

An A-Z Index of Windows VBScript commands

<code>Abs(number)</code>	Absolute (positive) value of <i>number</i>
<code>.AppActivate</code>	Activate running command
<code>.Application</code>	BrowseForFolder/Open
<code>Array(e11,e12,e13)</code>	Add values to an Array variable
<code>Arguments</code>	Command line arguments
<code>Asc(String)</code>	Return ASCII code for <i>string</i>
<code>AscB(String)</code>	Return the byte code for a character
<code>AscW(String)</code>	Return Unicode code for <i>string</i>
b	
Beep	- see <code>StdOut.Write</code>
<code>.BrowseForFolder</code>	Prompt the user to select a folder
c	
Call <i>subroutine (arguments) or subroutine arguments</i>	
<code>CBool(expression)</code>	Convert <i>expression</i> to Boolean (True/False) •
<code>CByte(expression)</code>	Convert <i>expression</i> to Byte (0-255) •
<code>CCur(expression)</code>	Convert <i>expression</i> to Currency (numeric) •
<code>CDate(expression)</code>	Convert <i>expression</i> to Date •
<code>CDBl(expression)</code>	Convert <i>expression</i> to Double (up to 1.79×10^{308}) •
<code>Chr(ChrCode)</code>	Return the string character for <i>ChrCode</i> (ASCII code)
<code>ChrB(ChrCode)</code>	Return the string character for <i>ChrCode</i> (Byte code)
<code>ChrW(ChrCode)</code>	Return the string character for <i>ChrCode</i> (Unicode/DBCS)
<code>CInt(expression)</code>	Convert <i>expression</i> to Integer •
<code>CLng(expression)</code>	Convert <i>expression</i> to long •
<code>CSng(expression)</code>	Convert <i>expression</i> to single •
<code>CStr(expression)</code>	Convert <i>expression</i> to a string •
<code>.CreateObject</code>	Create an automation object / run an external command
<code>.CreateShortcut</code>	Create Shortcut
<i>cscript</i>	Run a VBScript .vbs file
<code>.CurrentDirectory</code>	Retrieve or change the current directory
d	
Date()	The current system date
DateAdd	Add a time interval to a Date
DateDiff	Return the time interval between two dates
DatePart	Return part of a given date
DateSerial	Return a Date from a numeric Year, Month and Day
DateValue	Return a Date from a string expression
Day(date)	Return the day component of <i>date</i> (1-31)
Dim	Declare a new variable or array variable
Do..Loop	Repeat a block of statements
Drives	
Drive Map	<code>.MapNetworkDrive</code> - <code>.Enum</code> - <code>.Remove</code>
e	
Echo	Echo text to screen, also <code>StdOut.Write</code> / <code>.Popup</code>
End	End a procedure or code block
Environment Variables	- <code>Expand</code> - <code>.Delete/Remove</code>
Escape(String)	Return only ASCII characters from a Unicode string.
Eval(expr)	Evaluate an expression
Exec	Run a command
Execute	Execute one or more statements
Exp(n)	Return e (base of natural logs) raised to a power <i>n</i> .
Exit	Exit a block of code immediately
f	
FileSystemObject	Work with Drives, Folders and Files
Filter	Produce an array by filtering an existing array
Fix(number)	Return the integer portion of a number
For...Next	Repeat a block of statements a given number of times
For Each...	Loop through the items in a collection or array

