

הכרות ראשונית עם POWERSHELL



ozsaid@hotmail.com



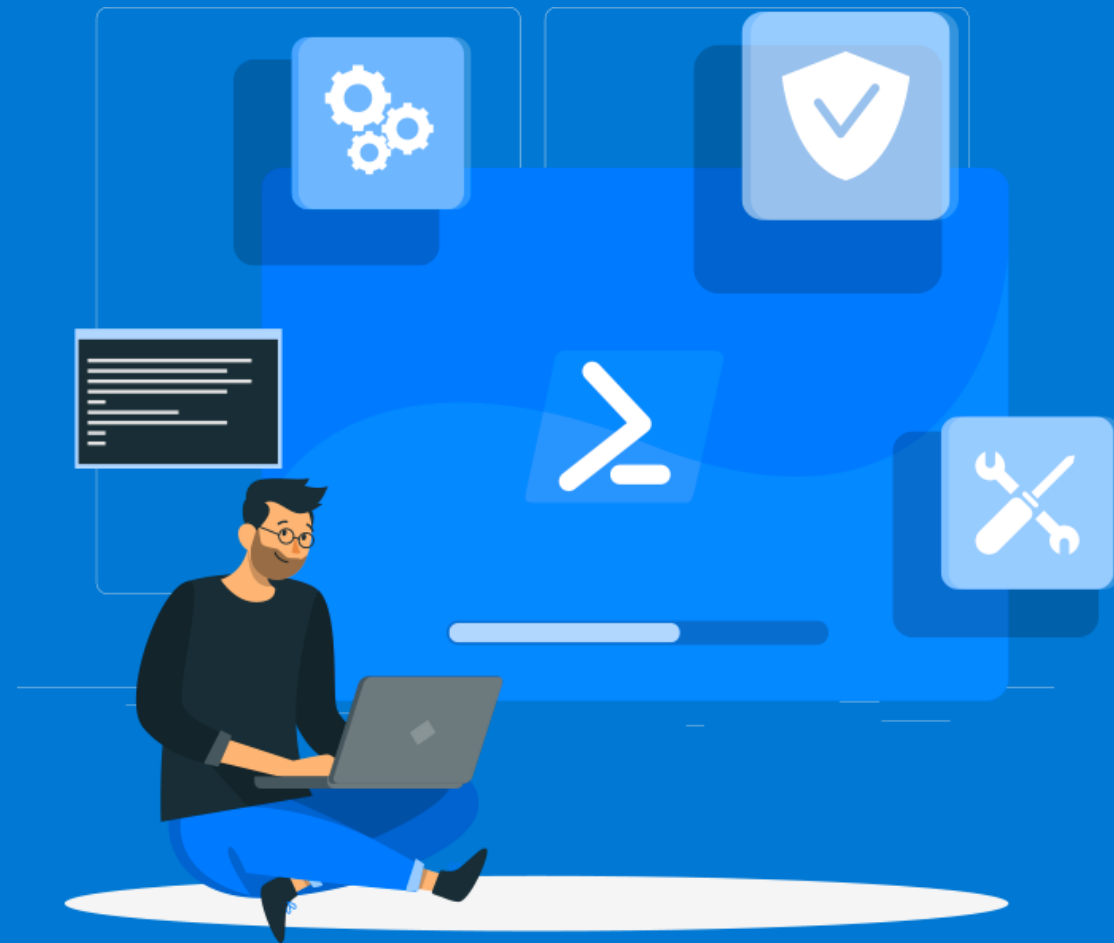
<https://www.linkedin.com/in/ozsaid/>



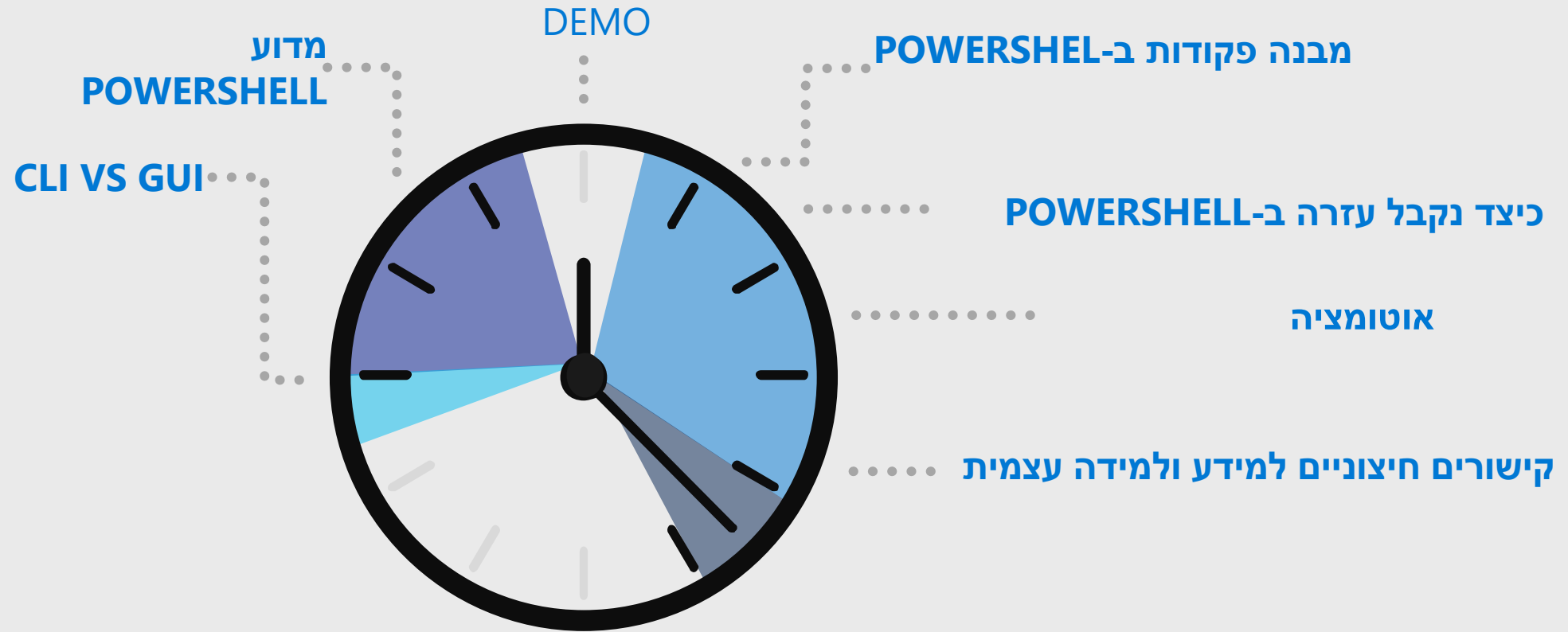
<https://github.com/ozsaid/ActiveDirectory-AutoLab>



