

2.

Auflage

dpunkt.verlag

Aktualisiert
auf
Python 3



Al Sweigart

Routineaufgaben mit Python automatisieren

Praktische Programmierlösungen für Einsteiger

Papier
plus⁺
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⁺:

www.dpunkt.plus

Al Sweigart

Routineaufgaben mit Python automatisieren

Praktische Programmierlösungen für Einsteiger

2., aktualisierte und erweiterte Auflage



dpunkt.verlag

Al Sweigart

Lektorat: Dr. Michael Barabas

Lektoratsassistentin: Anja Weimer

Fachgutachter/in: Ari Lacenski und Philip James

Übersetzung & Satz: G&U Language & Publishing Services GmbH, www.gundu.com

Copy-Editing: Ursula Zimpfer, Herrenberg

Herstellung: Stefanie Weidner

Umschlaggestaltung: Helmut Kraus, www.exclam.de

nach der Originalvorlage von No Starch Press

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 978-3-86490-753-1

PDF 978-3-96088-956-4

ePub 978-3-96088-957-1

mobi 978-3-96088-958-8

2., aktualisierte und erweiterte Auflage 2020

Copyright © 2020 dpunkt.verlag GmbH

Wieblinger Weg 17

69123 Heidelberg

Copyright © 2020 by Al Sweigart. Title of English-language original: Automate the Boring Stuff with Python, 2th Edition: Practical Programming for Total Beginners, ISBN 978-1-59327-992-9, 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.



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	xxvi
Der Fachgutachter	xxvi
Danksagung	xxvii
Einleitung	1
Für wen ist dieses Buch gedacht?	2
Programmierstil	3
Was ist Programmierung?	3
Was ist Python?	4
Programmierer müssen nicht viel Mathe können	4
Sie sind nie zu alt, um programmieren zu lernen	5
Programmierung ist kreativ	6
Der Aufbau dieses Buchs	6
Python herunterladen und installieren	9
Mu herunterladen und installieren	10
Mu starten	10
IDLE starten	11
Die interaktive Shell	11
Drittanbietermodule installieren	12
Hilfe finden	13
Sinnvolle Fragen stellen	14
Zusammenfassung	16

Teil 1	Grundlagen der Python-Programmierung	17
1	Grundlagen von Python	19
	Ausdrücke in die interaktive Shell eingeben	20
	Die Datentypen für ganze Zahlen, Fließkommazahlen und Strings	23
	Stringverkettung und -wiederholung	24
	Werte in Variablen speichern	25
	Zuweisungsanweisungen	25
	Variablenamen	27
	Ihr erstes Programm	28
	Analyse des Programms	30
	Kommentare	30
	Die Funktion print()	31
	Die Funktion input()	31
	Den Benutzernamen ausgeben	32
	Die Funktion len()	32
	Die Funktionen str(), int() und float()	33
	Zusammenfassung	36
	Wiederholungsfragen	37
2	Flusssteuerung	39
	Boolesche Werte	40
	Vergleichsoperatoren	41
	Boolesche Operatoren	43
	Binäre boolesche Operatoren	43
	Der Operator not	44
	Kombinierte Verwendung von booleschen und Vergleichsoperatoren	45
	Elemente zur Flusssteuerung	46
	Bedingungen	46
	Codeblöcke	46
	Programmausführung	47
	Flusssteuerungsanweisungen	47
	If-Anweisungen	47
	Else-Anweisungen	49
	Elif-Anweisungen	50
	While-Schleifen	57
	Break-Anweisungen	61

Continue-Anweisungen	62
For-Schleifen und die Funktion range()	66
Module importieren	69
From-import-Anweisungen	71
Programme mit sys.exit() vorzeitig beenden	71
Ein kurzes Programm: Zahlen raten	72
Ein kurzes Programm: Stein, Schere, Papier	74
Zusammenfassung	78
Wiederholungsfragen	78
3 Funktionen	81
Def-Anweisungen mit Parametern	83
Terminologie	84
Rückgabewerte und die Anweisung return	84
Der Wert None	86
Schlüsselwortargumente und print()	87
Der Aufrufstack	88
Lokaler und globaler Gültigkeitsbereich	90
Lokale Variablen können im globalen Gültigkeitsbereich nicht verwendet werden	92
Lokale Gültigkeitsbereiche können keine Variablen aus anderen lokalen Gültigkeitsbereichen verwenden	92
Globale Variablen können von einem lokalen Gültigkeitsbereich aus gelesen werden	93
Lokale und globale Variablen mit demselben Namen	94
Die Anweisung global	95
Ausnahmebehandlung	97
Ein kurzes Programm: Zickzackmuster	99
Zusammenfassung	102
Wiederholungsfragen	102
Übungsprojekte	103
Die Collatz-Folge	103
Eingabevalidierung	104
4 Listen	105
Der Datentyp für Listen	106
Einzelne Elemente einer Liste mithilfe des Index abrufen	106

