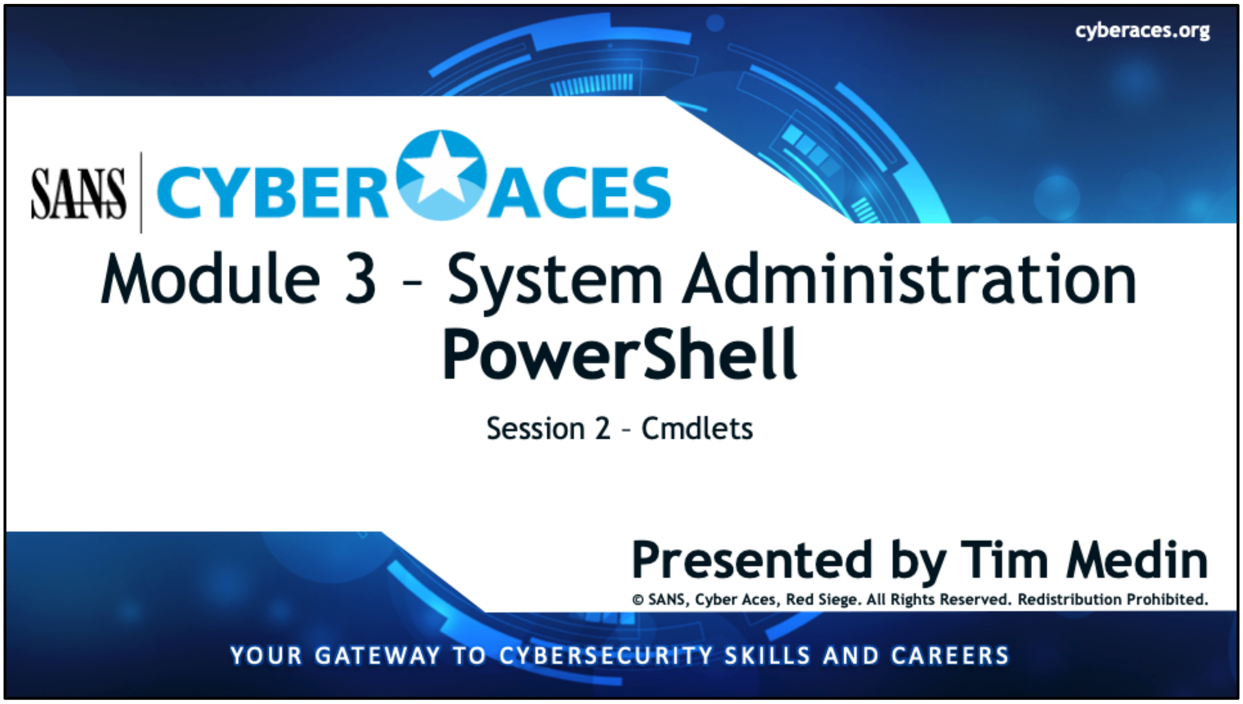


cyberaces.org

SANS | CYBER ACES

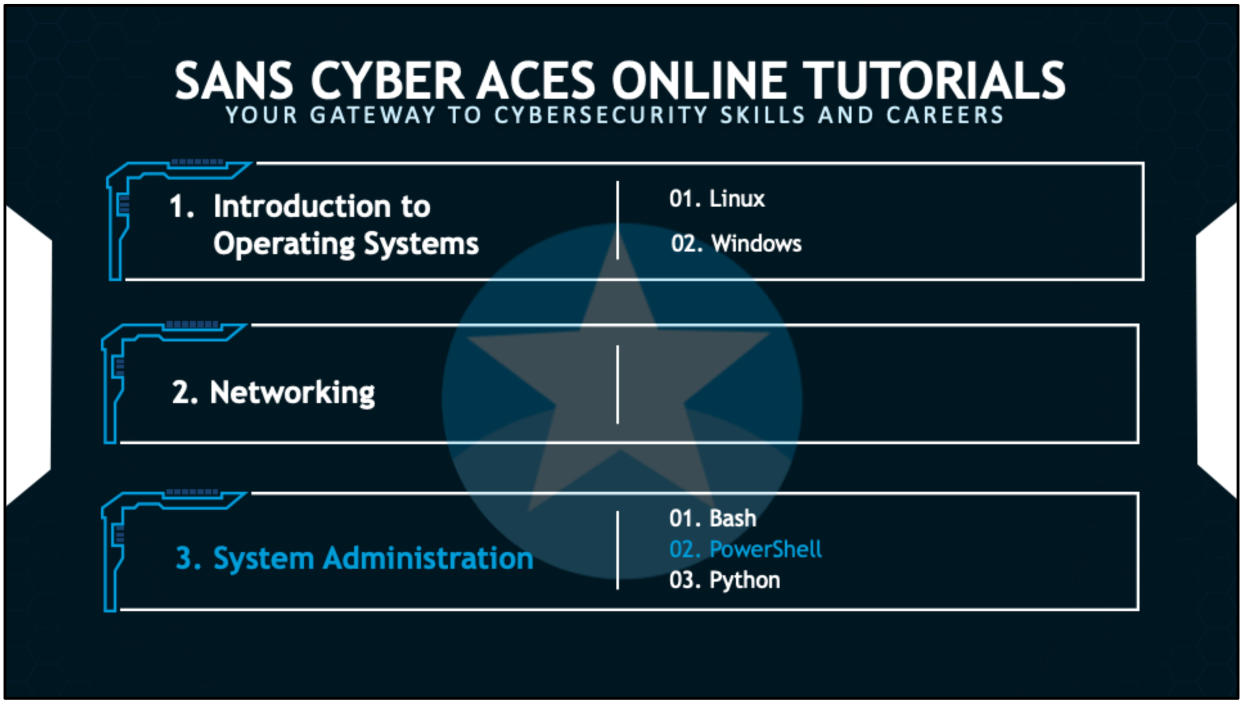
Module 3 – System Administration PowerShell

Session 2 – Cmdlets

Presented by Tim Medin

YOUR GATEWAY TO CYBERSECURITY SKILLS AND CAREERS

Welcome to Cyber Aces, Module 3! This module provides an introduction to the latest shell for Windows, PowerShell. In this session we'll discuss cmdlets.



