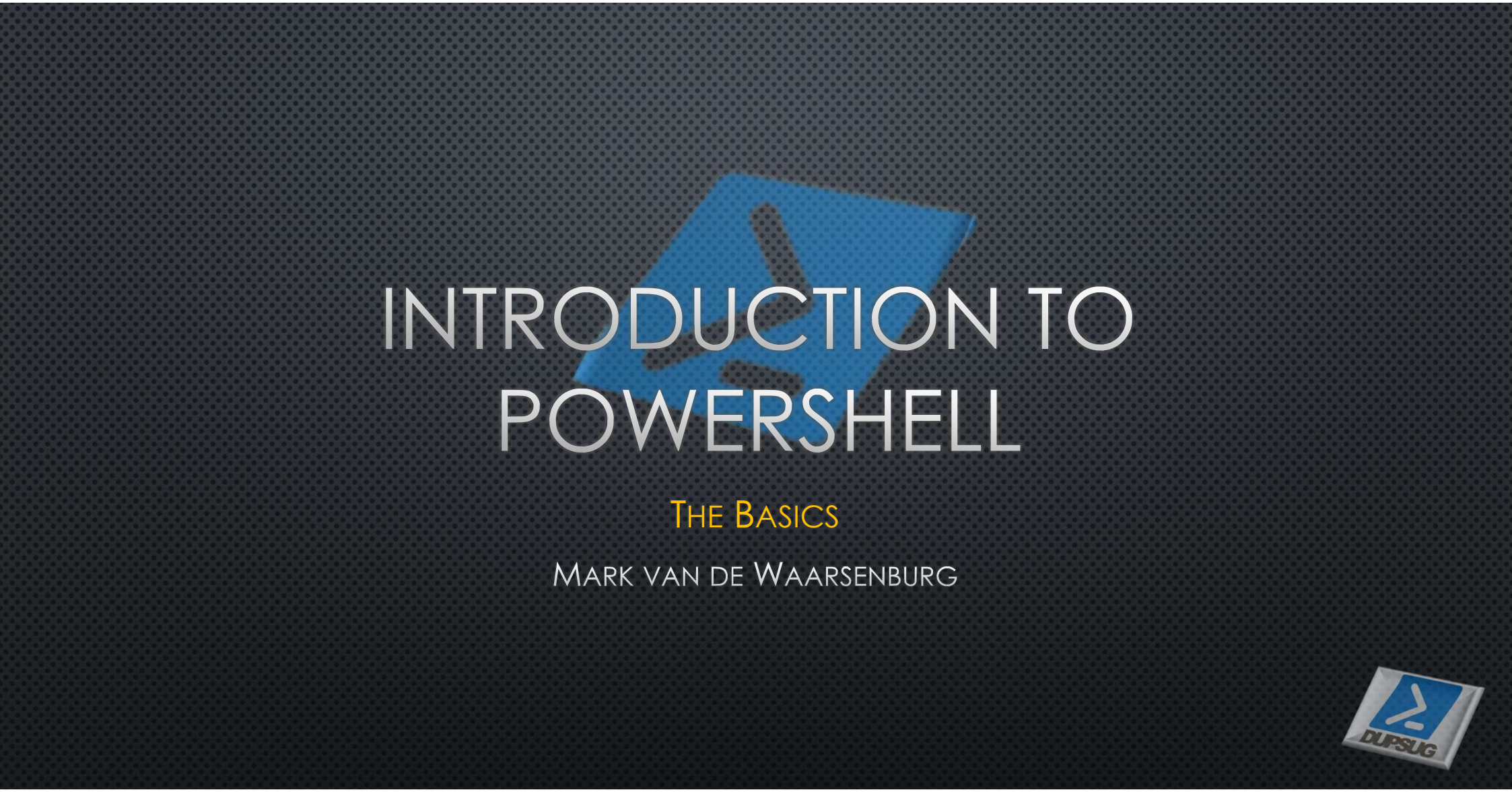


VOORSTELLEN: MARK VAN DE WAARSENBURG

- 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://github.com/D2CIT/IntroductionPowershe

https://d2c-it.nl

AGEN

01. Po

05. Fo

```powershell
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 destry 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, Conti
                $Continue = $true
            } Yes {
                if($Destroy){
                    Write-warning "Yes is not
                }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_
```

