PSDAY.UK

2018

Keynote

Richard Siddaway

DevOpsGroup System Frontier eSynergySolutions TECHSNIPS

Who am I?

Wrote first program in 1970

Working with PowerShell since v1 public beta 1

Started original UK user group in 2007

≥ World's first PowerShell UG

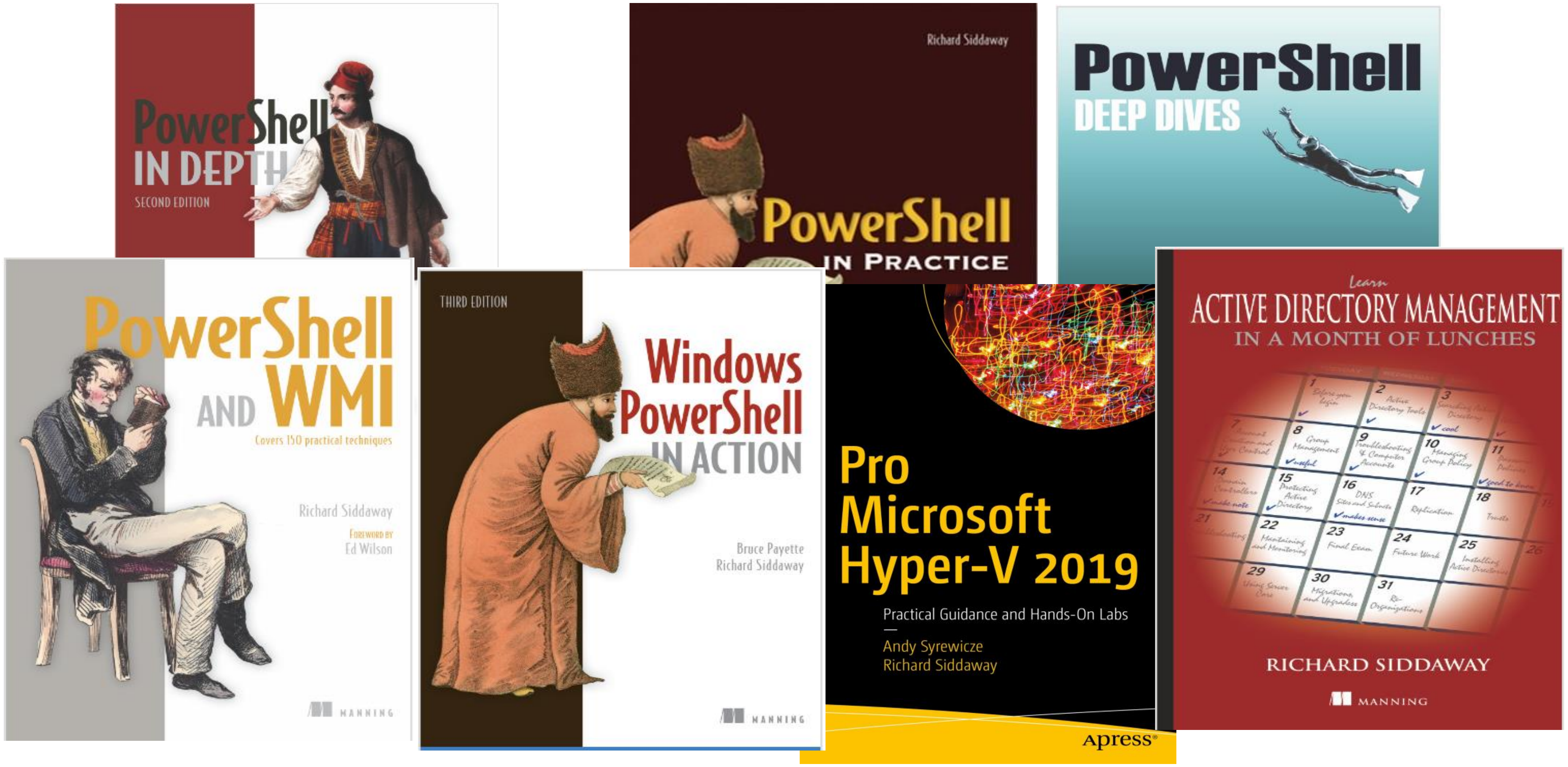
Established and grew PowerShell Summit

Blogger - https://itknowledgeexchange.techtarget.com/powershell/

Speaker

Author

My books



2018

The wonderful thing about keynotes ….

The wonderful thing about keynotes is that keynotes are wonderful things.

With apologies to A. A. Milne and Disney

Keynote session

Invited speaker

≥ Usually regarded as an expert in the topic

Establish main underlying theme

Establish the framework

Flag up any larger ideas

Summarise core message

Keynote session

Invited speaker

usually regarded as an expert in the topic

Establish main underlying theme

Establish the framework

Flag up any larger ideas

Summarise core message

Maybe add some controversy?

