



PowerShell Commands

Agenda


- ❖ Introduction
- ❖ What Makes PowerShell Unique?
- ❖ Why Learn PowerShell?
- ❖ 30 Fundamental PowerShell commands



Introduction



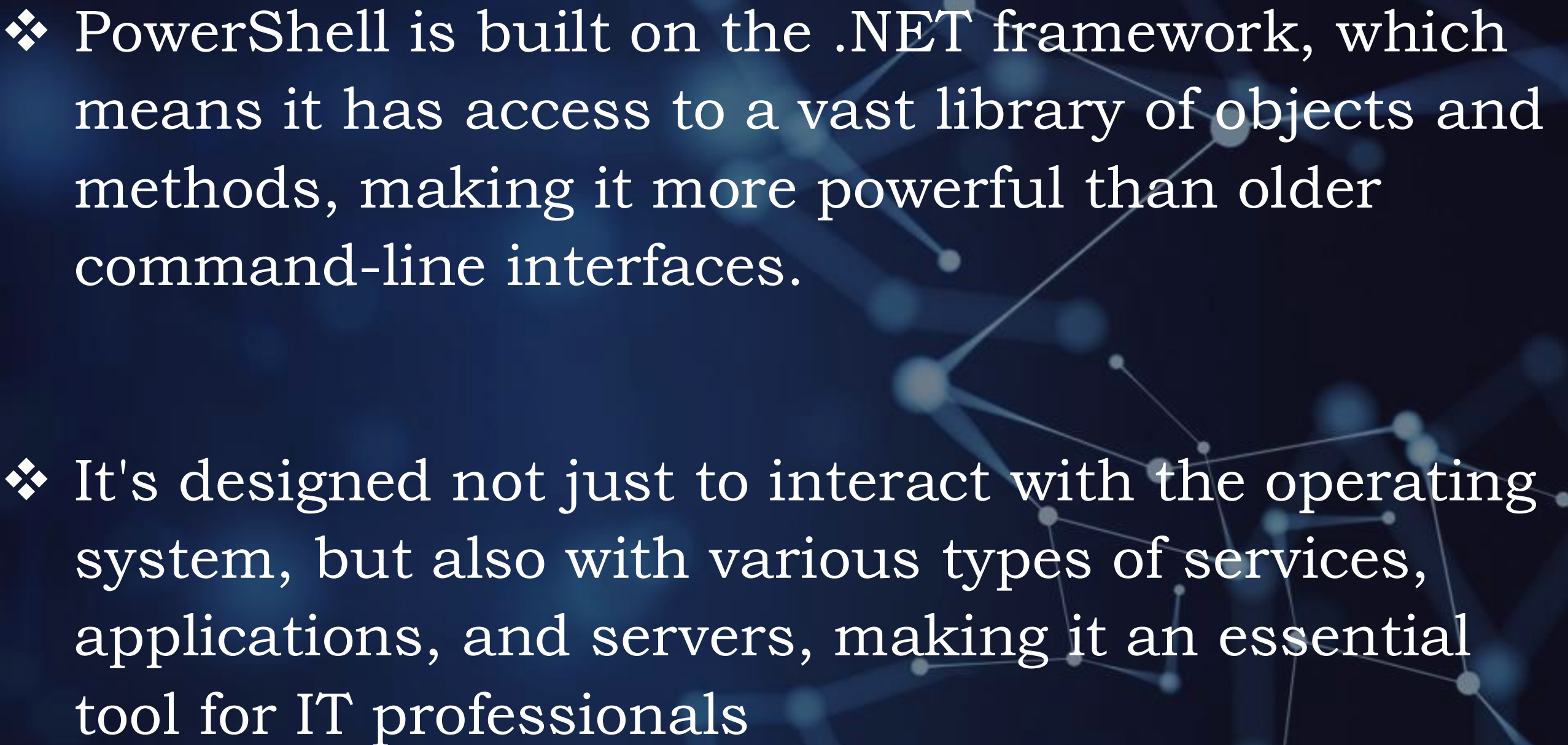
PowerShell is a powerful command-line shell and scripting language designed by Microsoft, primarily for system administration and automation.



Unlike the traditional Command Prompt, PowerShell provides a more advanced environment that allows users to manage computers and automate tasks using a wide array of commands known as cmdlets.

