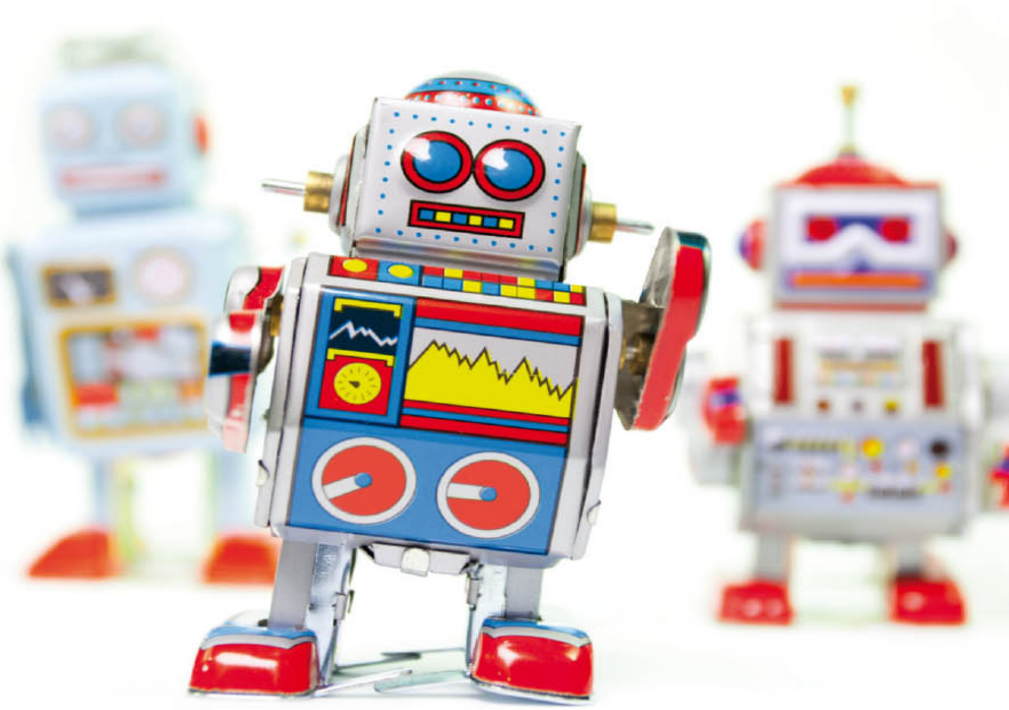
Mit vielen Beispielen aus der Praxis

PowerShell 7
und Windows PowerShell

Das komplette Praxiswissen für Administratoren und IT-Profis

Für Windows – Linux – macOS – Cloud



Dr. Tobias Weltner

O'REILLY®

Dr. Tobias Weltner

PowerShell 7 und Windows PowerShell

Das komplette Praxiswissen für Administratoren und IT-Profis

O'REILLY®

Dr. Tobias Weltner

Lektorat: Ariane Hesse
Lektoratsassistenz: Anja Weimer, Julia Griebel
Korrektorat: Sibylle Feldmann, *www.richtiger-text.de*
Satz: Gerhard Alfes, mediaService, *www.mediaservice.tv*
Herstellung: Stefanie Weidner
Umschlaggestaltung: Michael Oreal, *www.oreal.de*, unter Verwendung eines Fotos von iStock by Getty Images von querbeet, Stock-Fotografie-ID184997148

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen
Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über *http://dnb.d-nb.de* abrufbar.

ISBN:
Print 978-3-96009-163-9
PDF 978-3-96010-479-7
ePub 978-3-96010-480-3
mobi 978-3-96010-481-0

1. Auflage 2021

Wieblinger Weg 17
69123 Heidelberg

Dieses Buch erscheint in Kooperation mit O’Reilly Media, Inc. unter dem Imprint »O’REILLY«. O’REILLY ist ein Markenzeichen und eine eingetragene Marke von O’Reilly Media, Inc. und wird mit Einwilligung des Eigentümers verwendet.

