

2.

Auflage

dpunkt.verlag



# Python 3 Eric Matthes Crashkurs

Eine praktische,  
projektbasierte Programmier Einführung

Papier  
plus<sup>+</sup>  
PDF.

Zu diesem Buch – sowie zu vielen weiteren dpunkt.büchern – können Sie auch das entsprechende E-Book im PDF-Format herunterladen. Werden Sie dazu einfach Mitglied bei dpunkt.plus<sup>+</sup>:

[www.dpunkt.plus](http://www.dpunkt.plus)

**Eric Matthes**

# **Python 3 Crashkurs**

**Eine praktische, projektbasierte  
Programmier Einführung**

**2., überarbeitete und aktualisierte Auflage**



**dpunkt.verlag**

Eric Matthes

Lektorat: Dr. Michael Barabas  
Fachgutachter: Kenneth Love  
Copy-Editing: Ursula Zimpfer, Herrenberg  
Übersetzung & Satz: G&U Language & Publishing Services GmbH, Flensburg,  
[www.GundU.com](http://www.GundU.com)  
Herstellung: Stefanie Weidner  
Umschlaggestaltung: Helmut Kraus, [www.exclam.de](http://www.exclam.de)

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 978-3-86490-735-7  
PDF 978-3-96910-031-8  
ePub 978-3-96910-032-5  
mobi 978-3-96910-033-2

2., überarbeitete und aktualisierte Auflage 2020  
Translation Copyright für die deutschsprachige Ausgabe © 2020 dpunkt.verlag  
GmbH  
Wieblinger Weg 17  
69123 Heidelberg

Copyright © 2019 by Eric Matthes. Title of English-language original: Python  
Crash Course, 2nd Edition: A Hands-On, Project-Based Introduction to  
Programming, ISBN 978-1-59327-928-8, published by No Starch Press. German-  
language edition copyright © 2020 by dpunkt.verlag. All rights reserved.

*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](mailto:hallo@dpunkt.de).

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autor noch Verlag noch Übersetzer können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen.

5 4 3 2 1 0

# Inhalt

**Der Autor**

**Der Fachgutachter**

**Danksagung**

**Vorwort zur zweiten Auflage**

**Einleitung**

## **Teil 1 Grundlagen**

---

### **1 Erste Schritte**

Die Programmierumgebung einrichten

- Python-Versionen

- Python-Codeausschnitte ausführen

- Der Editor Sublime Text

Python auf verschiedenen Betriebssystemen

- Python unter Windows

- Python unter macOS

- Python unter Linux

Das Hello-World-Programm ausführen

- Sublime Text auf die richtige Python-Version einstellen

Hello\_world.py ausführen

Fehlersuche

Python-Programme im Terminal ausführen

Unter Windows

Unter Linux und macOS

Zusammenfassung

## **2 Variablen und einfache Datentypen**

Was bei der Ausführung von hello\_world.py wirklich geschieht

Variablen

Variablen benennen und verwenden

Fehler bei Variablennamen vermeiden

Variablen sind Etiketten

Strings

Groß- und Kleinschreibung mithilfe von Methoden ändern

Variablen in Strings verwenden

Weißraum hinzufügen

Weißraum entfernen

Syntaxfehler bei der Stringverarbeitung vermeiden

Zahlen

Integer

Fließkommazahlen

Integer und Fließkommazahlen

Unterstriche in Zahlen

Mehrfachzuweisung

Konstanten

Kommentare

Wie werden Kommentare geschrieben?

Was für Kommentare sind sinnvoll?

The Zen of Python

Zusammenfassung

### **3 Eine Einführung in Listen**

Was sind Listen?

