



Basic PowerShell Commands

Ref:(<https://docs.microsoft.com/en-us/powershell/scripting/overview?view=powershell-6>)


PowerShell uses "Verb-Noun" naming system

1. Get-host | select-object Version # to display PowerShell version
2. \$PSVersionTable.PSVersion # display PowerShell version
3. Get-Command -Verb Get # to list all cmdlets that uses the verb Get
4. Get-Command -Noun service # to list all commands available to manage services.
5. Get-Command *-Service # find a list of cmdlets with service in their name
6. Get-Command -Name Clear-Host
7. Get-Help or help or man # to get help
8. Get-Help Get-Service -example # Get help for cmdlet Get-Service
9. Get-Verb # show all the verbs
10. Clear; dir *.txt # multiple commands separated by semicolon
11. write-host "Hello World" # Famous hello world example
12. write-host -foregroundcolor red "Hello World"
13. Linux commands that can be used as such in PowerShell "cat, cd, chdir, clear, cls, copy, del, diff, dir, echo, erase, h, history, kill, lp, ls, mount, move, popd, ps, pushd, r, ren, rm rmdir, sleep, sort, tee, type, write,
14. Get-Item env: # show the environment variables
15. Get-Location # Equivalent to command 'pwd' in Unix
16. Get-ChildItem # Equivalent to "ls" in Unix
17. Get-Alias ls # Find the alias for ls command
18. Get-Alias | out-host -paging # Equivalent to more
19. Get-Alias | more # more
20. Get-ChildItem \$env:USERPROFILE\Desktop | Sort-Object -Property LastWriteTime (Equivalent to ls -ltr in Unix)
21. Get-ChildItem -Filter "commands" -Recurse -File (Equivalent to find command in Unix)



24. `New-Item -ItemType Directory -Name 'TestFolder' # Equivalent to "mkdir" in Unix`
25. `New-Item -ItemType File -Name 'TestFile' # Equivalent to "touch" in Unix`
26. `Get-Content <filename> # Equivalent to "cat" in Unix`
27. `Get-Content -Tail 2 <filename> # Display last two lines`
28. `Get-Content .\Alias.txt | select -first 10 # Equivalent to head -10 ; <Alias.txt> is a filename, replace that with your file name`
29. `Get-Content .\Alias.txt | select -last 10 # Equivalent to last -10`
30. `Get-Content -totalcount 10 .\Alias.txt # Equivalent to head -10`
31. `Get-Content .\Alias.txt | %{ $_ -replace '\d+', '($0)' } # Equivalent to sed; "%" is an alias for ForEach object "Get-Alias -Definition ForEach-Object"; It will also be a modulus operator like in $10 \% 3$.`
32. `Get-Content .\Alias.txt | %{ $n=0; }{$n++; "$n $_"} # Equivalent to cat -n`
33. `Get-Content .\Alias.txt | ForEach{ "[0,5]{1}" -f $_.ReadCount, $_ } # print with line numbers`
34. `Get-Content .\Alias.txt | %{ $n=0; }{$n++; if($n -gt 2 -and $n -lt 7){ "$n $_" }} # Equivalent to sed to print certain lines`
35. `Measure-Command { Get-EventLog "windows powershell" } # Equivalent to time command in Unix`
36. `Get-ChildItem | Measure-Object # count the number of files`
37. `Get-Process | where-object cpu -gt 100 # Display process using more than 100% CPU. Instead of get-process you can type "ps"`
38. `(Get-Command python).Path # Equivalent to "which python" in Unix`
39. `Get-CimInstance win32_operatingsystem # information about your operating system`
40. `Get-CimInstance -Class Win32_Process | more # process information`
41. `Test-Connection <ipaddress> # equivalent to ping`
42. `Select-String -Path .\commands "process" # Equivalent to "grep" in Unix; commands is a filename, replace it with your file name and string.`
43. `Get-Help Get-Service -ShowWindow # Pop up the help in a separate window`
44. `Get-Process | Where-Object { $_.WorkingSet -gt 200000000 } # Display process greater than certain size.`
45. `Get-Process powershell # Display details of the process powershell`
46. `Get-Process | Where { $_.Handles -gt 750 } | Sort PM -Descending # sort the output`
47. `Get-Process -IncludeUserName # Run it as Administrator to get the process username information`
48. `$process = (Get-Process)[0] # subExpressions`
49. `"Net*" | Get-Service # Show all services with name Net`



