PowerShell Scripting Fundamentals

Jeff Hicks
AUTHOR ~ TEACHER ~ SENSEI

@jeffhicks https://jdhitsolutions.com/blog

By default PowerShell will not run scripts

- .ps1, .psm1, .ps1xml files

Default application for ps1 files is Notepad

You must specify full path to script

Execution Policy

- Restricted
- RemoteSigned
- AllSigned
- Unrestricted
- ByPass

```
PS C:\> Set-ExecutionPolicy RemoteSigned
```

Execution Policy

Run in an elevated session

Only needs to be set once per machine

Can also be set via Group Policy

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

Execution Policy

PowerShell will tell you if you can’t run a script

```
PS C:\> Get-ExecutionPolicy

RemoteSigned
```

Execution Policy

Keep your execution policy simple but the most secure

```
PS C:\Scripts> .\MondayTasks.ps1
```

Use Full Path Name

Prevents command high-jacking

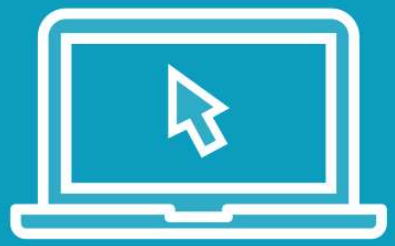
Extension not required

- But recommended

Use tab expansion

PowerShell scripting security is focused on the unintentional execution of code

Demo



Scripting security

PowerShell executes in a given *scope*

When the scope ends everything defined ends

Scripts run in their own scope

Scope

Global

Script

Private

Numbered Scopes

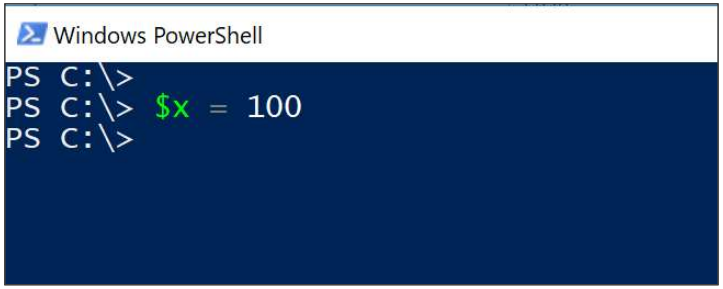
You reference *something*

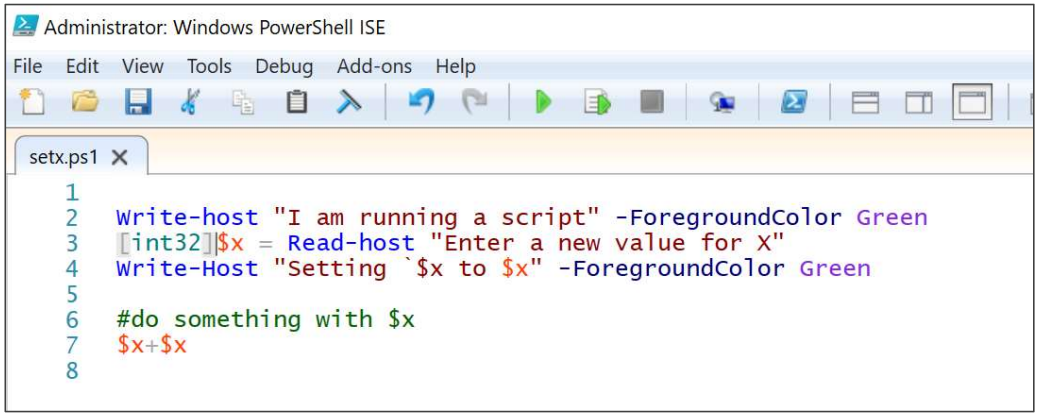
PowerShell looks in the current scope

If it finds it – Success!

