

INTRODUCTION TO POWERSHELL

THE BASICS

MARK VAN DE WAARSENBURG



VOORSTELLEN: MARK VAN DE WAARSENBURG

- WERKZAAM IN IT SINDS 1998
- TECHNISCH CLOUD CONSULTANT
- WERK SINDS 2010 MET POWERSHELL
- INTRODUCTION TO POWERSHELL AT



<https://www.linkedin.com/in/mark-van-de-waarsenburg-8b201414/>

<https://www.linkedin.com/company/d2c-it/>



<https://d2c-it.nl/blog/>



<https://github.com/D2CIT>



<https://d2c-it.nl>



AGE

01. F

05.

08. C

```
function get-continue {
    <#
        .Synopsis
            get-continue
        .DESCRIPTION
            get-continue
        .EXAMPLE
            get-continue -message "Would you like to continue (yes /no)"
    #>

    [cmdletbinding()]

    param(
        [Parameter(Mandatory=$false)]
        $message = "Would you like to continue (yes /no)" ,
        [switch]$destroy
    )

    Begin {
        if($Destroy){
            Write-host "#####" -for Red -BackgroundColor black
            Write-host " LET OP : Je gaat een destroy uitvoeren!" -for Red -BackgroundColor black
            Write-host "#####" -for Red -BackgroundColor black
        } #End If

        $ReadHost = read-host -Prompt $message
    }

    Process{
        Switch ($ReadHost) {
            Destroy {
                Write-verbose "Destroy, Continue";
                $Continue = $true
            } Yes {
                if($Destroy){
                    Write-warning "Yes is not the correct input"
                }else{
                    Write-verbose "Yes, Continue"; $Continue = $true
                }
            } No {
                Write-verbose "No, stop";
                $continue = $false
            } Default {
                Write-verbose "Default, stop"; $Continue=$false
            }
        }#end Switch
    }
    End {
        return $continue
    }
} #EndFunction
```

```
✖ Select Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> Install-WindowsFeature DHCP
```

WAT IS POWERSHELL?

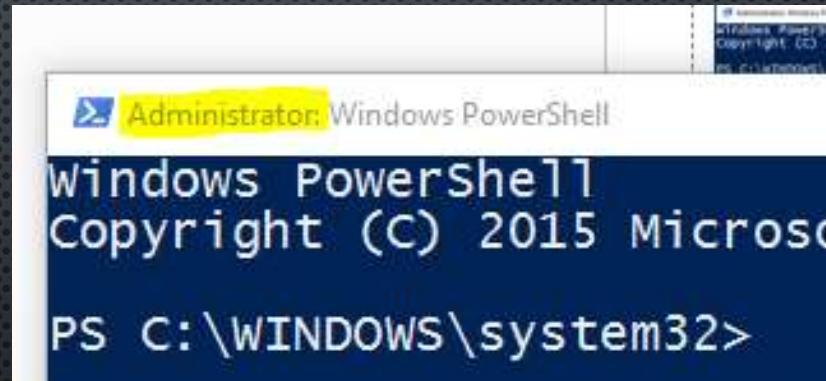
- MANAGEMENTTOOL VOOR SYSTEEMBEHEER
- GUI NIET OM TE AUTOMATISEREN
- COMMANDLINE TOOL, GEEN SCRIPTAAL ...
- VERGELIJKBAAR UNIX/LINUX
- OBJECT GEORIËNTEERDE SHELL (.NET)

BASIS : CONSOLE VERUS ISE

Powershell Console

Powershell ISE

Tip : Run as Administrator



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\WINDOWS\system32>
```

BASIS : CMDLETS AND ALIASES

Alias	CMdlet
dir, ls	Get-ChildItem
copy, cp	Copy-Item
dir /s	dir -recurse

```
PS C:\> get-alias -Definition get-childitem
 CommandType      Name
 -----          ---
 Alias           dir -> Get-ChildItem
 Alias           gci -> Get-ChildItem
 Alias           ls -> Get-ChildItem

PS C:\> get-Alias dir
 CommandType      Name
 -----          ---
 Alias           dir -> Get-ChildItem
```

Naamgeving: Werkwoord-Zelfstandig naamwoord (Verb-Noun)

