



Windows PowerShell Crashkurs

ppedv AG

Haiko Hertes
Dipl.Inf. (FH), M.Sc.

- Haiko Hertes
- 1986, verheiratet
- Informatik-Studium in Leipzig (Diplom (FH), Master of Science)
- Seit 2004 für verschiedene MS-Goldpartner tätig
- Seit 2011 für ppedv tätig
- MCT, MCTS, MCITP, MCSA & MCSE
- Kein „typischer Fanboy“ :)

Microsoft
CERTIFIED
Trainer

Microsoft
CERTIFIED
IT Professional

Microsoft
CERTIFIED
Technology Specialist

Microsoft
CERTIFIED
Solutions Associate

Microsoft
CERTIFIED
Solutions Expert



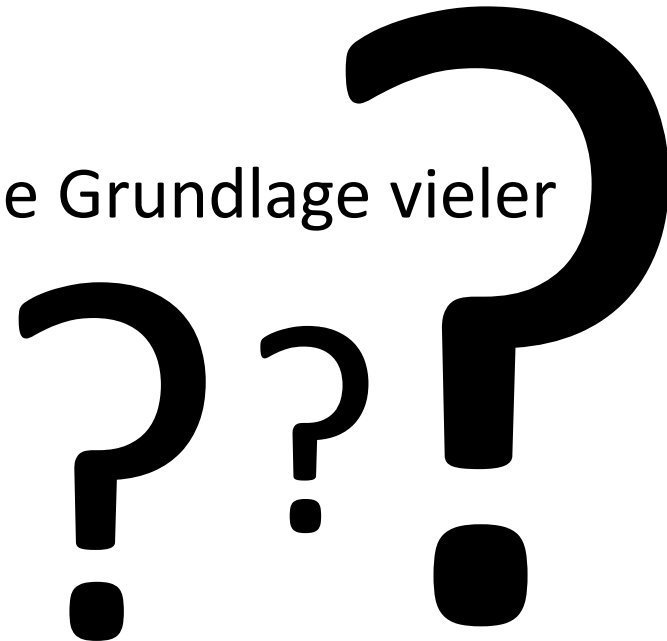
- Firmensitz in Burghausen
- Büros in verschiedenen Städten:
- Ca. 50-60 Mitarbeiter
- Schulungen für nahezu alle Microsoft-Technologien
- Konferenzen, Camps, Verlag (VisualStudioOne / VSOne)



- Was ist PowerShell?
- Was kann man damit machen?
- Variablen & Objekte

WAS IST POWERSHELL?

- Windows PowerShell ist...
 - Die vielleicht beste Skript-Sprache für Windows
 - Vielseitig einsetzbar
 - Leicht zu erlernen
 - Sehr umfangreich
 - Bereits seit längerem funktionale Grundlage vieler GUIs
 - So ein bisschen wie Linux



- Skript- & Kommundosprache
- (Bessere) Alternative zur CMD
- Wurde 2006 eingeführt
- Bereits seit Windows 7 / Server 2008 R2 fester Bestandteil der Microsoft-Betriebssysteme
- Nachinstallierbar (WMF)
- Objektorientiert
- Mitgelieferte IDE inkl. Debugger (ISE)
- Basiert auf .NET Framework

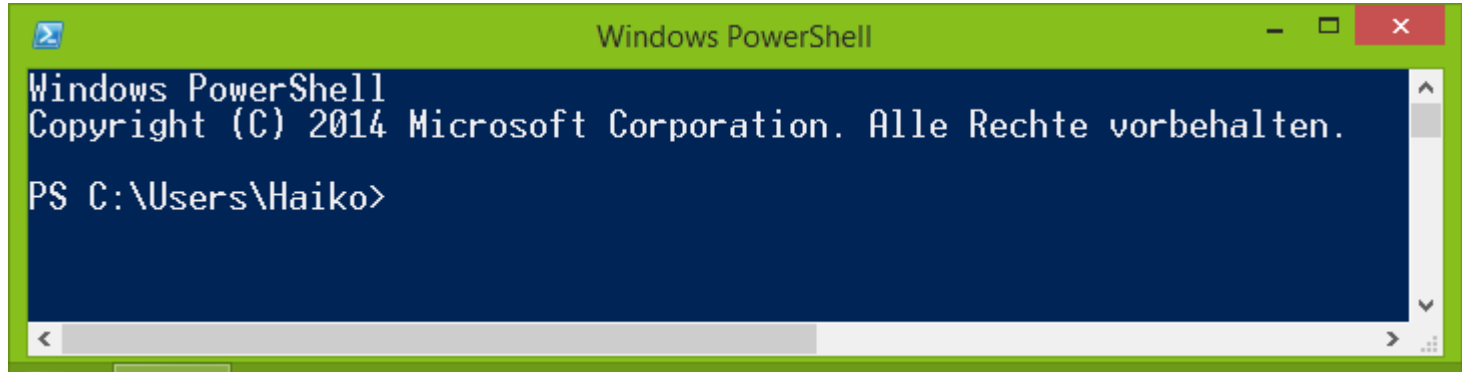
| | 2.0 | 3.0 | 4.0 |
|---|-------------|-------------------|-------------|
| Windows XP | Verfügbar | Nein | Nein |
| Windows Server 2003 | Verfügbar | Nein | Nein |
| Windows Vista | Verfügbar | Nein | Nein |
| Windows Server 2008 | Verfügbar | Verfügbar mit SP2 | Nein |
| Windows 7 | Installiert | Verfügbar mit SP1 | Verfügbar |
| Windows Server 2008 R2 | Installiert | Verfügbar mit SP2 | Verfügbar |
| Windows 8 | Nein | Installiert | Verfügbar |
| Windows Server 2012 | Nein | Installiert | Verfügbar |
| Windows 8.1 und Windows Server 2012 R2 | Nein | Nein | Installiert |

