.prototype.atEnd**  
**Enumerator.prototype.item**  
**Enumerator.prototype.moveFirst**    **Enumerator.prototype.moveNext**        **Enumerator**
  

```
if(x instanceof Enumerator)
```

[Enumerator](#) | [Files](#) | [Folders](#) | [Drives](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

**throw**      **try**      **catch**      **try**    **throw**      **try**    **catch**

- **try**      **catch**
- catch
- catch

[Error](#) | [throw](#) | [try...catch](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

' )'

\* + ?

•

- \( - JScript )

[Regular Expression](#) | [Regular Expression](#)

---

© 2000 Microsoft Corporation

JScript

---

'J'

/[abc]/ "a", "b", "c"

•

- \( - JScript )

[Regular Expression](#) | [Regular Expression](#)

---

© 2000 Microsoft Corporation

JScript

---

## **prototype**

**instanceof** , **prototype** (JScript )(, Internet Explorer  
window COM )

- **prototype** JScript

[Function](#) | [prototype](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## Function

<b>Function</b>	<b>Function prototype</b>	<b>foo</b>
var foo = new Object(); //		"foo"
var x = foo(); //	foo	

- **Function**   **Function prototype**
- 0

[Function](#) | [prototype](#) | [JScript](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

COM

•

(=)

---

[© 2000 Microsoft Corporation](#)

JScript

---

a-z 0-9 \w

var good = /[a-z]/; //  
var notGood = /[z-a]/; //

- a z  
- z a

•

[Regular Expression](#) | [Regular Expression](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **JScript**

-JScript JScript JScript

•

[JScript](#) | .

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **Number**

**Number    Number.prototype.toString    Number.prototype.valueOf  
Number**

**Number Number.prototype.toString or Number.prototype.valueOf**

[Number](#) | [number](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **Object**

**Object**   **Object.prototype.toString** **Object.prototype.valueOf**  
**Object**

**Object**   **Object.prototype.toString**   **Object.prototype.valueOf**

[Object | JScript](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

[Object](#) | [JScript](#) | [JScript](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

**RegExp    RegExp.prototype.toString    RegExp.prototype.valueOf**  
**RegExp**

- **RegExp            RegExp.prototype.toString    RegExp.prototype.valueOf**

[Regular Expression](#) | [Regular Expression](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **String**

**StringString.prototype.toString    String.prototype.valueOf**

- **String    String.prototype.toString    String.prototype.valueOf**

[String | toString](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## JScript

- JScript

[Regular Expression](#) | [Regular Expression](#) | [Regular Expression](#) | [Regular Expression](#) | [compile](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **Number.prototype.toExponential()toExponential()** 0 20 20

- **toExponential()**

[toExponential](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

**Number.prototype.toPrecision**      **toPrecision** 1 2121

- **toPrecision**

[toPrecision](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## **URI**

URIURI URI

**encodeURI**

**encodeURIComponent** JScript URI

URI

<Scheme>:<first>/<second>;<third>?<fourth>

“:”, “/”, “;” “?”

- URI JScript

[decodeURI](#) | [decodeURIComponent](#)

---

© 2000 Microsoft Corporation

JScript

---

## **URI**

URI URI Unicode

Unicode

URI

<Scheme>:<first>/<second>;<third>?<fourth>

“:”, “/”, “;” “?”

[encodeURI](#) | [encodeURIComponent](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## JScript

- - 
  -
- 
- **var**      **var** x)

[JScript](#) | . | [JScript](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

/ $\wedge$ +/

$\wedge$

$\wedge$	
\$	
\b	
\B	
*	
+	
?	
{n}	n
{n,}	n
{n,m}	n m , m

•

[Regular Expression](#) | [Regular Expression](#)

---

© 2000 Microsoft Corporation

JScript

---

## **V<sub>B</sub>Array**

Visual Basic safeArray

new V<sub>B</sub>Array(safeArray);

V<sub>B</sub>Arrays safeArray V<sub>B</sub>Array V<sub>B</sub>Array V<sub>B</sub>Array  
ActiveX

- **V<sub>B</sub>Array** V<sub>B</sub>Array

[V<sub>B</sub>Array](#) | .

---

[© 2000 Microsoft Corporation](#)

JScript

---

**'break'**

**break**      **break**      **switch**      **switch**

**break**

**break** labelname;

**switch**      **break**

- **break** switch

[Break](#) | . | [JScript](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## 'continue'

**continue**      **continue**

- **do-while**
- **while**
- **for**
- **for/in**

**continue**

- **do-while**
- **while**
- **for**
- **for/in**

[Continue](#) | . | [JScript](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

JScript @

/\*@cc\_on @\*/

- 
- /\*@cc\_on @\*/

| . | [@cc\\_on](#) | [@if](#) | [@set](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## 'switch' 'default'

switch      **default** Default switch

- **switch default** switch default

[Switch](#) | . | [JScript](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

'('

```
for (initialize; test; increment) {  
    statement;  
}
```

•

[JScript](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

')'

```
for (initialize; test; increment) {  
    statement;  
}
```

•

[JScript](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

'/'

(/) ( / )

•

[Regular Expression](#) | [Regular Expression](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

'.'

•

[\(?:\)](#) | [JScript](#) | [JScript](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

'.'

- 
- 
- **for**

[JScript](#) | [JScript](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

'@'

@set @

• @

@set @myvar = 1

@set | . | .

---

[© 2000 Microsoft Corporation](#)

JScript

---

'@end'

@end JScript      @if/@end

- @end

| .

---

[© 2000 Microsoft Corporation](#)

JScript

---

'J'

•

| [Array](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

'{'

•

[Function](#) | [JScript](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

'}'

**for**

•

[Function](#) | . | [JScript](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

'='

•

@set @myvar1 = 1

| .

---

[© 2000 Microsoft Corporation](#)

JScript

---

## 'catch'

**try            catch**

**try            throw    try            try catch**

- **catch**
- **finally    catch**

[try...catch | Error](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

- 
- 

| .

---

[© 2000 Microsoft Corporation](#)

JScript

---

Unicode \u00A9 A-F  
a-f Unicode

z = "\u1A5F";

- Unicode \u09 A-F a-f

t \u\_ (\u) -

[JScript](#) | [JScript](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

- 
- 
- 
- 
- 

[JScript | .](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

(“”):

```
var point = {x:1.2, y:-3.4};
```

•

(,)

---

[© 2000 Microsoft Corporation](#)

JScript

---

**'while'**

**do ... while**      **do while**

- **while .**

[while](#) | . | [JScript](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## JScript

- ASCII (.)
- ASCII
- 
- JScript

[JScript](#) | . | [JScript](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

- 
- 

| [switch](#) | [break](#) | [continue](#)

---

© 2000 Microsoft Corporation

JScript

---

•

[Labeled](#) | [switch](#) | [break](#) | [continue](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

**'return'**

**return**

**()        return**

**return [ expression ];**

**return                  undefined**

**return                  return    try            finally            finally**

**return                  undefined**

- **return**

[return](#) | [Function](#) | [caller](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## JScript

- 
- 

[Error](#) | [JScript](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

## Throw

**throw**           **throw**       **throw**

```
if (denominator == 0) {  
    throw new DivideByZeroException();  
}
```

- **throw**

[Error](#) | [throw](#) | [try...catch](#)

---

[© 2000 Microsoft Corporation](#)

JScript

---

"/\*\*" "\*/"

/\*  
 \*/

- “\*/” .

JScript

---

[String](#) | [toString](#)

---

[© 2000 Microsoft Corporation](#)