

Inhalt

Vorwort	XVII
Danksagung	XVIII
1 Einleitung	1
1.1 Einfach und schnell zu lernen	1
1.2 Zielgruppe des Buches	1
1.3 Aufbau des Buches	2
1.4 Programmieren lernen „interaktiv“	3
1.5 Download der Beispiele und Hilfe	4
1.6 Anregungen und Kritik	4
Teil I Einführung in Python 3 – Für Ein- und Umsteiger	5
2 Kommandos und Programme	7
2.1 Erste Schritte mit Python	7
2.1.1 Linux	7
2.1.2 Windows	8
2.2 Herkunft und Bedeutung des Begriffes interaktive Shell	9
2.2.1 Erste Schritte in der interaktiven Shell	9
2.3 Verlassen der Python-Shell	11
2.4 Benutzung von Variablen	11
2.5 Mehrzeilige Anweisungen in der interaktiven Shell	12
2.6 Programme schreiben oder schnell mal der Welt „Hallo“ sagen	12
3 Bytecode und Maschinencode	15
3.1 Einführung	15
3.2 Unterschied zwischen Programmier- und Skriptsprachen	15
3.3 Interpreter- oder Compilersprache	15

4	Datentypen und Variablen	19
4.1	Einführung	19
4.2	Datentypen	21
4.2.1	Ganze Zahlen	21
4.2.2	Fließkommazahlen	22
4.2.3	Zeichenketten	23
4.2.4	Boolesche Werte	23
4.2.5	Komplexe Zahlen	23
4.2.6	Operatoren	24
4.3	Statische und dynamische Typdeklaration	25
4.4	Typumwandlung	27
4.5	Datentyp ermitteln	27
5	Sequentielle Datentypen	29
5.1	Übersicht	29
5.1.1	Zeichenketten oder Strings	29
5.1.2	Listen	31
5.1.3	Tupel	32
5.1.4	Sequenz von Binärdaten	32
5.2	Indizierung von sequentiellen Datentypen	32
5.3	Slicing oder Ausschneiden	33
5.4	Die len-Funktion	35
5.5	Aufgaben	36
6	Dictionaries	39
6.1	Dictionaries und assoziative Felder	39
6.2	Definition und Benutzung	40
6.3	Fehlerfreie Zugriffe auf Dictionaries	42
6.4	Zulässige Typen für Schlüssel und Werte	43
6.5	Verschachtelte Dictionaries	44
6.6	Methoden auf Dictionaries	44
6.7	Operatoren	47
6.8	Die zip-Funktion	48
6.9	Dictionaries aus Listen erzeugen	49
6.10	Aufgaben	50

7	Mengen	53
7.1	Übersicht	53
7.2	Mengen in Python	53
7.2.1	Sets erzeugen	54
7.2.2	Mengen von unveränderlichen Elementen	54
7.3	Frozensets	55
7.4	Operationen auf „set“-Objekten	55
7.4.1	add(element)	55
7.4.2	clear()	55
7.4.3	copy	56
7.4.4	difference()	56
7.4.5	difference_update()	57
7.4.6	discard(el)	57
7.4.7	remove(el)	57
7.4.8	intersection(s)	58
7.4.9	isdisjoint()	58
7.4.10	issubset()	58
7.4.11	issuperset()	59
7.4.12	pop()	59
8	Eingaben	61
8.1	Eingabe mittels input	61
9	Verzweigungen	63
9.1	Anweisungsblöcke und Einrückungen	63
9.2	Bedingte Anweisungen in Python	66
9.3	Vergleichsoperatoren	67
9.4	Zusammengesetzte Bedingungen	68
9.5	Wahr oder falsch: Bedingungen in Verzweigungen	68
9.6	Aufgaben	69
10	Schleifen	71
10.1	Übersicht	71
10.2	while-Schleife	72
10.3	break und continue	73
10.4	die Alternative im Erfolgsfall: else	74
10.5	For-Schleife	75
10.6	Aufgaben	78

