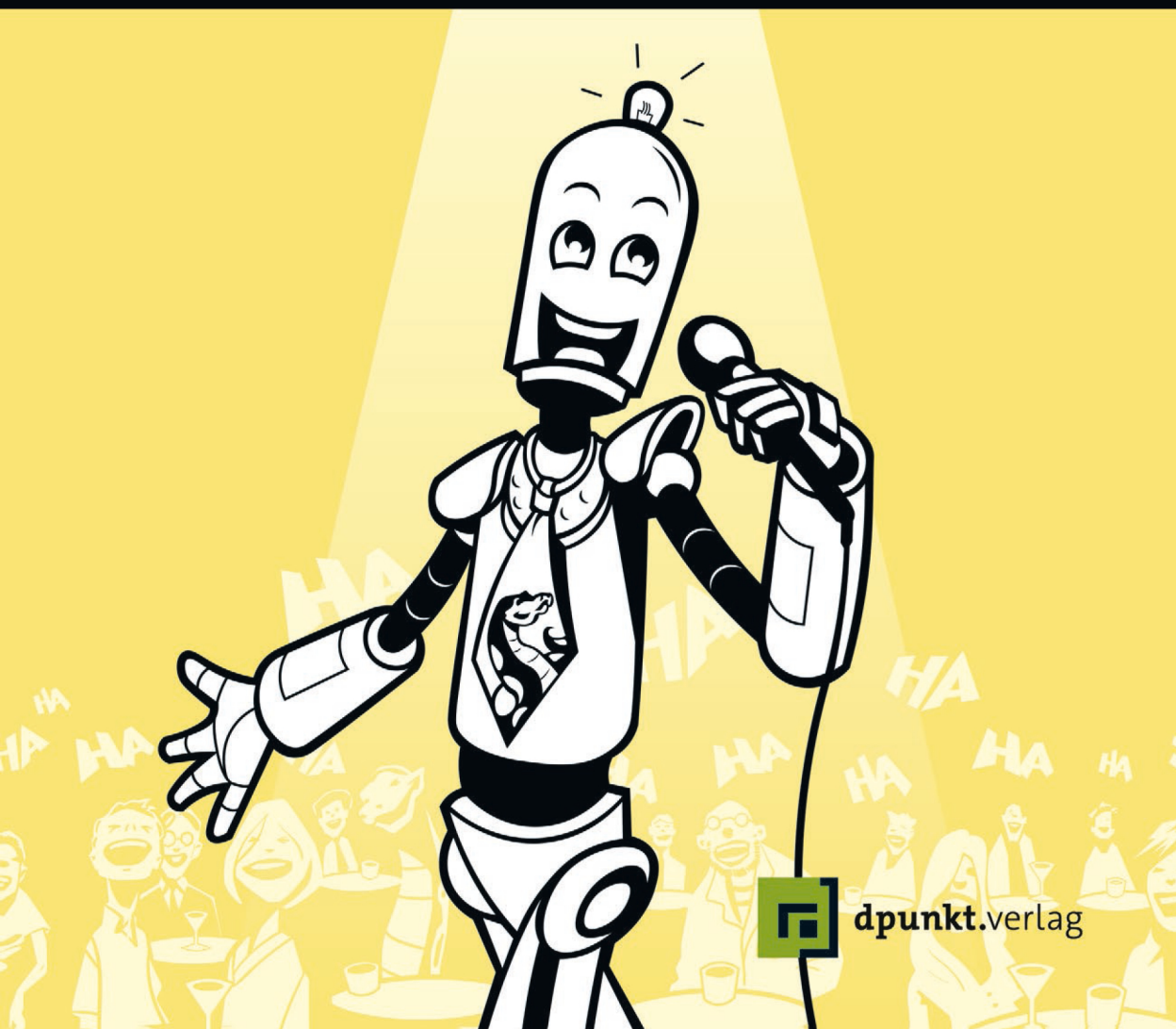
Python One-Liners

Profi-Programmierung durch kurz gefasstes Python

Christian Mayer



dpunkt.verlag

Christian Mayer hat einen Doktortitel in Informatik und ist der Gründer der beliebten Python-Site Finxter (*https://blog.finxter.com*). Mayer ist außerdem der Autor der Coffee Break Python-Reihe.

