

# The Secrets of PowerShell

Michael Bender  
Cloud Advocate - Microsoft



How Many People have  
Written Code in PowerShell or  
Another Language?

# The Secret to PowerShell?

When does this all make sense? I mean, it's like learning a language. And I know it comes with time. And it's slowly starting to sink in. But this isn't something where one video makes me an expert. I'm just curious how long it took you to be able to just pull a Matrix where you can look at code and know what's up.

Yesterday, 9:22 AM

I wish it were that easy. It's a constant learning curve. I've spent close to a decade using Power Shell, and the last 5-6 focused on learning it. The key is really finding ways to use it on a regular basis. Fundamentals are key with PowerShell

Yesterday, 6:21 PM ✓

# The Fundamentals

If you don't know the fundamentals, you don't know...

# What is PowerShell?

PowerShell is a **open-source** task-based command-line shell and scripting language built on .NET. PowerShell helps system administrators and power-users rapidly automate tasks that manage operating systems (Linux, macOS, and Windows) and processes.

PowerShell  
is...

- A command line interface
- An automation framework
- An infrastructure builder
- A ticket to the future

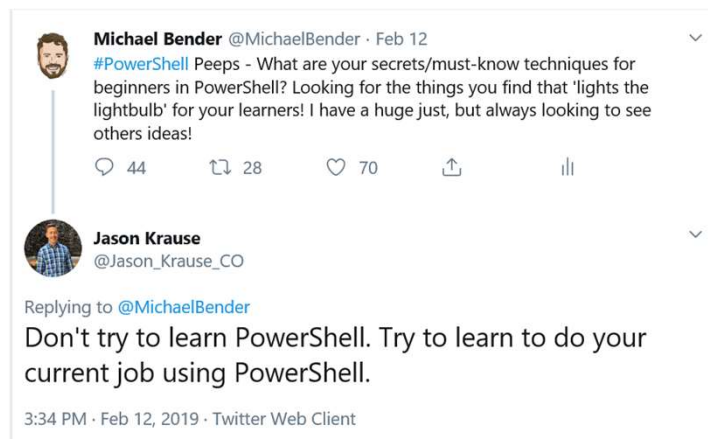
Why you need to learn  
PowerShell

'The future of Windows and system administration is automation and [#PowerShell](#) is the "One Ring" to rule them all.' - Jeff Hicks

"... or get used to saying 'Do you want fries with that?'"  
-Don Jones

Almost EVERYTHING in the  
Microsoft ecosystem supports  
#PowerShell

How to  
begin...



Tweet thread at <http://bit.ly/2C6SBsL>

# Verb-Noun

## Approved Verbs for PowerShell Commands

09/06/2018 • 17 minutes to read • Contributors

PowerShell uses a verb-noun pair for the names of cmdlets and for their derived Microsoft .NET Framework classes. For example, the `Get-Command` cmdlet provided by PowerShell is used to retrieve all the commands that are registered in PowerShell. The verb part of the name identifies the action that the cmdlet performs. The noun part of the name identifies the entity on which the action is performed.

### Note

PowerShell uses the term *verb* to describe a word that implies an action even if that word is not a standard verb in the English language. For example, the term *New* is a valid PowerShell verb name because it implies an action even though it is not a verb in the English language.

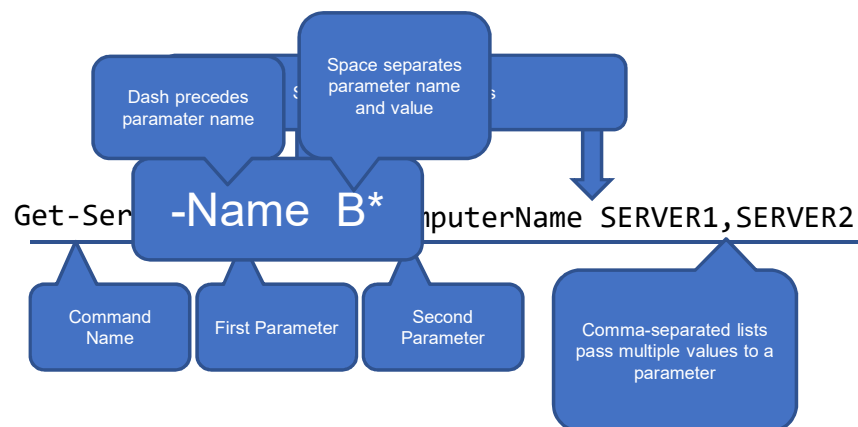
### Verb Naming Rules

The following list provides guidelines to consider when you choose the verb for a cmdlet name:

