

---

# Vorwort

Zunächst einmal bedanke ich mich bei Ihnen, dass Sie sich für dieses Buch entschieden haben. Hierin finden Sie einen fundierten und interaktiven Einstieg in die Programmierung mit Python. Dabei fangen wir mit den Grundlagen an und bauen Ihr Wissen immer weiter aus, sodass Sie nach der Lektüre in der Lage sind, eigene Experimente zu wagen, und bestenfalls Programmieren als neues Hobby lieben gelernt haben. Insbesondere die ungeheuren Möglichkeiten, kreativ zu werden und dabei immer wieder Neues zu entdecken, werden Sie bestimmt ähnlich faszinieren wie mich seit über 30 Jahren.

## Zielgruppe

Dieses Buch ist für Programmierneulinge gedacht. Es wendet sich somit an

- Schüler und Schülerinnen, die ein paar Tipps und Hilfestellungen suchen, die das Nachvollziehen des Informatikunterrichts erleichtern,
- Studierende, die ergänzende Erklärungen zu denen aus den Vorlesungen suchen, um Gelerntes schneller anwenden zu können oder besser für die nächste Prüfung vorbereitet zu sein,
- und alle, die einfach die wunderbare und vielfältige Welt der Programmierung mit Python kennenlernen möchten.

Zum Einstieg sind Programmiererfahrungen keine zwingende Voraussetzung – natürlich schaden diese nicht. Selbst dann nicht, wenn Sie sich vielleicht eher mit Java, C#, TypeScript oder JavaScript beschäftigt haben – aber für die Lektüre des Buchs ist es hilfreich, wenn Sie

- einigermaßen fit im Installieren von Programmen sind und
- wissen, was die Kommandozeile ist und sie grundlegend bedienen können.

## Was vermittelt dieses Buch?

Sie als Leser erhalten in diesem Buch einen Einstieg in Python. Allerdings ist die trockene Theorie auf ein Minimum reduziert und wir legen immer mit kleinen Beispielen los. Deshalb ist es auch ein Buch zum Mitmachen. Ich ermutige Sie ganz besonders, parallel zum Lesen auch immer ein paar Dinge auszuprobieren, vielleicht sogar mal das

eine oder andere abzuwandeln. Man lernt Programmieren einfach am besten, wenn man es praktiziert. Somit bietet es sich an, die abgebildeten Codeschnipsel abzutippen, also direkt im Python-Kommandozeileninterpreter einzugeben, oder später im Editor Ihrer Entwicklungsumgebung / IDE (Integrated Development Environment).

Damit Sie nicht über einfache Probleme stolpern, führt das Buch behutsam und schrittweise in die jeweilige Thematik ein und gibt Ihnen immer auch ein paar Hinweise, auf was man achten oder was man vielleicht sogar vermeiden sollte. Dazu dienen diverse Praxistipps mit Hintergrundinformationen.

### **Tipp: Praxistipp**

In derart formatierten Kästen finden sich im späteren Verlauf des Buchs immer wieder einige wissenswerte Tipps und ergänzende Hinweise zum eigentlichen Text.

## **Aufbau dieses Buchs**

Dieses Buch besteht aus jeweils in sich abgeschlossenen, aber aufeinander aufbauenden Kapiteln zu elementar wichtigen Bereichen der Programmiersprache Python. Abgerundet werden viele Kapitel mit Aufgaben und Musterlösungen, sodass das zuvor Gelernte direkt anhand neuer Problemstellungen praktiziert und das Wissen vertieft werden kann.

Für Ihren erfolgreichen Weg zur Python-Programmierung gliedert sich das Buch in die vier Teile Einstieg, Aufstieg, Praxisbeispiele und Schlussgedanken.