Christian Mayer

Python One-Liners

Profi-Programmierung durch kurz gefasstes Python

dpunkt.verlag

Christian Mayer

Lektorat: Gabriel Neumann
Lektoratsassistenz: Anja Weimer
Übersetzung: Kathrin Lichtenberg
Copy-Editing: Claudia Lötschert, *www.richtiger-text.de*
Satz: Veronika Schnabel
Herstellung: Stefanie Weidner
Umschlaggestaltung: Helmut Kraus, *www.exclam.de*
Druck und Bindung: mediaprint solutions GmbH, 33100 Paderborn

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über *http://dnb.d-nb.de* abrufbar.

ISBN:
Print 978-3-86490-805-7
PDF 978-3-96910-114-8
ePub 978-3-96910-115-5
mobi 978-3-96910-116-2

1. Auflage 2021

Wieblinger Weg 17
69123 Heidelberg

Hinweis:
Dieses Buch wurde auf PEFC-zertifiziertem Papier aus nachhaltiger Waldwirtschaft gedruckt. Der Umwelt zuliebe verzichten wir zusätzlich auf die Einschweißfolie.

Schreiben Sie uns:
Falls Sie Anregungen, Wünsche und Kommentare haben, lassen Sie es uns wissen: *hallo@dpunkt.de.*

5 4 3 2 1 0

Inhalt

Danksagung

Python One-Liners ist das Resultat von mehr als eintausend Zeitstunden Arbeit – einen großen Teil davon leisteten die Mitarbeiter des in San Francisco ansässigen Verlags No Starch Press. Mein besonderer Dank gilt dem Verlagsleiter Bill Pollock für seine Einladung, dieses Buch zu schreiben. Die brillanten Lektorinnen Liz Chadwick, Alex Freed und Janelle Ludowise brachten meine groben Entwürfe in eine besser lesbare Form. Dank Liz, Alex und Janelle erreichte *Python One-Liners* ein Maß an Klarheit, das ich allein nicht hätte erreichen können.

Professor Daniel Zingaro – Autor des populären Buchs *Algorithmic Thinking* – nutzte seine tiefgehenden Informatikkenntnisse, um meine Ungenauigkeiten in Text und Code zu beseitigen. Ohne seine Bemühungen hielten Sie jetzt ein Buch in ihren Händen, das nicht nur mehr Fehler enthielte, sondern auch schwerer zu lesen wäre. Selbstverständlich verbleiben alle übrigen Ungenauigkeiten meine eigenen.

Zu diesem Buch trugen direkt oder indirekt zahlreiche Menschen bei. Mein Doktorvater Professor Rothermel lehrte mich, dass es in der Informatikbildung weitaus effizienter ist, grundlegende Konzepte statt oberflächlicher Fakten zu vermitteln. Auch in der schnelllebigen Welt der Informatik bleiben Konzepte über Jahrzehnte bestehen, während sich Fakten und Technologien schnell verändern.

Meine schöne Frau Anna Altimira hat stets ein offenes Ohr, selbst für meine wildesten Ideen, und ich bin ihr zutiefst dankbar dafür! Für Inspiration und Motivation sorgten meine wunderbaren Kinder Amalie und Gabriel. In der Hoffnung, dass sie eines Tages dieses Vorwort lesen werden: Ich liebe euch sehr!

Meine größte Motivationsquelle waren aber stets die aktiven Mitglieder der Finxter-Community. Dieses Buch richtet sich an ambitionierte Programmierer – wie Sie –, die ihre Programmierfähigkeiten verbessern möchten, um sich den Herausforderungen des Informationszeitalters zu stellen. Nach einem langen Arbeitstag waren es häufig die aufbauenden Worte der Finxter-Mitglieder, die mich ermutigten, einen weiteren Abschnitt des Buchs zu schreiben.