```
get-eventlog -logname Security -ComputerName localhost,Server01 -Verbose
```

BASIS : TABCOMPLETION, AFKORTINGEN, ALIAS

- HANDIGE SHORTCUTS VOOR HET TYPEN VAN COMMANDO'S
- TAB COMPLETION
 - TYP: GET-SE[TAB]
 - TYP: GET-SERVICE –CO[TAB]
- PARAMETER AFKORTINGEN
 - TYP: GET-SERVICE –COMP LOCALHOST
- PARAMETER ALIASES
 - TYP: GET-SERVICE –CN LOCALHOST

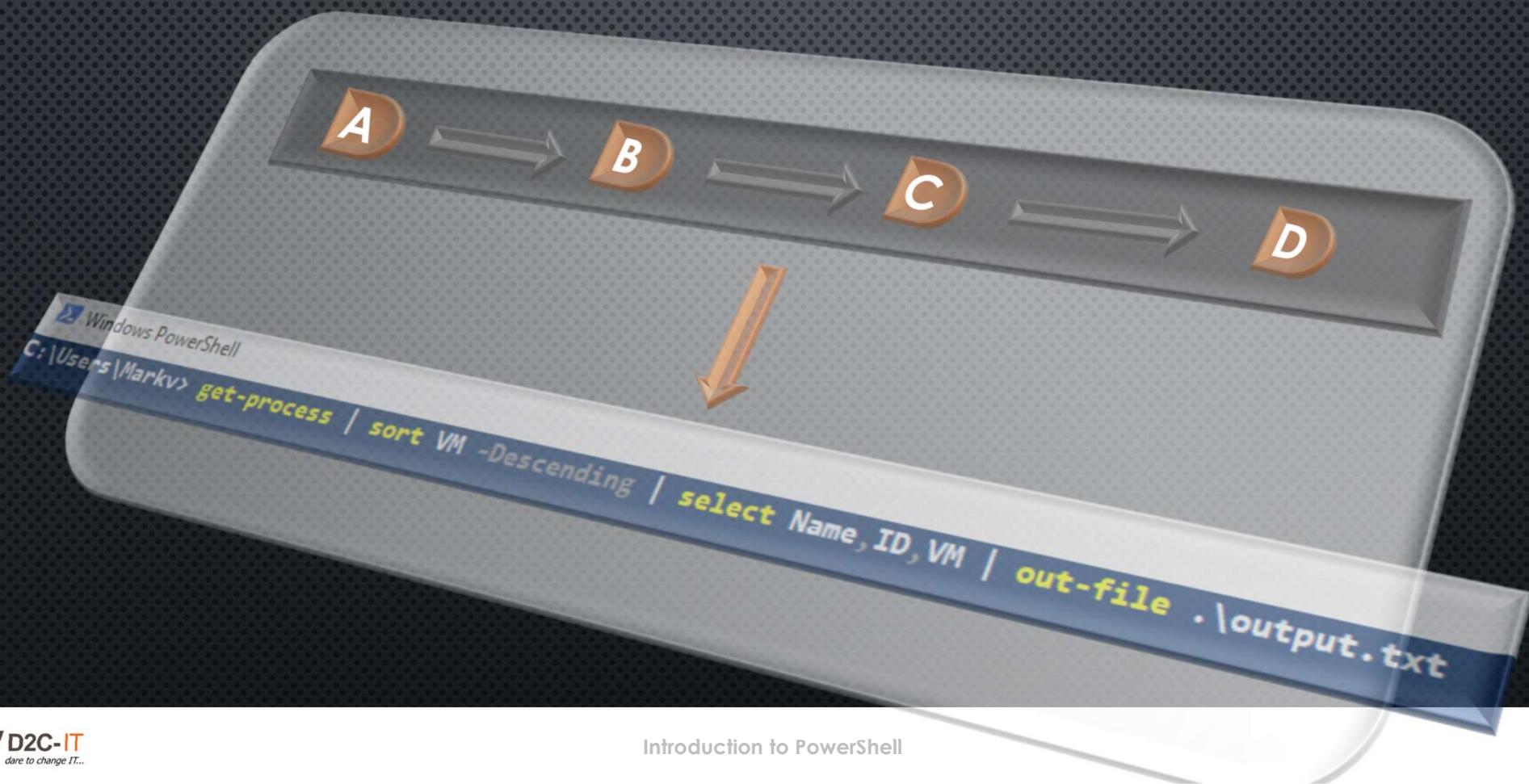
BASIS : PARAMETER

- NAMED EN POSITIONELE PARAMETERS
 - TYP: GET-SERVICE –COMPUTERNAME LOCALHOST
 - TYP: GET-SERVICE –NAME WUAUSERV
 - TYP: GET-SERVICE WUAUSERV

SYNTAX