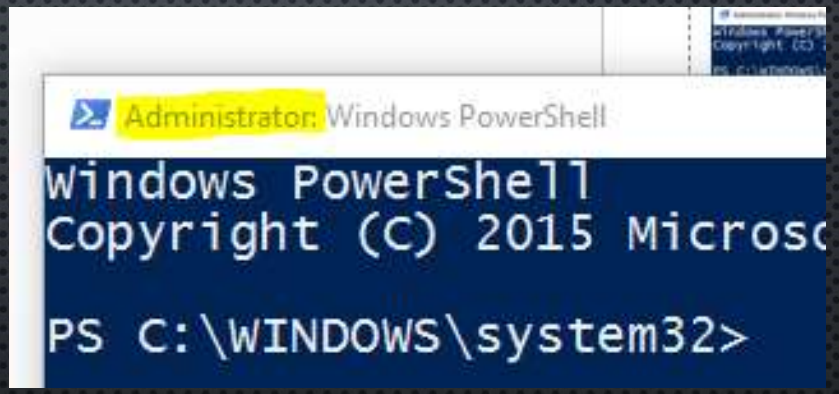
WAT IS POWERSHELL?

- 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.



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

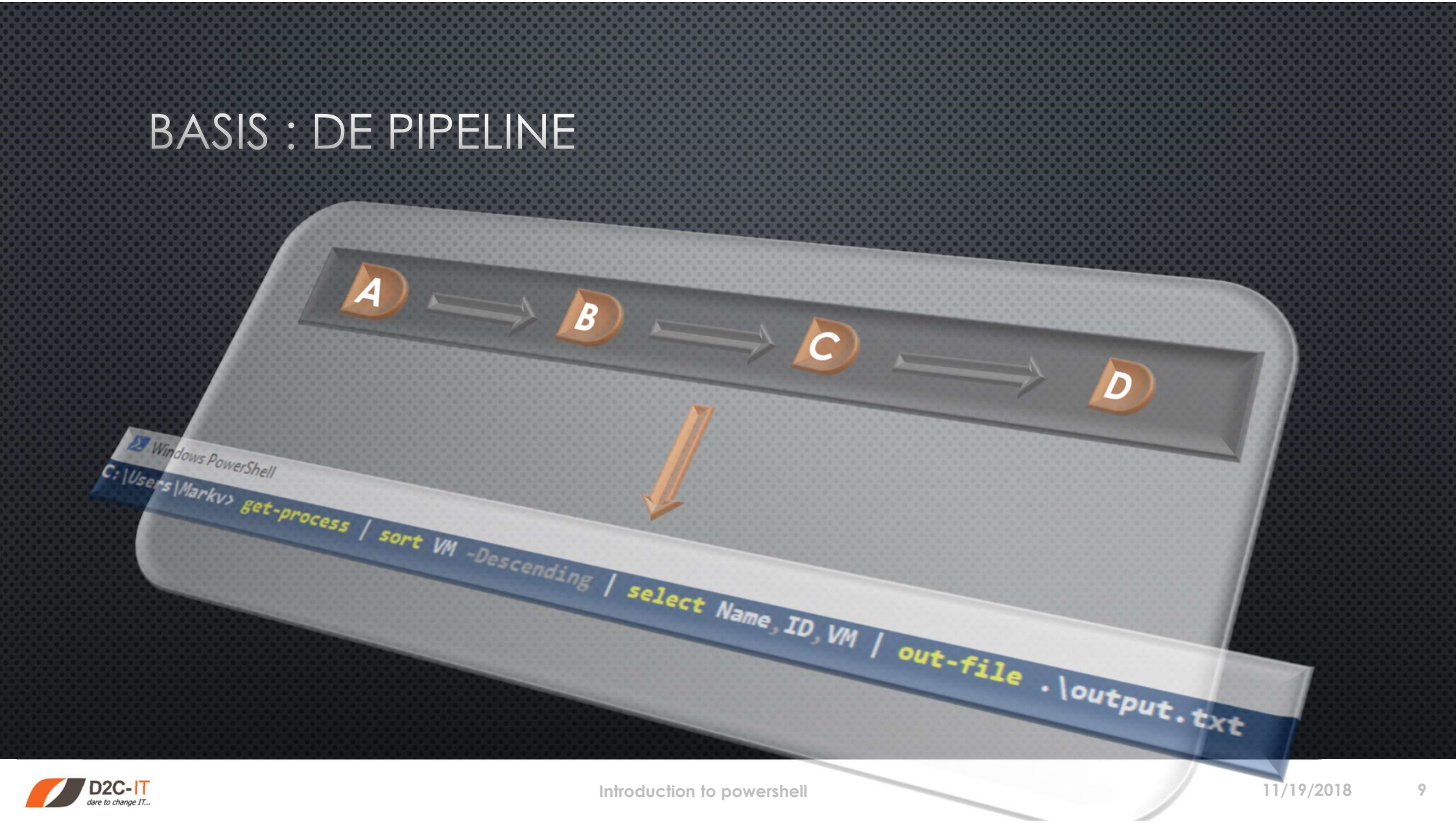
- HANDIGE SHORTCUTS VOOR HET TYPEN VAN COMMANDO'S
- TAB COMPLETION
  - TYP: GET-SE[TAB]
  - TYP: GET-SERVICE –CO[TAB]
- PARAMETER AFKORTINGEN
  - TYP: GET-SERVICE –COMP LOCALHOST
- PARAMETER ALIASES
  - TYP: GET-SERVICE –CN LOCALHOST

