PowerShell Commands Cheat Sheet

PowerShell - Files and Folders

Task	Command	Example
Creating Folders	New-Item -ItemType Directory <foldername></foldername>	New-Item -ItemType Directory "C:\NewFolder"
Creating Files	New-Item -ItemType File <filename></filename>	New-Item -ItemType File "C:\test.txt"
Copying Folders	Copy-Item -Path <source/> -Destination <dest> -Recurse</dest>	Copy-Item -Path "C:\Folder1" -Destination "D:\Backup" -Recurse
Copying Files	Copy-Item <sourcefile> <destination></destination></sourcefile>	Copy-Item "C:\file.txt" "D:\backup.txt"
Deleting Folders	Remove-Item -Path <foldername> -Recurse -Force</foldername>	Remove-Item -Path "C:\OldFolder" -Recurse -Force
Deleting Files	Remove-Item <filename></filename>	Remove-Item "C:\test.txt"
Moving Folders	Move-Item <sourcefolder> <destination></destination></sourcefolder>	Move-Item "C:\Folder1" "D:\NewLocation"
Moving Files	Move-Item <sourcefile> <destination></destination></sourcefile>	Move-Item "C:\test.txt" "D:\backup.txt"
Rename Folders	Rename-Item <oldname> <newname></newname></oldname>	Rename-Item "C:\OldFolder" "C:\NewFolder"
Rename Files	Rename-Item <oldname> <newname></newname></oldname>	Rename-Item "C:\old.txt" "C:\new.txt"
Retrieving Item	Get-Item <path></path>	Get-Item "C:\file.txt"
Check Folder Existence	Test-Path <foldername></foldername>	Test-Path "C:\Users" (Returns \$true or \$false)
Check File Existence	Test-Path <filename></filename>	Test-Path "C:\test.txt"

PowerShell - Dates and Timers

Task	Command	Example
Get System Date	Get-Date	Get-Date (Displays current date & time)
Set System Date	Set-Date -Date " <mm dd="" yyyy="">"</mm>	Set-Date -Date "02/18/2025"
Get System Time	(Get-Date).TimeOfDay	(Get-Date).TimeOfDay
Set System Time	Set-Date -Date " <mm dd="" hh:mm="" yyyy="">"</mm>	Set-Date -Date "02/18/2025 14:30"

PowerShell - Files I/O

Task	Command	Example
Create Text File	New-Item -ItemType File <filename></filename>	New-Item -ItemType File "C:\test.txt"
Read Text File	Get-Content <filename></filename>	Get-Content "C:\test.txt"
Create XML File	[xml]\$xml = " <root></root> "; \$xml.Save(" <filename>")</filename>	[xml]\$xml = " <root></root> "; \$xml.Save("C:\data.xml")
Read XML File	[xml]\$xml = Get-Content <filename></filename>	[xml]\$xml = Get-Content "C:\data.xml"
Create CSV File	New-Item -ItemType File <filename>.csv</filename>	New-Item -ItemType File "C:\data.csv"
Read CSV File	Import-Csv <filename></filename>	Import-Csv "C:\data.csv"
Create HTML File	New-Item -ItemType File <filename>.html</filename>	New-Item -ItemType File "C:\index.html"
Read HTML File	Get-Content <filename>.html</filename>	Get-Content "C:\index.html"
Erasing File Content	Set-Content <filename> ""</filename>	Set-Content "C:\test.txt" ""
Append Text Data	Add-Content <filename> "Text"</filename>	Add-Content "C:\test.txt" "Hello World!"

PowerShell - Advanced Cmdlets

Cmdlet	Description	Example
Get-Unique	Get unique values from input	`Get-Content file.txt
Group-Object	Group objects by property	`Get-Process
Measure-Object	Perform calculations on objects	`Get-ChildItem
Compare-Object	Compare two sets of objects	Compare-Object (Get-Content File1.txt) (Get-Content File2.txt)
Format-List	Format output as a list	`Get-Service
Format-Wide	Format output in columns	`Get-Command
Where-Object	Filter output	`Get-Process
Get-ChildItem	Get files & directories	Get-ChildItem "C:\Users"
ForEach-Object	Iterate through each object	`Get-Process
Start-Sleep	Pause script execution	Start-Sleep -Seconds 5
Read-Host	Prompt user for input	\$name = Read-Host "Enter your name"
Select-Object	Select specific properties	`Get-Process