Hinweis:
Dieses Buch wurde auf PEFC-zertifiziertem Papier aus nachhaltiger Waldwirtschaft gedruckt. Der Umwelt zuliebe verzichten wir zusätzlich auf die Einschweißfolie.

Schreiben Sie uns:
Falls Sie Anregungen, Wünsche und Kommentare haben, lassen Sie es uns wissen: *kommentar@oreilly.de*.

5 4 3 2 1 0

Inhalt

Vorwort

PowerShell begann vor 15 Jahren als Windows Powershell. Heute ist diese Shell plattformübergreifend auf Windows, Linux und macOS verfügbar und wird begeistert von einer immer größeren Anwendercommunity genutzt.

In diesem Buch werden Sie Schritt für Schritt erfahren, was PowerShell eigentlich ist und was diese Shell alles für Sie tun kann. Ob Sie Administrator »on-premise« sind und lokale Server betreuen, Ihr Unternehmen in der Cloud verwalten oder ob Sie heterogene Umgebungen »unter einen Hut« bringen wollen – PowerShell bietet die Werkzeuge dafür.

Aber auch zu Hause und in der Schule lässt es sich hervorragend einsetzen: PowerShell ist es nämlich egal, ob Sie damit eine Unternehmens-IT administrieren, Ihre Philips-Hue-Homeautomation verwalten oder Sonnenkollektoren steuern. Ob Sie den eigenen NAS-Server auf dem Dachboden sichern, automatisiert Dateien aus dem Internet laden oder vielleicht bloß Matherätsel knacken wollen.

Und genau deshalb ist PowerShell auch in Ausbildung und Schulen spannend: Kaum eine andere kostenlose Programmierumgebung und Shell unterstützt so viele

Programmierkonzepte auf so vielen Plattformen – angefangen von Befehlsaufrufen über moderne Programmiertechniken wie den Einsatz von Funktionen und Objekten bis hin zu nativer Klassenunterstützung, Vererbung und universellem Netzwerk-Remoting. Kaum eine andere Umgebung bietet so unmittelbares Feedback wie PowerShell. Ideal also, um im Unterricht und in der Ausbildung anhand von nachvollziehbaren Praxisbeispielen die Funktionsweisen moderner IT auszuprobieren und zu vertiefen.

Dieses Buch bietet zahlreiche Praxisbeispiele aus den unterschiedlichsten Einsatzbereichen und demonstriert schrittweise die vielfältigen Möglichkeiten der PowerShell.

So schlüpfen Sie zu Anfang des Buches in die Rolle des einfachen PowerShell-Anwenders. Sie generieren mit Einzeilern Excel- und HTML-Reports und können mit wenigen Schritten Cloud-Systeme steuern oder Notenblätter aus dem Internet herunterladen und als PDF exportieren.

Von diesen ersten Fingerübungen inspiriert lernen Sie den Minimal-Wortschatz der PowerShell kennen und fügen mehrere Befehle zu eigenen größeren Automationslösungen zusammen.

Die Anwendungsbeispiele werden immer komplexer, und Sie lernen schrittweise alle weiteren wichtigen Konzepte der PowerShell kennen. Dazu gehören Remotezugriffe, direkte Betriebssystemaufrufe, das Erstellen grafischer Oberflächen sowie eigene PowerShell-Befehle und -Module.

Am Ende dieses Buches beherrschen Sie dann eine der modernsten Automationssprachen, die für Windows, Linux und macOS kostenfrei zur Verfügung stehen, und haben

sich privat wie beruflich vom einfachen Anwender zum versierten IT-Automatisierer qualifiziert.

Damit Sie auf dieser Reise nicht allzuviel eintippen müssen, automatisiert PowerShell in diesem Buch auf Wunsch natürlich auch das Eintippen der Beispiele für Sie. Schon im ersten Kapitel lernen Sie den passenden Befehl kennen: durch Eingabe der jeweiligen Listingnummer fügt PowerShell den Quellcode aus dem Buch automatisch ein.