Windows PowerShell 1.0 und 2.0 benötigen .NET Framework 2.0

Windows PowerShell 3.0 benötigt .NET Framework 4.0

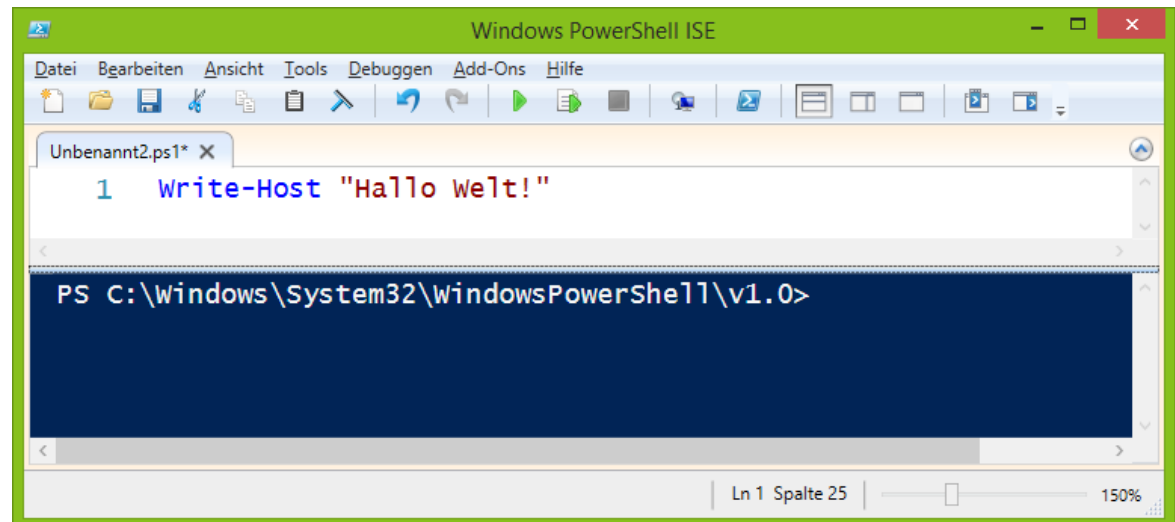
Windows PowerShell 4.0 benötigt .NET Framework 4.5

- PowerShell “klassisch” -> Shell



A screenshot of the classic Windows PowerShell console window. The title bar is green and says "Windows PowerShell". The background is dark blue with white text. The text inside reads: "Windows PowerShell", "Copyright (C) 2014 Microsoft Corporation. Alle Rechte vorbehalten.", and "PS C:\Users\Haiko>".

- PowerShell ISE -> Entwicklungsumgebung



A screenshot of the Windows PowerShell ISE (Integrated Scripting Environment) window. The title bar is green and says "Windows PowerShell ISE". The window has a menu bar with "Datei", "Bearbeiten", "Ansicht", "Tools", "Debuggen", "Add-Ons", and "Hilfe". Below the menu bar is a toolbar with various icons. The main area shows a script file named "Unbenannt2.ps1*" with a single line of code: "1 Write-Host 'Hallo Welt!'". Below the script editor is a console window with a dark blue background and white text, showing the prompt "PS C:\Windows\System32\WindowsPowerShell\v1.0>". The status bar at the bottom indicates "Ln 1 Spalte 25" and "150%".

GRUNDLAGEN

- Commandlet, eigentlich “cmdlet”
 - “Mini-Programme”
 - Immer einheitlicher Aufbau:
 - VERB-NOUN / VERB-SUBSTANTIV
 - Immer “Einzahl”
 - Bsp.: Get-Command
- Aliasse
- Functions
- Anzeigen aller bekannten Befehle:
“Get-Command”

- Jedes Cmdlet hat eine umfangreiche Hilfe
- Aufruf über `Get-Help CMDLET`
 - Alternativen (Alias!) `man` u. `help`
 - Detail-Grad: `-detailed`, `-examples`, `-full`, ...
- Seit PS 3.0: Hilfe nicht mehr default offline mitgeliefert
- Muss erst heruntergeladen werden

```
PS C:\Windows\System32\WindowsPowerShell\v1.0> Get-Help Get-Service
```

NAME

Get-Service

SYNTAX

```
Get-Service [[-Name] <string[]>] [-ComputerName <string[]>] [-DependentServices]
[-RequiredServices] [-Include <string[]>] [-Exclude <string[]>]
[<CommonParameters>]
```

```
Get-Service -DisplayName <string[]> [-ComputerName <string[]>]
[-DependentServices] [-RequiredServices] [-Include <string[]>] [-Exclude
<string[]>] [ <CommonParameters>]
```

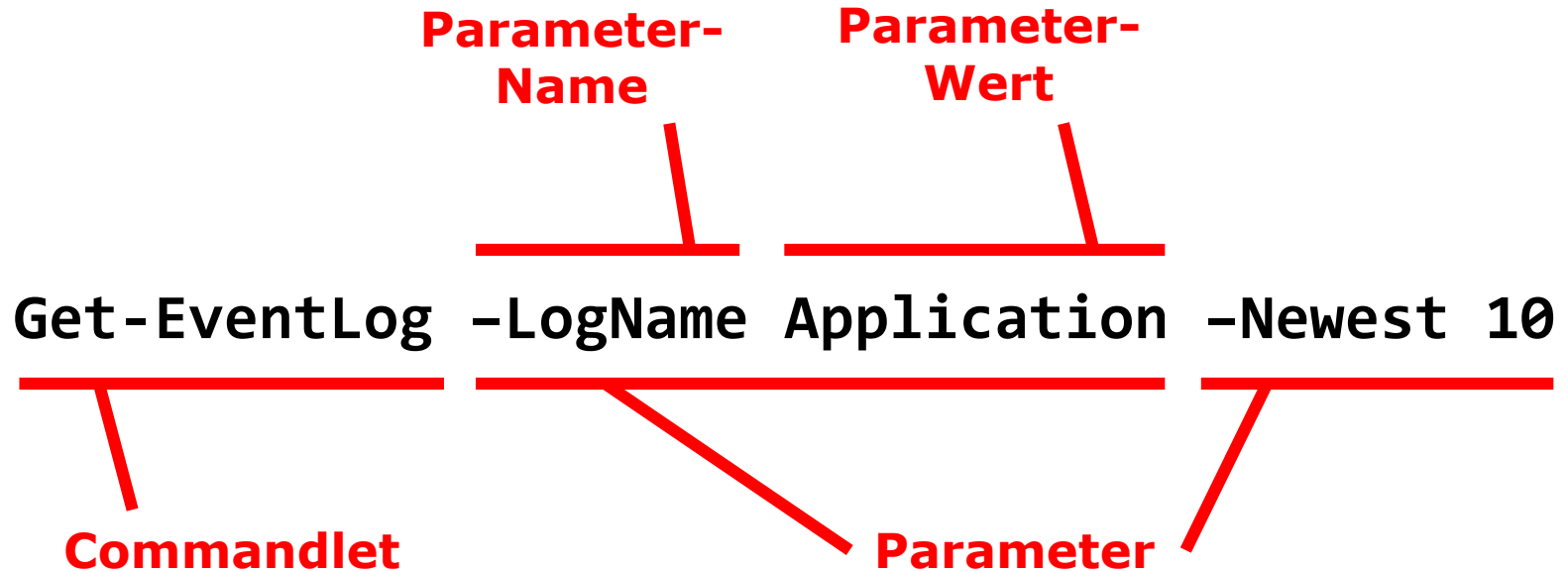
```
Get-Service [-ComputerName <string[]>] [-DependentServices] [-RequiredServices]
[-Include <string[]>] [-Exclude <string[]>] [-InputObject <ServiceController[]>]
[<CommonParameters>]
```