SANS CYBER ACES ONLINE TUTORIALS

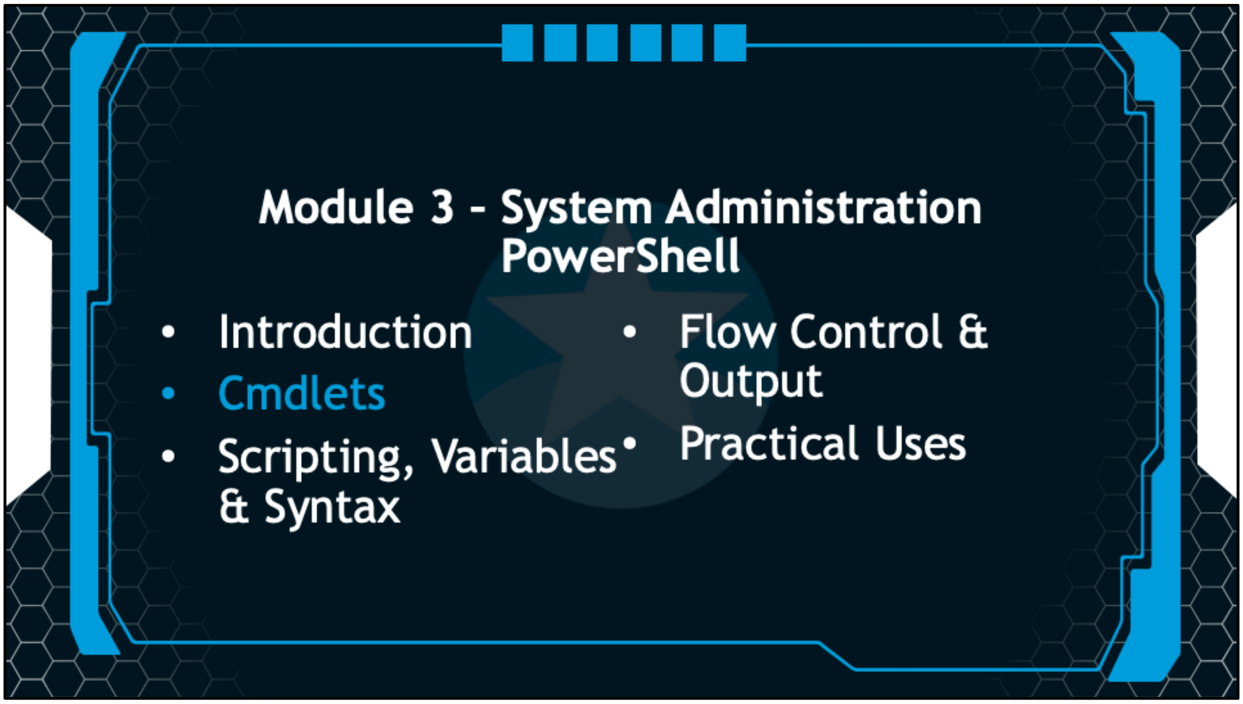
YOUR GATEWAY TO CYBERSECURITY SKILLS AND CAREERS

1. Introduction to Operating Systems
   - 01. Linux
   - 02. Windows
2. Networking
3. System Administration
   - 01. Bash
   - 02. PowerShell
   - 03. Python

This training material was originally developed to help students, teachers, and mentors prepare for the Cyber Aces Online Competition. This module focuses on the basics of what an operating systems is as well as the two predominant OS's, Windows and Linux. In this session we will provide a walkthrough of the installation a Windows VM using VMware Fusion (MacOS) and VMware Player (Windows & Linux). These sessions include hands-on labs, but before we begin those labs we need to install the operating systems used in those labs. We will be using VMware to virtualize these operating systems. You can use other virtualization technologies if you like, but instruction for their setup and use are not included in this training.

The three modules of Cyber Aces Online are Operating Systems, Networking, and System Administration.

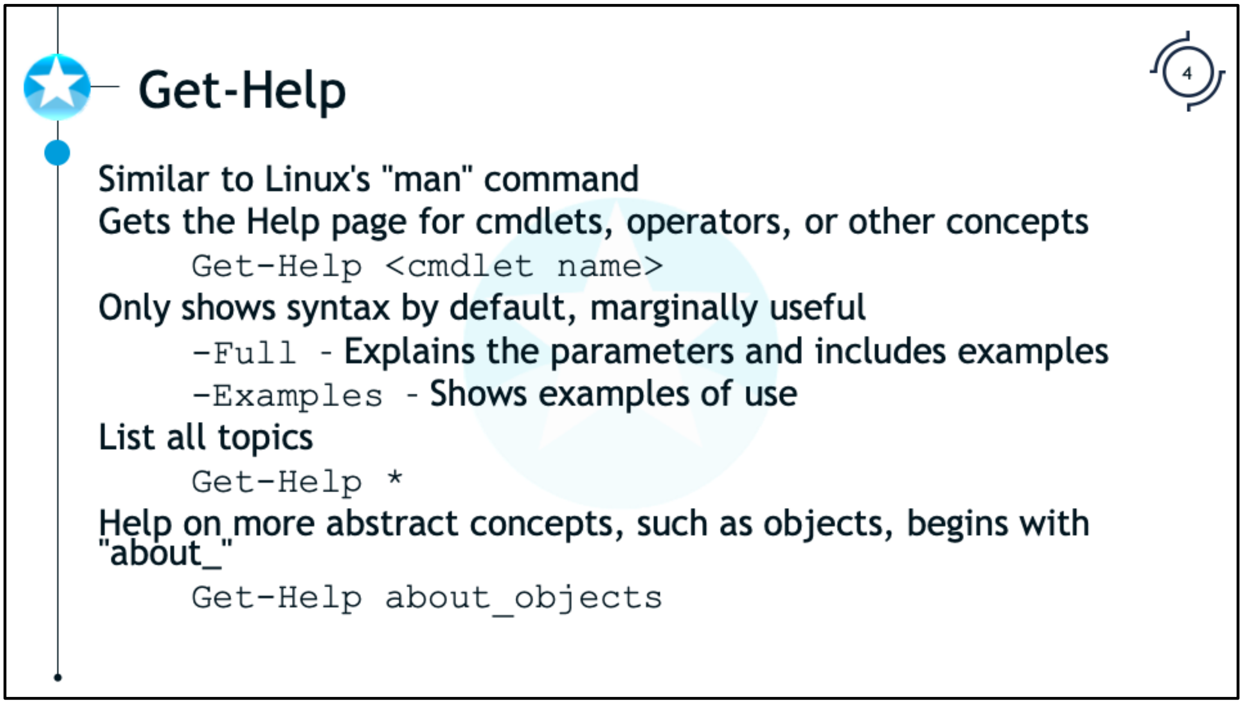
For more information about the Cyber Aces program, please visit the Cyber Aces website at https://CyberAces.org/.



