Powershell Einsteiger Workshop
Teil 1

PowerShell
UsergroupAustria

Roman Stadlmair
Microsoft PowerShell MVP
https://www.powershell.co.at

[Security.Principal.WindowsIdentity]::GetCurrent()

- Angestellt bei SEPPmail GmbH
- Roman Stadlmair [rconsult.at]

  Projekte-IT-Training
- PowerShell MVP (seit 2016)
- Gründer der PowerShell Usergroup Austria
- rs@rconsult.at
- XING, LinkedIn

Erwartungen

Fotos ?

Überblick Usergroup

- Gründung Jänner 2016
- XING: PowerShell Usergroup Austria
- www.powershell.co.at
- Wöchentliche Zusammenfassung von News „SnippetRace“
- 1 x Quartal Usergroup Treffen www.expertslive.at

Ehrenamtlich

Aktuelle Events

- Exertslive Cafe Ried im Innkreis
  - Office365 der Stand der Dinge
  - 10. März 2020
  - Infotech
- Expertslive Cafe Wien
  - Offie365 der Stand der Dinge
  - 12. März 2020
  - ITLS
- PowerShell 7 Launch Event
  - 30. März 2020
  - ITLS

Aktuelle Trainings

- Automatisieren mit PowerShell leicht gemacht
  - 23. März – 24. März 2020
  - ITLS – Roman
  - € 790
- Administrationswerkzeuge mit PowerShell entwickeln
  - 6. April – 7. April 2020
  - ITLS – Patrick
  - € 790

Dieser Kurs …

- Dient zum kennenlernen von PowerShell
- Hat den Fokus auf die Überwindung der wichtigsten Einstiegshürden
- Detaillierte Themen ➔ Folgeworkshops
- Erkennen von PS-Projekten ➔ Folgeprojekte

Wir werden

- Themen anschneiden
- Beispiele durchgehen
- Details manchmal übergehen müssen
- Neugier wecken
- Fragen gleich beantworten
  (sofern es die Zeit erlaubt)

Wichtigste 3 Regeln

1.) Niemand kann Powershell zu 100%
2.) Es gibt immer mehrere Möglichkeiten
3.) Jemand hat das schon mal gemacht ..

PowerShell starten

https://docs.microsoft.com/en-us/powershell/scripting/core-powershell/console/powershell.exe-command-line-help?view=powershell-5.1

- powershell.exe (5.1) oder pwsh.exe (7-Core)
- Parameter (die wichtigsten):
  - -noProfile
  - -noLogo
  - -nonInteractive
  - -executionPolicy
  - -MTA [Performance]
  - -EncodedCommand [Für komplexere Aufrufe]
  - -Command {PS Commands}

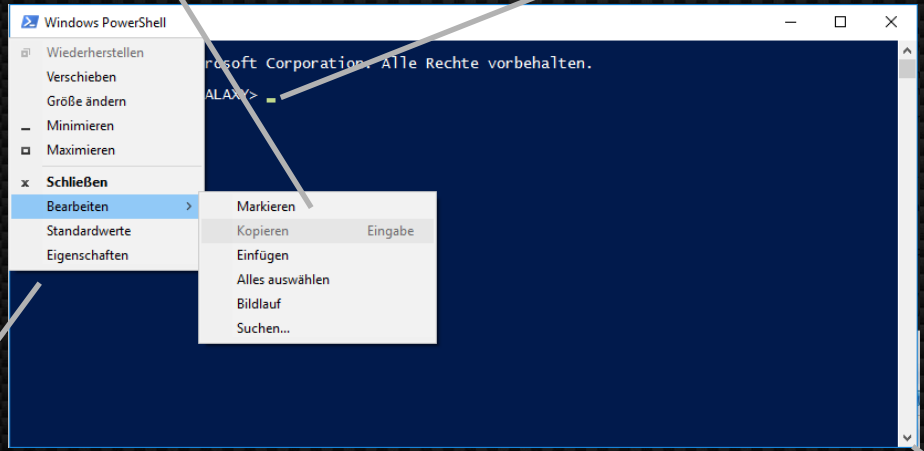
```
powershell.exe -nop -ex all -nol -noni -command "& {get-process|select Name}"
```

