

A Full Day of PowerShell

Chrissy Lemaire, Rob Sewell and Jaap Brasser

@cl

@sqldbawithbeard

@jaap_brasser



MS Cloud Summit Paris

Agile.Net – aOS – AZUG FR – CMD – GUSS

A Full Day of PowerShell

PowerShell, Azure & SQL Server



MS Cloud Summit Paris
Agile.Net – aOS – AZUG FR – CMD – GUSS

About_Chriissy



 @c1

SQL Server since 1999 & PowerShell since 2005

System Center Cloud and Datacenter Management MVP since
2015

DBA with General Dynamics IT at NATO Spec Ops HQ in Belgium

@cl on Twitter

Blogs at netnerds.net

Also RealCajunRecipes.com

Lead for the Belgian PowerShell User Group

Co-Lead for SQL PASS PowerShell Virtual Chapter



MS Cloud Summit Paris
Agile.Net – aOS – AZUG FR – CMD – GUSS

About_Rob

Name : Rob

Occupation : DBA, Automator

Website : sqldbawithAbeard.com

Twitter : @sqldbawithbeard

Interests : PowerShell, Automation And SQL (PaaS geddit?)

Interesting Fact : Has a Beard. (Still) Plays Cricket

Speaker : SQL Saturdays, SQL Relay, PowerShell Conference EU

Community : SQL South West , SQL Saturday Exeter , PowerShell

Virtual Chapter, Organiser for PowerShell Conference EU 2017

Officer, Lead for dbareports, contributor to dbatools



@sqldbawithbeard



MS Cloud Summit Paris
Agile.Net – aOS – AZUG FR – CMD – GUSS

About_Jaap

- Dutch PowerShell User Group
- Blogging
 - PowerShell Magazine
 - JaapBrasser.com
- Slack
- Reddit
- GitHub
- PowerShell Gallery
- TechNet Forums/Gallery



@Jaap_Brasser



MS Cloud Summit Paris
Agile.Net – aOS – AZUG FR – CMD – GUSS

Agenda for Today

- PowerShell: The beginning
- Defining variables
- Basic script structure
- Loops, operators and flow control
- Advanced PowerShell functionality
- PowerShell tips from the field





POWERSHELL: THE BEGINNING

What is PowerShell and how did it start

- Development started in 2001
- First released in 2006
- Created by a team led by Jeffrey Snover
- Was initially codenamed Monad
- It is written: **P**ower**S**hell