- Cmdlet finden -> Get-Command
- Hilfe lesen -> Get-Help
- Parameter beginnen mit “-”
- Keine Unterscheidung zwischen Groß- und Kleinschreibung
- Zwischen Cmdlet, Parametern und Parameterwerten jeweils Leerzeichen

**Parameter-
Name** **Parameter-
Wert**

Get-EventLog **-LogName** **Application** **-Newest** **10**

Commandlet **Parameter**



- Kompletter Aufruf vs. Kurzformen
 - Alias
 - Positionale Parameter
 - Abkürzungen bei den Parametern

Get-Service -Name BITS -ComputerName WIN2012

Alias

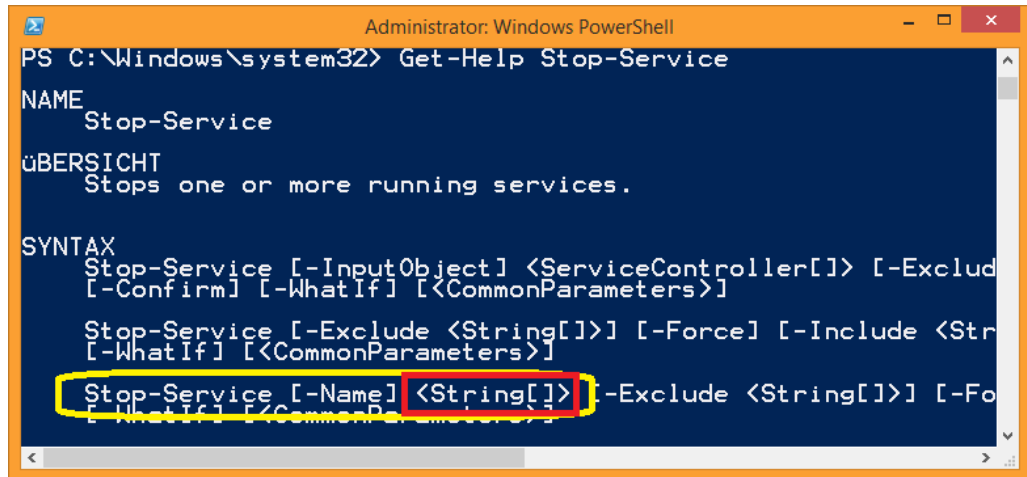
**Positionaler
Parameter**

**Verkürzter
Parameter-
Name**

gsv BITS -Comp WIN2012

- Einem Parameter können auch mehrere Werte übergeben werden:

-Name <String[]>



```
Administrator: Windows PowerShell
PS C:\Windows\system32> Get-Help Stop-Service

NAME
    Stop-Service

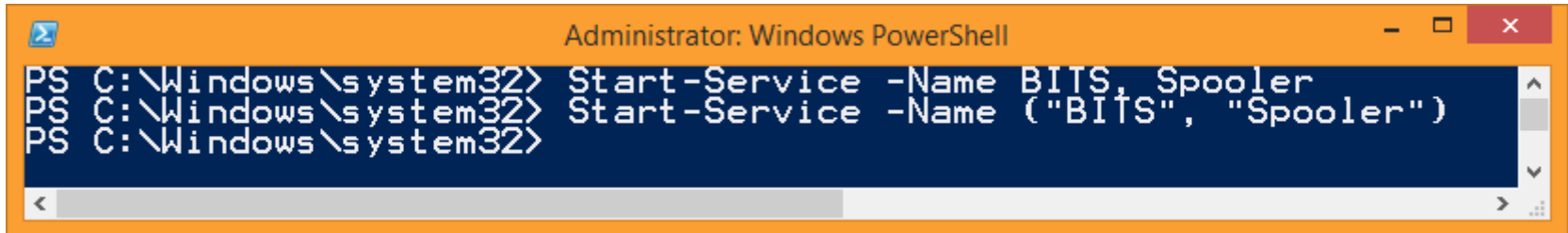
ÜBERSICHT
    Stops one or more running services.

SYNTAX
    Stop-Service [-InputObject] <ServiceController[]> [-Exclude
    [-Confirm] [-WhatIf] [<CommonParameters>]

    Stop-Service [-Exclude <String[]>] [-Force] [-Include <Str
    [-WhatIf] [<CommonParameters>]

    Stop-Service [-Name] <String[]> [-Exclude <String[]>] [-Fo
    [-WhatIf] [<CommonParameters>]
```

- Dazu erzeugt man eine Liste der Werte:



```
Administrator: Windows PowerShell
PS C:\Windows\system32> Start-Service -Name BITS, Spooler
PS C:\Windows\system32> Start-Service -Name ("BITS", "Spooler")
PS C:\Windows\system32>
```

- Jedes Cmdlet steckt in einem konkreten Modul
- Module müssen vor Benutzung der Cmdlets geladen werden
- PS 3.0 und neuer: Module werden automatisch bei Bedarf geladen
- Sonst (und wegen Abwärtskompatibilität!):
 - `Import-Module MODULNAME (Alias: ipmo)`

Windows PowerShell

Windows PowerShell
Copyright (C) 2013 Microsoft Corporation. Alle Rechte vorbehalten.

