

Auf einen Blick

Über den Autor	11
Einleitung	23
Teil I: Erste Schritte mit Python	27
Kapitel 1: Die Kommunikation mit Ihrem Computer	29
Kapitel 2: Installieren Sie sich Ihr eigenes Python	43
Kapitel 3: Mit Python arbeiten	63
Kapitel 4: Ihre erste Anwendung	83
Kapitel 5: Mit Anaconda arbeiten	111
Teil II: Elementare Python-Bausteine	129
Kapitel 6: Daten speichern und ändern	131
Kapitel 7: Daten verwalten	141
Kapitel 8: Entscheidungen treffen	161
Kapitel 9: Sich wiederholende Aufgaben ausführen	177
Kapitel 10: Fehlerbehandlung in Programmen	191
Teil III: Alltägliche Aufgabenstellungen	221
Kapitel 11: Mit Paketen arbeiten	223
Kapitel 12: Mit Zeichenketten arbeiten	251
Kapitel 13: Listen verwalten	267
Kapitel 14: Verschiedenartige Daten sammeln	285
Kapitel 15: Klassen erstellen und verwenden	305
Teil IV: Anspruchsvollere Aufgaben	327
Kapitel 16: Daten in Dateien speichern	329
Kapitel 17: E-Mails versenden	347
Teil V: Der Top-Ten-Teil	363
Kapitel 18: Zehn tolle Quellen für Programmierer	365
Kapitel 19: Zehn Möglichkeiten, mit Python Geld zu verdienen	375
Kapitel 20: Zehn Tools zur Erweiterung von Python	383
Kapitel 21: Gut zehn Bibliotheken, die Sie kennen sollten	393
Stichwortverzeichnis	401

Inhaltsverzeichnis

Über den Autor	11
Einleitung	23
Über dieses Buch	23
Törichte Annahmen über den Leser	24
Symbole in diesem Buch	25
Über das Buch hinaus	25
Wie geht es weiter?	25
TEIL I	
ERSTE SCHRITTE MIT PYTHON	27
Kapitel 1	
Die Kommunikation mit Ihrem Computer	29
Warum wollen Sie sich mit Ihrem Computer unterhalten?	30
Anwendungen als eine Art von Kommunikation	31
Alltägliche Arbeitsabläufe	31
Arbeitsabläufe aufschreiben	32
Anwendungen als gewöhnliche Arbeitsabläufe verstehen	33
Computer nehmen alles wörtlich	33
Was sind Anwendungen eigentlich?	33
Computer besitzen ihre eigene Sprache	34
Den Menschen helfen, sich mit dem Computer zu verständigen	34
Warum Python so cool ist	36
Warum man Python verwenden sollte	36
Wie Sie persönlich von Python profitieren können	37
Welche Organisationen verwenden Python?	39
Nützliche Python-Programme finden	39
Python mit anderen Sprachen vergleichen	40
Kapitel 2	
Installieren Sie sich Ihr eigenes Python	43
Die benötigte Version herunterladen	43
Python installieren	46
Mit Windows arbeiten	47
Mit dem Mac arbeiten	49
Mit Linux arbeiten	51
Auf Python auf Ihrem System zugreifen	55
Mit Windows arbeiten	56
Mit dem Mac	58
Mit Linux arbeiten	59
Ihre Installation testen	60

