POWERSHELL

THE BASICS

MARK VAN DE WAARSENBURG

D2C-IT
dare to change IT...

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

AGENDA

01. Powershell?	02. BASIS	03. Pipeline	04. Objects Core Commands
05. Formatting	06. Variables	07. Scripting	08. Credentials
08. Credentials			

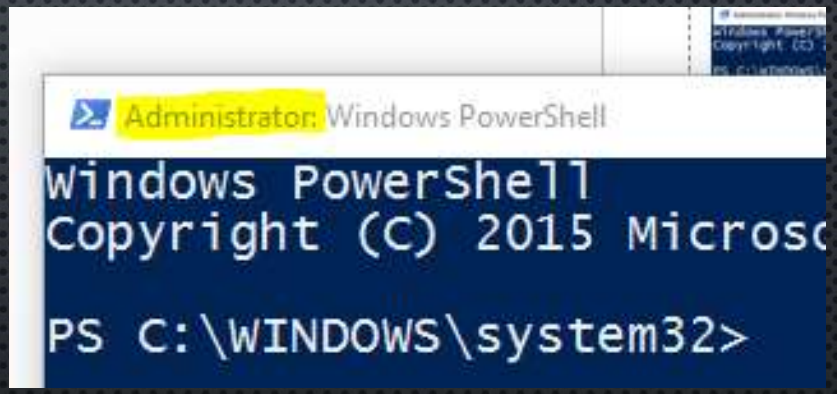
WAT IS POWERSHELL?

- 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.