PS C:\Users\Haiko> Get-Module

| ModuleType | Version | Name | ExportedCommands |
|------------|---------|---------------------------------|---------------------------------|
| Manifest | 3.1.0.0 | Microsoft.PowerShell.Management | {Add-Computer, Add-Content, Che |

Windows PowerShell

PS C:\Users\Haiko> Get-Module -ListAvailable

Verzeichnis: C:\SkyDrive\Dokumente\WindowsPowerShell\Modules

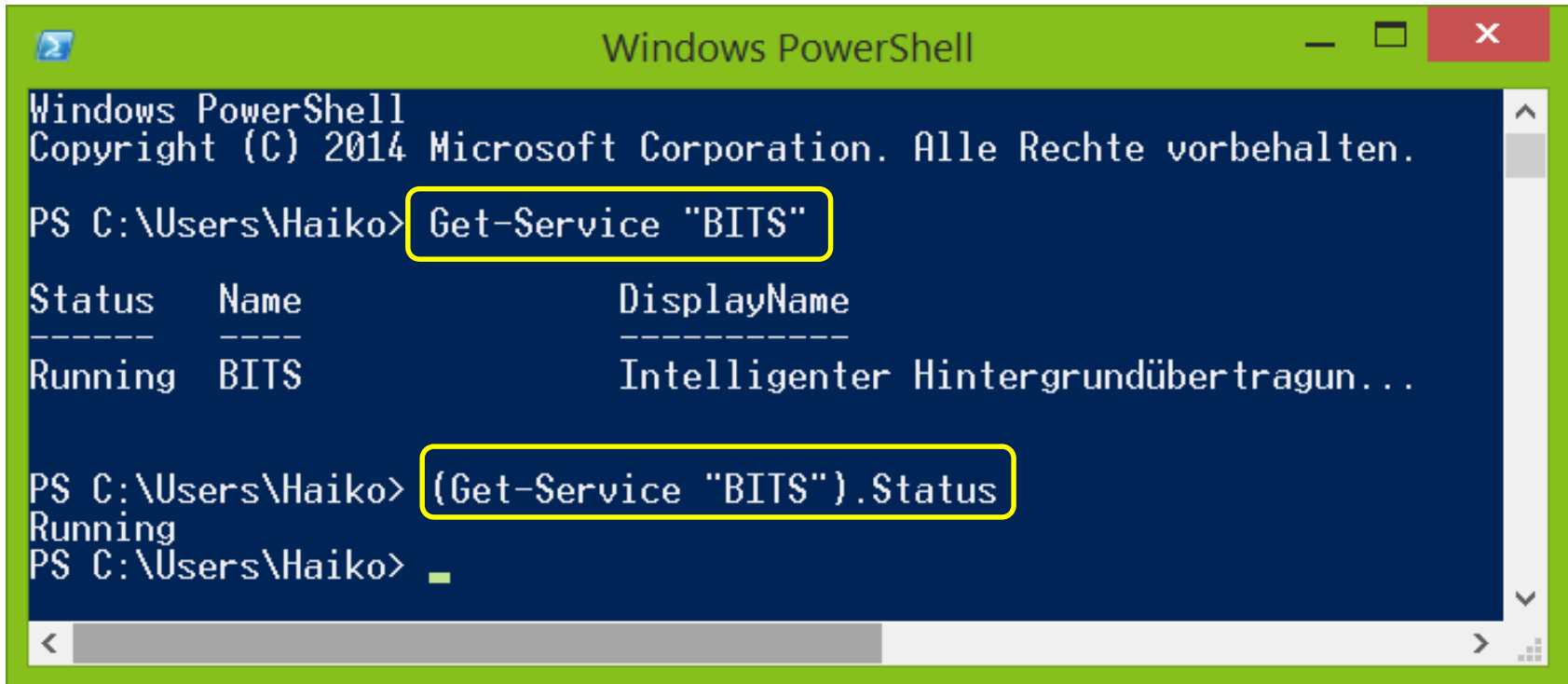
| ModuleType | Version | Name | ExportedCommands |
|------------|---------|----------------|---------------------------------|
| Script | 0.0 | DotNet | {Get-CommandWithParameterType, |
| Script | 1.0 | FileSystem | {Copy-ToZip, Get-DuplicateFile, |
| Script | 1.1 | IsePack | {Add-ForeachStatement, Add-IfSt |
| Script | 1.0 | PowerShellPack | {Add-ChildControl, Add-CodeGene |
| Script | 1.0 | PSCodeGen | {New-Enum, New-PInvoke, New-Scr |
| Script | 1.0 | PSImageTools | {Add-CropFilter, Add-RotateFlip |
| Script | 1.1 | PSRSS | {Get-Article, Get-Feed, New-Fee |
| Script | 1.0 | PSSystemTools | {Get-BootStatus, Get-DisplaySet |
| Script | 1.0 | PSUserTools | {Get-CurrentUser, Get-Everyone, |
| Script | 1.0 | TaskScheduler | {Add-TaskAction, Add-TaskTrigge |
| Script | 1.0 | WPK | {Add-ChildControl, Add-CodeGene |

Verzeichnis: C:\Windows\system32\WindowsPowerShell\v1.0\Modules

| ModuleType | Version | Name | ExportedCommands |
|------------|---------|-------------------|---------------------------------|
| Manifest | 1.0.0.0 | AppBackgroundTask | {Disable-AppBackgroundTaskDiagn |
| Manifest | 2.0.0.0 | AppLocker | {Get-AppLockerFileInformation, |
| Manifest | 2.0.0.0 | Appx | {Add-AppxPackage, Get-AppxPacka |
| Script | 1.0.0.0 | AssignedAccess | {Clear-AssignedAccess, Get-Assi |
| Manifest | 1.0.0.0 | BitLocker | {Unlock-BitLocker, Suspend-BitL |

