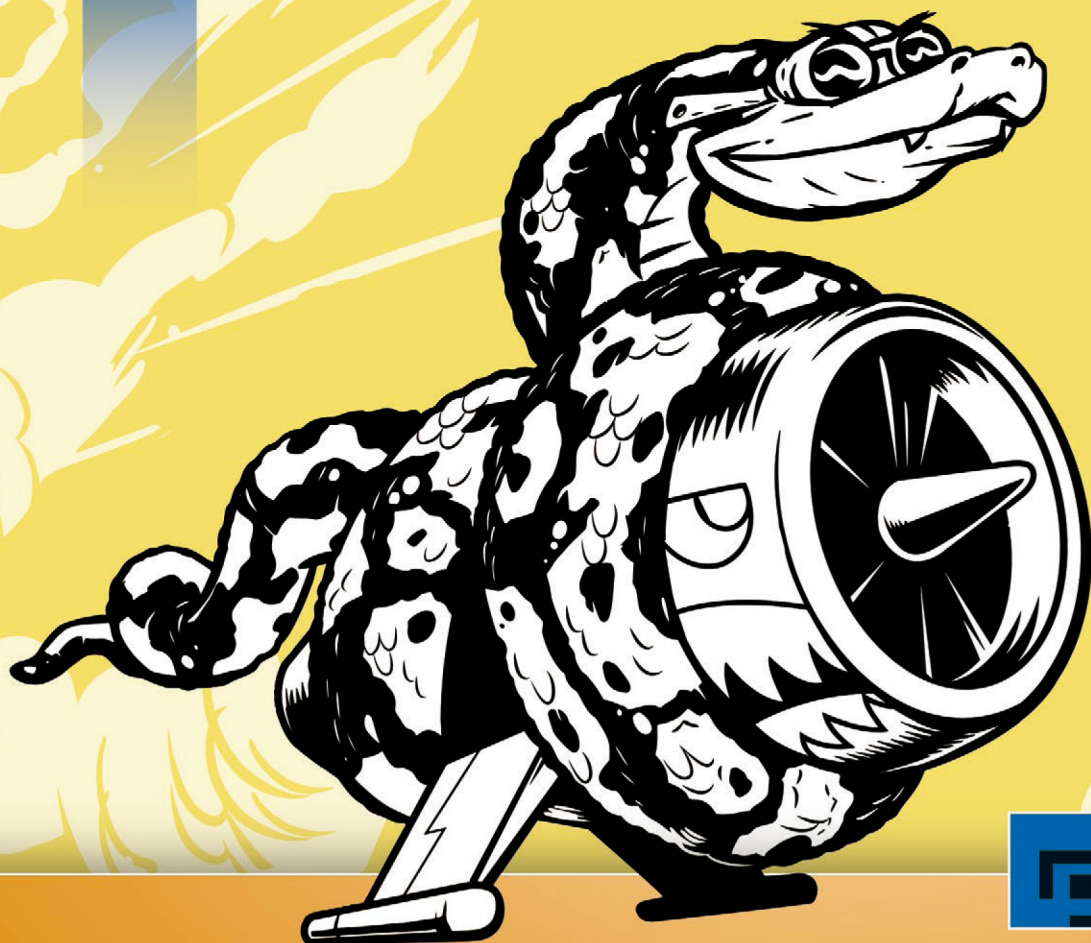


2.

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  
Programmiereinführung**

**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)

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 978-3-86490-735-7

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

Wiebinger 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.

# Inhalt

<b>Der Autor .....</b>	<b>xxii</b>
<b>Der Fachgutachter .....</b>	<b>xxii</b>
<b>Danksagung .....</b>	<b>xxiii</b>
<b>Vorwort zur zweiten Auflage .....</b>	<b>xxv</b>
<b>Einleitung .....</b>	<b>xxix</b>
<b>Teil 1      Grundlagen .....</b>	<b>1</b>
<b>1 Erste Schritte .....</b>	<b>3</b>
Die Programmierumgebung einrichten .....	3
Python-Versionen .....	4
Python-Codeausschnitte ausführen .....	4
Der Editor Sublime Text .....	5
Python auf verschiedenen Betriebssystemen .....	5
Python unter Windows .....	6
Python unter macOS .....	8
Python unter Linux .....	10
Das Hello-World-Programm ausführen .....	11
Sublime Text auf die richtige Python-Version einstellen .....	11
Hello_world.py ausführen .....	12
Fehlersuche .....	12
Python-Programme im Terminal ausführen .....	14
Unter Windows .....	14
Unter Linux und macOS .....	15
Zusammenfassung .....	16