- Elemente in einer Liste ansprechen
- Indizes beginnen bei 0, nicht bei 1
- Einzelne Werte aus einer Liste verwenden

Elemente ändern, hinzufügen und entfernen

- Elemente in einer Liste ändern
- Elemente zu einer Liste hinzufügen
- Elemente aus einer Liste entfernen

Listen ordnen

- Listen mit `sort()` dauerhaft sortieren
- Listen mit der Funktion `sorted()` vorübergehend sortieren
- Listen in umgekehrter Reihenfolge ausgeben
- Die Länge einer Liste ermitteln

Indexfehler vermeiden

Zusammenfassung

### **4 Mit Listen arbeiten**

Eine komplette Liste durchlaufen

- Die Schleife im Detail
- Weitere Aufgaben in einer `for`-Schleife erledigen
- Aktionen nach der `for`-Schleife

Einrückungsfehler vermeiden

- Vergessene Einrückung der ersten Zeile in einer Schleife
- Vergessene Einrückung nachfolgender Zeilen
- Unnötige Einrückung
- Unnötige Einrückung nach einer Schleife
- Vergessener Doppelpunkt

## Numerische Listen

- Die Funktion range()

- Numerische Listen mithilfe von range() aufstellen

- Einfache Statistiken für numerische Listen

- Listennotation

## Teillisten

- Einen Slice erstellen

- Einen Slice in einer Schleife durchlaufen

- Listen kopieren

## Tupel

- Ein Tupel definieren

- Die Werte in einem Tupel durchlaufen

- Tupel überschreiben

## Code formatieren

- Die Gestaltungsrichtlinien

- Einrückung

- Zeilenlänge

- Leerzeilen

## Zusammenfassung

## **5 if-Anweisungen**

Ein einfaches Beispiel

### Bedingungen

- Prüfung auf Gleichheit

- Groß- und Kleinschreibung bei der Prüfung auf Gleichheit

- Prüfung auf Ungleichheit

- Numerische Vergleiche

- Prüfung auf mehrere Bedingungen

- Prüfung auf Vorhandensein eines Werts in einer Liste

- Prüfung auf Abwesenheit eines Werts in einer Liste

- Boolesche Ausdrücke

## if-Anweisungen

- Einfache if-Anweisungen
- if-else-Anweisungen
- Die if-elif-else-Kette
- Mehrere elif-Blöcke
- Den else-Block weglassen
- Mehrere Bedingungen prüfen

## if-Anweisungen für Listen

- Prüfung auf besondere Elemente
- Prüfung auf nicht leere Liste
- Mehrere Listen verwenden

## if-Anweisungen gestalten

## Zusammenfassung

## **6 Dictionaries**

### Ein einfaches Dictionary

### Umgang mit Dictionaries

- Zugriff auf die Werte in einem Dictionary
- Schlüssel-Wert-Paare hinzufügen
- Ein leeres Dictionary als Ausgangspunkt
- Werte in einem Dictionary ändern
- Schlüssel-Wert-Paare entfernen
- Ein Dictionary aus ähnlichen Objekten
- Mit get() auf Werte zugreifen

### Dictionaries in einer Schleife durchlaufen

- Alle Schlüssel-Wert-Paare durchlaufen
- Alle Schlüssel in einem Dictionary durchlaufen
- Die Schlüssel in einem Dictionary geordnet durchlaufen
- Alle Werte in einem Dictionary durchlaufen

### Verschachtelung

- Dictionaries in einer Liste
- Listen in einem Dictionary

Dictionaries in einem Dictionary

Zusammenfassung

## **7 Benutzereingaben und while-Schleifen**

Die Funktion input()