PSDAY.UK

2018

PSDAY.UK

2018

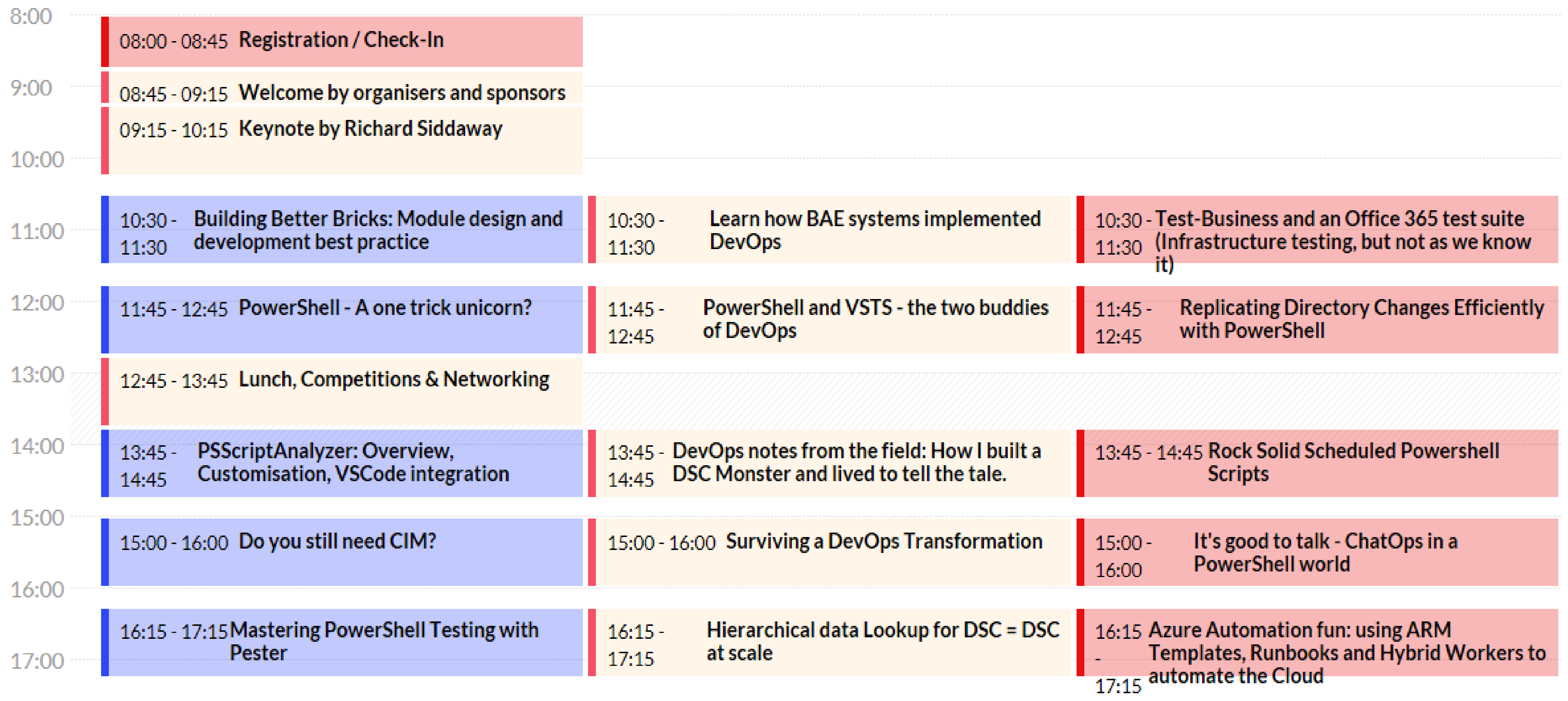
PowerShell doesn’t matter…

PSDAY.UK

2018

It’s what you do with that matters

Today – all about doing things



8:00 9:00 10:00 11:00 12:00 13:00 14:00 15:00 16:00 17:00

08:00 - 08:45 Registration / Check-In		
08:45 - 09:15 Welcome by organisers and sponsors		
09:15 - 10:15 Keynote by Richard Siddaway		
10:30 - 11:30 Building Better Bricks: Module design and development best practice	10:30 - 11:30 Learn how BAE systems implemented DevOps	10:30 - 11:30 Test-Business and an Office 365 test suite (Infrastructure testing, but not as we know it)
11:45 - 12:45 PowerShell - A one trick unicorn?	11:45 - 12:45 PowerShell and VSTS - the two buddies of DevOps	11:45 - 12:45 Replicating Directory Changes Efficiently with PowerShell
12:45 - 13:45 Lunch, Competitions & Networking		
13:45 - 14:45 PSScriptAnalyzer: Overview, Customisation, VSCode integration	13:45 - 14:45 DevOps notes from the field: How I built a DSC Monster and lived to tell the tale.	13:45 - 14:45 Rock Solid Scheduled Powershell Scripts
15:00 - 16:00 Do you still need CIM?	15:00 - 16:00 Surviving a DevOps Transformation	15:00 - 16:00 It's good to talk - ChatOps in a PowerShell world
16:15 - 17:15 Mastering PowerShell Testing with Pester	16:15 - 17:15 Hierarchical data Lookup for DSC = DSC at scale	16:15 - 17:15 Azure Automation fun: using ARM Templates, Runbooks and Hybrid Workers to automate the Cloud