<b>2 Variablen und einfache Datentypen .....</b>	<b>17</b>
Was bei der Ausführung von <code>hello_world.py</code> wirklich geschieht .....	17
Variablen .....	18
Variablen benennen und verwenden .....	19
Fehler bei Variablennamen vermeiden .....	20
Variablen sind Etiketten .....	21
Strings .....	22
Groß- und Kleinschreibung mithilfe von Methoden ändern .....	22
Variablen in Strings verwenden .....	24
Weißraum hinzufügen .....	25
Weißraum entfernen .....	26
Syntaxfehler bei der Stringverarbeitung vermeiden .....	27
Zahlen .....	29
Integer .....	29
Fließkommazahlen .....	30
Integer und Fließkommazahlen .....	31
Unterstriche in Zahlen .....	31
Mehrfachzuweisung .....	32
Konstanten .....	32
Kommentare .....	33
Wie werden Kommentare geschrieben? .....	33
Was für Kommentare sind sinnvoll? .....	33
The Zen of Python .....	34
Zusammenfassung .....	36
 <b>3 Eine Einführung in Listen .....</b>	 <b>37</b>
Was sind Listen? .....	37
Elemente in einer Liste ansprechen .....	38
Indizes beginnen bei 0, nicht bei 1 .....	39
Einzelne Werte aus einer Liste verwenden .....	39
Elemente ändern, hinzufügen und entfernen .....	40
Elemente in einer Liste ändern .....	41
Elemente zu einer Liste hinzufügen .....	41
Elemente aus einer Liste entfernen .....	43
Listen ordnen .....	48
Listen mit <code>sort()</code> dauerhaft sortieren .....	48
Listen mit der Funktion <code>sorted()</code> vorübergehend sortieren .....	48

Listen in umgekehrter Reihenfolge ausgeben .....	49
Die Länge einer Liste ermitteln .....	50
Indexfehler vermeiden .....	51
Zusammenfassung .....	53
<b>4 Mit Listen arbeiten .....</b>	<b>55</b>
Eine komplette Liste durchlaufen .....	55
Die Schleife im Detail .....	56
Weitere Aufgaben in einer for-Schleife erledigen .....	57
Aktionen nach der for-Schleife .....	59
Einrückungsfehler vermeiden .....	60
Vergessene Einrückung der ersten Zeile in einer Schleife .....	60
Vergessene Einrückung nachfolgender Zeilen .....	61
Unnötige Einrückung .....	61
Unnötige Einrückung nach einer Schleife .....	62
Vergessener Doppelpunkt .....	63
Numerische Listen .....	64
Die Funktion range() .....	64
Numerische Listen mithilfe von range() aufstellen .....	65
Einfache Statistiken für numerische Listen .....	66
Listennotation .....	67
Teillisten .....	68
Einen Slice erstellen .....	68
Einen Slice in einer Schleife durchlaufen .....	70
Listen kopieren .....	71
Tupel .....	74
Ein Tupel definieren .....	74
Die Werte in einem Tupel durchlaufen .....	75
Tupel überschreiben .....	75
Code formatieren .....	76
Die Gestaltungsrichtlinien .....	77
Einrückung .....	77
Zeilenlänge .....	78
Leerzeilen .....	78
Zusammenfassung .....	79

<b>5 if-Anweisungen</b>	<b>81</b>
Ein einfaches Beispiel	81
Bedingungen	82
Prüfung auf Gleichheit	82
Groß- und Kleinschreibung bei der Prüfung auf Gleichheit	83
Prüfung auf Ungleichheit	84
Numerische Vergleiche	85
Prüfung auf mehrere Bedingungen	85
Prüfung auf Vorhandensein eines Werts in einer Liste	87
Prüfung auf Abwesenheit eines Werts in einer Liste	87
Boolesche Ausdrücke	88
if-Anweisungen	89
Einfache if-Anweisungen	89
if-else-Anweisungen	90
Die if-elif-else-Kette	91
Mehrere elif-Blöcke	93
Den else-Block weglassen	93
Mehrere Bedingungen prüfen	94
if-Anweisungen für Listen	97
Prüfung auf besondere Elemente	97
Prüfung auf nicht leere Liste	98
Mehrere Listen verwenden	99
if-Anweisungen gestalten	101
Zusammenfassung	102
 <b>6 Dictionaries</b>	 <b>103</b>
Ein einfaches Dictionary	104
Umgang mit Dictionaries	104
Zugriff auf die Werte in einem Dictionary	105
Schlüssel-Wert-Paare hinzufügen	106
Ein leeres Dictionary als Ausgangspunkt	106
Werte in einem Dictionary ändern	107
Schlüssel-Wert-Paare entfernen	109
Ein Dictionary aus ähnlichen Objekten	109
Mit get() auf Werte zugreifen	111
Dictionaries in einer Schleife durchlaufen	113
Alle Schlüssel-Wert-Paare durchlaufen	113
Alle Schlüssel in einem Dictionary durchlaufen	115



