

# Corona

Montag, 23. März 2020 07:48

Siehe auch

[https://github.com/steinam/2019\\_10FI5](https://github.com/steinam/2019_10FI5)

steinam  
#2974

Aus <<https://discordapp.com/channels/@me>>

## 10FI5 Wochenplan Blockwoche 09

Montag	Dienstag	Mittwoch	Donnerstag	Freitag
9.00 Uhr – 10.00 Uhr VK Infos & Orga Klassenleitung HIN + Deutsch: Projektmanagement	08:30 – 10:00 STE: AWP Arbeit an Dateisystem		NEU: 08:30 – 09:30 VK VSY LAM Verkabelungsarten	08:30 – 10:00 STE: AWP: Dateisystem
11:00 Uhr – 11.30 Uhr VK VSY LAM Kick Off				10:00 – 11:00 Uhr VK VSY LAM 10FI5 Verkabelungsarten Auswahl Verk.
12:00 – 12:30 VK AWP STE			13:30 – 15:00 ITS Besprechung Zilles	
12:30 – 13:30 ITS Zeichencodes und Präfixe Zilles				

# C\_Arbeit mit Dateisystem

Donnerstag, 19. März 2020 08:00

[PowerShell Dateien und Ordner verwalten](#)

## 1 Arbeiten im Dateisystem

Das Arbeiten mit dem Dateisystem kann für Systemintegratoren eine wichtige Aufgabe sein. Nachfolgende Tabelle vermittelt einen Überblick über die wichtigsten Befehle zum Verwalten des Dateisystems in Powershell.

Tabelle 31.1: Wichtige Commandlets für die Arbeit mit dem Dateisystem

PowerShell-Commandlet	PowerShell-Alias	Befehl Klassische Windows-Kommandozeile	Befehl Unix „sh“	Beschreibung
Clear-Item	Cli	–	–	Inhalt leeren
Copy-Item	Cpi, cpp, cp, copy	Copy	Cp	Kopieren von Elementen
Get-Content	Gc	Type	Cat	Holt den Inhalt
Get-Location	Gl, pwd	Pwd	Pwd	Holt das aktuelle Verzeichnis
Move-Item	Mi, move, mv, ml	Move	Mv	Bewegen von Elementen
New-Item	Ni (Funktion md)	–	–	Element anlegen
Remove-Item	Ri, rp, rm, rmdir, del, erase, rd	del, rd	rm, rmdir	Löschen von Elementen
Rename-Item	Rni, ren	Rn	Ren	Umbenennen eines Elements
Set-Content	Sc	(Umleitungen >)	(Umleitungen >)	Festlegen des Inhalts
Set-Item	Si	–	–	Inhalt festlegen
Set-Location	Sl, cd, chdir	cd, chdir	cd, chdir	Setzt das aktuelle Verzeichnis

Test-Path

### Arbeitsauftrag 1

Erläutern Sie das Cmdlet Get-Item (oben nicht aufgeführt). Worin besteht der Unterschied zu Get-Childitem?

### Arbeitsauftrag 2

Bereiten Sie eine kurze Präsentation vor, in der Sie Ihre Mitschüler die von Ihnen gewählte Funktion kurz erklären. Erstellen Sie hierzu eine Powerpointfolie, die die wesentlichen Informationen zu Ihrem Thema zusammenfasst. Überlegen Sie sich zudem eine kleine Übungsaufgabe. Bearbeitungszeit: 35 Minuten, Präsentations- und Übungszeit: 5-10 Minuten. Nutzen Sie für Ihre Recherche folgende Internetseite: <http://technet.microsoft.com/de-de/scriptcenter/dd772285.aspx>

Folgende Funktionen stehen zur Auswahl:

[Copy Files or Folders](#)

[Create a New File or Folder](#)

[Delete a File or Folder \(Or Other Objects\)](#)

[Move a File or Folder](#)

[Rename a File or Folder](#)

[Retrieve a Specific Item](#)

[Verify the Existence of a File or Folder](#)

`get-childitem $env:windir`

`get-item $env:windir`

**Arbeitsauftrag (Alle Aufgaben sind als Funktion zu codieren)!**

1. Löschen Sie eine von Ihnen zuvor erstellte Datei Ihrer Wahl.

2. Erstellen Sie einen Ordner mit mehreren Textdateien mit unterschiedlichen, frei wählbaren Inhalten. Speichern Sie auch eine Textdatei ohne Inhalt. Schreiben Sie anschließend ein Skript, das alle Textdateien löscht, deren Größe gleich 0KB ist.