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

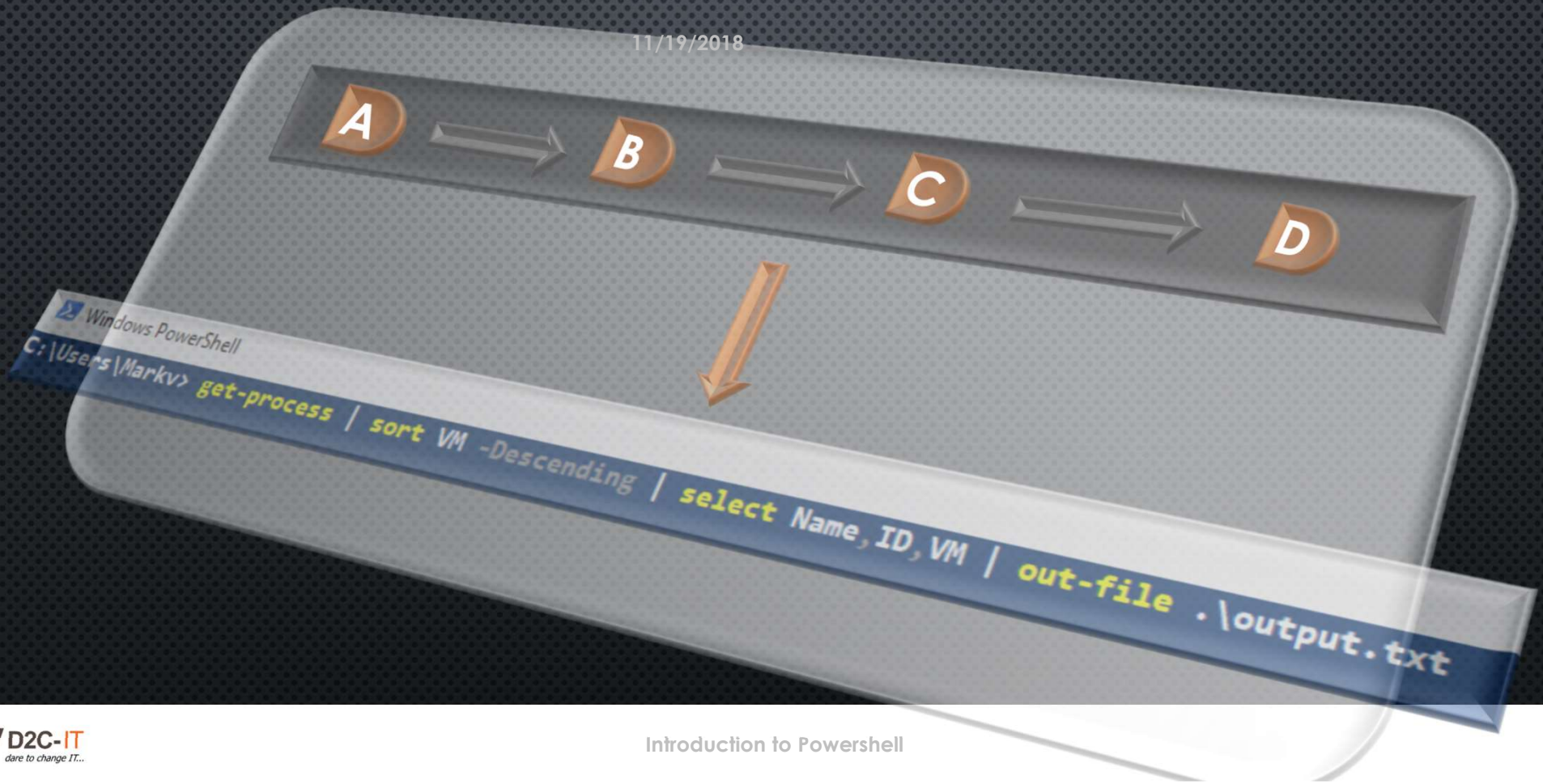
BASIS : PARAMETER

- 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

11/19/2018



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

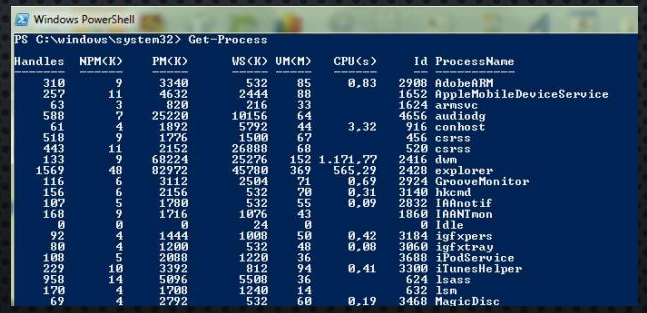
- HELP INTERPRETATIE → GET-HELP <CMDLET>
  - [-VERPLICHT] <STRING> [-OPTIONEEL <STRING[]>]
    [[-POSITIONEEL] <STRING[]>] [<COMMONPARAMETERS>]

- VOLLEDIGE HELP → HELP <CMDLET> -FULL
- PARAMETER WAARDES → (STRING/INT/DATETIME)
- ABOUT HELP → GET-HELP ABOUT
- ONLINE HELP → GET-HELP <CMDLET> -ONLINE
  GET-HELP <CMDLET> -SHOWWINDOW

WRITE-OUTPUT VS WRITE-HOST

Write-output → Out-default

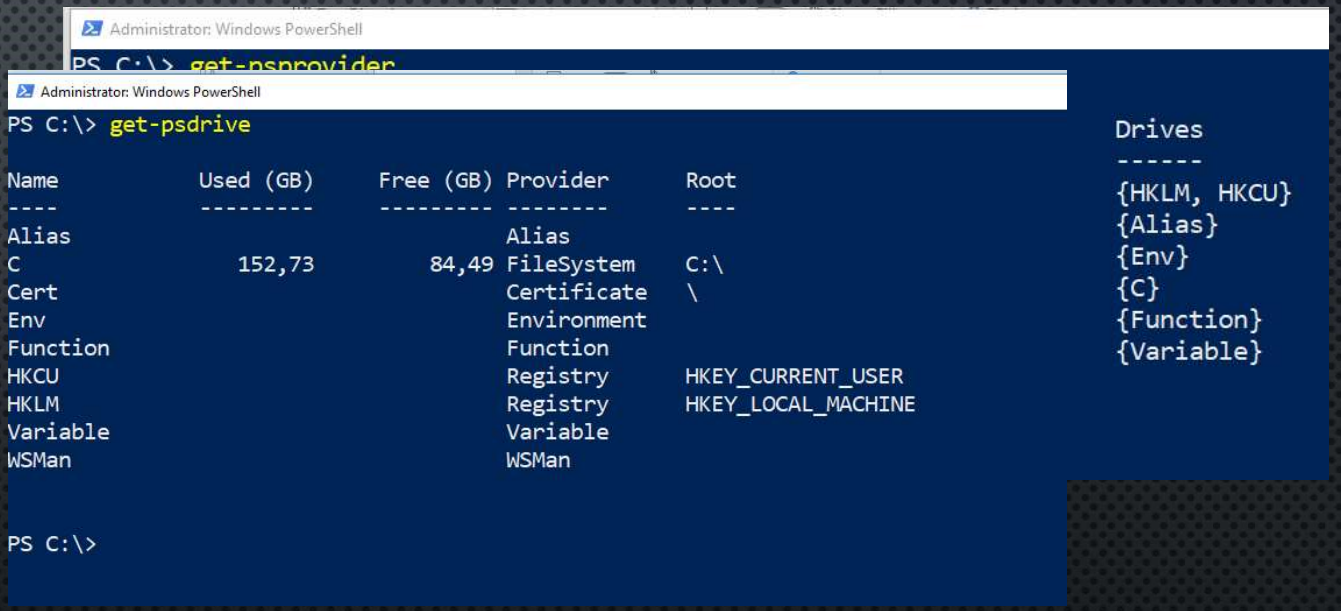
Write-Host →



PROVIDER

- WAT IS EEN PROVIDER?