- When you specify the verb, we recommend that you use one of the predefined verb names provided by PowerShell (aliases for these predefined verbs are included in the following tables). When you use a predefined verb, you ensure consistency between the cmdlets that you create, the cmdlets that are provided by PowerShell, and the cmdlets that are designed by others.
- Use the predefined verbs to describe the general scope of the action, and use parameters to further refine the action of the cmdlet.
- To enforce consistency across cmdlets, do not use a synonym of an approved verb.
- Use only the form of each verb that is listed in this topic. For example, use "Get", but do not use "Getting" or "Gets".
- Use Pascal casing for verbs. In Pascal casing the initial letter of each word is capitalized, such as "ForEach".
- Do not use the following reserved verbs or aliases. These verbs are used by the PowerShell language, or by special case cmdlets provided by PowerShell.

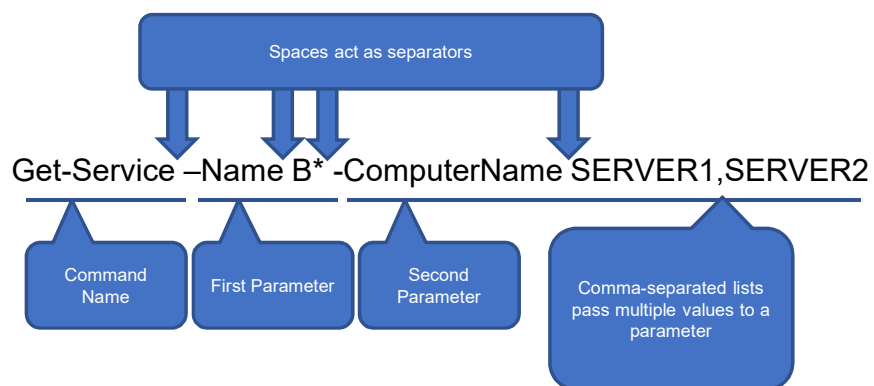
<https://aka.ms/psverbs>

## Anatomy of a Command



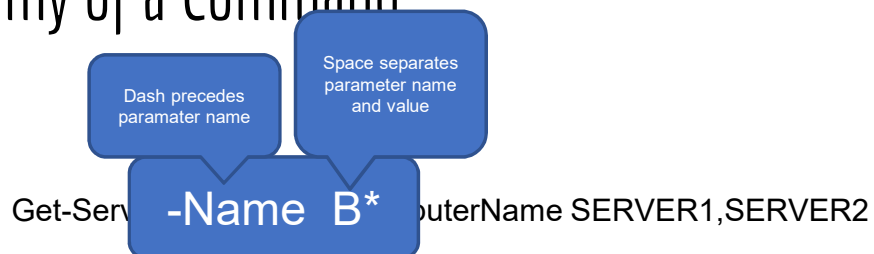
&gt;

## Anatomy of a Command



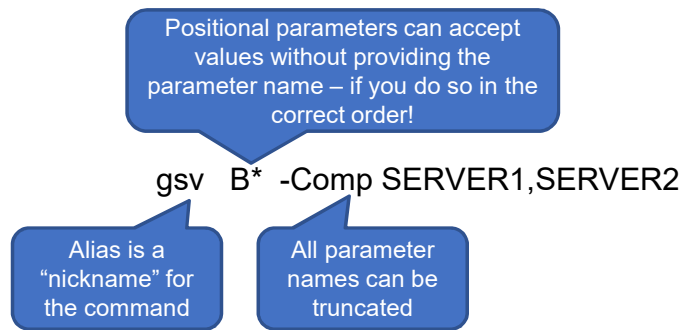
&gt;