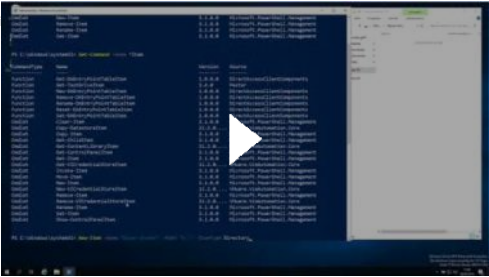
3. Prüfen Sie anhand eines Skripts, ob es im Verzeichnis von Aufgabe 2 eine Datei mit dem Namen "report.txt" gibt. Sofern nicht, legen Sie diese an. Sofern diese bereits existiert, löschen Sie diese und legen Sie die Datei danach dann "frisch" neu an.

4. Kopieren Sie alle vorhandenen Dateien mit der Endung „txt“ aus dem aktuellen Verzeichnis in ein Unterverzeichnis mit dem Namen „Textdateien“. Prüfen Sie auch hier wiederum, ob das entsprechende Unterverzeichnis bereits existiert und legen Sie dieses bei Bedarf zuvor noch an.

```
if((test-Path -Path "C:\test_10fi5\Textdateien") -eq $false)
{
    Write-Host "Neuer Ordner wird angelegt"
    New-Item -ItemType Directory "C:\test_10fi5\Textdateien" -Force
}
Copy-Item -Path "C:\test_10fi5\*.txt" -Destination "C:\test_10fi5\Textdateien"
Get-ChildItem -Path "C:\test_10fi5\Textdateien"

if(($container = test-Path -Path "C:\test_10fi5\Textdateien -pathType Container){
}

$container = test-Path -Path "C:\test_10fi5\Textdateien -pathType Container
```



```

if(($container = test-Path -Path "C:\test_10fi5\Textdateien -pathtype Container){

}

$container = test-Path -Path "C:\test_10fi5\Textdateien -pathtype Container

If($container-eq $true)

```

5. Erstellen Sie ein Backup-Skript, das bei jeder Ausführung einen Ordner erstellt, der das aktuelle Datum als Bestandteil seines Namens trägt. Lassen Sie im Rahmen dieses Skriptes alle Inhalte eines bestehenden anderen Ordners Ihrer Wahl dort als Kopie einfügen.

```

#Aufgabe 5
##Achtung test-Path muss in Klammern gesetzt werden, sonst eq ein Parameter von test-Path

[String]$aktuellesDatumOrdner = [string](Get-Date).Day+[string](Get-Date).Month+[string](Get-Date).Year
$aktuellesDatumOrdner+="Backup" # # get-date -format yyyy_MM_dd

##Ordner anlegen
# $false entspricht dem Zahlenwert 0
if ((test-Path "C:\test_10fi5\Documents\$aktuellesDatumOrdner") -eq $false)
{
    New-Item C:\test_10fi5\Documents\$aktuellesDatumOrdner -type directory
}

##test_10fi5dateien anlegen
if ((test_10fi5-Path "C:\test_10fi5\Documents\test_10fi5ordner\test_10fi5file2.txt") -eq 0) # Zahlenwert 0 entspricht $false
{
    New-Item C:\test_10fi5\Documents\test_10fi5ordner\test_10fi5file2.txt -type file
}

##Zwingendes Kopieren mit -force
Copy-Item -LiteralPath C:\test_10fi5\Documents\test_10fi5ordner\ -Destination C:\test_10fi5\Documents\$aktuellesDatumOrdner -force -Recurse
##test_10fi5
Get-ChildItem -Path C:\test_10fi5\Documents\$aktuellesDatumOrdner

# Datum schneller in die Variable:
$datum = get-date -Format ddMMyy

#$datum = get-date
#$datum2 = $datum.ToString("dd.MM.yy")
#$datum2

```

6. Stellen Sie sich folgendes Szenario vor: In einer Abteilung wird gemeinschaftlich ein Scanner genutzt. Der netzwerktaugliche Scanner speichert alle gescannten Dokumente in einem speziellen Ordner auf einem Netzwerklaufwerk ab. Unglücklicherweise vergessen verschiedene Kollegen, ihre Scanunterlagen nach der Nutzung zu löschen. Ihre Aufgabe ist es, ein Skript zu schreiben, mit dessen Hilfe alle Dateien in diesem Ordner gelöscht werden, die älter als 10 Tage sind.