Kapitel 3	
Mit Python arbeiten	63
Python über die Kommandozeile starten	63
Python starten	64
Die Kommandozeile nutzbringend einsetzen	65
Python-Umgebungsvariablen sinnvoll einsetzen	68
Einzelne Befehle eingeben	69
Dem Computer sagen, was er machen soll	70
Dem Computer mitteilen, dass Sie fertig sind	70
Ergebnisse anzeigen	70
Die Hilfe verwenden	71
In den Hilfemodus wechseln	73
Um Hilfe bitten	74
Den Hilfemodus verlassen	77
Sich unmittelbar Hilfe holen	77
Die Python-Kommandozeile schließen	79
Kapitel 4	
Ihre erste Anwendung	83
Warum IDEs derart wichtig sind	84
Besseren Code erstellen	84
Debugging-Funktionalität	84
Warum Notebooks nützlich sind	85
So erhalten Sie Ihre Anaconda-Kopie	86
Analytics Anaconda herunterladen	86
Anaconda unter Linux installieren	87
Anaconda unter MacOS installieren	88
Anaconda unter Windows installieren	89
Datensätze und Beispielcode herunterladen	93
Jupyter Notebook nutzen	93
Das Code-Repository definieren	96
Eine Anwendung erstellen	101
Zellen verstehen	101
Dokumentationszellen hinzufügen	102
Weitere Zellinhalte	103
Einrückungen verstehen	104
Kommentare einfügen	106
Kommentare verstehen	106
Kommentare als Erinnerungsstützen verwenden	107
Mit Kommentaren die Ausführung von Code verhindern	108
Jupyter Notebook schließen	108
Kapitel 5	
Mit Anaconda arbeiten	111
Ihren Code herunterladen	112
Mit Checkpoints arbeiten	113
Checkpoints nutzen	113

Einen Checkpoint speichern	114
Einen Checkpoint wiederherstellen	115
Zellen bearbeiten	115
Verschiedene Zelltypen hinzufügen	115
Zellen teilen und verbinden	116
Zellen verschieben	116
Zellen ausführen	117
Ausgabe umschalten	117
Das Aussehen von Jupyter Notebook ändern	118
Befehle über die Befehlspalette suchen	120
Mit Zeilennummern arbeiten	120
Die Funktionen der Zell-Werkzeugleiste nutzen	121
Mit dem Kernel zusammenarbeiten	122
Hilfe anfordern	123
Magische Funktionen nutzen	125
Laufende Prozesse anzeigen	128

TEIL II
ELEMENTARE PYTHON-BAUSTEINE 129

Kapitel 6
Daten speichern und ändern 131

Daten speichern	131
Variablen als Aufbewahrungskisten betrachten	132
Die richtige Kiste zum Speichern der Daten	132
Grundlegende Datentypen von Python	133
Daten in Variablen ablegen	133
Numerische Datentypen	133
Boolesche Werte	137
Der Datentyp String	138
Datum und Uhrzeit	139

Kapitel 7
Daten verwalten 141

Steuern, wie Python Daten interpretiert	142
Vergleiche durchführen	142
Wie Computer Vergleiche durchführen	143
Mit Operatoren arbeiten	143
Operatoren definieren	144
Vorrangregeln für Operatoren	149
Erstellen und Verwenden von Funktionen	150
Funktionen als Codepäckchen betrachten	150
Die Wiederverwendbarkeit von Code ist wichtig	151
Funktionen definieren	152
Funktionen ausführen	153
Daten an Funktionen übergeben	154

Datenrückgabe von Funktionen	158
Rückgabewerte vergleichen	159
Benutzereingaben	159

**Kapitel 8
Entscheidungen treffen 161**

Einfache Entscheidungen mit der »if«-Anweisung treffen	162
Die »if«-Anweisung	162
Die »if«-Anweisung in Anwendungen verwenden	162
Mit der »if...else«-Anweisung Optionen auswählen	167
Die »if...else«-Anweisung	167
Die »if...else«-Anweisung in einer Anwendung verwenden	167
Die »if...elif«-Anweisung in einer Anwendung verwenden	169
Verschachtelte Entscheidungen nutzen	172
Mehrere »if«- oder »if...else«-Anweisungen verwenden	172
Mehrere Entscheidungen miteinander kombinieren	173