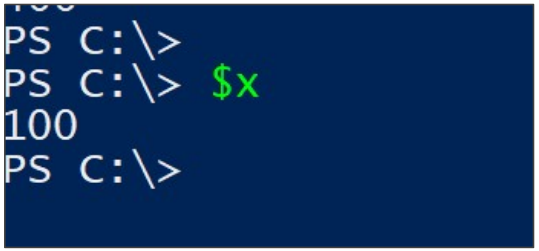
- Else it looks to the parent
  - And then the grandparent
    - Until it finds it
    - You may or may not get an error

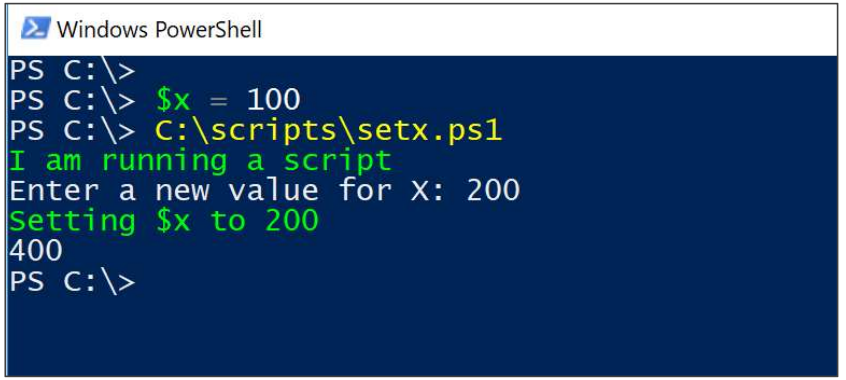
But -- PowerShell can only write or create in the current scope*