Legen Sie zu diesem Zweck einen Ordner mit einer passenden Anzahl von Dokumenten an und schreiben Sie ein Skript, das die gewünschte Bereinigung vornimmt.

```

#Aufgabe 6
cls
$element = get-childitem "C:\test_10fi5\Documents\test_10fi5ordner"
$limit = (get-date).AddDays(-10)
foreach ($i in $element)
{
    if($i.lastwritetime -lt $limit)
    {
        remove-item $i.FullName
    }
}

#Löschen=Get-ChildItem C:\test_10fi5\Documents\test_10fi5ordner | Where-Object{$_ .LastWriteTime.Days -lt (Get-#Date).TotalDays+10} | remove-item

```

7. Ein Kollege hat sich mächtig über Ihr eigenmächtiges Vorgehen beschwert. Nach heftigen Diskussionen haben Sie vereinbart, vor dem Löschen dieser Datei, diese zu kopieren und in einen Archivierungsordner zu verschieben. Passen Sie Ihr Skript entsprechend an.

```

#Aufgabe 7
cls
$element = get-childitem "C:\test_10fi5\Documents\test_10fi5ordner"
$limit = (get-date).AddDays(-10)
foreach ($i in $element)
{
    if($i.lastwritetime -lt $limit)
    {
        Copy-Item $i.FullName -destination "C:\test_10fi5\Archivordner" -Force
        remove-item $i.FullName
    }
}

Remove-Item -Path "C:\test_10fi5\Documents\text3.txt"

```

Klasse:: Statische Methode

MessageBox.show() geht nicht, weil wir kein Objekt von MessageBox haben

8. Erweitern Sie das Skript, so dass der Inhalt des neu entstandenen Archivierungsordners jeweils am 31.12 eines Jahres gelöscht wird. Falls das aktuelle Datum nicht der 31.12. des Jahres ist, soll eine entsprechende Meldung erfolgen. Die Meldung erfolgt im Hilfe-Feld von MessageBox. Informieren Sie sich wie diese eingebunden werden kann, ohne eine Objekt der Klasse zu benötigen

```

cls
Class MessageBox
[string]$aktuellesDatum= get-date -Format dd.MM.
If($aktuellesDatum -eq "31.12.")
{
    New-Item -itemtype file -Path "C:\test_10fi5\Documents\Doc1.txt" -Value "hallo" -Force
    New-Item -itemtype file -Path "C:\test_10fi5\Documents\Doc2.txt" -Value "guten morgen" -Force
    New-Item -itemtype file -Path "C:\test_10fi5\Documents\Doc3.txt" -Value "" -Force
    $dateien = Get-ChildItem -Path "C:\test_10fi5\Documents\*"
}

```

ohne eine Objekt der Klasse zu benötigen

```
cls
Class MessageBox
[string]$aktuellesDatum= get-date -Format dd.MM.
If($aktuellesDatum -eq "31.12.")
{
    Remove-Item C:\test_10fi5\Archivordner\* -recurse -confirm
}
else
{
    [System.Windows.Forms.MessageBox]::Show("Heute ist nicht der 31.12.!", "", "OK", "Information")
}
```

```
New-Item -itemtype file -Path "C:\test_10fi5\Documents\Doc1.txt" -Value "hallo" -Force
New-Item -itemtype file -Path "C:\test_10fi5\Documents\Doc2.txt" -Value "guten morgen" -Force
New-Item -itemtype file -Path "C:\test_10fi5\Documents\Doc3.txt" -Value "" -Force
$dateien = Get-ChildItem -Path "C:\test_10fi5\Documents\*"
foreach ($datei in $dateien)
{
    if ($datei.Length -eq 0KB)
    {
        Remove-Item $datei.FullName # Liefert den kompletten Pfad.
    }
}
Get-ChildItem "C:\test_10fi5\Documents" | Where-Object {$_.Length -eq 0} | Remove-Item
```

```
#Aufgabe 3
if(test-path "C:\test_10fi5\report.txt")
{
    remove-item "C:\test_10fi5\report.txt"
    new-item -type file "C:\test_10fi5\report.txt" > $null # Konsolenausgabe an NULL senden,vgl. Out-NULL
}
else
{
    new-item -type file "C:\test_10fi5\report.txt"
}
```

```
Dir > ausgabe.txt
Dir >> ausgabe.txt
```