Klar verständliche Eingabeaufforderungen schreiben

Verwendung von int() für numerische Eingaben

Der Modulo-Operator

while-Schleifen

while-Schleifen in Aktion

Programmbeendigung durch den Benutzer

Flags

Eine Schleife mit break verlassen

Die Anweisung continue

Endlosschleifen vermeiden

while-Schleifen für Listen und Dictionaries

Elemente von einer Liste in eine andere verschieben

Alle Vorkommen eines Wertes aus einer Liste entfernen

Ein Dictionary mit Benutzereingaben füllen

Zusammenfassung

## **8 Funktionen**

Funktionen definieren

Informationen an eine Funktion übergeben

Argumente und Parameter

Argumente übergeben

Positionsabhängige Argumente

Schlüsselwortargumente

Standardwerte

Verschiedene Formen für Funktionsaufrufe

Argumentfehler vermeiden

Rückgabewerte

Einen einfachen Wert zurückgeben

Optionale Argumente

Ein Dictionary zurückgeben

Funktionen in einer while-Schleife

Eine Liste übergeben

Eine Liste mithilfe einer Funktion ändern

Die Änderung einer Liste in einer Funktion verhindern

Beliebig viele Argumente übergeben

Positionsabhängige Argumente und Argumente

beliebiger Anzahl kombinieren

Beliebig viele Schlüsselwortargumente übergeben

Funktionen in Modulen speichern

Ein komplettes Modul importieren

Einzelne Funktionen importieren

Eine Funktion mit »as« umbenennen

Ein Modul mit »as« umbenennen

Alle Funktionen eines Moduls importieren

Gestaltung von Funktionen

Zusammenfassung

## **9 Klassen**

Eine Klasse erstellen und verwenden

Die Klasse Dog erstellen

Eine Instanz einer Klasse anlegen

Mit Klassen und Instanzen arbeiten

Die Klasse Car

Einen Standardwert für ein Attribut festlegen

Attributwerte bearbeiten

Vererbung

Die Methode `__init__()` für eine Kindklasse

Attribute und Methoden der Kindklasse definieren

Methoden der Elternklasse überschreiben

Instanzen als Attribute  
Reale Objekte modellieren

Klassen importieren

Eine einzelne Klasse importieren  
Mehrere Klassen in einem Modul speichern  
Mehrere Klassen aus einem Modul importieren  
Ein gesamtes Modul importieren  
Alle Klassen eines Moduls importieren  
Ein Modul in ein Modul importieren  
Aliase verwenden  
Ihren eigenen Arbeitsablauf finden

Die Standardbibliothek von Python

Gestaltung von Klassen

Zusammenfassung

## **10 Dateien und Ausnahmen**

Aus Dateien lesen

Eine gesamte Datei lesen  
Dateipfade  
Zeilenweises Lesen  
Eine Liste aus den Zeilen einer Datei erstellen  
Dateiinhalte verarbeiten  
Große Dateien: eine Million Stellen  
Ist Ihr Geburtsdatum in Pi enthalten?

In Dateien schreiben

In eine leere Datei schreiben  
Mehrere Zeilen schreiben  
Text an eine Datei anhängen

Ausnahmen

Division durch null  
try-except-Blöcke  
Abstürze mithilfe von Ausnahmen verhindern

- Der else-Block
- Datei nicht gefunden
- Text analysieren
- Umgang mit mehreren Dateien
- Fehler stillschweigend übergehen
- Welche Fehler sollten Sie melden und welche nicht?

- Daten speichern
  - json.dump() und json.load()
  - Benutzergenerierte Daten speichern und lesen
  - Refactoring

- Zusammenfassung

## **11 Code testen**

- Funktionen testen
  - Unit Tests und Testfälle
  - Ein bestandener Test
  - Ein nicht bestandener Test
  - Was tun bei einem nicht bestandenen Test?
  - Neue Tests hinzufügen

- Klassen testen
  - Verschiedene Zusicherungsmethoden
  - Eine Beispielklasse zum Testen
  - Die Klasse AnonymousSurvey testen
  - Die Methode setUp()