Zur deutschen Ausgabe

Ein besonderer Dank geht an den dpunkt.verlag für das Erstellen dieser deutschsprachigen Ausgabe von *Python One-Liners*. Insbesondere möchte ich meine Dankbarkeit gegenüber Gabriel Neumann und Kathrin Lichtenberg aussprechen für ihre gewissenhafte und unermüdliche Arbeit an der deutschsprachigen Ausgabe. Eine großartige Übersetzung anzufertigen, ist eine Meisterleistung, und ich bin dankbar, dass die fähigen Lektoren und Übersetzer des dpunkt.verlags sich dieser Aufgabe annahmen.

Das Buch verwendet an verschiedenen Stellen die maskuline, an anderen die feminine Form. Diese Lösung habe ich der leichteren Lesbarkeit halber gewählt. Sie impliziert keine Benachteiligung anderer Geschlechter, sondern ist im Sinne der sprachlichen Vereinfachung als geschlechtsneutral zu verstehen.

Vorwort

In Deutschland habe ich Informatik studiert und promoviert. Es ist mir daher ein besonderes Vergnügen, das Vorwort für diese deutsche Ausgabe zu schreiben.

Der Trend zu Digitalisierung und Programmierung ist ungebrochen in Deutschland. Auf jeden Informatikabsolventen kommen zahlreiche offene Stellen im produzierenden Gewerbe, in der Automobilindustrie, im E-Commerce sowie im öffentlichen Dienst. Programmierfähigkeiten sind gefragt – und mit dem Aufkommen von selbstfahrenden Autos, leistungsfähigen Industrierobotern und der raschen Ausbreitung neuer Technologien im maschinellen Lernen gibt es kaum noch einen Wirtschaftssektor, der von der Digitalisierung unberührt bleibt. Auch neuere Entwicklungen wie die zunehmende Bedeutung von Arbeit im Homeoffice verstärken die Nachfrage nach Informatikern und Programmierern, die mithilfe ihrer Programmierfähigkeiten die Grundlage dafür schaffen. Auch in den kommenden Jahrzehnten können Sie sich darauf verlassen, dass die Nachfrage nach Python-Entwicklern weiter steigen wird, insbesondere im deutschsprachigen Raum.

Meine Python Community[1] zur Informatikbildung, Finxter, erfreute sich in den letzten Jahren großen Zulaufs. Je mehr Menschen ich durch Python begegnete,

1 Nachfolgend der Link zur Teilnahme an der kostenlosen Finxter Python Email Academy mit herunterladbaren Python PDFs und regelmäßigen Video-Lektionen des Finxter-Teams: *https://blog.finxter.com/email-academy/*

desto mehr kristallisierte sich für mich heraus, dass Python nicht nur eine reine Programmiersprache zur Bedienung von Maschinen ist – Deutsch ist ja auch nicht nur eine Sprache zur Bedienung von Menschen! Wie eine sogenannte natürliche Sprache verbindet Python Millionen von Menschen unterschiedlicher Herkunft. Wie eine natürliche Sprache erleben Millionen Menschen, wie Python ihrer schöpferischen Kraft Ausdruck verleiht.

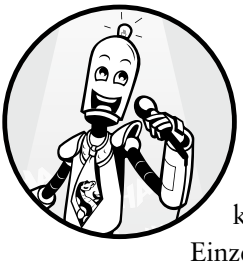
Ich bin überzeugt, dass jeder moderne Mensch davon profitiert, Python zu lernen. Python ist die Sprache der Algorithmen. Und diese durchdringen mehr und mehr Bereiche des täglichen Lebens. Algorithmen finden kürzeste Wege zur Arbeit, lenken selbstfahrende Autos und helfen bei der effizienten Verteilung von Gütern in weltweiten Mobilitätsnetzen. Python ermöglicht der Menschheit eine effizientere und skalierbarere Organisation von Arbeit. Mithilfe von Computern und Automatisierung sind wir nun in der Lage, eine Gesellschaft zu entwerfen, in der jeder Menschen die Freiheit besitzt, sich seinen Alltag weitestgehend nach seinen eigenen Vorstellungen zu gestalten.

Das Verständnis von Python wird auch Ihrer Karriere nützen. Einige meiner Freunde, Kommilitonen und Kollegen in der Informatikforschung schulden ihren internationalen Erfolg zu einem nicht unwesentlichen Teil ihrer Fähigkeit, Python zu sprechen. Viele der größten Unternehmen und Organisationen unserer Zeit sind mehr denn je an »deutscher Ingenieurskunst« und aufstrebenden Programmierern interessiert, die der Sprache Python mächtig sind.

