

# Vorwort

Ein Script ist ein kleines Stück Code zur Lösung eines konkreten Problems oder zur Automatisierung einer lästigen Aufgabe. Sie brauchen zur Entwicklung eines Scripts weder eine Entwicklungsumgebung noch einen Compiler – ein Editor genügt. Beim Scripting ist Minimalismus die Devise. Es geht darum, mit minimalem Mittel maximale Wirkung zu erzielen. Salopp formuliert:

**Scripting ist die Kunst,  
IT-Probleme in zehn Zeilen Code zu lösen.**

In meinem Arbeitsumfeld sind Scripts allgegenwärtig. Ich verwende Scripts, um den Energiesparmodus meines Notebooks einzustellen, um aus Markdown-Dateien die PDF-Datei für den Druck dieses Buchs zu erzeugen, um auf Servern das Backup zu automatisieren, um E-Books mit Wasserzeichen auszustatten, um in einer Webapplikation neue Kunden einzurichten, um Dutzende von Datenbanken nach logischen Fehlern zu durchsuchen, um virtuelle Maschinen für den Unterricht zu erstellen, um Fotos zu sortieren usw.

## Bash, PowerShell oder Python?

Die »klassische« Script-Sprache ist die Bash. Ihr Name *Bourne Again Shell* ist ein Wortspiel. Der Vorgänger war die unter Unix verbreitete Bourne-Shell. Für Linux war die Lizenz der Bourne-Shell ungeeignet, weswegen ein neues Projekt entstand. Heute ist die Bash die dominierende Linux-Shell, sowohl im interaktiven Betrieb zur Ausführung von Kommandos als auch zur Script-Programmierung. macOS setzt auf die weitgehend kompatible Zsh, die auch unter Linux immer mehr Anhänger findet.

Die Bash ist populär, Ihre antiquierte Syntax gewinnt aber keine Schönheitspreise. Im Gegenteil, Bash-Scripts sehen mitunter grauenhaft aus. Insofern kann man verstehen, dass Microsoft gar nicht erst versucht hat, die Bash für Windows zu adaptieren. Stattdessen hat Microsoft mit der PowerShell Grundideen klassischer Unix-Shells mit den Konzepten objektorientierter Programmiersprachen zu einer vollkommen neuen Sprache kombiniert. Das ist überraschend gut gelungen! Nicht ohne Grund schwören Windows-Administratorinnen und -Administratoren auf die PowerShell.

Python ist eigentlich keine typische Script-Sprache. Je nach Einsatzzweck ist Python die Basis für KI-Entwicklungen, ein Tool für (Natur-)Wissenschaftler oder die erste Sprache für den Einstieg in die Programmierung. Es gibt wohl keine andere Sprache, die so universell eingesetzt wird! Die Eleganz der Python-Syntax und das schier uner-

schöpfliche Angebot von Erweiterungsmodulen haben dazu geführt, dass Python auch zur Systemadministration, zur Umwandlung von Dateien zwischen verschiedenen Formaten, als Datenbank-Tool oder zur Hardware-Steuerung (Raspberry Pi) verwendet wird. Python spielt seine Stärken umso mehr aus, je komplexer die Aufgabenstellung ist.

## Über dieses Buch

Weil es nicht die eine perfekte Script-Sprache gibt, startet dieses Buch nach der Einleitung mit Crash-Kursen für die Sprachen Bash, PowerShell und Python. Wenn Sie möchten, können Sie sich für den Start auf eine dieser drei Sprachen konzentrieren und Ihren Sprachwortschatz nach und nach vergrößern.