Wie Sie dieses Buch nutzen

Dieses Buch setzt keinerlei Grundkenntnisse voraus, wenn Sie von vorn zu lesen beginnen – und das ist auch empfehlenswert. Die Kapitel bauen aufeinander auf. Am Anfang jedes Kapitels finden Sie eine kurze Zusammenfassung, falls es einmal eilig ist.

Die PowerShell-Beispiele im Buch sind jeweils in einer anderen Schriftart formatiert. Damit Sie leichter erkennen, welche Eingaben von Ihnen erwartet werden, wird bei allen Eingaben die PowerShell-Eingabeaufforderung `PS>` (einschließlich der Leerstelle hinter dem >) vorangestellt. Diese Eingabeaufforderung kann bei Ihnen auch anders aussehen und sollte in den Beispielen natürlich nicht mit eingegeben werden.

Achtung

Bitte verwenden Sie die Begleitmaterialien immer im Kontext des entsprechenden Buchkapitels. Viele der Beispiele funktionieren nur, wenn Sie die entsprechenden Vorarbeiten im Kapitel beachtet haben, oder können auch unerwartete Resultate liefern, wenn man die Beispiele aus dem Zusammenhang des Kapitels reißt.

Noch mehr Unterstützung

Falls trotz aller Sorgfalt einmal Fragen offenbleiben oder Sie weitere Ideen und Themenwünsche haben, besuchen Sie einfach das interaktive Leserforum zu diesem Buch: *https://github.com/TobiasPSP/OReilly/discussions*

Damit bleibt mir an dieser Stelle nur noch, Ihnen viel Spaß zu wünschen bei der Lektüre dieses Buchs und bei der Arbeit mit der faszinierenden PowerShell! Ich würde mich freuen, von Ihnen im Diskussionsforum zu hören.

Herzlichst,

Tobias Weltner

Kapitel 1
PowerShell: Erste Schritte

In diesem Kapitel:

Ausführlich werden in diesem Kapitel die folgenden Aspekte erläutert:

- **Windows PowerShell:** In Windows integrierte PowerShell, die auf dem klassischen *.NET Framework Version 4.5* oder höher basiert und auch künftig für Automationsaufgaben im Windows-Umfeld eingesetzt werden kann. Der Startbefehl lautet `powershell.exe`, und die aktuelle Version ist 5.1.

- **PowerShell:** Neuartige, plattformunabhängige PowerShell, die als portable Anwendung bei Windows parallel zur *Windows PowerShell* verwendet werden kann und auch auf Linux, macOS und weiteren Betriebssystemen zur Verfügung steht. Der Startbefehl lautet `pwsh.exe`. Diese PowerShell beruht auf dem plattformunabhängigen neuen *.NET Framework Core*, das weitgehend kompatibel zum klassischen .NET Framework ist. Die *PowerShell* ist im Gegensatz zur *Windows PowerShell* quelloffen (Open Source).
- **Autovervollständigung:** Mit einigen Tastendrücken kann man sich bei der Eingabe von Befehlen, Parametern und Argumenten Tipparbeit sparen: ↹ vervollständigt Eingaben. Drücken Sie die Taste mehrmals, zeigt PowerShell bei jedem Drücken einen weiteren Vorschlag. ⇧ + ↹ geht in der Reihenfolge wieder einen Schritt zurück. Mit Strg + Leertaste werden Eingabevorschläge als vollständige Auswahlliste präsentiert. Editoren zeigen dazu ein IntelliSense-Menü an. In der Konsole erscheint ein textbasiertes Auswahlfeld. Unabhängig von den Autovervollständigungsvarianten können Sie mit ↑ und ↓ frühere Eingaben aus Ihrer Befehlshistorie anzeigen lassen. Diese Liste ist anfangs leer und wächst mit der Verwendung der Konsole.
- **Zeilen löschen und Befehlsabbruch:** Um die gesamte aktuelle Zeile zu löschen, drücken Sie Esc. Um einen Befehl abzubrechen, drücken Sie Strg + C.
- **Groß- und Kleinschreibung:** PowerShell selbst unterscheidet bei Befehlsnamen und Parametern nicht zwischen Groß- und Kleinschreibung. Ob die

Groß- und Kleinschreibung bei Befehlsargumenten (wie zum Beispiel Pfadnamen oder anderen Angaben) wichtig ist, hängt vom jeweiligen Befehl und dem verwendeten Betriebssystem ab. Bei Kennworteingaben beispielsweise kommt es natürlich immer auf die richtige Groß- und Kleinschreibung an.