Bevor Sie mit dem Sprechen ganzer Sätze beginnen, müssen Sie zunächst einzelne Wörter verstehen. Und bevor Sie mit dem Erzählen ganzer Geschichten beginnen, müssen Sie zunächst ganze Sätze bilden können. Das trifft auch auf Programmiersprachen zu! Jedes noch so komplizierte Python-Projekt besteht aus einer Reihe von – oftmals Tausenden – Python One-Liners. Dieses Buch wird Ihnen ein fundiertes Verständnis der einzelnen Zeile von Python-Code vermitteln, sodass Sie nach dessen Lektüre die Python-Vokabeln beherrschen.

Mit diesem Buch möchte ich technikbegeisterten Menschen helfen, die Sprache der Algorithmen zu sprechen, Computer zu beherrschen und deren Kraft gegen die großen Probleme unserer Zeit zu richten.

Einführung



Mit diesem Buch möchte ich Ihnen helfen, zum Python-Experten zu werden. Wir konzentrieren uns dabei auf *Python-Einzeiler*: knappe Programme, gepackt in eine einzige Zeile Python. Durch die Einzeiler lernen Sie, Code schneller und präziser zu lesen und zu schreiben und Ihr Verständnis der Sprache zu verbessern.

Es gibt außerdem noch fünf weitere Gründe, weshalb ich glaube, dass Python-Einzeiler Ihnen helfen, sich zu verbessern, und deshalb lohnenswerte Studienobjekte sind.

Erstens, durch das Verbessern Ihrer Python-Kernfähigkeiten werden Sie in die Lage versetzt, viele der kleinen Programmierschwächen zu überwinden, die Sie bremsen. Es ist nicht leicht, Fortschritte zu erzielen, wenn man die Grundlagen nicht richtig verstanden hat. Einzelne Codezeilen sind die Grundbausteine jedes Programms. Wenn Sie diese Grundbausteine verstehen, können Sie auch komplexere Probleme meistern, ohne dass Sie diese überfordern.

Zweitens lernen Sie, die unglaublich beliebten Python-Bibliotheken auszunutzen, wie diejenigen für Data Science und Machine Learning. Das Buch besteht aus

fünf Einzeiler-Kapiteln, die jeweils einem eigenen Bereich von Python gewidmet sind, von regulären Ausdrücken bis zum Machine Learning. Durch diese Vorgehensweise erhalten Sie einen Überblick über mögliche Python-Anwendungen und lernen darüber hinaus, diese machtvollen Bibliotheken einzusetzen.

Drittens lernen Sie, Code auf Python-spezifischere Weise zu schreiben. Python-Anfänger, speziell solche, die vorher mit anderen Programmiersprachen gearbeitet haben, schreiben Code oft auf sehr Python-untypische Weise. Wir behandeln Python-spezifische Konzepte wie List Comprehensions, Mehrfachzuweisungen und Slicing, mit deren Hilfe Sie Code schreiben können, der gut lesbar ist und sich dafür eignet, ihn mit anderen Programmierern zu teilen.

Viertens zwingt die Beschäftigung mit Python-Einzeilern Sie dazu, klar und präzise zu denken. Wenn jedes einzelne Codesymbol zählt, bleibt kein Platz für zerstreutes und unkonzentriertes Programmieren.

Fünftes erlauben es Ihnen Ihre neuen Fähigkeiten, übermäßig komplizierte Python-Codeprojekte zu durchschauen und Freunde wie Arbeitgeber gleichermaßen zu beeindrucken. Außerdem dürfte es Ihnen Spaß machen, anspruchsvolle Programmierprobleme mit nur einer einzigen Zeile Code zu lösen. Und Sie wären damit nicht allein: Eine aktive Online-Community aus Python-Geeks wetteifert um die kompaktesten, Python-typischsten Lösungen für verschiedene praktische (und nicht so praktische) Probleme.