Negative Indizes	108
Teillisten mithilfe von Slices abrufen	108
Die Länge einer Liste mit len() abrufen	109
Werte in einer Liste mithilfe des Index ändern	109
Listenverkettung und -wiederholung	110
Elemente mit del aus einer Liste entfernen	110
Listen verwenden	110
Listen in for-Schleifen	112
Die Operatoren in und not in	113
Der Trick mit der Mehrfachzuweisung	114
Die Funktion enumerate() für Listen	115
Die Funktionen random.choice() und random.shuffle() für Listen	115
Erweiterte Zuweisungsoperatoren	116
Methoden	117
Elemente in einer Liste mit der Methode index() finden	117
Elemente mit den Methoden append() und insert() zu Listen hinzufügen	118
Elemente mit remove() aus Listen entfernen	119
Elemente in einer Liste mit sort() sortieren	119
Reihenfolge der Listeneinträge mit reverse() umkehren	121
Beispielprogramm: Magic 8 Ball mithilfe einer Liste	122
Sequenzielle Datentypen	122
Veränderbare und unveränderbare Datentypen	123
Der Datentyp für Tupel	125
Typen mit den Funktionen list() und tuple() umwandeln	126
Verweise	127
Identität und die Funktion id()	129
Verweise übergeben	130
Die Funktionen copy() und deepcopy() des Moduls copy	131
Ein kurzes Programm: Conways Spiel des Lebens	132
Zusammenfassung	137
Wiederholungsfragen	138
Übungsprojekte	139
Kommacode	139
Münzwurffolgen	139
Zeichenbildraster	140

5 Dictionaries und Datenstrukturen	143
Der Datentyp für Dictionaries	143
Dictionaries und Listen im Vergleich	144
Die Methoden keys(), values() und items()	146
Das Vorhandensein eines Schlüssels oder Wertes im Dictionary ermitteln	148
Die Methode get()	148
Die Methode setdefault()	149
Saubere Ausgabe	150
Datenstrukturen zur Modellierung realer Objekte	151
Ein Tic-Tac-Toe-Brett	153
Verschachtelte Dictionaries und Listen	158
Zusammenfassung	160
Wiederholungsfragen	160
Übungsprojekte	161
Validierer für Schach-Dictionary	161
Inventar für ein Fantasyspiel	161
Eine Funktion zum Hinzufügen von Listeninhalten zum Inventar- Dictionary	162
6 Stringbearbeitung	163
Umgang mit Strings	164
Stringlitterale	164
Strings indizieren und Slices entnehmen	167
Die Operatoren in und not in für Strings	168
Strings in andere Strings einfügen	168
Nützliche Stringmethoden	169
Die Stringmethoden upper(), lower(), isupper() und islower()	169
Die isX-Stringmethoden	171
Die Stringmethoden startswith() und endswith()	173
Die Methoden join() und split()	173
Strings mit der Methode partition() aufteilen	175
Text mit rjust(), ljust() und center() ausrichten	176
Weißraum mit strip(),rstrip() und lstrip() entfernen	178
Die Funktionen ord() und chr() für numerische Zeichenwerte	178
Strings mit dem Modul pyperclip kopieren und einfügen	179

Projekt: Automatisierte Nachrichten mithilfe einer Mehrfach-Zwischenablage	180
Schritt 1: Programmdesign und Datenstrukturen	181
Schritt 2: Befehlszeilenargumente verarbeiten	181
Schritt 3: Die richtige Antwort kopieren	182
Projekt: Aufzählungspunkte zu einem Wiki-Markup hinzufügen	183
Schritt 1: Text von und zur Zwischenablage übertragen	184
Schritt 2: Textzeilen trennen und Sternchen hinzufügen	184
Schritt 3: Die veränderten Zeilen zusammenfügen	185
Ein kurzes Programm: Pig Latin	186
Zusammenfassung	190
Wiederholungsfragen	191
Übungsprojekt	192
Tabellenausgabe	192
Bots für Zombie Dice	192