- Zusammenfassung

## **Teil 2 Projekte**

---

- Alien Invasion - ein Python-Spiel
- Datenvisualisierung
- Webanwendungen

---

# Projekt 1: Alien Invasion

## 12 Das eigene Kampfschiff

Das Projekt planen

Pygame installieren

Erste Schritte für das Spielprojekt

- Ein Pygame-Fenster anlegen und auf Benutzereingaben reagieren

- Die Hintergrundfarbe festlegen

- Eine Klasse für Einstellungen anlegen

Das Bild eines Raumschiffs hinzufügen

- Die Klasse Ship

- Das Schiff auf den Bildschirm zeichnen

Refactoring: Die Methoden `_check_events()` und `_update_screen()`

- Die Methode `_check_events()`

- Die Methode `_update_screen()`

Das Schiff bewegen

- Auf Tastenbetätigungen reagieren

- Kontinuierliche Bewegung

- Bewegung nach rechts und links

- Die Geschwindigkeit des Schiffes anpassen

- Den Bewegungsbereich des Schiffes einschränken

- Refactoring von `_check_events()`

- Beenden mit Q

- Das Spiel im Vollbildmodus ausführen

Zwischenstand

- `alien_invasion.py`

- `settings.py`

- `ship.py`

Geschosse

- Einstellungen für Geschosse hinzufügen

- Die Klasse Bullet
- Geschosse in Gruppen speichern
- Geschosse abfeuern
- Alte Geschosse löschen
- Die Anzahl der Geschosse begrenzen
- Die Methode `_update_bullets()`

Zusammenfassung

## **13 Die Außerirdischen**

Überblick über das Projekt

Das erste Invasionsschiff

- Die Klasse Alien
- Eine Instanz von Alien erstellen

Die Invasionsflotte erstellen

- Wie viele Invasionsschiffe passen in eine Reihe?
- Reihen von Invasionsschiffen erstellen
- Refactoring von `_create_fleet()`
- Reihen hinzufügen

Die Flotte in Bewegung setzen

- Die Invasoren nach rechts bewegen
- Einstellungen für die Flugrichtung der Flotte
- Auf Randberührungen prüfen
- Sinken und Flugrichtung ändern

Invasoren abschießen

- Kollisionen von Geschossen erkennen
- Größere Geschosse zu Testzwecken
- Die Flotte auffüllen
- Die Geschosse beschleunigen
- Refactoring von `_update_bullets()`

Spielende

- Kollisionen zwischen Invasoren und dem eigenen Schiff erkennen

Auf Kollisionen zwischen Invasoren und dem eigenen Schiff reagieren  
Wenn Invasoren den unteren Bildschirmrand erreichen  
Game over!  
Welche Teile des Spiels müssen ausgeführt werden?

Zusammenfassung

## **14 Das Wertungssystem**

Eine Play-Schaltfläche hinzufügen

- Die Klasse Button
- Die Schaltfläche auf den Bildschirm zeichnen
- Das Spiel starten
- Das Spiel zurücksetzen
- Die Play-Schaltfläche deaktivieren
- Den Mauszeiger ausblenden

Levels

- Die Geschwindigkeitseinstellungen ändern
- Die Geschwindigkeit zurücksetzen

Die Punktwertung

- Den Punktestand anzeigen
- Eine Anzeigetafel erstellen
- Den Punktestand bei jedem Abschuss erhöhen
- Den Punktestand zurücksetzen
- Alle Treffer berücksichtigen
- Den Punktwert erhöhen
- Den Punktestand runden
- Highscore
- Das Level anzeigen
- Die Anzahl der verfügbaren Schiffe anzeigen