- Variablen werden durch „\$“ gekennzeichnet
- Zuweisung von Werten idR mit „=„
 - Bsp.: `$Text = „Hallo Welt“`
- Müssen nicht initialisiert werden!
- Variablen können enthalten:
 - (primitive) Werte (Int, String, Char, ...)
 - Arrays von Werten
 - Objekte
 - Arrays von Objekten

- PowerShell ist objektorientiert
- Cmdlets liefern idR Objekte oder Arrays mit Objekten
- Objekte werden auf Konsole durch Text dargestellt
- Können zur weiteren Verarbeitung „ge-pipe-d“ werden
- Innerhalb eines Objektes kann auch auf einzelne Attribute zugegriffen werden



```
Windows PowerShell
Copyright (C) 2014 Microsoft Corporation. Alle Rechte vorbehalten.

PS C:\Users\Haiko> Get-Service "BITS"

Status      Name      DisplayName
-----
Running     BITS      Intelligenter Hintergrundübertragun...

PS C:\Users\Haiko> (Get-Service "BITS").Status
Running
PS C:\Users\Haiko>
```

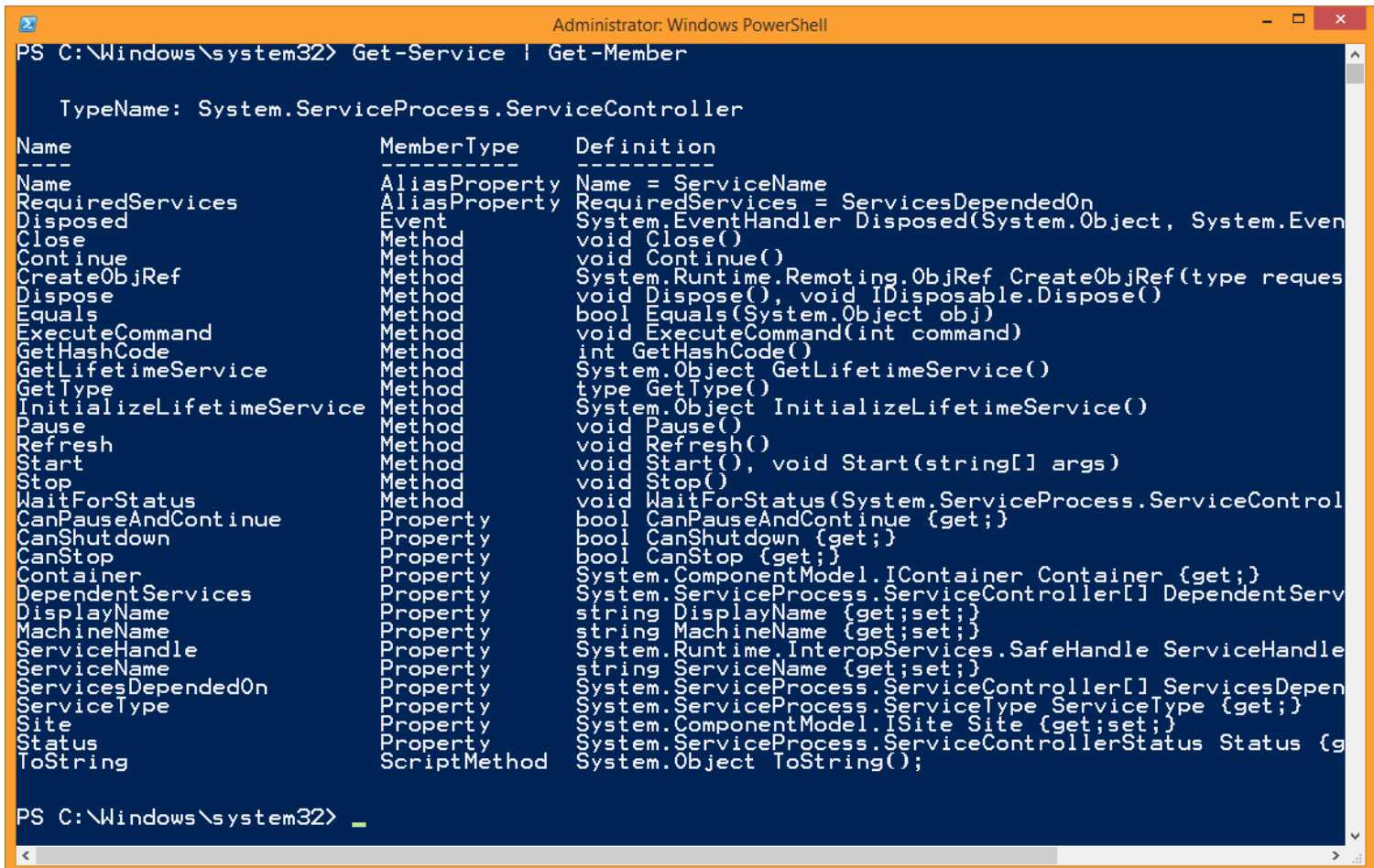
DIE PIPELINE

- PowerShell Kommandos werden in einer *Pipeline* ausgeführt
- Pipeline kann ein oder mehrere Commandlets enthalten
- Mehrere Commandlets werden durch „|“ verknüpft
- Commandlets werden von links nach rechts ausgeführt
- Auf der Konsole wird die Ausgabe des letzten Commandlets gezeigt

- Bsp.: **Get-Service | Out-File Services.txt**
- Das ist also die Kombination aus **Get-Service** und **Out-File**
- (Hier wird dann auch keine Ausgabe erzeugt!)
- In diesem Fall entspricht das etwa dem „> Services.txt“ aus DOS
- Funktioniert aber anders!