Teil 2 Aufgaben automatisieren **197**

7 Mustervergleich mit regulären Ausdrücken	199
Textmuster ohne reguläre Ausdrücke finden	200
Textmuster mithilfe regulärer Ausdrücke finden	203
Regex-Objekte erstellen	203
Vergleiche mit einem Regex-Objekt	204
Zusammenfassung: Mustervergleich mit regulären Ausdrücken	205
Weitere Möglichkeiten für den Mustervergleich mithilfe regulärer Ausdrücke	205
Gruppierung durch Klammern	205
Mithilfe der Pipe nach Übereinstimmungen mit mehreren Gruppen suchen	207
Optionale Übereinstimmung mit dem Fragezeichen	208
Mit dem Sternchen nach null oder mehr Übereinstimmungen suchen ..	209
Mit dem Pluszeichen nach einer oder mehr Übereinstimmungen suchen	209
Mit geschweiften Klammern nach einer genauen Zahl von Wiederholungen suchen	210
Gieriger und nicht gieriger Mustervergleich	211
Die Methode findall()	212
Zeichenklassen	213

Eigene Zeichenklassen bilden	214
Zirkumflex und Dollarzeichen	214
Das Jokerzeichen	215
Beliebige Übereinstimmungen mit Punkt-Stern finden	216
Zeilenumbrüche mit dem Punktsymbol finden	217
Übersicht über Regex-Symbole	217
Übereinstimmungen ohne Berücksichtigung der Groß- und Kleinschreibung	218
Strings mit der Methode sub() ersetzen	218
Umgang mit komplizierten regulären Ausdrücken	219
Die Variablen re.IGNORECASE, re.DOTALL und re.VERBOSE kombinieren	220
Projekt: Extraktionsprogramm für Telefonnummern und E-Mail-Adressen	221
Schritt 1: Einen regulären Ausdruck für Telefonnummern erstellen	222
Schritt 2: Einen regulären Ausdruck für E-Mail-Adressen erstellen	223
Schritt 3: Alle Übereinstimmungen im Inhalt der Zwischenablage finden	224
Schritt 4: Die gefundenen Übereinstimmungen zu einem String kombinieren	225
Das Programm ausführen	225
Ideen für ähnliche Programme	226
Zusammenfassung	226
Wiederholungsfragen	227
Übungsprojekte	229
Datumserkennung	229
Passwortstärke ermitteln	229
Regex-Version von strip()	229
8 Eingabevalidierung	231
Das Modul PyInputPlus	232
Die Schlüsselwortargumente min, max, greaterThan und lessThan	234
Das Schlüsselwortargument blank	235
Die Schlüsselwortargumente limit, timeout und default	235
Die Schlüsselwortargumente allowRegexes und blockRegexes	236
Eine eigene Validierungsfunktion an inputCustom() übergeben	238
Projekt: Einen Trottler stundenlang beschäftigen	239
Projekt: Multiplikationstest	241
Zusammenfassung	243
Wiederholungsfragen	244

Übungsprojekte	244
Sandwichzubereiter	244
Ein eigenes Programm für den Multiplikationstest	245
9 Dateien lesen und schreiben	247
Dateien und Dateipfade	247
Backslash unter Windows und Schrägstrich unter macOS und Linux	248
Pfade mit dem Operator / zusammenfügen	250
Das aktuelle Arbeitsverzeichnis	252
Das Benutzerverzeichnis	253
Absolute und relative Pfade	253
Neue Ordner mit os.makedirs() erstellen	254
Absolute und relative Pfade verwenden	255
Die Komponenten eines Dateipfads abrufen	257
Dateigrößen und Ordnerinhalte ermitteln	260
Eine Dateiliste mit Glob-Mustern bearbeiten	261
Die Gültigkeit von Pfaden prüfen	262
Dateien lesen und schreiben	263
Dateien mit der Funktion open() öffnen	265
Die Inhalte einer Datei lesen	266
Dateien schreiben	267
Variablen mit dem Modul shelve speichern	268
Variablen mit der Funktion pprint.pformat() speichern	270
Projekt: Zufallsgenerator für Tests	271
Schritt 1: Die Daten für den Test in einem Dictionary speichern	272
Schritt 2: Die Fragebogendatei erstellen und die Fragen mischen	273
Schritt 3: Die Auswahl der möglichen Antworten zusammenstellen	274
Schritt 4: Den Inhalt der Dateien für die Frage- und Lösungsbogen schreiben	275
Projekt: Aktualisierbare Mehrfach-Zwischenablage	276
Schritt 1: Kommentare und Vorbereitungen für die Shelf-Daten	277
Schritt 2: Den Inhalt der Zwischenablage unter einem Schlüssel- wort speichern	278
Schritt 3: Schlüsselwörter auflisten und Inhalte laden	279
Zusammenfassung	280
Wiederholungsfragen	280