Zusammenfassung

## **Projekt 2: Datenvisualisierung**

## **15 Daten generieren**

Matplotlib installieren

Einfache Liniendiagramme

- Beschriftung und Linienstärke ändern

- Das Diagramm korrigieren

- Vordefinierte Formatierungen verwenden

- Einzelne Punkte mit scatter() darstellen und gestalten

- Eine Folge von Punkten mit scatter() ausgeben

- Daten automatisch berechnen

- Eigene Farben festlegen

- Eine Colormap verwenden

- Diagramme automatisch speichern

Zufallsbewegungen

- Die Klasse RandomWalk

- Richtungen wählen

- Den Zufallspfad als Diagramm ausgeben

- Mehrere Zufallspfade erstellen

- Den Pfad gestalten

Würfeln mit Plotly

- Plotly installieren

- Die Klasse Die

- Würfeln

- Die Ergebnisse analysieren

- Ein Histogramm erstellen

- Ergebnisse bei zwei Würfeln

- Würfel unterschiedlicher Flächenzahl

Zusammenfassung

## **16 Daten herunterladen**

Das Dateiformat CSV

- CSV-Spaltenköpfe analysieren

- Die Spaltenköpfe und ihre Position ausgeben

- Daten entnehmen und lesen

- Daten in einem Temperaturdiagramm darstellen
- Das Modul datetime
- Datumsangaben im Diagramm darstellen
- Ein Diagramm für einen längeren Zeitraum
- Eine zweite Datenreihe darstellen
- Einen Diagrammbereich einfärben
- Fehlerprüfung
- Daten selbst herunterladen

- Globale Daten im JSON-Format visualisieren
- Erdbebendaten herunterladen
- JSON-Daten untersuchen
- Eine Liste aller Erdbeben aufstellen
- Die Stärken entnehmen
- Ortsdaten entnehmen
- Eine Weltkarte zeichnen
- Eine andere Möglichkeit zur Angabe von Diagrammdaten
- Die Größe der Markierungen anpassen
- Die Farben der Markierungen anpassen
- Weitere Farbpaletten
- Maustext hinzufügen

Zusammenfassung

## **17 APIs**

Web-APIs

- Git und GitHub
- Daten mithilfe eines API-Aufruf anfordern
- Das Paket requests installieren
- API-Antworten verarbeiten
- Das Antwort-Dictionary verarbeiten
- Ein Überblick über die höchstbewerteten Repositories
- Grenzwerte für die API-Aufruftrate

- Angaben zu Repositories mit Plotly visualisieren
- Plotly-Diagramme verbessern

- Eigenen Maustext hinzufügen
- Links zu dem Diagramm hinzufügen
- Mehr über Plotly und die GitHub-API

Die API von Hacker News

Zusammenfassung

## **Projekt 3: Webanwendungen**

### **18 Erste Schritte mit Django**

Ein Projekt einrichten

- Eine Spezifikation schreiben
- Eine virtuelle Umgebung erstellen
- Die virtuelle Umgebung aktivieren
- Django installieren
- Ein Projekt in Django erstellen
- Die Datenbank erstellen
- Das Projekt anzeigen

Eine App anlegen

- Modelle definieren
- Modelle aktivieren
- Die Admin-Site von Django
- Das Modell für die Einträge definieren
- Das Modell Entry in die Datenbank aufnehmen
- Das Modell Entry auf der Admin-Site registrieren
- Die Django-Shell