```
Get-Service [[-Name] [<String[]>]] [-ComputerName [<String[]>]] [-DependentServices] [-Exclude [<String[]>]] [-Incl  
ude [<String[]>]] [-InformationAction {SilentlyContinue | Stop | Continue | Inquire | Ignore | Suspend}] [-Informat  
ionVariable [<System.String>]] [-RequiredServices] [<CommonParameters>]
```

BASIS : DE PIPELINE



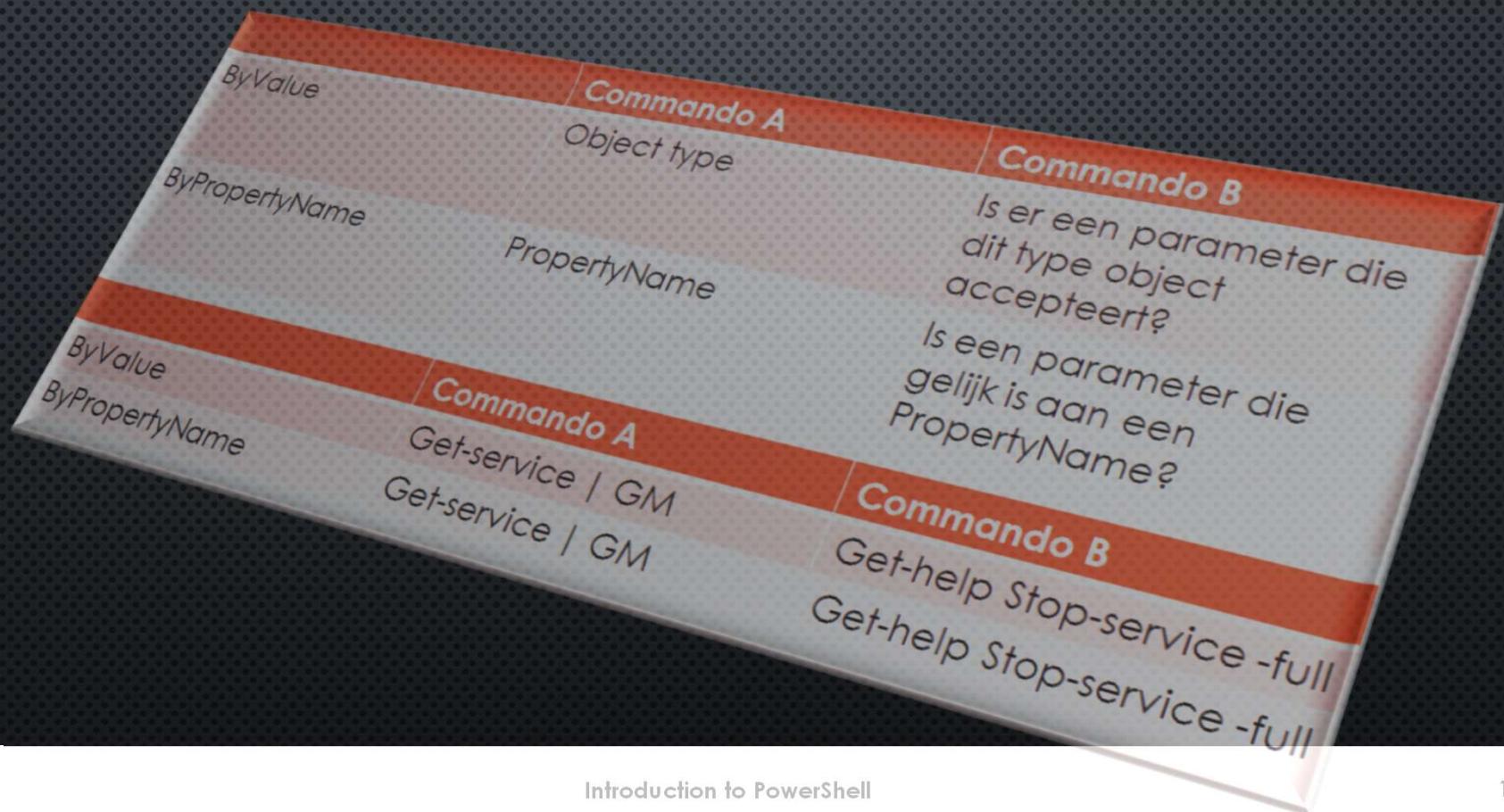
QUIZ & DEMO

- Open je browser en open :

<https://kahoot.it/>

→ of installeer app op je telefoon

DE PIPELINE (PARAMETER BINDING)



DE PIPELINE (PARAMETER BINDING)

- OBJECTEN WORDEN DOOR DE PIPELINE VERPLAATS VAN HET ENE CMDLET NAAR DE ANDERE
- HOE STUURT POWERSHELL DATA DOOR DE PIPELINE?
 - **ByValue**