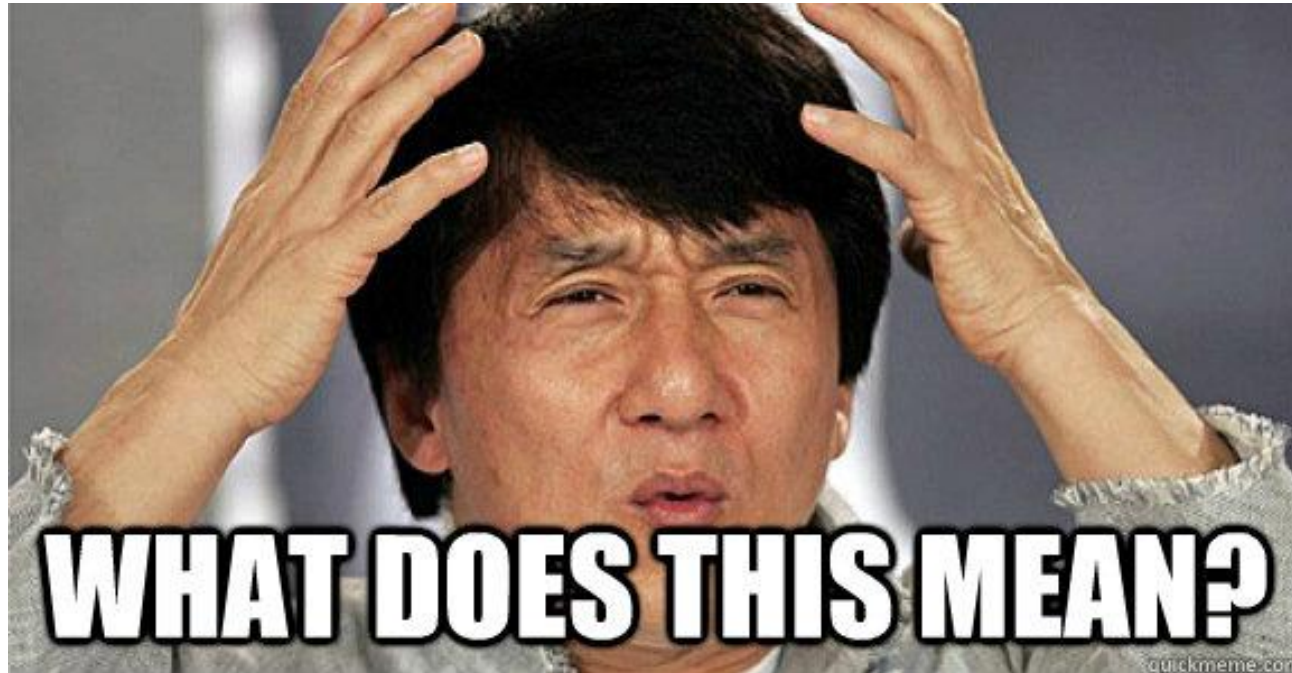


PowerShell naming convention

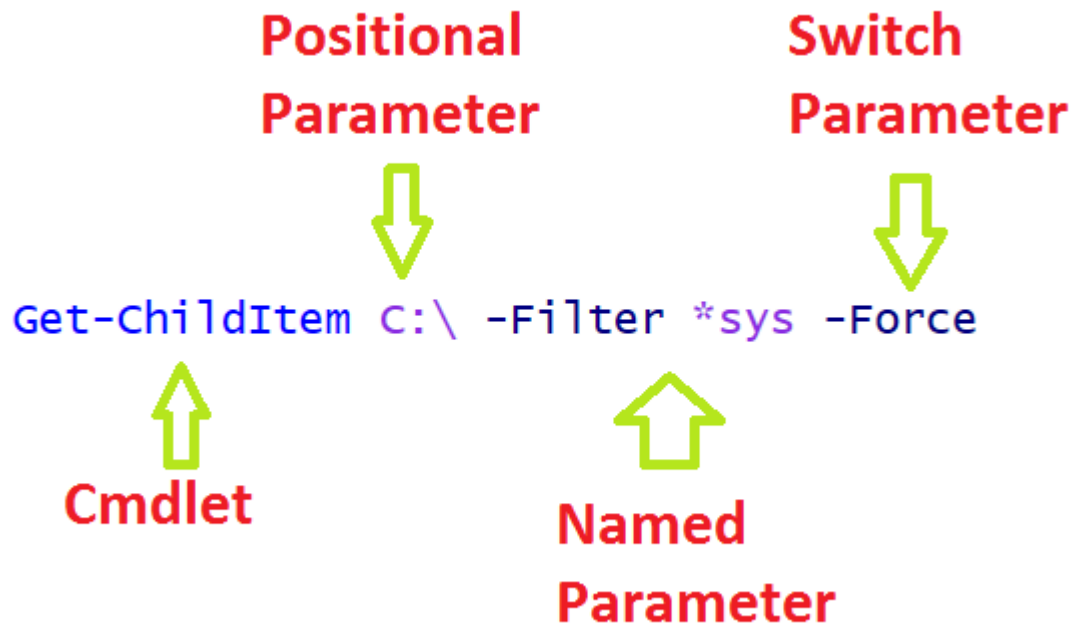
- Verb-SingularNoun
- Verb has to be 'approved'
- Noun should be descriptive, can contain a prefix



How to read the PowerShell language



The PowerShell language



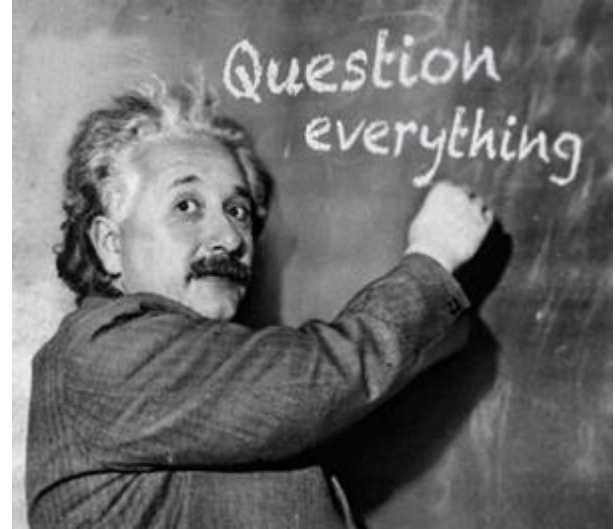
Demo – Get-Verb / Get-ChildItem

- Get-Verb lists all approved verbs
- If you omit a parameter name, PowerShell will use 'positional parameter'



Question

- What happens when you use a non-approved verb?