Ein Beispiel für einen Python-Einzeiler

Die zentrale These dieses Buchs ist, dass das Lernen von Python-Einzeilern entscheidend für das Verständnis anspruchsvollerer Codeprojekte sowie ein ausgezeichnetes Werkzeug für die Verbesserung Ihrer Fähigkeiten ist. Bevor Sie verstehen, was in einer Codebasis mit Tausenden von Zeilen passiert, müssen Sie die Bedeutung einer einzelnen Codezeile verstehen.

Schauen wir uns einen Python-Einzeiler an. Es macht nichts, wenn Sie nicht alles verstehen. Sie werden diesen Einzeiler in Kapitel 6 meistern.

```
q = lambda l: q( ❶[x for x in l[1:] if x <= l[0]]) + [l[0]] + q([x for x in
l if x > l[0]]) if l else []
```

Dieser Einzeiler ist eine wunderbare und präzise Möglichkeit, den berühmten Quicksort-Algorithmus kurz und knapp zusammenzufassen, auch wenn seine Bedeutung für viele Anfänger und selbst für fortgeschrittene Anfänger schwierig zu durchschauen sein dürfte.

Python-Einzeiler bauen oft aufeinander auf, sodass die Einzeiler im Laufe des Buchs immer komplexer werden. Wir beginnen hier mit einfachen Einzeilern, die dann später die Grundlage für komplexere Einzeiler bilden werden. So ist zum

Beispiel der gezeigte Quicksort-Einzeiler schwierig und lang, basiert aber auf dem einfacheren Konzept der List Comprehension ❶. Schauen Sie sich die folgende einfachere List Comprehension an, die eine Liste aus Quadratzahlen erzeugt:

```
lst  = [x**2 for x in range(10)]
```

Wir können diesen Einzeiler in noch einfachere Einzeiler zerlegen, die uns wichtige Python-Grundlagen lehren, wie Variablenzuweisungen, mathematische Operatoren, Datenstrukturen, `for`-Schleifen, Zugehörigkeitsoperatoren und die `range()`-Funktion – und all dies in einer einzigen Zeile Python!

Seien Sie sich bewusst, dass *grundlegend* nicht gleichbedeutend ist mit *trivial*. Alle Einzeiler, die wir uns anschauen, sind sinnvoll und nützlich, und jedes Kapitel befasst sich mit einem eigenen Bereich oder einer Disziplin der Informatik, sodass Sie am Ende einen umfassenden Eindruck von der Stärke gewonnen haben werden, die Python Ihnen bietet.

Ein Hinweis zur Lesbarkeit

The Zen of Python fasst 19 Leitsätze für die Programmiersprache Python zusammen. Sie können sie in Ihrer Python-Shell lesen, indem Sie `import this` eingeben:

```
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
-- schnipp --
```

Laut *The Zen of Python* ist Lesbarkeit wichtig – »Readability counts«. Einzeiler sind minimalistische Programme zum Lösen von Problemen. In vielen Fällen wird durch das Umformulieren eines Stücks Code in einen Python-Einzeiler die Lesbarkeit verbessert, und der Code wird Python-artiger. Ein Beispiel ist die Verwendung von *List Comprehension*, um das Erzeugen von Listen auf eine einzelne Codezeile zu reduzieren. Schauen Sie sich das folgende Beispiel an:

```
# VORHER
squares = []

for i in range(10):
    squares.append(i**2)
```

```
print(squares)
# [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

In diesem Codeausschnitt brauchen wir fünf Zeilen Code, um eine Liste der ersten zehn Quadratzahlen zu erzeugen und auf der Shell auszugeben. Da ist es doch viel besser, wenn man eine Einzeilerlösung benutzt, die dasselbe in einer besser lesbaren und präzisen Weise erreicht:

```
# NACHHER
print([i**2 for i in range(10)])
# [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

Das Ergebnis ist identisch, doch der Einzeiler setzt auf das Python-spezifische Konzept der List Comprehension. Er ist leichter lesbar und prägnanter.