Übungsprojekte	281
Erweiterte Mehrfach-Zwischenablage	281
Lückentextspiel	281
Regex-Suche	282
10 Dateien verwalten	283
Das Modul shutil	284
Dateien und Ordner kopieren	284
Dateien und Ordner verschieben und umbenennen	285
Dateien und Ordner unwiederbringlich löschen	286
Sicheres Löschen mit dem Modul send2trash	287
Einen Verzeichnisbaum durchlaufen	288
Dateien mit dem Modul zipfile komprimieren	290
ZIP-Dateien lesen	291
ZIP-Dateien entpacken	292
ZIP-Dateien erstellen und Inhalte hinzufügen	293
Projekt: Amerikanische Datumsangaben in Dateinamen in europäische Datumsangaben ändern	293
Projekt: Amerikanische Datumsangaben in Dateinamen in	293
Schritt 1: Einen regulären Ausdruck für amerikanische Datumsan- gaben definieren	294
Schritt 2: Die einzelnen Teile der Datumsangabe in den Dateina- men ermitteln	296
Schritt 3: Die neuen Dateinamen zusammenstellen und die Dateien umbenennen	297
Vorschläge für ähnliche Programme	298
Projekt: Einen Ordner in einer ZIP-Datei sichern	298
Schritt 1: Den Namen der ZIP-Datei bestimmen	298
Schritt 2: Die neue ZIP-Datei erstellen	300
Schritt 3: Den Verzeichnisbaum durchlaufen und Inhalte zur ZIP- Datei hinzufügen	301
Vorschläge für ähnliche Programme	302
Zusammenfassung	302
Wiederholungsfragen	303
Übungsprojekte	303
Selektives Kopieren	303
Nicht mehr benötigte Dateien löschen	303
Lücken entfernen	304

11 Debugging	305
Ausnahmen auslösen	306
Traceback als String abrufen	308
Zusicherungen (Assertions)	309
Zusicherungen in einem Ampelsimulator	311
Protokollierung	312
Das Modul logging verwenden	313
Kein Debugging mit print()	315
Protokolliergrade	315
Die Protokollierung deaktivieren	317
Protokollierung in eine Datei	317
Der Debugger von Mu	318
Continue	318
Step In	319
Step Over	319
Step Out	319
Stop	319
Debugging eines Additionsprogramms	319
Haltepunkte	321
Zusammenfassung	323
Wiederholungsfragen	323
Übungsprojekt	324
Münzwurfprogramm	324
12 Web Scraping	325
Projekt: mapIt.py mit dem Modul webbrowser	326
Schritt 1: Die URL herausfinden	327
Schritt 2: Befehlszeilenargumente verarbeiten	327
Schritt 3: Den Inhalt der Zwischenablage verarbeiten und den Browser starten	328
Vorschläge für ähnliche Programme	329
Dateien mithilfe des Moduls requests aus dem Web herunterladen	329
Eine Webseite mit der Funktion requests.get() herunterladen	330
Auf Fehler prüfen	331
Heruntergeladene Dateien auf der Festplatte speichern	332
HTML	334
Quellen zu HTML	334
Ein kleiner Auffrischkurs	334

Den HTML-Quellcode einer Webseite einsehen	335
Die Entwicklertools des Browsers öffnen	336
HTML-Elemente mithilfe der Entwicklertools finden	338
HTML mit dem Modul bs4 durchsuchen	340
Ein BeautifulSoup-Objekt aus dem HTML-Text erstellen	340
Elemente mit der Methode select() finden	341
Daten aus den Attributen eines Elements abrufen	344
Projekt: Alle Suchergebnisse öffnen	344
Schritt 1: Die Befehlszeilenargumente abrufen und die Suchergeb- nisseite anfordern	345
Schritt 2: Alle Ergebnisse finden	345
Schritt 3: Browsertabs für jedes Suchergebnis öffnen	346
Vorschläge für ähnliche Programme	347
Projekt: Alle XKCD-Comics herunterladen	348
Schritt 1: Den Aufbau des Programms festlegen	349
Schritt 2: Die Webseite herunterladen	350
Schritt 3: Das Bild des Comics finden und herunterladen	351
Schritt 4: Das Bild speichern und den vorherigen Comic suchen	352
Vorschläge für ähnliche Programme	353
Den Browser mit dem Modul selenium steuern	353
Einen seleniumgesteuerten Browser starten	354
Elemente auf der Seite finden	356
Auf Elemente klicken	358
Formulare ausfüllen und absenden	359
Die Betätigung von Sondertasten simulieren	359
Auf Browserschaltflächen klicken	360
Weitere Informationen über Selenium	361
Zusammenfassung	361
Wiederholungsfragen	361
Übungsprojekte	362
E-Mail-Programm für die Befehlszeile	362
Downloadprogramm für Fotowebsites	362
2048	363
Linküberprüfung	363