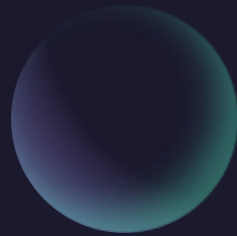


What Makes PowerShell Unique?

- 
- The background of the slide features a dark blue gradient with a subtle, abstract pattern of light blue dots and thin white lines, resembling a network or molecular structure.
- ❖ PowerShell is built on the .NET framework, which means it has access to a vast library of objects and methods, making it more powerful than older command-line interfaces.
 - ❖ It's designed not just to interact with the operating system, but also with various types of services, applications, and servers, making it an essential tool for IT professionals

Why Learn PowerShell?

- ❖ Automates repetitive tasks.
- ❖ Simplifies system administration.
- ❖ Powerful tool for both beginners and IT professionals





30 Fundamental PowerShell commands

1. ***Get-Help*** :

This command provides a full description, including parameters and examples, helping you understand how to use Set-Service effectively

You're new to PowerShell and need to understand how a specific cmdlet, like Set-Service, works.

2. *Get-Command* :

This command lists all commands related to file retrieval, helping you find the appropriate cmdlet, functions, workflows, aliases and scripts for your task.

You're developing a script that requires file manipulation, but you're unsure which cmdlets handle files.

```
PS C:\Users\Admin>  
PS C:\Users\Admin> get-command
```

CommandType	Name	Version	Source
Alias	Add-AppPackage	2.0.1.0	Appx
Alias	Add-AppPackageVolume	2.0.1.0	Appx
Alias	Add-AppProvisionedPackage	3.0	Dism

3. *Get-Process* :

This command helps you identify the processes consuming the most CPU, allowing you to take appropriate action.

A server is running slowly, and you suspect a process is consuming too many resources

```
PS C:\Users\Admin> get-process
```

Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
354	21	10008	29096	0.17	4472	1	ApplicationFrameHost
566	14	12928	19704	0.09	4992	0	audiodg
1644	55	104992	201184	602.75	1744	1	chrome
588	46	453504	494996	1,290.03	2124	1	chrome

4. ***Stop-Process*** :

This command forcefully stops the various processes, freeing up system resources.

A rogue process is consuming excessive CPU resources, and you need to terminate it.

```
PS C:\Users\Admin> stop-process -Name "Dropbox"  
PS C:\Users\Admin> stop-process -id 6544
```


Practical example

```
PS C:\Users\Admin> get-process | sort-object Workingset -Descending | select-object -First 5
```

Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
582	45	463704	509404	1,437.95	2124	1	chrome
5576	171	333368	417872	140.25	6544	1	Dropbox
405	47	410408	408392	650.08	5320	1	chrome
520	40	268784	273348	341.52	7696	1	chrome
1640	55	104836	201012	630.91	1744	1	chrome

```
PS C:\Users\Admin> get-process | sort-object Workingset -Descending | select-object -First 5 | stop-process -Force
```

5 Get-Service :

The Get-Service command in PowerShell is used to retrieve the status of services on a local or remote computer.

You can use it to list all services, filter by service name, or check the status of specific services.

```
PS C:\Users\Admin> get-service
```

Status	Name	DisplayName
Stopped	AarSvc_330ec	Agent Activation Runtime_330ec

```
PS C:\Users\Admin> get-service -name "Power"
```

Status	Name	DisplayName
Running	Power	Power

6 Start-Service :

This command in `systemd` is used to start one or more services on a local or remote computer.

This is useful when you need to ensure that critical services are running or when you're automating service management tasks

7. *Stop-Service:*

This command is used stop one or more services in the a local or remote computer.

This is necessary for maintenance, troubleshooting or when a service needs to be temporarily disabled.

```
PS C:\Users\Admin> stop-process -Name "Dropbox"  
PS C:\Users\Admin> stop-process -id 6544
```


8. *Restart-Service:*

This command is used to stop and then start one or more services in the a local or remote computer.

This is necessary for , refreshing the state of a service without having to manually stop and start it.

```
PS C:\Users\Admin> restart-service -Name "bthserv"  
WARNING: Waiting for service 'Bluetooth Support Service (bthserv)' to stop...  
WARNING: Waiting for service 'Bluetooth Support Service (bthserv)' to stop...  
WARNING: Waiting for service 'Bluetooth Support Service (bthserv)' to stop...
```

9. *Get-EventLog* :

This command is used to retrieve event log entries for a computer.