**Kapitel 9
Sich wiederholende Aufgaben ausführen 177**

Daten mit der »for«-Schleife verarbeiten	178
Die »for«-Schleife	178
Eine einfache »for«-Schleife erstellen	178
Die Programmausführung mit der »break«-Anweisung steuern	179
Die Ausführung mit »continue« steuern	181
Die Ausführung mit »pass« steuern	183
Die Ausführung mit der »else«-Klausel steuern	184
Daten mit der »while«-Anweisung verarbeiten	185
Die »while«-Anweisung	185
Die »while«-Anweisung in einer Anwendung verwenden	186
Schleifen verschachteln	187

**Kapitel 10
Fehlerbehandlung in Programmen 191**

Warum versteht Python mich nicht?	192
Fehlerquellen erkennen	193
Klassifizieren, wann Fehler auftreten	194
Fehlertypen unterscheiden	195
Ausnahmen abfangen	197
Die grundlegende Ausnahmebehandlung	197
Spezifischere Behandlung weniger spezifischer Ausnahmefehler	208
Verschachtelte Ausnahmebehandlung	211
Ausnahmefehler auslösen	214
Ausnahmen unter außergewöhnlichen Umständen auslösen	214
Fehlerinformationen an den Aufrufer übergeben	215
Benutzerdefinierte Ausnahmen erstellen und verwenden	216
Die »finally«-Klausel	218

TEIL III ALLTÄGLICHE AUFGABENSTELLUNGEN 221

Kapitel 11 Mit Paketen arbeiten 223

Code gruppieren	224
Die Pakettypen verstehen	226
Den Cache für Pakete betrachten	227
Pakete importieren	228
Die »import«-Anweisung verwenden	230
Die »from...import«-Anweisung verwenden	231
Pakete auf Datenträgern finden	233
Pakete aus anderen Quellen herunterladen	235
Die Anaconda-Eingabeaufforderung starten	236
Mit conda-Paketen arbeiten	237
Pakete mit pip installieren	243
Sich den Paketinhalt ansehen	243
Die Paketdokumentation verwenden	246
Die pydoc-Anwendung öffnen	246
Die Links für den Schnellzugriff verwenden	247
Einen Suchbegriff eingeben	248
Die Ergebnisse anschauen	250

Kapitel 12 Mit Zeichenketten arbeiten 251

Zeichenketten sind anders	252
Einzelne Zeichen durch Zahlen definieren	252
Zeichen zur Erstellung von Zeichenketten verwenden	253
Zeichenketten mit Sonderzeichen erstellen	254
Einzelne Zeichen auswählen	256
Zeichenketten schneiden und würfeln	258
Einen Wert in einem String finden	261
Zeichenketten formatieren	263

Kapitel 13 Listen verwalten 267

Daten in einer Anwendung strukturieren	268
Strukturen mit Listen nachbilden	268
Wie Computer Listen sehen	269
Listen erstellen	270
Auf Listen zugreifen	272
Listen durchlaufen	273
Listen bearbeiten	274
Listen durchsuchen	277
Listen sortieren	279
Listen drucken	280
Mit dem Counter-Objekt arbeiten	282

Kapitel 14

Verschiedenartige Daten sammeln	285
So funktionieren Collections	285
Mit Tupeln arbeiten	287
Mit Dictionaries arbeiten	290
Erstellen und Verwenden von Dictionarys	291
Die »switch«-Anweisung durch ein Dictionary ersetzen	294
Stapel mithilfe von Listen erstellen	296
Mit Warteschlangen arbeiten	299
Mit Deques arbeiten	301

Kapitel 15

Klassen erstellen und verwenden	305
Klassen als Strukturierungshilfe	306
Komponenten einer Klasse	307
Eine Klassendefinition schreiben	308
Die integrierten Klassenattribute	309
Mit Methoden arbeiten	310
Mit Konstruktoren arbeiten	312
Mit Variablen arbeiten	314
Methoden mit variablen Parameterlisten verwenden	316
Operatoren überladen	318
Eine Klasse erstellen	320
MeineKlasse definieren	320
Eine Klasse auf Datenträger speichern	321
Die Klasse in einer Anwendung verwenden	321
Eine Klasse erweitern, um neue Klassen zu schaffen	323
Eine Kindklasse schreiben	323
Die Klasse in einer Anwendung testen	324