2018

Organisers

Daniel Krebs

Jonathan Medd

Martynas Valkunas

Ryan Yates

Iain Brighton

Rob Sewell

Ebru Cucen

2018

PSDAY.UK

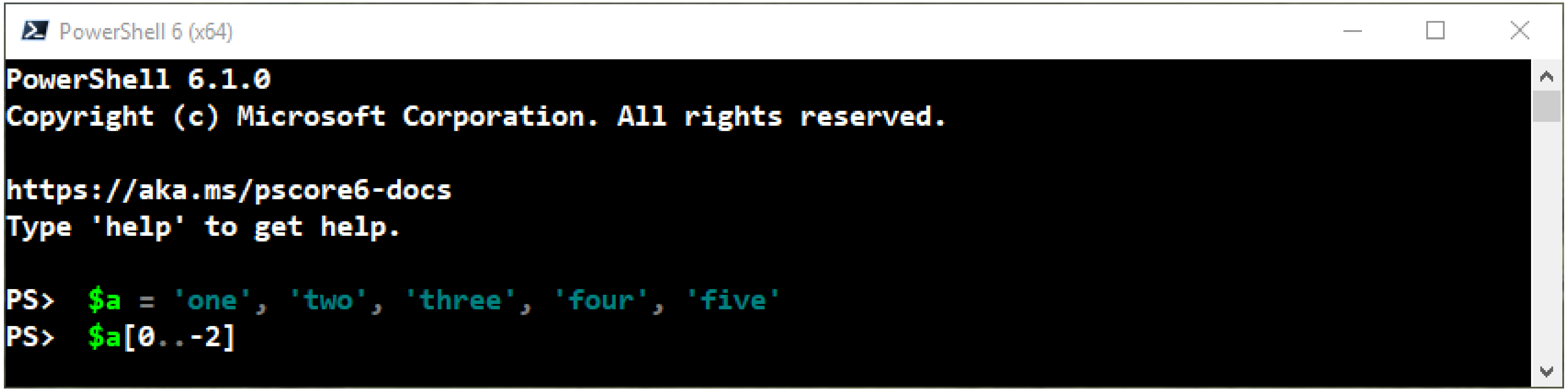
2018

This is your day

Ask questions

Talk to people

Coding aside: What’s the result?


PowerShell 6 (x64)

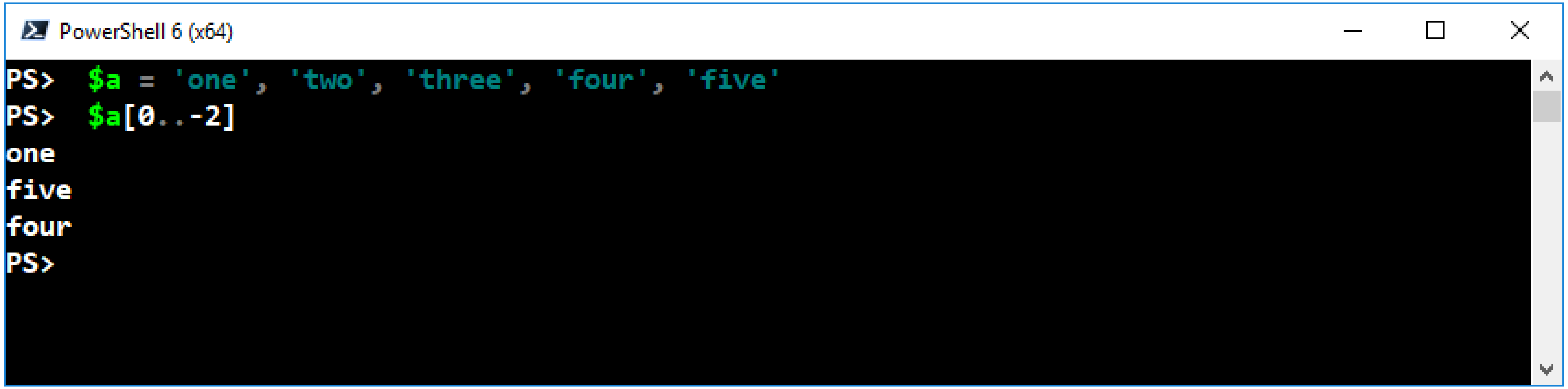
```
PowerShell 6.1.0
Copyright (c) Microsoft Corporation. All rights reserved.

https://aka.ms/pscore6-docs
Type 'help' to get help.

PS>  $a = 'one', 'two', 'three', 'four', 'five'
PS>  $a[0..-2]
```