 - **ByPropertyName**

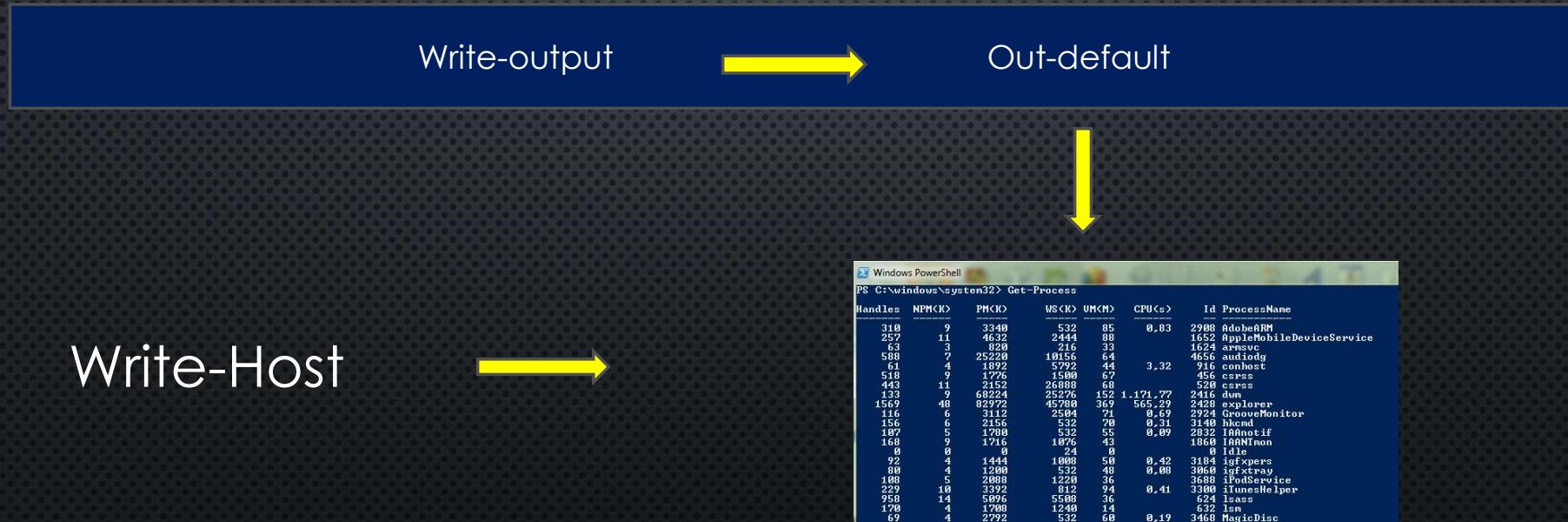
HELP SYSTEEM

- HOE VINDEN WE NU AL DIE COMMANDO'S?
 - GET-COMMAND
 - GET-HELP, HELP, MAN
- HELP! ONZE HELP IS NIET UP TO DATE!
 - UPDATE-HELP ZORGT VOOR EEN UP TO DATE HELP
- HOE GEBRUIKEN WE DE HELP?
 - GET-HELP *SERV*
 - GET-HELP GET-SERVICE
 - GET-HELP GET-SERVICE –EXAMPLE

HELP SYSTEEM

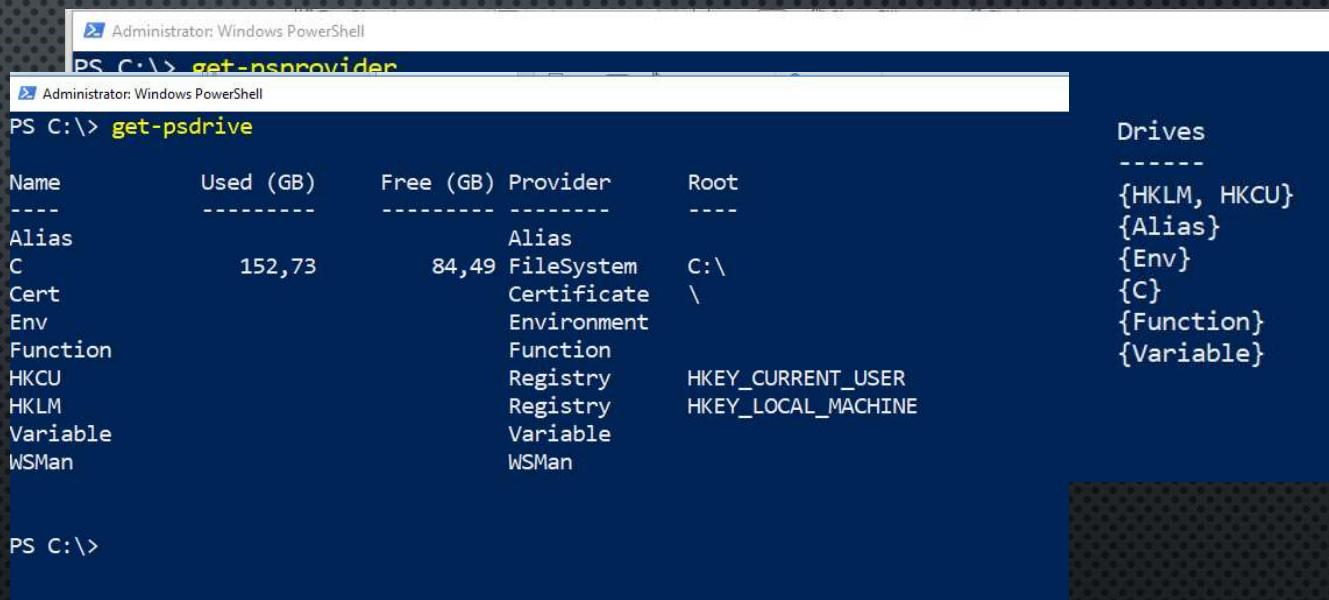
- HELP INTERPRETATIE → GET-HELP <CMDLET>
 - [-VERPLICHT] <STRING> [-OPTIONEEL <STRING[]>]
[[-POSITIONEEL] <STRING[]>] [<COMMONPARAMETERS>]
- VOLLEDIGE HELP → HELP <CMDLET> -FULL
- PARAMETER WAARDEN → (STRING/INT/DATETIME)
- ABOUT HELP → GET-HELP ABOUT
- ONLINE HELP → GET-HELP <CMDLET> -ONLINE
GET-HELP <CMDLET> -SHOWWINDOW

WRITE-OUTPUT VS WRITE-HOST



PROVIDER

- WAT IS EEN PROVIDER?
- GET-PSPROVIDER
- GET-PSDRIVE



The screenshot shows two PowerShell windows. The top window runs the command `get-psprovider`, which lists various providers:

Name	Used (GB)	Free (GB)	Provider	Root
Alias			Alias	
C	152,73	84,49	FileSystem	C:\
Cert			Certificate	\
Env			Environment	
Function			Function	
HKCU			Registry	HKEY_CURRENT_USER
HKLM			Registry	HKEY_LOCAL_MACHINE
Variable			Variable	
WSMan			WSMan	

The bottom window runs the command `get-psdrive`, which lists the drives:

Drives

{HKLM, HKCU}
{Alias}
{Env}
{C}
{Function}
{Variable}

PROVIDER

- SET-LOCATION → CD