More PowerShell terms

- Cmdlets
- Functions
- Aliases
- External tools



PowerShell pipeline

- Pipe using the pipe symbol: **|**
- Output from one command can be passed on to the next
- Transfers information using first out, first in



Demo – PowerShell help & pipeline

- How to find help
- Utilizing the PowerShell pipeline





DEFINING VARIABLES

Define a variable in PowerShell

- Using the '\$Variable = ...' notation
- Using the -Variable cmdlets
- Using Tee-Object



Object types in PowerShell

- Each value displayed in the console or stored in a variable has an object type
- Get-Member can be used to reveal the object
- Examples of object types:
 - String, Integer, Boolean, Array
 - File, Folder, Registry entry, ACL



Objects in PowerShell

- Each object has set of properties & methods
- These can be discovered by using Get-Member
- It is possible for a property to have multiple sub properties
- Methods are executed by using () notation



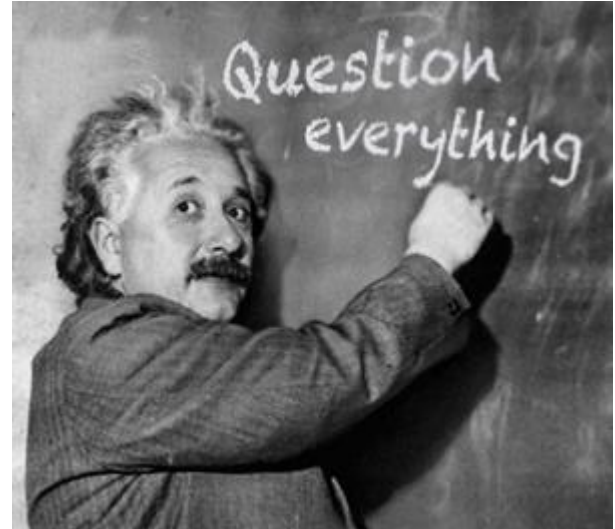
Demo – Defining variables

- Use different methods of creating variables
- Access methods and properties on objects
- Simple calculations in the PowerShell console



Question

- What is another method of creating a variable in PowerShell?



The string is the thing

- Expanding strings

`" "`

`@" "@`

- Literal strings

`' '`

`@' '@`



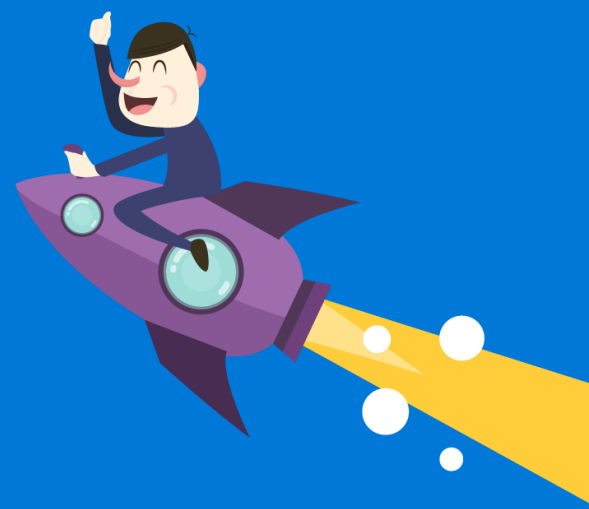
Arrays and hash tables



Demo – Strings, arrays & hashtables

- Use literal strings unless something needs to be expanded
- Arrays can contain any type of object
- Hash tables are used to define custom objects





BASIC SCRIPT STRUCTURE

Best practices

- Do not use aliases
- Write useful comments
- Do not use one-liners outside of console
- Write 'simple' code if possible
- Write multiple small functions instead of a do-it-all script