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



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

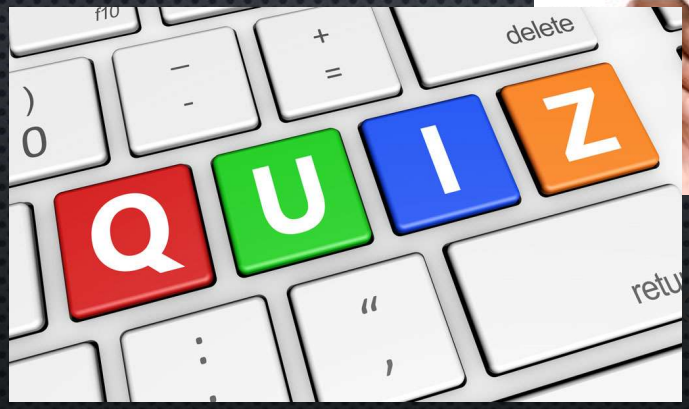
HELP SYSTEEM

- 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

QUIZ & DEMO

- Open je browser en op



WAT IS HET JUISTE COMMANDO OM DE EIGENSCHAPPEN (PROPERTIES) VAN DE WINDOWS UPDATE SERVICE TE ZIEN ?

(NAMED & POSITIONELE PARAMETERS)

get-process –name wuauserv

get-service wuauserv

get-process wuauserv

get-process –computers .

MET WELK COMMANDO KUN JE NIET ALLE E BEGINNEN MET GET?

Get-command –name get

Get-help –name get-*

Man get-*

Help get*

WELKE COMMANDO'S ZIJN ONJUIST?

Get-service –name wuauserv –c .

gsv wuauserv –computername .

Get-services -name wuauserv –com .

Get-service localhost –name wuauserv

HOE CHECK JE WAT DE ALIAS VAN HET CMDLET GET-PROCESS IS?

Get-Alias –name get-process

Get-Alias –Definition get-process

Get-Alias get-process

Get-Alias –name gps

WE RUNNEN ONDERSTAAND COMMAND. WAT IS HET RESULTAAT?
GET-PROCESS | SELECT-OBJECT NAME,ID |SORT-OBJECT VM -DECENDING

Alle processen gesorteerd op naam in alfabetische volgorde

Alle processen met alleen de properties naam en ID, gesorteerd op vm van klein naar groot.

Alle processen met alleen de properties naam en ID, gesorteerd op vm van groot naar klein.

Geen output.

WAT IS HET KORTSTE MOGELIJKE COMMAND VOOR :