Im Teil »Einstieg« werden Grundlagen behandelt. Hier empfiehlt es sich wirklich, die Kapitel in der Reihenfolge des Buchs zu lesen, da mit jedem Kapitel neue Grundlagen und Themen hinzukommen, die im Anschluss vorausgesetzt und verwendet werden. Dann folgt der Teil »Aufstieg«. Dort beschäftigen wir uns mit leicht fortgeschrittenen Themen. Hier können Sie zwar nach Lust und Laune eins der Kapitel zur Lektüre auswählen, aber auch hier bauen einige Themen aufeinander auf. Der Teil »Praxisbeispiele« verdeutlicht die bisherigen Lerninhalte anhand von vereinfachten, didaktisch aufbereiteten Beispielen aus der Praxis. Hier haben Sie viel Spielraum zum Experimentieren und Einbringen eigener Ideen. Im Teil »Schlussgedanken« wird ein Ausblick gegeben, etwa auf Programmierstil und Testen. Das Buch endet dann mit einem Rückblick und Hinweisen für nächste Schritte.

## **Einstieg**

**Kapitel 1 – Einführung** Dieses Kapitel gibt zunächst einen kurzen Überblick über Pythons mittlerweile fast 30-jährige Geschichte. Bevor wir dann mit dem Lernen von Python als Programmiersprache loslegen können, müssen wir ein paar Installationen vornehmen. Zum einen benötigen wir Python an sich und zum anderen wird eine Entwicklungsumgebung / IDE (Integrated Development Environment) im Verlauf des Buchs mit zunehmender Komplexität der Beispiele immer nützlicher.

**Kapitel 2 – Schnelleinstieg** Dieses Kapitel bietet einen Schnelleinstieg und stellt viele wesentliche Elemente von Python vor. Dabei nehmen wir ganz behutsam Fahrt auf: Wir beginnen mit einer einfachen Ausgabe eines Textes, ganz traditionell »Hello World«, und lernen dann, wie wir das mithilfe von Variablen variieren. Zudem schauen wir uns die bedingte Ausführung mit Fallunterscheidungen und Wiederholungen mit Schleifen an.

**Kapitel 3 – Strings** Zeichenketten oder Strings sind aus kaum einem Programm wegzudenken. Variablen vom Typ `str` repräsentieren Zeichenketten und dienen zur Verwaltung von textuellen Informationen. In diesem Kapitel behandle ich die Thematik genauer.

**Kapitel 4 – Klassen und Objektorientierung** Immer wieder hört man, das Python auch die objektorientierte Programmierung unterstützt. Doch was bedeutet das? Zum Verständnis gibt dieses Kapitel einen Einblick in den objektorientierten Entwurf von Software. Dazu vermittele ich die grundlegenden Ideen von Zustand (Daten) in Kombination mit Verhalten (Funktionen auf diesen Daten) und wie man dies in Python formuliert.

**Kapitel 5 – Collections** Collections bzw. Containerklassen bieten Flexibilität und Komfort beim Verwalten von Daten. In Python unterstützen die vordefinierten Listen, Mengen und Schlüssel-Wert-Abbildungen bei der Verwaltung anderer Objekte.

**Kapitel 6 – Ergänzendes Wissen** In diesem Kapitel werden verschiedene wichtige Themen angesprochen, die in den vorherigen Kapiteln aus didaktischen Gründen bewusst ausgelassen wurden. Warum? Deren Beschreibung erfordert mehr Wissen rund um Python, was Sie mittlerweile erworben haben – vorher wäre das Ganze aber zu tief in die Details gegangen und hätte zu viel anderes Wissen vorausgesetzt. Hier angelangt lohnt es sich aber, das bisherige Wissen etwa mit Informationen zu Besonderheiten bei Parametern, dem Ternary-Operator, Aufzählungen mit `Enum` usw. zu komplettieren.

## Aufstieg

**Kapitel 7 – Collections Advanced** Dieses Kapitel beschreibt diverse Funktionalitäten rund um Collections. Dort gehen wir beispielsweise auf sequenzielle Datentypen, Iteratoren und Generatoren ein. Spezielle benannte Datencontainer lassen sich mit Named Tuples erzeugen. Schließlich schauen wir uns noch einführend Lambdas an.

**Kapitel 8 – Verarbeitung von Dateien** Dieses Kapitel beschäftigt sich mit der Verarbeitung von Informationen aus Dateien. Dies ist für viele Anwendungen von großer Bedeutung, da diverse Informationen nicht nur während der Programmlaufzeit von Interesse sind, sondern vor allem auch darüber hinaus – denken Sie etwa an die Highscore-Liste Ihres Lieblingsspiels.