## Anatomy of a Command





## Anatomy of a Command



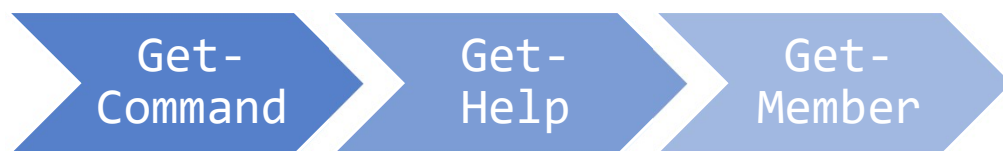
## 3 Keys to PowerShell

Get-Command

Get-Help

Get-Member

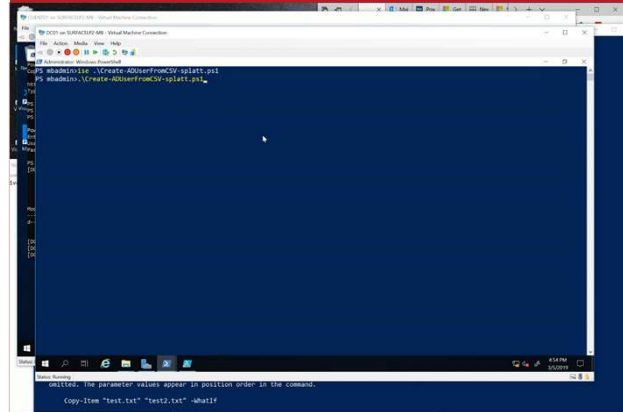
# The 3 Keys of PowerShell



Finding Info in the  
Shell Demo

## Power of the Shell

How many users can you create in 2 minutes?



## Tale of Two Shells

## Windows PowerShell

- Based on .NET Standard
- Built into Windows
- Windows only
- Feature complete
- Full set of Windows PowerShell commands

## PowerShell Core

- Based on .NET Core
- Download from GitHub
- Runs on Windows, Linux, and Mac OS
- Open source development
- Subset of Windows PowerShell commands

## Dueling Shells Demo

## Objects

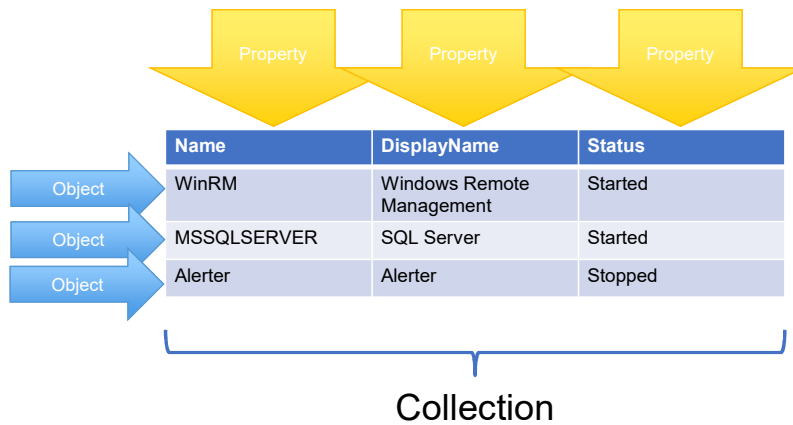
# Objects in PowerShell



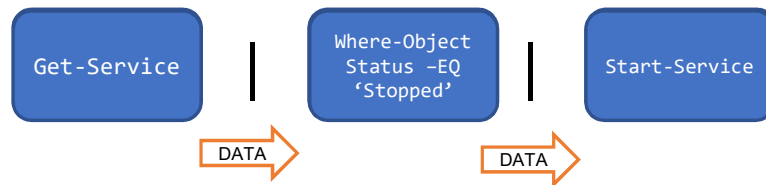
- Not text-based
- Contain Properties and Methods
- Examples
  - ◆ Color
  - ◆ Doors
  - ◆ Horespower

## “Objects”

### Get-Service



## Pipelining in PowerShell



## Variables

→ It's Where you store stuff!!!

