

# Einleitung

Jeder Programmierer kann Ihnen erzählen, wie er gelernt hat zu programmieren und sein erstes Programm geschrieben hat. Ich habe als Kind damit angefangen, als mein Vater für die Digital Equipment Corporation arbeitete, einem der bahnbrechenden Unternehmen des modernen Computerzeitalters. Mein erstes Programm schrieb ich auf einem Computer aus einem Bausatz, den mein Vater im Keller zusammengebastelt hatte. Er bestand lediglich aus einem nackten Motherboard mit einer Tastatur und einer Bildröhre als Monitor. Mein erstes Programm war ein einfaches Zahlenratespiel, das wie folgt ablief:

Ich habe mir eine Zahl ausgedacht! Versuche sie zu erraten: **25**  
Zu niedrig! Versuche es noch einmal: **50**  
Zu hoch! Versuche es noch einmal: **42**  
Richtig! Möchtest du noch einmal spielen? (ja/nein) **nein**  
Danke fürs Spielen!

Ich werde nie vergessen, wie stolz ich war, dass meine Familie ein Spiel spielte, das ich selbst geschrieben hatte und das tatsächlich so funktionierte wie beabsichtigt.

Diese erste Erfahrung hatte eine bleibende Wirkung auf mich. Es ist sehr befriedigend, etwas zu erschaffen, das einen Zweck erfüllt oder ein Problem löst. Heute schreibe ich Software, die wichtigere Bedürfnisse erfüllt als mein Programm aus Kindertagen, aber die Befriedigung, die ich daraus gewinne, ist immer noch die gleiche.

## Zielgruppe

Dieses Buch soll Sie so schnell wie möglich mit Python vertraut machen, sodass Sie eigene Programme schreiben können – beispielsweise Spiele, Datenvisualisierungen und Webanwendungen –, Ihnen aber auch Grundkenntnisse in Programmierung vermitteln, die Ihnen auch bei der Verwendung anderer Sprachen von Nutzen sind. Es richtet sich an Leser aller Altersgruppen, die noch nie in Python oder überhaupt noch nie programmiert haben. Wenn Sie die Grundlagen der Programmierung schnell erlernen wollen, sodass Sie sich interessanten Projekten zuwenden können, und wenn Sie Ihre neu erworbenen Kenntnisse an praxisnahen Problemen erproben wollen, so ist dies das richtige Buch für Sie. Es ist auch ideal für Lehrer der Sekundarstufe geeignet, die ihren Schülern eine projektbezogene Einführung in die Programmierung geben wollen. Wenn Sie einen Hochschulkurs besuchen und sich eine benutzerfreundlichere Einführung in Python wünschen als mit den dort behandelten Texten, kann Ihnen dieses Buch auch helfen, in diesem Kurs besser zurechtzukommen.

## Lernstoff

Dieses Buch soll Sie zu einem guten Programmierer im Allgemeinen und zu einem guten Python-Programmierer im Besonderen machen. Während ich Ihnen eine solide Grundlage in allgemeinen Programmierprinzipien gebe, eignen Sie sich auch gute Programmiergewohnheiten an. Nachdem Sie dieses Buch durchgearbeitet haben, sind Sie bereit, sich fortgeschrittenen Python-Techniken zuzuwenden, und können auch andere Programmiersprachen leichter erlernen.

Im ersten Teil dieses Buches lernen Sie grundlegende Programmierprinzipien kennen und erfahren, wie Sie Python-Programme schreiben. Diese Prinzipien sind die gleichen, die Sie auch bei fast allen anderen Programmiersprachen befolgen müssen. Sie lernen, welche verschiedenen Arten von Daten es gibt, wie Sie sie in Listen und Dictionaries speichern und wie Sie solche Zusammenstellungen von Daten rationell abarbeiten. Außerdem erfahren Sie, wie Sie mit `while`-Schleifen und `if`-Anweisungen dafür sorgen, dass je nachdem, welche Umstände vorliegen, unterschiedliche Codeblöcke ausgeführt werden, was eine wichtige Grundlage für die Automatisierung von Vorgängen ist.

Sie lernen, wie Sie Eingaben von Benutzern entgegennehmen, um Ihre Programme interaktiv zu machen; wie Sie die Programme so lange am Laufen halten, wie der Benutzer es wünscht; wie Sie Funktionen erstellen, um Teile Ihrer Programme wiederverwenden zu können, sodass Sie einen Codeblock für eine bestimmte Funktion nur einmal schreiben müssen und dann so oft ausführen können, wie Sie wollen; wie Sie dieses Prinzip mithilfe von Klassen auf komplexere Ver-

haltensweisen ausdehnen, sodass relativ einfache Programme auf viele verschiedene Situationen reagieren können; und wie Sie dafür sorgen, dass Ihre Programme mit auftretenden Fehlern umgehen können. Nachdem Sie all diese Grundlagen gründlich durchgearbeitet haben, schreiben Sie kurze Programme, um genau definierte Probleme zu lösen. Schließlich unternehmen Sie auch einen ersten Schritt auf dem Weg zum fortgeschrittenen Anfänger, indem Sie Tests für Ihren Code schreiben, sodass Sie Ihre Programme weiterentwickeln können, ohne befürchten zu müssen, dabei Fehler einzuführen. Alles, was Sie in Teil I lernen, bereitet Sie darauf vor, umfangreiche, anspruchsvolle Projekte durchzuführen.

