

# PowerShell Basic Cheat Sheet

PowerShell is a task based command line shell and scripting language. To run it, click Start, type PowerShell, run PowerShell ISE or PowerShell as Administrator. Commands are written in verb-noun form, and named parameters start with a dash.

## Basics

Cmdlet	Commands built into shell written in .NET
Functions	Commands written in PowerShell language
Parameter	Argument to a Cmdlet/Function/Script
Alias	Shortcut for a Cmdlet or Function
Scripts	Text files with .ps1 extension
Applications	Existing windows programs
Pipelines	Pass objects Get-process word   Stop-Process
Ctrl+c	Interrupt current command
Left/right	Navigate editing cursor
Ctrl+left/right	Navigate a word at a time
Home / End	Move to start / end of line
Up/down	Move up and down through history
Insert	Toggles between insert/overwrite mode
F7	Command history in a window
Tab / Shift-Tab	Command line completion

## Help

Get-Command	Get all commands
Get-Command -Module RGHS	Get all commands in RGHS module
Get-Command Get-p*	Get all commands starting with get-p
Get-help get-process	Get help for command
Get-Process   Get-Member	Get members of the object
Get-Process  format-list -properties *	Get-Process as list with all properties

## Writing output and reading input

"This displays a string"	String is written directly to output
Write-Host "color" -ForegroundColor Red -NoNewLine	String with colors, no new line at end
\$age = Read-host "Please enter your age"	Set \$age variable to input from user
\$pwd = Read-host "Please enter your password" -asSecureString	Read in \$pwd as secure string
Clear-Host	Clear console

Example command: dir C:\users\example -recurse -File | ?{\$\_.LastWriteTime -gt [datetime]::Today} | Select LastWriteTime,CreationTime,Length,FullName | sort LastWriteTime -descending | ft -AutoSize  
This gets all files under C:\users\example, filters by lastwritetime today, only returns lastwritetime, creationtime, length and fullname, sorts by lastwritetime and outputs results in an autosized table

## Variables

\$var = "string"	Assign variable
\$a,\$b = 0 or \$a,\$b = 'a','b'	Assign multiple variables
\$a,\$b = \$b,\$a	Flip variables
\$var=[int]5	Strongly typed variable

## Assignment, Logical, Comparison Operators

=,+=,-=,++,--	Assign values to variable
-and,-or,-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
-replace	"Hi" -replace "H", "P"
-match,-notmatch	Regular expression match
-like,-notlike	Wildcard matching
-contains,-notcontains	Check if value in array
-in, -notin	Reverse of contains,notcontains.

## Parameters

-Confirm	Prompt whether to take action
-Whatif	Displays what command would do

## Cmdlets

Get-EventLog	Get-WinEvent
	Get-Date
Start-Sleep	Compare-Object
Start-Job	Get-Credential
Test-Connection	New-PSsession
Test-Path	Split-Path
Get-ADUser	Get-ADComputer
Get-History	New-ISESnippet
Get-WMIObject	Get-CimInstance

## Arrays, Objects

\$arr = "a", "b"	Array of strings
\$arr = @()	Empty array
\$arr[5]	Sixth array element
\$arr[-3..-1]	Last three array elements
\$arr[1,4+6..9]	Elements at index 1,4, 6-9
\$arr[1] += 200	Add to array item value
\$z = \$arA + \$arB	Two arrays into single array
[pscustomobject]@{x=1;z=2}	Create custom object
(Get-Date).Date	Date property of object

## Scripts

Set-ExecutionPolicy -ExecutionPolicy Bypass	Set execution policy to allow all scripts
."\\c-is-ts-91\c\$\scripts\script.ps1"	Run Script.PS1 script in current scope
&"\\c-is-ts-91\c\$\scripts\script.ps1"	Run Script.PS1 script in script scope
.\Script.ps1	Run Script.ps1 script in script scope
\$profile	Your personal profile that runs at launch

## Importing, Exporting, Converting

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

## Flow Control

If(){ } Elseif(){ } Else{ }
while(){ }
For(\$i=0; \$i -lt 10; \$i++){ }
Foreach(\$file in dir C:){\$file.name}
1..10   foreach{\$_}

## Comments, Escape Characters

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

## Aliases for common commands

Gcm	Get-Command
Foreach,%	Foreach-Object
Sort	Sort-Object
Where,?	Where-Object
Diff,compare	Compare-Object
Dir, ls, gci	Get-ChildItem
Gi	Get-Item
Copy,cp,cpi	Copy-Item
Move,mv,mi	Move-Item
Del,rm	Remove-Item
Rni,ren	Rename-Item
Ft	Format-Table
Fl	Format-List
Gcim	Get-CimInstance
Cat,gc,type	Get-Content
Sc	Set-Content
h,history,ghy	Get-History
Ihy,r	Invoke-History
Gp	Get-ItemProperty
Sp	Set-ItemProperty
Pwd,gl	Get-Location
Gm	Get-Member
Sls	Select-String
Cd,chdir,sl	Set-Location
Cls,clear	Clear-Host

## Cmdlets

Set-Location
Get-Content
Add-Content
Set-Content
Out-File
Out-String
Copy-Item
Remove-Item
Move-Item
Set-Item
New-Item