<https://www.mrcloud.co.il/>



מפגש הכרות בס'סי



מטרת המפגש:
הכרות בסיסית עם ממשק PS והיכולות המרובות שלו



CLI vs GUI

מדוע להשתמש בממשק פקודה במקום ממשק גרפי ?

- מהיר
- לא תלוי שפה
- מאפשר עבודה מרחוק מבלי להפריע למשתמש
- לא תמיד יש ממשק גרפי זמין
- אוטומציה
- יש פקודות שלא קיימות בממשק גרפי (PING)

POWERSHELL VS CMD

OLD VS NEW....

PowerShell



PowerShell was introduced in the year 2006.

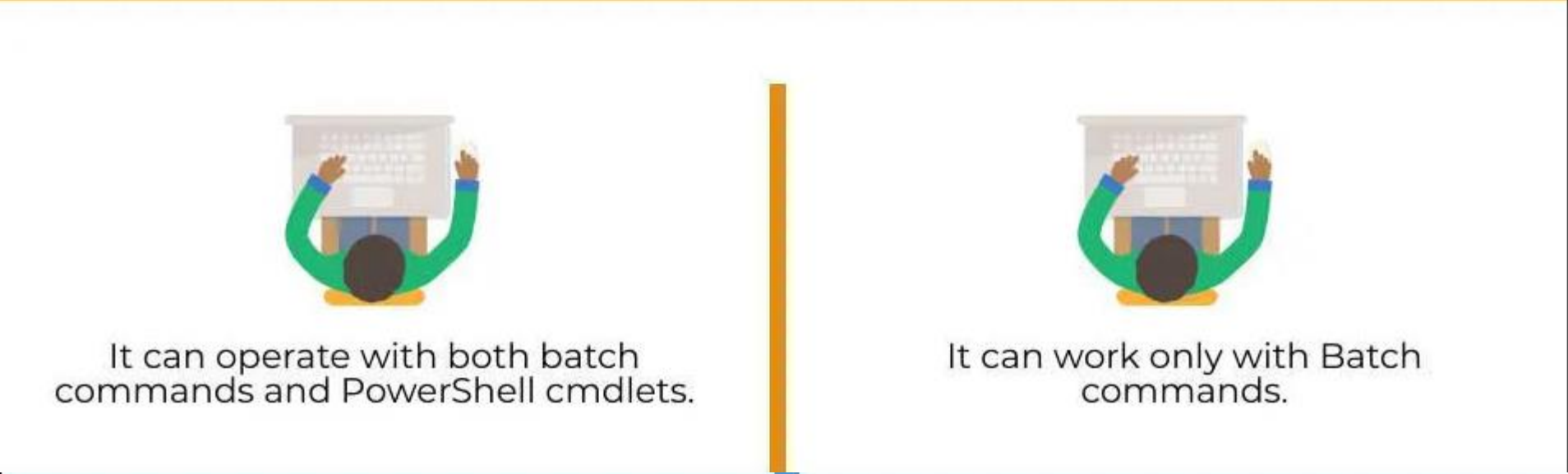
Command Prompt



cmd was introduced in the year 1981.

