

Powershell

8. Custom Objects

Computersystemen 2 - Windows

Overzicht

- Custom Objects
- Oefeningen

Custom Objects

- Hash Table (key/value pair)
 - `$person = @{Name="Bob"; Age=22}`
 - `$person["Name"]`
 - `$person["Age"]`
- Custom Objects
 - `[array] $personlist = $null`
 - `$person1 = [pscustomobject]@{Name="Bob"; Age=22}`
 - `$person1.Name`
 - `$personlist += $person1`
 - `$person2 = [pscustomobject]@{Name="Ann"; Age=20}`
 - `$personlist += $person2`
 - `$personlist`

Overzicht

- Custom Objects
- Oefeningen

Oefeningen 8

1. Maak fruit.csv aan met volgende inhoud

Fruit,Aantal
Appels,10
Peren,8
Kersen,5
Bananen,23

- Importeer deze file in de variabele *\$fruit*. Kan je deze sorteren op de property *aantal*? Waarom (niet)?
- Zet *[array] \$CopiedFruit = \$null* . Kopieer de lijst *\$fruit* in deze lijst van custom objecten. Maak gebruik van *Foreach-Object*. Converteer hierbij *aantal* naar integer. Kan je deze nu sorteren op de property *aantal*?

Oefeningen 8

2. Schrijf een powershell script Gemiddelde.ps1. Het script bevat 2 functies en roept deze dan na elkaar op:

a. functie 1: Lees-Punten:

- vraagt aantal vakken en aantal studenten
- leest punten in voor alle vakken en alle studenten
- schrijft deze weg in file met formaat:

```
vak1,punt1,punt2,punt3
```

```
vak2,punt1,punt2,punt3
```

b. functie 2: Schrijf-Gemiddelde:

- leest file aangemaakt in LeesPunten in
- schrijft gemiddelde per vak op het scherm

Kan je gebruik maken van Export-Csv en Import-Csv?
Waarom (niet)?



Oefeningen 8

3. Start je Gitea server. Schrijf een Powershell script genaamd "Check-Site.ps1"
 - a. Dat elke 2 sec op je host machine checkt:
 - of je Windows Server up-and-running is (reageert op ICMP)
 - of je website up-and-running
 - b. Schrijf telkens als output een lijn in een file "\$logfile" met:
 - SEVER UP/DOWN, de huidige tijd en de naam van de server
 - WEB UP/DOWN, de huidige tijd en de naam van de website
 - c. Zoek hiervoor op welke Cmdlet je nodig hebt voor
 - pingen
 - http request



Oefeningen 8

4. Schrijf een Powershell script genaamd "Overtredingen.ps1"
 - a. Lees file *a_overtredingen.csv* in (zie Canvas) in de lijst *\$overtredingen*. Check op fouten.
 - b. Schrijf functie *More-Overtredingen* (*\$overtredingen*, *\$aantal*)
 - Deze geeft een lijst terug met enkel de properties *datum_vaststelling*, *opnameplaats_straat*, *aantal_passanten* en *aantal_overtredingen_roodlicht* van items waar het aantal overtredingen \geq *\$aantal*.
 - c. Schrijf functie *Get-Streets* (*\$overtredingen*)
 - Deze geeft een unieke lijst terug met alle straten in de csv file
 - d. Schrijf functie *Sum-Overtredingen* (*\$overtredingen*, *\$straat*)
 - Deze geeft het totale aantal overtredingen in een straat (int)

Oefeningen 8

4. Schrijf een Powershell script genaamd "Overtredingen.ps1"
 - e. Schrijf functie *All-Overtredingen (\$overtredingen)*
 - Deze geeft een lijst terug met alle straten en de som van het aantal overtredingen voor deze straat(maak gebruik van vorige functies)
 - Extra: kan je deze ook sorteren (straat met meeste overtredingen eerst)?

Oefeningen 8

5. Een bedrijf wil een inventaris maken van alle MAC-adressen van alle computers. Schrijf hiervoor een Powershell script "Get-Mac.ps1" die MAC-adressen van een computer opvraagt en stockeert in een file "MacTable.csv"
- Vervolledig de functie Delete-LinesCsv die alle lijnen uit een CSV file verwijdert waar een bepaalde \$property gelijk is aan een bepaalde \$value
 - Schrijf de functie Write-Mac, die alle MAC-adressen van fysieke adapters van type Ethernet (enkel deze!) van een computer ophaalt.
 - Je kan hiervoor gebruik maken van "Get-CimInstance Win32_NetworkAdapter"
 - Deze worden gestockeerd in MacTable.csv (zie file op Canvas), samen met de naam van de adapter, de naam van de computer en een timestamp (dit is datum en uur waarop de informatie opgehaald werd).
 - Indien het MAC-adres reeds bestaat in de file wordt deze informatie overschreven (of m.a.w. de bestaande lijn wordt ge-delete). Maak hiervoor gebruik van de functie Delete-LinesCsv. Dus: elk MAC-adres mag slechts één maal voorkomen in de file. PS: alle velden zijn van het type string, deze moeten niet noodzakelijk tussen quotes staan.

Oefeningen 8

5. Een bedrijf wil een inventaris maken van alle MAC-adressen van alle computers. Schrijf hiervoor een Powershell script "Get-Mac.ps1" die MAC-adressen van een computer opvraagt en stockeert in een file "MacTable.csv"
- c. Schrijf de functie Get-Mac, die alle lijnen uit de csv-file op het scherm toont waarvoor de computernaam aan een bepaald patroon voldoet, bv. "KDG*"
 - d. Schrijf de functie Clean-Mac, die alle lijnen uit de file verwijderd die ouder zijn dan \$maxtimespan dagen. Tip: je kan met Get-Date "20/12/2017 15:27:48" een string omzetten naar een datum object.
 - e. Schrijf het hoofdprogramma:
 - Indien geen parameters meegegeven worden aan Get-Mac wordt de functie Write-Mac uitgevoerd.
 - Indien als 1ste parameter -show meegegeven worden, wordt de functie Get-Mac aangeroepen (de 2de parameter is het patroon voor de computernaam). Bv: Get-Mac.ps1 -show KDG*
 - Indien als 1ste parameter -clean meegegeven worden, wordt Clean-Mac aangeroepen (de 2de parameter is de maximale timespan). Bv: Get-Mac.ps1 -clean 10