Achtung

Wenn Sie es eilig haben und die Grundlagen der PowerShell schon kennen, dürfen Sie diesen Einleitungsteil selbstverständlich überspringen. Sie sollten aber in jedem Fall wenigstens den Abschnitt »Zusammenfassung« am Ende dieses Kapitels beachten. Dort werden wichtige Grundeinstellungen besprochen, die die Voraussetzung für viele Beispiele in den folgenden Kapiteln sind.

PowerShell ist eine verblüffend flexible und machtvolle plattformunabhängige Automationssprache, die mit geringem Aufwand ein enormes Spektrum von Aufgaben automatisieren kann.

Dazu zählen typische IT-Administrationsaufgaben ebenso wie völlig andere Einsatzbereiche aus Mathematik, Forschung und Lehre, in der Büroautomation und nicht zuletzt in Hobby und Tüftelei: Schon im nächsten Kapitel werden wir uns kurz der Musikkomposition und Steuerung von MIDI-Musikinstrumenten widmen, und in *Kapitel 13* erfahren Sie zum Beispiel, wie PowerShell sogar Lampen und Steckdosen in Ihrem Zuhause fernsteuert. Die Grundlagen und Strategien sind dabei indes immer dieselben.

Bei all diesen Beispielen geht es also ausschließlich um zweierlei: kurzweilig und verständlich möglichst viele

Einsatzszenarien der PowerShell vorzustellen, um Ideen zu wecken und eine breite Leserschicht anzusprechen, und die stets gleichen allgemeinen Mechanismen zu verstehen, die dabei zum Einsatz kommen.

Denn mit dem Wissen, das Sie beispielsweise in der Musikkomposition oder auch bei der Fernsteuerung von Elektrogeräten benötigen, können Sie natürlich auch Server aufsetzen oder Drittanbietersoftware für Backup-Lösungen steuern – und umgekehrt. Hier wird deutlich, dass PowerShell ebenfalls eine ideale Plattform für Ausbildung und Schule ist, denn alle modernen IT-Grundlagen lassen sich damit anschaulich und unterhaltsam vermitteln.

Zunächst aber muss PowerShell vollständig eingerichtet werden, und genau darum geht es in diesem ersten Kapitel. Dabei werden auch einige wichtige Sicherheitseinstellungen besprochen, und der Unterschied zwischen *PowerShell* und *Windows PowerShell* wird Ihnen schnell klar werden.

Danach folgt ein kleiner Exkurs zu den wichtigsten kostenlosen PowerShell-Tools, mit denen Sie Ihren Computer in eine moderne PowerShell-Entwicklungsumgebung verwandeln.

Zum Abschluss des Kapitels erfahren Sie, wie die vielen Hundert Skriptbeispiele in diesem Buch ohne Tipparbeit direkt von PowerShell aus dem Internet heruntergeladen und ausgeführt werden und wie offengebliebene Fragen unmittelbar an den Autor gerichtet werden können.

Hinweis

In diesem Buch wird das Wort »PowerShell« immer dann kursiv gesetzt, wenn spezifisch die neue

plattformübergreifende *PowerShell* gemeint ist. Der nicht kursiv gesetzte Begriff »PowerShell« bezieht sich allgemein auf die Sprache PowerShell und gilt sowohl für *Windows PowerShell* als auch für *PowerShell*.