13 Excel-Arbeitsblätter	365
Excel-Dokumente	366
Das Modul openpyxl installieren	366
Excel-Dokumente lesen	367
Excel-Dokumente mit OpenPyXL öffnen	368
Arbeitsblätter aus der Arbeitsmappe abrufen	368
Zellen eines Arbeitsblatts abrufen	369
Umrechnen zwischen Kennbuchstaben und Nummern	370
Zeilen und Spalten eines Arbeitsblatts abrufen	371
Arbeitsmappen, Arbeitsblätter und Zellen	373
Projekt: Daten in einer Arbeitsmappe lesen	373
Schritt 1: Die Daten der Arbeitsmappe lesen	374
Schritt 2: Die Datenstruktur füllen	375
Schritt 3: Die Ergebnisse in eine Datei schreiben	377
Vorschläge für ähnliche Programme	378
Excel-Dokumente schreiben	379
Excel-Dokumente erstellen und speichern	379
Arbeitsblätter erstellen und entfernen	380
Werte in Zellen schreiben	381
Projekt: Ein Arbeitsblatt aktualisieren	381
Schritt 1: Eine Datenstruktur mit den neuen Informationen einrichten	382
Schritt 2: Alle Zeilen prüfen und die falschen Preise korrigieren	383
Vorschläge für ähnliche Programme	384
Die Schrift in den Zellen gestalten	384
Font-Objekte	385
Formeln	387
Das Erscheinungsbild von Zeilen und Spalten festlegen	388
Zeilenhöhe und Spaltenbreite festlegen	388
Zellen verbinden und aufteilen	389
Bereiche fixieren	390
Diagramme	391
Zusammenfassung	393
Wiederholungsfragen	394
Übungsprojekte	394
Multiplikationstabellen erstellen	395
Leere Zeilen einfügen	395
Zellen transponieren	396
Textdateien in Arbeitsblätter umwandeln	397
Arbeitsblätter in Textdateien umwandeln	397

14 Google Tabellen	399
EZSheets installieren und einrichten	399
Anmeldeinformationen und Tokendateien beziehen	400
Die Datei mit den Anmeldeinformationen widerrufen	402
Spreadsheet-Objekte	403
Tabellen erstellen, hochladen und auflisten	403
Tabellenattribute	405
Tabellen hoch- und herunterladen	406
Tabellen löschen	406
Sheet-Objekte	407
Daten lesen und schreiben	408
Tabellenblätter erstellen und löschen	412
Tabellenblätter kopieren	414
Grenzwerte für Google Tabellen	415
Zusammenfassung	415
Wiederholungsfragen	416
Übungsprojekte	416
Daten von Google Formulare herunterladen	416
Tabellen in andere Formate umwandeln	417
Fehler in einer Tabelle finden	417
15 PDF- und Word-Dokumente	419
PDF-Dokumente	419
Text aus PDFs entnehmen	420
PDFs entschlüsseln	422
PDFs erstellen	423
Projekt: Ausgewählte Seiten aus mehreren PDFs kombinieren	428
Schritt 1: Alle PDF-Dateien finden	429
Schritt 2: Die einzelnen PDFs öffnen	430
Schritt 3: Die einzelnen Seiten hinzufügen	430
Schritt 4: Die Ergebnisse speichern	431
Vorschläge für ähnliche Programme	432
Word-Dokumente	432
Word-Dokumente lesen	433
Den kompletten Text einer .docx-Datei abrufen	434
Absätze und Run-Objekte formatieren	435
Word-Dokumente mit anderen als den Standardformaten erstellen ...	437
Run-Attribute	438

Word-Dokumente schreiben	440
Überschriften hinzufügen	442
Zeilenwechsel und Seitenumbrüche hinzufügen	443
Bilder einfügen	443
PDF-Dateien aus Word-Dokumenten erstellen	444
Zusammenfassung	444
Wiederholungsfragen	445
Übungsprojekte	446
PDF-Paranoia	446
Personalisierte Einladungen als Word-Dokument	446
Brute-Force-Passwortknacker für PDFs	447
16 CSV-Dateien und JSON-Daten	449
Das Modul csv	450
Reader-Objekte	451
Daten in einer for-Schleife aus reader-Objekten lesen	452
Writer-Objekte	453
Die Schlüsselwortargumente delimiter und lineterminator	454
DictReader- und DictWriter-Objekte	455
Projekt: Kopfzeilen aus CSV-Dateien entfernen	457
Schritt 1: Alle CSV-Dateien durchlaufen	458
Schritt 2: Die CSV-Datei lesen	458
Schritt 3: Die CSV-Datei ohne die erste Zeile schreiben	459
Vorschläge für ähnliche Programme	460
JSON und APIs	461
Das Modul json	462
JSON-Daten mit der Funktion loads() laden	462
JSON-Daten mit der Funktion dumps() schreiben	463
Projekt: Die aktuellen Wetterdaten abrufen	463
Schritt 1: Den Standort aus dem Befehlszeilenargument entnehmen ..	464
Schritt 2: Die JSON-Daten herunterladen	465
Schritt 3: JSON-Daten laden und die Wettervorhersage ausgeben ...	466
Vorschläge für ähnliche Programme	467
Zusammenfassung	468
Wiederholungsfragen	468
Übungsprojekt	469
Excel-in-CSV-Konverter	469