Powershell Konsole

Konsolenelemente

Einfügen, markieren/kopieren

Benutzerkontext



Eigenschaften

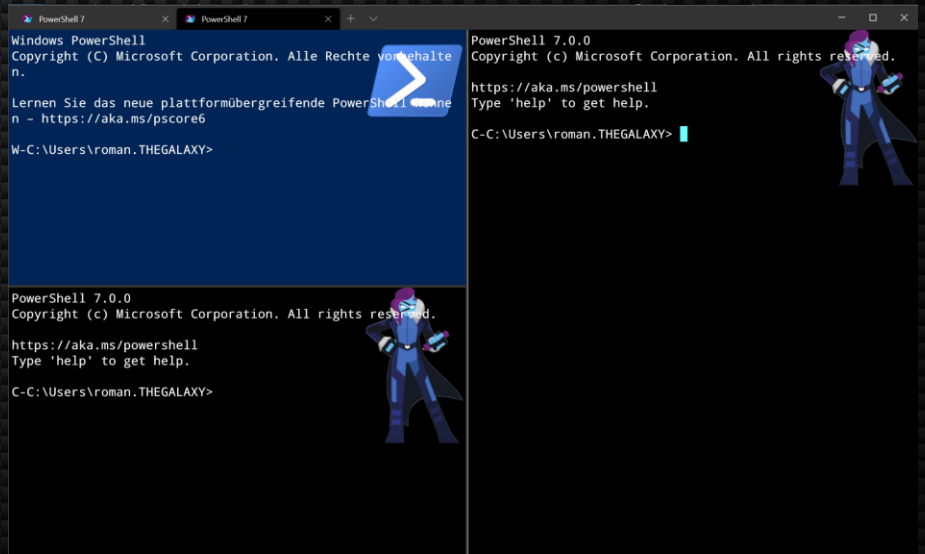
Fenstergröße und Puffer

Powershell Konsole

- Dos-Box Ersatz ?
- Öffnen und Eigenschaften
- Eingaben (Befehle, Rechnen, Vollständigkeit)
- Tastenkombinationen
  - Cursors, CPM
  - Autocomplete/Intellisense
  - History

Windows Terminal

- Mehrere Sessions
- Unterschiedliche Terminals
- Mit JSON konfigurierbar
- Fenster
  - <Alt>+<Shift>+<+>
  - <Alt>+<Shift>+<->
  - <Alt>+<Shift>+<Cursor>
- Einfach schön …