Die Schlüssel in einem Dictionary geordnet durchlaufen	117
Alle Werte in einem Dictionary durchlaufen	117
Verschachtelung	120
Dictionaries in einer Liste	120
Listen in einem Dictionary	123
Dictionaries in einem Dictionary	125
Zusammenfassung	127
<b>7 Benutzereingaben und while-Schleifen</b>	<b>129</b>
Die Funktion input()	130
Klar verständliche Eingabeaufforderungen schreiben	130
Verwendung von int() für numerische Eingaben	131
Der Modulo-Operator	133
while-Schleifen	134
while-Schleifen in Aktion	134
Programmbeendigung durch den Benutzer	135
Flags	137
Eine Schleife mit break verlassen	138
Die Anweisung continue	139
Endlosschleifen vermeiden	140
while-Schleifen für Listen und Dictionaries	141
Elemente von einer Liste in eine andere verschieben	142
Alle Vorkommen eines Wertes aus einer Liste entfernen	143
Ein Dictionary mit Benutzereingaben füllen	143
Zusammenfassung	145
<b>8 Funktionen</b>	<b>147</b>
Funktionen definieren	148
Informationen an eine Funktion übergeben	148
Argumente und Parameter	149
Argumente übergeben	150
Positionsabhängige Argumente	150
Schlüsselwortargumente	152
Standardwerte	153
Verschiedene Formen für Funktionsaufrufe	154
Argumentfehler vermeiden	155
Rückgabewerte	156
Einen einfachen Wert zurückgeben	157

Optionale Argumente .....	157
Ein Dictionary zurückgeben .....	159
Funktionen in einer while-Schleife .....	160
Eine Liste übergeben .....	162
Eine Liste mithilfe einer Funktion ändern .....	163
Die Änderung einer Liste in einer Funktion verhindern .....	166
Beliebig viele Argumente übergeben .....	167
Positionsabhängige Argumente und Argumente beliebiger Anzahl kombinieren .....	168
Beliebig viele Schlüsselwortargumente übergeben .....	169
Funktionen in Modulen speichern .....	171
Ein komplettes Modul importieren .....	171
Einzelne Funktionen importieren .....	172
Eine Funktion mit »as« umbenennen .....	173
Ein Modul mit »as« umbenennen .....	174
Alle Funktionen eines Moduls importieren .....	174
Gestaltung von Funktionen .....	175
Zusammenfassung .....	176
<b>9 Klassen .....</b>	<b>179</b>
Eine Klasse erstellen und verwenden .....	180
Die Klasse Dog erstellen .....	180
Eine Instanz einer Klasse anlegen .....	182
Mit Klassen und Instanzen arbeiten .....	185
Die Klasse Car .....	185
Einen Standardwert für ein Attribut festlegen .....	186
Attributwerte bearbeiten .....	187
Vererbung .....	191
Die Methode __init__() für eine Kindklasse .....	191
Attribute und Methoden der Kindklasse definieren .....	193
Methoden der Elternklasse überschreiben .....	194
Instanzen als Attribute .....	194
Reale Objekte modellieren .....	197
Klassen importieren .....	198
Eine einzelne Klasse importieren .....	198
Mehrere Klassen in einem Modul speichern .....	200
Mehrere Klassen aus einem Modul importieren .....	201

