



1

Einleitung

Warum soll man das Programmieren erlernen?

Programmieren fördert die Kreativität, das logische Denken und die Fähigkeit, Probleme zu lösen. Programmierer und Programmiererinnen haben die Möglichkeit, etwas aus dem Nichts zu erschaffen. Mithilfe der Logik bringen sie Programmstrukturen in eine Form, sodass ein Computer damit funktioniert. Und wenn die Dinge nicht ganz so gut funktionieren wie erwartet, können sie durch die Fähigkeit zur Problemlösung herausfinden, was schiefgelaufen ist. Programmieren macht Spaß, ist manchmal schwierig (gelegentlich frustrierend), und die Fähigkeiten, die man dabei erwirbt, können sowohl in der Schule als auch bei der Arbeit nützlich sein – selbst wenn Dein Berufsleben später nichts mit Computern zu tun haben sollte.

Außerdem ist das Programmieren ein prima Zeitvertreib bei miesem Wetter.

1.1 Warum Python?

Python ist eine leicht zu erlernende Programmiersprache, die für den Programmieranfänger einige nützliche Eigenschaften hat. Der Code ist im Vergleich zu anderen Programmiersprachen recht einfach zu lesen, und es gibt eine interaktive Shell, in die man seine Programme eingeben und sehen kann, wie sie laufen. Zusätzlich zu seiner einfachen Programmstruktur und seiner interaktiven Shell hat Python einige Merkmale, die den Lernvorgang sehr bereichern und mit denen Du einfache Animationen zum Erstellen Deiner eigenen Spiele zusammenbauen

kannst. Eines davon ist das Modul `turtle`, das von Turtle Graphics inspiriert wurde (das in den 1960er-Jahren von der Programmiersprache Logo verwendet wurde) und für Lernzwecke geschaffen wurde. Ein weiteres Modul ist `tkinter`, mit dem man auf das Tk GUI Toolkit zugreifen kann, um damit ziemlich einfach ein bisschen anspruchsvollere Grafiken und Animationen zu erstellen.

1.2 Wie man das Programmieren lernt

Wie bei allem, was man zum ersten Mal probiert, ist es am besten, mit den Grundlagen anzufangen. Beginne daher mit den ersten Kapiteln, und blättere nicht voller Ungeduld zu den Kapiteln weiter hinten. Niemand kann beim ersten Mal, wenn er ein Musikinstrument in die Hand nimmt, im Sinfonieorchester mitspielen. Flugschüler fliegen auch nicht, bevor sie die grundlegenden Steuerelemente verstanden haben, und Turner kriegen (normalerweise) beim ersten Versuch keinen Salto rückwärts hin. Wenn Du zu Anfang zu ungeduldig bist, haben die grundlegenden Prinzipien keine Zeit, sich richtig in Deinem Kopf festzusetzen. Dir wird dann der Inhalt der Kapitel weiter hinten viel komplizierter vorkommen, als er in Wirklichkeit ist.

Während Du dieses Buch durchliest, solltest Du jedes Beispiel selbst ausprobieren, um zu sehen, wie es funktioniert. Am Ende der meisten Kapitel gibt es auch Programmier-Puzzles, die Du lösen kannst. Sie werden Deine Programmierfähigkeiten fördern. Denke immer daran: Je besser Du die Grundlagen verstanden hast, desto leichter werden Dir die komplizierteren Konzepte später vorkommen.

Wenn Dich etwas frustriert oder Dir zu schwierig vorkommt, hier ein paar Ratschläge, die ich sehr hilfreich finde:

- Teile das Problem in kleinere Teile auf. Versuche zu verstehen, was ein kleiner Teil des Codes macht, oder denke nur an einen kleinen Teil einer komplexen Stelle. (Konzentriere Dich lieber auf einen kleinen Teil des Codes, statt alles auf einmal verstehen zu wollen.)
- Wenn das alles nichts hilft, ist es manchmal am besten, wenn man es für eine Weile einfach liegen lässt. Schlafe drüber, und mache an einem anderen Tag weiter. Auf diese Weise lösen sich viele Probleme von allein – besonders Programmierprobleme.

1.3 Wer dieses Buch lesen sollte

Dieses Buch ist für jeden geschrieben, der sich für das Programmieren interessiert, ganz egal, ob man nun Kind oder Erwachsener ist, wenn man zum ersten Mal programmiert. Wenn man lernen will, wie man seine eigene Software schreibt, anstatt nur von anderen entwickelte Programme zu nutzen, ist *Python kinderleicht* ein toller Einstieg.

In den folgenden Kapiteln erfährst Du, wie man Python installiert, die Python-Shell startet, einfache Berechnungen anstellt, Text auf den Bildschirm bekommt und Listen erstellt. Du lernst, wie man einfache Fallunterscheidungen mit `if`-Anweisungen und `for`-Schleifen durchführt. (Und natürlich erfährst Du, was `if`-Anweisungen und `for`-Schleifen eigentlich sind!) Du erfährst, wie man Code mit Funktionen wiederverwendet. Du lernst die Grundlagen von Klassen und Objekten kennen und bekommst Beschreibungen der vielen in Python eingebauten Funktionen und Module.

