

04_Powershell

Modul 122

Abläufe mit einer Scriptsprache automatisieren

Mitteilungen

19.03.2023

> Heute keine Mitteilungen

Themen

- > Prüfungsbesprechung
- > Navigation, Ordnerstruktur
- > Verzeichnis- und Datenmanagement
- > Piping
- > Parameter

Handlungsnotwendige Kenntnisse

HANOK	Kenntnisse
2.1	Kennt grundlegende Funktionalitäten der eingesetzten Scriptsprache.

Navigation, Ordnerstruktur

- > Inhalt eines Verzeichnisses anzeigen
 - > Befehl: `ls, dir`
 - > Um versteckte Ordner und Verzeichnisse auszugeben, kann der Parameter `-a` mitgegeben werden: `ls -a`
 - > Gibt Dateien und Verzeichnisse aus
- > Verzeichnis wechseln
 - > Befehl: `cd`
 - > In Unterordner "example" wechseln: `cd example`
 - > In Übergeordneter Ordner wechseln: `cd ..`
 - > Befehl um Laufwerk wechseln (Windows), Bsp. Laufwerk D: `D:`

Navigation, Ordnerstruktur

- > Root Directory
 - > Dargestellt als (Unix): /
 - > Dargestellt als (Windows): C:\
 - > Das oberste Verzeichnis
 - > Beinhaltet Daten und Verzeichnisse zum Betriebssystem, Programmen, etc.
- > Home Directory
 - > Dargestellt als (Unix): ~
 - > Dargestellt als (Windows): C:\Users\<USER>
 - > Verzeichnis zu den User-Daten
 - > Beinhaltet persönliche Einstellungen, Daten, etc.
- > Current Directory
 - > Dargestellt als: .

Navigation, Ordnerstruktur

Übung

10'

- > Lass dir im Terminal den Inhalt deines Download-Ordners ausgeben
 - > Wie lauten die Befehle? `"cd Downloads"` und dann `"ls"`
 - > Gibt es mehrere Möglichkeiten, um diese Aufgabe zu lösen?

- > Wie kann auf das Root-Verzeichnis navigiert werden?
 - > Wie lauten die Befehle? `cd /`
 - > Gibt es mehrere Möglichkeiten, um diese Aufgabe zu lösen?

- > Wie kann auf das Home-Verzeichnis navigiert werden?
 - > Wie lauten die Befehle? `cd`
 - > Gibt es mehrere Möglichkeiten, um diese Aufgabe zu lösen?

Verzeichnis- und Datenmanagement

- > Datei erstellen
 - > Befehl (Bash): `touch <file>`
 - > Befehl (PS): `new-item <file>`
- > Datei löschen
 - > Befehl: `rm <file>`
- > Verzeichnis erstellen
 - > Befehl: `mkdir <folder>`
- > Verzeichnis löschen
 - > Befehl (Inhalt wird gelöscht): `rm -r <folder>`
 - > Befehl (funktioniert nur bei leeren Verzeichnissen): `rmdir <folder>`

Verzeichnis- und Datenmanagement

- > Datei / Verzeichnis verschieben / umbenennen
 - > Befehl: `mv <file/folder> <destination>`
- > Datei / Verzeichnis kopieren
 - > Befehl: `cp <file/folder> <destination>`
- > Dateiinhalt anzeigen
 - > Befehl: `cat <file>`

Verzeichnis- und Datenmanagement Übung

15'

- > Erstelle mit einem Skript diese Verzeichnisstruktur

```

├── admin
├── README.md
└── src
    ├── bin
    └── test
    
```

- > README.md ist eine Datei, alles andere sind Verzeichnisse
- > Navigiere in dein Download Verzeichnis und verschiebe die erstellte Verzeichnisstruktur in das aktuelle Verzeichnis.
- > Entferne die Verzeichnisstruktur mit einem Befehl.

Piping

- Mit Piping können Resultate zwischen Commands weitergegeben werden.
 - Befehl: |
 - Beispiel (Bash): `ls | grep Downloads`
 - Beispiel (PS): `ls | Select-String Downloads`

Piping Übung

5'

- > Sortiere eine Ausgabe
 - > Erstelle eine Datei "data", in welcher auf unterschiedlichen Zeilen eine zufällige, einstellige Zahl steht.
 - > Gib den Inhalt der Datei "data" sortiert aus.
Tipp: Verwende zur Lösung den Befehl sort.

Parameter

- > Über Parameter kann das Verhalten eines Commands beeinflusst werden.
- > Parameter werden nach dem Befehl eingegeben.
 - > Beispiel bestimmt der Parameter (IP Adresse) nach dem Ping-Befehl, auf welche Adresse der Ping gesendet wird: `ping 8.8.8.8`
 - > Es können mehrere Parameter eingegeben werden
 - > Bash: `ping 8.8.8.8 -c 2`
 - > PS: `ping 8.8.8.8 -n 2`

Parameter. Benannt

Benannte Parameter

Benannte Parameter sind Parameter, die Sie mit ihrem Namen angeben müssen. Nach einem Leerzeichen folgt dann zwingend der Wert für den Parameter. Die allgemeine Syntax sieht wie folgt aus:

```
<Cmdlet> -Parameter Wert
```

Beispiel

```
Get-Command -Noun service ↵
```

Es werden alle Befehle ausgegeben, die das Substantiv `service` verwenden, also alle Befehle rund um *Dienste*.