	<code>FormatCurrency</code>	Format a number with a currency symbol
	<code>FormatNumber</code>	Format a number
	<code>FormatPercent</code>	Format a number with a % symbol
	<code>FormatDateTime</code>	Format a Date/Time value
	<code>Function</code>	Define a function procedure
g	<code>GetLocale()</code>	Return the Regional <code>LocaleID</code>
	<code>.GetObject</code>	Get an Automation object
h	<code>Hex(number)</code>	Return the hex value of <i>number</i>
	<code>Hour(time)</code>	Return the hour component of <i>time</i>
i	<code>If..Then</code>	Conditionally execute a block of statements
	<code>InputBox</code>	Prompt for user input
	<code>InStr</code>	Find one string within another
	<code>InStrRev</code>	Find one string within another, starting from the end
	<code>Int(number)</code>	Return the integer portion of a number
	<code>IsArray(varname)</code>	Is <i>varname</i> an array?
	<code>IsDate(expression)</code>	Is <i>expression</i> a <code>DateTime</code> ?
	<code>IsEmpty(expression)</code>	Is <i>expression</i> initialised?
	<code>IsNull(expression)</code>	Is <i>expression</i> NULL?
	<code>IsNumeric(expression)</code>	Is <i>expression</i> a <code>Numeric</code> ?
	<code>IsObject(expression)</code>	Is <i>expression</i> an object?
	<code>Join</code>	Combine the contents of an array into a single variable.
l	<code>LBound</code>	Return the smallest subscript for an array.
	<code>LCase(String)</code>	Return <i>String</i> in lower case
	<code>Left(String,len)</code>	Return the leftmost <i>len</i> characters of <i>string</i>
	<code>LeftB(String,len)</code>	Return the leftmost <i>len</i> bytes of <i>string</i>
	<code>Len(String)</code>	Return length of <i>string</i> in characters
	<code>LenB(String)</code>	Return length of <i>string</i> in Bytes
	<code>Log(number)</code>	Return natural log of <i>number</i>
	<code>.LogEvent</code>	Log an item in the Event log
	<code>LTrim(String)</code>	Remove any leading spaces from a string expression
m	<code>.MapNetworkDrive</code>	Drive Map
	<code>Mid</code>	Return a mid-section from a string
	<code>Minute(time)</code>	Return the minute component of <i>time</i>
	<code>Month(date)</code>	Return the month component of <i>date</i>
	<code>MonthName</code>	Convert a month number to a descriptive Month
	<code>MsgBox</code>	Display a dialogue box message
n	<code>.Network</code>	Access network resources
	<code>Now</code>	Return the current Date and Time
o	<code>On Error</code>	Error handling
	<code>Option Explicit</code>	Force all variables to be <i>defined</i>
p	<code>.AddPrinterConn / .AddWindowsPrConn</code>	Add Printer connection
	<code>.EnumPrinterConnections</code>	List Printer connections
	<code>.SetDefaultPrinter</code>	Set default printer
	<code>.RemovePrinterConnection</code>	Remove printer connection
	<code>Private VarName</code>	Declare a local <i>variable</i> /array variable
	<code>Public VarName</code>	Declare a public <i>variable</i> /array variable
q	<code>.Quit</code>	Quit
r	<code>Randomize(number)</code>	Initialise the random number generator
	<code>.ReadLine</code>	Accept user text input

ReDim	Resize a dynamic array
RegExp	Regular expression search object
.RegDelete	Registry, delete
.RegRead	Registry, read
.RegWrite	Registry, write
REM	Add a Comment
Replace	Find and replace text
RGB(<i>Red, Green, Blue</i>)	Return a system color code
Right(<i>String, len</i>)	Return the rightmost <i>len</i> characters of <i>string</i>
RightB(<i>String, len</i>)	Return the rightmost <i>len</i> bytes of <i>string</i>
Rnd	Return a random number
Round	Round a number to <i>n</i> decimal places
RTrim(<i>String</i>)	Remove any trailing spaces from a string expression
.Run a command	
S	
Second(<i>time</i>)	Return the second component of <i>time</i>
Select...Case	Conditionally execute a block of statements
.SendKeys	
Set <i>variable</i> = <i>object</i>	Assign an object reference
SetLocale(<i>LocaleID</i>)	Set the Regional LocaleID
Sgn(<i>number</i>)	The sign of a number
.Shell + Shell.Application	objects/methods
.ShellExecute	Run a script/application in the Windows Shell
Shortcut create .CreateShortcut	
.Sleep	
Space	Return a string consisting of spaces.
.SpecialFolders	Path to Desktop, Start menu, Programs...
Split	Parse a string of delimited values into an array
Sqr(<i>number</i>)	Square root
StdOut.Read	Obtain User Input from the command prompt
StdOut.Write	Echo text to the screen, also beep.
StrComp	Compare two strings
String	Create a string with a single character repeated
StrReverse	Reverse a string
Sub	Declare a sub procedure
t	
Time()	The current system time
Timer()	The number of seconds since midnight
TimeSerial	Construct a Time value from Hours, Minutes and seconds
TimeValue	Convert a string to a Time value
Trim(<i>String</i>)	Remove leading and trailing spaces from <i>string</i>
TypeName	Return variable type (variant subtype).
u	
UBound	Return the largest subscript for an array dimension
UCase(<i>String</i>)	Uppercase <i>String</i>
UnEscape(<i>Str</i>)	Return Unicode characters from an escaped ASCII string
w	
Weekday	Return the day of the week (1-7)
WeekdayName	Return the named day of the week
While...Wend	Conditionally repeat a block of statements
With	Assign multiple properties of an object
Year(<i>date</i>)	Return the year component of <i>date</i> (1-12)
' Remark	Add a Comment

• All VBScript variables are **variants** the conversion functions set the variant subtype (an internal flag indicating the **DataType**) .

Syntax and Examples

[Docs.Microsoft.com - VB Script language reference](https://docs.microsoft.com/en-us/previous-versions/vbnet-10/aa904436(v=vs.100)?r=65)
Other sources of help.