Ein gesamtes Modul importieren . . . . .	202
Alle Klassen eines Moduls importieren . . . . .	202
Ein Modul in ein Modul importieren . . . . .	203
Aliase verwenden . . . . .	204
Ihren eigenen Arbeitsablauf finden . . . . .	205
Die Standardbibliothek von Python . . . . .	205
Gestaltung von Klassen . . . . .	207
Zusammenfassung . . . . .	207
<b>10 Dateien und Ausnahmen . . . . .</b>	<b>209</b>
Aus Dateien lesen . . . . .	210
Eine gesamte Datei lesen . . . . .	210
Dateipfade . . . . .	212
Zeilenweises Lesen . . . . .	213
Eine Liste aus den Zeilen einer Datei erstellen . . . . .	214
Dateiinhalte verarbeiten . . . . .	215
Große Dateien: eine Million Stellen . . . . .	216
Ist Ihr Geburtsdatum in Pi enthalten? . . . . .	217
In Dateien schreiben . . . . .	218
In eine leere Datei schreiben . . . . .	218
Mehrere Zeilen schreiben . . . . .	219
Text an eine Datei anhängen . . . . .	220
Ausnahmen . . . . .	221
Division durch null . . . . .	221
try-except-Blöcke . . . . .	222
Abstürze mithilfe von Ausnahmen verhindern . . . . .	223
Der else-Block . . . . .	224
Datei nicht gefunden . . . . .	225
Text analysieren . . . . .	226
Umgang mit mehreren Dateien . . . . .	227
Fehler stillschweigend übergehen . . . . .	229
Welche Fehler sollten Sie melden und welche nicht? . . . . .	230
Daten speichern . . . . .	231
json.dump() und json.load() . . . . .	232
Benutzergenerierte Daten speichern und lesen . . . . .	233
Refactoring . . . . .	235
Zusammenfassung . . . . .	238

**11 Code testen ..... 239**

Funktionen testen .....	240
Unit Tests und Testfälle .....	241
Ein bestandener Test .....	241
Ein nicht bestandener Test .....	243
Was tun bei einem nicht bestandenem Test? .....	244
Neue Tests hinzufügen .....	246
Klassen testen .....	247
Verschiedene Zusicherungsmethoden .....	247
Eine Beispielklasse zum Testen .....	248
Die Klasse AnonymousSurvey testen .....	250
Die Methode setUp() .....	252
Zusammenfassung .....	254

**Teil 2     Projekte ..... 257**

Alien Invasion – ein Python-Spiel .....	257
Datenvisualisierung .....	258
Webanwendungen .....	258

**Projekt 1: Alien Invasion ..... 259****12 Das eigene Kampfschiff ..... 261**

Das Projekt planen .....	262
Pygame installieren .....	263
Erste Schritte für das Spielprojekt .....	263
Ein Pygame-Fenster anlegen und auf Benutzereingaben reagieren ....	263
Die Hintergrundfarbe festlegen .....	265
Eine Klasse für Einstellungen anlegen .....	266
Das Bild eines Raumschiffs hinzufügen .....	267
Die Klasse Ship .....	269
Das Schiff auf den Bildschirm zeichnen .....	270
Refactoring: Die Methoden _check_events() und _update_screen() .....	272
Die Methode _check_events() .....	272
Die Methode _update_screen() .....	273
Das Schiff bewegen .....	274
Auf Tastenbetätigungen reagieren .....	274
Kontinuierliche Bewegung .....	275

Bewegung nach rechts und links	277
Die Geschwindigkeit des Schiffes anpassen	278
Den Bewegungsbereich des Schiffes einschränken	280
Refactoring von <code>_check_events()</code>	281
Beenden mit Q	282
Das Spiel im Vollbildmodus ausführen	282
Zwischenstand	283
<code>alien_invasion.py</code>	283
<code>settings.py</code>	284
<code>ship.py</code>	284
Geschosse	284
Einstellungen für Geschosse hinzufügen	285
Die Klasse <code>Bullet</code>	285
Geschosse in Gruppen speichern	287
Geschosse abfeuern	287
Alte Geschosse löschen	289
Die Anzahl der Geschosse begrenzen	290
Die Methode <code>_update_bullets()</code>	291
Zusammenfassung	292
<b>13 Die Außerirdischen</b>	<b>293</b>
Überblick über das Projekt	294
Das erste Invasionsschiff	294
Die Klasse <code>Alien</code>	295
Eine Instanz von <code>Alien</code> erstellen	296
Die Invasionsflotte erstellen	298
Wie viele Invasionsschiffe passen in eine Reihe?	298
Reihen von Invasionsschiffen erstellen	299
Refactoring von <code>_create_fleet()</code>	300
Reihen hinzufügen	301
Die Flotte in Bewegung setzen	304
Die Invasoren nach rechts bewegen	304
Einstellungen für die Flugrichtung der Flotte	305
Auf Randberührungen prüfen	306
Sinken und Flugrichtung ändern	307
Invasoren abschießen	308
Kollisionen von Geschossen erkennen	308
Größere Geschosse zu Testzwecken	310