Schon in diesem Kapitel wird Ihnen bereits PowerShell-Code begegnen, der aber hier als reines »Werkzeug« für die Einrichtung Ihres Computers dient. Sie können den Code natürlich bereits neugierig mustern, aber bitte brüten Sie nicht zu lange über einzelnen Codebeispielen. Es kommt in diesem Kapitel nur auf die Ergebnisse der Codebeispiele an, nicht auf den Code selbst und seine Mechanismen. Diese stehen ab dem zweiten Kapitel im Vordergrund und werden dort dann auch systematisch besprochen.

PowerShell installieren

PowerShell ist plattformunabhängig und kann auf verschiedenen Betriebssystemen ausgeführt werden, zum Beispiel Windows, Linux oder macOS.

Historisch ist PowerShell allerdings als Teil des Windows-Betriebssystems entstanden und dort als sogenannte *Windows PowerShell* immer automatisch enthalten. Seit 2016 ist diese Fassung in Version 5.1 fertiggestellt, wird auf diesem Stand gehalten und weiter von Microsoft gewartet.

Alle Entwicklungsarbeit geht seit 2016 in die neue, plattformunabhängige *PowerShell*, bei Drucklegung dieses Buchs in Version 7.2, die bei allen Betriebssystemen zuerst nachinstalliert werden muss. Wie das geschieht, erfahren Sie jetzt, und zwar in separaten Abschnitten für die jeweiligen Betriebssysteme.

Windows-Betriebssystem

In allen Windows-Betriebssystemen ab Server 2008 und Windows 7 ist die PowerShell bereits fester Bestandteil – genauer gesagt die *Windows PowerShell*. Hier könnten Sie also sofort loslegen, sollten aber wenigstens prüfen, ob Ihre *Windows PowerShell* auf dem aktuellen Stand ist (dazu gleich mehr).

Noch empfehlenswerter ist es, zusätzlich zur integrierten *Windows PowerShell* auch die neue *PowerShell* hinzuzuinstallieren. Beide laufen friedlich parallel nebeneinander, und so können Sie sich ein eigenes Bild von den Unterschieden machen.

Immer vorhanden: Windows PowerShell

Sie starten die *Windows PowerShell* mit dem Befehl `powershell.exe` (oder kurz `powershell`) – wie jedes andere Programm auch. Drücken Sie zum Beispiel [⊞] + [R], um das Dialogfeld *Ausführen* zu öffnen (siehe Abbildung 1.1), und geben den Befehl `powershell` ein. Dann klicken Sie auf *OK*.

In der Taskleiste erscheint ein blaues Symbol, das Sie per Rechtsklick und *An Taskleiste anheften* am besten dauerhaft dort anpinnen. Ein Rechtsklick auf das Symbol öffnet die sogenannte Sprungliste, sozusagen das Cockpit der *Windows PowerShell* (siehe Abbildung 1.2).