Sort-Object	Sort output	`Get-Process
Write-Warning	Display warning message	Write-Warning "Low disk space!"
Write-Host	Print output	Write-Host "Hello, World!" -ForegroundColor Green
Invoke-Item	Open a file or folder	Invoke-Item "C:\test.txt"
Invoke-Expression	Run a command string	Invoke-Expression "Get-Process"
Measure-Command	Measure execution time	Measure-Command { Get-Process }
Invoke-History	Run a command from history	Invoke-History 3
Add-History	Add command to history	Add-History -InputObject "Get-Process"
Get-History	Show command history	Get-History
Get-Culture	Get system language settings	Get-Culture

Getting Help

Task	Command	Example
Get help for a command	Get-Help <cmdlet></cmdlet>	Get-Help Get-Service
Show examples	Get-Help <cmdlet> -Examples</cmdlet>	Get-Help Get-Process -Examples
Show detailed help	Get-Help <cmdlet> -Detailed</cmdlet>	Get-Help Get-Service -Detailed
List available commands	Get-Command	Get-Command
Find cmdlets by keyword	Get-Command *service*	Get-Command *disk*

System Information

Task	Command	Example
Get system information	Get-ComputerInfo	Get-ComputerInfo
Get OS details	Get-WmiObject Win32_OperatingSystem	Get-WmiObject Win32_OperatingSystem
Get system manufacturer	Get-WmiObject Win32_ComputerSystem	Get-WmiObject Win32_ComputerSystem
List running processes	Get-Process	Get-Process
List installed updates	Get-Hotfix	Get-Hotfix

File and Folder Management

Task	Command	Example
List files/folders	Get-ChildItem <path></path>	Get-ChildItem C:\Users
Create a file	New-Item -ItemType File <filename></filename>	New-Item -ItemType File "C:\test.txt"
Create a folder	New-Item -ItemType Directory <foldername></foldername>	New-Item -ItemType Directory "C:\NewFolder"
Copy a file	Copy-Item <source/> <destination></destination>	Copy-Item "C:\file.txt" "D:\backup.txt"
Move a file	Move-Item <source/> <destination></destination>	Move-Item "C:\file.txt" "D:\backup.txt"
Delete a file	Remove-Item <filename></filename>	Remove-Item "C:\test.txt"
Delete a folder	Remove-Item <foldername> -Recurse -Force</foldername>	Remove-Item "C:\OldFolder" -Recurse -Force

User and Group Management

Task	Command	Example
List local users	Get-LocalUser	Get-LocalUser
Create a new user	New-LocalUser -Name <username> -Password <securestring></securestring></username>	New-LocalUser -Name "TestUser" -Password (ConvertTo-SecureString "P@ssword123" -AsPlainText -Force)
Delete a user	Remove-LocalUser -Name <username></username>	Remove-LocalUser -Name "TestUser"
List local groups	Get-LocalGroup	Get-LocalGroup
Add user to a group	Add-LocalGroupMember -Group <group> -Member <user></user></group>	Add-LocalGroupMember -Group "Administrators" -Member "TestUser"

Networking

Task	Command	Example
Get IP address	Get-NetIPAddress	Get-NetIPAddress
Ping a host	Test-Connection <host></host>	Test-Connection google.com
Get network adapter info	Get-NetAdapter	Get-NetAdapter
Show active connections	Get-NetTCPConnection	Get-NetTCPConnection
Resolve a DNS name	Resolve-DnsName <domain></domain>	Resolve-DnsName google.com

Managing Processes & Services

Task	Command	Example
List running processes	Get-Process	Get-Process
Kill a process	Stop-Process -Name <processname> -Force</processname>	Stop-Process -Name "notepad" -Force
Start a process	Start-Process < AppName >	Start-Process "notepad.exe"
List all services	Get-Service	Get-Service
Start a service	Start-Service -Name <servicename></servicename>	Start-Service -Name "wuauserv"
Stop a service	Stop-Service -Name <servicename></servicename>	Stop-Service -Name "wuauserv"
Restart a service	Restart-Service -Name <servicename></servicename>	Restart-Service -Name "wuauserv"

Windows Updates

Task	Command	Example
Check for updates	Get-WindowsUpdate	Get-WindowsUpdate
Install updates	Install-WindowsUpdate -AcceptAll	Install-WindowsUpdate -AcceptAll