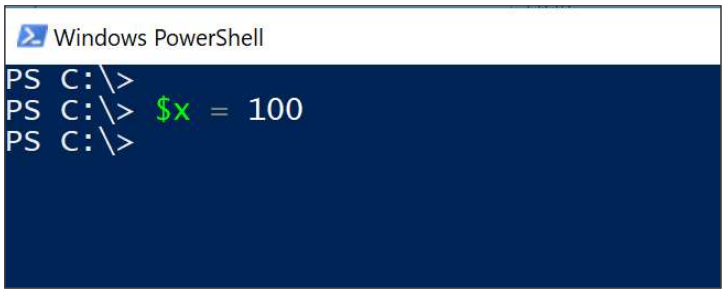


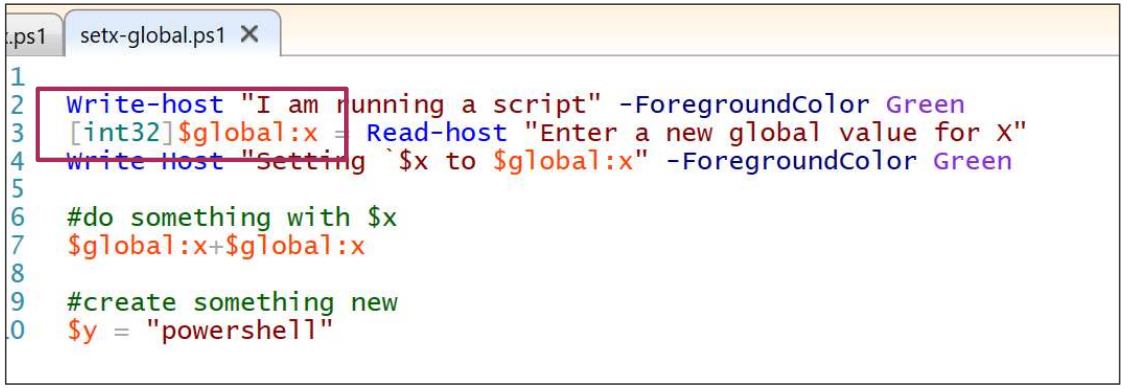


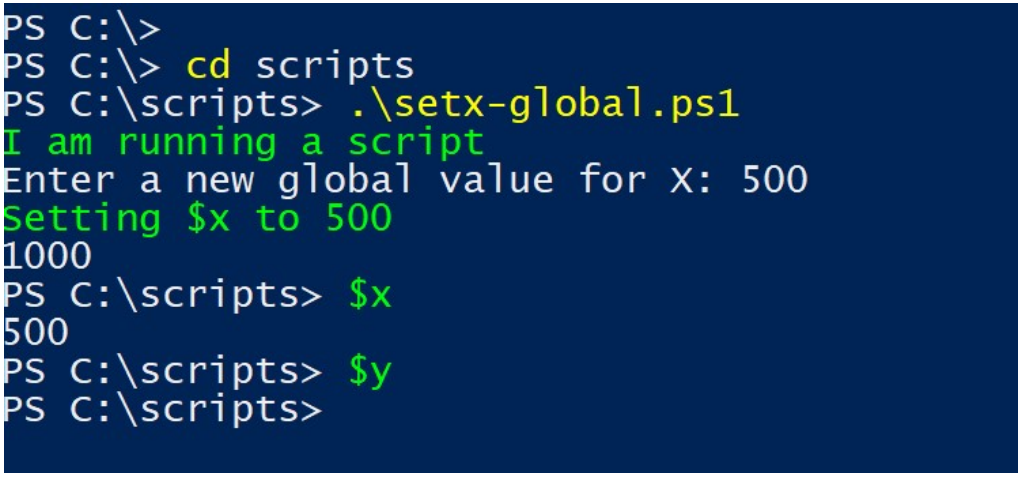
First Rule of Scope:
Avoid referencing out of scope items






Windows PowerShell
PS C:\>
PS C:\> $x = 100
PS C:\>


.ps1
setx-global.ps1
1
2 Write-host "I am running a script" -ForegroundColor Green
3 [int32]$global:x = Read-host "Enter a new global value for X"
4 Write-Host "Setting `$x to $global:x" -ForegroundColor Green
5
6 #do something with $x
7 $global:x+$global:x
8
9 #create something new
10 $y = "powershell"


PS C:\>
PS C:\> cd scripts
PS C:\scripts> .\setx-global.ps1
I am running a script
Enter a new global value for X: 500
Setting $x to 500
1000
PS C:\scripts> $x
500
PS C:\scripts> $y
PS C:\scripts>

Scope Guidelines

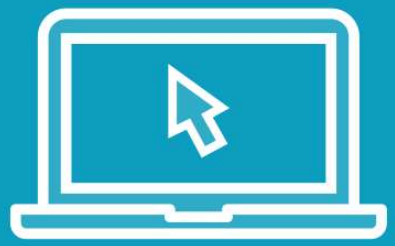
Referencing global items like PSDrives typically OK

Some cmdlets have a –Scope parameter

Using scope specifiers is the exception

Everything in the ISE runs in the global scope

Demo



Understanding scope

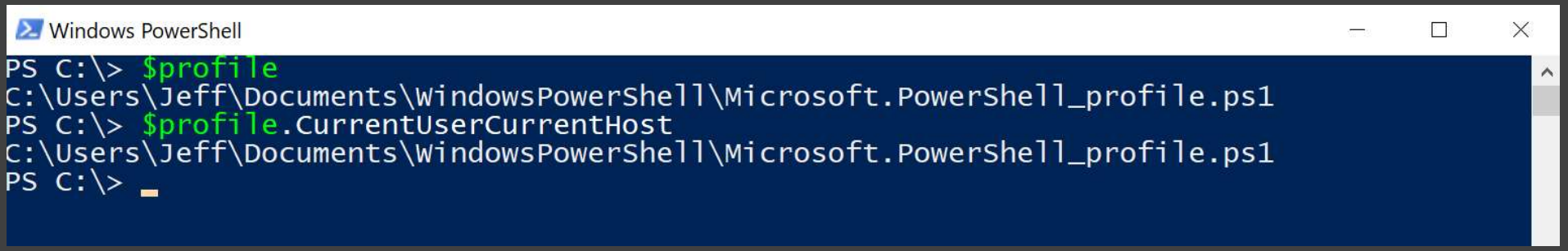
Configure your PowerShell session to meet your needs

Create a profile script

- Hard coded file paths
- You will have to create them

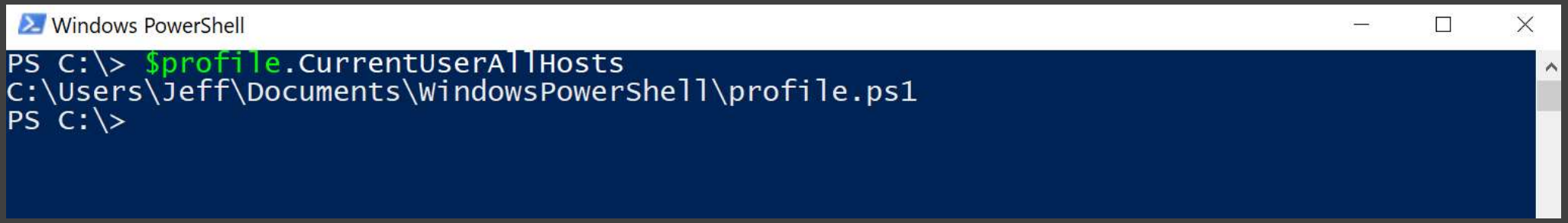
PowerShell tools and shells may have additional profiles