Es kann aber auch schwierig sein, Python-Einzeiler zu verstehen. In manchen Fällen ist eine Python-Einzeilerlösung nicht lesbarer. Doch genau wie ein Schachmeister, der alle möglichen Züge kennen muss, bevor er sich für einen entscheiden kann, der ihm als der beste erscheint, müssen Sie alle Möglichkeiten kennen, Ihre Gedanken in Code auszudrücken, um zu entscheiden, welche die beste ist. Nach der *schönsten* Lösung zu streben, ist gar nicht so unwichtig; schließlich findet sich dieses Prinzip im Herzen des Python-Ökosystems. Wie uns *The Zen of Python* lehrt, ist schön besser als hässlich: »Beautiful is better than ugly«.

An wen richtet sich dieses Buch?

Sind Sie Anfänger oder fortgeschrittener Anfänger? Wie viele Ihrer Kollegen stecken auch Sie möglicherweise im Programmierprozess. Dieses Buch kann Ihnen aus dieser Lage heraushelfen. Sie haben online viele Programmieranleitungen gelesen. Sie haben Ihren eigenen Quellcode geschrieben und erfolgreich kleine Projekte ausgeliefert. Sie haben einen Programmiergrundkurs abgeschlossen und ein oder zwei Fachbücher zur Programmierung gelesen. Vielleicht haben Sie sogar einen Kurs an der Uni absolviert, bei dem Sie die Grundlagen von Informatik und Programmierung kennengelernt haben.

Vielleicht werden Sie durch bestimmte Überzeugungen beschränkt, etwa, dass die meisten Programmierer Quellcode viel schneller verstehen als Sie oder dass Sie auf keinen Fall zu den zehn Prozent besten Programmierern gehören. Falls Sie lernen wollen, besser zu programmieren und zur Spitze vorzudringen, müssen Sie neue, nützliche Fähigkeiten erwerben.

Ich kann Ihre Probleme nachvollziehen. Als ich vor zehn Jahren begann, Informatik zu studieren, war ich überzeugt, dass ich nichts über das Programmieren wusste. Gleichzeitig schien mir, als seien alle meine Kommilitonen bereits sehr erfahren und kompetent.

Ich möchte Ihnen mit diesem Buch helfen, diese einschränkenden Überzeugungen zu überwinden und einen entscheidenden Schritt zum Meistern von Python zu machen.

Was werden Sie lernen?

Es folgt ein Überblick über das, was Sie lernen werden.

- **Kapitel 1: Python-Auffrischungskurs** Führt Sie in die Grundlagen von Python ein, um Ihr Wissen aufzufrischen.
- **Kapitel 2: Python-Tricks** Enthält zehn Einzeilertricks, mit denen Sie die Grundlagen meistern können, wie etwa List Comprehensions, Dateieingabe, die Funktionen `lambda`, `map()` und `zip()`, den Quantor `all()`, Slicing und einfache Listenberechnungen. Sie lernen außerdem, wie Sie Datenstrukturen benutzen und manipulieren, um verschiedene Standardprobleme zu lösen.
- **Kapitel 3: Data Science** Enthält zehn Einzeiler für das Data Science, die auf der NumPy-Bibliothek aufbauen. NumPy bildet den Kern der leistungsstarken Machine-Learning- und Data-Science-Fähigkeiten von Python. Sie eignen sich elementare NumPy-Grundlagen wie Array, Shape, Typ, Broadcasting, fortgeschrittene Indexierung, Slicing, Sortieren, Suchen, Sammeln und Statistiken an.
- **Kapitel 4: Machine Learning** Behandelt zehn Einzeiler für das Machine Learning mit Pythons scikit-learn-Bibliothek. Sie lernen Regressionsalgorithmen kennen, die Werte vorhersagen. Dazu gehören lineare Regression, k-Nearest Neighbors und neuronale Netzwerke. Außerdem lernen Sie Klassifikationsalgorithmen kennen wie logistische Regression, Decision-Tree Learning, Support-Vector Machines und Random Forests. Darüber hinaus erfahren Sie, wie Sie einfache Statistiken mehrdimensionaler Datenfelder sowie den k-Means-Algorithmus für Unsupervised Learning berechnen. Diese Algorithmen und Methoden gehören zu den wichtigsten auf dem Gebiet des Machine Learning.
- **Kapitel 5: Reguläre Ausdrücke** Enthält zehn Einzeiler, mit denen Sie mehr aus regulären Ausdrücken herausholen. Sie lernen verschiedene einfache reguläre Ausdrücke kennen, die Sie kombinieren (und neu kombinieren) können, um komplexere reguläre Ausdrücke zu erzeugen, und verwenden Gruppierung und benannte Gruppen, negative Lookaheads, Escape-Zeichen, Whitespaces, Zeichenmengen (und negative Zeichenmengen) sowie gierige/nicht gierige Operatoren.