```
SET-LOCATION -PATH C:\WINDOWS  
CD C:\WINDOWS
```

- GET-CHILDITEM → DIR

```
GET-CHILDITEM HKCU:\SOFTWARE
```

- NEW-ITEM → CREËER EEN NIEUW ITEM

```
MKDIR TESTFOLDER
```

- WILDCARDS / LITERALPATH

```
GET-ITEM *.EXE  
SET-LOCATION -LITERALPATH HKCU:\SOFTWARE\*
```

OBJECTEN

- IN POWERSHELL BESTAAT ALLES UIT OBJECTEN
- TERMINOLOGIE:
- OBJECT -> DE KUBUS
- PROPERTY -> BREEDTE
- METHOD -> DRAAIEN
- COLLECTION -> MEERDERE KUBUSSEN



OBJECTEN : CORE COMMANDS

- CMDLET ALIAS
- SELECT-OBJECT SELECT
- SORT-OBJECT SORT
- WHERE-OBJECT WHERE / ?
- FOREACH-OBJECT FOREACH / %
- TIP: WAT DOET POWERSHELL MET HET OBJECT !!!!

OBJECTEN : CORE COMMANDS

SORTEREN EN SELECTEREN

- SORT-OBJECT

```
get-process | Sort-object VM -Descending
```

- SELECT-OBJECT

```
get-process | select-object -Property name, ID, VM, PM
```

- GET-MEMBER

```
get-process | Sort-object VM -Descending | select-object -Property name, ID, VM
```

```
get-process | Sort-object VM -Descending | select-object -Property name, ID, VM | GM
```

OBJECT : CORE COMMANDS

- **FILTEREN**

TWEE MODELLEN VOOR VERMINDEREN VAN RESULTATEN:

➤ LINKS FILTEREN

→ FILTEREN MET 1E CMDLET

```
get-service -name wuauserv
```

➤ RECHTS/ITERATIEF FILTEREN

→ PIPELINE CMDLET

```
get-service | where-object {$_.name -eq "wuauserv"}  
get-service | where name -eq wuauserv
```

OBJECTEN : CORE COMMANDS

- **VERGELIJKEN**
VERGELIJKEN VAN TWEE OBJECTEN LEVERT ALTIJD TRUE OF FALSE
- -EQ / -NE
- -GE / -LE
- -GT / -LT
- -CEQ / -CNE / -CGT / CGE / CLE (HOOFDLETTER GEVOELIG)
- -AND / -OR
- -LIKE

OBJECTEN : CORE COMMANDS

- FILTEREN OBJECTEN UIT DE PIPELINE

WHERE-OBJECT

(ALIAS → WHERE OF ?)