Event logs are records of system, security and application-related events that can help in diagnosing issues, monitoring security, or auditing activities on the system

```
PS C:\Users\Admin> get-eventlog  
cmdlet Get-EventLog at command pipeline position 1  
Supply values for the following parameters:  
LogName: ■
```

10. *Clear-EventLog* :

This clears all entries in the Event log, making it easier to spot new issues

It can be useful when you need to clear logs for troubleshooting, manage disk space, or reset the events log for monitoring.

```
PS C:\Users\Admin> clear-eventlog
```

```
cmdlet Clear-EventLog at command pipeline position 1  
Supply values for the following parameters:  
LogName[0]:
```

11. Get-Location :

This is used to display the present working directory("pwd" which can be used interchangeably with get-location) in which the PowerShell session is operating.

This helps to Know exactly where you are in the file system or when working with relative paths.

```
PS C:\Users\Admin> get-location
```

```
Path
```

```
C:\Users\Admin
```

```
PS C:\Users\Admin> pwd
```

```
Path
```

```
C:\Users\Admin
```

```
PS C:\Users\Admin> clear-ev  
cmdlet Clear-Eventlog at co  
Supply values for the follo  
LogName[0]:
```


12. Set-Location :

This command is used to change the current directory to a different location within the file system, registry, or other supported providers.

It's the equivalent of the `cd`(change directory) command in other shells like command prompt or linux. And also used to move to the Users directory.

```
PS C:\Users\Admin> set-location -path "Desktop"
PS C:\Users\Admin\Desktop> pwd
```

Path

C:\Users\Admin\Desktop

```
PS C:\Users\Admin> cd -path "Desktop"
PS C:\Users\Admin\Desktop> pwd
```

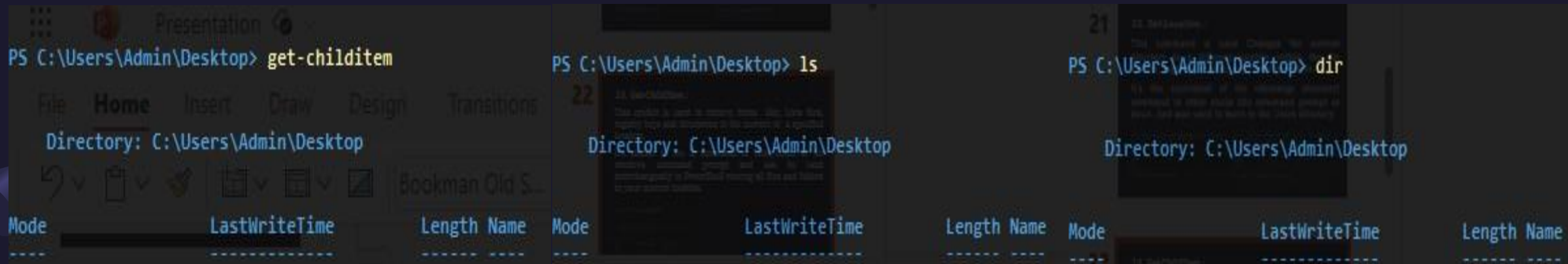
Path

C:\Users\Admin\Desktop

13. Get-ChildItem :

This cmdlet is used to retrieve items like; Lists files, registry keys and directories in the current or a specified location.

It's similar to the ls command in Unix/Linux or dir windows command prompt and can be used interchangeably in PowerShell viewing all files and folders in your current location.



The image displays three side-by-side screenshots of a PowerShell window, each showing the output of a different command used to list files and directories in the current location (C:\Users\Admin\Desktop).

Left Screenshot (get-childitem): The command `PS C:\Users\Admin\Desktop> get-childitem` is entered. The output shows a table with columns: Mode, LastWriteTime, Length, and Name. The table lists the contents of the directory.

Middle Screenshot (ls): The command `PS C:\Users\Admin\Desktop> ls` is entered. The output shows a table with columns: Mode, LastWriteTime, Length, and Name. The table lists the contents of the directory.