Disk Management

Task	Command	Example	
List all volumes	Get-Volume	Get-Volume	
List all disks	Get-Disk	Get-Disk	
Show partitions	Get-Partition	Get-Partition	
Wipe a disk (Caution!)	Clear-Disk -Number <disknumber> -Confirm:\$false</disknumber>	Clear-Disk -Number 1 -Confirm:\$false	

Task Scheduler

Task	Command	Example	
List scheduled tasks	Get-ScheduledTask Get-ScheduledTask		
Create a new task	Register-ScheduledTask -TaskName "MyTask" -Trigger (New- ScheduledTaskTrigger -At 12:00PM -Daily) -Action (New- ScheduledTaskAction -Execute "notepad.exe") -User "SYSTEM" - RunLevel Highest -Force		

Security & Permissions

Task	Command	Example	
Check execution policy	Get-ExecutionPolicy	Get-ExecutionPolicy	
Set execution policy Set-ExecutionPolicy RemoteSigned		Set-ExecutionPolicy RemoteSigned	

Active Directory (For Domain Admins)

Task	Command	Example	
Get all AD users	Get-ADUser -Filter *	Get-ADUser -Filter *	
Create an AD user	New-ADUser -Name "John Doe" -SamAccountName jdoe - UserPrincipalName jdoe@domain.com -Path "OU=Users,DC=domain,DC=com" -Enabled \$true		
Delete an AD user	Remove-ADUser -Identity jdoe	Remove-ADUser -Identity jdoe	

Remote Sessions

Task	Command	Example
Start a remote session	Enter-PSSession -ComputerName <server> Enter-PSSession -ComputerName Server01</server>	
Run a command remotely	Invoke-Command -ComputerName <server> -ScriptBlock { Get- Service }</server>	