- Windows PowerShell ist objektorientiert
- Ausgabe von Commandlets erzeugt i.d.R. Objekte
- Diese werden meist in Textform dargestellt
- Objekte können enthalten:
 - Eigenschaften („Properties“)
 - Methoden („Methods“)
 - Ereignisse („Events“)

- Commandlets, die Objekte generieren, können an **Get-Member** übergeben werden:



```
Administrator: Windows PowerShell
PS C:\Windows\system32> Get-Service | Get-Member

TypeName: System.ServiceProcess.ServiceController
-----
Name      MemberType Definition
-----
Name      AliasProperty Name = ServiceName
RequiredServices AliasProperty RequiredServices = ServicesDependedOn
Disposed  Event         System.EventHandler Disposed(System.Object, System.Event
Close      Method        void Close()
Continue   Method        void Continue()
CreateObjRef Method        System.Runtime.Remoting.ObjRef CreateObjRef(type reques
Dispose    Method        void Dispose(), void IDisposable.Dispose()
Equals     Method        bool Equals(System.Object obj)
ExecuteCommand Method        void ExecuteCommand(int command)
GetHashCode Method        int GetHashCode()
GetLifetimeService Method        System.Object GetLifetimeService()
GetType    Method        type GetType()
InitializeLifetimeService Method        System.Object InitializeLifetimeService()
Pause      Method        void Pause()
Refresh    Method        void Refresh()
Start      Method        void Start(), void Start(string[] args)
Stop       Method        void Stop()
WaitForStatus Method        void WaitForStatus(System.ServiceProcess.ServiceControl
CanPauseAndContinue Property       bool CanPauseAndContinue {get;}
CanShutdown Property       bool CanShutdown {get;}
CanStop    Property       bool CanStop {get;}
Container  Property       System.ComponentModel.IContainer Container {get;}
DependentServices Property       System.ServiceProcess.ServiceController[] DependentServ
DisplayName Property       string DisplayName {get;set;}
MachineName Property       string MachineName {get;set;}
ServiceHandle Property       System.Runtime.InteropServices.SafeHandle ServiceHandle
ServiceName Property       string ServiceName {get;set;}
ServicesDependedOn Property       System.ServiceProcess.ServiceController[] ServicesDepen
ServiceType Property       System.ServiceProcess.ServiceType ServiceType {get;}
Site       Property       System.ComponentModel.ISite Site {get;set;}
Status     Property       System.ServiceProcess.ServiceControllerStatus Status {g
ToString   ScriptMethod   System.Object ToString();

PS C:\Windows\system32>
```

- Pipeline ermöglicht
 - komplexe Aufgaben in sehr kurzen Aufrufen (dazu später mehr)
 - Formatierung der Ausgabe
 - Filterung
 - Sortierung
 - uvm.

Die Pipeline – Formatierung der Ausgabe

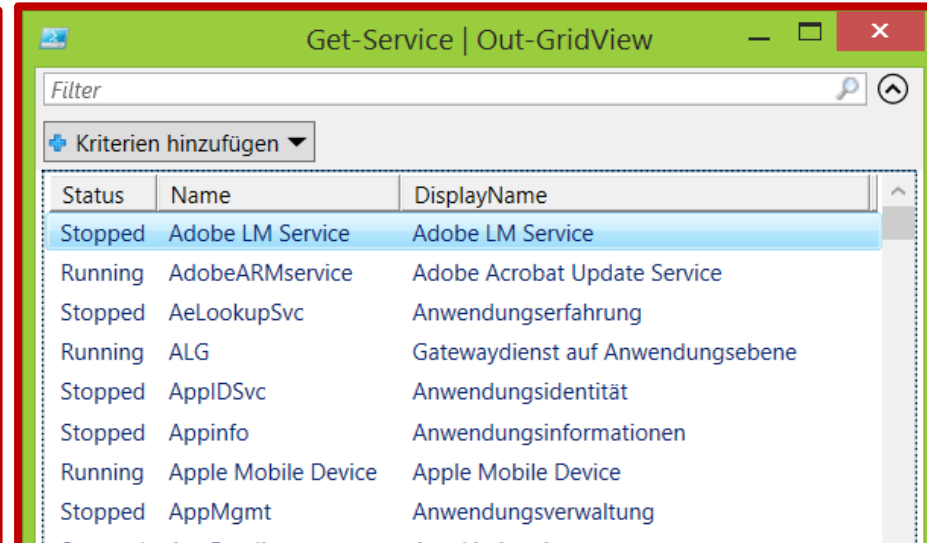
```
PS C:\Users\Haiko> Get-Service
```

| Status | Name | DisplayName |
|---------|------------------|-----------------------------------|
| Stopped | Adobe LM Service | Adobe LM Service |
| Running | AdobeARMservice | Adobe Acrobat Update Service |
| Stopped | AeLookupSvc | Anwendungserfahrung |
| Running | ALG | Gatewaydienst auf Anwendungsebene |
| Stopped | AppIDSvc | Anwendungsidentität |

```
PS C:\Users\Haiko> Get-Service | Format-List
```

```
Name                : Adobe LM Service
DisplayName          : Adobe LM Service
Status              : Stopped
DependentServices    : {}
ServicesDependedOn   : {}
CanPauseAndContinue : False
CanShutdown          : False
CanStop              : False
ServiceType          : Win32OwnProcess

Name                : AdobeARMservice
DisplayName          : Adobe Acrobat Update Service
Status              : Running
```



| Status | Name | DisplayName |
|---------|---------------------|-----------------------------------|
| Stopped | Adobe LM Service | Adobe LM Service |
| Running | AdobeARMservice | Adobe Acrobat Update Service |
| Stopped | AeLookupSvc | Anwendungserfahrung |
| Running | ALG | Gatewaydienst auf Anwendungsebene |
| Stopped | AppIDSvc | Anwendungsidentität |
| Stopped | Appinfo | Anwendungsinformationen |
| Running | Apple Mobile Device | Apple Mobile Device |
| Stopped | AppMgmt | Anwendungsverwaltung |

```
PS C:\Users\Haiko> Get-Service | Format-Table Name,Status
```

| Name | Status |
|------------------|---------|
| Adobe LM Service | Stopped |
| AdobeARMservice | Running |
| AeLookupSvc | Stopped |
| ALG | Running |

- **Bsp.:** `Get-Childitem
C:\Testfiles*.txt -Recurse |
Remove-Item -WhatIf`
- **Bsp.:** `Get-Service | Sort-Object
Status | Select-Object -First
10 | Start-Service`

WICHTIGE BASIS-CMDLETS

- Get-Command
- Get-Help
- Format-Table / Format-List / Format-Wide
- Sort-Object
- Select-Object
- Where-Object
- ForEach-Object