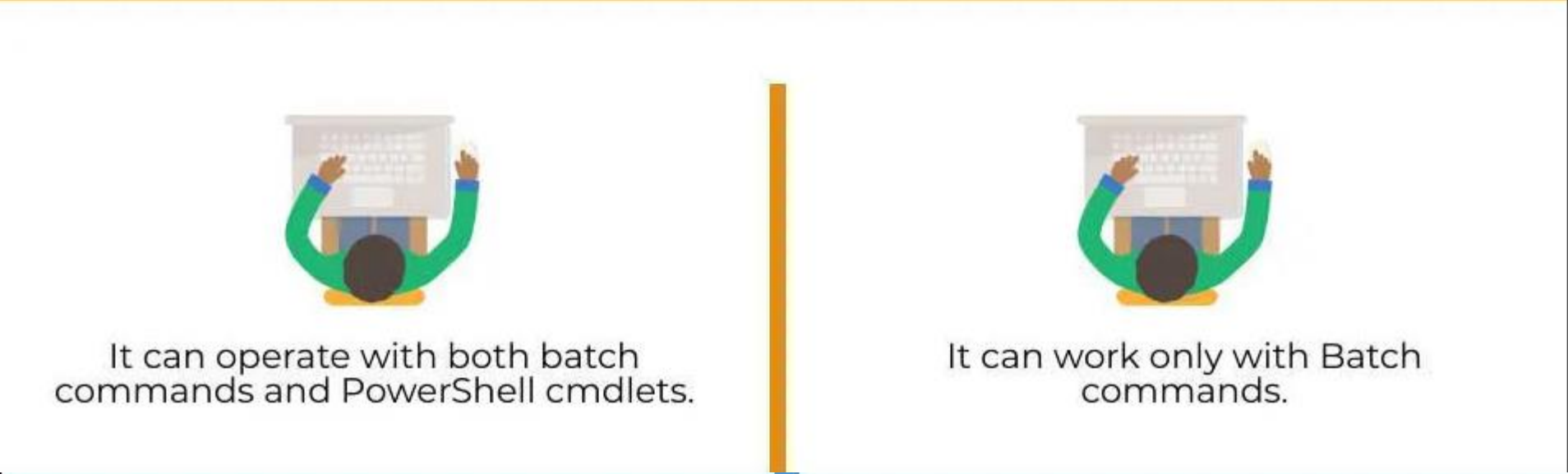
PowerShell Command Prompt

PowerShell Command Prompt



The diagram illustrates the scope of command-line tools. It is divided into two sections by a vertical orange line.

- Left Section:** A person is shown from behind, looking at a presentation board. The board displays a mix of batch commands (`dir`, `cd`) and PowerShell cmdlets (`Get-Childitem`, `Set-Location`). Below the board, the text reads: "It can operate with both batch commands and PowerShell cmdlets."
- Right Section:** A person is shown from behind, looking at a presentation board. The board displays only batch commands (`dir`, `cd`). Below the board, the text reads: "It can work only with Batch commands."

[illegible]

```

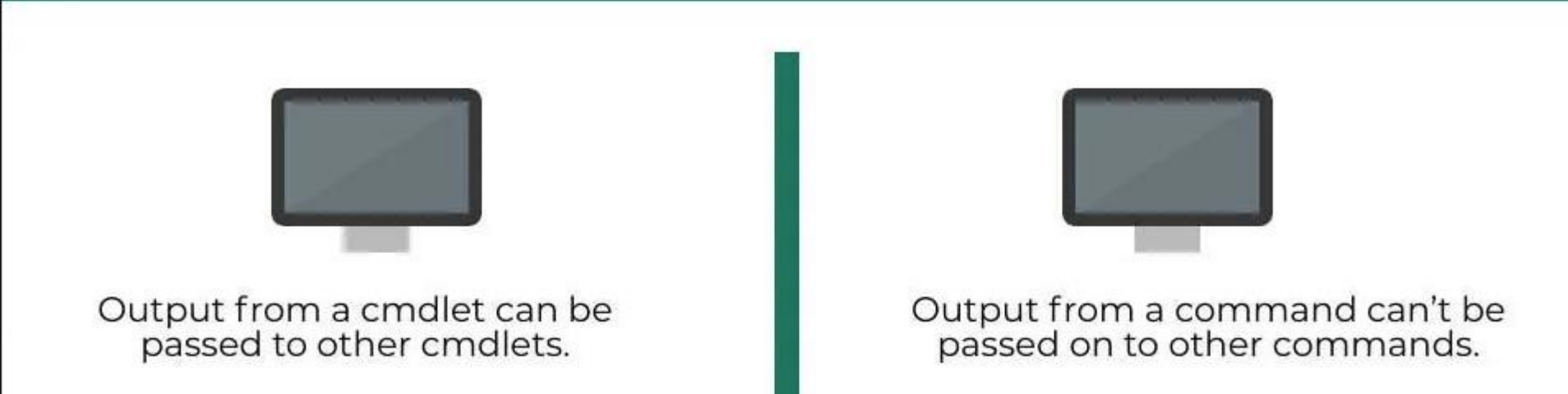
graph LR
    PowerShell[PowerShell] --> CommandPrompt[Command Prompt]
  
```

```

graph LR
    PowerShell[PowerShell] --> CommandPrompt[Command Prompt]
  
```



The diagram is divided into two sections by a vertical green line. The left section shows a monitor icon above the text: "Output from a cmdlet can be passed to other cmdlets." The right section shows a monitor icon above the text: "Output from a command can't be passed on to other commands."



The diagram illustrates the difference in output handling between cmdlets and commands. On the left, a monitor icon is positioned above the text "Output from a cmdlet can be passed to other cmdlets." On the right, another monitor icon is positioned above the text "Output from a command can't be passed on to other commands." A vertical green line separates the two scenarios.