Right Screenshot (dir): The command `PS C:\Users\Admin\Desktop> dir` is entered. The output shows a table with columns: Mode, LastWriteTime, Length, and Name. The table lists the contents of the directory.

14. *New-Item* :

- ❖ This cmdlet is used to create new items like; Lists files, registry keys and directories.
- ❖ It's is a versatile command that helps to automate the creation of various types of objects within a file system, registry, or other supported providers.
- ❖ Also, when this command is used to create a new item, depending on the context its can create either a file, directory or registry keys.
- ❖ New-Item -Name "Example.txt" -ItemType File

```
PS C:\Users\Admin> new-item
```

```
Home Insert Draw Design Transitions Anim
```

```
cmdlet New-Item at command pipeline position 1
```

```
Supply values for the following parameters:
```

```
Path[0]:
```



15. Copy-Item :

- ❖ This cmdlet is used copy items, files and directories from one location to another.
- ❖ It's a versatile tool that allows you to perform simple copies as well as more complex operations involving multiple files or entire directory structures.
- ❖ `Copy-Item -Path "SourcePath" -Destination "DestinationPath"`

```
PS C:\Users\Admin\Desktop> Copy-Item -path "C:\Users\Admin\Desktop\6b. Linux 2.mp4" -destination "C:\Users\Admin\Desktop\video"
```


16. *Move-Item* :

- ❖ This cmdlet is used move items, files and directories from one location to another.
- ❖ Unlike *Copy-Item* which duplicates the item, *Move-Item* transfer the items to the new location and removes them from the original location.
- ❖ `Move-Item -Path "SourcePath" -Destination "DestinationPath"`

```
PS C:\Users\Admin\Desktop> move-Item -path "C:\Users\Admin\Desktop\6b. Linux 2.mp4" -destination "C:\Users\Admin\Desktop\video"
PS C:\Users\Admin\Desktop> █
```

17. *Remove-Item* :

- ❖ This cmdlet is used delete items, registry keys, files and directories from one location to another.
- ❖ It's a powerful command that should be used with caution, as deleted items are not sent to the recycle bin –they are permanently removed.
- ❖ `Remove-Item -Path "SourcePath"`
- ❖ This cmdlet can be used with some added parameters like; -Recurse (used with the main cmdlet to delete a dir with sub-dir), -Force (for protected files)

```
PS C:\Users\Admin\Desktop> Remove-Item -path "C:\Users\Admin\Desktop\step6 .mp4"  
PS C:\Users\Admin\Desktop> █
```

18. Get-Content :

- ❖ This cmdlet is used to read the contents of files and other items such as strings from a pipeline or lines from a file.
- ❖ It's often used to display the contents of a text file in the contents of a text file in the console or process data within scripts and displays the content of file.txt line by line.
- ❖ Get-Content -Path "SourcePath"
- ❖ This cmdlet can be used with some added parameters like; Select-Object - "line to viewed "(used to view specific details in file e.g -first 2 or -last 3 which come after select-object).

```
PS C:\Users\Admin> Get-content -path "C:\Users\Admin\Downloads\demo31.pem"
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAkefM1LpAshfk7taW4Udev0okoQdDVTNCv+sV00rK6QJyoqiZ
```

```
PS C:\Users\Admin> Get-content -path "C:\Users\Admin\Downloads\demo31.pem" | select-object -Last 7
6WNYtybrdf9ytmFzuHzXkR8CgYBD8dsNilTPLiQForD9urbtdiCkVyLJK9+5Ly6T
ISwrurQgt14G0tQrvmfFE0X7zZjh8sd9ySntMfMwFAxBGC6fz+6EF0yzaQHOG4wj
0jQNd0oB/r6CogLWIjblRHeXuq6LN1G3nEU1W0rwy4/K7bsnuQhOQKAr6ezHb8wq
TfL1KQKBgHHELWmN3N5SPTprjNSwwaN1/zgEZAyiU4rVT+pGbgJiVCBRPpz6tp8y
qfRr8UoYbFD6jHpQTBBr0eSgsaKjYPoUsau3j5rbaYC0Jy2/M8Tmqqs11I52Td0HB
Yw/g1J13W1RVHLwcWha5d5MOFzVcjCsc/t7VSgaVhT1LV9H0IXT3
-----END RSA PRIVATE KEY-----
PS C:\Users\Admin>
```