Powershell
Intregrated Scripting Environment

Powershell_ISE.exe

Layout

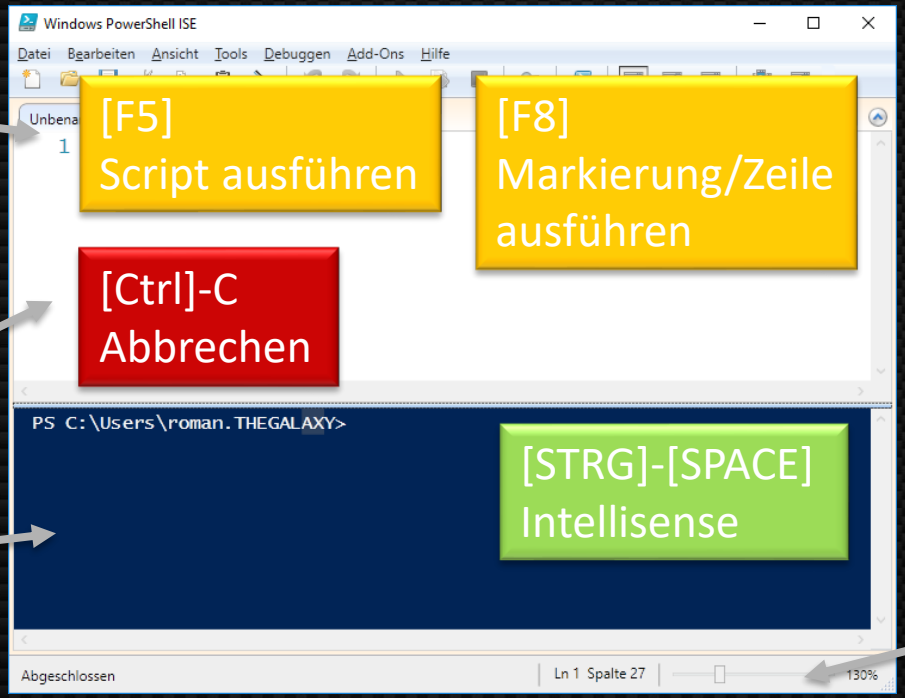
[STRG]-N
Neues Script

[STRG]-O
Script öffnen

[STRG]-T
Neuer Tab

[STRG]-[Shift]+R
Neuer Remote Tab

[STRG]-R
On/OFF



#[STRG]-[SPACE]
Befehlschronik

[STRG]-I
Scriptbereich

[STRG]-D
Konsole

[STRG]-[+]/[-]
Zoom

Steuerung der ISE durch Variablen (Auszug)

- $PsISE.Options
  - .ConsolePaneBackgroundColor
- $PsISE.CurrentFile
  - .DisplayName
  - .FullPath
- $PsISE.PowerShellTabs
- $PsISE.CurrentPowerShellTab

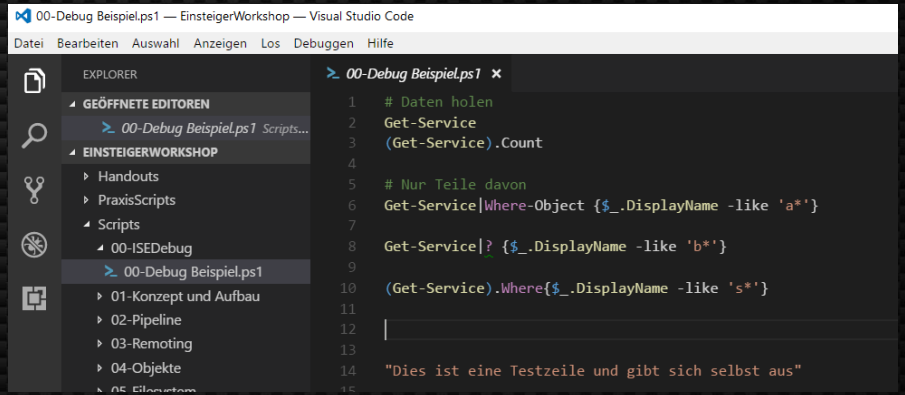
ISE Debugging Tastenkürzel

[F9]	Haltepunkt setzen
[Strg]+[Shift]+[F9]	Alle Haltepunkte löschen
[F11]	Einzelschritt
[F10]	Überspringen
[Shift]-[F11]	Rücksprung
[F5]	Starten
[Shift]-[F5]	Beenden
[Strg]+[Shift]+L	Alle Haltepunkte auflisten
[Strg]+[Shift]+D	Aufrufliste anzeigen

Visual Studio Code

VS Code

- Wird konstant weiterentwickelt
- Hat viele Plugins (PowerShell)
- Atom-basiert
- GitHub Integration
- Community-driven
- Wird immer besser
  - Debugging …
  - Breadcrumbs …



Blog Posts zu VS Code

- VS Code wir die ISE einrichten
  https://www.powershell.co.at/visual-studio-code-als-ersatz-zur-powershell-ise-installieren-und-einrichten/
- Debuggen mit VS Code
  https://code.visualstudio.com/docs/editor/debugging
- Debug PowerShell mit VS Code
  https://rkeithhill.wordpress.com/2015/12/27/debugging-powershell-script-with-visual-studio-code/

Entscheidungshilfe

- Linux oder macOS ?
- Client oder Server ?
- Fokus Window Administration ?

Konzept

Warum Powershell ?