GET-SERVICE -COMPUTER LOCALHOST| SORT-OBJECT STATUS | WHERE {$_.NAME –LIKE "W*"}

gsv -c . | sort status | ? name -like w*

gsv –n w* | sort-object status

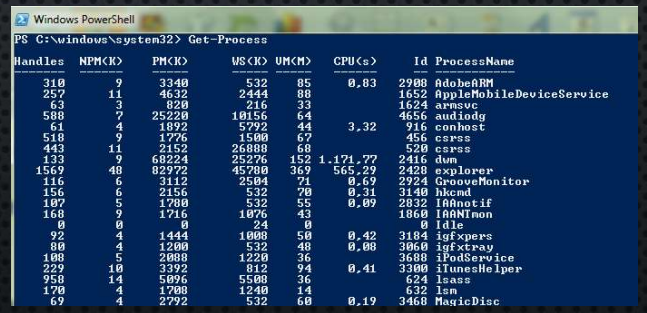
gsv w* | sort status

gsv | sort -status | ? name –like w*

WRITE-OUTPUT VS WRITE-HOST

Write-output → Out-default

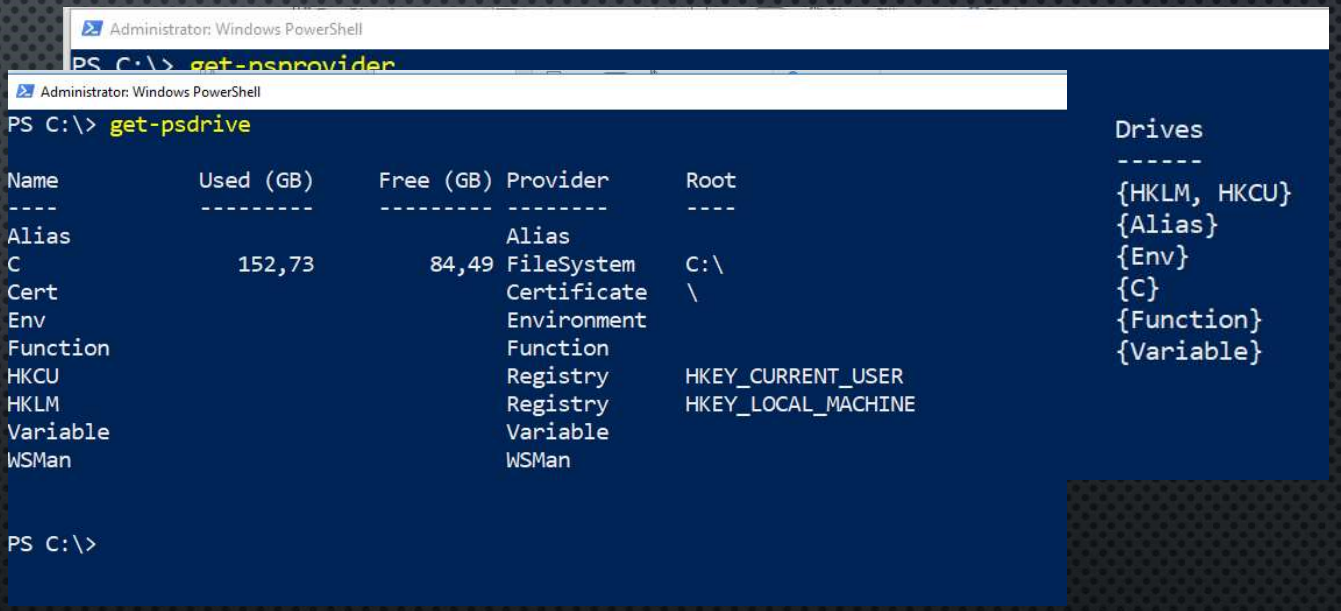
Write-Host →



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

- GET-MEMBER

```
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 %)

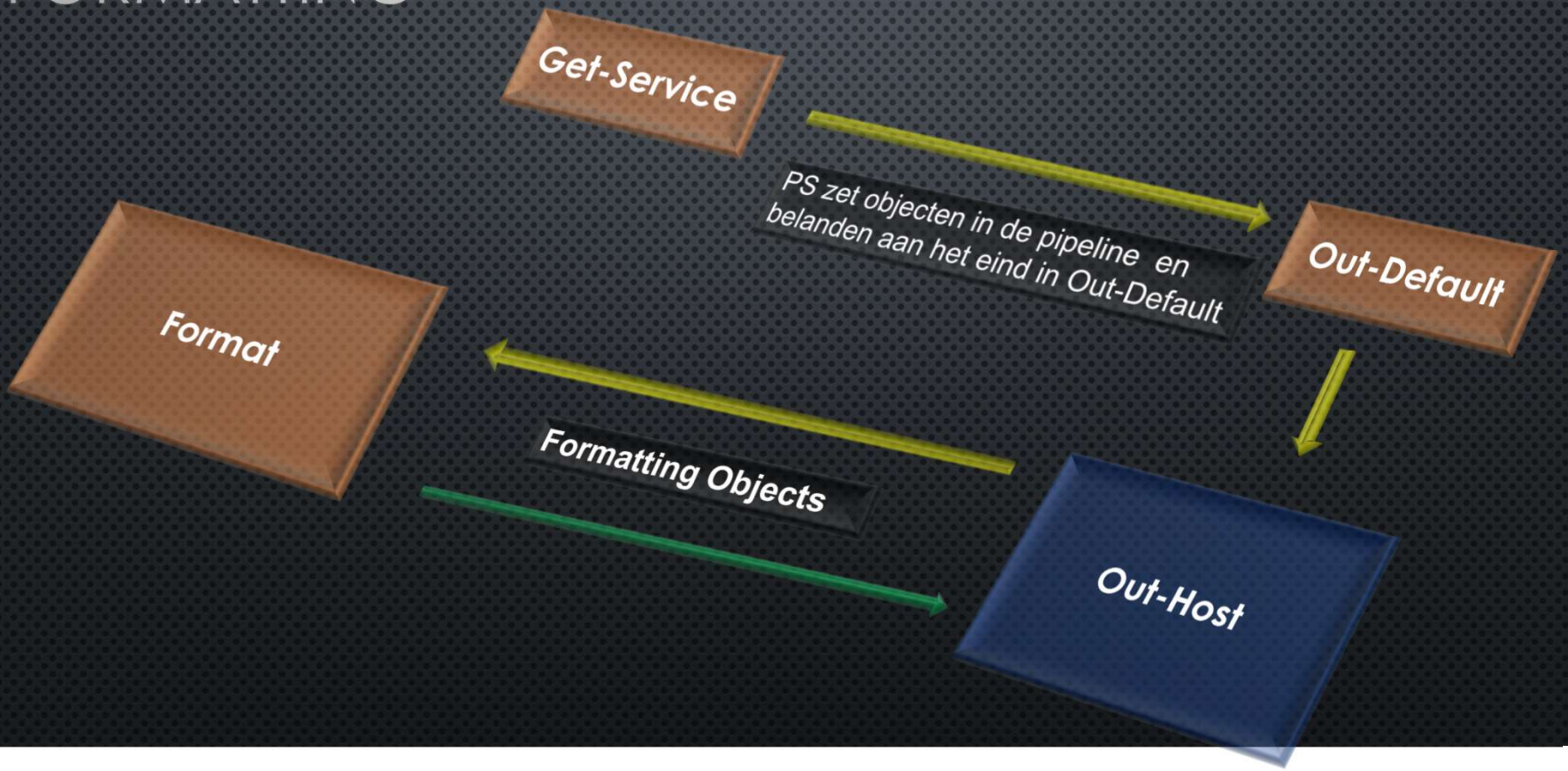
```
Get-content .\computers.txt |
    Foreach{
            Get-Service -ComputerName $_| Where { $_.Status -eq 'Running'}
    }#Foreach

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

LIST ALLE RUNNING SERVICES EN SELECTEER ALLEEN DE RUNNING SERVICES ?

Get-service –status running

Get-service | where {$_.status –like “running”}

get-status | where { $_.service –eq “running”}

Get-service | select-object running

CHECK WANNEER HET USERACCOUNT USERNAME VOOR HET LAATST IS AANGEPAST?

get-aduser -Identity username | select whenChanged

get-aduser -Identity username -Properties whenChanged