TEIL IV

ANSPRUCHSVOLLERE AUFGABEN	327
--	------------

Kapitel 16

Daten in Dateien speichern	329
Wie die permanente Datenspeicherung funktioniert	330
Daten für die permanente Speicherung erstellen	332
Eine Datei erstellen	335
Dateiinhalte lesen	339
Dateiinhalte aktualisieren	341
Eine Datei löschen	345

Kapitel 17

E-Mails versenden	347
Das passiert, wenn man E-Mails versendet	349
E-Mails als herkömmliche Briefe betrachten	349
Die Bestandteile des Umschlags	350
Die Bestandteile einer Nachricht	355

Eine E-Mail-Nachricht erstellen	359
Eine Textnachricht versenden	359
Eine HTML-Nachricht versenden	360
Sich die E-Mail-Ausgabe anschauen	362

TEIL V

DER TOP-TEN-TEIL..... 363

Kapitel 18

Zehn tolle Quellen für Programmierer..... 365

Die Python-Online-Dokumentation nutzen	365
Das Tutorial LearnPython.org nutzen.....	366
Webanwendungen mit Python programmieren.....	367
Zusätzliche Bibliotheken beschaffen	368
Schneller Anwendungen mit einer IDE erstellen.....	369
Einfachere Syntaxprüfung	370
XML sinnvoll nutzen	371
Die üblichen Anfängerfehler in Python vermeiden	372
Unicode verstehen	373
Machen Sie Ihrer Python-Anwendung Beine.....	374

Kapitel 19

Zehn Möglichkeiten, mit Python Geld zu verdienen..... 375

In der Qualitätssicherung arbeiten	376
IT-Mitarbeiter in einer kleinen Firma werden	377
Spezialskripte für Anwendungen schreiben	377
Netzwerkadministration.....	378
Programmierkenntnisse vermitteln	378
Leuten dabei helfen, einen bestimmten Ort zu finden	379
Data Mining nutzen.....	379
Mit eingebetteten Systemen arbeiten	380
Wissenschaftliche Aufgaben erledigen.....	380
Echtzeitanalyse von Daten.....	381

Kapitel 20

Zehn Tools zur Erweiterung von Python..... 383

Programmfehler mit dem Roundup Issue Tracker dokumentieren	384
Eine virtuelle Umgebung mit VirtualEnv erstellen	385
Ihre Anwendung mit PyInstaller installieren	386
Eine Entwicklerdokumentation mit pdoc erzeugen	387
Anwendungscode mit Komodo Edit entwickeln	388
Ihre Anwendung mit pydbgr debuggen.....	389
Interaktive Umgebungen mit IPython betreten	390
Anwendungen mit PyUnit testen.....	390
Ihren Code mit Isort aufräumen	391
Versionskontrolle mit Mercurial	391

Kapitel 21

Gut zehn Bibliotheken, die Sie kennen sollten 393

- Eine sichere Umgebung mit PyCrypto entwickeln 394
- Auf Datenbanken mit SQLAlchemy zugreifen 394
- Die Welt mit Google Maps bereisen 395
- Benutzeroberflächen mit TkInter erstellen 395
- Tabellarische Daten mit PrettyTable ansprechend präsentieren..... 396
- Anwendungen mit PyAudio und Soundausgabe aufwerten 396
- Abbildungen mit PyQtGraph bearbeiten 397
- Daten mit IRLib finden 398
- Interoperable Java-Umgebungen mit JPype erstellen 399
- Auf lokale Netzwerkressourcen mit Twisted Matrix zugreifen 399
- Auf Internet-Ressourcen mit Bibliotheken zugreifen..... 400

Stichwortverzeichnis 401