



LEARN POWERSHELL BY BUILDING A COOL ALARM CLOCK

Mike Nelson

VMware Explore VMTN 2022

Get-Mike

 mikenelsonio

 nelmedia

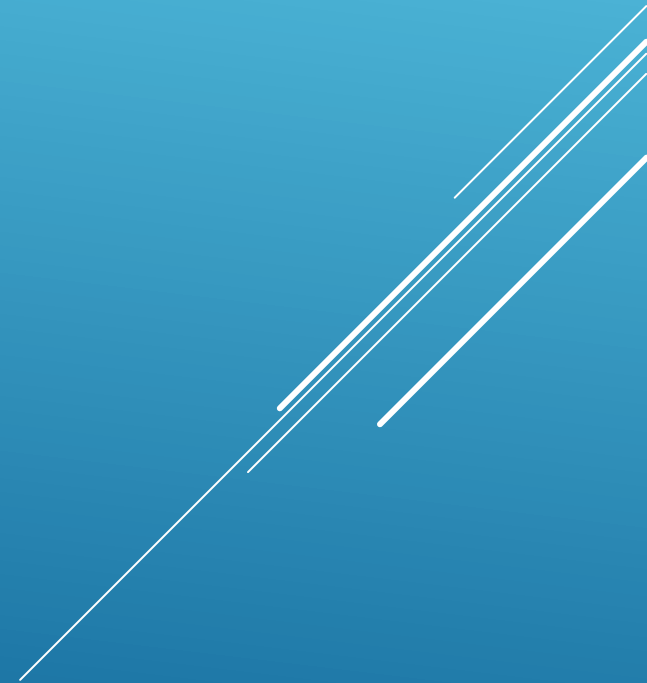
 mikenelson-io



- ▶ 35+ years in tech
- ▶ Principal Technologist @ Cohesity
- ▶ Experience from Helpdesk to Architect
- ▶ Scripter, not a coder
- ▶ Passion for community, teaching, learning
- ▶ Beer, BBQ, & Gadgets