(get-aduser -Identity username -Properties * | where {$_.whenChanged -eq "Changed"}

(get-aduser -Identity username -Properties *).whenChanged

LIST ALLE PROCESSEN DIE BEGINNEN MET DE LETTER W ?

Get-process –name w

Get-process | where{$_.name –like “w*”}

Get-process –name w*

Get-process | where {$_.Name –eq “w*”)

HOE KUN JE ZIEN WAT VOOR PROPERTIES, METHODS EN TYPE OBJECT DE OUTPUT IS?

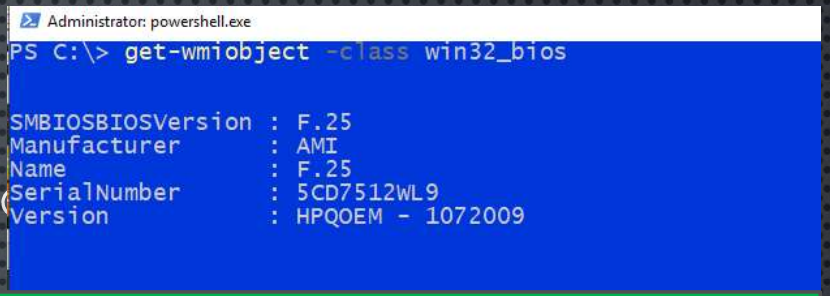
Get-wmiobject | where{$_.object –like Property}

Get-wmiobject | select -object

Get-wmiobject –type object

Get-wmiobject | get-member

VERANDER DE DEFAULT OUTPUT VAN HET COMMANDO "
EN LAAT ALLEEN DE NAAM EN HET SERIENUMMER ZIEN.
ZORG ERVOOR DAT DE OUTPUT NOG BRUIKB



get-wmiobject -class win32_bios | format-table -Property name,serialnumber

get-wmiobject -class win32_bios | select name,serialnumber

get-wmiobject -class win32_bios -Property name,serialnumber | format-table

get-wmiobject -class win32_bios | select name,serialnumber | format-table

EIND DAGDEEL 1

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 op

https:

HOE KUN MAAK JE EEN VARIABLE MET DE NAAM XYZ EN DE WAARDE 123 AAN IN POWERSHELL?