52. Get-ComputerInfo | more # info about your computer.
53. Get-Service | out-file -FilePath C:\Users\\service.txt # Store the output to a file
54. Get-Service | export-csv -Path C:\Users\\service.txt # output to CSV format.
55. Get-EventLog -LogName System -Newest 10 -EntryType error # show event logs
56. Get-Service | Select-Object -Property name, status | Sort-Object -property status -Descending # Show running and stopped services sorted.
57. Get-ChildItem -path C:\Users\\ | Sort-Object -Property length -Descending # Show the files sorted according to size
58. Get-ExecutionPolicy # Show execution policy (Restricted, RemoteSigned etc.)
59. Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser # Set execution policy
60. Get-CimInstance -ClassName Win32_LogicalDisk -filter "DeviceID='c:'" | select @{n="FreeGB"; e={\$_.FreeSpace / 1gb -as [int]}} # Show the free space in GB
61. Get-CimInstance -ClassName Win32_LogicalDisk -filter "DeviceID='c:'" | Select-Object -Property @{n="FreeGB"; e={\$_.FreeSpace / 1gb -as [int]}} # show the free space in GB and change the column heading as well.
62. start firefox <https://www.google.com> # start firefox from powershell
63. ise # start powershell ise (Other commands are notepad, calc etc)
64. Get-History # Equivalent to history command in Unix
65. Get-History | Foreach-Object { \$_.CommandLine } > .\history_script.ps1 # save the history to a file.
66. New-Alias -Name grep -Description grep Select-String # Alias for grep in Unix
67. Get-ChildItem env: | out-host -paging # Equivalent to less command in Unix
68. dir -r | Select-String "searchforthis" # Recursively search for a string.
69. ps | sort -p ws | select -last 5 # Find the five process using most memory
70. ps | sort -p cpu | select -last 5 # Find the five process using most CPU
71. Get-WmiObject -Class Win32_LogicalDisk # show information about the hard drive free space and total space.
72. Get-WmiObject -Class Win32_ComputerSystem # Get information about the model of this computer
73. Get-WmiObject -Class Win32_BIOS -ComputerName . # Get BIOS information
74. Get-WmiObject -Class Win32_ComputerSystem -Property UserName -ComputerName . # Get the information about the user logged into the machine.
75.  Get-WmiObject -Class Win32_Product -ComputerName . | Format-Wide -Column 1 # List all the applications that are installed on this computer

- 77. `Get-WmiObject -class Win32_OperatingSystem` # operating system information
- 78. `Get-Netadapter` # Get detailed information about the network adapters, the mac address, link speed, interface description etc.
- 79. `Get-Netadapter | format-table -Autosize`
- 80. `Get-NetIPAddress | Where-Object {$_.AddressFamily -eq 'IPv4'} | ForEach-Object IPAddress` # Get all the IPV4 addresses.
- 81. `Get-Eventlog application -newest 5 | format-table -Wrap` # to Wrap the output
- 82. `New-Alias grep findstr` # use grep (like in unix) for findstr
- 83. `ls | grep txt` # display all files with "txt" string in it.
- 84. `'Hello', 'HELLO' | Select-String -Pattern 'HELLO' -CaseSensitive -SimpleMatch` # select-string can also provide grep like functionality
- 85. `Set-Location c:\` # Equivalent to "cd" in Unix
- 86. `Start-Process ((Resolve-Path "C:\Users\\file.pdf").Path)` # open a pdf file
- 87. `Find-Package <pkgname>` # find packages from a repository Eg: `Find-Package ssh | Install-Package as administrator`
- 88. `Get-PSDrive` # equivalent to df command
- 89. `Measure-Command { Get-EventLog "windows powershell" }` # Equivalent to time command

