

PowerShell Cheat Sheet

Variables

\$var = "string"	Assign variable
[Type]\$var="typedVar"	Assign strong typed variable
[ValidateRange(1,9)][int]\$x=1	Assign strong typed attribute controlled variable
\$a,\$b,\$c = 0 or \$a,\$b = 'a','b'	Assign multiple variables
\$a,\$b = \$b,\$a	Flip variables
Scopes	global, local, private or script
\$global:var = "var"	Assign global scoped variable

Arrays

"a", "b", "c"	Array of strings
@{}	Empty array
1,(2,3),4	Array within array
,"hi"	Array of one element
\$arr[5]	Sixth element of array
\$arr[2..20]	Return elements 3 thru 21
\$arr[-1]	Return last array element
\$arr[-3...-1]	Display last three elements of array
\$arr[1,4+6..9]	Elements at index positions 1,4, 6 to 9
@(Get-Process)	Force result to an array
\$arr[((\$arr.length-1)..0]	Reverse array
\$arr[1] += 200	Add to existing array item value
\$b = \$arr[0,1 + 3..6]	New array from elements of \$arr array
\$z = \$arrA + \$arrB	Combine two arrays into single array

Associative Arrays (Hash tables)

\$hash = @{}	Create empty hash table
@{foo=1; bar=value2}	Create, initialize hash table
[ordered]@{a=1; b=2; c=3}	Create ordered dictionary
\$hash.key1 = 1	Assign 1 to key key1
\$hash.key1	Return value of key1
\$hash["key1"]	Return value of key1
\$hash.GetEnumerator sort Key	Sort hash table by Key
[pscustomobject]@{x=1;z="z"}	Create custom object

Strings

"\$var expand"	String with expansion "
'\$var no expand'	String with no expansion '
@"	Here-String - quotes, expressions, etc. Single quotes for no expressions
@	Here-String
"@"	Here-String

Comments, Escape Characters, Backtick

#Comment	Comment
<# comment #>	Multiline comment
"A \"test\""	Escape char `
`t	Tab
`n	New line
`	Line continuation

Basics of Text and Files

Get-Location	Get current directory
Set-Location	Change directory
Get-Content	Get content of file
Add-Content	Append content
Set-Content	Set content of file
Out-File	Formatted text to file
Out-Null	Discard output
Out-String	Convert to strings
Copy-Item	Copy items
Remove-Item	Remove items
Move-Item	Move items
Rename-Item	Rename item
Set-Item	Set contents of file
Clear-Item	Clear contents of file
New-Item	New empty file or dir

Objects

(Get-Date).Date	Date property of object from Get-Date
Get-Date Get-Member	List properties and methods of object
[DateTime]::Now	Static properties referenced with "::"
"string".ToUpper()	Use ToUpper() Method on string
[system.Net.Dns]::GetHostByAddress("127.0.0.1")	Use static method to get host name with "::"
\$excel = new-object -com excel.application	Create a new Excel COM object to work with

Flow Control

If(\$x -eq 5){} Elseif(\$x -gt 5){} Else{ }	If
\$x = 1; while(\$x -lt 10){\$x;\$x++}	While
For(\$i=0; \$i -lt 10; \$i++){ \$i }	For
Foreach(\$file in dir C:\){\$file.Name}	Foreach
1..10 foreach(\$_)	Foreach
Switch -options (<values to switch on>){ PatternX {statement} Default {Default Statement} }	Switch

Assignment, Logical, Comparison Operators

+=, +=, -=, *=, /=, %%, ++, --	Assign one or more values to variable
-and, -or, -xor, -not, !	Connect expressions / statements
-eq, -ne	Equal, not equal
-gt, -ge	Greater than, greater than or equal
-lt, -le	Less than, less than or equal to
-replace	Replacement - "Hi" -replace "H", "P"
-match, -notmatch	Regular expression match
-like, -notlike	Wildcard matching
-contains, -notcontains	TRUE if value on right in array on left
-in, -notin	Reverse of contains, notcontains

Other Operators

-Split	Split a string "abcdefghi" -split "de"
-join	Joins multiple strings "abc", "def" -join ","
..	Range operator 1..10 foreach { \$_ * 5 }
-is, -isnot	Boolean - is object instance of specified .NET type
-as	Convert input object to specified .NET type
-f	Format strings 1..10 foreach { "{0:N2}" -f \$_ }
[]	Cast operator. [datetime]\$birthday = "1/10/66"
\$()	Subexpression operator
@()	Array subexpression operator
&	The call/invoke operator.

Filter, Sort, Group and Format (aliases for brevity)

dir C:\pub where-object LastWriteTime -gt (Get-Date).addDays(-1)	Files in C:\pub with lastwritetime greater than yesterday
ps where-object {\$_.path -like "C:\windows\system32*" -and \$_.company -notlike "Microsoft"}	Processes where path includes system32 and company doesn't start with Microsoft
ps Explorer select-object -Property ProcessName -ExpandProperty Modules format-list	Get explorer processes, select processname, expand modules property array
ps Sort-Object -Property WorkingSet Select-Object -Last 5	Sort Processes by workingset, select last 5
"a","b","a" Select-Object -Unique	Return only unique - returns @(a b)
Get-Service Group-Object Status	Group services by their Status
dir Group-Object {\$_ .Length -gt 100KB}	Group objects bigger/smaller than 100 KB
Get-Content C:\pcs.txt Select-String "q-" sls "win7"	Select strings with "q-", "win7" from pcs.txt
ps Format-Table -Property Name, StartTime -AutoSize	Format ps output showing Name, StartTime properties, autosize the table
ps Format-table ProcessName, @{ Label = "Total Run Time"; Expression={(Get-Date) - \$_.StartTime}}	Table showing processname, custom label/expression showing run time.
Get-EventLog -Log System Select -first 5 Format-table -wrap	Get first 5 events in system log, wrap display
gci C:\Users format-list -property *	Get all properties from C:\Users in list format
"{0}\t{1}n" -f \$a, \$b	-f operator to construct strings. {0} replaced with \$a, {1} with \$b etc.

Common commands

Get-EventLog	Get-WinEvent
Get-CimInstance	Get-Date
Start-Sleep	Compare-Object
Start-Job	Get-Credential
Test-Connection	New-PSSession
Test-Path	Split-Path

Importing, Exporting and Converting

Export-CliXML	Import-CliXML
ConvertTo-XML	ConvertTo-HTML
Export-CSV	Import-CSV
ConvertTo-CSV	ConvertFrom-CSV

Automatic variables

\$_ \$PSItem	Current pipeline object
\$Args	Script or function arguments
\$Error	Errors from commands
\$True,\$False	Boolean value for true,false
\$null	Empty
\$profile	Array of profile locations

PSDrives

Alias:	Aliases in current session
Cert:	Certificate store for user
Env:	Environment variables
Function:	All functions in current session
HKLM:	Hkey Local Machine Hive
HKCU:	Hkey Current User Hive
Variable:	Variables in the current session
WSMan:	WinRM configuration / credentials
AD:	Active Directory
Set-location HKLM:	HKLM Registry hive
gci variable:	Variables in current session

Regular Expressions

\w	Any word character [a-zA-Z0-9]
\W	Any non-word character
\s	Any whitespace character
\S	Any non-whitespace character
\d \D	Any digit or non-digit
{n} {n,} {n,m}	Match n through m instances of a pattern.
More	Google .NET Regular Expressions