Create-Variable –name XYZ –value 123

New-variable –name XYZ –value 123

$XYZ = 123

New-variable –name $XYZ –value 123

HOE PAS JE DE WAARDE AAN VAN EEN VARIABLE?

$XYZ = 345

get-variable XYZ | set-variable 345

get-variable –name xyz –value 345

Set-variable –name xyz –value 345

HOE VERWIJDER JE EEN VARIABLE?

Remove-variable $XYZ

remove-variable –name XYZ

get-variable –name xyz | remove-variable

(dir variable:) | where {$_.name -like "xyz"} | remove-variable

SCRIPTING

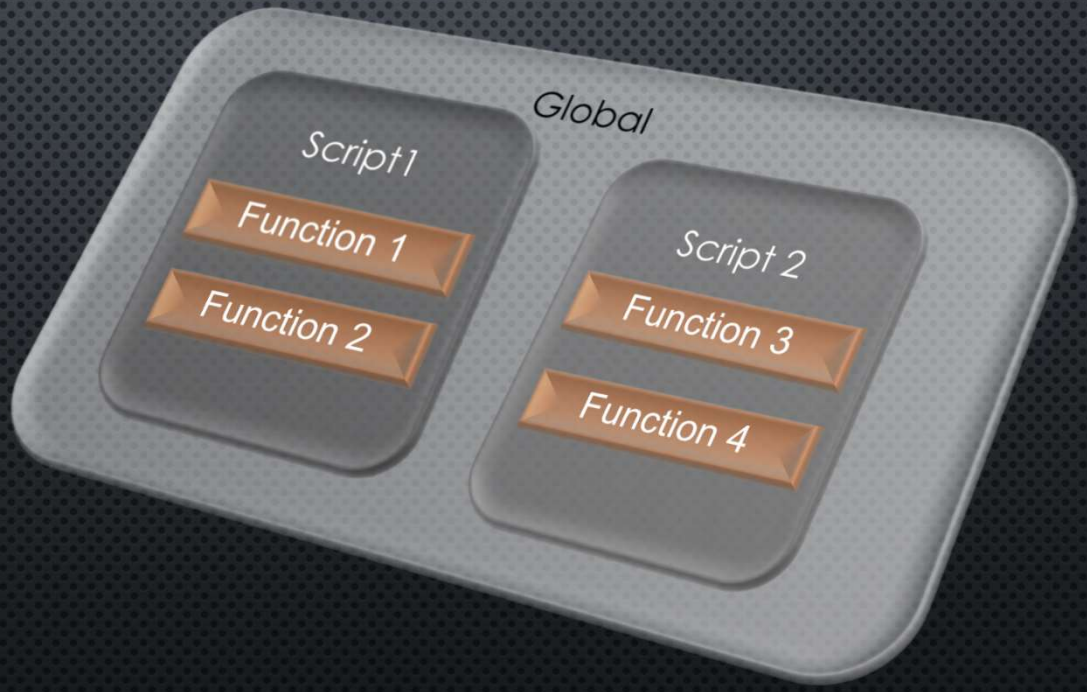
- HET PLAATSEN VAN OPDRACHTEN IN EEN FILE
- EXTENTIE IS <FILENAME>.PS1
- SCRIPT-EDITORS
  - POWERSHELL ISE
  - VISUAL CODE
  - POWERGUI (QUEST)
  - NOTEPAD ++
  - PRIMALFORMS (NIET GRATIS)

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
```

BELANGRIJKE CMDLETS

- Get-help
- Set-executionpolicy
- Get-service
- ConvertTo-Html
- Export-CSV
- Select-Object
- Get-eventlog
- Get-process
- Stop-process
- Get-childitem
- Get-alias
- Get-command
- Sort-object
- Where-object
- Foreach-Object
- Format-list /Format-table
- Get-psprovider
- Get-psdrive
- New-variable (get/set/remove)