Module 3 - System Administration PowerShell

- Introduction
- Cmdlets
- Scripting, Variables & Syntax
- Flow Control & Output
- Practical Uses

Is this section, we'll spend time discussing cmdlets. We'll cover the help system and how to find cmdlets. We'll also discuss aliases for cmdlets and some of the common cmdlets.



Get-Help

Similar to Linux's "man" command

Gets the Help page for cmdlets, operators, or other concepts

```
Get-Help <cmdlet name>
```

Only shows syntax by default, marginally useful

- `-Full` - Explains the parameters and includes examples
- `-Examples` - Shows examples of use

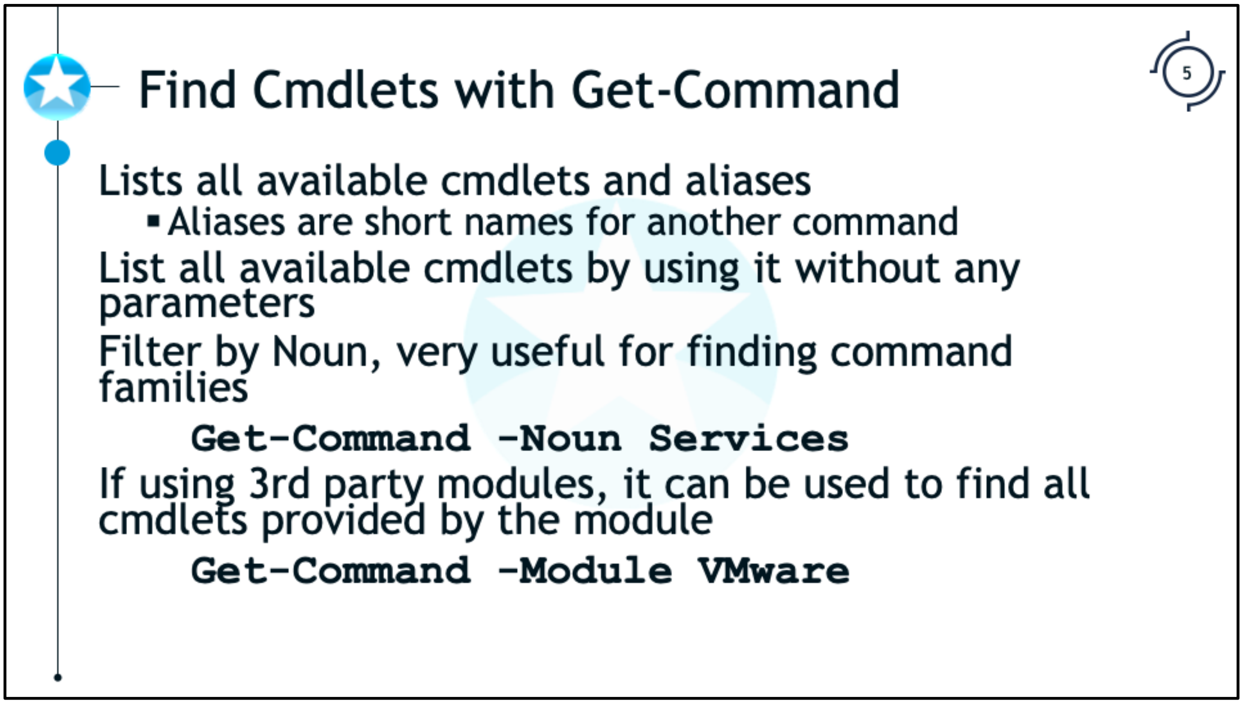
List all topics

```
Get-Help *
```

Help on more abstract concepts, such as objects, begins with "about_"

```
Get-Help about_objects
```

The most important commands to know are the ones that get more help and information. The two most important commands in this regard are Get-Help and Get-Command. The Get-Help cmdlet is the PowerShell equivalent of "man" on Linux. It displays information on PowerShell's commands and concepts. When used with the name of a cmdlet, it returns the synopsis and syntax for the command. To get examples of the cmdlet in use, use the "-Examples" switch. For the full output, including synopsis, syntax, parameter descriptions and examples, use the "-Full" switch. The formatting of Get-Help's output is very similar to that of Linux's man.



Find Cmdlets with Get-Command

- Lists all available cmdlets and aliases
  - Aliases are short names for another command
- List all available cmdlets by using it without any parameters
- Filter by Noun, very useful for finding command families

```
Get-Command -Noun Services
```

- If using 3rd party modules, it can be used to find all cmdlets provided by the module

```
Get-Command -Module VMware
```

The Get-Command cmdlet "gets basic information about cmdlets and other elements of Windows PowerShell commands." Its most common use is to find other cmdlets based on a verb or noun by using the "-Verb" or "-Noun" parameters. To see all the commands used to manage services we can use the following command:

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

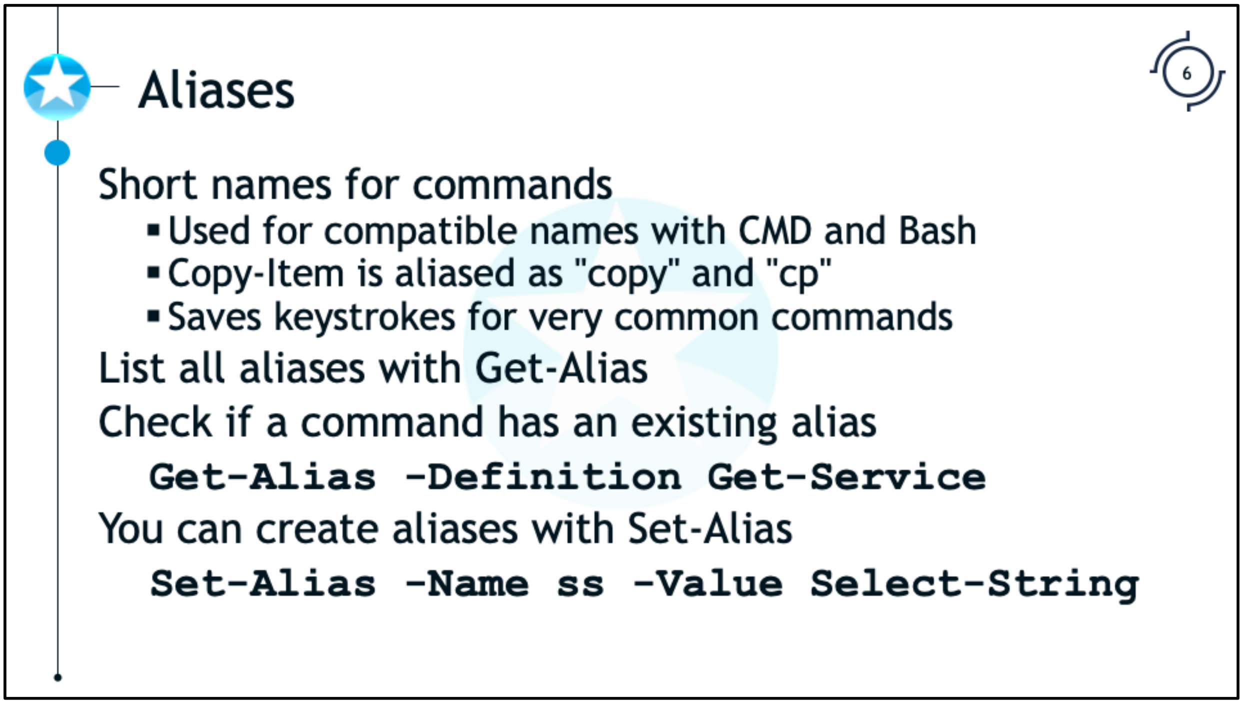
The -Verb parameter is available as well. To list all cmdlets that use the Get verb we can use the following command:

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

The -Module parameter can be used to find command specific to a loaded module. Many 3rd party products have a PowerShell interface which loads another module. We can list all loaded modules with “Get-Module”. To see the commands specific to a loaded module we use “Get-Command -Module ModuleName”.

All of these parameters take wildcard characters and they can be combined to provide a more granular search.

```
PS C:\> Get-Command -Module Vm* -Verb Get
```



Aliases

- Short names for commands
  - Used for compatible names with CMD and Bash
  - Copy-Item is aliased as "copy" and "cp"
  - Saves keystrokes for very common commands
- List all aliases with Get-Alias
- Check if a command has an existing alias

```
Get-Alias -Definition Get-Service
```

- You can create aliases with Set-Alias

```
Set-Alias -Name ss -Value Select-String
```

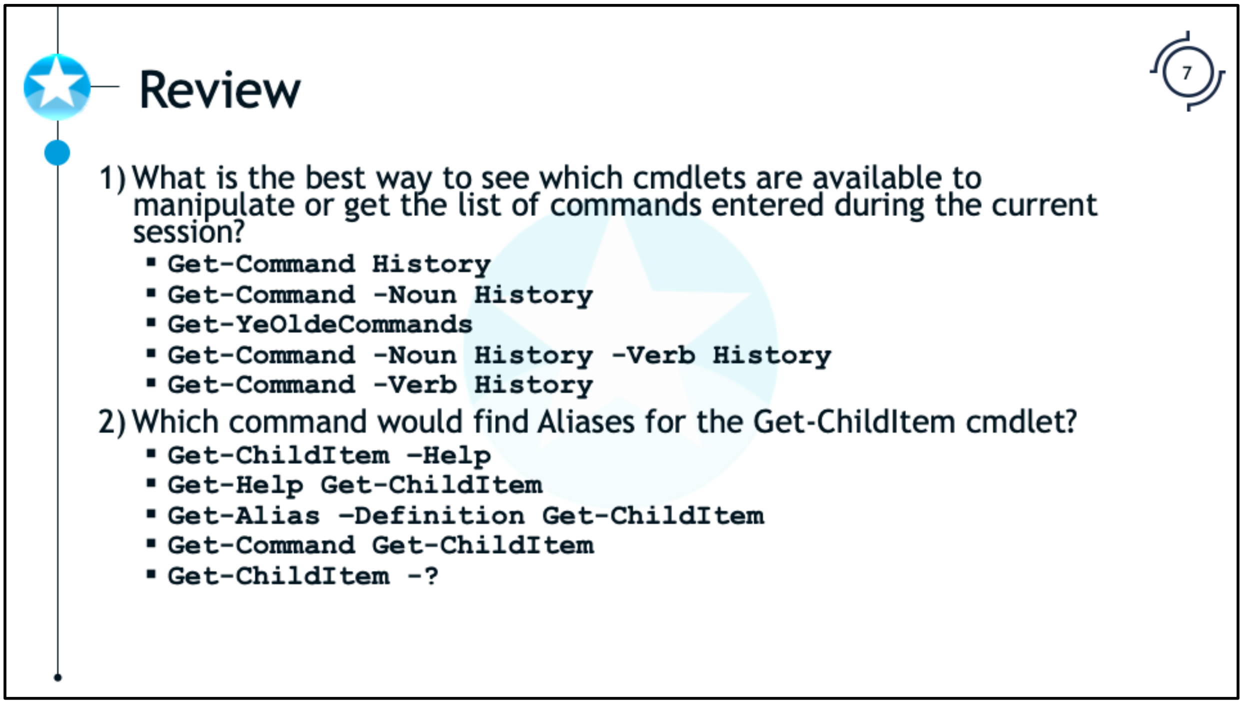
Aliases are a very handy way to simplify the commands that you use and make typing faster and more efficient. Many commands that are implemented in CMD or Bash are aliased using the respective shell's command name. To copy an item in CMD the command "copy" is used, in Bash the command is "cp". Both of these are aliases for the "Copy-Item" cmdlet.