11 Dateien lesen und schreiben	81
11.1 Dateien	81
11.2 Text aus einer Datei lesen	81
11.3 Schreiben in eine Datei	83
11.4 In einem Rutsch lesen: readlines und read	83
11.5 Aufgaben	84
12 Formatierte Ausgabe und Strings formatieren	87
12.1 Wege, die Ausgabe zu formatieren	87
12.2 Details zur print-Funktion	88
12.2.1 Unterschiede zu Python 2	88
12.2.2 Import aus der Zukunft: print_function	88
12.2.3 print-Funktion in Python 3	89
12.3 Stringformatierung im C-Stil	90
12.4 Der pythonische Weg: Die String-Methode „format“	95
12.5 Benutzung von Dictionaries beim Aufruf der „format“-Methode	98
12.6 Benutzung von lokalen Variablen in „format“	99
12.7 Weitere String-Methoden zum Formatieren	100
13 Flaches und tiefes Kopieren	103
13.1 Einführung	103
13.2 Kopieren einer Liste	104
13.3 Kopie mit Teilbereichsoperator	106
13.4 Kopieren mit deepcopy	107
13.5 Deepcopy für Dictionaries	107
14 Funktionen	109
14.1 Allgemein	109
14.2 Funktionen in Python	109
14.3 Optionale- und Schlüsselwortparameter	111
14.4 Docstring	112
14.5 Rückgabewerte	113
14.6 Mehrere Rückgabewerte	114
14.7 Lokale und globale Variablen in Funktionen	114
14.8 Parameterübergabe im Detail	116
14.9 Effekte bei veränderlichen Objekten	118
14.10 Kommandozeilenparameter	119
14.11 Variable Anzahl von Parametern / Variadische Funktionen	120
14.12 * in Funktionsaufrufen	121

14.13 Beliebige Schlüsselwortparameter	122
14.14 Doppeltes Sternchen im Funktionsaufruf.....	122
14.15 Aufgaben	123
15 Rekursive Funktionen.....	125
15.1 Definition und Herkunft des Begriffs	125
15.2 Definition der Rekursion	126
15.3 Rekursive Funktionen in Python.....	126
15.4 Die Tücken der Rekursion	127
15.5 Fibonacci-Folge in Python.....	128
15.6 Aufgaben	132
16 Listen und Tupel im Detail.....	135
16.1 Stapelspeicher.....	135
16.2 Stapelverarbeitung in Python: pop und append.....	136
16.3 extend	136
16.4 ,+-Operator oder append	137
16.5 Entfernen eines Wertes.....	138
16.6 Prüfen, ob ein Element in Liste enthalten ist	139
16.7 Finden der Position eines Elementes	139
16.8 Einfügen von Elementen.....	140
16.9 Besonderheiten bei Tupel	140
16.9.1 Leere Tupel	141
16.9.2 1-Tupel.....	141
16.9.3 Mehrfachzuweisungen, Packing und Unpacking	141
16.10 Die veränderliche Unveränderliche	142
16.11 Sortieren von Listen	143
16.11.1 „sort“ und „sorted“.....	143
16.11.2 Umkehrung der Sortierreihenfolge	144
16.11.3 Eigene Sortierfunktionen	144
16.12 Aufgaben	147
17 Modularisierung.....	149
17.1 Module	149
17.1.1 Namensräume von Modulen	150
17.1.2 Namensräume umbenennen	151
17.1.3 Modularten.....	151
17.1.4 Suchpfad für Module	152
17.1.5 Inhalt eines Modules.....	153

17.1.6	Eigene Module	153
17.1.7	Dokumentation für eigene Module	154
17.2	Pakete.....	155
18	Globale und lokale Variablen	157
18.1	Einführung.....	157
18.2	Globale und lokale Variablen in Funktionen.....	157
19	Alles über Strings	161
19.1	... fast alles	161
19.2	Aufspalten von Zeichenketten	162
19.2.1	split.....	162
19.2.2	Standardverhalten und „maxsplit“	164
19.2.3	rsplit.....	165
19.2.4	Folge von Trennzeichen	167
19.2.5	splitlines	168
19.2.6	partition	168
19.3	Zusammenfügen von Stringlisten mit join	169
19.4	Suchen von Teilstings	169
19.4.1	„in“ oder „not in“	169
19.4.2	s.find(substring[, start[, end]]).....	169
19.4.3	s.rfind(substring[, start[, end]]).....	170
19.4.4	s.index(substring[, start[, end]])	170
19.4.5	s.rindex(substring[, start[, end]])	171
19.4.6	s.count(substring[, start[, end]])	171
19.5	Suchen und Ersetzen	171
19.6	Nur noch Kleinbuchstaben oder Großbuchstaben	172
19.7	capitalize und title	172
19.8	Stripping Strings	173
19.9	Strings ausrichten.....	173
19.10	String-Tests	174
19.11	Aufgaben	176
20	Ausnahmebehandlung.....	179
20.1	Abfangen mehrerer Exceptions	181
20.2	except mit mehrfachen Ausnahmen	181
20.3	Die optionale else-Klausel	182
20.4	Fehlerinformationen über sys.exc_info	182
20.5	Exceptions generieren.....	183
20.6	Finalisierungsaktion	184