```
Get-Service | Where-Object -filter { $_.Status -eq 'Running' }
Get-Service | Where { $_.Status -eq 'Running' }
Gsv | ? { $_.Status -eq 'Running' }
```

FOREACH-OBJECT

(ALIAS → FOREACH OF %)

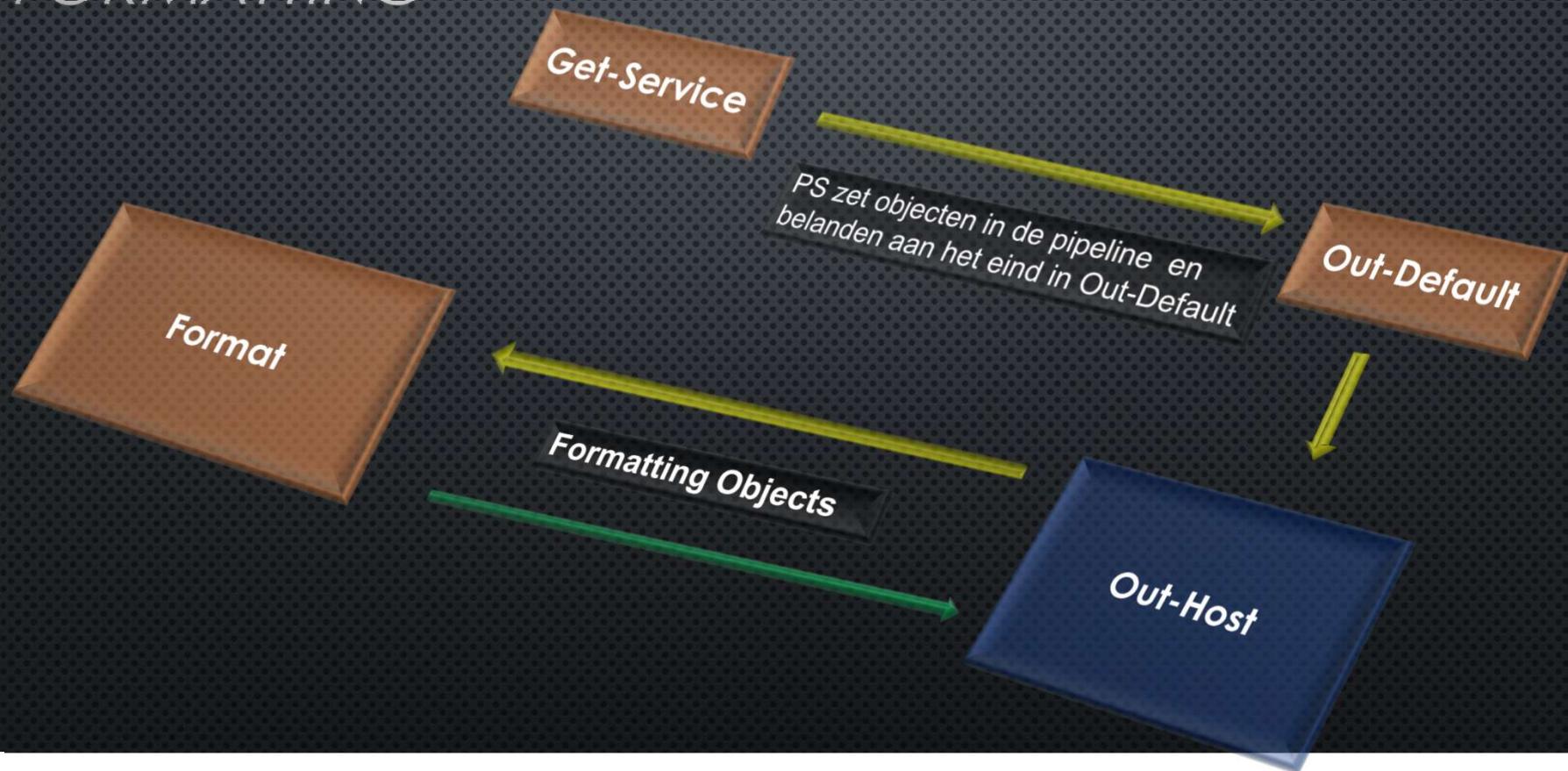
```
Get-content .\computers.txt |
  Foreach{
    Get-Service -ComputerName $_ | Where { $_.Status -eq 'Running' }
  }#Foreach

  Foreach($_ in (Get-content .\computers.txt)){
    Get-Service -ComputerName $_ | Where { $_.Status -eq 'Running' }
  }#Foreach
```

FORMATTING

- AAN HET EIND VAN DE PIPELINE KOMT ALTIJD OUT-DEFAULT
- STUURT OBJECTEN NAAR OUT-HOST
- OUT-HOST BEKIJKT HET OBJECT EN ZIJN REGELS
- REGELS EN DEFINITIES ZIJN OPGESLAGEN IN SPECIALE XML FILES IN \$PSHOME
 - DEFAULT VIEW?
 - DEFAULT PROPERTY SET?
 - HOEVEEL PROPERTIES?
- OUT-HOST GEBRUIKT DE FORMATTING CMDLETS
- JE KUNT DE DEFAULT VIEW ZELF AANPASSEN.

FORMATTING



VARIABELEN

- ZIE EEN VARIABELE ALS EEN DOOS
- HOE MAAK JE EEN VARIABELE?