Die Flotte auffüllen . . . . .	311
Die Geschosse beschleunigen . . . . .	312
Refactoring von <code>_update_bullets()</code> . . . . .	312
Spielende . . . . .	313
Kollisionen zwischen Invasoren und dem eigenen Schiff erkennen . . .	313
Auf Kollisionen zwischen Invasoren und dem eigenen Schiff reagieren	314
Wenn Invasoren den unteren Bildschirmrand erreichen . . . . .	317
Game over! . . . . .	318
Welche Teile des Spiels müssen ausgeführt werden? . . . . .	319
Zusammenfassung . . . . .	320
<b>14 Das Wertungssystem . . . . .</b>	<b>321</b>
Eine Play-Schaltfläche hinzufügen . . . . .	321
Die Klasse Button . . . . .	322
Die Schaltfläche auf den Bildschirm zeichnen . . . . .	324
Das Spiel starten . . . . .	325
Das Spiel zurücksetzen . . . . .	326
Die Play-Schaltfläche deaktivieren . . . . .	327
Den Mauszeiger ausblenden . . . . .	327
Levels . . . . .	328
Die Geschwindigkeitseinstellungen ändern . . . . .	329
Die Geschwindigkeit zurücksetzen . . . . .	330
Die Punktwertung . . . . .	331
Den Punktestand anzeigen . . . . .	332
Eine Anzeigetafel erstellen . . . . .	333
Den Punktestand bei jedem Abschuss erhöhen . . . . .	335
Den Punktestand zurücksetzen . . . . .	335
Alle Treffer berücksichtigen . . . . .	336
Den Punktwert erhöhen . . . . .	337
Den Punktestand runden . . . . .	338
Highscore . . . . .	339
Das Level anzeigen . . . . .	341
Die Anzahl der verfügbaren Schiffe anzeigen . . . . .	344
Zusammenfassung . . . . .	347

<b>Projekt 2: Datenvisualisierung .....</b>	<b>349</b>
<b>15 Daten generieren .....</b>	<b>351</b>
Matplotlib installieren .....	352
Einfache Liniendiagramme .....	353
Beschriftung und Linienstärke ändern .....	354
Das Diagramm korrigieren .....	355
Vordefinierte Formatierungen verwenden .....	356
Einzelne Punkte mit scatter() darstellen und gestalten .....	358
Eine Folge von Punkten mit scatter() ausgeben .....	359
Daten automatisch berechnen .....	360
Eigene Farben festlegen .....	361
Eine Colormap verwenden .....	362
Diagramme automatisch speichern .....	363
Zufallsbewegungen .....	364
Die Klasse RandomWalk .....	364
Richtungen wählen .....	365
Den Zufallspfad als Diagramm ausgeben .....	366
Mehrere Zufallspfade erstellen .....	367
Den Pfad gestalten .....	368
Würfeln mit Plotly .....	373
Plotly installieren .....	373
Die Klasse Die .....	374
Würfeln .....	374
Die Ergebnisse analysieren .....	375
Ein Histogramm erstellen .....	376
Ergebnisse bei zwei Würfeln .....	378
Würfel unterschiedlicher Flächenzahl .....	380
Zusammenfassung .....	382
<b>16 Daten herunterladen .....</b>	<b>383</b>
Das Dateiformat CSV .....	384
CSV-Spaltenköpfe analysieren .....	384
Die Spaltenköpfe und ihre Position ausgeben .....	385
Daten entnehmen und lesen .....	386
Daten in einem Temperaturdiagramm darstellen .....	387
Das Modul datetime .....	388