Operators

+ Addition 5 + 3 8 Arithmetic -* Multiplication 10 - 4 6	Operator	Description	Example	Output	Category
* Multiplication 6 * 3 18 / Division 12 / 4 3 % Modulus (Remainder) 10 % 3 1 ++ Increment \$x-5; \$x++ 6 Decrement \$x-5; \$x- 4 Assign value \$a = 10 \$a is 10 Assignment += Add & assign \$a + 5 \$a becomes 7 -= Subtract & assign \$a + 2 \$a becomes 7 -= Multiply & assign \$a + 2 \$a becomes 20 -= Pulvide & assign \$a / 4 \$a becomes 20 -= Pulvide & assign \$a / 4 \$a becomes 20 -= Pulvide & assign \$a / 4 \$a becomes 20 -= Pulvide & assign \$a / 4 \$a becomes 20 -= Equal to \$ -9 + 95 True Comparison	+	Addition	5 + 3	8	Arithmetic
f Division 12 / 4 3 % Modulus (Remainder) 10 % 3 1 ++ Increment \$x=5; \$x++ 6 Decrement \$x=5; \$x 4 Decrement \$x=5; \$x 4 Assign value \$a = 10 \$a is 10 Assignment += Add & assign \$a + 5 \$a becomes 15 Subtract & assign \$a - 3 \$a becomes 20 Divide & assign \$a * 2 \$a becomes 2.5 Divide & assign \$a / 4 \$a becomes 2.5 Polivide & assign \$a / 4 \$a becomes 2.5 Polivide & assign \$a / 4 \$a becomes 2.5 Polivide & assign \$a / 4 \$a becomes 2.5 True Comparison To Not equal to \$ -eq 5 True Comparison Greater than 10 -gt 3 True Logical True Logical <tr< td=""><td>-</td><td>Subtraction</td><td>10 - 4</td><td>6</td><td></td></tr<>	-	Subtraction	10 - 4	6	
% Modulus (Remainder) 10 % 3 1 ++ Increment \$x=5; \$x++ 6 Decrement \$x=5; \$x 4 = Assign value \$a = 0 \$a is 10 Assignment += Add & assign \$a + 5 \$a becomes 15 -= Subtract & assign \$a - 3 \$a becomes 20 -* Multiply & assign \$a / 4 \$a becomes 2.5 -eq Equal to \$-eq 5 True Comparison -ne Not equal to \$-eq 5 True Comparison -ne Not equal to \$-eq 5 True Comparison -ne Not equal to \$-eq 5 True Comparison -reg Greater than 10 -gt 5 True True -reg Greater than 2 -lt 4 True Logical -reg Greater than or equal 3 -le 5 True Logical -reg Greater than or equal 3 -le 5 True Logical	*	Multiplication	6 * 3	18	
++ Increment \$x=5; \$x+- 4 Decrement \$x=5; \$x 4 = Assign value \$a = 10 \$a is 10 Assignment += Add & assign \$a + 5 \$a becomes 15 -= Subtract & assign \$a + 2 \$a becomes 20 += Multiply & assign \$a * 2 \$a becomes 20 -= Polvide & assign \$a / 2 \$a becomes 20 e Equal to \$ - eq \$ \$a becomes 2.5 e True Comparison -ne Not equal to \$ - eq \$ True Comparison -ne Not equal to \$ - ne 3 True Comparison -ne Not equal to \$ - ne 3 True Comparison -it Less than 2 - 14 True True -ie Greater than or equal \$ - eg 5 True Logical -or Logical AND (5 - gt 10) - or (3 - 14 4) True Logical -not Logical NOT	/	Division	12 / 4	3	
Decrement \$x=5; \$x 4 = Assign value \$a = 10 \$a is 10 Assignment += Add & assign \$a += 5 \$a becomes 15 -= Subtract & assign \$a -= 3 \$a becomes 20 += Multiply & assign \$a *= 2 \$a becomes 20 /- Divide & assign \$a /= 4 \$a becomes 2.5 -eq Equal to 5 -eq 5 True Comparison -ne Not equal to 5 -eq 5 True Comparison -ne Not equal to 5 -eq 5 True Comparison -it Less than 2 -tt 4 True True -it Less than or equal 5 -eg 5 True Logical -and Logical AND (5 -eg 10) -or (3 -it 4) True Logical -or Logical OR (5 -eg 10) -or (3 -it 4) True String -iike Wildcard mismatch "PowerShell"-inet "Power*" True String -motth Regex match <	%	Modulus (Remainder)	10 % 3	1	
= Assign value \$a = 10 \$a is 10 Assignment += Add & assign \$a += 5 \$a becomes 15 -= Subtract & assign \$a -= 3 \$a becomes 7 *= Multiply & assign \$a *= 2 \$a becomes 20 -eq Equal to 5 -eq 5 True Comparison -ne Not equal to 5 -eq 5 True Comparison -ne Not equal to 5 -ne 3 True Comparison -ne Not equal to 5 -ne 3 True Comparison -it Less than 2 -lt 4 True Comparison -it Less than or equal 5 -ge 5 True Logical -ie Less than or equal (5 -gt 2) -and (3 -lt 4) True Logical -or Logical AND (5 -gt 2) -and (3 -lt 4) True String -in Logical OR (5 -gt 10) -or (3 -lt 4) True String -ilike Wildcard match "PowerShell"-ilike "Power*" True String	++	Increment	\$x=5; \$x++	6	