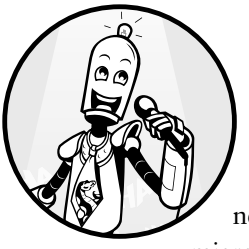
- **Kapitel 6: Algorithmen** Enthält zehn Einzeiler-Algorithmen zu einem breiten Spektrum an Informatikthemen, darunter Anagramme, Palindrome, Potenzmengen, Permutationen, Fakultäten, Primzahlen, Fibonacci-Zahlen, Verschleiern, Suche und algorithmisches Sortieren. Vieles davon dient als Grundlage für komplexere Algorithmen und enthält den Keim für eine gründliche algorithmische Ausbildung.
- **Nachwort** Beschließt das Buch und entlässt Sie in die wirkliche Welt, in der Sie Ihre neuen und verbesserten Python-Programmierkenntnisse anwenden können.

Online-Ressourcen

Zur Ergänzung des Übungsmaterials in diesem Buch habe ich weitere Ressourcen bereitgestellt, die Sie online unter *https://pythononeliners.com/* oder *http://www.nostarch.com/pythononeliners/* finden. Zu den interaktiven Ressourcen gehören:

- **Python-Cheat-Sheets** Sie können diese Python-Schummelseiten als PDFs herunterladen, ausdrucken und sich an die Wand heften. Diese Seiten enthalten wichtige Python-Sprachfunktionen und helfen Ihnen dabei, Ihre Python-Kenntnisse aufzufrischen und Wissenslücken zu schließen.
- **Einzeiler-Video-Lektionen** Als Teil meines Python-E-Mail-Kurses habe ich viele Python-Einzeiler-Lektionen aus diesem Buch aufgezeichnet, die Sie kostenlos abrufen können. Diese Lektionen sollen Ihnen beim Lernen helfen und bieten ein multimediales Lernerlebnis.
- **Python-Rätsel** Sie können die Online-Ressourcen besuchen, um Python-Rätsel zu lösen, und mit der kostenlosen *Finxter.com*-App Ihre Python-Fertigkeiten testen und trainieren sowie Ihren Lernfortschritt beim Durcharbeiten des Buchs messen.
- **Code-Dateien und Jupyter-Notebooks** Sie müssen die Ärmel hochkrempeln und tatsächlich mit Code arbeiten, um zur Python-Expertin zu werden. Spielen Sie mit den verschiedenen Parameterwerten und Eingabedaten herum. Zu Ihrer Bequemlichkeit habe ich alle Python-Einzeiler als ausführbare Codedateien hinzugefügt.

1
Python-Auffrischungskurs



Zweck dieses Kapitels ist es, Ihre Kenntnisse der grundlegenden Python-Datenstrukturen, -Schlüsselwörter, Kontrollflussoperationen usw. aufzufrischen. Ich habe dieses Buch für Python-Programmierer geschrieben, die bereits gewisse Fähigkeiten besitzen. Um zum Experten zu werden, müssen Sie natürlich die Grundlagen beherrschen.

Das Verständnis der Grundlagen erlaubt es Ihnen, einen Schritt zurückzutreten und das breitere Bild zu betrachten – eine wichtige Fähigkeit, ob Sie nun den Aufstieg bei Google schaffen, Informatikprofessorin werden oder einfach nur gut programmieren können wollen. Informatikprofessoren besitzen z. B. oft ein unglaublich tiefgreifendes Wissen der Grundlagen in ihrem Gebiet, das es ihnen ermöglicht, über Grundprinzipien zu diskutieren und Forschungslücken zu erkennen, statt sich von der allerneuesten Technologie blenden zu lassen. Dieses Kapitel präsentiert die wichtigsten Python-Grundlagen, die als Fundament für die höheren Themen in diesem Buch dienen.