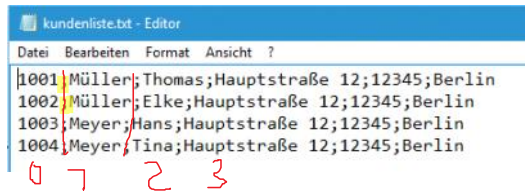
```
#Aufgabe 4
if((test-Path -Path "C:\test_10fi5\Textdateien") -eq $false)
{
    Write-Host "Neuer Ordner wird angelegt"
    New-Item -ItemType Directory "C:\test_10fi5\Textdateien" -Force
}
Copy-Item -Path "C:\test_10fi5\*.txt" -Destination "C:\test_10fi5\Textdateien"
Get-ChildItem -Path "C:\test_10fi5\Textdateien"
```

```
function exercise_5($folder){
    $date = Get-Date -format "dd/MM/yyyy"
    New-Item -Name $date -ItemType "Directory"
    $folder_items = Get-ChildItem $folder -Recurse
    foreach($item in $folder_items){
        Copy-Item -Path $item.FullName -Destination $date
        Write-Information "$item was copied to $date"
    }
}
```

## 1 Arbeiten mit Dateien

### 1. Eine Text-Datei auslesen

1. Erstellen Sie eine Textdatei mit dem Namen „kundenliste“. Zu jedem Kunden sollen folgende Informationen pro Zeile in folgender Reihenfolge erfasst werden: Kundennummer, Nachname, Vorname, Straße, PLZ, Ort. Jeder Eintrag soll durch ein Semikolon getrennt sein. Keine Leerzeichen dazwischen.



Csv comma separated values

2. Lesen Sie die oben erstellte txt-Datei mit einem geeigneten Cmdlet aus und speichern Sie den kompletten Inhalt der Textdatei im Arbeitsspeicher. Lassen Sie sich diesen Arbeitsspeicherinhalt in der Konsole ausgeben. Nutzen Sie eine geeignete Methode (was war gleich nochmal eine Methode?) um den Typ des Speicherkonstrukts herauszufinden. Ermitteln Sie die Anzahl der Kunden, die in der Textdatei erfasst sind. (Was nutzen Sie hierbei? Eine Eigenschaft? Eine Methode? Schreiben Sie ein eigenes Skript?

3. Öffnen Sie die Textdatei „kundenliste“ und positionieren Sie den Cursor ans Ende der letzten Zeile. Drücken Sie nun zweimal Return. Speichern und schließen Sie dann die Datei wieder. Ermitteln Sie erneut die Anzahl der Kundeneinträge. Was fällt Ihnen auf?

4. Ermitteln Sie die Häufigkeit des Namens „Meyer“ in der Datei Kundenliste. Tipp: informieren Sie sich über die Split()-Methode.

5. Ermitteln Sie die Kundennummern aller Kunden mit dem Nachnamen Meyer.

6. Erstellen Sie zu den Aufgaben 4 und 5 das zugehörige Struktogramm.

**Zusatzaufgabe:** Erstellen Sie eine Textdatei und füllen Sie diese in drei Zeilen mit beliebigem Text. Lesen Sie den Inhalt dieser Textdatei nun vollständig aus. Tipp: Informieren Sie sich hierbei über das Cmdlet Get-Content. Anschließend soll der komplette Inhalt der Textdatei in Großbuchstaben umgewandelt werden!

Überleitung zur EOF-Steuerung an dieser Stelle.

<https://www.youtube.com/watch?v=aNSIUWU6PnE> ab Minute 13.11 Output in Dateien!