Teil II des Buchs stellt Werkzeuge und Arbeitstechniken vor, die Sie in Scripts typischerweise verwenden: Dazu zählen Kommandos zur Verarbeitung von Textdateien, CmdLets zur Anwendung regulärer Ausdrücke und Funktionen zum Umgang mit JSON- und XML-Dateien. Ich zeige Ihnen, wie Sie Scripts regelmäßig und automatisiert ausführen, wie Sie dank SSH Code auch auf anderen Rechnern ausführen bzw. Dateien dorthin kopieren und wie Sie dank Git die Versionsverwaltung Ihres Codes und Scripting kombinieren. Viele praktische Beispiele runden das Informationsangebot an.

In Teil III geht es schließlich um konkrete Anwendungen: Zu den wichtigsten Themen zählen Backups, Bildverarbeitung, Web Scraping, die Nutzung von REST-APIs, Datenbanken, die Nutzung der Cloud und die Administration virtueller Maschinen.

Ich habe mich bemüht, das Buch so gut wie möglich nach einem Baukastensystem zusammenzusetzen. Sie müssen also nicht alle Kapitel linear lesen, sondern können gezielt nach den Informationen suchen, die Ihnen gerade wichtig sind. Alle Kapitel von Teil II und III beginnen mit einem kurzen Informationskasten, der die Voraussetzungen für das Kapitel zusammenfasst.

## Scripting und KI-Tools

In den letzten Monaten hat ChatGPT einen Umbruch in vielen Berufen eingeleitet. Wie so oft hängt es von der Sichtweise ab, ob Sie das Glas halb voll oder halb leer sehen: KI-Tools sind heute alles andere als perfekt, aber sie sind definitiv (schon jetzt und noch mehr in den kommenden Jahren) ein Hilfsmittel, das die Software-Entwicklung revolutionieren wird.

Wenn Sie ChatGPT oder den GitHub Copilot in VSCode um ein PowerShell-Script bitten, das eine JSON-Datei lesen und die darin enthaltenen Daten in einer PostgreSQL-Datenbank speichern soll, wird es Ihnen innerhalb von Sekunden ein plausibel aussehendes Script ausspucken.

Im Idealfall funktioniert das Script auf Anhieb. Nach meinen Erfahrungen ist der Code aber selten optimal – umso weniger, je komplexer die Fragestellung ist. Oft enthält der Code eindeutige Fehler, z. B. nicht existierende Optionen oder Kommandos. Mitunter gibt es auch logische Fehler, die schwerer zu erkennen sind. Mit etwas Scripting-Erfahrung lassen sich die Probleme zumeist rasch beheben. Sie können das KI-Tool sogar ganz gezielt bitten, einen bestimmten Aspekt des Codes zu verbessern.

Mein persönlicher Eindruck ist, dass sich KI-Tools umso besser einsetzen lassen, je solider das vorhandene Vorwissen ist. Ein KI-Tool und eine erfahrene Entwicklerin, ein routinierter Programmierer – das ist aus meiner Sicht das Dream-Team der Zukunft. Dieses Buch vermittelt genau das dafür erforderliche Fundament, ohne sich mit allzu vielen enzyklopädischen Details aufzuhalten. Für das Programmieren der Zukunft müssen Sie nicht hundert Syntaxvarianten von regulären Ausdrücken auswendig kennen. Aber Sie müssen wissen, dass es reguläre Ausdrücke gibt, wie sie prinzipiell funktionieren und wo die Fallstricke sind. Genau da liegt der Fokus dieses Buchs.

### **Scripting als Kernkompetenz für effiziente IT-Arbeit**

Ganz egal, in welcher Sparte der IT-Industrie Sie tätig sind, an welchen Projekten Sie gerade arbeiten: Ein wenig Scripting-Know-how macht Sie in Ihrer Arbeit noch effizienter. Konzentrieren Sie sich auf das Wesentliche, überlassen Sie die lästigen Nebenaufgaben einem Script! Dabei wünsche ich Ihnen viel Erfolg!

Michael Kofler (<https://kofler.info>)

PS: Alle Beispieldateien zu diesem Buch finden Sie hier zum Download:

<https://rheinwerk-verlag.de/5672>