19. Set-Content :

- ❖ This cmdlet is designed to write or replace the content of a file. Unlike some other content-related cmdlets, such as Add-Content, Set-Content is primarily focused on the setting the content –this means it writes the specified data to the file and, by default, replaces any existing content in that file.
- ❖ Set-Content -Path "SourcePath" -Value "Content"

```
PS C:\Users\Admin> set-content -path "C:\Users\Admin\Downloads\stivo.pem" -value " the rainfalls"  
PS C:\Users\Admin> Get-content -path "C:\Users\Admin\Downloads\stivo.pem"  
the rainfalls
```

20. *Clear-Host:*

- ❖ This cmdlet is designed to clear the current content displayed in the console window.
- ❖ It removes all the text on the screen, giving you a clean slate to work with.
- ❖ Note that, This cmdlet does not delete or modify any variables, function, or settings in your session, it only clears visible text from the console.
- ❖ It can be used interchangeably with 'cls' or just 'clear' in short.

21. Add-Content:

- ❖ This cmdlet is used to Append content to a file or other types of storage content.
- ❖ It provides a medium to add data to an existing file without overwriting the existing content.
- ❖ *Add-Content* -Path "SourcePath" -Value "Content"

```
PS C:\Users\Admin> add-content -path "C:\Users\Admin\Downloads\stivo.pem" -value "And the sunshines"
PS C:\Users\Admin> get-content -path "C:\Users\Admin\Downloads\stivo.pem"
the rainfalls
And the sunshines
```

22. *Test-Path* :

- ❖ This cmdlet is used to verify whether a path to a file, folder, or other item exists with the system.
- ❖ It returns True if the specified path exists, and False if it does not.
- ❖ `Test-Path -Path "path_to_test"`

```
PS C:\Users\Admin> test-path -path "C:\Users\Admin\Downloads\stivo.pem"  
True
```

23. Get-Date:

- ❖ This cmdlet is used to retrieve the current date and time.
- ❖ It also provides various options to format the output, perform date arithmetic, and extract specific parts of the date.
- ❖ the -Format parameter with specific format strings. For example, yyyy-MM-dd gives the date as 2024-08-19, and HH:mm:ss shows the time as 14:05:33. You can mix and match these components to create various custom date and time outputs, such as dddd, MMMM dd, yyyy for a full date format like Monday, August 19, 2024
- ❖ Date arithmetic in PowerShell allows you to add or subtract time from a date using methods like .AddDays() and .AddMonths(). For example, (Get-Date).AddDays(10) returns the date 10 days in the future.

```
PS C:\Users\Admin> get-date  
Tuesday, August 20, 2024 3:10:15 AM
```

```
PS C:\Users\Admin> Get-Date -Format "yyyy-MM-dd HH:mm:ss"  
>> @ Upgrade plan  
2024-08-20 03:13:26
```

24. Measure-Object:

- ❖ This cmdlet in PowerShell is used to calculate statistical properties of objects in a collection. It's
- ❖ commonly used to obtain metrics such as count, sum, average, minimum, and maximum values. Here's a breakdown of how it can be used:
- ❖ Common parameters for Measure-Object include *-Property* to specify the property to measure, and *-Sum*, *-Average*, *-Minimum*, *-Maximum*, and *-Count* to calculate respective statistical values.

```
Previous 7 Days
PS C:\Users\Admin> Get-ChildItem -Path "C:\Users\Admin\Desktop" -File | Measure-Object -Property Length -Minimum -Maximum -Average
>> Top PowerShell Commands

Counters: 17 Top PowerShell Cmdlets
Average : 159915162.823529
Sum      : 27185577627.999
Maximum  : 1618097742
Minimum  : 555
Property : Length

This command provides statistics on the file sizes, including the sma
size.

On 17 17 17 17 17
Message ChatGPT
```

25. *Export-Csv*:

- ❖ This command is used to export data to a CSV (Comma-Separated Values) file.
- ❖ It's useful for saving the output of your commands into a file that can be opened with spreadsheet applications like Microsoft Excel or analyzed further.
- ❖ The key parameters for *Export-Csv* are *-Path*, which specifies the output file location, and *-NoTypeInfoInformation*, which omits the type information header from the CSV file.
- ❖ *Export-Csv -Path "FilePath" -NoTypeInfoInformation*

```
PS C:\Users\Admin> Get-ChildItem -Path "C:\Users\Admin\Downloads\steveAdmin.pem" | Export-Csv -Path "C:\Users\Admin\OneDrive\Documents.csv" -NoTypeInfoInformation
```


