

## Übungen für den ersten Tag

Auf den folgenden Seiten werden kleine Aufgaben, ergänzt durch Fragen, zusammengestellt, die im Rahmen einer Übung am Nachmittag die Gelegenheit bieten sollen, sich in Ruhe mit den Grundlagen der PowerShell-Syntax beschäftigen zu können.

Letzte Aktualisierung: 05/09/2020

Die „Lernpunkte“ fassen das zusammen, was im Rahmen einer Übung gelernt werden soll.

**Wichtig:** Die Fragen sind natürlich kein Test – wenn man eine Antwort gar nicht weiß, bitte einfach das Feld leer lassen oder den Dozenten fragen (man findet die Antwort natürlich auch im Internet☺). Außerdem muss man die Aufgaben nicht von Anfang an der Reihe nach durchgehen. Das Ziel ist es, die PowerShell-Syntax an einer vorgegebenen Aufgabenstellung anwenden zu können.

### Übung 1: Die Schreibweise von Befehlen herausfinden

**Ziel:** Die Syntax (Schreibweise) von drei PowerShell-Befehlen herausfinden

1) Wie ist die „Schreibweise“ der Cmdlets (Befehle) Stop-Computer und Restart-Computer?

-----  
-----

2) Lässt sich anhand der Syntax erkennen wodurch sich die beiden Befehle unterscheiden?

-----  
-----

3) Warum wird die Syntax bei Restart-Computer zwei Mal angezeigt?

-----  
-----

## Übung 2: Die Schreibweise von Befehlen kennenlernen

**Ziel:** Kennenlernen der Befehlsschreibweise (Syntax) bei PowerShell-Befehlen

1) Was macht der Befehl Restore-Computer?

---

---

2) Wie viele Parameter besitzt der Befehl Restore-Computer? (Tipp: Die Parameter Whatif und Confirm sowie die Common Parameters zählen nicht, da sie allgemeine Parameter sind)

---

---

## Übung 3: Cmdlets (Befehle) und ihre Parameter

1) Warum besitzt der Befehl Restore-Computer nur einen einzigen Parameter?

---

---

**Lernpunkt:** Bei der PowerShell können ähnlich geschriebene Befehle eine komplett andere Bedeutung besitzen.

2) Die ISE zeigt über das Befehls-Add-On nach Eingabe von „Computer“ alle Befehle mit „Computer“ im Namen an – wie erhält man diese Liste in der Konsole (Tipp: Get-Command)

3) Eine allgemeine Frage zu den Parametern: Der Umstand, dass die Namen der Parameter in der Regel in eckige Klammern gesetzt werden, besitzt einen Grund. Welchen?

---

---

4) Einige Parameternamen sind in der Syntaxbeschreibung in zwei eckige Klammern gesetzt. Was hat das für den Parameter zu bedeuten?

---

---

**Lernpunkt:** Wird der Name eines Parameters von einem paar eckiger Klammern eingerahmt, ist der Parameter im Allgemeinen ein Pflichtparameter und ein Positionsparameter, so dass nur ein Wert, aber nicht der Name des Parameters angegeben werden muss.

**Lernpunkt:** Wird der Name eines Parameters von keinen eckigen Klammern eingerahmt, ist der Parameter kein Pflichtparameter, dessen Name angegeben werden muss.

**Lernpunkt:** Wird der Name eines Parameters von zwei Paar eckiger Klammern eingerahmt, ist der Parameter ein optionaler Positionsparameter.

## Übung 4: Verwenden der Pipeline und des Pipe-Operators

- 1) Ein Get-Service gibt die Eckdaten zu allen vorhandenen Systemdiensten aus. Ein Systemdienst kann ausführen oder nicht. Dies erkennt man an dem Wert der Eigenschaft „Status“. Ein Status = „Running“ bedeutet, dass der Dienst ausführt, ein Status = „Stopped“ bedeutet, dass der Systemdienst aktuell nicht ausführt (in der Regel wurde der Dienst nicht gestartet)
- 2) Wie lautet ein Aufruf von Get-Service, der nur die Systemdienste anzeigt, die nicht ausführen? Tipp: Der anzuhängende Befehl ist **Where-Object**.
- 3) Wie lautet ein Aufruf von Get-Service, der nur die Systemdienste anzeigt, die ausführen? Tipp: Anstelle von -eq einfach -ne verwenden.
- 4) Die Ausgabe soll in eine Datei umgeleitet werden.
- 5) Wie Aufgabe 4, nur sollen nur die Namen der Systemdienste in die Datei geschrieben werden. **Tipp:** Auf den Befehl Where-Object muss ein weiterer Befehl folgen: **Select-Object**. Über ihn werden die anzuzeigenden Eigenschaften festgelegt. In diesem Fall soll nur die Name-Eigenschaft ausgegeben werden.

**Tipp:** Möchte man eine Ausgabe ohne „Überschrift“, muss beim Select-Object-Befehl die Eigenschaft (in diesem Fall die Eigenschaft Name) über den Parameter **ExpandProperty** ausgewählt werden.

## Übungen 5: Zählen von „Gegenständen“

- 1) Wie viele Cmdlets gibt es in der aktuellen PowerShell-Umgebung? **Tipp:** Get-Command, CommandType-Parameter nicht vergessen und gezählt wird am einfachsten durch Setzen des Befehls in runde Klammern und anhängen eines „.count“.

-----

- 2) Wie viele Cmdlets gibt es, die mit dem Verb „get“ anfangen? Es geht nur um die Cmdlets.

-----

- 3) Falls ein AD zur Verfügung steht: Wie viele Computerkonten gibt es in der Domäne?

-----

## Übung 6: Umgang mit Platzhaltern in Parameterwerten

1) Gesucht ist ein Aufruf von Get-Service, der alle Systemdienste ausgibt, deren Name mit „S“ beginnt.

---

---

2) Gesucht ist ein Aufruf von Get-Service, der alle Systemdienste ausgibt, deren Beschreibung mit „S“ beginnt.

---

---

3) Gesucht ist ein Aufruf von Get-Service, der alle Systemdienste ausgibt, in deren Name ein „Sys“ enthalten ist.

---

---

4) Gesucht ist ein Aufruf von Get-Service, der alle Systemdienste ausgibt, deren Name mit „n“ beginnt und im Namen ein „svc“ enthalten ist.

---

---

5) Gesucht ist ein Aufruf von Get-Service, der alle Systemdienste ausgibt, deren Name mit „S“, „T, oder „W“ beginnt und mit „svc“ endet.

---

---

\*\*\* Ende der Übungen für den 1. Tag \*\*\*