

# Einleitung

PowerShell ist als Automationssprache entwickelt worden: IT Administratoren und ambitionierte Computeranwender können damit Routineaufgaben automatisch erledigen. Das spart Zeit und lästige Handgriffe und ist aus der modernen IT nicht mehr wegzudenken.

Darüber hinaus ist PowerShell inzwischen aber auch eine sehr effektive Programmiersprache geworden, mit der man ohne komplexes Vorwissen und in nur wenigen Zeilen Code kleine Programme und Tools schreiben kann. Das ist für nahezu jeden ambitionierten Computeranwender interessant, nicht mehr lediglich für IT-Profis.

Und spätestens mit PowerShell 6 laufen PowerShell Skripte nun sogar nicht mehr nur auf Windows, sondern auch auf Linux und MacOS. Das macht PowerShell Fachwissen noch erheblich vielseitiger.

Dieses Buch beginnt bei null und führt Sie Schritt für Schritt in alle wichtigen Grundlagen der PowerShell ein. Ob Sie PowerShell beruflich nutzen oder privat, quer über das Netzwerk auf Hunderten von Computern Programme konfigurieren oder bloß auf dem eigenen Notebook Urlaubsbilder umbenennen wollen: Dieses Buch liefert zahlreiche Praxisbeispiele und erklärt die eingesetzten Techniken so, dass Sie die Beispiele leicht an eigene Bedürfnisse anpassen können.

Sieben didaktisch aufeinander aufbauende Teile führen Sie ohne notwendiges Vorwissen in die Möglichkeiten der PowerShell ein.

Im ersten Kapitel erfahren Sie, wie PowerShell auf dem Computer eingerichtet wird und welche essenziellen Einstellungen und ersten Schritte wichtig sind, um PowerShell Befehle ausführen zu können.

In **Teil A** wird die PowerShell als praktische interaktive Konsole betrachtet – schnelle Ergebnisse mit ganz wenigen Eingaben. Sie entdecken, welche Befehle die PowerShell versteht, wie man schnell die benötigten Befehle findet und neue Befehle nachrüsten kann. In diesem Teil wird auch das universelle Laufwerkkonzept der PowerShell vorgestellt, mit dem beinahe alles – vom Dateisystem über Zertifikate und Registrierung bis hin zu Datenbanken und Benutzerverwaltung – als Laufwerk gesehen werden kann, auf dem man Informationen findet.

**Teil B** zeigt dann, wie aus mehreren Einzelbefehlen und der PowerShell Pipeline komplexere und mächtigere Automationskripte entstehen, mit denen sich vielfältige Aufgaben vollständig automatisieren lassen.

## Einleitung

**Teil C** beschäftigt sich mit Operatoren, die oft unterschätzt werden. Hier erfahren Sie, wie man intelligente Entscheidungen (Bedingungen) formuliert. Einen besonderen Schwerpunkt bilden die Textoperatoren und sogenannten »regulären Ausdrücke«, mit denen PowerShell die für Sie wichtigen Informationen aus Texten, Logdateien und von Webseiten fischen kann.

PowerShell kommt bereits mit Hunderten eigener Befehle, doch bietet es vor allem eine Schnittstelle zu den unzähligen Betriebssystemfunktionalitäten, die im .NET Framework bereitgestellt werden. In **Teil D** erfahren Sie, wie PowerShell auf all diese Funktionen zugreifen kann. So erhalten Ihre PowerShell Skripte Zugriff auf genau die gleichen Möglichkeiten, die auch Anwendungsentwicklern anderer .NET Sprachen zur Verfügung stehen, und können von DNS-Auflösung über Sprachausgabe bis hin zu eigenen Fenstern und Oberflächen alles nachrüsten, was noch nicht über vorgefertigte PowerShell-Cmdlets erreichbar war.

Weil PowerShell eine dynamische Sprache ist, kann man den Befehlssatz sehr leicht erweitern. **Teil E** erklärt zuerst, wie Ihr PowerShell Code zu einem PowerShell-Befehl wird, der dann in allen PowerShells zur Verfügung steht. Danach erfahren Sie, wie PowerShell-Module erzeugt werden: Mit ihnen lassen sich Befehlssammlungen an Kollegen und Kunden weitergeben. Und schließlich wird das brandneue PowerShellGet vorgestellt, über das Module automatisiert verteilt werden können.

Wenn PowerShell dabei einmal nicht genau das tut, was Sie sich vorstellen, hilft **Teil F** und zeigt, wie Fehler im Code erkannt und »behandelt« werden können. Dazu zählt auch das Mitprotokollieren von Fehlern und ungewöhnlichen Zuständen.

In **Teil G** wird zum Abschluss das »PowerShell Remoting« vorgestellt. Mit diesem extrem mächtigen Feature kann PowerShell-Code remote auf einem oder vielen anderen Computern parallel ausgeführt werden. Weil PowerShell inzwischen auch auf Linux und MacOS ausführbar ist, kann man darüber unter anderem auch Informationen in heterogenen Welten austauschen.

Dieses Buch ist eine überarbeitete und gekürzte Fassung des über 1.100 Seiten starken »Windows PowerShell 5: Windows Automation für Einsteiger & Profis«, das sich auf die für Einsteiger wichtigen Inhalte fokussiert.

## 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 bei der Arbeit mit diesem Buch Fragen auftauchen oder Sie Anregungen haben, besuchen Sie mich: <http://www.powershell.com>. Oder senden Sie mir eine Nachricht an meine Mail adresse [tobias.weltner@email.de](mailto:tobias.weltner@email.de).

Bevor ich Ihnen viel Spaß und Erfolg mit PowerShell wünsche, geht noch ein großes Danke schön an meine Lektorin Ariane Hesse und die Korrektorin Sibylle Feldmann, die dieses Buch mit allergrößtem Sachverstand und mit Sorgfalt begleitet haben.

Herzlichst Ihr

Dr. Tobias Weltner