```
$process = Get-Process  
$process = (get-process -name explorer)
```

- ALTERNATIEVE METHODES:

```
New-Variable -name Process -value (Get-Process)  
Get-Variable -name Process  
Set-Variable -name Process -Value (get-process -name explorer)
```



VARIABELEN

- VARIABELEN ZIJN PER POWERSHELL SESSIE
 - ALLE VARIABELEN ZITTEN IN EEN PSDRIVE VARIABLE:
 - NAMEN MOGEN HEEL LANG ZIJN
 - HOUDT DE NAMEN LOGISCH \$SERVICE = GET-SERVICE
 - MEESTAL LETTERS, CIJFERS EN underscores
 - SPATIES MOGEN, MAAR.. \${VARIABELE NAAM}
 - NIET MEER VARIABELE PREFIXEN MET TYPE \$STRSERVICE

VARIABELEN

- ER ZIJN TWEE TYPEN QUOTES
- HET ESCAPE KARAKTER (BACKTICK)
- MEERDERE OBJECTEN IN EEN VARIABELE (ARRAYS/HASHTABLES)
- COLLECTIE VARIABELEN BINNEN QUOTES (SUBEXPRESSION)
- TYPE BEPALEN VAN VARIABELEN

'WAARDE' & "WAARDE"

` , `N

\$VAR = 1,2,3

\$HT = @{'KEY'='VALUE'}

"\$(\$VAR[0])"

\$VAR.GETTYPE()

QUIZ & DEMO

- Open je browser en open :

<https://kahoot.it/>

→ of installeer app op je telefoon

SCRIPTING

- HET PLAATSEN VAN OPDRACHTEN IN EEN FILE
- EXTENTIE IS <FILENAME>.PS1
- SCRIPT-EDITORS
 - POWERSHELL ISE
 - VISUAL CODE
 - POWERGUI (QUEST)
 - NOTEPAD ++
 - PRIMALFORMS (NIET GRATIS)

SCRIPTING

- PS1 FILENAME EXTENSION STAAT NIET NAAR POWERSHELL
- SCRIPTS DRAAIEN ALLEEN MET HET VOLLEDIGE PAD
 - D:\SCRIPT.ps1
 - .\SCRIPT.ps1
 - SCRIPT.ps1 [TAB]
- SCRIPTS DRAAIEN NIET DOOR DE EXECUTION POLICY
 - RESTRICTED
 - ALLSIGNED
 - REMOTESIGNED
 - **Unrestricted**
 - **Bypass**

SCRIPTING : SCOPE

- GLOBAL
- SCRIPT
- FUNCTION



SCRIPTING

- PARAMETISE
- PARAM()
- BEGIN{ }
- PROCESS { }
- END { }
- TIP : CTRL +

```
1 Function New-Folder {
2     <#
3     .Synopsis
4
5     .Description
6     .Parameter Path
7     .InputType String
8     .OutputType System.String
9     .Parameter Force
10    .InputType Boolean
11    .Defaultvalue $false
12
13    .Process {
14        #check if Folder exists
15        If(!(Test-Path -Path $Path)){
16            Write-verbose "      - Create Folder $Path"
17            Try{
18                mkdir "$Path" -ErrorAction stop | Out-Null
19            }
20            End{
21                Write-host "Endtime      : $((get-date).ToString('HH:mm:ss'))" -ForegroundColor Yellow
22                Write-host "#####" -ForegroundColor Yellow
23
24                return $result
25            }#End
26
27        }
28        $result = $true
29    }Else{
30        Write-verbose "      - Error creating folder $path !!"
31        $result = $false
32    }#end IF
33
34    }Else{
35        Write-verbose " No action needed. Folder already exists"
36        $result = $true
37    } #end IF
38
39    }#process
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54}
```