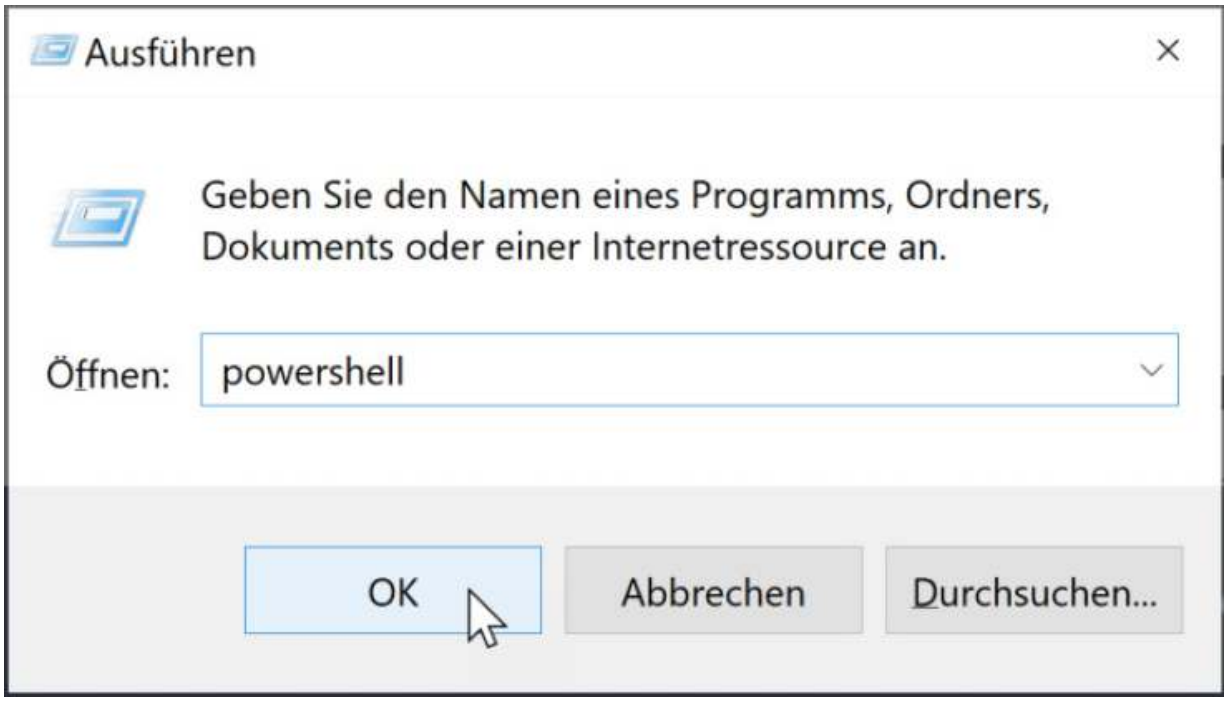


Abbildung 1.1: Windows PowerShell starten

Von hier aus können Sie die PowerShell zum Beispiel mit vollen Administratorrechten starten, und auch der in *Windows PowerShell* integrierte Editor *ISE* (*Integrated Script Environment*) steht hier bereit.

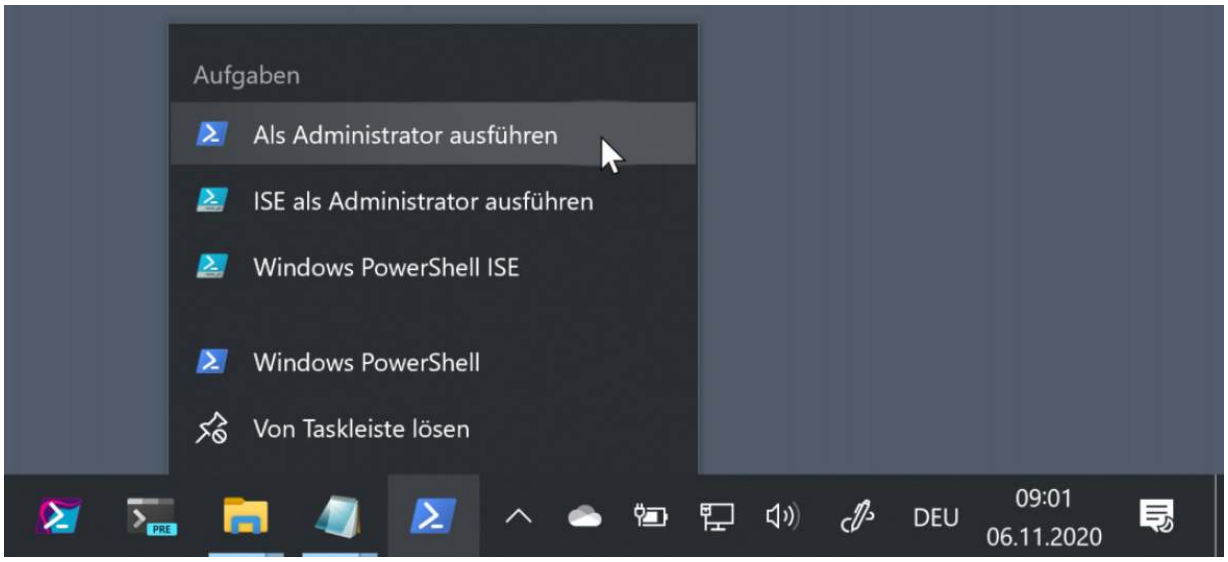


Abbildung 1.2: Sprungliste der Windows PowerShell

Die *Windows PowerShell* selbst erscheint als blaue oder schwarze Konsole, in die Sie Befehle eingeben können (siehe Abbildung 1.3). Auch wenn es Ihnen bereits in den Fingern juckt: Verschieben Sie Ihre Experimente bitte noch einen Moment – was Sie alles mit PowerShell tun können, veranschaulicht der Rundkurs im nächsten Kapitel.