Many times it is useful to create an alias for a commonly used command. The most commonly used command without an alias is Select-String and the common alias is ss. To create the alias for the command we use this command:

```
PS C:\> Set-Alias -Name ss -Value Select-String
```

The Set-Alias takes positional parameters, so it knows the first input is the alias name and the second is the command we want to alias. We could type this command instead.

```
PS C:\> Set-Alias ss Select-String
```



Review

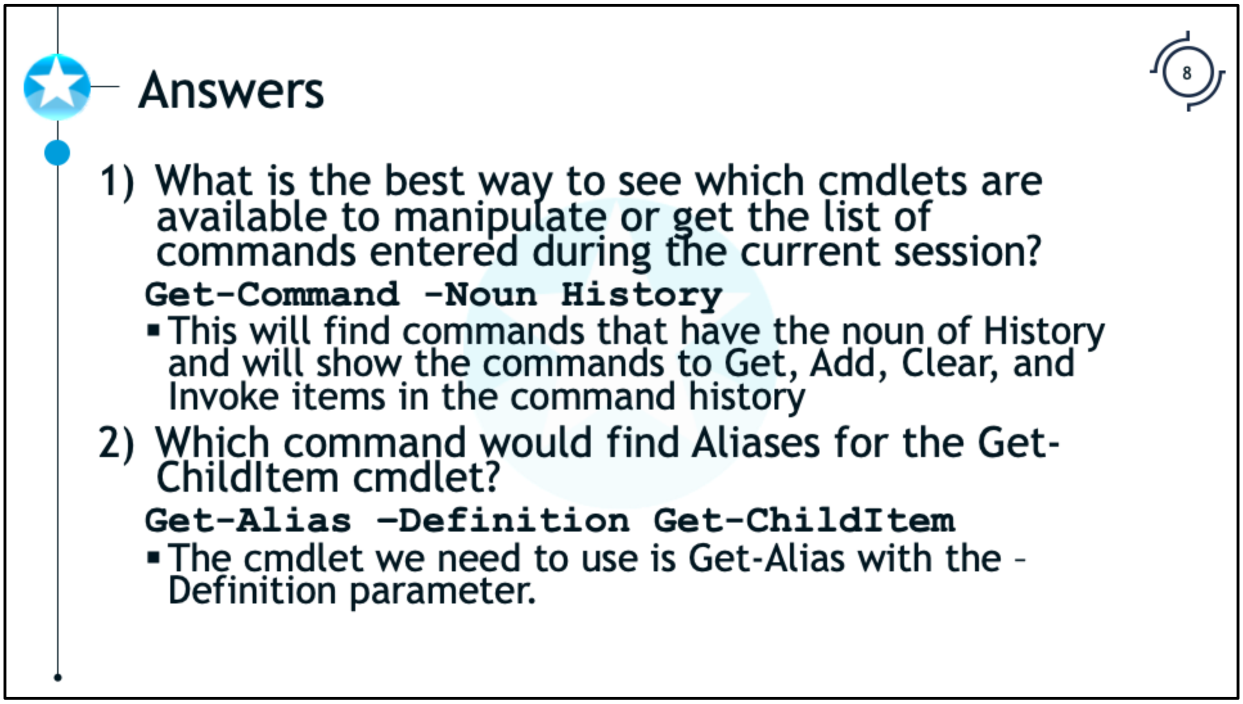
1) What is the best way to see which cmdlets are available to manipulate or get the list of commands entered during the current session?
   - `Get-Command History`
   - `Get-Command -Noun History`
   - `Get-YeOldeCommands`
   - `Get-Command -Noun History -Verb History`
   - `Get-Command -Verb History`
2) Which command would find Aliases for the Get-ChildItem cmdlet?
   - `Get-ChildItem –Help`
   - `Get-Help Get-ChildItem`
   - `Get-Alias –Definition Get-ChildItem`
   - `Get-Command Get-ChildItem`
   - `Get-ChildItem -?`

1) What is the best way to see which cmdlets are available to manipulate or get the list of commands entered during the current session?

```
Get-Command History
Get-Command -Noun History
Get-YeOldeCommands
Get-Command -Noun History -Verb History
Get-Command -Verb History
```

2) Which command would find Aliases for the Get-ChildItem cmdlet?

```
Get-ChildItem -Help
Get-Help Get-ChildItem
Get-Alias –Definition Get-ChildItem
Get-Command Get-ChildItem
Get-ChildItem -?
```



Answers

8

1) What is the best way to see which cmdlets are available to manipulate or get the list of commands entered during the current session?

   ```
   Get-Command -Noun History
   ```

   - This will find commands that have the noun of History and will show the commands to Get, Add, Clear, and Invoke items in the command history

2) Which command would find Aliases for the Get-ChildItem cmdlet?

   ```
   Get-Alias –Definition Get-ChildItem
   ```

   - The cmdlet we need to use is Get-Alias with the -Definition parameter.

1) What is the best way to see which cmdlets are available to manipulate or get the list of commands entered during the current session?

   ```
   Get-Command -Noun History
   ```

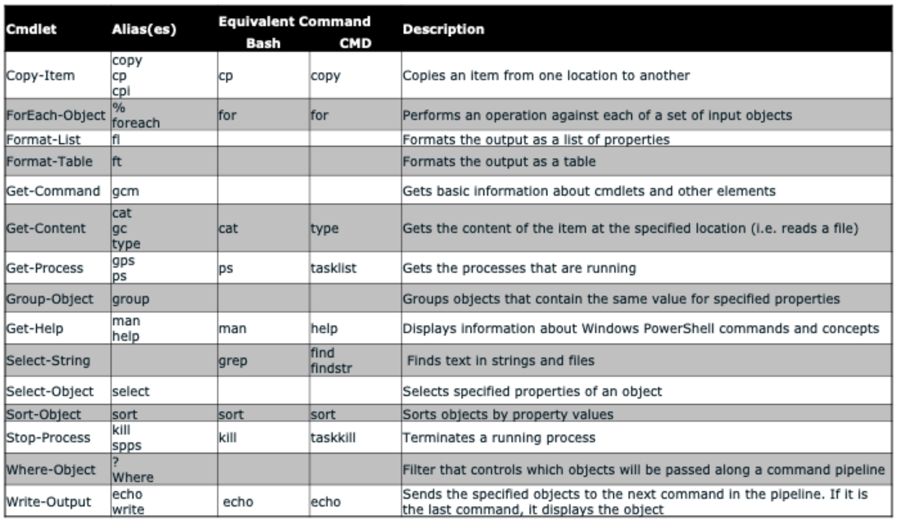
   This will find commands that have the noun of History and will show the commands to Get, Add, Clear, and Invoke items in the command history

2) Which command would find Aliases for the Get-ChildItem cmdlet?