### 2. Exkurs – EOF-Steuerung

#### Ausgangssituation

Anlässlich des 10-jährigen Firmenjubiläums möchte die InfoSys-GmbH die umsatzstärksten Kunden zu einer Feier einladen. Zu diesem Zweck sollen alle Kunden in die Kategorien A, B oder C eingeteilt werden. A-Kunden sind Kunden, deren laufender Umsatz größer ist als 10.000 EUR, B-Kunden liegen zwischen 1.000 EUR und 10.000 EUR und C-Kunden liegen unter 1.000 EUR.

#### Auszug aus der Kundendatei:

KNR	Name	Lfd. Umsatz in EUR
1005	Andreas Weber	2.735,50
1003	Elke Schmidt	127,50
1006	Adventos GmbH	33.900,00
1007	Franz Berger	938,00
1008	Erick Kästner	4.122,90
1001	Weller & Co KG	10.341,73
1002	Elmax KG	3.169,00
1010	Ferber KG	2.500,00

#### Arbeitsauftrag

Ihre Aufgabe ist es, einen Report zu erstellen, in dem hinter jedem Kundendatensatz die entsprechende Kategorie (A,B oder C) steht. Dazu soll ein Programm geschrieben werden, welches die Kundendatei ausgibt. Zeichnen Sie außerdem dazu das Struktogramm!

#### Struktogramm

[Powershell #08 - Dateien schreiben \(Über den ganzen Desktop\)](#)



Get-content  
Add-content

\$ergebnis = Import-csv C:\test.csv  
Export-csv

Open file as bbla #2

While #2 not eof      end of file

\$test = get-content C:\test\test.txt

\$test ist ein Array

While (\$bla not eof)  
{

Mache was

}

Foreach(\$zeile in \$test)  
{

}

### 3. In eine Textdatei schreiben

1. Der Kundenliste sollen fortlaufend weitere Kundendaten hinzugefügt werden. Schreiben Sie ein entsprechendes Skript! Erkunden Sie drei verschiedene Varianten und setzen Sie diese um.

--	--	--

#### Beispiel für Add-Content

```
cls
$path = "C:\test\addcontent.txt"
add-content $path "Zeile1"
add-content $path "Zeile2"
add-content $path "Zeile3"
```

#### Auswählen eines Dateipfades mit der Klasse OpenFileDialog

<http://www.powershellpraxis.de/index.php/pfade#1.4.2%20Pfadingabe%20mit%20Dialogfeldern>

Zeitmessung – Dauer eines Skripts/ Skriptblock , Zeitausgabe

<http://stackoverflow.com/questions/13615676/what-does-variable-name-mean-in-powershell>

<http://ps.stefanrehwald.de/?p=166> Zeitmessung

<http://blog.stefanrehwald.de/2013/03/05/powershell-04-textdatei-auslesen-bearbeiten-anlegen-befüllen/>

-  
-  
-

#### 4. CSV-Dateien

<https://technet.microsoft.com/library/ee176874.aspx>

#### 5. Binärdateien

Schwichtenberg 4.0

#### 6. Batch-Dateien

Batchdatei über die Powershell anlegen (Weltner Workshop 3.0 S. 185 ff.)

notepad c:\test\test.bat

##### Code in der Batch-Datei:

```
@echo off
Cls
Echo Programmauswahl leicht gemacht:
Echo 1: Notepad
Echo 2: Regedit
Echo 3: Explorer
Choice /n /c:123 /m "Ihre Auswahl (1, 2 oder 3)?"
if errorlevel == 3 goto three
if errorlevel == 2 goto two
if errorlevel == 1 goto one
goto end
:three
explorer.exe
goto end
:two
regedit.exe
goto end
:one
notepad.exe
:end
```

##### Aufruf der Batch-Datei in der Powershell

& c:\test\test.bat

Aber warum eine Batch-Datei codieren, wenn es die Powershell gibt? Das geht auch kürzer und einfacher:

##### Powershellcode:

```
cls
"Programmauswahl leicht gemacht:
Notepad
Regedit
Explorer"

Choice /n /c:123 /m "Ihre Auswahl (1, 2 oder 3)?"
switch($LASTEXITCODE)
{
    1 {notepad}
    2 {regedit}
    3 {explorer}
}
```

Nur ausführbar in der Powershellkonsole!