**Kapitel 9 – Fehlerbehandlung mit Exceptions** Sicher kennen Sie es: Manchmal tritt ein Programmfehler auf und das Programm stürzt ab. Wichtige Daten gehen mitunter verloren. So etwas ist immer ärgerlich. Daher gehört auch die Behandlung von Fehlern zum guten Ton beim Programmieren. Diese Kapitel führt in die Thematik ein.

**Kapitel 10 – Datumsverarbeitung** Python bietet diverse praktische Funktionalitäten zur Datumsverarbeitung, die in diesem Kapitel einführend dargestellt werden.

## Praxisbeispiele

**Kapitel 11 – Praxisbeispiel: Tic Tac Toe** In diesem Kapitel entwickeln wir eine einfache Version des Strategiespiels Tic Tac Toe, das auf einem Spielfeld mit  $3 \times 3$  Feldern gespielt wird. Dabei wird verdeutlicht, warum wir Programme sinnvoll in kleine Methodenbausteine untergliedern sollten.

**Kapitel 12 – Praxisbeispiel: CSV-Highscore-Liste einlesen** In diesem Kapitel verdeutlicht ein weiteres Praxisbeispiel die Verarbeitung von Dateien und kommaseparierten Daten, auch CSV (Comma Separated Values) genannt. Um das Ganze unterhaltsam zu gestalten, werden wir statt trockener Anwendungsdaten eine Liste von Spielständen als Eingabe nutzen.

**Kapitel 13 – Praxisbeispiel: Worträtsel** Dieses dritte Praxisbeispiel umfasst eine etwas komplexere Programmieraufgabe, nämlich die Erstellung von Worträtseln, die man aus Zeitschriften kennt. Dabei sollen aus einem scheinbaren »Buchstabensalat« verschiedene dort versteckte Begriffe extrahiert werden. Dieses Kapitel vermittelt, wie man Aufgaben in verschiedene kleine Problemstellungen untergliedert und diese jeweils mit eigenen Klassen realisieren kann. Schließlich ist es dann Aufgabe der eigentlichen Applikation, wie ein Dirigent zu wirken und die Einheiten passend zusammenzufügen. Dabei lernen wir beispielsweise den Import möglicher Wörter aus Dateien, die Modellierung des Rätsels und einen einfachen Export nach HTML kennen.

## Schlussgedanken

**Kapitel 14 – Gute Angewohnheiten** Dieses Kapitel stellt Ihnen ein paar Dinge zu gutem Programmierstil vor. Das umfasst sogenannte Coding Conventions, also Regeln beim Programmieren. Außerdem zeige ich noch, wie sich einige davon mit Tools prüfen lassen und wie man Programme mit Pytest testen und dadurch Fehler vermeiden kann.

**Kapitel 15 – Schlusswort** Hier rekapitulieren wir kurz, was Sie durch die Lektüre gelernt haben sollten und wie Sie möglicherweise weitermachen können.

## Anhang

**Anhang A – Schlüsselwörter im Überblick** In Python existiert eine Reihe von Schlüsselwörtern, die reserviert sind und nicht als Bezeichner für Variablen, Funktionen, Methoden, Klassen oder anderes verwendet werden dürfen. Hier erhalten Sie einen Überblick.

**Anhang B – Schnelleinstieg REPL** In diesem Buch werden diverse Beispiele direkt auf der Konsole ausprobiert. Der Grund ist vor allem, dass Python einen interaktiven Kommandozeileninterpreter als REPL (Read-Eval-Print-Loop) bietet.

**Anhang C – Wesentliche Neuerungen aus Python 3.10 im Kurzüberblick** Dieser Anhang stellt einige der für das kommende Python 3.10 umgesetzten Erweiterungen vor, die für Sie möglicherweise von Relevanz sind und die weiteren Gehversuche erleichtern können.