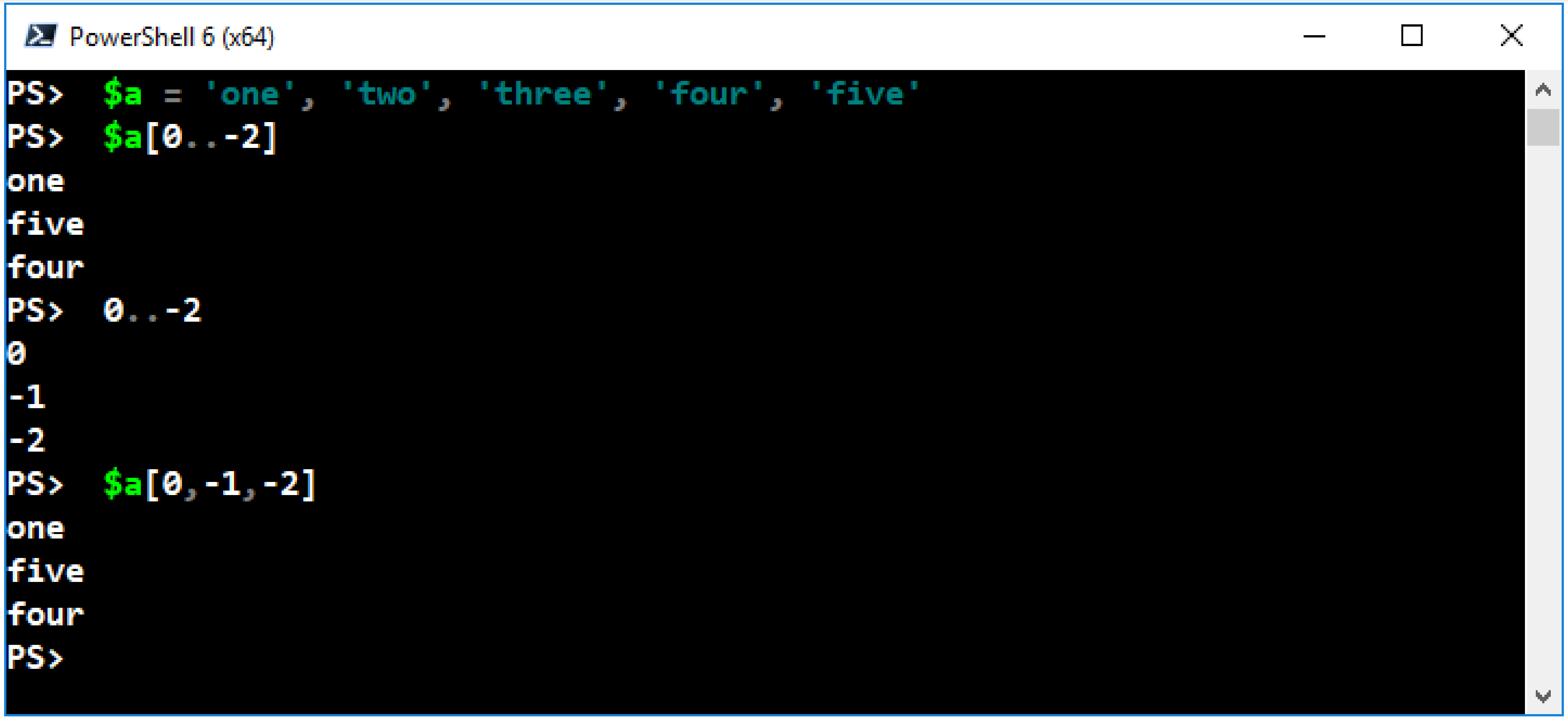
2018

Coding aside: What's happening?



```
PS>  $a = 'one', 'two', 'three', 'four', 'five'
PS>  $a[0..-2]
one
five
four
PS>
```

Coding aside: Range and indexing



PowerShell 6 (x64)

```
PS>  $a = 'one', 'two', 'three', 'four', 'five'
PS>  $a[0..-2]
one
five
four
PS>  0..-2
0
-1
-2
PS>  $a[0,-1,-2]
one
five
four
PS>
```

PSDAY.UK

2018

PowerShell: Where are we?

PSDAY.UK
2018

Which version of PowerShell did you start with?

The past: Monad Manifesto and more

PowerShell v1	PowerShell and objects

PowerShell v2	PowerShell remoting (WSMAN);
	PowerShell Jobs;
	ISE; WMI cmdlets; Modules

PowerShell v3	Workflows CIM cmdlets; CIM sessions; CDXML Cmdlet coverage growth
PowerShell v4	DSC
PowerShell v5.x	DSC improvements PowerShell classes
PowerShell v6	Open source; Linux & mac; SSH remoting
PowerShell v6.1	Markdown cmdlets; Threadjobs; Measure-Object –ALLstats; return of [wmi] & [adsi] accelerators
PowerShell vNext	?????????????

2018

Things that never took off

Transactions

≥ Registry provider only

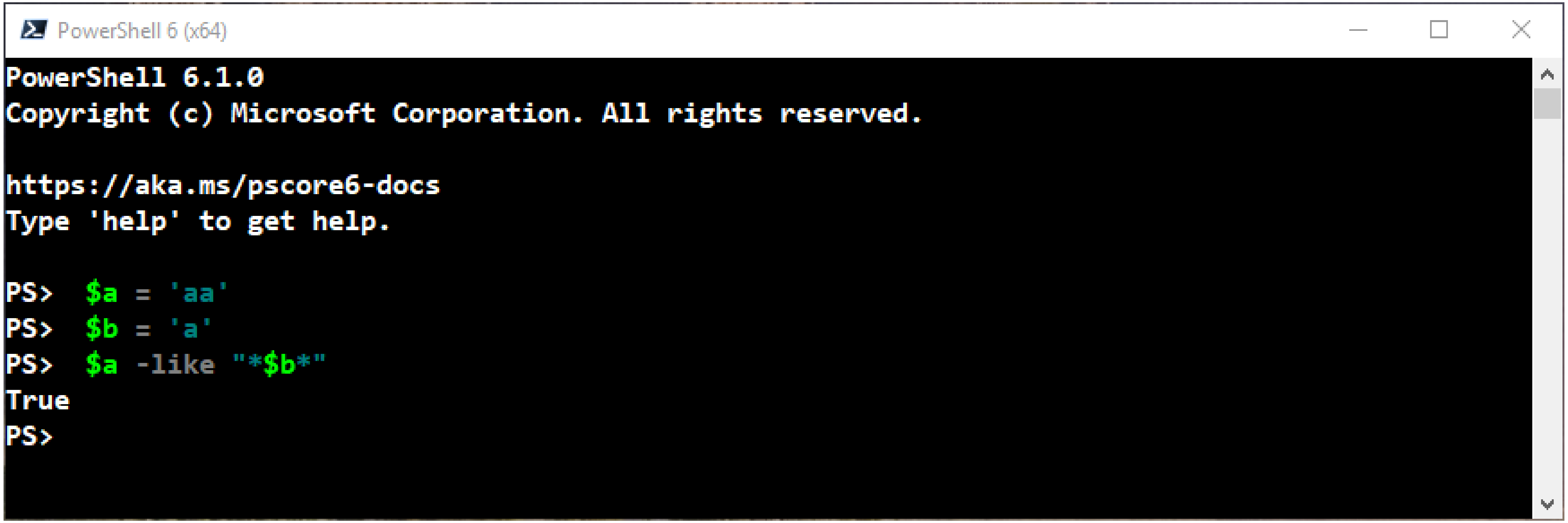
Workflows

≥ Same but different

≥ Too confusing

2018

Coding aside: Anybody surprised that this works?