PowerShell

Command Prompt



Output is in the form of an object.



Output from a command is just text.

PowerShell

Command Prompt



Can execute sequence of cmdlets put together in a script.



In cmd a command must be finished before the next command is run.

PowerShell

Command Prompt



It has access to programming libraries as it is built on .net framework.



No such access to libraries.

PowerShell

Command Prompt



Can integrate directly with WMI.



Need some external plugin for WMI interaction.



Can connect with Microsoft cloud products.



Doesn't have the ability to connect with MS online products.

PowerShell

Command Prompt



Supports Linux Systems.



It doesn't support Linux systems.

DEMO

MAKE POWERSHELL TALK...

POWERSHELL VERSIONS

Version	Release date	Notes
PowerShell 7.2	November 2021	Built on .NET 6.0.
PowerShell 7.1	November 2020	Built on .NET 5.0.
PowerShell 7.0	March 2020	Built on .NET Core 3.1.
PowerShell 6.0	September 2018	Built on .NET Core 2.0. First release that's installable on Windows, Linux, and macOS.
PowerShell 5.1	August 2016	Released in Windows 10 Anniversary Update and Windows Server 2016 and as part of Windows Management Framework (WMF) 5.1.
PowerShell 5.0	February 2016	Integrated in Windows 10 version 1511. Released in Windows Management Framework (WMF) 5.0. Can be installed on Windows Server 2008 R2, Windows Server 2012, Windows 10
PowerShell 4.0	October 2013	Integrated in Windows 8.1 and Windows Server 2012 R2.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Mahesh> $PSVersionTable