   ```
   Get-Alias –Definition Get-ChildItem
   ```

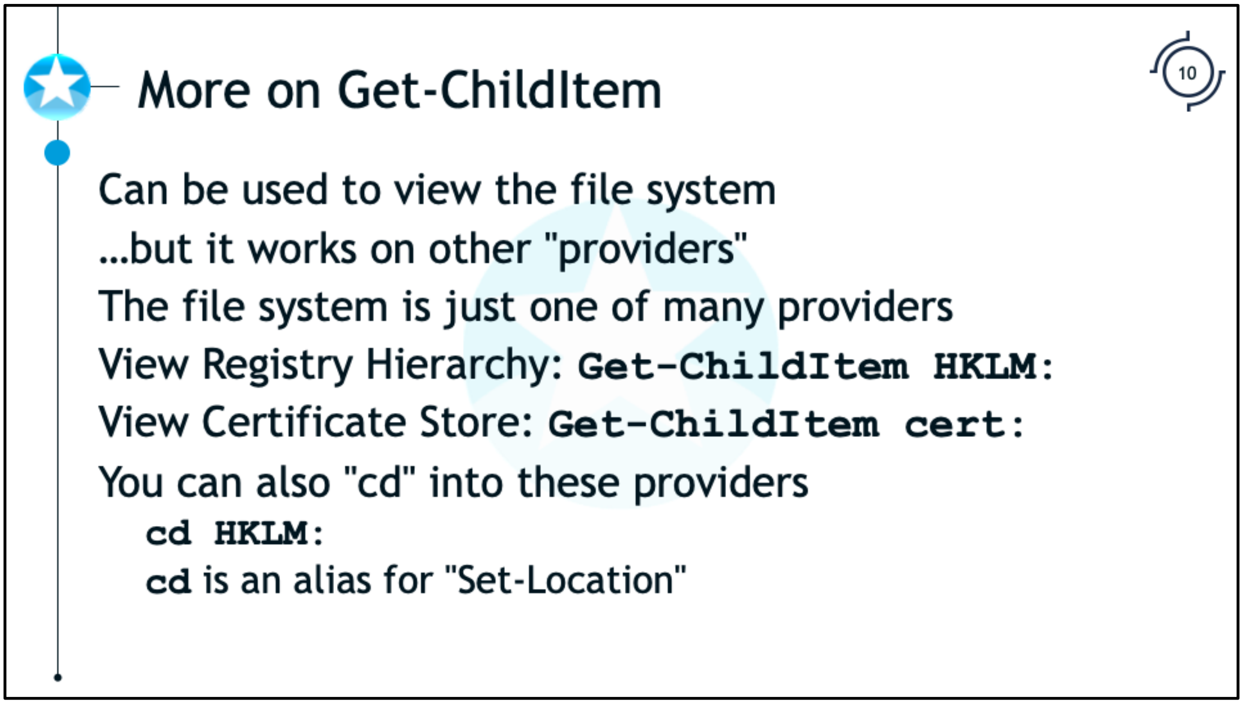
   The cmdlet we need to use is Get-Alias with the –Definition parameter.

Common Cmdlets



Cmdlet	Alias(es)	Equivalent Command		Description
		Bash	CMD	
Copy-Item	copy cp cpi	cp	copy	Copies an item from one location to another
ForEach-Object	% foreach	for	for	Performs an operation against each of a set of input objects
Format-List	fl			Formats the output as a list of properties
Format-Table	ft			Formats the output as a table
Get-Command	gcm			Gets basic information about cmdlets and other elements
Get-Content	cat gc type	cat	type	Gets the content of the item at the specified location (i.e. reads a file)
Get-Process	gps ps	ps	tasklist	Gets the processes that are running
Group-Object	group			Groups objects that contain the same value for specified properties
Get-Help	man help	man	help	Displays information about Windows PowerShell commands and concepts
Select-String		grep	find findstr	Finds text in strings and files
Select-Object	select			Selects specified properties of an object
Sort-Object	sort	sort	sort	Sorts objects by property values
Stop-Process	kill spps	kill	taskkill	Terminates a running process
Where-Object	? Where			Filter that controls which objects will be passed along a command pipeline
Write-Output	echo write	echo	echo	Sends the specified objects to the next command in the pipeline. If it is the last command, it displays the object

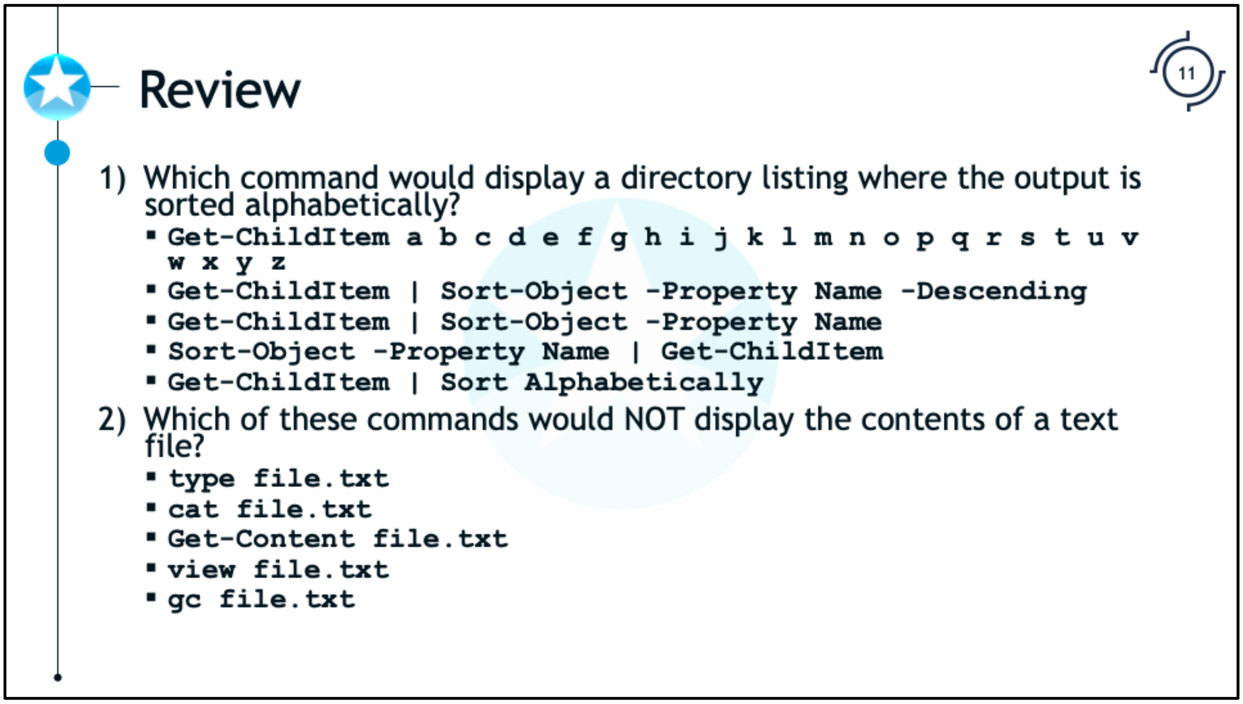
Above is a list of the common cmdlets and the equivalent commands in Bash and CMD.