26. *Import-Csv*:

- ❖ This command is used to read data from a CSV (Comma-Separated Values) file and convert it into PowerShell objects.
- ❖ This allows you to work with the data programmatically.
- ❖ `Import-Csv -Path "FilePath"`

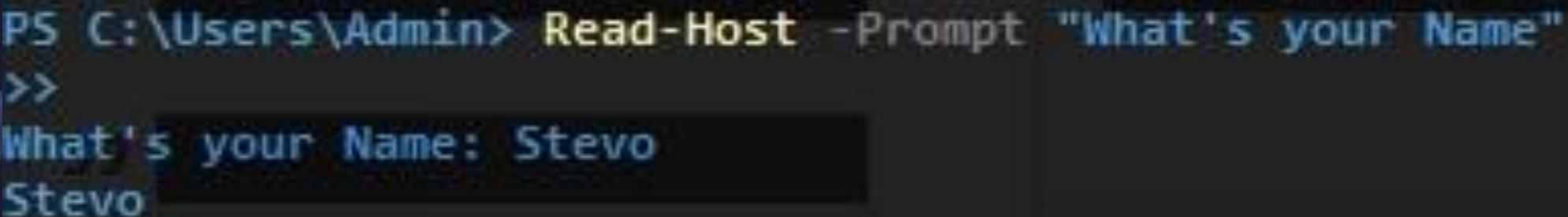
```
PS C:\Users\Admin> $data = Import-Csv -Path "C:\Users\Admin\OneDrive\Documents.csv"
>> Yesterday
```

27. *Out-File* :

- ❖ This command is used send output to a file, creating or appending to a file as specified.
- ❖ It's often used to save the results of commands or scripts to a text file.
- ❖ Out-File -FilePath "Processes.txt"

28. *Read-Host*:

- ❖ This command is used Prompts the user for input returns that input as a string.
- ❖ It's useful for interactive scripts where user input is required.
- ❖ `Read-Host -Prompt "PromptText"`



```
PS C:\Users\Admin> Read-Host -Prompt "What's your Name"  
>>  
What's your Name: Stevo  
Stevo
```

29. Start-Process :

- ❖ This cmdlet in PowerShell is used to start one or more processes on the local computer.
- ❖ It can be used to launch applications, scripts, or commands, and can include arguments, specify working directories, and control how the process is run
- ❖ The key parameters include -FilePath to specify the executable or file to start, -ArgumentList to provide arguments to the process, and -WorkingDirectory to set the directory where the process runs. The -NoNewWindow parameter runs the process in the current console, and -Wait pauses the script until the process completes

```
PS C:\Users\Admin> Start-Process -FilePath "notepad.exe" -ArgumentList "C:\Users\Admin\OneDrive\Documents\Document.docx"  
PS C:\Users\Admin> Start-Process "Desktop\6b. Linux 2.mp4" ■
```



30. Get-Alias :

- ❖ This cmdlet in PowerShell is used to retrieve a list of all aliases defined in the current session.
- ❖ Aliases are shortcut names for cmdlets, functions, and scripts, making it quicker to type common commands.

```
PS C:\Users\Admin> Get-Alias  
>>
```

CommandType	Name
Alias	% -> ForEach-Object
Alias	? -> Where-Object
Alias	acr -> Add-Content
Alias	asnp -> Add-PSSnapin
Alias	cat -> Get-Content
Alias	cd -> Set-Location

Examples

Version	Source
1	Get All Aliases

```
powerShell
```

```
Get-Alias
```


Conclusion

- ❖ In conclusion, mastering these 30 fundamental PowerShell commands equips you with a powerful toolkit for managing and automating tasks across your systems.
- ❖ These commands are not just building blocks; they provide the foundation for more advanced scripting and process automation.
- ❖ As you continue to explore and practice, you'll unlock even greater potential within PowerShell, making you more proficient and efficient in your daily tasks.

Thank you

Stephen Ngulle N

237 673 252 2872

Stephenngulle@gmail.com