Name                           Value
----                           -
PSVersion                      5.1.19041.906
PSEdition                      Desktop
PSCompatibleVersions           {1.0, 2.0, 3.0, 4.0...}
BuildVersion                   10.0.19041.906
CLRVersion                     4.0.30319.42000
WSManStackVersion              3.0
PSRemotingProtocolVersion      2.3
SerializationVersion           1.1.0.1
```

Windows PowerShell applications

- Windows PowerShell console includes:
 - Basic command-line interface.
 - Maximum support for PowerShell features.
 - Minimal editing capabilities.
- Windows PowerShell ISE includes:
 - Script editor and console combination.
 - Rich editing capabilities.
- PowerShell Core doesn't support Windows PowerShell ISE. It uses VS Code with the PowerShell extension.

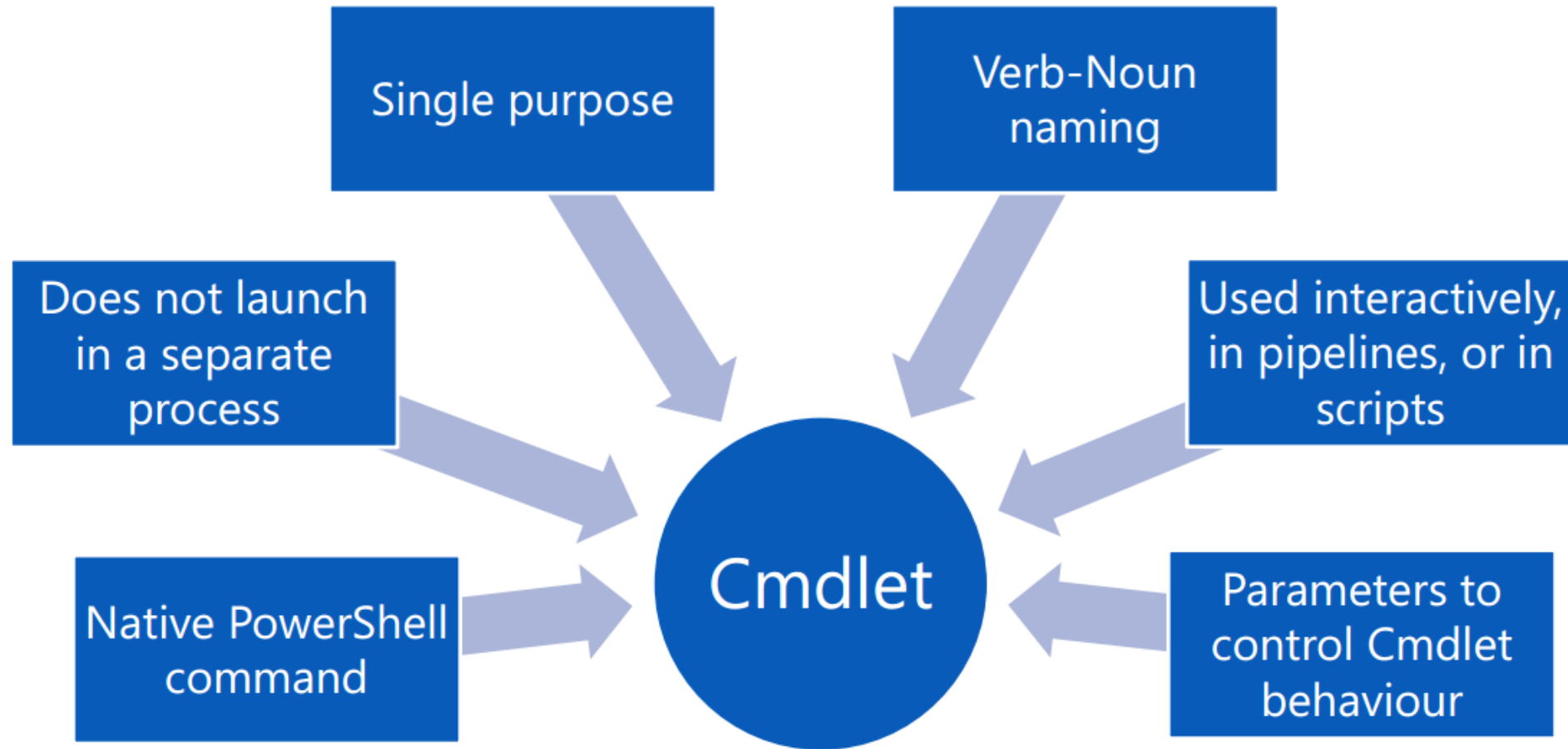
- When using PowerShell, you should:
 - Install and use PowerShell side-by-side with Windows PowerShell:
 - PowerShell uses a separate installation path and executable name (pwsh.exe).
 - PowerShell uses a separate PSModulePath, profile, and event logs.
 - You identify the PowerShell version by using **\$PSVersionTable**.
 - Run PowerShell using Administrative credentials:
 - 64-bit operating systems include both 64-bit and 32-bit versions of PowerShell.
 - The Windows title bar must display **Administrator** if you need administrative privileges in Windows PowerShell.
 - When UAC is enabled, you must right-click the application icon or activate its context menu to run as Administrator.
 - Identify and modify the execution policy in PowerShell:
 - Use **Get-ExecutionPolicy** to identify the effective execution policy in PowerShell.
 - Be aware that **Restricted** is the default for Windows clients and **RemoteSigned** is the default for Windows servers.

Using Visual Studio Code with PowerShell

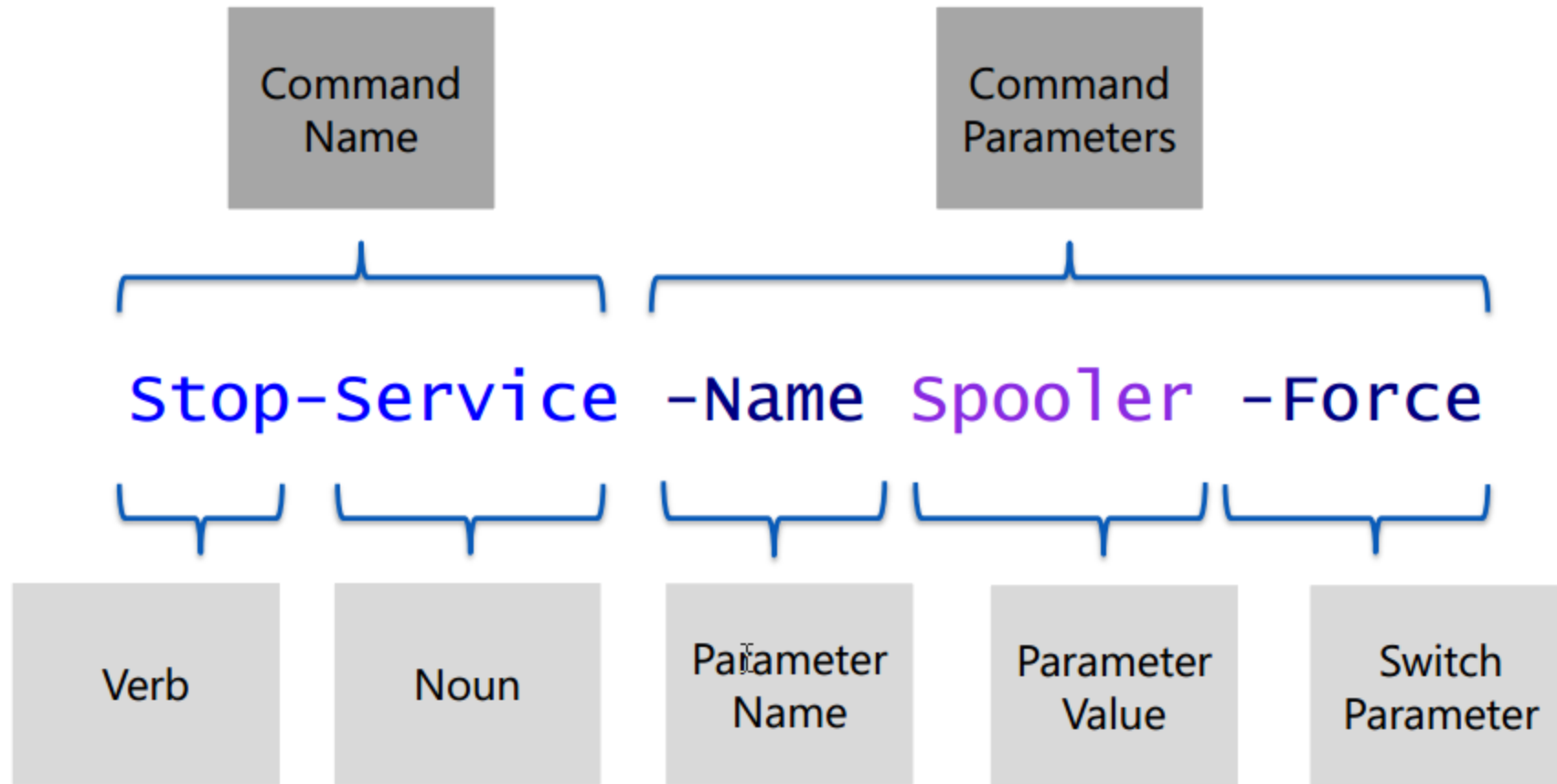
- Visual Studio Code is a Microsoft Script Editor that offers a similar experience to the PowerShell ISE when you're using the PowerShell extension add-on.
- Supports the following:
 - PowerShell Core 6, 7, and newer for Windows, macOS, and Linux.
 - PowerShell 5.1 for Windows.
- Supports ISE mode, which enables:
 - Mapping keyboard functions in VS Code so they match those used in the ISE.
 - Replicating the VS Code user interface to resemble the ISE.
 - Enabling ISE-like tab completion.
 - Providing several ISE themes to make the VS Code editor look like the ISE.

מבנה פקודות POWERSHELL

What is a Cmdlet?



Anatomy of a Cmdlet



Cmdlet Examples