+= Add & assign \$a += 5 \$a becomes 15 -= Subtract & assign \$a -= 3 \$a becomes 7 *= Multiply & assign \$a *= 2 \$a becomes 20 /= Divide & assign \$a /= 4 \$a becomes 2.5 -eq Equal to 5 -eq 5 True Comparison -ne Not equal to 5 -eq 5 True Comparison -ne Not equal to 5 -ne 3 True True -le Greater than 10 -gt 5 True True -lt Less than 2 -lt 4 True True -le Less than or equal 5 -ge 5 True Dejical -nad Logical AND (5 -gt 2) -and (3 -lt 4) True Logical -not Logical OR (5 -gt 10) -or (3 -lt 4) True String -not Logical NOT -not (5 -eq 5) False -like Wildcard match "PowerShell" -like "Power*" True String -notlike Wildcard mismatch		Decrement	\$x=5; \$x	4	
= Subtract & assign \$a -= 3 \$a becomes 7 *= Multiply & assign \$a *= 2 \$a becomes 20 /= Divide & assign \$a /= 4 \$a becomes 2.5 -eq Equal to 5 -eq 5 True Comparison -ne Not equal to 5 -ne 3 True Comparison -gt Greater than 10 -gt 5 True True -lt Less than 2 -lt 4 True True -le Less than or equal 5 -ge 5 True Logical -and Logical AND (5 -gt 2) -and (3 -lt 4) True Logical -or Logical OR (5 -gt 10) -or (3 -lt 4) True Logical -not Logical NOT -not (5 -eq 5) False False -like Wildcard match "PowerShell" -inke "Power*" True String -notlike Wildcard mismatch "PowerShell" -notlike "Win*" True True -notmatch Regex mismatch "abc" -notmatch "\d+" "True Arr	=	Assign value	\$a = 10	\$a is 10	Assignment
**= Multiply & assign \$a *= 2 \$a becomes 20 /= Divide & assign \$a /= 4 \$a becomes 2.5 -eq Equal to 5 -eq 5 True Comparison -ne Not equal to 5 -ne 3 True	+=	Add & assign	\$a += 5	\$a becomes 15	
/= Divide & assign \$a /= 4 Sa becomes 2.5 -eq Equal to 5 -eq 5 True Comparison -ne Not equal to 5 -ne 3 True True -gt Greater than 10 -gt 5 True True -lt Less than or equal 5 -ge 5 True True -le Less than or equal 3 -le 5 True Logical Nor True Logical Nor Logical (5 -gt 2) -and (3 -lt 4) True Logical Nor True Logical Nor True String Incutation String String String String String Incutation String String String Incutation String String Incutation String Incutation String Incutation String Incutation String Incutation String <td< td=""><td>-=</td><td>Subtract & assign</td><td>\$a -= 3</td><td>\$a becomes 7</td><td></td></td<>	-=	Subtract & assign	\$a -= 3	\$a becomes 7	
-eq Equal to 5-eq 5 True Comparison -ne Not equal to 5 -ne 3 True -gt Greater than 10 -gt 5 True -lt Less than 2 -lt 4 True -ge Greater than or equal 5 -ge 5 True -le Less than or equal 3 -le 5 True -and Logical AND (5 -gt 2) -and (3 -lt 4) True Logical -or Logical OR (5 -gt 10) -or (3 -lt 4) True Logical -not Logical NOT -not (5 -eq 5) False -not (5 -eq 5) False -like Wildcard match "PowerShell"-like "Power*" True String -notlike Wildcard mismatch "PowerShell"-anotlike "Win*" True -notmatch -match Regex match "abc123"-match "\d+" True -notmatch -notmatch Regex mismatch "abc123"-notmatch "\d+" True -notmatch -replace Replace text "Hello World"-replace "World", "PowerShell" "Hello	*=	Multiply & assign	\$a *= 2	\$a becomes 20	
-ne Not equal to 5 -ne 3 True -gt Greater than 10 -gt 5 True -lt Less than 2 -lt 4 True -ge Greater than or equal 5 -ge 5 True -le Less than or equal 3 -le 5 True -and Logical AND (5 -gt 2) -and (3 -lt 4) True Logical -or Logical OR (5 -gt 10) -or (3 -lt 4) True True -not Logical NOT -not (5 -eq 5) False -like Wildcard match "PowerShell" -like "Power*" True String -notlike Wildcard mismatch "PowerShell" -notlike "Win*" True True -match Regex match "abc123" -natch "\d+" True True -notmatch Regex mismatch "abc* -notmatch "\d+" True Array & Collection -replace Replace text "Hello World" -replace "World", "PowerShell" "Hello PowerShell" -contains Array does not contain value @(1,2,3) -nottains 2 True Array & Collect	/=	Divide & assign	\$a /= 4	\$a becomes 2.5	