SKRIPTE



- Skriptsicherheit soll:
 - einen uninformierten Benutzer
 - welcher unbeabsichtigt
 - versucht, ein nicht-vertrauenswürdiges Skript auszuführen
 - verlangsamen.
- Sie kann nicht verhindern, dass ein informierter Benutzer gewollt ein Skript ausführt
- Sie ersetzt auch keinen Anti-Malware-Schutz

- Fünf „Execution Policy“ Einstellungen:
 - Restricted (Standard)
 - AllSigned
 - RemoteSigned
 - Unrestricted
 - Bypass
- Drei Wege, die Einstellung zu ändern:
 - **Set-ExecutionPolicy**
 - GPO
 - **-ExecutionPolicy** Paramater an **PowerShell.exe**

| WERT | AUSWIRKUNG |
|---------------------|--|
| Restricted | Es werden keine Konfigurationsdateien geladen und keine Scripts ausgeführt (Standard) |
| AllSigned | Signierte Scripts und Konfigurationsdateien von einem vertrauenswürdigen Herausgeber werden ausgeführt. Auch lokal erstellte Scripts müssen signiert sein. |
| RemoteSigned | Aus dem Internet heruntergeladenen Scripts und Konfigurationsdateien müssen von einem vertrauenswürdigen Herausgeber signiert sein. |
| Unrestricted | Alle Konfigurationsdateien und alle Scripts werden ausgeführt. Bei nicht signierten Scripts aus dem Internet muss man jede Ausführung am Prompt bestätigen |
| Bypass | Keinerlei Blockade, keine Warnungen oder Prompts. |
| Undefined | Entfernt die gerade zugewiesene Richtlinie (nur für lokal zugewiesene Richtlinien, nicht für GPO-applizierte) |



- Zu Beginn: Einfacher Aufruf mit spezifischen Parametern
- Testen, ob Syntax und Ausgabe passen

```
Get-EventLog -LogName Security  
-ComputerName localhost |  
Where EventID -eq 4624 |  
Select -First 50
```

- Dann: Werte identifizieren, die sich beim nächsten Einsatz evtl. ändern könnten:

```
Get-EventLog -LogName Security  
-ComputerName localhost |  
Where EventID -eq 4624 |  
Select -First 50
```

- Parameter einführen:

```
[CmdletBinding()]  
Param(  
    [Parameter(Mandatory=$True)]  
    [string] $ComputerName,  
  
    [int] $EventID = 4624  
)  
Get-EventLog -LogName Security -ComputerName  
$ComputerName |  
Where EventID -eq $EventID |  
Select -First 50
```

**Standard-
Wert; kann
geändert
werden**

- Im Skript kann eine Dokumentation hinterlegt werden
- Schlagworte:
 - Synopsis
 - Description
 - Parameter
 - Example
 - Weitere
- **help about_comment_based_help** (sehr umfangreich!)

Beispiel:

```
<#  
.SYNOPSIS  
Retrieves network adapter information from a computer.  
.DESCRIPTION  
Uses CIM to retrieve information about physical adapters only.  
.PARAMETER ComputerName  
The name of the computer to query.  
.EXAMPLE  
.\Get-NetAdapterInfo.ps1 -ComputerName LON-DC1 -Verbose  
#>
```

- Parameter können/sollten auf Gültigkeit getestet werden
- Eine Variante:

Windows PowerShell ISE

Parameter_ohne_Check.ps1

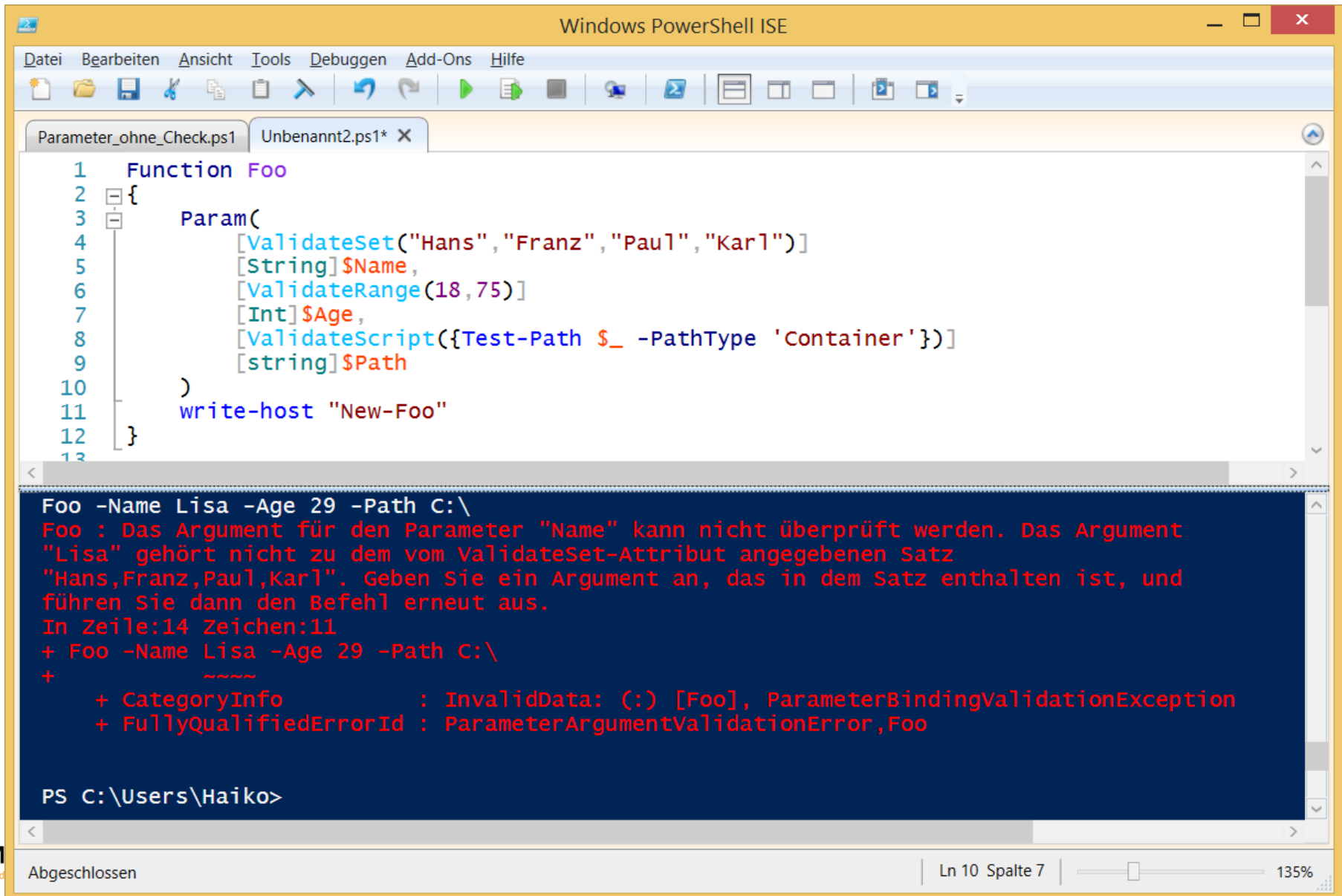
```
1 Function Foo
2 {
3     Param(
4         [String] $Name,
5         [Int] $Age,
6         [string] $Path
7     )
8     If ($Name -NotContains "Hans","Franz","Paul","Karl")
9         {Throw "$($Name) is not a valid name! Please use Hans, Franz, Paul or Karl"}
10    If ($Age -lt 18 -OR $Age -gt 75)
11        {Throw "$($Age) is not a between 18 and 75"}
12    IF (-NOT (Test-Path $Path -PathType 'Container'))
13        {Throw "$($Path) is not a valid folder"}
14
15    # All parameters are valid so New-stuff"
16    write-host "New-Foo"
17 }
18
19 Foo -Name Lisa -Age 29 -Path C:\
```