Datumsangaben im Diagramm darstellen .....	389
Ein Diagramm für einen längeren Zeitraum .....	391
Eine zweite Datenreihe darstellen .....	392
Einen Diagrammbereich einfärben .....	393
Fehlerprüfung .....	394
Daten selbst herunterladen .....	398
Globale Daten im JSON-Format visualisieren .....	399
Erdbebedaten herunterladen .....	400
JSON-Daten untersuchen .....	400
Eine Liste aller Erdbeben aufstellen .....	403
Die Stärken entnehmen .....	403
Ortsdaten entnehmen .....	404
Eine Weltkarte zeichnen .....	405
Eine andere Möglichkeit zur Angabe von Diagrammdaten .....	406
Die Größe der Markierungen anpassen .....	407
Die Farben der Markierungen anpassen .....	408
Weitere Farbpaletten .....	410
Maustext hinzufügen .....	410
Zusammenfassung .....	412
<b>17 APIs .....</b>	<b>415</b>
Web-APIs .....	415
Git und GitHub .....	416
Daten mithilfe eines API-Aufruf anfordern .....	416
Das Paket requests installieren .....	417
API-Antworten verarbeiten .....	418
Das Antwort-Dictionary verarbeiten .....	419
Ein Überblick über die höchstbewerteten Repositories .....	421
Grenzwerte für die API-Aufruftrate .....	422
Angaben zu Repositories mit Plotly visualisieren .....	423
Plotly-Diagramme verbessern .....	425
Eigenen Maustext hinzufügen .....	427
Links zu dem Diagramm hinzufügen .....	429
Mehr über Plotly und die GitHub-API .....	430
Die API von Hacker News .....	430
Zusammenfassung .....	434



<b>Projekt 3: Webanwendungen</b>	<b>435</b>
<b>18 Erste Schritte mit Django</b>	<b>437</b>
Ein Projekt einrichten	438
Eine Spezifikation schreiben	438
Eine virtuelle Umgebung erstellen	438
Die virtuelle Umgebung aktivieren	439
Django installieren	440
Ein Projekt in Django erstellen	440
Die Datenbank erstellen	441
Das Projekt anzeigen	442
Eine App anlegen	444
Modelle definieren	444
Modelle aktivieren	446
Die Admin-Site von Django	447
Das Modell für die Einträge definieren	450
Das Modell Entry in die Datenbank aufnehmen	451
Das Modell Entry auf der Admin-Site registrieren	452
Die Django-Shell	453
Seiten erstellen: die Startseite von Learning Log	455
Eine URL zuordnen	456
Eine Ansicht schreiben	458
Eine Vorlage schreiben	458
Weitere Seiten erstellen	460
Vererbung bei Vorlagen	460
Die Seite Topics	463
Einzelne Fachgebietsseiten	466
Zusammenfassung	470
<b>19 Benutzerkonten</b>	<b>473</b>
Dateneingabe durch die Benutzer	474
Neue Fachgebiete hinzufügen	474
Neue Einträge hinzufügen	479
Einträge bearbeiten	483
Benutzerkonten einrichten	487
Die App users	487
Die Anmeldeseite	488

Abmelden .....	491
Die Registrierungsseite .....	493
Die Benutzer als Besitzer ihrer eigenen Daten .....	496
Den Zugriff mit @login_required beschränken .....	496
Daten mit Benutzern verknüpfen .....	498
Den Zugriff auf die Fachgebiete auf die zuständigen Benutzer einschränken .....	502
Die Fachgebiete eines Benutzers schützen .....	502
Die Seite edit_entry schützen .....	503
Neue Fachgebiete dem aktuellen Benutzer zuordnen .....	504
Zusammenfassung .....	505
<b>20 Eine App gestalten und bereitstellen .....</b>	<b>507</b>
Learning Log gestalten .....	508
Die App django-bootstrap4 .....	508
Learning Log mit Bootstrap gestalten .....	509
Änderungen an base.html .....	510
Die Startseite mit einem Jumbotron gestalten .....	514
Das Anmeldeformular gestalten .....	516
Die Seite Topics gestalten .....	517
Einträge auf den Fachgebieten Seiten gestalten .....	518
Learning Log bereitstellen .....	520
Ein Heroku-Konto anlegen .....	520
Die Heroku-Befehlszeile installieren .....	520
Die erforderlichen Pakete installieren .....	521
Die Datei requirements.txt erstellen .....	521
Die Python-Laufzeitversion angeben .....	522
Die Datei settings.py für Heroku anpassen .....	523
Ein Procfile zum Starten der Prozesse erstellen .....	523
Mit Git den Überblick über die Projektdateien bewahren .....	523
Die Datenbank auf Heroku einrichten .....	528
Die Heroku-Bereitstellung verbessern .....	528
Das Onlineprojekt schützen .....	530
Änderungen mit Commit bestätigen und übertragen .....	531
Umgebungsvariablen auf Heroku einrichten .....	533
Eigene Fehlerseiten erstellen .....	533
Weiterentwicklung des Projekts .....	536