21 Objektorientierte Programmierung	185
21.1 Einführung	185
21.2 Klassen in Python	187
21.2.1 Objekte und Instanzen einer Klasse	187
21.2.2 Kapselung von Daten und Methoden	188
21.2.3 Ein minimale Klasse in Python	188
21.2.4 Eigenschaften und Attribute	189
21.3 Methoden	191
21.3.1 Instanzvariablen	191
21.3.2 Die <code>__init__</code> -Methode	193
21.4 Datenkapselung, Datenabstraktion und Geheimnisprinzip	195
21.4.1 Begriffsbestimmungen	195
21.4.2 Die <code>__str__</code> - und die <code>__repr__</code> -Methode	197
21.5 Public-, Protected- und Private-Attribute	201
21.6 Destruktor	205
21.7 Klassenattribute	207
21.8 Statische Methoden	210
21.9 Klassenmethoden	211
21.10 Properties	213
21.11 Public-Attribute statt private Attribute	217
21.12 Vererbung	219
21.12.1 Oberbegriffe und Oberklassen	219
21.12.2 Ein einfaches Beispiel	219
21.12.3 Überladen, Überschreiben und Polymorphie	220
21.12.4 Vererbung in Python	223
21.13 Mehrfachvererbung	226
21.13.1 Einführung	226
21.13.2 Beispiel: <code>CalendarClock</code>	227
21.13.3 Diamand-Problem oder „deadly diamond of death“	234
21.13.4 <code>super</code> und MRO	236
21.14 Magische Methoden und Operatorüberladung	240
21.14.1 Einführung	240
21.14.2 Übersicht magische Methoden	241
21.14.3 Beispielklasse: <code>Length</code>	242
21.15 Standardklassen als Basisklassen	245
21.16 Aufgaben	246

Teil II	Weiterführende Themen	249
22	Tests und Fehler	251
22.1	Einführung	251
22.2	Modultests	253
22.3	Modultests unter Benutzung von <code>__name__</code>	254
22.4	<code>doctest</code> -Modul	256
22.5	Testgetriebene Entwicklung oder „Im Anfang war der Test“	259
22.6	<code>unittest</code>	260
22.7	Methoden der Klasse <code>TestCase</code>	262
22.8	Aufgaben	265
23	Systemprogrammierung	267
23.1	Systemprogrammierung	267
23.2	Häufig falsch verstanden: Shell	267
23.3	<code>os</code> -Modul	268
23.3.1	Vorbemerkungen	268
23.3.2	Umgebungsvariablen	269
23.3.3	Dateiverarbeitung auf niedrigerer Ebene	271
23.3.4	Die <code>exec</code> -„Familie“	276
23.3.5	Weitere Funktionen im Überblick	282
23.3.6	<code>os.path</code> - Arbeiten mit Pfaden	295
23.4	<code>shutil</code> -Modul	303
24	Forks	309
24.1	<code>Fork</code>	309
24.2	<code>Fork</code> in Python	309
25	Daten konservieren	313
25.1	Persistente Speicherung	313
25.2	<code>Pickle</code> -Modul	314
25.2.1	Daten „einpökeln“ mit <code>pickle.dump</code>	314
25.2.2	<code>pickle.load</code>	315
25.3	Ein persistentes Dictionary mit <code>shelve</code>	315
26	Reguläre Ausdrücke	319
26.1	Ursprünge und Verbreitung	319
26.2	Stringvergleiche	319
26.3	Überlappungen und Teilstrings	321
26.4	Das <code>re</code> -Modul	321

26.5	Matching-Problem	322
26.6	Syntax der regulären Ausdrücke	324
26.6.1	Beliebiges Zeichen	324
26.7	Zeichenauswahl	324
26.8	Endliche Automaten	325
26.9	Anfang und Ende eines Strings	326
26.10	Vordefinierte Zeichenklassen	328
26.11	Optionale Teile	330
26.12	Quantoren	330
26.13	Gruppierungen und Rückwärtsreferenzen	333
26.13.1	Match-Objekte	333
26.14	Umfangreiche Übung	335
26.15	Alles finden mit findall	337
26.16	Alternativen	338
26.17	Kompilierung von regulären Ausdrücken	339
26.18	Aufspalten eines Strings mit oder ohne regulären Ausdruck	340
26.18.1	split-Methode der String-Klasse	340
26.18.2	split-Methode des re-Moduls	342
26.18.3	Wörter filtern	343
26.19	Suchen und Ersetzen mit sub	344
26.20	Aufgaben	345
27	lambda, map, filter und reduce	347
27.1	lambda	347
27.2	map	350
27.3	Filtern von sequentiellen Datentypen mittels „filter“	352
27.4	reduce	352
27.5	Aufgaben	354
28	Listen-Abstraktion/List Comprehension	355
28.1	Die Alternative zu Lambda und Co.	355
28.2	Syntax	356
28.3	Weitere Beispiele	356
28.4	Die zugrunde liegende Idee	357
28.5	Anspruchsvolleres Beispiel	358
28.6	Mengen-Abstraktion	358
28.7	Rekursive Primzahlberechnung	359
28.8	Generatoren-Abstraktion	359
28.9	Aufgaben	360