Es gibt Kapitel über einfache und fortgeschrittene Turtle-Grafiken und über die Benutzung des Moduls `tkinter`, um auf dem Computerbildschirm zu zeichnen. Am Ende vieler Kapitel gibt es Programmier-Puzzles mit unterschiedlichen Schwierigkeitsgraden, die dabei helfen, das gerade Gelernte zu verfestigen. Sie bieten Dir auch die Möglichkeit, selbst kleine Programme zu schreiben.

Wenn Du Dir die Grundlagen des Programmierens angeeignet hast, wirst Du lernen, wie Du Deine eigenen Spiele schreiben kannst. Du wirst zwei grafische Spiele entwickeln und etwas über Kollisionsdetektion, Events und diverse Animationstechniken erfahren.

Die meisten Beispiele in diesem Buch benutzen die IDLE-Shell (*Integrated DeveLopment Environment*; integrierte Entwicklungsumgebung) von Python. IDLE bietet Syntax-Markierung, eine Kopieren- und Einfügen-Funktionalität (so, wie Du es von anderen Anwendungen kennst) und ein Editor-Fenster, in dem Du Deinen Code für den späteren Gebrauch speichern kannst. IDLE ist daher eine Entwicklungsumgebung zum Experimentieren und hat auch ein bisschen was von einem Text-Editor. Die Beispiele funktionieren genauso gut in der Standard-Konsole und in einem üblichen Text-Editor, aber die Syntax-Markierung und die benutzerfreundlichere Umgebung von IDLE helfen Dir, den Code schneller zu verstehen. Deshalb wird im ersten Kapitel erklärt, wie man IDLE einrichtet.

1.4 Was in diesem Buch steht

Hier ist ein kurzer Überblick, was Dich in den einzelnen Kapiteln erwartet:

- **Kapitel 2** ist eine Einführung in das Programmieren. Außerdem findest Du Anleitungen zur ersten Installation von Python.
- **Kapitel 3** führt einfache Berechnungen und Variablen ein.
- **Kapitel 4** erklärt einige der grundlegenden Python-Elemente, wie etwa Strings, Listen und Tupel.
- **Kapitel 5** bietet Dir einen Vorgeschmack auf das Modul `turtle`. Wir springen dabei von den Grundlagen des Programmierens zum Bewegen einer Schildkröte (engl. *turtle*, die aber hier die Form eines Pfeils hat) über den Bildschirm.
- **Kapitel 6** behandelt die Varianten der Bedingungen und `if`-Anweisungen, und **Kapitel 7** macht bei den `for`- und `while`-Schleifen weiter.

- In **Kapitel 8** beginnen wir mit der Benutzung und Erstellung von Funktionen, und in **Kapitel 9** geht es um Klassen und Objekte. Wir decken in diesen beiden Kapiteln so viel von den grundsätzlichen Prinzipien der Programmier-Techniken ab, dass wir in den weiteren Kapiteln zur Spiele-Entwicklung übergehen können. Von dort an wird es ein bisschen komplizierter.
- **Kapitel 10** stellt die meisten der eingebauten Funktionen von Python vor, und **Kapitel 11** macht mit ein paar Modulen (die im Prinzip Behälter voller nützlicher Funktionalität sind) weiter, die automatisch mit Python installiert wurden.
- **Kapitel 12** kehrt zum `turtle`-Modul zurück, da Du jetzt lernst, mit komplexeren Formen umzugehen. **Kapitel 13** geht zum Modul `tkinter` über – und damit zu fortgeschritteneren grafischen Kreationen.
- In den **Kapiteln 14 und 15** programmieren wir unser erstes Spiel, »Bounce!«, das auf dem Erlernten aus den vorigen Kapiteln aufbaut.
- In den **Kapiteln 16 bis 19** programmieren wir unser zweites Spiel: »Mr. Stick – Man rennt zum Ausgang.« In den Spieleentwicklungs-Kapiteln können die Dinge aus dem Ruder laufen. Wenn nichts mehr geht, lädst Du den Code von der Website zu diesem Buch (www.dpunkt.de/python) herunter und vergleichst Deinen Code mit den funktionierenden Beispielen von dort.
- Im **Nachwort** fassen wir das Gelernte mit einem Blick auf PyGame und andere beliebte Programmiersprachen zusammen.
- Zum Schluss sind im **Anhang** noch einmal alle Python-Schlüsselwörter genau erklärt, und im **Glossar** findest Du alle Definitionen der Programmierbegriffe, die in diesem Buch verwendet werden.

1.5 Die Website zum Buch

Wenn Du meinst, dass Du während des Lesens Hilfe brauchst, kannst Du die Website www.dpunkt.de/python aufsuchen, wo Du Downloads für alle Beispiele in diesem Buch und noch mehr Programmier-Puzzles findest. Du findest dort auch die Lösungen für alle Programmier-Puzzles in diesem Buch, falls Du nicht mehr weiter weißt oder Deine Programme überprüfen möchtest.

1.6 Viel Vergnügen!

Vergiss beim Durcharbeiten dieses Buches nie, dass Programmieren Spaß machen kann. Sieh es nicht als Arbeit an: Das Programmieren ist eine Möglichkeit, lustige Spiele oder Anwendungen zu erzeugen, die Du mit Deinen Freunden oder anderen teilen kannst.

Programmieren zu lernen ist ein tolles Training fürs Gehirn, und die Ergebnisse können sehr bereichernd sein. Aber vor allem gilt: Egal was Du tust, hab Spaß dabei!