## Running with Privileges

- ExecutionPolicy
- Caching Credentials

## Fun with Remoting

- PSSession (1 to 1)
- Invoke-Command (1 to Many)
- Implicit Remoting



## Finding your way in AD

## Splatting

VS Code

GIT

→ Centralized version control for your

- ◆ Code
- ◆ Documentation

## Cloud Advocacy at Microsoft



## Cloud Advocacy at Microsoft

- I work in Engineering so I'm not interested in **selling you anything.**
- I AM interested in:
  - ◆ Your Feedback
  - ◆ Helping you Learn

Share Everything



'You don't have to  
share your life; just  
share what you know.' –  
Don Jones

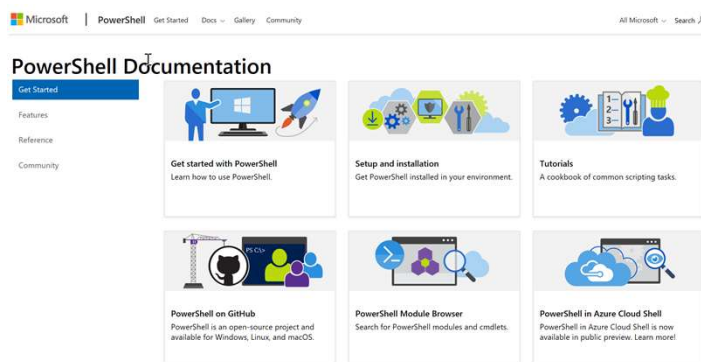
## Call to Action

- Remember the Fundamentals
- Think of 5 tasks at your work that you can automate with PowerShell, and Automate 1 one in the next week
- Learn Git and put your code in a repo to share
- Share What you know...Always!

## Resources

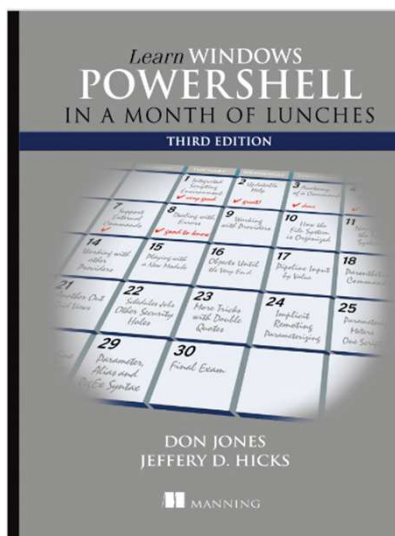
- Docs.Microsoft.com – <https://aka.ms/pslearn>
- PowerShell.org
- Pluralsight.com – Learning Path
- Download Visual Studio Code – <https://aka.ms/vscode-mb>
- Github repo for demo : <https://aka.ms/ps-secrets>

# Read the Docs!!!



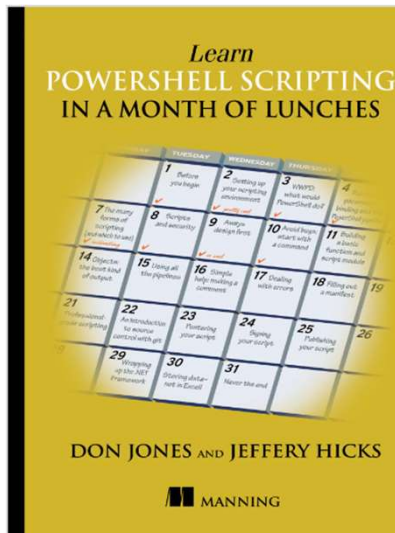
<https://aka.ms/pslearn>

# Read and Do This!



<https://www.manning.com/books/learn-windows-powershell-in-a-month-of-lunches-third-edition>

Then This...



<https://www.manning.com/books/learn-powershell-scripting-in-a-month-of-lunches>

## Questions

- Itsallgeek2mike.com
- ltopstalk.com
- Twitter - @MichaelBender
- michael.bender@Microsoft.com



Thank You!