17 Zeit und Aufgabenplanung	471
Das Modul time	472
Die Funktion time.time()	472
Die Funktion time.sleep()	473
Zahlen runden	474
Projekt: Superstoppuhr	475
Schritt 1: Das Programm auf die Zeitmessung vorbereiten	475
Schritt 2: Intervalldauern messen und anzeigen	476
Vorschläge für ähnliche Programme	477
Das Modul datetime	478
Der Datentyp timedelta	479
Anhalten bis zu einem bestimmten Zeitpunkt	481
datetime-Objekte in Strings umwandeln	481
Strings in datetime-Objekte umwandeln	483
Die Zeitfunktionen von Python im Überblick	483
Multithreading	484
Argumente an die Zielfunktion eines Threads übergeben	486
Probleme der Nebenläufigkeit	487
Projekt: Multithread-Version des XKCD-Downloadprogramms	488
Schritt 1: Eine Funktion für den Download verwenden	488
Schritt 2: Threads erstellen und starten	490
Schritt 3: Auf das Ende aller Threads warten	490
Andere Programme von Python aus starten	491
Befehlszeilenargumente an Popen() übergeben	494
Taskplaner, launchd und cron	494
Websites mit Python aufrufen	495
Andere Python-Skripte ausführen	495
Dateien in ihren Standardanwendungen öffnen	495
Projekt: Ein einfaches Countdown-Programm	496
Schritt 1: Der Countdown	497
Schritt 2: Die Klangdatei abspielen	497
Vorschläge für ähnliche Programme	498
Zusammenfassung	499
Wiederholungsfragen	500
Übungsprojekte	501
Elegantere Stoppuhr	501
Webcomic-Downloadprogramm mit Zeitplanung	501

18 E-Mails und Textnachrichten	503
E-Mails mit der Gmail-API senden und empfangen	504
Die Gmail-API aktivieren	505
E-Mails von einem Gmail-Konto senden	505
E-Mails in einem Gmail-Konto lesen	506
E-Mails in einem Gmail-Konto suchen	508
Anhänge von einem Gmail-Konto herunterladen	508
SMTP	509
E-Mails senden	510
Verbindung mit einem SMTP-Server aufnehmen	510
Die »Hallo«-Nachricht an den SMTP-Server senden	512
Die TLS-Verschlüsselung einleiten	512
Am SMTP-Server anmelden	513
Eine E-Mail senden	513
Die Verbindung zum SMTP-Server trennen	514
IMAP	514
E-Mails mit IMAP abrufen und löschen	515
Verbindung mit einem IMAP-Server aufnehmen	515
Am IMAP-Server anmelden	516
Nach E-Mails suchen	517
E-Mails abrufen und als gelesen markieren	521
E-Mail-Adressen aus einer Rohnachricht gewinnen	522
Den Rumpf aus einer Rohnachricht gewinnen	523
E-Mails löschen	524
Die Verbindung zum IMAP-Server trennen	524
Projekt: E-Mails über ausstehende Mitgliedsbeiträge senden	525
Schritt 1: Die Excel-Datei öffnen	526
Schritt 2: Alle säumigen Mitglieder finden	527
Schritt 3: Personalisierte E-Mail-Mahnungen senden	528
Textnachrichten über SMS-E-Mail-Gateways senden	529
Textnachrichten mit Twilio senden	531
Ein Twilio-Konto einrichten	532
Textnachrichten senden	532
Projekt: Das Modul »Just Text Me«	534
Zusammenfassung	535
Wiederholungsfragen	536

Übungsprojekte	537
Zufällige Zuweisung von Arbeiten	537
Regenschirmhinweis	537
Automatischer Entregistrierer	537
Den Computer per E-Mail steuern	538
19 Bildbearbeitung	541
Grundlagen zur Bilddarstellung auf Computern	542
Farben und RGBA-Werte	542
Koordinaten und Rechtecktuplel	543
Bildbearbeitung mit Pillow	545
Mit dem Datentyp Image arbeiten	546
Bilder beschneiden	548
Bilder kopieren und in andere Bilder einfügen	549
Die Bildgröße ändern	552
Bilder drehen und spiegeln	553
Einzelne Pixel ändern	555
Projekt: Ein Logo hinzufügen	556
Schritt 1: Das Logobild öffnen	557
Schritt 2: Alle Dateien durchlaufen und die Bilder öffnen	558
Schritt 3: Die Bildgröße ändern	559
Schritt 4: Logo hinzufügen und Änderungen speichern	560
Vorschläge für ähnliche Programme	562
Bilder zeichnen	563
Formen zeichnen	563
Text zeichnen	565
Zusammenfassung	567
Wiederholungsfragen	568
Übungsprojekte	568
Das Logoprogramm erweitern und verbessern	569
Fotoordner auf der Festplatte finden	569
Personalisierte Tischkarten	570
20 GUI-Automatisierung	573
Das Modul PyAutoGUI installieren	574
Zugriff auf macOS erlauben	575