Die Einstellung SECRET_KEY .....	537
Projekte auf Heroku löschen .....	537
Zusammenfassung .....	539

<b>Nachwort .....</b>	<b>541</b>
-----------------------	------------

<b>Anhang .....</b>	<b>543</b>
---------------------	------------

<b>A Installation und Fehlerbehebung .....</b>	<b>543</b>
--	------------

Python unter Windows .....	543
Den Python-Interpreter finden .....	544
Python zur Pfadvariablen hinzufügen .....	544
Python neu installieren .....	545
Python unter macOS .....	545
Homebrew installieren .....	546
Python unter Linux .....	547
Schlüsselwörter und integrierte Funktionen .....	547
Python-Schlüsselwörter .....	548
Integrierte Python-Funktionen .....	548

<b>B Texteditoren und IDEs .....</b>	<b>549</b>
--------------------------------------	------------

Die Einstellungen von Sublime Text anpassen .....	550
Tabulatoren in Leerzeichen umwandeln .....	550
Den Zeilenlängenmarker festlegen .....	551
Codeblöcke einrücken und Einrückungen aufheben .....	551
Codeblöcke auskommentieren .....	551
Die Konfiguration speichern .....	551
Weitere Anpassungen .....	552
Weitere Texteditoren und IDEs .....	552
IDLE .....	552
Geany .....	552
Emacs und Vim .....	553
Atom .....	553
Visual Studio Code .....	553
PyCharm .....	553
Jupyter Notebooks .....	554

<b>C</b>	<b>Hilfe finden</b>	<b>555</b>
Erste Schritte		555
Versuchen Sie es erneut		556
Legen Sie eine Pause ein		556
Nutzen Sie das Onlinematerial zu diesem Buch		557
Online nach Hilfe suchen		557
Stack Overflow		557
Die offizielle Python-Dokumentation		558
Offizielle Dokumentation der Bibliotheken		558
r/learnpython		558
Blogs		558
IRC (Internet Relay Chat)		559
Ein IRC-Konto anlegen		559
Hilfreiche Kanäle		560
IRC-Kultur		560
Slack		560
Discord		561
<b>D</b>	<b>Versionssteuerung mit Git</b>	<b>563</b>
Git installieren		564
Git unter Windows installieren		564
Git unter macOS installieren		564
Git unter Linux installieren		564
Git konfigurieren		564
Ein Projekt anlegen		565
Dateien ignorieren		565
Ein Repository initialisieren		565
Den Projektstatus überprüfen		566
Dateien zum Repository hinzufügen		566
Einen Commit durchführen		567
Das Protokoll einsehen		568
Der zweite Commit		568
Änderungen zurücknehmen		569
Vorherige Commits auschecken		571
Das Repository löschen		572
	<b>Stichwortverzeichnis</b>	<b>575</b>

*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 können, und einige

Abschnitte auf den neuesten Stand gebracht, um neue, einfachere Möglichkeiten darzustellen, die es jetzt zur Erledigung von Aufgaben in Python gibt. Außerdem habe ich einige Abschnitte deutlicher formuliert, in denen Aspekte der Sprache nicht klar genug dargestellt wurden. Alle Projekte wurden unter Zuhilfenahme weitverbreiteter, gut gepflegter Bibliotheken komplett überarbeitet, die Sie auch vertrauensvoll für Ihre eigenen Projekte einsetzen können.

Der folgende Überblick führt besondere Änderungen auf, die ich in dieser zweiten Auflage vorgenommen habe:

- In Kapitel 1 wurden die Anweisungen zur Installation von Python für die Benutzer der am weitesten verbreiteten Betriebssysteme vereinfacht. Außerdem empfehle ich jetzt den Editor Sublime Text, der sowohl bei Einsteigern als auch bei Profis beliebt ist und in allen Betriebssystemen gut funktioniert.
- Kapitel 2 gibt jetzt eine korrekte Erklärung darüber, wie Variablen in Python implementiert sind. Sie werden als *Etiketten* für Werte beschrieben, was deutlicher macht, wie sich Variablen in Python verhalten. Außerdem werden in diesem Buch jetzt die in Python 3.6 eingeführten F-Strings verwendet, die eine viel einfachere Methode bieten, um Variablenwerte in Strings einzufügen. Die Verwendung von Unterstrichen als Tausendertrennzeichen, z.B. in 1\_000\_000, wurde ebenfalls in Python 3.6 eingeführt und wird in dieser Auflage vorgestellt. Die Mehrfachzuweisung von Variablen war in der ersten Auflage in einem der Projekte eingeführt worden. Diese Erläuterung wurde verallgemeinert und in das Kapitel 2 verlegt. Schließlich wurde auch eine eindeutige Konvention für die Darstellung von konstanten Werten in Python in dieses Kapitel eingefügt.
- In Kapitel 6 führe ich die Methode `get()` ein, um Werte aus einem Dictionary abzurufen. Sie kann einen Standardwert zurückgeben, wenn der angegebene Schlüssel nicht existiert.
- Das Projekt *Alien Invasion* (Kapitel 12 bis 14) ist in dieser Version komplett auf Klassen aufgebaut. Das Spiel selbst ist jetzt eine Klasse und nicht mehr eine Folge von Funktionen. Das vereinfacht die Grundstruktur des Spiels sehr stark und reduziert die Anzahl der erforderlichen Funktionsaufrufe und Parameter erheblich. Leser, die mit der ersten Auflage vertraut sind, werden die Einfachheit dieser neuen, klassengestützten Herangehensweise zu schätzen wissen. Pygame kann jetzt auf allen Systemen mit nur einer Zeile installiert werden. Außerdem besteht die Möglichkeit, das Spiel im Vollbildmodus oder in einem Fenster auszuführen.
- Bei den Datenvisualisierungsprojekten wurden die Installationsanweisungen für Matplotlib für alle Betriebssysteme vereinfacht. Bei den Visualisierungen mit Matplotlib wird jetzt die Funktion `subplots()` verwendet, die eine einfa-

chere Grundlage bildet, wenn Sie das Erstellen komplizierterer Visualisierungen lernen. Für das Würfelprojekt in Kapitel 15 wird jetzt Plotly eingesetzt, eine gut gepflegte Visualisierungsbibliothek mit einer übersichtlichen Syntax und einer schönen, komplett anpassbaren Ausgabe.

- Das Wetterprojekt aus Kapitel 16 basiert jetzt auf den Daten der NOAA, deren Website in den nächsten Jahren etwas stabiler sein sollte als diejenige, die in der ersten Auflage verwendet wurde. Bei dem Kartierungsprojekt geht es dieses Mal um die Erdbebenaktivität weltweit. Zum Abschluss des Projekts haben Sie eine atemberaubende Visualisierung, die durch die Darstellung der Positionen aller Erdbeben in einem gegebenen Zeitraum die Ränder der tektonischen Platten aufzeigt. Dabei lernen Sie, jegliche Art von Datenmengen mit geografischem Bezug grafisch darzustellen.
- In Kapitel 17 wird wiederum Plotly verwendet, um Aktivitäten rund um Python in den Open-Source-Projekten auf GitHub zu visualisieren.
- Das Projekt Learning Log (Kapitel 18 bis 20) wird mit der neuesten Version von Django erstellt und mit der neuesten Version von Bootstrap gestaltet. Die Bereitstellung des Projekts auf Heroku wurde mit dem Paket `django-heroku` vereinfacht und nutzt Umgebungsvariablen, anstatt die Datei `settings.py` zu ändern. Diese Vorgehensweise ist einfacher und auch stärker daran angelehnt, wie professionelle Programmierer moderne Django-Projekte bereitstellen.
- Anhang A wurde komplett auf den neuesten Stand gebracht, um die aktuell empfohlenen Vorgehensweisen zur Installation von Python widerzuspiegeln. Anhang B enthält jetzt ausführliche Anleitungen zur Einrichtung von Sublime Text und kurze Beschreibungen der wichtigsten Texteditoren und IDEs, die zurzeit in Gebrauch sind. Anhang C gibt Hinweise zu neueren und populäreren Onlineressourcen, in denen Sie Hilfe erhalten können, und in Anhang D finden Sie weiterhin einen Mini-Crashkurs zur Verwendung von Git für die Versionssteuerung.
- Der Index wurde gründlich überarbeitet und ermöglicht es Ihnen, dieses Buch auch als Nachschlagewerk für Ihre zukünftigen Python-Projekte zu verwenden.

Vielen Dank dafür, dass Sie *Python Crashkurs* lesen! Wenn Sie irgendwelche Rückmeldungen oder Fragen an mich haben, können Sie sich gern an mich wenden.



# 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-