- Lese Informationen von vielen Geräten
- Standardisierung von Aufgaben
- Automatisierung
- Berichte
- Werkzeuge selbst erstellen
- … Deine Ideen ☺

PowerShell Geschichte

https://technet.microsoft.com/en-us/library/hh847833%28v=wps.620%29.aspx

- 1.0: Download für XP, Windows 2003
- 2.0: ISE, Module/Snapins
- 3.0: Verbesserungen/Vereinfachungen, z.B.
  - Syntax: Get-Service | Where { $_.Status -eq 'running' } ➔ Get-Service | Where Status -eq 'running'
  - Lokale Variablen in Remote Sessions, Variablenvalidierung,…
- 4.0: DSC, Enhanced Debugging,Workflows, WebServices
  - -PipelineVariable
  - (Get-Process).where{$_.Name -match 'powershell'}

Powershell 5.1

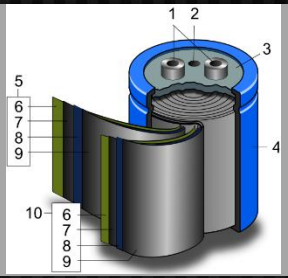
https://blogs.msdn.microsoft.com/powershell/2017/01/19/windows-management-framework-wmf-5-1-released/

- Klassen definieren
- Convert-FromString (MS-Research)
- Neue Module:
  - Microsoft.Powershell.Archive
  - OneGet (Package Manager)
  - PowershellGet (PS Ressource Gallery, oder internes Repo)
  - Network-Switch (Windows Server logo certified switches)
  - Local Users/Groups
- Wait-Debugger –> Script anhalten
- Neuigkeiten im DSC und vieles mehr

PowerShell 7 (aka PowerShell Core)

- Auf Guithub
  https://github.com/PowerShell/PowerShell
- Open Source
- Multi Plattform
  - Windows
  - Mac
  - Linux (dzt. 10)
- .Net Core ➔ reduzierte Funktionalität
- Sprung 6 auf 7 ➔ Wieder mehr Windows Funktionalität

Aufbau



Wo ist PowerShell im System ?

- $psHome = Powershell Programmverzeichnis
- Get-ChildItem $pSHome zeigt u.a.:
  - EXE Files (Core Engine)
  - Module
  - Beispiele
  - Hilfe (Sprachspezifisch)
- $env:PsModulePath = User und Systemspezifische Module

Hilfe finden

- Get-Help Thema –online
- Get-Help Thema | code -
- Get-Help about_*
- Get-Help Parameter
  - -Parameter
  - -Examples
- | more oder *Out-Host -paging*
- | clip oder Set-Clipboard

Links

- Liste der erlaubten CmdLet Verbs
  - https://technet.microsoft.com/en-us/library/ms714428%28v=VS.85%29.aspx
  - Get-Verb

Befehle raten

- Ausgabe aktuelles Datum?
- Eventlog auslesen ?
- Installierte Hotfixes ansehen ?
- Computer neu starten ?

Welche Befehle kann ich ausführen ?

- Dir, ls ,cat, Get-Process
- Powershell akzeptiert:
  - CmdLets
  - Aliase
  - Funktionen
  - Workflows
  - Anwendungen

Reihenfolge der Befehlsausführung

- Aliase
- Funktionen
- CmdLets
- Anwendungen
- PowerShell Scripts
- Andere Skripts und Dokumente
- Rechnen (Grundrechnungsarten und IT-Einheiten)

$alias:ls

$function:v:

Get-Command

- Zeigt Befehle durch PS und Module verfügbar sind.
- Wichtigste Parameter
  - -Listimported (Alle in der aktuellen Session verfügbaren)
  - -CommandType
  - -Module
  - -Verb -Noun
- Hilfe bei der Ausführung mit Show-Command

Befehlsaliase

- Dir,ls,gcm,ft sind Aliase für Befehle
- Get-Alias
- Cd alias:\
- Set-Alias Parameter

Befehlhistorie

- Get-History
  - Zeigt Befehlsgeschichte an (ID, Befehl, Ausführungsstatus, Start und EndZeit)
- Invoke-History
  - “alten” Befehl ausführen
- Clear-History
- Add- History
  - get-history |Export-Csv PSHist.CSV
  - import-csv pshist.csv |add-history