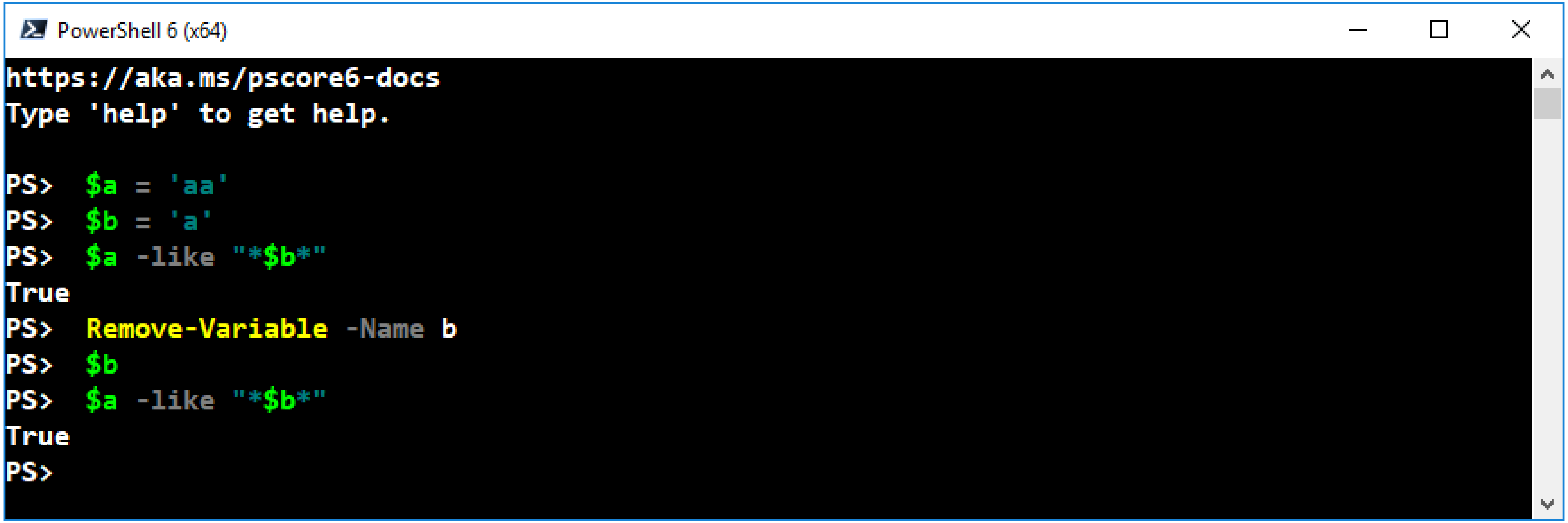
```
PowerShell 6.1.0
Copyright (c) Microsoft Corporation. All rights reserved.

https://aka.ms/pscore6-docs
Type 'help' to get help.

PS>  $a = 'aa'
PS>  $b = 'a'
PS>  $a -like "*$b*"
True
PS>
```

2018

Coding aside: What about this?



```
https://aka.ms/pscore6-docs
Type 'help' to get help.

PS>  $a = 'aa'
PS>  $b = 'a'
PS>  $a -like "*$b*"
True
PS>  Remove-Variable -Name b
PS>  $b
PS>  $a -like "*$b*"
True
PS>
```

PSDAY.UK

2018

PowerShell: Where are we going?

What’s needed?

Good parallelisation option

- Workflows didn’t work
- PoshRSJobs from the gallery?
- Threadjobs in v6.1?

Cmdlet coverage on Linux

Domain Specific languages ?

- E.g. Pester

Easier OpenSSH setup and configuration

YOUR opportunity to input to PowerShell

Innovation

New features in PowerShell

New ways of using existing functionality

- E.G use Pester for troubleshooting
- Think about SSH remoting for non-domain scenarios

New uses for PowerShell

Where does innovation come from?

PSDAY.UK

2018

On which OS
do you use PowerShell?