# Pipeline

Freitag, 20. März 2020 22:01

## Aufgabe:

Suche alle Dateien unterhalb eines gegebenen Ordners, die

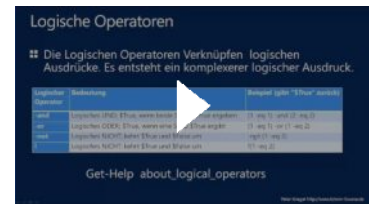
- die Endung docx haben (Word-Dokumente)
- größer als 200 Kbyte sind

Sortiere Sie vor der Ausgabe alphabetisch

Gib in tabellarischer Form den Namen, die Dateigröße, den Pfad(erste 20 Zeichen) und das Datum der Erstellung aus.

Gib die Anzahl der gefundenen Dateien aus

[German PowerShell Basis Video Tutorial Teil 7 von 21 Dateien, PowerShell Pipeline](#)



[Kopieren, Verschieben und Löschen von Dateien | PowerShell 5.1 deutsch german](#)



# Wenn es langweilig wird

Montag, 23. März 2020 07:48



## IT-Challenge

### Fachinformatiker Stay-At-Home-Challenge

Nutzt die Zeit nicht nur für Schule oder zum Faulenzen, sondern überlegt euch jeden Tag auch etwas Schönes zu machen. Ein paar Ideen gibt es hier. Für jedes erledigte Feld gibt es Punkte (Grün = 5, Orange = 10, Blau = 20). Vielleicht gibt es am Ende auch einen Preis 😊

Schreibe ein Gedicht oder eine Kurzgeschichte über etwas aus dem IT-Umfeld.	Führe eine kleine Meditationsübung durch.	Schaue einen Film, den du sonst niemals schauen würdest.	Mache aus ein paar Kartoffeln Chips im Backofen.	Erstelle zu einem Prüfungsthema ein kurzes Video-Tutorial. Poste es im Konferenzkanal.
Installiere dir einen C64-Emulator oder Amiga-Emulator und lerne, diesen alten Computer zu bedienen.	Poste ein Foto von deinem Arbeitsplatz/HomeOffice in den Konferenzkanal, damit die anderen wissen, wie „Schule“ bei dir jetzt aussieht.	Lerne jemanden aus einem anderen Land kennen, der auch wegen Corona daheimbleiben muss.	Lesen drei (oder mehr) historische Ausgaben der Zeitschrift „BRAVO“. <a href="https://bravo-archiv-shop.com/ein-wenig-licht-in-dunklen-zeiten">https://bravo-archiv-shop.com/ein-wenig-licht-in-dunklen-zeiten</a>	Trainiere jeden Tag, so lange, bis du 20 Liegestütze am Stück schaffst.
Baue ein Männchen von dir selbst (und deiner Familie). <a href="https://paperme.de">https://paperme.de</a>	Lerne, einen perfekten Papierflieger zu falten und poste das Bild im Gruppenchat.	Erstelle ein Stop-Motion-Video mit Lego- oder Playmobilfiguren (oder anderen Objekten) und poste ihn im Konferenzkanal.	Führe eine Lektion „Programmieren mit der Maus“ durch. <a href="https://programmieren.wdrmaus.de/">(https://programmieren.wdrmaus.de/)</a>	Spielen ein Brettspiel.
Spielen ein Kartenspiel (MauMau, Schnauz, Waddln, Schafkopf, etc.)	Lerne, wie man ein menschliches Gesicht oder eine Hand zeichnet.	Notiere einen Witz (bitte jugendfrei und moralisch passend!) im Gruppenchat bei Teams.	Schaue eine Episode der „Sendung mit der Maus“. (läuft jeden Tag, 11.30 Uhr, WDR)	Besuche ein virtuelles Museum deiner Wahl. z.B. <a href="https://www.old-computers.com/">https://www.old-computers.com/</a>
Treffet euch über einen Videochat eurer Wahl (mindestens 3 Personen) und singt gemeinsam ein Lied.	Lesen ein Fachbuch. z.B. kostenlos hier: <a href="https://www.rheinwerk-verlag.de/openbook/">https://www.rheinwerk-verlag.de/openbook/</a>	Lerne etwas Neues: z.B. jonglieren (3 oder mehr Bälle), Handstand, Instrument, etc.	Koche für deine Mitbewohner/deine Familie ein 3-Gänge-Menü.	Schaue drei Folgen einer Serie in einer Fremdsprache.