PS C:\Users\Haiko> D:\SkyDrive\Dokumente\!!ppedv\Kurse\PowerShell - Administration automatisieren\Lisa is not a valid name! Please use Hans, Franz, Paul or Karl
In D:\SkyDrive\Dokumente\!!ppedv\Kurse\PowerShell - Administration automatisieren\Samples\Skriptsammlung\Parameter_ohne_Check.ps1:9 Zeichen:10
+ {Throw "\$(\$Name) is not a valid name! Please use Hans, Franz, Paul or Ka ...
+
+ CategoryInfo : OperationStopped: (Lisa is not a v...z, Paul or Karl:Str
ing) [], RuntimeException
+ FullyQualifiedErrorId : Lisa is not a valid name! Please use Hans, Franz, Paul o

Abgeschlossen | Ln 19 Spalte 33 | 135%

- Besser: Gültige Parameter-Werte bei Definition festlegen!

```
Param(  
    [ValidateSet("Hans", "Franz", "Paul", "Karl")]  
    [String] $Name,  
    [ValidateRange(18, 75)]  
    [Int] $Age,  
    [ValidateScript({Test-Path $_ -PathType 'Container'})]  
    [string] $Path  
)
```



The screenshot shows the Windows PowerShell ISE interface. The top menu bar includes 'Datei', 'Bearbeiten', 'Ansicht', 'Tools', 'Debuggen', 'Add-Ons', and 'Hilfe'. The toolbar contains various icons for file operations and execution. The script editor shows a file named 'Unbenannt2.ps1' with the following PowerShell code:

```
1 Function Foo
2 {
3     Param(
4         [ValidateSet("Hans","Franz","Paul","Karl")]
5         [String] $Name,
6         [ValidateRange(18,75)]
7         [Int] $Age,
8         [ValidateScript({Test-Path $_ -PathType 'Container'})]
9         [string] $Path
10    )
11    write-host "New-Foo"
12 }
13
```

The console window shows the execution of the script with the command `Foo -Name Lisa -Age 29 -Path C:\`. The output is an error message in red text:

```
Foo : Das Argument für den Parameter "Name" kann nicht überprüft werden. Das Argument
"Lisa" gehört nicht zu dem vom ValidateSet-Attribut angegebenen Satz
"Hans,Franz,Paul,Karl". Geben Sie ein Argument an, das in dem Satz enthalten ist, und
führen Sie dann den Befehl erneut aus.
In Zeile:14 Zeichen:11
+ Foo -Name Lisa -Age 29 -Path C:\
+ ~~~~
+ CategoryInfo          : InvalidData: (:) [Foo], ParameterBindingValidationException
+ FullyQualifiedErrorId : ParameterArgumentValidationError,Foo

PS C:\Users\Haiko>
```

The status bar at the bottom indicates 'Abgeschlossen' (Completed) and shows the current position as 'Ln 10 Spalte 7' with a zoom level of 135%.

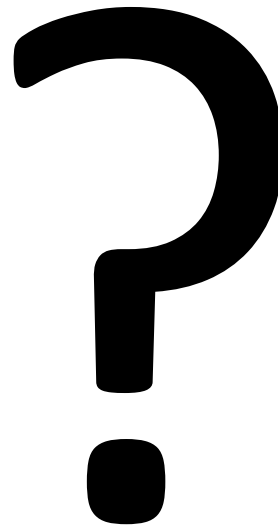
DESIRED STATE CONFIGURATION

- Just a Demo – enjoy the show! ;)



WAS ICH (WAHRSCHEINLICH) ALLES NICHT ANGESPROCHEN HABE

- Was tut die Pipeline im Hintergrund?
- Schleifen
- Aufzählen
- Verzweigungen
- Import, Export, Konvertierung
- WMI/CIM
- Error-Handling
- Eigene Module / Funktionen schreiben
- PowerShell Web-Access
- PSRemoting
- ...



Und zum Schluss...

A gold-colored crown icon with a cross on top, set against a blue background.

KEEP
CALM
AND
LEARN
POWERSHELL



- Fr., 31. Juli 2015
- Straßenbahnhof Angerbrücke, Leipzig
- MVPs, Insider, Community-Leader
- Buffet, Getränke, Give-Aways
- Kostenlos!!

<http://www.sysadminday2015.de>

Haiko Hertes

HaikoH@ppedv.de

vCard:

www.hertes.net



Profil auf XING, LinkedIn, Twitter,
Facebook, Flickr,