Powershell oefeningen

Parameters

Maak enkel gebruik van de cursus, je notities en de PowerShell help functionaliteit voor het oplossen van de onderstaande oefeningen.

Als er een Cmdlet bestaat voor het uitvoeren van de opdrachten dien je deze te verkiezen boven een ander commando.

**Internetgebruik is niet noodzakelijk om deze oefeningen op te lossen.**

# Script: NumberOfFiles

Maak een script NumberOfFiles.ps1 dat het recursief aantal files weergeeft van de meegegeven directory (argument).

## NumberOfFiles.ps1 C:\powershell

Afbeelding met tekst

Automatisch gegenereerde beschrijving

param(

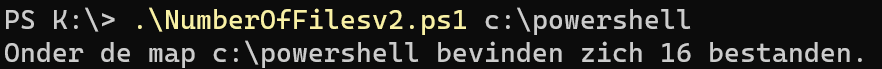
$path

)

$aantalfiles = (Get-ChildItem -File $path -Recurse).count

Write-Output "$aantalfiles"

## NumberOfFilesv2.ps1 C:\powershell



param(

$path

)

$aantalfiles = (Get-ChildItem -File $path -Recurse).count

Write-Output "onder de map $path bevind zich $aantalfiles"

# Script: FreeDiskSpace

## FreeDiskSpace.ps1

## Maak een script FreeDiskSpace.ps1 dat de vrije ruimte weergeeft op basis van het meegegeven argument, zijnde de drive, bv. .\FreeDiskSpace.ps1 C:

**EXRA INFORMATIE**

Get-CimInstance -ClassName CIM\_LogicalDisk | where DriveType -eq 3

Als je **enkel** de C:-drive wil, moet je natuurlijk verder filteren

* + - Get-CimInstance CIM\_LogicalDisk |   
       Where-Object {$\_.DriveType -eq 3 -and $\_.DeviceID -eq "C:"}

Afbeelding met tekst

Automatisch gegenereerde beschrijving

param(

$drive

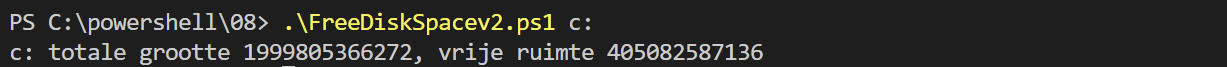
)

Get-CimInstance -ClassName CIM\_LogicalDisk | Where-Object{$\_.DeviceID -eq $drive}

## FreeDiskSpacev2.ps1

Idem A, maar met de volgende output.

**Output:**

****

**param(**

**$drive**

**)**

**$groote = (Get-CimInstance -ClassName CIM\_LogicalDisk | Where-Object{$\_.DeviceID -eq $drive}).Size**

**$freespaceover = (Get-CimInstance -ClassName CIM\_LogicalDisk | Where-Object{$\_.DeviceID -eq "$drive"}).FreeSpace**

**Write-Output "de totale groote $groote , de vrije ruimte $freespaceover"**

## FreeDiskSpacev3.ps1

Idem B, indien de gebruiker geen parameter meegeeft, wordt automatisch “C:” als waarde doorgegeven.

**Output:**



param(

$drive = "c:"

)

$groote = (Get-CimInstance -ClassName CIM\_LogicalDisk | Where-Object{$\_.DeviceID -eq $drive}).Size

$freespaceover = (Get-CimInstance -ClassName CIM\_LogicalDisk | Where-Object{$\_.DeviceID -eq "$drive"}).FreeSpace

Write-Output "de totale groote $groote , de vrije ruimte $freespaceover"

## FreeDiskSpacev4.ps1

Idem C, maar nu zijn er twee mogelijke argumenten.  
argument1 = driveLetter (standaardwaarde c:).  
argument2 = dataUnit (de gewenste data eenheid, standaardwaarde TB.

Zorg dat enkel de waarden MB, GB en TB mogelijke waarden zijn.

Rond de uitvoer van de totale grootte en vrije ruimte af met 2 cijfers achter de komma.

Tip: Voorbeeld

[math]::Round(11.244,2)  
  
**Output:**

Afbeelding met tekst

Automatisch gegenereerde beschrijving

param(

[ValidateSet("MB","GB","TB",ErrorMessage="allen mb,gb of tb")]

$driveunit = "TB",

$drive = "c:"

)

$disk = Get-CimInstance -ClassName CIM\_LogicalDisk | Where-Object { $\_.DeviceID -eq $drive }

if ($disk) {

$size = $disk.Size

$freespace = $disk.FreeSpace

if ($driveunit -eq "GB") {

$size = [math]::Round($size / 1GB, 2)

$freespace = [math]::Round($freespace / 1GB, 2)

}

elseif ($driveunit -eq "MB") {

$size = [math]::Round($size / 1MB, 2)

$freespace = [math]::Round($freespace / 1GB, 2)

}

else{

$size = [math]::Round($size / 1TB, 2)

$freespace = [math]::Round($freespace / 1TB, 2)

}

# If $driveunit is "TB", no conversion is needed as the size is already in terabytes.

Write-Output "De totale grootte van schijf $drive : $size $driveunit en de vrije ruimte $freespace $driveunit"

} else {

Write-Host "Schijf met letter $drive niet gevonden."

}

# Script: Rekenen.ps1

Pas het script Rekenen.ps1 aan zodat je het eerste en tweede getal verplicht via argumenten dient in te geven.

## RekenenV2.ps1

.\RekenenV2.ps1 [-my\_Getal1] <double> [-my\_Getal2] <double>

De volgende syntax dien je implementeren.



**Output:**

Afbeelding met tekst

Automatisch gegenereerde beschrijving

param(

[double]$getal1,

[double]$getal2

)

$som = $getal1 + $getal2

$product = $getal1 \* $getal2

Write-Host "som : $som"

Write-Output "product : $product"

Maak de oefening opnieuw kijk naar de eerste foto van de oefening

# Script: MakeBackup.ps1

Pas het script MakeBackup.ps1 aan met de volgende argumenten.

## MakeBackupV2.ps1

### 

Alle argumenten zijn verplicht.

**Output:**

Afbeelding met tekst

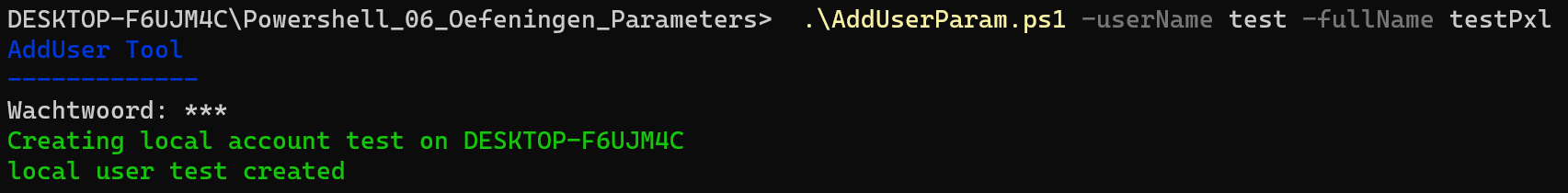
Automatisch gegenereerde beschrijving

## Script: AddUserParam.ps1

### Pas het script AddUser.ps1 aan met de volgende argumenten.

## 

**Output:**



**-------------------------------------------------------------------------------------------------**

**EINDE OEFENING**