```
PS C:\> Get-Process
```

Handles	NPM(K)	PM(K)	WS(K)	VM(M)	CPU(s)	Id	ProcessName
83	7	1084	4124	45	0.09	7784	armsvc
179	13	1892	8216	89	0.66	6540	BDAppHost
143	12	1840	7320	76	0.22	11148	BDExtHost
...							

```
PS C:\> Restart-Service -Name Spooler -Verbose
```

```
VERBOSE: Performing the operation "Restart-Service" on target "Print Spooler (Spooler)".
```

```
PS C:\> Test-Connection -ComputerName 2012R2-MS -Count 1 -Quiet  
True
```


Cmdlet Syntax - Command Name

Syntax Definition

```
<Command-Name> -<Required Parameter Name> <Required Parameter Value>  
[-<Optional Parameter Name> <Optional Parameter Value>]  
[-<Optional Switch Parameters>]  
[-<Optional Parameter Name>] <Required Parameter Value>  
<Multiple Parameter Values>[]
```

Syntax Sample

```
PS C:\> Get-Command -Name Add-Computer -Syntax
```

```
Add-Computer [-DomainName] <string> -Credential <pscredential> [-ComputerName  
<string[]>] [-LocalCredential  
<pscredential>] [-UnjoinDomainCredential <pscredential>] [-OUPath <string>] [-  
Server <string>] [-Unsecure] [-Options  
<JoinOptions>] [-Restart] [-PassThru] [-NewName <string>] [-Force] [-WhatIf] [-  
Confirm] [<CommonParameters>]
```

Common Parameters (with alias in parenthesis)

-Debug (db)	Displays programmer-level detail
-ErrorAction (ea)	Determines how cmdlet responds to errors
-ErrorVariable (ev)	Stores error messages in a specified variable
-OutVariable (ov)	Stores output in a specified variable
-OutBuffer (ob)	Determines number of output objects to accumulate in a buffer
-PipelineVariable (pv)	Stores value of current pipeline* element as a variable
-Verbose (vb)	Displays detailed information
-WarningAction (wa)	Determines how cmdlet responds to warnings
-WarningVariable (wv)	Stores warnings in a specified variable

Example:
Common
Parameters
in Use -
ErrorAction

```
PS C:\> Get-Process Netlogon
```

```
Get-Process : Cannot find a process with the name "Netlogon".  
Verify the process name and call the cmdlet again.
```

```
At line:1 char:1
```

```
+ Get-Process Netlogon
```

```
+ ~~~~~
```

```
    + CategoryInfo          : ObjectNotFound:  
(Netlogon:String) [Get-Process], ProcessCommandException
```

```
PS C:\>
```

```
PS C:\> Get-Process Netlogon -ErrorAction SilentlyContinue
```

```
PS C:\>
```



Common Parameter

Error Action

Example:
Store
command
output in a
specified
variable name

```
PS C:\> Get-FileHash .\iExploreProcesses.csv -OutVariable csvhash
```

Common
Parameter

Variable
Name

Use the variable to retrieve the command output

```
PS C:\> $csvhash
```

Algorithm	Hash	Path
-----	----	----
SHA256	6A78...	C:\iExploreProcesses.csv

Note: \$ prefix denotes a variable in PowerShell

Risk Mitigation Parameters

- Many cmdlets also offer risk mitigation parameters
- Typically when the cmdlet changes the system or application

-WhatIf (wi)	Displays message describing the effect of the command, instead of executing the command
-Confirm (cf)	Prompts for confirmation before executing command

Termination Characters

- To complete a command, use either a:
 - Newline character (enter) , or a
 - Semi-colon



- Semi-colon can be used to execute more than one statement on a single line

Example:
Use
command
termination
character

Semi-colon command termination

```
PS C:\> Get-Service BITS ; Get-Process System
```