More on Get-ChildItem

- Can be used to view the file system
- ...but it works on other "providers"
- The file system is just one of many providers
- View Registry Hierarchy: **`Get-ChildItem HKLM:`**
- View Certificate Store: **`Get-ChildItem cert:`**
- You can also "cd" into these providers
  - **`cd HKLM:`**
  - **`cd`** is an alias for "Set-Location"

At first glance, you might wonder why cmd.exe's "dir" command has been replaced by something as weird sounding as "Get-ChildItem". Well, "Get-ChildItem" does more than just list files and directories, and that is why the name is more generic. This cmdlet returns objects from any container, and the filesystem is just one of many containers. For example, it can also be used to list the system certificates ("Get-ChildItem cert:") and the registry ("Get-ChildItem HKLM:").



Review

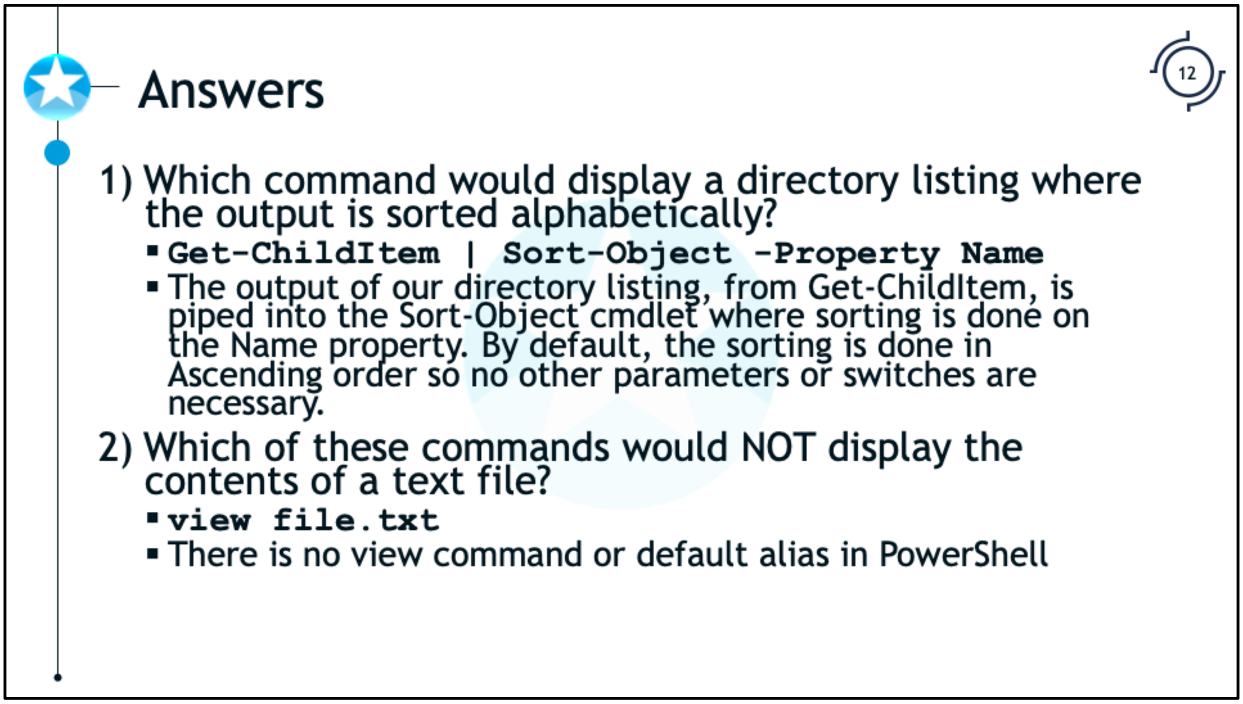
1) Which command would display a directory listing where the output is sorted alphabetically?
   - `Get-ChildItem a b c d e f g h i j k l m n o p q r s t u v w x y z`
   - `Get-ChildItem | Sort-Object -Property Name -Descending`
   - `Get-ChildItem | Sort-Object -Property Name`
   - `Sort-Object -Property Name | Get-ChildItem`
   - `Get-ChildItem | Sort Alphabetically`
2) Which of these commands would NOT display the contents of a text file?
   - `type file.txt`
   - `cat file.txt`
   - `Get-Content file.txt`
   - `view file.txt`
   - `gc file.txt`

1) Which command would display a directory listing where the output is sorted alphabetically?

```
Get-ChildItem a b c d e f g h I j k i m n o p q r s t u v w x y z
Get-ChildItem | Sort-Object -Property Name -Descending
Get-ChildItem | Sort-Object -Property Name
Sort-Object -Property Name | Get-ChildItem
Get-ChildItem | Sort Alphabetically
```

2) Which of these commands would NOT display the contents of a text file?

```
type file.txt
cat file.txt
Get-Content file.txt
view file.txt
gc file.txt
```



Answers

1) Which command would display a directory listing where the output is sorted alphabetically?
   - `Get-ChildItem | Sort-Object -Property Name`
   - The output of our directory listing, from Get-ChildItem, is piped into the Sort-Object cmdlet where sorting is done on the Name property. By default, the sorting is done in Ascending order so no other parameters or switches are necessary.
2) Which of these commands would NOT display the contents of a text file?
   - `view file.txt`
   - There is no view command or default alias in PowerShell

1) Which command would display a directory listing where the output is sorted alphabetically?