Parameter. Switch

Switch-Parameter

Ein Switch-Parameter ist ein Schalter, der ohne Wert auskommt. Wird der Switch-Parameter angegeben, ist er aktiv. Wird er nicht angegeben, ist er inaktiv. Die allgemeine Syntax ist einfacher als bei benannten Parametern:

```
<Cmdlet> -Parameter
```

Beispiel

```
Get-ChildItem -Recurse ↵
```

Mithilfe des Cmdlets `Get-ChildItem` wird der Inhalt des aktuellen Verzeichnisses angezeigt. Der Switch-Parameter `-Recurse` sorgt dafür, dass ebenfalls der Inhalt aller Unterverzeichnisse angezeigt wird.

Parameter. Position

Positionsparameter sind als schnelle Lösung für eine direkte Eingabe erfahrener Anwender gedacht. Lernen Sie den Umgang mit der PowerShell, sollten Sie die Befehle immer komplett eingeben. In allen anderen Fällen, z. B. in Skripten oder bei Arbeiten im Team, empfiehlt sich die komplette Schreibweise.

Positionsparameter sind wie folgt zu notieren:

```
<Cmdlet> [-Parameter] Wert
```


Beispiele

```
Get-Process [-Name] powershell ↵
```

Das Cmdlet `Get-Process` ruft Informationen zu laufenden Prozessen ab, hier zum Prozess *powershell*, der auf Ihrem Computer ausgeführt wird. Der Positionsparameter `-Name` ist optional, es kann auch einzig der Wert *powershell* angegeben werden.

```
Copy-Item [-Path] c:\alt [-Destination] c:\neu ↵
```

`Copy-Item` kopiert ein Element, hier das Verzeichnis *C:\alt* nach *C:\neu*. Das Cmdlet besitzt zwei Positionsparameter `-Path` und `-Destination` für die Angabe der Quelle und des Ziels. Eine verkürzte Schreibweise `Copy-Item c:\alt c:\neu` ist möglich.

Im Folgenden wird darauf verzichtet, bei jedem Beispiel explizit darauf hinzuweisen, dass die Befehle mit der Taste  abgeschlossen werden. Erst nach Betätigung dieser Taste erfolgt die Verarbeitung durch die PowerShell.

Parameter. Allgemein

Allgemeine Parameter		
Parameter	Typ	Beschreibung
-Verbose	S	Zeigt sehr ausführliche Informationen zu dem Vorgang an, der mit dem Befehl ausgeführt wird
-Debug	S	Zeigt zusätzlich Informationen für Programmierer, wenn Warnungen und Fehler bei dem ausgeführten Vorgang auftreten
-WarningAction	N	Bestimmt das Verhalten des Cmdlets, falls eine Warnung auftritt. Gültige Werte sind: ✓ <i>SilentlyContinue</i> : Warnung unterdrücken, Ausführung fortsetzen ✓ <i>Continue</i> : Warnung ausgeben, Ausführung fortsetzen ✓ <i>Inquire</i> : Warnung ausgeben, weitere Ausführung nachfragen ✓ <i>Stop</i> : Warnung ausgeben, Ausführung beenden
-WarningVariable	N	Speichert auftretende Warnmeldungen in der angegebenen Variablen
-ErrorAction	N	Bestimmt das Verhalten des Cmdlets, falls ein Fehler auftritt. Gültige Werte sind: ✓ <i>SilentlyContinue</i> : Warnung unterdrücken, Ausführung fortsetzen ✓ <i>Continue</i> : Warnung ausgeben, Ausführung fortsetzen ✓ <i>Inquire</i> : Warnung ausgeben, weitere Ausführung nachfragen ✓ <i>Stop</i> : Warnung ausgeben, Ausführung beenden
-ErrorVariable	N	Speichert auftretende Fehlermeldungen in der angegebenen Variablen
-OutVariable	N	Zeigt die Ausgabeobjekte des Befehls an und speichert sie in der angegebenen Variablen

Parameter Übung

5'

- > Wie Commands zu verwenden sind, ist meistens (besser oder schlechter) dokumentiert. Um die Anleitung – Manpage genannt – eines Befehls anzuzeigen, kann der Command man genutzt werden: `man ls`
 - > Finde mit Hilfe der Manpage heraus, wie mit `ls` zusätzlich alle Unterdateien und Unterverzeichnisse ausgegeben werden. `ls -R`
- > Dokumentiere 4 Parameter, welche für den Command `rm` verwendet werden können.

Für den `rm`-Befehl in PowerShell gibt es vier nützliche Parameter:

- Path: Mit diesem Parameter kann der Pfad zur zu löschenden Datei oder zum Verzeichnis angegeben werden.
 - Recurse: Verwenden Sie diesen Parameter, um das Löschen von Verzeichnissen und deren Inhalten zu erzwingen.
 - Force: Wenn Sie diesen Parameter verwenden, werden alle Dateien und Verzeichnisse ohne Bestätigung gelöscht.
 - WhatIf: Mit diesem Parameter können Sie sehen, welche Dateien und Verzeichnisse gelöscht würden, wenn der Befehl ausgeführt wird, ohne tatsächlich Dateien zu löschen.
- Dies kann nützlich sein, um zu überprüfen, ob der Befehl die gewünschten Dateien löscht, bevor er tatsächlich ausgeführt wird.

Vertiefungsaufgabe

- > Führe die Übungen unter einem anderen Betriebssystem durch.
 - > Wo sind Unterschiede? Was ist identisch? – Halte deine Erkenntnisse schriftlich fest.

Hausaufgaben

> Fragen Sie Ihre Lehrperson

Quellen

- > BBB. (2022). M122-master

Vielen Dank für Ihre Aufmerksamkeit!

KANTON
LUZERN



**Berufsbildungszentrum Wirtschaft,
Informatik und Technik BBZW**

www.bbzw.lu.ch