Powershell oefeningen

Variabelen

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

# Opdracht:

## Wat is een variabele in Powershell en geef de 3 hoofdbestandsdelen waaruit deze bestaat?

$ , naam, gelijk aanteken

## Maak een variabele “myFirstName” aan en ken je voornaam aan deze variabele toe. Bewijs dat deze variabele een object is.

$naam = "lahcen"

$naam.GetType()

Dan zie je dat het en systemobject is

## Toon alle Members van deze variabele. Welke properties heeft deze variabele (object) en geef telkens een voorbeeld?

Afbeelding met tekst, schermopname, Lettertype, software

Automatisch gegenereerde beschrijving

Length property

## Toon alle Methods van deze variabele (object). Geeft ook het aantal.

Afbeelding met tekst, schermopname, Lettertype, software

Automatisch gegenereerde beschrijving

## Toon op basis van de variabele de lengte van je voornaam.

Afbeelding met tekst, Lettertype, schermopname

Automatisch gegenereerde beschrijving

## Toon je voornaam in hoofdletters (waarde van de variabele niet wijzigen).

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

## Toon enkel de eerste letter van je voornaam. Toon de laatste letter van je voornaam.

Afbeelding met tekst, Lettertype, schermopname

Automatisch gegenereerde beschrijving

## Maak een variabele “myLastName” aan en ken je achternaam aan deze variabele toe. Maak een variabele “myName” aan met je voornaam (spatie) achternaam (spatie) op basis van de aangemaakte variabelen. Toon de waarde van de variabele “myName”. Toon de lengte van de variabele “mName”.

$lastname = "oualach"

$myname = "lahcen " + "$lastname"

Write-Output $myname

$myname.length

## Toon de lengte van de variabele “myName”.

$myname.length

## Verwijder de achterste spatie uit je “myName” variabele en toon vervolgens opnieuw de lengte van je variabele.

### $myname.TrimEnd()

## Maak een variabele “myHosts” aan en ken de volgende items aan deze variabele toe.

* Server01
* Server02
* WorkstationA
* WorkstationB
* WorkstationC

Oplossing:

$myHosts = "Server01", "Server02", "WorkstationA", "WorkstationB", "WorkstationC"

Write-Output $myHosts

### Toon de volledige inhoud van de variabele “myHosts”.

$myHosts = "Server01", "Server02", "WorkstationA", "WorkstationB", "WorkstationC"

Write-Output $myHosts

### Toon het aantal items aanwezig in de variabele “myHosts”. Output:

  
$myhosts2.count

**Oplossing:**

**Write-Output $myHosts.Count**

### Toon de eerste 2 items (1 commando)

Write-Output $myname | select -First 2

### Toon de laatste twee items (1 commando)

Write-Output $myname | select -Last 2

## Opdracht:

### Start de applicatie “Windows Calculator” (niet afsluiten) Maak een variabele “myProces” aan Plaats de systeemprocessen in deze variabele toe en toon vervolgens de uitvoer op het scherm. Bewijs dat deze variabele een object is.

$myprocessor = Get-Process -Name CalculatorApp

$myprocessor.GetType()

### Toon de properties en de methods van deze variabele

$myprocessor | Get-Member

### Bewijs dat deze variabele een object is. Toon door enkel gebruik te maken van de huidige inhoud van de variabele.

### De namen van de processen.

$myprocessor.Name

### Toon door enkel gebruik te maken van de huidige inhoud van de variabele.

### Geef een overzicht van de eerste 3 processen, aflopend gesorteerd op hun starttijd. Toon daarbij enkel de volgende items: id, naam en starttijd.

$myprocessor | select -First 3 | Sort-Object -Descending StartTime | select -Property Id,Name,StartTime

### Maak gebruik van de inhoud van de variabele “$myProces” om de calculator processen te tonen. Voeg een Cmdlet toe om het calculator proces te sluiten.

## Script: Rekenen.ps1

### Maak een script dat 2 getallen (kommagetal mag) vraagt om in te geven en vervolgens de volgende berekeningen uitvoert. Toon 2 cijfers na de komma bij de berekening.

**Uitvoer:**

### Afbeelding met tekst Automatisch gegenereerde beschrijving

**Oplossing:**

# Vraag de gebruiker om twee kommagetallen in te voeren

$nummer1 = Read-Host "Voer het eerste kommagetal in"

$nummer2 = Read-Host "Voer het tweede kommagetal in"

# Voer berekeningen uit en toon resultaten met 2 cijfers na de komma

$optelling = [math]::Round(($nummer1 + $nummer2), 2)

$aftrekking = [math]::Round(($nummer1 - $nummer2), 2)

(vermenigvuldiging = [math]::Round(($nummer1 \* $nummer2), 2))

(deling = [math]::Round(($nummer1 / $nummer2), 2))

# Toon de resultaten

Write-Host "Optelling: $optelling"

Write-Host "Aftrekking: $aftrekking"

Write-Host "Vermenigvuldiging: $vermenigvuldiging"

Write-Host "Deling: $deling"

## Script: MakeBackup.ps1 Maak de volgende mappen aan: c:\pxl\_data (plaats enkele bestanden in deze map) c:\backups

Het script maakt een back-up (Compress-Archive) van een bron-map en plaats deze in de doel-map.

Na het opstarten van het script (.\MakeBackup.ps1) wordt er gevraagd naar een bron-map (opgeslaglocatie van de te back-uppen bestanden) en vervolgens naar de doel-map (opslaglocatie van de back-up).

Er wordt ook telkens getoond of de map al dan niet bestaat (enkel tonen, geen gevolg aan koppelen)

De bestandsnaam (back-up bestand) heeft de volgende opmaak.  
 20221023041419\_mydata.zip (jaar maand dag uur minuut sec \_ backupnaam.zip)

**Output:**

Afbeelding met tekst

Automatisch gegenereerde beschrijving

**Oplossing:**

## Script: AddUser.ps1 (extra)

## Maak een script dat een lokale gebruiker aanmaakt. Bij de uitvoer van het script wordt er gevraag naar de gebruikersnaam en het wachtwoord (encryptie).

**Tip: New-LocalUser**

**Output:**

## Afbeelding met tekst Automatisch gegenereerde beschrijving

**Je kan nadien de gebruiker eenvoudig verwijderen met het Cmdlet   
 Remove-LocalUser <gebruikersnaam>  
 Get-LocalUser**

**Oplossing:**

**EINDE OEFENING**