/MYPRESENTATIONS/2022-AUGUST_VMWARE EXPLORE VMTN



GET-VERSION



≤ 5.1

Windows

.Net Framework
powershell.exe



~~6.x~~

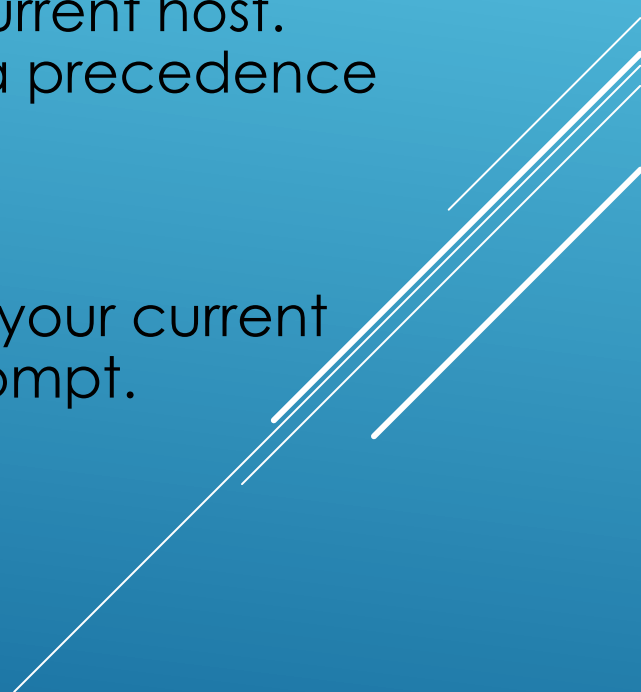
Windows-Linux-MacOS



7.x

.NET Core
pwsh.exe
-UseWindowsPowerShell

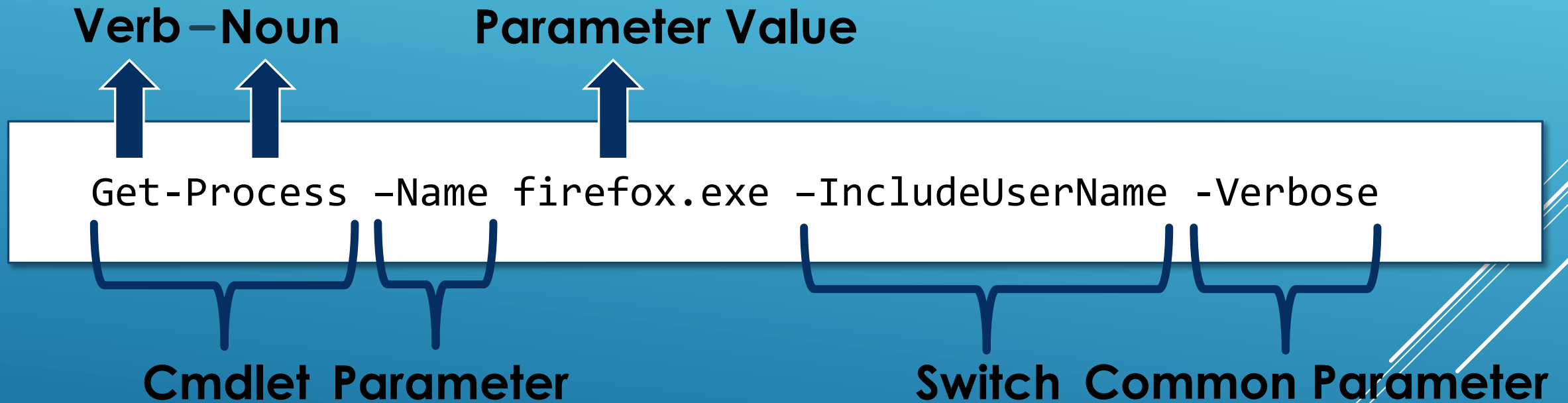
SHOW-PROFILE

- ▶ The PowerShell profile is a script that runs when a PowerShell session is started (unless the `-nopprofile` switch is used)
 - ▶ Basically, it is a logon script for PowerShell containing commands, aliases, variables, drives, functions, modules, etc.
 - ▶ Profiles can be for all users, the current user, all hosts, and the current host. You can have a mix of none, some, or all of these and there is a precedence order.
 - ▶ There is no default profile
 - ▶ Your current user profile is stored in the `$profile` variable. To edit your current user profile with VSCode, type `code $profile` at a PowerShell prompt.
- 
- A series of three parallel white diagonal lines are located in the bottom right corner of the slide, extending from the middle of the right edge towards the bottom left.


SHOW-VARIABLE

- ▶ A unit of **memory** in which a value is stored
- ▶ PowerShell variables are text strings represented by the dollar sign "\$" prefix (ex. \$a, \$my_var, \$var1, etc.)
- ▶ Although special characters and spaces allowed, variable names should be kept simple
- ▶ Types of variables:
 - ▶ User – user defined and deleted on exit (add to your PowerShell Profile to sustain)
 - ▶ Automatic – defined by Posh & not editable (ex. \$PSHOME)
 - ▶ Preference – defaults defined & are user editable
- ▶ Get-Variable to show all variables defined in a session


GET-SYNTAX



CMDLETS (COMMAND-LETS)

- ▶ A type of command in PowerShell
 - ▶ Common syntax & options
 - ▶ Usually take object input & return objects
 - ▶ Run standalone, combined, or stored in `.ps1` file as scripts
- 
- A series of three parallel white diagonal lines in the bottom right corner of the slide.

SHOW-PARAMETER

- ▶ Allow for users to provide input or options
 - ▶ A pre-hyphen ("-") is not always necessary (ie. positional)
 - ▶ Some parameters have default values (dev decision)
 - ▶ Different Types:
 - ▶ Named -> default full name of parameter
 - ▶ Positional -> typed in a relative order (caution)
 - ▶ Dynamic -> only available under special conditions
 - ▶ Common -> built-in parameters
 - ▶ Sets -> expose different parameters & return different information
- 
- A series of three parallel white diagonal lines extending from the bottom right towards the top right of the slide.

SHOW-PIPELINING

Pipeline operator



```
PS>Get-Process -Name firefox.exe -IncludeUserName | Stop-Process
```

Object returned by first cmdlet sent to second cmdlet

“One-liner”

SHOW-FUNCTION


A list of PowerShell statements that run like you had entered them on the command line.

```
function Get-FirefoxProcess { Get-Process firefox.exe }  
function Get-FirefoxProcess {  
    $a = Get-Process firefox.exe  
    if ($a -eq $null) {  
        Write-Host "No Firefox process present"}  
    return $a  
}
```


To run a function, you call it by name on the command line, in a script, or in a module.

```
function Get-FirefoxProcess {  
    $a = Get-Process firefox.exe  
    if ($a -eq $null) {  
        Write-Host "No Firefox process present"}  
    return $a  
}  
Get-FirefoxProcess
```

SHOW-MODULE

- ▶ Modules are a **.psm1** file or a **.dll** that contain commands, providers, variables, functions, help context, aliases, & workflows
 - ▶ Imported as itself (automatically via autoloading or manually via the Import-Module cmdlet) or launched from **.psd1** manifest file (see next slide)
 - ▶ Can be stored anywhere, but common paths is best (ex. \$env:PSModulePath)
 - ▶ Easy lifecycle management with Install-Module & Update-Module
- 
- A series of three parallel white diagonal lines extending from the bottom right towards the top right of the slide.

SHOW-MODULEMANIFEST

- ▶ Module can be defined by a manifest file which has an extension of `.psd1`
 - ▶ A manifest is not required for a module.
 - ▶ Manifests are made up of a hash table of keys and values that describe the contents & attributes of a module, define prerequisites, and determine how components are processed.
 - ▶ Think of it as a glossary and instruction set for a module for PowerShell.
- 
- A series of three parallel white diagonal lines in the bottom right corner of the slide.

CORE CMDLETS TO KNOW



- ▶ Get-Help
- ▶ Update-Help
- ▶ Get-Command
- ▶ Show-Command
- ▶ Get-Member
- ▶ Update-Module

THANK YOU!

@mikenelsonio

Github - mikenelson-io

LinkedIn - nelmedia

PS C:\PowerShell> █