29	Generatoren und Iteratoren	361
29.1	Einführung	361
29.2	Iteration in for-Schleifen	361
29.3	Generatoren	363
29.4	Beispiele	365
29.4.1	Permutationen	365
29.4.2	Variationen und Kombinationen	366
29.5	Generatoren zähmen mit firstn und islice	367
29.6	send-Methode	368
29.7	Generator-Ausdrücke	369
29.8	Aufgaben	370
30	Memoisation	371
30.1	Bedeutung und Herkunft des Begriffes	371
30.2	Memoisation mit Dekorateur-Funktionen	372
30.3	Memoize in einer Class	373
30.4	Dekoratoren in Python	373
30.5	Überprüfung von Argumenten durch Dekoratoren	375
31	NumPy	377
31.1	Übersicht	377
31.2	Arrays in NumPy	379
31.3	Arrays flach machen	381
31.4	Arrays umdimensionieren	383
31.5	Arrays konkatenieren	384
31.6	Array, neue Dimension hinzufügen	386
31.7	Array mit Nullen und Einsen initialisieren	386
31.8	Matrizenarithmetik	387
31.9	Vektoraddition und Vektorsubtraktion	388
31.10	Matrix-Klasse	390
31.10.1	Matrix-Produkt	391
31.10.2	Eine einfache praktische Anwendung	392
31.11	Inverse Matrix	393
31.12	Kreuzprodukt / Vektorprodukt	394
31.13	Lineare Gleichungssysteme	394
31.14	Polynome	396
31.15	Aufgaben	397

Teil III	Umfassende Beispiele	399
32	Bruchklasse	401
32.1	Brüche à la 1001 Nacht	401
32.2	Zurück in die Gegenwart	402
32.3	Rechenregeln	405
32.3.1	Multiplikation von Brüchen	405
32.3.2	Division von Brüchen	406
32.3.3	Addition von Brüchen	407
32.3.4	Subtraktion von Brüchen	407
32.4	Integer plus Bruch	408
32.4.1	Die Bruchklasse im Überblick	409
32.5	Die Fraction-Klasse im fractions-Modul	411
33	Mastermind	413
33.1	Die Ursprünge des Spiels	413
33.2	Die Spielregeln	414
33.2.1	„Bulls and Cows“	414
33.2.2	Mastermind	414
33.3	Kombinatorikmodul	415
33.4	Mastermind in Python	416
34	Textklassifikation	421
34.1	Einführung in die Textklassifikation	421
34.2	Textklassifikation: Aufgabe	423
34.3	Naive-Bayes-Klassifikator	423
34.3.1	Definition	423
34.3.2	Bayes-Theorem	423
34.4	Formale Herleitung der Naive-Bayes-Klassifikation	424
34.5	Textklassifikation in Python	426
34.5.1	BagOfWords-Klasse	426
34.5.2	Document-Klasse	427
34.5.3	DocumentClass-Klasse	429
34.5.4	Pool-Klasse	430

Teil IV Lösungen zu den Aufgaben	433
35 Lösungen zu den Aufgaben	435
35.1 Lösungen zu Kapitel 5 (Sequentielle Datentypen)	435
35.2 Lösungen zu Kapitel 6 (Dictionaries)	438
35.3 Lösungen zu Kapitel 9 (Verzweigungen)	439
35.4 Lösungen zu Kapitel 10 (Schleifen)	442
35.5 Lösungen zu Kapitel 11 (Dateien lesen und schreiben)	445
35.6 Lösungen zu Kapitel 16 (Listen und Tupel im Detail)	446
35.7 Lösungen zu Kapitel 14 (Funktionen)	449
35.8 Lösungen zu Kapitel 15 (Rekursive Funktionen)	453
35.9 Lösungen zu Kapitel 19 (Alles über Strings ...)	457
35.10 Lösungen zu Kapitel 21 (Objektorientierte Programmierung)	460
35.11 Lösungen zu Kapitel 22 (Tests und Fehler)	468
35.12 Lösungen zu Kapitel 26 (Reguläre Ausdrücke)	468
35.13 Lösungen zu Kapitel 27 (lambda, map, filter und reduce)	474
35.14 Lösungen zu Kapitel 28 (Listen-Abstraktion/List Comprehension)	475
35.15 Lösungen zu Kapitel 29 (Generatoren und Iteratoren)	476
35.16 Lösungen zu Kapitel 31 (NumPy)	479
Stichwortverzeichnis	481