-gt Greater than 10 -gt 5 True -lt Less than 2 -lt 4 True -ge Greater than or equal 5 -ge 5 True -le Less than or equal 3 -le 5 True -and Logical AND (5 -gt 2) -and (3 -lt 4) True Logical -or Logical OR (5 -gt 10) -or (3 -lt 4) True -not Logical NOT -not (5 -eq 5) False -like Wildcard match "PowerShell" -like "Power*" True String -notlike Wildcard mismatch "PowerShell" -notlike "Win*" True -match Regex match "abc123" -match "(4+" True -notmatch Regex mismatch "abc123" -match "(4+" True -replace Replace text "Hello World" -replace "World", "PowerShell" "Hello PowerShell" -contains Array contains value @(1,2,3) -contains 2 True Array & Collection -nottcontains Array does not contain value @(1,2,3) -notcontains 5 True -in Value in array 5 -in @(1,2,3,5) True -notin Value not in array 6 -notin @(1,2,3,5) True -notin Value not in array 6 -notin @(1,2,3,5) True -notin Value not in array 6 -notin @(1,2,3,5) True -Redirect output (overwrite) Get-Process > processes.txt Appends to file Redirection to the contain value Appends to file Redirect output (append) Get-Process > processes.txt Appends to file	-eq	Equal to	5 -eq 5	True	Comparison
-It Less than 2-It 4 True -ge Greater than or equal 5 - ge 5 True -le Less than or equal 3 - le 5 True -and Logical AND (5 - gt 2) - and (3 - lt 4) True Logical -or Logical OR (5 - gt 10) - or (3 - lt 4) True -not Logical NOT -not (5 - eq 5) False -like Wildcard match "PowerShell" -like "Power*" True String -notlike Wildcard mismatch "PowerShell" -notlike "Win*" True -match Regex match "abc123" -match "\d+" True -notmatch Regex mismatch "abc123" -match "\d+" True -replace Replace text "Hello World" -replace "World", "PowerShell" "Hello PowerShell" -contains Array contains value @(1,2,3) -contains 2 True Array & Collection -notcontains Array does not contain value @(1,2,3) -notcontains 5 True -in Value in array 5 - in @(1,2,3,5) True -notin Value not in array 6 -notin @(1,2,3,5) True -notin Value not in array 6 -notin @(1,2,3,5) True -Redirect output (overwrite) Get-Process > processes.txt Appends to file Redirection -Redirect output (append) Get-Process > processes.txt Appends to file	-ne	Not equal to	5 -ne 3	True	
-geGreater than or equal5 -ge 5True-leLess than or equal3 -le 5True-andLogical AND(5 -gt 2) -and (3 -lt 4)TrueLogical-orLogical OR(5 -gt 10) -or (3 -lt 4)True	-gt	Greater than	10 -gt 5	True	
-le Less than or equal 3 -le 5 True -and Logical AND (5 -gt 2) -and (3 -lt 4) True Logical -or Logical OR (5 -gt 10) -or (3 -lt 4) True -not Logical NOT -not (5 -eq 5) False -like Wildcard match "PowerShell" -like "Power*" True String -notlike Wildcard mismatch "PowerShell" -notlike "Win*" True -match Regex match "abc123" -match "\d+" True -notmatch Regex mismatch "abc" -notmatch "\d+" True -replace Replace text "Hello World" -replace "World", "PowerShell" "Hello PowerShell" -contains Array contains value @(1,2,3) -contains 2 True Array & Collection -notcontains Array does not contain value @(1,2,3) -notcontains 5 True -in Value in array 5 -in @(1,2,3,5) True -notin Value not in array 6 -notin @(1,2,3,5) True > Redirect output (overwrite) Get-Process > processes.txt Writes to file Redirection >> Redirect output (append) Get-Process >> processes.txt Appends to file 2> Redirect errors Get-Command xyz 2> errors.txt Saves errors to file	-lt	Less than	2 -lt 4	True	
-and Logical AND (5 -gt 2) -and (3 -lt 4) True Logical -or Logical OR (5 -gt 10) -or (3 -lt 4) True -not Logical NOT -not (5 -eq 5) False -like Wildcard match "PowerShell" -like "Power*" True String -notlike Wildcard mismatch "PowerShell" -notlike "Win*" True -match Regex match "abc123" -match "\d+" True -notmatch Regex mismatch "abc" -notmatch "\d+" True -replace Replace text "Hello World" -replace "World", "PowerShell" "Hello PowerShell" -contains Array contains value @(1,2,3) -contains 2 True Array & Collection -notcontains Array does not contain value @(1,2,3) -notcontains 5 True -in Value in array 5 -in @(1,2,3,5) True -notin Value not in array 6 -notin @(1,2,3,5) True > Redirect output (overwrite) Get-Process > processes.txt Writes to file Redirection >> Redirect output (append) Get-Process >> processes.txt Appends to file 2> Redirect errors Get-Command xyz 2> errors.txt Saves errors to file	-ge	Greater than or equal	5 -ge 5	True	