- GET-PSPROVIDER
- GET-PSDRIVE



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 %)

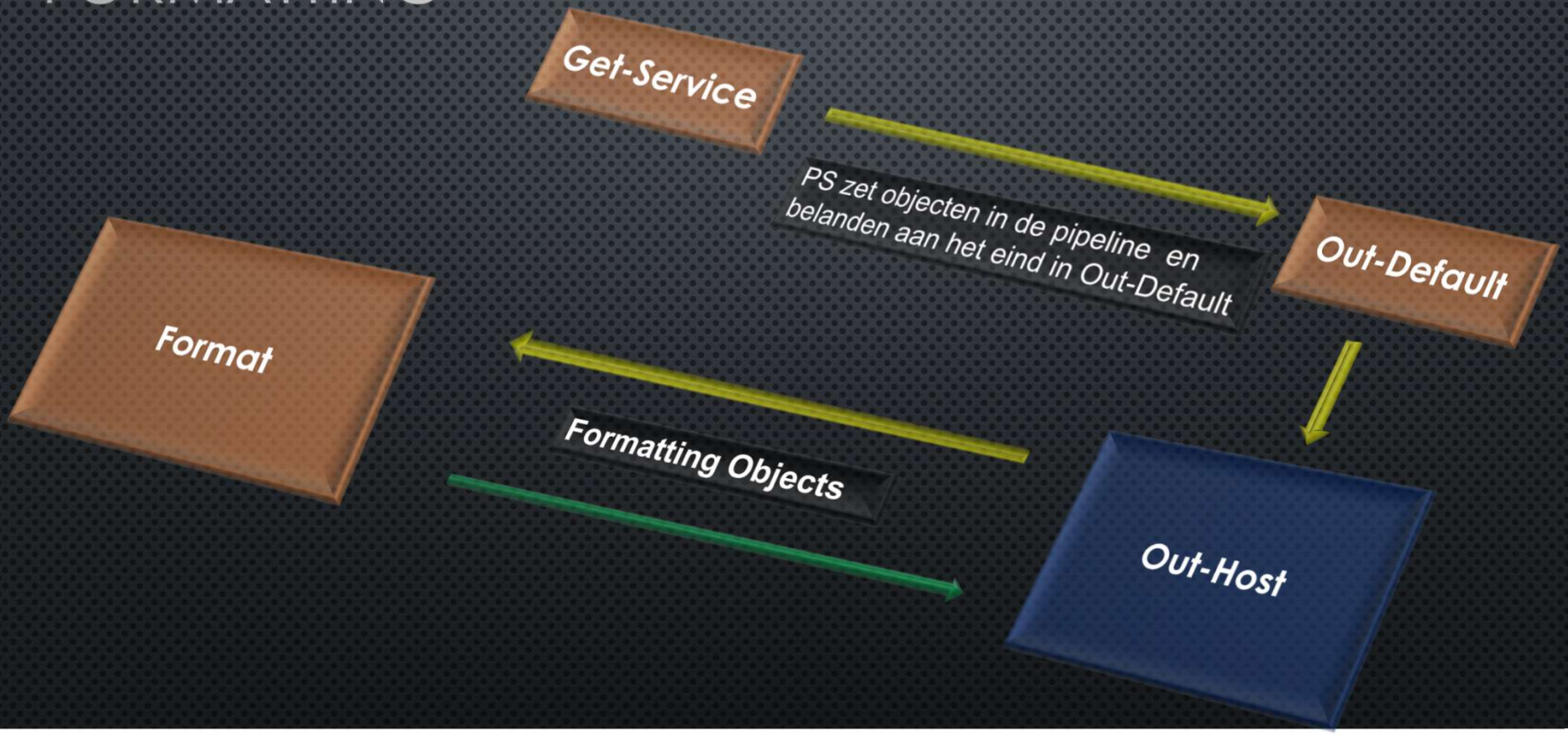
```
Get-content .\computers.txt |
    Foreach{
            Get-Service -ComputerName $_| Where { $_.Status -eq 'Running'}
    }#Foreach

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


Get-Service
PS zet objecten in de pipeline en belanden aan het eind in Out-Default
Out-Default
Format
Formatting Objects
Out-Host

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
- MEESTAL LETTERS, CIJFERS EN UNDERSCORES
- SPATIES MOGEN, MAAR..
- NIET MEER VARIABELE PREFIXEN MET TYPE

$SERVICE = GET-SERVICE

${VARIABELE NAAM}

$STRSERVICE

VARIABELEN

- Er zijn twee typen quotes — 'Waarde' & "Waarde"
- Het escape karakter (Backtick) — ` , `n
- Meerdere objecten in een variabele (Arrays/HashTables) — $var = 1,2,3 / $HT = @{'key'='value'}
- Collectie variabelen binnen quotes (Subexpression) — "$($var[0])"
- Type bepalen van variabelen — $var.GetType()

QUIZ & DEMO

- Open je browser en open :

https://kahoot.it/

→ of installeer app op je telefoon

SCRIPTING

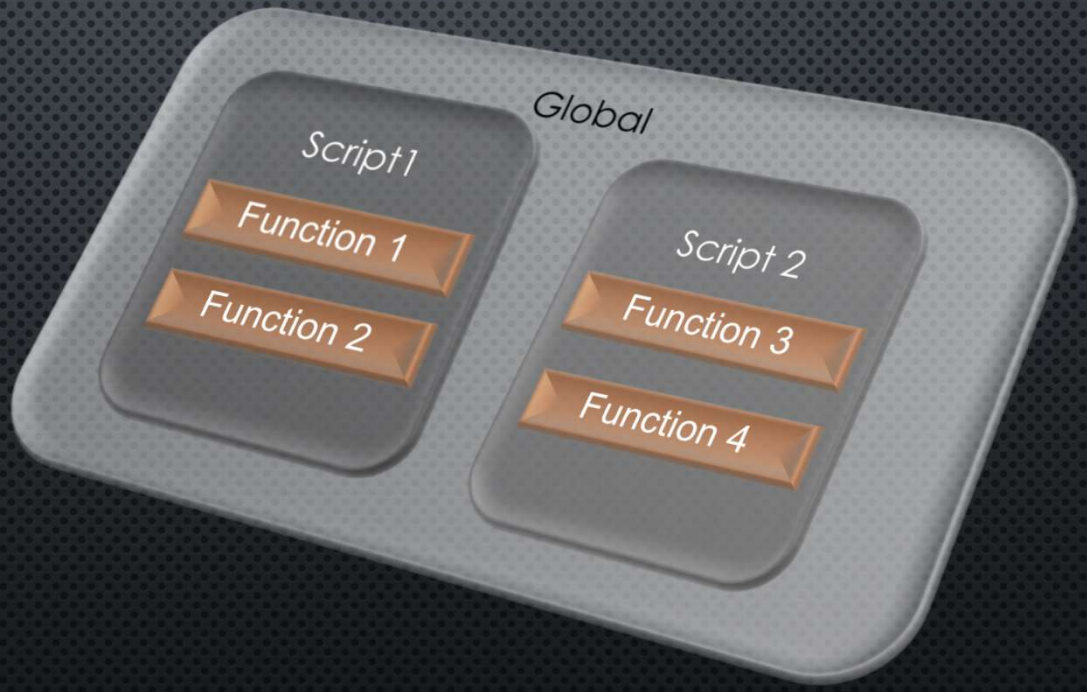
- HET PLAATSEN VAN OPDRACHTEN IN EEN FILE
- EXTENTIE IS <FILENAME>.PS1
- SCRIPT-EDITORS
  - POWERSHELL ISE
  - VISUAL CODE
  - POWERGUI (QUEST)
  - NOTEPAD ++
  - PRIMALFORMS (NIET GRATIS)

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



SCRIPTI

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

```
1  Function New-Folder {
2      <#
3      .Synopsis
30     Process{
31         #check if Folder exists
32             If(!(Test-Path -Path $Path)){
33                 Write-verbose "    - Create Folder $Path"
34                 Try{
35                     mkdir "$Path" -ErrorAction stop | Out-Null
55     End{
56
57         Write-host "Endtime    : $((get-date).ToShortTimeString())"  -ForegroundColor Yellow
58         Write-host "##########################################"  -ForegroundColor Yellow
59
60         return $result
61     }#End
43                     $result = $true
44                 }Else{
45                     Write-verbose "    - Error creating folder $path !!"
46                     $result = $false
47                 }#end IF
48             }else{
49                 Write-verbose " No action needed. Folder already exitsts"
50                 $result = $true
51             }  #end IF
52
53     }#process
54
```