Status	Name	DisplayName
-----	----	-----
Running	BITS	Background Intelligent Transfer Ser...

```
Id      : 4
Handles : 1308
CPU     : 1213.59375
Name    : System
```

Line Continuation

- When a statement is not syntactically complete and there is a newline character, PowerShell enters a line continuation



- Still in the same statement
- Complete syntax and include an empty line to finish the statement and execute
- **Ctrl-C** to break out and abort statement and line continuation
 - Useful when line continuation is accidental (Ctrl-C followed by Up-Arrow gets you back)

Example:
Line
Continuation

```
PS C:\> "This is a multi-line  
>> string that continues  
>> on several lines  
>> until the syntax is completed"  
>>
```

```
This is a multi-line  
string that continues  
on several lines  
until the syntax is completed
```

```
PS C:\>
```

Getting Some Help

Get-Command

- Discover Commands (cmdlets, functions, scripts, aliases)
- Can show command syntax
- Can also discover external commands (.exe, .cpl, .msc)



Example:
Wildcard
in Name

```
PS C:\> Get-Command -Name *user*
```

CommandType	Name
-----	----
Function	UpdateDefaultPreferencesWi...
Cmdlet	Get-WinUserLanguageList
Cmdlet	New-WinUserLanguageList
Cmdlet	Set-WinUserLanguageList
Cmdlet	Test-UserGroupMembership
Application	DsmUserTask.exe
Application	quser.exe
Application	UserAccountBroker.exe
Application	UserAccountControlSettings...
Application	userinit.exe

Example:
List Cmdlets
By Verb

```
PS C:\> Get-Command -Verb Get
```

CommandType	Name	ModuleName
-----	----	-----
Alias	Get-GPPermissions	GroupPolicy
Alias	Get-ProvisionedAppxPackage	Dism
Function	Get-AppBackgroundTask	AppBackgroundTask
...		

Example:
List
Cmdlets
By Noun

```
PS C:\> Get-Command -Noun Service
```

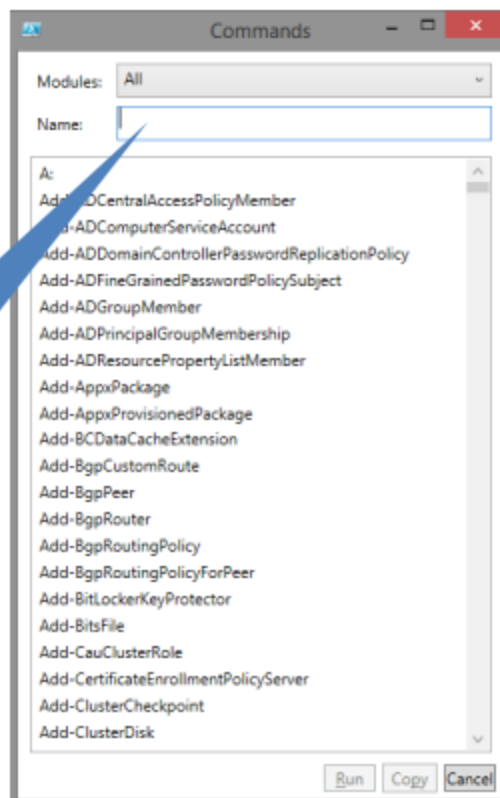
CommandType	Name	ModuleName
-----	----	-----
Cmdlet	Get-Service	Microsoft.PowerShell.Management
Cmdlet	New-Service	Microsoft.PowerShell.Management
Cmdlet	Restart-Service	Microsoft.PowerShell.Management
Cmdlet	Resume-Service	Microsoft.PowerShell.Management
...		

Show-Command

- Show-Command cmdlet launches GUI Command Browser
- Populate Parameters and Insert or Execute

PS C:\> Show-Command

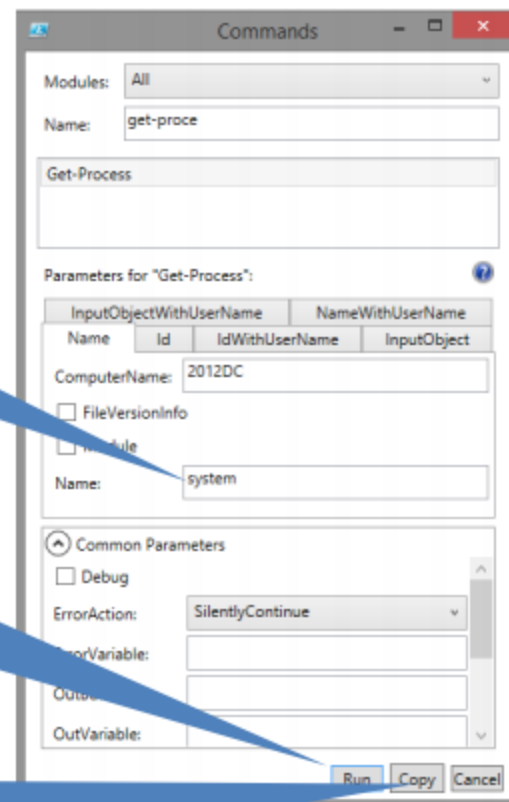
Start Typing
Command
Name
and/or click
on command
in list



