Writing Award Winning PowerShell

Functions and Script Modules

mikefrobbins.com

Get-Help –Name 'about_Presenter'

- Mike F Robbins

- Microsoft MVP

- Mississippi PowerShell User Group

- PowerShell 101: The No-Nonsense Beginner's Guide

- Windows PowerShell TFM 4th Edition

- PowerShell Deep Dives

- Advanced Category in the 2013 Scripting Games

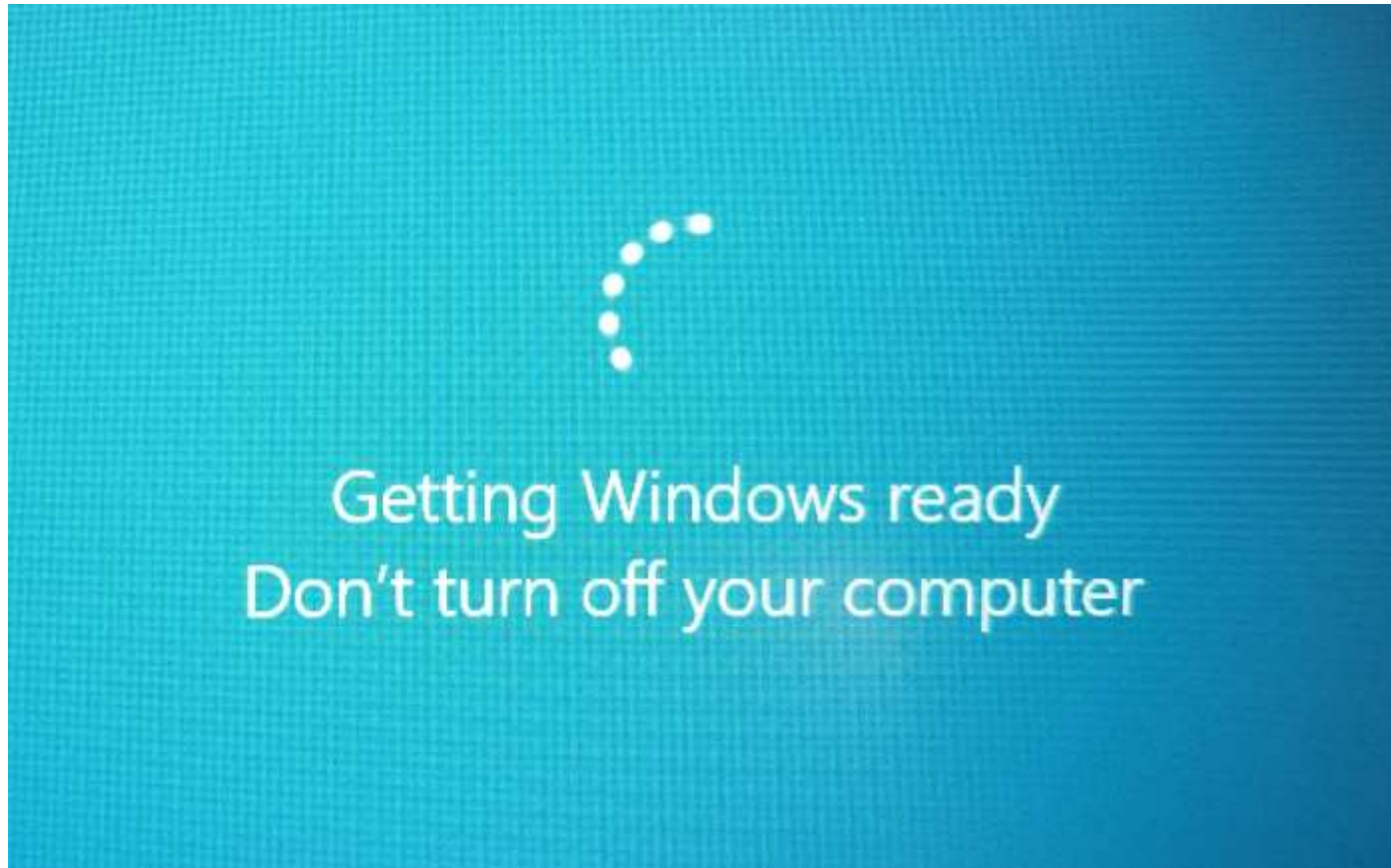- Learn more about me @ mikefrobbins.com

Write-Output 'Questions for Audience'