-or Logical OR (5 -gt 10) -or (3 -lt 4) True -not Logical NOT -not (5 -eq 5) False -like Wildcard match "PowerShell" -like "Power*" True String -notlike Wildcard mismatch "PowerShell" -notlike "Win*" True -match Regex match "abc123" -match "\d+" True -notmatch Regex mismatch "abc" -notmatch "\d+" True -replace Replace text "Hello World" -replace "World", "PowerShell" "Hello PowerShell" -contains Array contains value @(1,2,3) -contains 2 True Array & Collection -notcontains Array does not contain value @(1,2,3) -notcontains 5 True -in Value in array 5 -in @(1,2,3,5) True -notin Value not in array 6 -notin @(1,2,3,5) True > Redirect output (overwrite) Get-Process > processes.txt Writes to file Redirection >> Redirect output (append) Get-Process >> processes.txt Appends to file 2> Redirect errors Get-Command xyz 2> errors.txt Saves errors to file	-le	Less than or equal	3 -le 5	True	
-not Logical NOT -not (5 -eq 5) False -like Wildcard match "PowerShell" -like "Power*" True String -notlike Wildcard mismatch "PowerShell" -notlike "Win*" True -match Regex match "abc123" -match "\d+" True -notmatch Regex mismatch "abc" -notmatch "\d+" True -replace Replace text "Hello World" -replace "World", "PowerShell" "Hello PowerShell" -contains Array contains value @(1,2,3) -contains 2 True Array & Collection -notcontains Array does not contain value @(1,2,3) -notcontains 5 True -in Value in array 5 -in @(1,2,3,5) True -notin Value not in array 6 -notin @(1,2,3,5) True > Redirect output (overwrite) Get-Process > processes.txt Writes to file Redirection >> Redirect output (append) Get-Process >> processes.txt Appends to file 2> Redirect errors Get-Command xyz 2> errors.txt Saves errors to file	-and	Logical AND	(5 -gt 2) -and (3 -lt 4)	True	Logical
-like Wildcard match "PowerShell" -like "Power*" True String -notlike Wildcard mismatch "PowerShell" -notlike "Win*" True -match Regex match "abc123" -match "\d+" True -notmatch Regex mismatch "abc" -notmatch "\d+" True -replace Replace text "Hello World" -replace "World", "PowerShell" "Hello PowerShell" -contains Array contains value @(1,2,3) -contains 2 True Array & Collection -notcontains Array does not contain value @(1,2,3) -notcontains 5 True -in Value in array 5 -in @(1,2,3,5) True -notin Value not in array 6 -notin @(1,2,3,5) True > Redirect output (overwrite) Get-Process > processes.txt Writes to file Redirection >> Redirect output (append) Get-Process >> processes.txt Appends to file 2> Redirect errors Get-Command xyz 2> errors.txt Saves errors to file	-or	Logical OR	(5 -gt 10) -or (3 -lt 4)	True	
-notlike Wildcard mismatch "PowerShell" -notlike "Win*" True -match Regex match "abc123" -match "\d+" True -notmatch Regex mismatch "abc" -notmatch "\d+" True -replace Replace text "Hello World" -replace "World", "PowerShell" "Hello PowerShell" -contains Array contains value @(1,2,3) -contains 2 True Array & Collection -notcontains Array does not contain value @(1,2,3) -notcontains 5 True -in Value in array 5 -in @(1,2,3,5) True -notin Value not in array 6 -notin @(1,2,3,5) True > Redirect output (overwrite) Get-Process > processes.txt Writes to file Redirection >> Redirect output (append) Get-Process >> processes.txt Appends to file 2> Redirect errors Get-Command xyz 2> errors.txt Saves errors to file	-not	Logical NOT	-not (5 -eq 5)	False	
-match Regex match "abc123" -match "\d+" True -notmatch Regex mismatch "abc" -notmatch "\d+" True -replace Replace text "Hello World" -replace "World", "PowerShell" "Hello PowerShell" -contains Array contains value @(1,2,3) -contains 2 True Array & Collection -notcontains Array does not contain value @(1,2,3) -notcontains 5 True -in Value in array 5 -in @(1,2,3,5) True -notin Value not in array 6 -notin @(1,2,3,5) True > Redirect output (overwrite) Get-Process > processes.txt Writes to file Redirection >> Redirect output (append) Get-Process >> processes.txt Appends to file 2> Redirect errors Get-Command xyz 2> errors.txt Saves errors to file	-like	Wildcard match	"PowerShell" -like "Power*"	True	String
-notmatch Regex mismatch "abc" -notmatch "\d+" True -replace Replace text "Hello World" -replace "World", "PowerShell" "Hello PowerShell" -contains Array contains value @(1,2,3) -contains 2 True Array & Collection -notcontains Array does not contain value @(1,2,3) -notcontains 5 True -in Value in array 5 -in @(1,2,3,5) True -notin Value not in array 6 -notin @(1,2,3,5) True > Redirect output (overwrite) Get-Process > processes.txt Writes to file Redirection >> Redirect output (append) Get-Process >> processes.txt Appends to file 2> Redirect errors Get-Command xyz 2> errors.txt Saves errors to file	-notlike	Wildcard mismatch	"PowerShell" -notlike "Win*"	True	