In Teil II wenden Sie das, was Sie in Teil I gelernt haben, in drei großen Projekten an. Sie können diese Projekte in der Reihenfolge angehen, die Ihnen am besten liegt. Im ersten Projekt (Kapitel 12 bis 14) erstellen Sie ein stark an Space Invaders angelehntes Ballerspiel namens *Alien Invasion* mit Levels von immer höherem Schwierigkeitsgrad. Wenn Sie dieses Projekt durchgearbeitet haben, sind Sie gut gerüstet, um eigene 2D-Spiele zu entwickeln.

Im zweiten Projekt (Kapitel 15 bis 17) geht es um Datenvisualisierung. Datenforscher versuchen, die Unmengen an Informationen, die ihnen zur Verfügung stehen, zu deuten, indem sie verschiedene Visualisierungstechniken darauf anwenden. In diesen Kapiteln arbeiten Sie mit Datenmengen, die Sie durch Code erzeugen, aus Onlinequellen herunterladen oder automatisch von Ihren Programmen herunterladen lassen. Nachdem Sie dieses Projekt durchgearbeitet haben, können Sie eigene Programme schreiben, die große Datenmengen durchforsten und grafische Darstellungen der darin gespeicherten Informationen erstellen.

Im dritten Projekt (Kapitel 18 bis 20) richten Sie eine kleine Webanwendung namens Learning Log ein, mit der es möglich ist, ein Tagebuch über Erlerntes zu einem bestimmten Thema zu führen. Sie erfahren hier, wie Sie getrennte Tagebücher für unterschiedliche Fachgebiete führen und wie Sie anderen Benutzern gestatten, ein Konto anzulegen und ihre eigenen Tagebücher zu schreiben. Außerdem lernen Sie, das Projekt öffentlich bereitzustellen, sodass es überall online zugänglich ist.

## Onlinematerial

Das gesamte Zusatzmaterial zu diesem Buch finden Sie online auf <http://www.dpunkt.de/python3crashcourse> und [http://ehmatthes.github.io/pcc\\_2e/](http://ehmatthes.github.io/pcc_2e/) (auf Englisch). Dazu gehört Folgendes:

**Anleitungen zur Einrichtung** Diese Anleitungen sind identisch mit denjenigen in diesem Buch, enthalten aber Links zu allen einzelnen Teilen. Schauen Sie hier nach, wenn Sie Probleme bei der Einrichtung haben.

**Aktualisierungen** Python wird ebenso wie andere Sprachen ständig weiterentwickelt. Ich pflege einen umfangreichen Satz von Aktualisierungen. Sollte also irgendwelcher Code nicht funktionieren, so schauen Sie hier nach, ob sich Anweisungen geändert haben.

**Lösungen zu den Übungen** Wenn Sie sich an den Übungen in den Abschnitten mit dem Titel »Probieren Sie es selbst aus!« versuchen, sollten Sie sich zunächst ausreichend Zeit nehmen, um sie selbstständig zu bearbeiten. Sollten Sie aber wirklich nicht mehr weiterwissen, können Sie zu den meisten dieser Übungen Lösungen online nachschlagen.

**Spickzettel** Auch ein ganzer Satz von Spickzetteln zum Herunterladen steht online zur Verfügung. Sie sind zum schnellen Nachschlagen aller wichtigen Prinzipien geeignet.

## Warum Python?

Jedes Jahr überlege ich, ob ich weiterhin Python verwenden oder auf eine andere Programmiersprache umsteigen soll, vielleicht eine neuere, bleibe aber jedes Mal bei Python, und das aus vielen Gründen. Python ist eine unglaublich effiziente Sprache: Python-Programme können in weniger Zeilen mehr erledigen, als das bei anderen Programmiersprachen der Fall ist. Die Syntax von Python hilft Ihnen auch, »sauberen« Code zu schreiben, der sich leicht lesen, leicht korrigieren und leicht erweitern lässt.

Python wird für viele Zwecke eingesetzt: für Spiele, für Webanwendungen, zum Lösen von Problemen in der Geschäftswelt und zur Entwicklung interner Werkzeuge für alle möglichen Arten von Unternehmen. Auch in Wissenschaft und Forschung ist Python stark vertreten.

Einer der wichtigsten Gründe dafür, dass ich bei Python bleibe, ist die Python-Community, der sehr unterschiedliche und sehr offene Menschen angehören. Die Community ist für das Programmieren sehr wichtig, da Programmierung keine Aufgabe für Einzelkämpfer ist. Selbst die erfahrensten Programmierer brauchen hin und wieder Rat von anderen, die bereits ähnliche Probleme gelöst haben. Eine gut vernetzte und hilfsbereite Community ist zur Problemlösung unverzichtbar, und die Python-Community hilft Menschen wie Ihnen, die Python als erste Programmiersprache lernen, gern weiter.

Python ist eine großartige Sprache. Beginnen wir also damit, sie zu erlernen!