PowerShell: The future

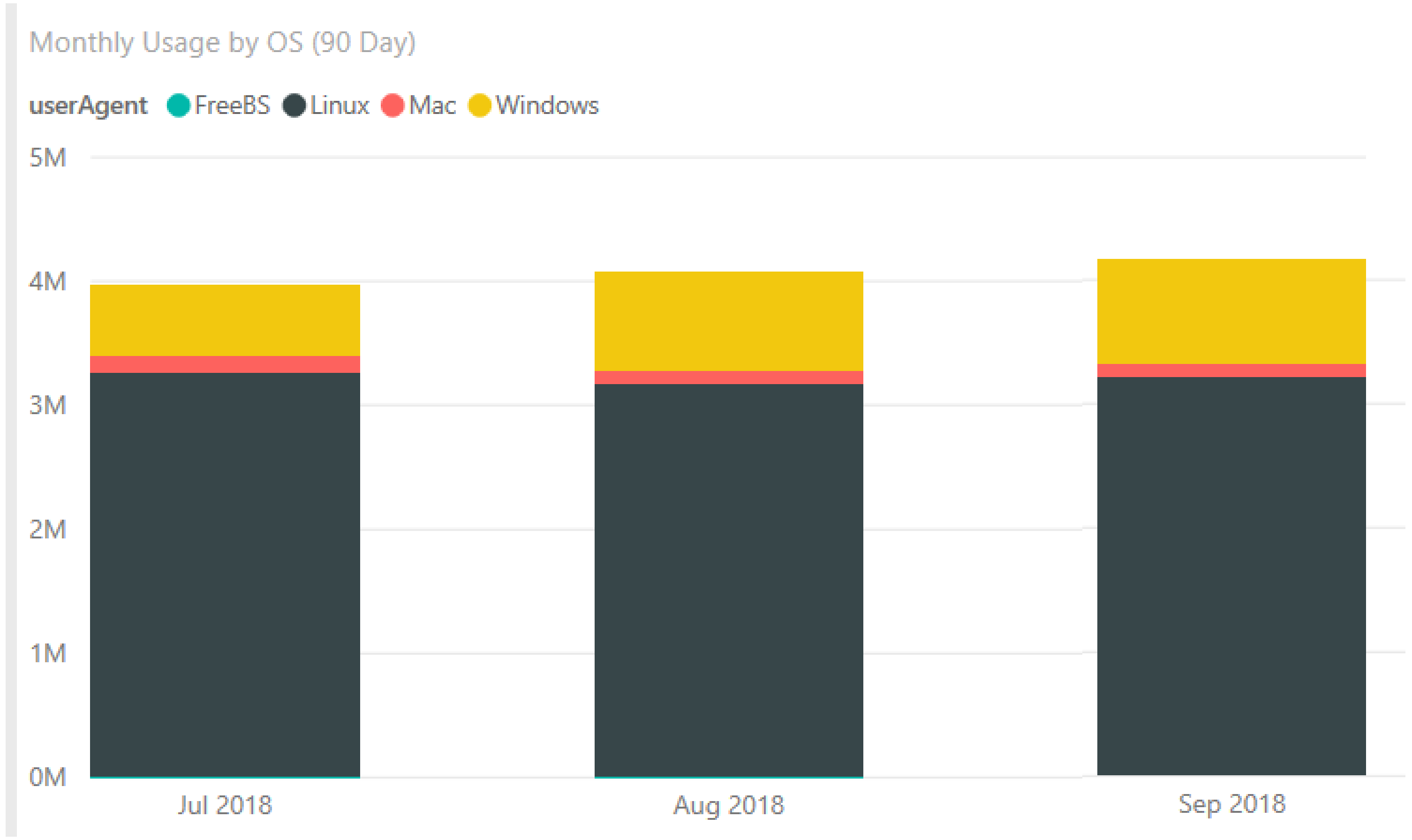
… the intent is to never have a v7 otherwise we'll end up fragmenting the PowerShell community further between 5.x and 6.

Steve Lee. PowerShell Team Microsoft

https://github.com/PowerShell/PowerShell/issues/3302#issuecomment-423693097

2018

PowerShell v6 usage



2018

https://msit.powerbi.com/view?r=eyJrIjoiYTYyN2U3ODgtMjBlMi00MGM1LWI0ZjctMmQ3MzE2ZDNkMzIyIiwidCI6IjcyZjk4OGJmLTg2ZjEtNDFhZi05MWFiLTJkN2NkMDExZGI0NyIsImMiOjV9&pageName=ReportSection5

What does this mean?

From the PowerShell v6.1 release notes:

PowerShell Core sends basic telemetry data to Microsoft when it is launched. The data includes the OS name, OS version, and PowerShell version. This data allows us to better understand the environments where PowerShell is used and enables us to **prioritize** new features and fixes.

My emphasis

PSDAY.UK

2018

You need to be involved with the PowerShell project to get the PowerShell you want

PSDAY.UK

2018

Question:

What’s the state of the PowerShell community?