Lassen Sie uns zunächst alles einsatzbereit machen.

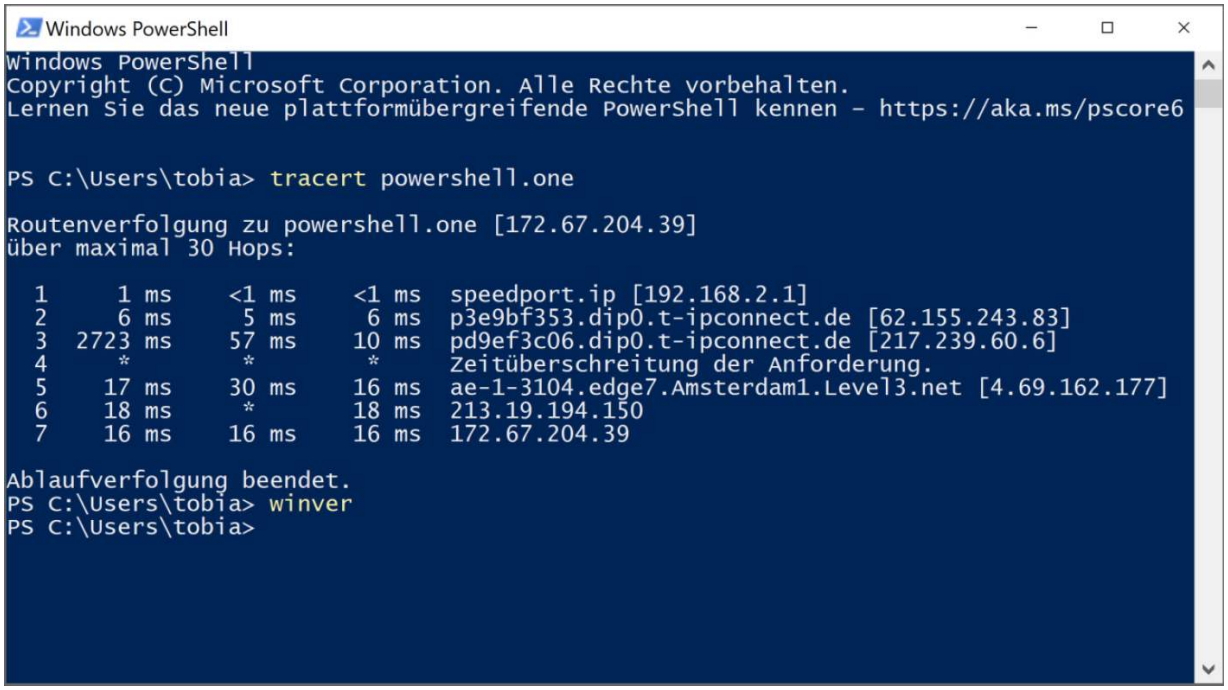


Abbildung 1.3: Die in Windows integrierte Windows PowerShell

Ist Ihre Windows PowerShell noch aktuell?

Falls Sie Windows 10 oder Server 2016 (und besser) verwenden, ist alles gut: Diese Betriebssysteme enthalten *Windows PowerShell* in Version 5.1, der letzten und ausgereiften Version. Diese Version wird über die gesamte Laufzeit von Windows 10 weitergepflegt und automatisch aktualisiert.

Wer ein älteres Windows-Betriebssystem nutzt, muss *Windows PowerShell* manuell aktualisieren. Das ist sinnvoll, weil die Codebeispiele dieses Buchs die