Profile scripts follow script execution rules



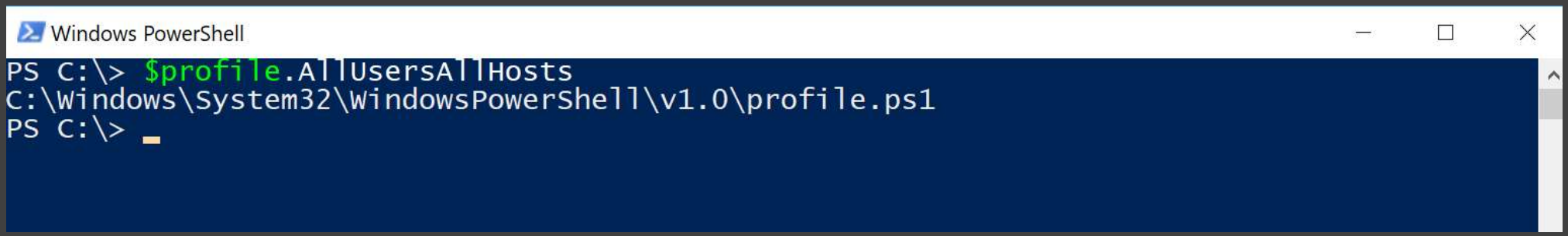
Current User Current Host

WindowsPowerShell folder does not exist by default



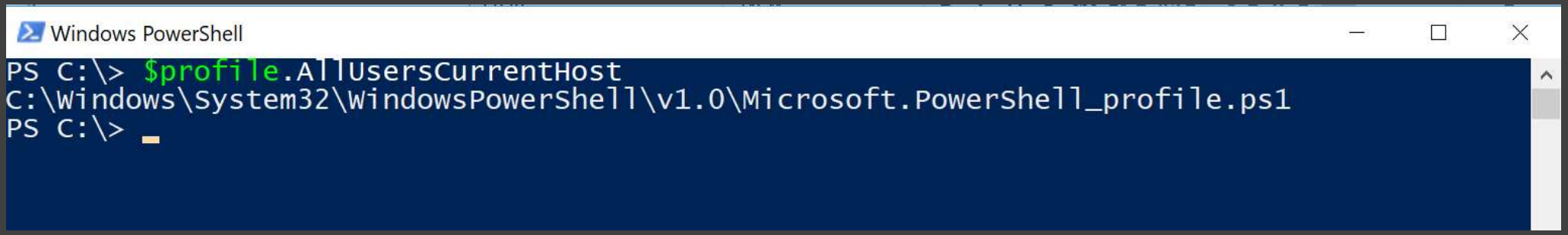
Current User All Hosts

Applies to console, ISE, VSCode and others



All Users All Hosts

Requires local admin rights to create or modify



All Users Current Host

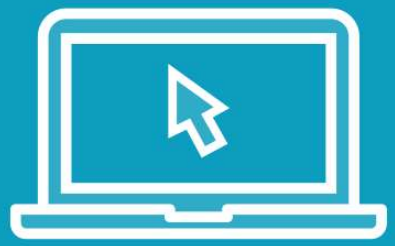
Requires local admin rights to create or modify

Other PowerShell Platforms

```
PS C:\> $profile
C:\Users\Jeff\Documents\WindowsPowerShell\Microsoft.PowerShellISE_profile.ps1

PS C:\> $profile.AllUsersCurrentHost
C:\Windows\System32\WindowsPowerShell\v1.0\Microsoft.PowerShellISE_profile.ps1
```

Demo



PowerShell profile scripts

Summary

- Enable the most secure execution policy
- You need to understand scope
- Use PowerShell profile scripts to configure your work environment
- Take time to read the about topics