PowerShell community strengths

Openness

Numbers

Information suppliers

User groups

Conferences

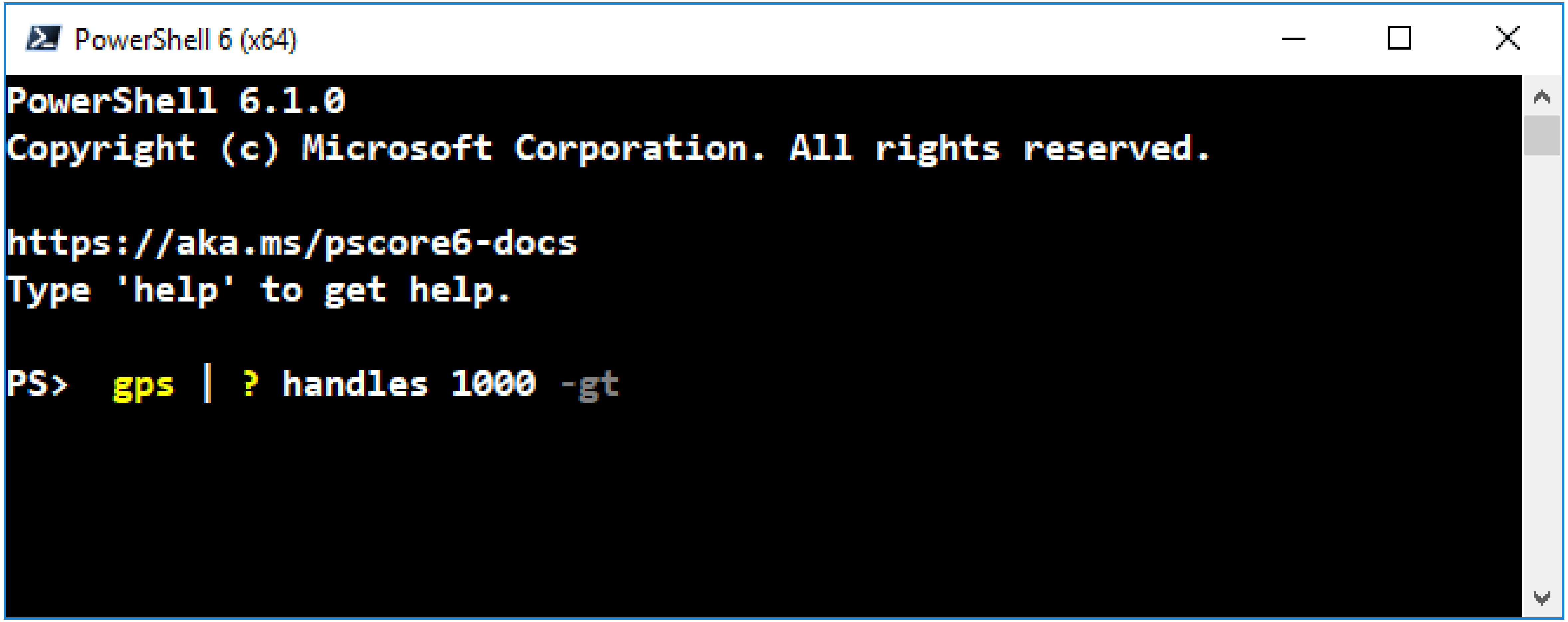
DIVERSITY

PSDAY.UK

2018

I think the community is currently in a very strong position.

Coding aside: What’s this doing



PowerShell 6 (x64)

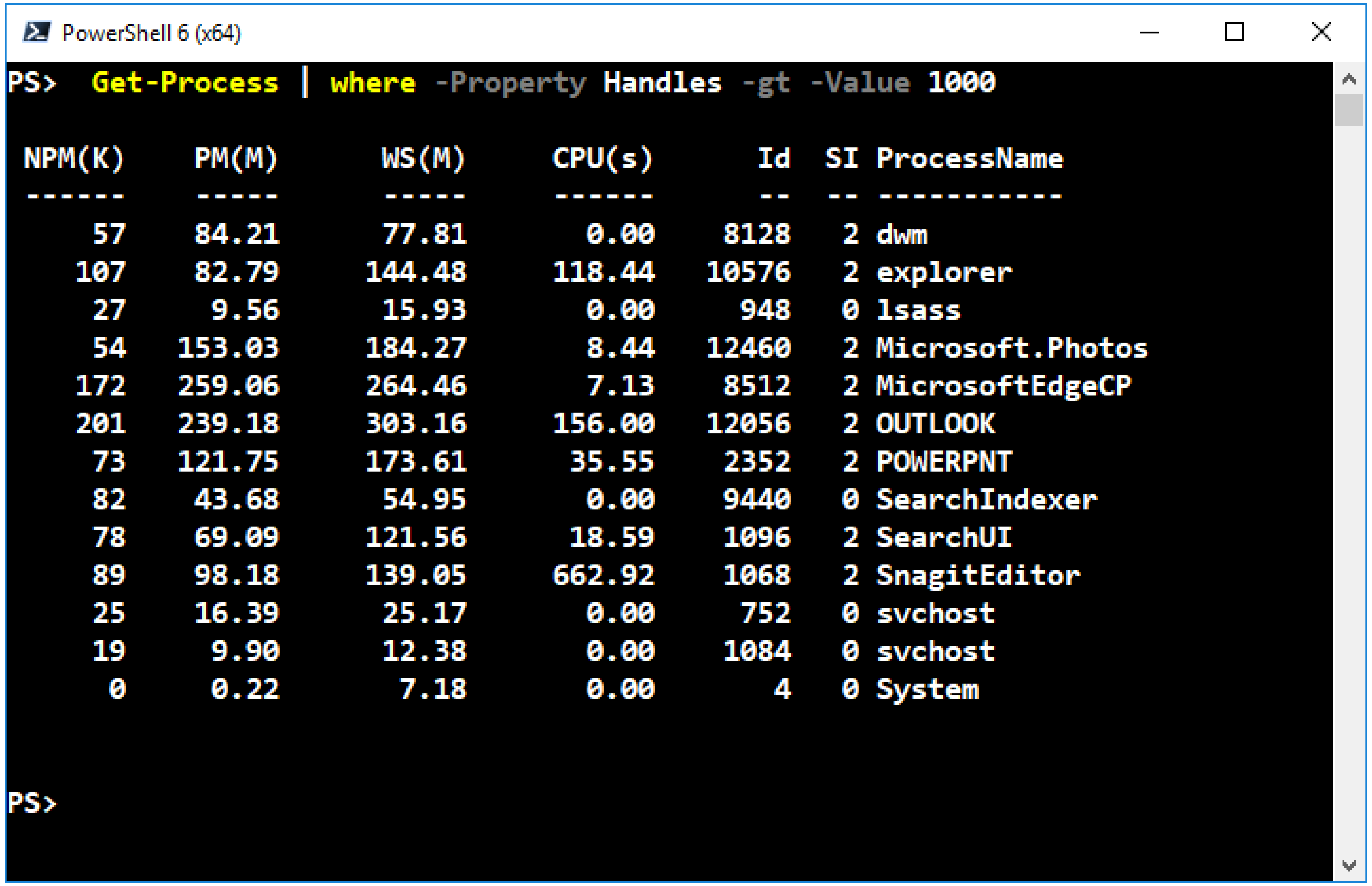
```
PowerShell 6.1.0
Copyright (c) Microsoft Corporation. All rights reserved.

https://aka.ms/pscore6-docs
Type 'help' to get help.

PS>  gps | ? handles 1000 -gt
```