-replace Replace text "Hello World" -replace "World", "PowerShell" "Hello PowerShell" -contains Array contains value @(1,2,3) -contains 2 True Array & Collection -notcontains Array does not contain value @(1,2,3) -notcontains 5 True -in Value in array 5 -in @(1,2,3,5) True -notin Value not in array 6 -notin @(1,2,3,5) True > Redirect output (overwrite) Get-Process > processes.txt Writes to file Redirection >> Redirect output (append) Get-Process >> processes.txt Appends to file 2> Redirect errors Get-Command xyz 2> errors.txt Saves errors to file	-match	Regex match	"abc123" -match "\d+"	True	
-contains Array contains value @(1,2,3) -contains 2 True Array & Collection -notcontains Array does not contain value @(1,2,3) -notcontains 5 True -in Value in array 5 -in @(1,2,3,5) True -notin Value not in array 6 -notin @(1,2,3,5) True > Redirect output (overwrite) Get-Process > processes.txt Writes to file Redirection >> Redirect output (append) Get-Process >> processes.txt Appends to file 2> Redirect errors Get-Command xyz 2> errors.txt Saves errors to file	-notmatch	Regex mismatch	"abc" -notmatch "\d+"	True	
-notcontains Array does not contain value @(1,2,3) -notcontains 5 True -in Value in array 5 -in @(1,2,3,5) True -notin Value not in array 6 -notin @(1,2,3,5) True > Redirect output (overwrite) Get-Process > processes.txt Writes to file Redirection >> Redirect output (append) Get-Process >> processes.txt Appends to file 2> Redirect errors Get-Command xyz 2> errors.txt Saves errors to file	-replace	Replace text	"Hello World" -replace "World", "PowerShell"	"Hello PowerShell"	
-in Value in array 5 -in @(1,2,3,5) True -notin Value not in array 6 -notin @(1,2,3,5) True > Redirect output (overwrite) Get-Process > processes.txt Writes to file Redirection >> Redirect output (append) Get-Process >> processes.txt Appends to file 2> Redirect errors Get-Command xyz 2> errors.txt Saves errors to file	-contains	Array contains value	@(1,2,3) -contains 2	True	Array & Collection
-notin Value not in array 6 -notin @(1,2,3,5) True > Redirect output (overwrite) Get-Process > processes.txt Writes to file Redirection >> Redirect output (append) Get-Process >> processes.txt Appends to file 2> Redirect errors Get-Command xyz 2> errors.txt Saves errors to file	-notcontains	Array does not contain value	@(1,2,3) -notcontains 5	True	
> Redirect output (overwrite) Get-Process > processes.txt Writes to file Redirection >> Redirect output (append) Get-Process >> processes.txt Appends to file 2> Redirect errors Get-Command xyz 2> errors.txt Saves errors to file	-in	Value in array	5 -in @(1,2,3,5)	True	
>> Redirect output (append) Get-Process >> processes.txt Appends to file 2> Redirect errors Get-Command xyz 2> errors.txt Saves errors to file	-notin	Value not in array	6 -notin @(1,2,3,5)	True	
2> Redirect errors Get-Command xyz 2> errors.txt Saves errors to file	>	Redirect output (overwrite)	Get-Process > processes.txt	Writes to file	Redirection
·	>>	Redirect output (append)	Get-Process >> processes.txt	Appends to file	
2>> Append errors Get-Command xyz 2>> errors.txt Appends errors to file	2>	Redirect errors	Get-Command xyz 2> errors.txt	Saves errors to file	
	2>>	Append errors	Get-Command xyz 2>> errors.txt	Appends errors to file	
-is Check object type 5 -is [int] True Type	-is	Check object type	5 -is [int]	True	Туре
-as Convert object type "123" -as [int] 123	-as	Convert object type	"123" -as [int]	123	
Range of numbers 15 1 2 3 4 5 Range		Range of numbers	15	12345	Range
& Execute script/command & "C:\script.ps1" Runs script Call	&	Execute script/command	& "C:\script.ps1"	Runs script	Call
\$() Evaluate expression \$(Get-Date) Outputs current date Subexpression	\$()	Evaluate expression	\$(Get-Date)	Outputs current date	Subexpression