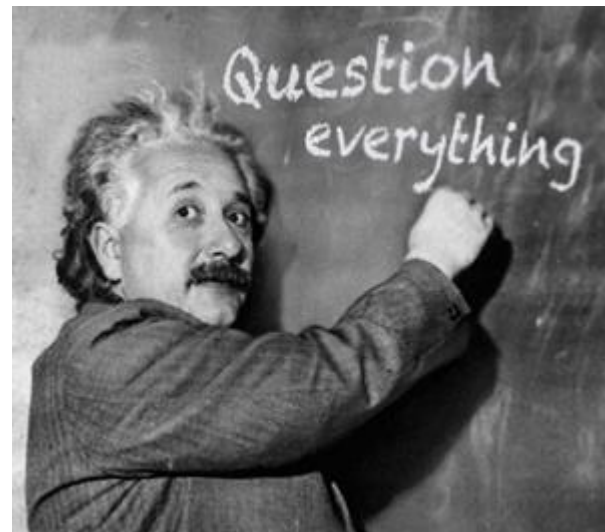
Demo – From bad to Better

- Created a function
- Wrote a comment
- Did not use any aliases
- Used the pipeline to filter and select the desired results



Question

- What could be better?



Best practices

- Make sure that your code only runs until you have the information you need
- Use Error handling where appropriate
- Use comment-based help



Demo – A different approach

- Created a function
- Wrote comment-based help
- Used the error handling to stop processing after result is found





OPERATORS, LOOPS & FLOW CONTROL

Comparison operators

- -eq
- -ne
- -gt
- -ge
- -lt
- -le
- -Like
- -NotLike
- -Match
- -NotMatch
- -Contains
- -NotContains
- -In
- -NotIn
- -Replace



Loops

- Foreach () {}
- For...
- | ForEach-Object
- Do {} while ()



Flow control

- If – elseif – else
- Switch



Demo – Strings, arrays & hashtables

- Use literal strings unless something needs to be expanded
- Arrays can contain any type of object
- Hash tables are used to define custom objects





POWERSHELL: ADVANCED FUNCTIONALITY

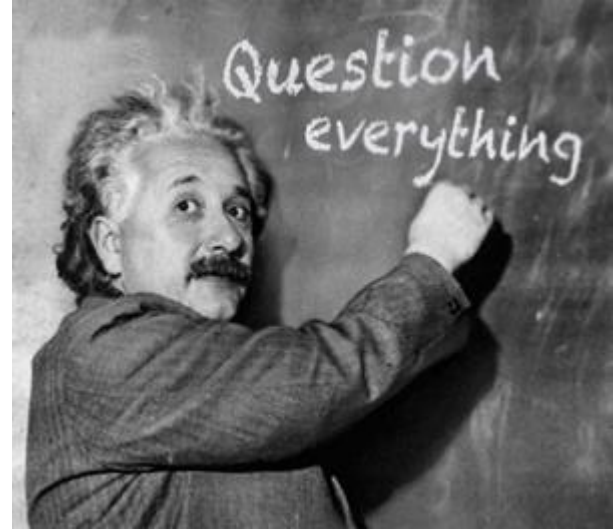
Demo – My first advanced function

- Writing a function in PowerShell
- Create a parameter block
- Use comments
- Create comment-based help
- Set mandatory parameters
- Enable pipeline support



Question

- What is the advantage of writing a function over writing a question?



Demo – My first advanced function

- Using SupportsShouldProcess
 - -WhatIf
 - -Verbose
 - -Confirm
- ValidateSet to verify a set of possible arguments



POWERSHELL TIPS FROM THE FIELD



What would you like to see?

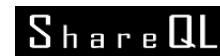
- Fast folder size
- [adsi] and [adsisearcher]
- Powershell –command
- Efficient loops in PS
- Convert word doc to pdf
- Check compression of zip
- Check open folders
- Change a drive letter
- Copy untitled tabs clip
- Create simple GUI
- List non-admin shares
- Rename a disk
- Create OpenFileDialog
- Query MSDN from PS
- Test local credentials
- List AD attributes in use
- 'Special' variables



Merci beaucoup à nos sponsors!
Thank you to all our sponsors!

Join the conversation

#MSCloudSummit
@MSCloudSummit





MS Cloud Summit Paris

Agile.Net – aOS – AZUG FR – CMD - GUSS

Merci Beaucoup! Thank you!

Join the conversation

#MSCloudSummit
@MSCloudSummit