Fill in
Parameters

Execute
Command
Directly

Insert
Command
with
Parameters
Populated



PS C:\> Get-Process -ComputerName 2012DC
-Name system -ErrorAction SilentlyContinue

Get-Help

- Cmdlet help
- Concept Help
- Command Examples
- Detailed Syntax

Example:
Help for
Cmdlets –
Full View

```
PS C:\> Get-Help Get-ChildItem
PS C:\> Get-Help Get-ChildItem -Full
PS C:\> Get-Help Get-ChildItem -Examples
PS C:\> Get-Help Get-ChildItem -Detailed
```

Default Help Sections (no params)	All Help Sections (-Full)
NAME SYNOPSIS SYNTAX DESCRIPTION RELATED LINKS REMARKS	NAME SYNOPSIS SYNTAX DESCRIPTION PARAMETERS INPUTS OUTPUTS NOTES EXAMPLES RELATED LINKS

Example:
Listing all aliases

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

CommandType	Name	ModuleName
-----	----	-----
Alias	% -> ForEach-Object	
Alias	? -> Where-Object	
Alias	ac -> Add-Content	
Alias	asnp -> Add-PSSnapin	

Built-in Aliases

- PowerShell provides short names for frequently used cmdlets
- Ease of PowerShell adoption for Windows cmd.exe and *Nix administrators
- Saves time when typing interactive commands

Example:
Using built-in aliases

Full cmdlet name

```
PS C:\> Get-ChildItem C:\windows
```

Cmdlet Alias - Windows

```
PS C:\> dir C:\windows
```

Cmdlet Alias - *nix

```
PS C:\> ls C:\windows
```

Cmdlet Alias - PowerShell

```
PS C:\> gci C:\windows
```

Example:
Creating a
custom alias

New Alias (list) for Get-ChildItem cmdlet

```
PS C:\> New-Alias -Name list -Value Get-ChildItem
```

Using New Alias (list)

```
PS C:\> list
```

Directory: C:\


Mode	LastWriteTime	Length	Name
----	-----	-----	----
d----	5/09/2013 1:40 PM		Intel
d-r--	21/10/2013 1:31 PM		Program Files
d-r--	10/12/2013 10:26 AM		Program Files (x86)
d----	1/12/2013 1:32 PM		Scripts

Can you Create you own?

What are modules?

- Modules:
 - Are containers for related cmdlets.
 - Are provided as part of management tools for various software packages.
 - Must be loaded into your current session.
 - May only support specific operating systems.
- Windows PowerShell version 3.0 and newer support autoloading.
- Windows PowerShell and PowerShell Core support different module paths as indicated by the *\$Env:PSModulePath* environment variable.

POWERSHELL GALLERY

 PowerShell Gallery

PackagesPublishStatisticsDocumentation

Sign in

Welcome to the PowerShell Gallery

The central repository for sharing and acquiring PowerShell code including PowerShell modules, scripts, and DSC resources.


Search PowerShell packages:

Az, etc...

10,594
Unique Packages

6,073,878,058
Total package downloads

150,667
Total packages



DEMO : INSTALL NEW MODULE

Working with the Windows PowerShell pipeline

- Consider the following regarding the PowerShell pipeline:
 - Windows PowerShell runs commands in a pipeline.
 - Each console command line is a pipeline.
 - Commands are separated by a pipe character (|).
 - Commands execute from left to right.
 - Output of each command is *piped* (passed) to the next.
 - The output of the last command in the pipeline is what you notice on your screen.
 - Piped commands typically follow the pattern **Get | Set**, **Get | Where**, or **Select | Set**.

Pipeline output

- Windows PowerShell commands produce objects as their output.
- An object is like a table of data in memory.
- Allows the **Get | Set** pattern to work.

The diagram illustrates a PowerShell pipeline output. A table represents the data, with a blue header row and two data rows. Annotations in red text with arrows point to specific parts of the table: 'Object' points to the first column, 'Property' points to the header row, and 'Collection' points to the entire table structure.

Name	Status	DisplayName
WinRM	Running	Windows Remote Management
VDS	Running	Virtual Disk

PIPELINE DEMO

PSPROVIDERS – REGISTRY DEMO

Where to Start Learning....

Free Resources – PowerShell.org - אתר מעולה ללמוד ולהכיר את השפה

<https://powershell.org/free-resources/>



כמה קיצורי דרך שימושיים :

<https://cdn.comparitech.com/wp-content/uploads/2018/08/Comparitech-Powershell-cheatsheet.pdf>



כמה דפים שיעזרו לעשות סדר בראש מומלץ להדפיס ... הפעם מאת מייקרוסופט

Download Windows PowerShell 4.0 and Other Quick Reference Guides from Official Microsoft Download Center

<https://www.microsoft.com/en-us/download/details.aspx?id=42554>



קישור לספר בסיסי בנושא

<https://leanpub.com/s/g64yN-wS7v73IMoJJWilHw.pdf>



סדרת סרטונים מעולה לנושאים בסיסיים :

Learn Windows PowerShell in a Month of Lunches - YouTube

<https://www.youtube.com/playlist?list=PL6D474E721138865A>

<https://github.com/doctordns>

<https://sid-500.com/>

<https://adamtheautomator.com/tutorials/>

<https://lazyadmin.nl/>