## Sourcecode und ausführbare Programme

Ich hatte schon angedeutet, dass es zum Erlernen des Programmierens hilfreich ist, die Beispiele und Codeschnipsel auch auszuprobieren und abzutippen. Um Ihnen ein wenig Tipparbeit und Mühe zu sparen, finden Sie viele der Beispiele als Programme in einem PyCharm-Projekt. Dieses steht unter [www.dpunkt.de/Einfach-Python](http://www.dpunkt.de/Einfach-Python) zur Verfügung. Weitere Informationen zum genauen Vorgehen finden Sie auf der Download-Seite.

### Verwendete Python-Version

Die Beispiele dieses Buchs wurden mit der im Sommer 2021 aktuellen Version 3.9.6 von Python entwickelt. In einem Anhang gehe ich auf einige relevante Features aus dem im Oktober 2021 erscheinenden Python 3.10 ein, damit Sie auch für die Zukunft informiert sind. Wenn Ihre Python-Version noch nicht ganz so aktuell ist, wird dies eher selten ein Problem darstellen. Die allermeisten Programme sollten nämlich ohne Änderungen auch in etwas älteren Python-Versionen wie 3.8 oder 3.7 funktionieren.

## Konventionen

### Verwendete Zeichensätze

In diesem Buch gelten folgende Konventionen bezüglich der Schriftart: Neben der vorliegenden Schriftart werden wichtige Textpassagen *kursiv* oder ***kursiv und fett*** markiert. Englische Fachbegriffe werden eingedeutscht großgeschrieben, etwa Event Handling. Zusammensetzungen aus englischen und deutschen (oder eingedeutschten) Begriffen werden mit Bindestrich verbunden, z. B. Plugin-Manager. Listings mit Sourcecode sind

in der Schrift `Courier` gesetzt, um zu verdeutlichen, dass diese einen Ausschnitt aus einem Python-Programm darstellen. Auch im normalen Text wird für Klassen, Funktionen, Methoden, Variablen und Parameter diese Schriftart genutzt.

## Verwendete Abkürzungen

Im Buch verwende ich die in der nachfolgenden Tabelle aufgelisteten Abkürzungen. Weitere Abkürzungen werden im laufenden Text in Klammern nach ihrer ersten Definition aufgeführt und anschließend bei Bedarf genutzt.

Abkürzung	Bedeutung
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
IDE	Integrated Development Environment

## Danksagung

Wie schon bei einigen meiner Bücher hat mich auch diesmal Michael Kulla wieder ganz besonders unterstützt, wie üblich breitgefächert vom Aufdecken von Tippfehlern bis hin zu diversen inhaltlichen Hinweisen. Das gilt insbesondere für die auf Java ausgerichtete Version dieses Buchs.

Selbstverständlich geht ein Dankeschön an das Team des dpunkt.verlags (Dr. Michael Barabas, Anja Weimer und Stefanie Weidner) für die tolle Zusammenarbeit. Außerdem möchte ich mich bei Tobias Overkamp für die fundierte fachliche Durchsicht sowie bei Ursula Zimpfer für ihre Adleraugen beim Copy-Editing bedanken.

Abschließend geht ein lieber Dank an meine Frau Lilija für ihr Verständnis und die Unterstützung. Ihren ganz besonderen Anteil hat unser kleiner Sonnenschein Sophie Jelena dazu beigetragen, indem sie den Papa immer wieder zum Lachen gebracht hat.

## Anregungen und Kritik

Trotz großer Sorgfalt und mehrfachen Korrekturlesens lassen sich missverständliche Formulierungen oder sogar Fehler leider nicht vollständig ausschließen. Falls Ihnen etwas Derartiges auffallen sollte, so zögern Sie bitte nicht, mir dies mitzuteilen. Gerne nehme ich auch Anregungen oder Verbesserungsvorschläge entgegen. Kontaktieren Sie mich bitte per Mail unter:

michael\_inden@hotmail.com

Zürich, im September 2021  
Michael Inden

Diese Leseprobe haben Sie beim  
 [edv-buchversand.de](http://edv-buchversand.de) heruntergeladen.  
Das Buch können Sie online in unserem  
Shop bestellen.  
[Hier zum Shop](#)