Kleine Probleme beheben	575
Die PyAutoGUI-Notfallsicherung	575
Beenden durch Abmelden	576
Den Mauszeiger steuern	576
Den Mauszeiger bewegen	577
Die Position des Mauszeigers abrufen	578
Mausinteraktionen	579
Klicken	579
Ziehen	579
Scrollen	582
Mausbewegungen planen	582
Auf dem Bildschirm arbeiten	584
Einen Screenshot aufnehmen	584
Einen Screenshot analysieren	584
Bildererkennung	586
Informationen über das Fenster abrufen	588
Das aktive Fenster abrufen	588
Andere Möglichkeiten zum Abrufen von Fenstern	589
Fenster bearbeiten	590
Die Tastatur steuern	592
Strings von der Tastatur senden	592
Tastennamen	594
Tasten drücken und loslassen	595
Tastenkombinationen	595
GUI-Automatisierungsskripte einrichten	596
Übersicht über die Funktionen von PyAutoGUI	597
Projekt: Formulare automatisch ausfüllen	599
Schritt 1: Den Ablauf herausfinden	601
Schritt 2: Die Koordinaten ermitteln	601
Schritt 3: Daten eingeben	603
Schritt 4: Auswahllisten und Optionsschalter	604
Schritt 5: Das Formular absenden und warten	605
Dialogfelder anzeigen	606
Zusammenfassung	608
Wiederholungsfragen	608

Übungsprojekte	609
Beschäftigung vortäuschen	609
Textfelder über Zwischenablage lesen	609
Instant-Messenger-Bot	610
Tutorial für einen Spiele-Bot	611

Anhang **613**

A Drittanbietermodule installieren **613**

Pip	613
Drittanbietermodule installieren	614
Module für Mu installieren	616

B Programme ausführen **619**

Programme im Terminalfenster ausführen	619
Python-Programme unter Windows ausführen	621
Python-Programme unter macOS ausführen	622
Programme unter Ubuntu Linux ausführen	623
Python-Programme mit ausgeschalteten Zusicherungen ausführen	624

C Antworten auf die Wiederholungsfragen **625**

Kapitel 1	626
Kapitel 2	626
Kapitel 3	628
Kapitel 4	629
Kapitel 5	630
Kapitel 6	630
Kapitel 7	631
Kapitel 8	632
Kapitel 9	632
Kapitel 10	633
Kapitel 11	633
Kapitel 12	634
Kapitel 13	635

Kapitel 14	636
Kapitel 15	636
Kapitel 16	637
Kapitel 17	637
Kapitel 18	638
Kapitel 19	638
Kapitel 20	639
Stichwortverzeichnis	641

Für meinen Neffen Jack

Der Autor

Al Sweigart ist Softwareentwickler und Fachbuchautor. Seine Lieblingsprogrammiersprache ist Python und er hat bereits mehrere Open-Source-Module dafür entwickelt. Seine anderen Bücher sind auf seiner Website <http://www.inventwithpython.com/> unter einer Creative-Commons-Lizenz kostenlos erhältlich. Seine Katze wiegt 11 Pfund.

Der Fachgutachter

Philip James arbeitet schon mehr als zehn Jahre mit Python und tritt in der Python-Community häufig als Redner zu unterschiedlichen Themen von Unix-Grundlagen bis zu Social Media auf Open-Source-Basis auf. Er ist einer der Hauptbeteiligten am BeeWare-Projekt und lebt mit seiner Partnerin Nic und deren Katze River in der Bay Area von San Francisco.

Danksagung

Es ist irreführend, dass nur mein Name auf dem Umschlag steht, denn um dieses Buch schreiben zu können, habe ich die Hilfe vieler Menschen benötigt. Ich möchte meinem Herausgeber Bill Pollock, meinen Lektoren Laurel Chun, Leslie Shen, Greg Poulos, Jennifer Griffith-Delgado und Frances Saux sowie allen anderen Mitarbeitern bei No Starch Press für ihre unschätzbare Hilfe danken. Ein Dank geht auch an meine Fachgutachter Ari Lacenski und Philip James für ihre hervorragenden Vorschläge und Verbesserungen sowie für ihre Unterstützung.