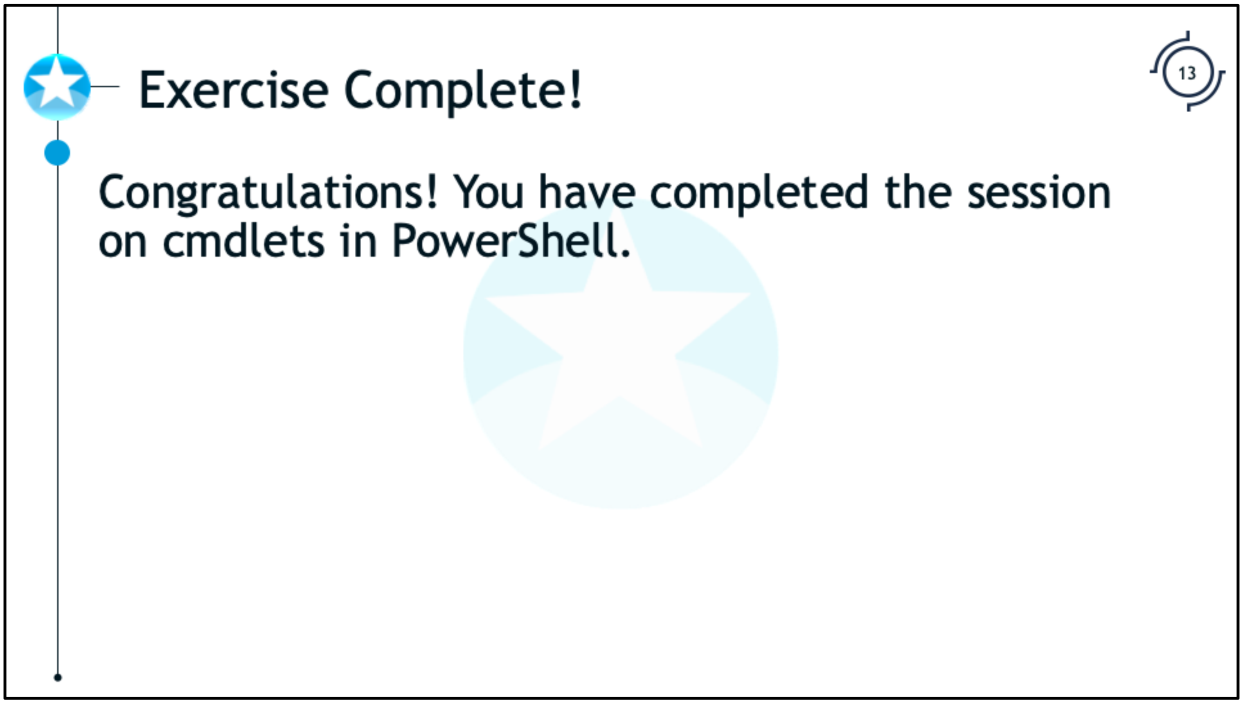
```
Get-ChildItem | Sort-Object -Property Name
```

The output of our directory listing, from Get-ChildItem, is piped into the Sort-Object cmdlet where sorting is done on the Name property. By default, the sorting is done in Ascending order so no other parameters or switches are necessary.

2) Which of these commands would NOT display the contents of a text file?

```
view file.txt
```

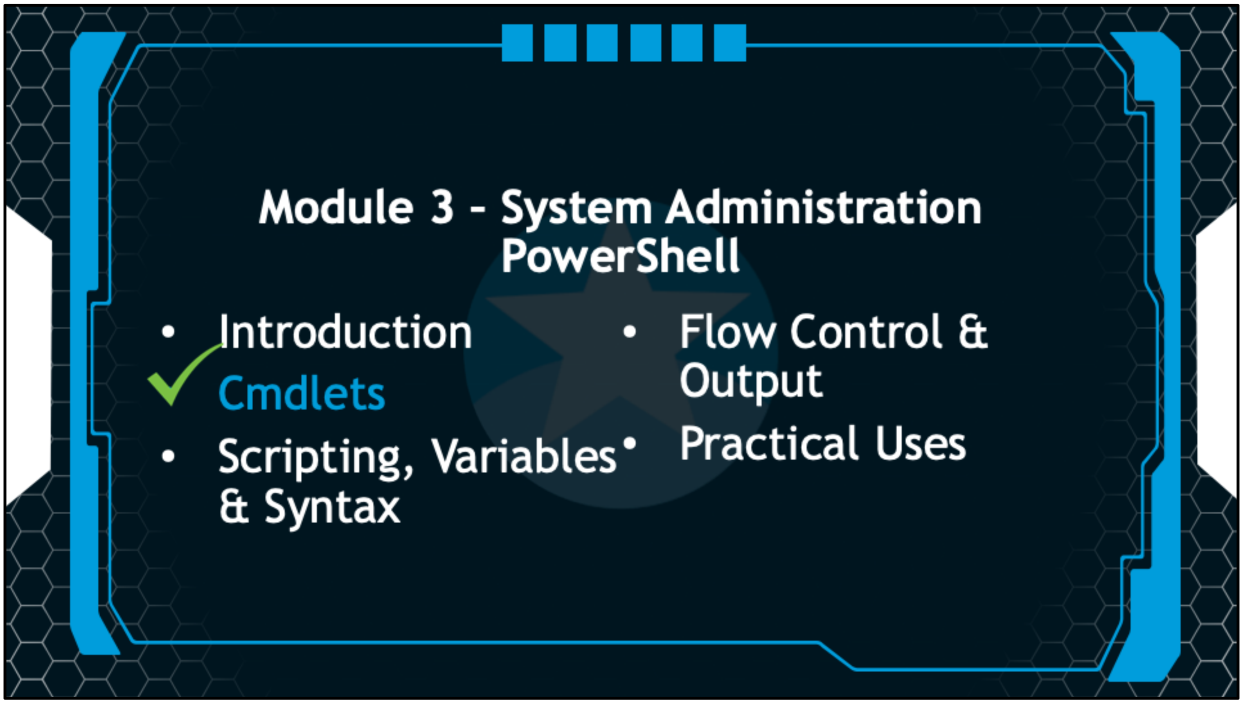
There is no view command or default alias in PowerShell



Exercise Complete!

Congratulations! You have completed the session on cmdlets in PowerShell.

Exercise Complete



Module 3 - System Administration PowerShell

- Introduction
- ✓ Cmdlets
- Scripting, Variables & Syntax
- Flow Control & Output
- Practical Uses

This portion intentionally left blank.