- IT Pro's

- Developers

- Unit Testing

- Source Control

- VSCode (Visual Studio Code)

- ISE (Integrated Scripting Environment)

- PowerShell Core

- PowerShell Experts

Get-HotFix

Getting Windows ready
Don't turn off your computer

Quick & Dirty Code 6 Months Later

Set-Content 'Information to Cover'

- Functions and Script Modules
  - Parameters
  - Parameter Validation
  - CmdletBinding
  - SupportsShouldProcess
  - Pipeline Input
  - Error Handling
  - Module Manifest
  - Custom Formatting
  - And More ☺

Start-Process 'Demo'

Demo Code: github.com/mikefrobbins/Presentations

Introduction

- Think

- Plan

- Learning –ne Searching the Internet

- Stealing (I mean borrowing) Code

- What Problem are you trying to solve?

- Don't Reinvent the Wheel

- Don't Overcomplicate Things

Thought Process

- Aliases and Positional Parameters

- Readability

- Unnecessary code

- Attention to Detail

- Clarity

- Manageable Steps

- Simplicity

Create a Script or Function?

- Tool Oriented

- Autoloading

- NuGet Repository

- Proprietary Data

- Opensource

- Ideas and Suggestions

- Contributions

- Better Code

What is a Function?

- Wrong Conference ☺

- List of Commands or Instructions
  - Perform a Specific Task
  - Packaged as one unit

- PowerShell Function
  - Retrieve Data
  - Process Data
  - Output Data

- Modular

Function Naming

- Pascal Case Name
  - Capitalize First Letter in Verb
  - All Terms in Noun

- Approved Verb
  - Import Module Warning

- Singular Noun

- Prefix Noun
  - Naming Collisions

- Example
  - ApprovedVerb-PrefixSingularNoun

Resume-Process 'Demo'

Demo Code: github.com/mikefrobbins/Presentations

Variables

- Static Values
  - Variables
  - Parameters

- Hungarian Notation
  - ~~$strOutFile~~
  - $OutFile

- Naming

- Variable Reuse

Don't Reuse Variable Names

June 15, 2014 at 2:46 pm   #16242

Before that line, $VM is a string. Afterward, it's one of the objects returned from Get-VM.

Generally speaking, it's confusing to reuse a variable in that fashion, and I try to avoid it. I would prefer something like "$vmObject = $VMs | Where-Object Name -eq $VM", where it's more clear that $VM and $vmObject are different thingies. (A very technical term, I know.)

Dave Wyatt
Moderator

- Source: https://powershell.org/forums/topic/redundancy-of-code-in-advanced-function/

Parameter Naming

- **Built-In Cmdlets**
    - Azure Cmdlets

- **Plural Parameter Names**
    - Avoid Them
    - Single Element
    - Multi Element

- **Pascal Case**

Resume-Process 'Demo'

Demo Code: github.com/mikefrobbins/Presentations

Where to put braces?

rkeithhill commented on May 29, 2015   + 😀

I agree with @sunmorgus, I'd go for option 1.5 - Stroustrup style (no cuddled else):

```
function Get-Noun {
    end {
        if($Wide) {
            Get-Command | Sort-Object Noun -Unique | Format-Wide Noun
        }
        else {
            Get-Command | Sort-Object Noun -Unique | Select-Object -Expand Noun
        }
    }
}
```

I don't like cuddled else and catch. It's easier IMO to line-select a whole else clause and delete it. With K&R style, you can easily wind up deleting the closing } on the if statement.

👍 1

- Source: https://github.com/PoshCode/PowerShellPracticeAndStyle/issues/24#issuecomment-106860973

Great Code 6 Months Later

Select-Object –Property 'Resources'

- [PowerShell Documentation](#)

- [PowerShell Best Practices and Style Guide](#)

- [Microsoft Virtual Academy](#)

- [PowerShell.org](#)

- [GitHub](#)

- [User Groups & Blogs](#)

- [Twitter](#)

- [Slack](#)

Get-Contact –Identity 'Presenter'

- Blog: mikefrobbins.com (or MrPowerShell.com)

- Twitter: @mikefrobbins

- LinkedIn: www.linkedin.com/in/mikefrobbins

- E-Mail: See mikefrobbins.com/about/

- User Group: mspsug.com (or MsPowerShell.com)