PowerShell Commands Cheat Sheet

PowerShell - Files and Folders

Task	Command	Example
Creating Folders	New-Item -ItemType Directory <FolderName>	New-Item -ItemType Directory "C:\NewFolder"
Creating Files	New-Item -ItemType File <FileName>	New-Item -ItemType File "C:\test.txt"
Copying Folders	Copy-Item -Path <Source> -Destination <Dest> -Recurse	Copy-Item -Path "C:\Folder1" -Destination "D:\Backup" -Recurse
Copying Files	Copy-Item <SourceFile> <Destination>	Copy-Item "C:\file.txt" "D:\backup.txt"
Deleting Folders	Remove-Item -Path <FolderName> -Recurse -Force	Remove-Item -Path "C:\OldFolder" -Recurse -Force
Deleting Files	Remove-Item <FileName>	Remove-Item "C:\test.txt"
Moving Folders	Move-Item <SourceFolder> <Destination>	Move-Item "C:\Folder1" "D:\NewLocation"
Moving Files	Move-Item <SourceFile> <Destination>	Move-Item "C:\test.txt" "D:\backup.txt"
Rename Folders	Rename-Item <OldName> <NewName>	Rename-Item "C:\OldFolder" "C:\NewFolder"
Rename Files	Rename-Item <OldName> <NewName>	Rename-Item "C:\old.txt" "C:\new.txt"
Retrieving Item	Get-Item <Path>	Get-Item "C:\file.txt"
Check Folder Existence	Test-Path <FolderName>	Test-Path "C:\Users" (Returns $true or $false)
Check File Existence	Test-Path <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>"	Set-Date -Date "02/18/2025"
Get System Time	(Get-Date).TimeOfDay	(Get-Date).TimeOfDay
Set System Time	Set-Date -Date "<MM/DD/YYYY HH:MM>"	Set-Date -Date "02/18/2025 14:30"

PowerShell - Files I/O

Task	Command	Example
Create Text File	New-Item -ItemType File <FileName>	New-Item -ItemType File "C:\test.txt"
Read Text File	Get-Content <FileName>	Get-Content "C:\test.txt"
Create XML File	[xml]$xml = "<root></root>"; $xml.Save("<FileName>")	[xml]$xml = "<root></root>"; $xml.Save("C:\data.xml")
Read XML File	[xml]$xml = Get-Content <FileName>	[xml]$xml = Get-Content "C:\data.xml"
Create CSV File	New-Item -ItemType File <FileName>.csv	New-Item -ItemType File "C:\data.csv"
Read CSV File	Import-Csv <FileName>	Import-Csv "C:\data.csv"
Create HTML File	New-Item -ItemType File <FileName>.html	New-Item -ItemType File "C:\index.html"
Read HTML File	Get-Content <FileName>.html	Get-Content "C:\index.html"
Erasing File Content	Set-Content <FileName> ""	Set-Content "C:\test.txt" ""
Append Text Data	Add-Content <FileName> "Text"	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>	Get-Help Get-Service
Show examples	Get-Help <Cmdlet> -Examples	Get-Help Get-Process -Examples
Show detailed help	Get-Help <Cmdlet> -Detailed	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>	Get-ChildItem C:\Users
Create a file	New-Item -ItemType File <FileName>	New-Item -ItemType File "C:\test.txt"
Create a folder	New-Item -ItemType Directory <FolderName>	New-Item -ItemType Directory "C:\NewFolder"
Copy a file	Copy-Item <Source> <Destination>	Copy-Item "C:\file.txt" "D:\backup.txt"
Move a file	Move-Item <Source> <Destination>	Move-Item "C:\file.txt" "D:\backup.txt"
Delete a file	Remove-Item <FileName>	Remove-Item "C:\test.txt"
Delete a folder	Remove-Item <FolderName> -Recurse -Force	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>	New-LocalUser -Name "TestUser" -Password (ConvertTo-SecureString "P@ssword123" -AsPlainText -Force)
Delete a user	Remove-LocalUser -Name <UserName>	Remove-LocalUser -Name "TestUser"
List local groups	Get-LocalGroup	Get-LocalGroup
Add user to a group	Add-LocalGroupMember -Group <Group> -Member <User>	Add-LocalGroupMember -Group "Administrators" -Member "TestUser"

Networking

Task	Command	Example
Get IP address	Get-NetIPAddress	Get-NetIPAddress
Ping a host	Test-Connection <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>	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	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>	Start-Service -Name "wuauserv"
Stop a service	Stop-Service -Name <ServiceName>	Stop-Service -Name "wuauserv"
Restart a service	Restart-Service -Name <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:$false	Clear-Disk -Number 1 -Confirm:$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
Run a command remotely	Invoke-Command -ComputerName <Server> -ScriptBlock { Get-Service }	

Operators

Operator	Description	Example	Output	Category
+	Addition	5 + 3	8	**Arithmetic**
-	Subtraction	10 - 4	6	
*	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 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 -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	
-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	
2>>	Append errors	Get-Command xyz 2>> errors.txt	Appends errors to file	
-is	Check object type	5 -is [int]	True	**Type**
-as	Convert object type	"123" -as [int]	123	
..	Range of numbers	1..5	1 2 3 4 5	**Range**
&	Execute script/command	& "C:\script.ps1"	Runs script	**Call**
$( )	Evaluate expression	$(Get-Date)	Outputs current date	**Subexpression**