Seiten erstellen: die Startseite von Learning Log

- Eine URL zuordnen
- Eine Ansicht schreiben
- Eine Vorlage schreiben

Weitere Seiten erstellen

- Vererbung bei Vorlagen
- Die Seite Topics

Einzelne Fachgebieten Seiten

Zusammenfassung

## **19 Benutzerkonten**

Dateneingabe durch die Benutzer

- Neue Fachgebiete hinzufügen

- Neue Einträge hinzufügen

- Einträge bearbeiten

Benutzerkonten einrichten

- Die App users

- Die Anmeldeseite

- Abmelden

- Die Registrierungsseite

Die Benutzer als Besitzer ihrer eigenen Daten

- Den Zugriff mit @login\_required beschränken

- Daten mit Benutzern verknüpfen

- Den Zugriff auf die Fachgebiete auf die zuständigen

- Benutzer einschränken

- Die Fachgebiete eines Benutzers schützen

- Die Seite edit\_entry schützen

- Neue Fachgebiete dem aktuellen Benutzer zuordnen

Zusammenfassung

## **20 Eine App gestalten und bereitstellen**

Learning Log gestalten

- Die App django-bootstrap4

- Learning Log mit Bootstrap gestalten

- Änderungen an base.html

- Die Startseite mit einem Jumbotron gestalten

- Das Anmeldeformular gestalten

- Die Seite Topics gestalten

- Einträge auf den Fachgebieten Seiten gestalten

## Learning Log bereitstellen

- Ein Heroku-Konto anlegen
- Die Heroku-Befehlszeile installieren
- Die erforderlichen Pakete installieren
- Die Datei requirements.txt erstellen
- Die Python-Laufzeitversion angeben
- Die Datei settings.py für Heroku anpassen
- Ein Procfile zum Starten der Prozesse erstellen
- Mit Git den Überblick über die Projektdateien bewahren
- Die Datenbank auf Heroku einrichten
- Die Heroku-Bereitstellung verbessern
- Das Onlineprojekt schützen
- Änderungen mit Commit bestätigen und übertragen
- Umgebungsvariablen auf Heroku einrichten
- Eigene Fehlerseiten erstellen
- Weiterentwicklung des Projekts
- Die Einstellung SECRET\_KEY
- Projekte auf Heroku löschen

## Zusammenfassung

## **Nachwort**

## **Anhang**

---

### **A Installation und Fehlerbehebung**

#### Python unter Windows

- Den Python-Interpreter finden
- Python zur Pfadvariablen hinzufügen
- Python neu installieren

#### Python unter macOS

- Homebrew installieren

Python unter Linux

Schlüsselwörter und integrierte Funktionen

Python-Schlüsselwörter

Integrierte Python-Funktionen

## **B Texteditoren und IDEs**

Die Einstellungen von Sublime Text anpassen

Tabulatoren in Leerzeichen umwandeln

Den Zeilenlängenmarker festlegen

Codeblöcke einrücken und Einrückungen aufheben

Codeblöcke auskommentieren

Die Konfiguration speichern

Weitere Anpassungen

Weitere Texteditoren und IDEs

IDLE

Geany

Emacs und Vim

Atom

Visual Studio Code

PyCharm

Jupyter Notebooks

## **C Hilfe finden**

Erste Schritte

Versuchen Sie es erneut

Legen Sie eine Pause ein

Nutzen Sie das Onlinematerial zu diesem Buch

Online nach Hilfe suchen

Stack Overflow

Die offizielle Python-Dokumentation

Offizielle Dokumentation der Bibliotheken

r/learnpython

Blogs

IRC (Internet Relay Chat)  
Ein IRC-Konto anlegen  
Hilfreiche Kanäle  
IRC-Kultur

Slack

Discord

## **D Versionssteuerung mit Git**

Git installieren

Git unter Windows installieren

Git unter macOS installieren

Git unter Linux installieren

Git konfigurieren

Ein Projekt anlegen

Dateien ignorieren

Ein Repository initialisieren

Den Projektstatus überprüfen

Dateien zum Repository hinzufügen

Einen Commit durchführen

Das Protokoll einsehen

Der zweite Commit

Änderungen zurücknehmen

Vorherige Commits auschecken

Das Repository löschen

## **Stichwortverzeichnis**

*Für meinen Vater, der sich immer Zeit genommen hat,  
meine Fragen über Programmierung zu beantworten, und  
für Ever, der gerade anfängt, mir seine Fragen zu stellen.*

# Der Autor

Eric Matthes ist High-School-Lehrer für Naturwissenschaften und Mathematik in Alaska und gibt dort auch Einführungskurse in Python. Programme schreibt er seit dem Alter von fünf Jahren. Zurzeit konzentriert er sich darauf, Software zu entwickeln, die Lernverfahren effizienter machen soll und die Vorteile von Open-Source-Programmen auf den Bildungsbereich überträgt. Seine Freizeit verbringt er mit Klettern und mit seiner Familie.