2018

Coding aside: Positional parameters are fun



PowerShell 6 (x64)

```
PS>  Get-Process | where -Property Handles -gt -Value 1000

 NPM(K)    PM(M)      WS(M)     CPU(s)      Id  SI ProcessName
 ------    -----      -----     ------      --  -- -----------
     57    84.21      77.81       0.00    8128   2 dwm
    107    82.79     144.48     118.44   10576   2 explorer
     27     9.56      15.93       0.00     948   0 lsass
     54   153.03     184.27       8.44   12460   2 Microsoft.Photos
    172   259.06     264.46       7.13    8512   2 MicrosoftEdgeCP
    201   239.18     303.16     156.00   12056   2 OUTLOOK
     73   121.75     173.61      35.55    2352   2 POWERPNT
     82    43.68      54.95       0.00    9440   0 SearchIndexer
     78    69.09     121.56      18.59    1096   2 SearchUI
     89    98.18     139.05     662.92    1068   2 SnagitEditor
     25    16.39      25.17       0.00     752   0 svchost
     19     9.90      12.38       0.00    1084   0 svchost
      0     0.22       7.18       0.00       4   0 System


PS>
```

PSDAY.UK
2018

Best practice.
Fact or fiction?

PowerShell practice

Relative easy definitions:

- ≥ Bad practice
- ≥ Good practice

What’s “best” practice

Pareto principle

Who decides “best” practice

2018

Coding practices

Bad

Cmdlet and parameter aliases

Positional parameters

Good

Output object

≥ Single type

“Best” practice conflicts?

https://github.com/PoshCode/PowerShellPracticeAndStyle/blob/master/Style-Guide/Code-Layout-and-Formatting.md

Open braces on the same line

PSScriptAnalyzer

Place open braces either on the same line as the preceding expression or on a new line.

Anyone vote for always on new line?

Good or bad?

```
Get-CimInstance -ClassName Win32_OperatingSystem |
foreach {
  Write-Host -Object "Operating System : $($_.Caption)"
  Write-Host -Object "Version          : $($_.Version)"
  Write-Host -Object "Boot Time        : $($_.LastBootUpTime)"
  Write-Host -Object "Date             : $($_.LocalDateTime)"
  Write-Host -Object "Install Date     : $($_.InstallDate)"
}
```

Good or bad?

```
Get-CimInstance -ClassName Win32_OperatingSystem |
foreach {
  Write-Host -Object "Operating System : $($_.Caption)"
  Write-Host -Object "Version          : $($_.Version)"
  Write-Host -Object "Boot Time        : $($_.LastBootUpTime)"
  Write-Host -Object "Date             : $($_.LocalDateTime)"
  Write-Host -Object "Install Date     : $($_.InstallDate)"
}
```

```
Get-CimInstance -ClassName Win32_OperatingSystem |
select @{N='Operating System'; E={$_.Caption}},Version,
@{N='Boot Time'; E={$_.LastBootUpTime}},
@{N='Date'; E={$_.LocalDateTime}},
@{N='Install Date'; E={$_.InstallDate}}
```

Script analyser doesn’t like select or trailing white space on line 5

2018

Another one

```
$a = New-Object -TypeName PSobject

$a |
Add-Member -MemberType NoteProperty -Name P1 -Value 1 -PassThru |
Add-Member -MemberType NoteProperty -Name P2 -Value 2 -PassThru |
Add-Member -MemberType NoteProperty -Name P3 -Value 3 -PassThru
```

Another one

```
$a = New-Object -TypeName PSobject

$a |
Add-Member -MemberType NoteProperty -Name P1 -Value 1 -PassThru |
Add-Member -MemberType NoteProperty -Name P2 -Value 2 -PassThru |
Add-Member -MemberType NoteProperty -Name P3 -Value 3 -PassThru
```

```
New-Object -TypeName PSobject -Property @{
  P1 = 1
  P2 = 2
  P3 = 3
}
```

```
## may use ordered hash table instead
$props = @{
  P1 = 1
  P2 = 2
  P3 = 3
}

New-Object -TypeName PSobject -Property $props
```

```
[PSCustomObject]@{
  P1 = 1
  P2 = 2
  P3 = 3
}
```

2018

My view of good code

Does the job

≥ Include error handling etc as required

Runs in a satisfactory time frame

≥ Beware of looking for unnecessary performance increases

Easy to read, understand and maintain

PSDAY.UK

2018

Perfect is the mortal enemy of good enough

PSDAY.UK
2018

Perfect is the mortal enemy of good enough

Developers want everything perfect.
Admins just want to go home

The future of IT?

DevOps

≥ The principles

≥ Cultural change

Post DevOPs

≥ Anti-DevOPs

≥ DevOps the business

CloudOps

NoOps

≥ Automation?

?

2018

Call to action

Get involved with the community

≥ Teach someone to use PowerShell

≥ Blog

≥ Speak

≥ Organise

≥ Contribute code

≥ Answer forum questions

Get involved with the PowerShell project

≥ Use PowerShell v6

≥ Comment

≥ Discuss

≥ Submit issues

PSDAY.UK

2018

PSDAY.UK
2018

Thank you

DevOpsGroup System Frontier eSynergySolutions TECHSNIPS

PSDAY.UK

2018