

## Übung 2.1.3

- Starten Sie auf Ihrem Windows Server 2022 System die Windows PowerShell Konsole
- Überlegen Sie kurz, wann die Windows PowerShell (x86) benutzt werden sollte, wählen Sie aber die 64-Bit Variante
- Benutzen Sie ein Cmdlet, das Ihnen die installierte PowerShell Version anzeigt
- Interpretieren Sie die Ausgabe
- Benutzen Sie die Autoausfüllfunktion, indem Sie eingeben
  - `$PS` und dann die Tab-Taste drücken
- Öffnen Sie nach Belieben Cmdlets, die Ihnen angezeigt werden und interpretieren Sie diese
- Öffnen Sie eine Windows PowerShell ISE
- Tippen Sie dort ein „`$`“
- Welche Hilfe erhalten Sie?
- Was passiert, wenn Sie das Cmdlet „`$PSVersionTable`“ eingeben
- Ab wann wird nur noch „`$PSVersionTable`“ angezeigt?

## Lösung 2.1.3

### PowerShell starten

- Linke Maustaste auf „Start“
- Herunterscrollen bis zu „Windows PowerShell“
- Klicken Sie auf den kleinen Pfeil nach unten rechts daneben
- Sie sehen vier Varianten:
  - Windows PowerShell
  - Windows PowerShell (x86)
  - Windows PowerShell ISE
  - Windows PowerShell ISE (x86)
- Die x86-Variante ist nur für die Abwärtskompatibilität und wird nur selten verwendet
- Auswahl „Windows PowerShell“

### Anzeigen der Version

- `$PSVersionTable`

### Interpretation

- `PSVersion`: Version der PowerShell, hier 5.1
- `PSEdition`: Edition der PowerShell, hier die Desktop-Variante
- `PSCompatibleVersions`: Anzeige der Abwärtskompatibilität
- `BuildVersion`: Genaue aktuelle Version, abhängig vom Patchstand
- `WSManStackVersion`: Version des Web-Service Managers
- `PSRemotingProtocolVersion`: Die Version des Windows PowerShell remote Management Protokolls
- `SerializationVersion`: Versionsnummer der Serialisation. Dies ist Teil der Klasseneinteilung der Programmierung

## Autoausfüllfunktion

- z.B.
  - \$PS
  - 4 x Tab
  - Enter
- Sie haben das Cmdlet „\$PSCulture“ geöffnet, das Ihnen die Spracheinstellung ausgibt
- Versuchen Sie noch einige Cmdlets

## Öffnen der Windows PowerShell ISE

- Linke Maustaste auf „Start“
- Herunterscrollen bis zu „Windows PowerShell“
- Klicken Sie auf den kleinen Pfeil nach unten rechts daneben
- Auswahl „Windows PowerShell ISE“

## Eintippen von „\$“

- Ich erhalte alle cmdlets, die mit „\$“ beginnen

## Eintippen der weiteren Buchstaben des Cmdlets

- Ich erhalte eine immer kleiner werdende Auswahl an Cmdlets, die mit der eingegebenen Buchstabenkombination beginnt

## Ab wann wird nur noch „\$PSVersionTable“ angezeigt?

- Ab „\$PSV“

## Übung 2.2.1

- Wie ist der prinzipielle Aufbau eines Cmdlets?
- Zeigen Sie alle Dienste an, die mit dem Buchstaben „P“ beginnen
- Interpretieren Sie die Ausgabe

## Lösung 2.2.1

### Wie ist der prinzipielle Aufbau eines Cmdlets?

- Verb-Substantiv –Parameter

### Zeigen Sie alle Dienste an, die mit dem Buchstaben „P“ beginnen

- Get-Service –Name „P\*“

### Interpretation der Ausgabe

- Spalte Status: Dienst gestartet oder gestoppt
- Spalte Name: Systemname des Diensts
- Spalte Displayname: Anzeigename des Diensts

## Übung 2.3

- Lassen Sie sich in einer PowerShell Konsole alle Verben anzeigen, die mit dem Buchstaben „A“ beginnen

- Hinweis: Der benötigte Parameter lautet „-Verb“

### Lösung 2.3

- Get-Verb –Verb „a\*“

### Übung 2.4

- Suchen Sie in einer PowerShell Konsole nach allen Cmdlets, die mit dem Verb „Get“ beginnen
- Suchen Sie in einer PowerShell Konsole nach allen Cmdlets, die mit dem Verb „Get“ beginnen und einen Bezug zur Speicherverwaltung haben
  - Hinweis: Cmdlets mit Bezug auf die Speicherverwaltung haben die Zeichenkette „Memory“ im Substantiv
- Wie beurteilen Sie die Schreibweise der Substantive
  - Windowerrorreporting
  - WindowsErrorReporting

### Lösung 2.4

#### Alle Cmdlets, die mit dem Verb „Get“ beginnen

- Get-Command –Verb Get

#### Alle Cmdlets, die mit dem Verb „Get“ beginnen, mit Bezug zur Speicherverwaltung

- Get-Command –Verb Get –noun \*memory\*

#### Beurteilung der Schreibweise

- Beide Schreibweisen sind korrekt, aber es ist übersichtlicher, jeweils den ersten Buchstaben eines Wortes groß zu schreiben und bei zusammengesetzten Wörtern den ersten Buchstaben des zweiten Wortes ebenfalls

### Übung 2.5.1a

- Was zeigen Sie mit dem Cmdlet „Get-Service –Name Win\*“ an?
- Führen Sie dieses Cmdlet aus
- Wie müssen Sie es verändern, wenn Sie aus dieser Auswahl den Dienst „WinRM“ **nicht** sehen möchten?
- Führen Sie dieses Cmdlet aus
- Müssen die Parameter in einer bestimmten Reihenfolge sein? Probieren Sie es aus
- Was bedeutet das?

### Lösung 2.5.1a

#### Was zeigen Sie mit dem Cmdlet „Get-Service –Name Win\*“ an?

- Alle Dienste, die mit der Zeichenkette „Win“ beginnen

#### Keine Anzeige von WinRM

- Get-Service –Name Win\* -Exclude WinRM

### Müssen die Parameter in einer bestimmten Reihenfolge sein?

- Nein

### Warum?

- Es sind benannte Parameter, die in willkürlicher Reihenfolge stehen können

## Übung 2.5.1b

- Geben Sie ein „Get-Service Win\*“
  - Funktioniert dieses Cmdlet auch ohne Angabe des Parameters „Name“?
  - Warum?
- Sie möchten erfahren, ob die Parameter „Name“ und „Exclude“ benannte Parameter sind
  - Wie lauten die Cmdlets dafür?
- Welche wichtige Aussage erkennen Sie in der Ausgabe?

## Lösung 2.5.1b

- Ja, es funktioniert, da „Name“ ein Positionsparameter ist

### Überprüfung, ob benannte Parameter

- Get-Help -Name Get-Service -Parameter Name
- Get-Help -Name Get-Service -Parameter Exclude

### Aussage der Ausgabe

- Parameter Exclude:
  - Position „Benannt“ heißt, dies ist ein benannter Parameter, der immer in der Form
    - -Parameter Argument
  - eingegeben werden muss, die Position ist aber egal
- Parameter „Name“
  - Position 0 heißt, dies ist ein Positionsparameter, der an einer beliebigen Position stehen kann

## Übung 2.6

- Sie möchten Hilfe über das Cmdlet „Get-TimeZone“ erhalten
  - Wie lautet das Cmdlet? Benutzen Sie es

## Lösung 2.6

- Get-Help -Name Get-TimeZone