Vielen Dank auch allen Mitgliedern der Python Software Foundation für ihre großartige Arbeit. Die Python-Community ist die beste, die ich in der IT-Branche gefunden habe.

Schließlich möchte ich noch meiner Familie, meinen Freunden und der Clique bei Shotwell's danken, weil sie es mir nicht übel genommen haben, dass ich während des Schreibens an diesem Buch dauernd beschäftigt war. Dankeschön!

Einleitung



»Du hast gerade in zwei Stunden das erledigt, woran wir drei sonst zwei Tage lang sitzen!«

Mein Mitbewohner in den frühen 2000er-Jahren arbeitete bei einem Elektronikhändler, bei dem gelegentlich eine Tabelle mit Tausenden von Produktpreisen anderer Läden auftauchte. Drei Mitarbeiter druckten diese Tabelle dann auf einem dicken Stapel Papier aus und teilten diesen unter sich auf. Für jeden Produktpreis schlugen sie den Preis ihres eigenen Arbeitgebers nach und notierten alle Produkte, die die Konkurrenz billiger anbot. Damit waren sie gewöhnlich zwei Tage beschäftigt.

»Wenn ich die Originaldatei bekomme, kann ich ein Programm schreiben, das die Arbeit für euch erledigt«, schlug mein Mitbewohner ihnen vor, als er sah, wie sie inmitten eines Riesenhaufens Papier auf dem Fußboden hockten.

Nach ein paar Stunden hatte er ein kurzes Programm geschrieben, das die Preisliste der Konkurrenten aus der Datei auslas, die Produkte in der Datenbank des Elektronikladens nachschlug und einen Vermerk machte, wenn die Konkurrenz billiger war. Er war immer noch ein Anfänger in Sachen Programmierung

und hatte den Großteil dieser Stunden damit zugebracht, die Dokumentation in einem Programmierbuch nachzuschlagen. Die Ausführung des fertigen Programms dauerte nur wenige Sekunden. An dem Tag gönnten sich mein Mitbewohner und seine Kollegen eine besonders lange Mittagspause.

Das zeigt das Potenzial der Programmierung. Ein Computer ist wie ein Schweizer Messer und lässt sich für zahllose Aufgaben einrichten. Viele Leute bringen Stunden mit Klicken und Tippen zu, um monotone Aufgaben auszuführen, ohne zu ahnen, dass der Computer diese Arbeit in wenigen Sekunden erledigen könnte, wenn er nur die richtigen Anweisungen dafür bekäme.

Für wen ist dieses Buch gedacht?

Software bildet die Grundlage vieler unserer Geräte, die wir für die Arbeit und in der Freizeit verwenden: Fast jeder nutzt soziale Netzwerke zur Kommunikation, die Telefone vieler Menschen enthalten Computer mit Internetzugriff und für die meisten Büroarbeiten ist Computerarbeit erforderlich. Daher ist die Nachfrage nach Personen, die programmieren können, sprunghaft angestiegen. Unzählige Bücher, interaktive Webtutorials und Schulungen für Entwickler werden mit dem Versprechen beworben, ehrgeizige Anfänger zu Softwareingenieuren zu machen, die sechsstellige Gehälter verlangen können.

Dieses Buch ist jedoch nicht für diese Leute gedacht, sondern für alle anderen.

Mit diesem Buch allein können Sie nicht zu einem professionellen Softwareentwickler werden, genauso wenig, wie ein paar Gitarrenstunden Sie zu einem Rockstar machen. Wenn Sie aber Büroangestellter, Administrator oder Akademiker sind oder auch nur zur Arbeit oder zum Vergnügen einen Computer benutzen, so werden Sie hier die Grundlagen der Programmierung kennenlernen, um einfache Aufgaben wie die folgenden zu automatisieren:

- Tausende von Dateien verschieben und umbenennen und in Ordner sortieren
- Onlineformulare ausfüllen, ohne Text eingeben zu müssen
- Dateien von einer Website herunterladen oder Texte von dort kopieren, sobald dort neues Material bereitgestellt wird
- Sich von Ihrem Computer per SMS benachrichtigen lassen
- Excel-Arbeitsblätter bearbeiten und formatieren
- Nach neuen E-Mails suchen und vorformulierte Antworten senden

Diese Aufgaben sind einfach, aber zeitraubend, und sie sind häufig so trivial oder so spezifisch, dass es keine fertige Software dafür gibt. Mit einigen Programmierkenntnissen können Sie Ihren Computer diese Aufgaben für Sie erledigen lassen.