PowerShell 7 Cheat Sheet

FINDING CMDLETS AND HELP

Get-Command	List available commands. Use -Module, -Noun, -Verb. Wildcards help too
Get-Member	List properties and methods of an object
Get-Help	Help for a command. Use -Online to get latest

WORKING WITH OBJECTS

Common pattern: Get | Filter/Group/Sort | Modify/Delete/Output/Convert

Where-Object	Filters objects based on value of property
Select-Object	Choose properties of an object to include in pipeline
Group-Object	Group based on property values
Sort-Object	Sort results by property values
Foreach-Object	Act on each object in pipeline
-Parallel	Act on each object in pipeline at the same time
Measure-Object	Measure property values or number of objects

BUILT-IN VARIABLES

$Args	Arguments passed into script.
$error	Array of errors. $Error[0] is latest.
$host	Details on application running PS
$IsLinux	Returns TRUE on Linux OS
$isMacos	Returns TRUE on Mac OS
$IsWindows	Returns TRUE on Windows OS
$Profile	Path to PowerShell profiles
$PSBoundParameterValues	List parameters and current values.
$PSCommandPath	Full path of script being run
$PSItem / $_	Current object in the pipeline
$PSScriptRoot	Directory the script is run from
$PSVersionTable	Details on PowerShell version

OPERATORS

Pipeline	|, **?? (If error), && (If success)**
Arithmetic	+, -, *, /, %
Assignment	=, +=, -=, *=, /=, %=
Null Coalescing	??
Comparison	-eq, -ne, -gt, -lt, -le, -ge
Wildcard Compare	-like, -notlike
Regex Compare	-match, -notmatch, -replace
Contain Comparison	-in, -notin, -contains, -notcontains
Logical	-and, -or, -xor, -not, !
Type	-is, -isnot, -as
Ternary	(statement) ? (if true) : (if false)

USEFUL CMDLETS

How To...	Commands
Zip Files	Compress-Archive, Expand-Archive
Date/Time	Get-Date, Set-Date, Get-TimeZone, Set-TimeZone
Event Logs	Get-WinEvent, New-WinEvent
Performance	Get-Counter, Export-Counter, Import-Counter
Clipboard	Get-Clipboard, Set-Clipboard
Reboot	Restart-Computer
Send Output	Out-Printer, Out-Null, Out-File
User Input	Read-Host
Use Jobs	Start-Job, Stop-Job, Get-Job, Receive-Job, Remove-Job
Wait	Start-Sleep
Map Drives	Get-PSDrive, New-PSDrive, Remove-PSDrive
Navigate	Get-Location, Set-Location, Test-Path
File/Folders	New-Item, Get-Item, Get-ChildItem, Get-Content, Set-Content Move-Item, Rename-Item, Copy-Item, Remove-Item
Display in GUI form	Out-Gridview (with -OutputMode and -Passthru) to select one or more items and return to shell

COMMON PARAMETERS

-WHATIF	Don't make the changes, but output what would
-CONFIRM	Prompt before making changes
-VERBOSE	Display verbose output
-DEBUG	Display debug-level output
-ERRORACTION	Override $ErrorActionPreference variable
-OUTVARIABLE	Redirect output to a variable
-?	Display help for the cmdlet

KEYBOARD SHORTCUTS

Esc	Clear line
Tab	Complete partially entered cmdlet/parameter
CTRL+C	Stop processing current command
Up/Down Arrow	Navigate command history
CTRL+S/CTRL+R	Search forward/reverse through history
CTRL+ALT+?	Show all keybindings

STRINGS

How To...	Commands/Examples
Grep / Search Text	Select-String
Split into array	"one,two,three" **-split ","**
Join	"one", "two", "three" **-join ", and a "**
Replace	"http://mysite.com" **-replace "http:","https:"**
# Decimal Places	"Pi is **{0:N2}**" **-f [Math]::Pi**
Format Currency	"The price is **{0:C}**" **-f 1.23**
Format All Caps	$a = "I'm yelling"; **$a.ToUpper()**
Format All Lower	$a = "TOO LOUD"; **$a.ToLower()**
Insert Characters	$a = "abcghij"; **$a.Insert(3,"def")**

LOOPS & BRANCHES

Loop/Branch	How To Use
If/Then/Else	If ($true) { <this> } else { <that> }
For	For ($a=0; $a -lt 10; $a++) { <this> }
Do…While	Do { <this> } While ($evaluation)
Do…Until	Do { <this> } Until ($evaluation)
While	While ($evaluation) { <this> }
Foreach	Foreach ($a in $b) { <this $a> }
Switch	Switch ($a) { "one" { <this happens if $a is "one"> } Default { <this $a is none of above>} }

ARRAYS

How To...	Commands
Create Array	$a = @()
Single Item Array	$a = @("one")
Item Reference	$a[index] (0 is first)
Range Reference	$a[0..4] (Returns first 5)
Last Item Reference	$a[-1]

MODULES AND PACKAGES

Find-Module	Search PSGallery for PowerShell modules
Find-Package	Search PSGallery, nuget.org for software
Get-Module	Find modules/packages installed on system
Get-Package	Software installed by package management
Other Verbs	Install, Uninstall, Update
Register-PackageSource	Allow package sources for installation
Install-PackageProvider	Allow additional package providers (Gist) or specific versions

ADVANCED FUNCTION PARAMETER ATTRIBUTES

Mandatory	Will prompt if missing
Position	Allows params in order instead of by name
ValueFromPipeline	Allows pipeline input to parameter
ValueFromPipelineByPropertyName	Pipeline accepted if property name matches
HelpMessage	Sets param help msg
ValidateSet("Choice1","Choice2")	Gives choice, allows tab-complete
ValidateScript({ evaluation script })	Processes an evaluation script
ValidateRange([1..10])	Enforces param values in range