# Der Fachgutachter

Kenneth Love arbeitet schon seit vielen Jahren als Python-Programmierer, Lehrer und Organisator von Konferenzen. Er ist auf vielen Konferenzen als Redner und Lehrer aufgetreten und war selbstständig tätig als Python- und Django-Programmierer. Zurzeit arbeitet er als Softwareingenieur für O'Reilly Media. Außerdem gehört er zu den Urhebern des Pakets `django-braces`, das mehrere praktische Mixins für die klassengestützten Ansichten in Django bietet. Auf Twitter können Sie ihm als `@kennethlove` folgen.

# Danksagung

Ohne die wunderbaren und äußerst professionellen Mitarbeiter bei No Starch Press wäre dieses Buch nicht möglich gewesen. Bill Pollock lud mich ein, ein Einführungsbuch zu schreiben, und ich weiß dieses ursprüngliche Angebot sehr zu schätzen. Tyler Ortman hat mir bei der Konzeption dabei geholfen, meinen Gedanken Form zu geben, die ersten Rückmeldungen von Liz Chadwick und Leslie Shen zu den einzelnen Kapiteln waren von unschätzbarem Wert, und Anne Marie Walker hat mir geholfen, viele Teile des Buches klarer zu gestalten. Riley Hoffman hat alle meine Fragen darüber beantwortet, wie man ein komplettes Buch zusammenstellt, und aus meinem Werk mit viel Geduld ein wunderbares, fertiges Produkt gemacht.

Ich möchte auch Kenneth Love danken, dem Fachgutachter für dieses Buch. Ich habe ihn auf einem PyCon kennengelernt habe, und seitdem war sein Engagement für Python und die Python-Community mir stets eine Quelle professioneller Anregung. Kenneth hat nicht nur einfach die Fakten überprüft, sondern das Buch darauf durchgesehen, dass es Anfängern ein solides Verständnis von Python und der Programmierung im Allgemeinen gibt. Für jegliche Unsauberkeiten, die noch in dem Buch vorhanden sein sollten, bin ich jedoch ganz allein verantwortlich.

Des Weiteren möchte ich meinem Vater danken, der mich schon in sehr jungen Jahren in die Programmierung einführte und dabei keine Angst davor hatte, dass ich

seinen Computer beschädigen könnte. Meiner Frau Erin danke ich für ihre Unterstützung und Ermutigung, während ich dieses Buch schrieb, und meinem Sohn Ever für seine Neugier, die mich an jedem Tag aufs Neue inspiriert.

# Vorwort

## zur zweiten Auflage

Die Reaktion auf die erste Auflage von *Python Crashkurs* war überwältigend positiv. Es wurden einschließlich der Übersetzungen in acht Sprachen mehr als 500.000 Exemplare gedruckt. Ich erhielt Briefe und E-Mails von Lesern, sowohl von Zehnjährigen als auch von Rentnern, die in ihrer Freizeit programmieren lernen möchten. *Python Crashkurs* wird im Sekundarstufenunterricht, aber auch in Hochschulkursen eingesetzt. Studenten, an die anspruchsvollere Lehrbücher ausgeteilt wurden, haben *Python Crashkurs* als ergänzenden Text für ihren Kurs benutzt und waren sehr zufrieden damit. Das Buch wird verwendet, um die Programmierfähigkeit für den Beruf zu verbessern und um an Privatprojekten zu arbeiten. Kurzum, es wird so vielfältig genutzt, wie ich es mir gewünscht habe.

Es war überaus erfreulich, eine zweite Auflage von *Python Crashkurs* zu schreiben. Auch wenn Python eine ausgereifte Sprache ist, entwickelt sie sich wie jede andere Sprache weiter. Meine Absicht bei der Überarbeitung des Buches bestand darin, es effizienter und einfacher zu machen. Da es keinen Grund mehr gibt, Python 2 zu lernen, konzentriert sich diese Auflage ausschließlich auf Python 3. Da sich viele Python-Pakete jetzt leichter installieren lassen, wurden auch die Anweisungen zur Einrichtung und Installation vereinfacht. Des Weiteren habe ich einige Themen ergänzt, von denen Sie als Leser profitieren