

# Inhalt

<b>Vorwort</b> .....	<b>XIX</b>
<b>Danksagung</b> .....	<b>XX</b>
<b>Teil I: Einleitung</b> .....	<b>1</b>
<b>1 Einleitung</b> .....	<b>3</b>
1.1 Einfach und schnell zu lernen .....	3
1.2 Geschichte von Python .....	3
1.3 Zen von Python .....	4
1.4 Zielgruppe des Buches .....	5
1.5 Aufbau des Buches .....	6
1.6 Programmieren lernen „interaktiv“ .....	7
1.7 Download der Beispiele und Hilfe .....	8
1.8 Anregungen und Kritik .....	8
<b>Teil II: Grundlagen</b> .....	<b>9</b>
<b>2 Kommandos und Programme</b> .....	<b>11</b>
2.1 Erste Schritte mit Python .....	11
2.1.1 Linux .....	11
2.1.2 Windows .....	12
2.1.3 macOS .....	13
2.2 Herkunft und Bedeutung des Begriffes interaktive Shell .....	13
2.2.1 Erste Schritte in der interaktiven Shell .....	14
2.3 Verlassen der Python-Shell .....	15
2.4 Benutzung von Variablen .....	15
2.5 Mehrzeilige Anweisungen in der interaktiven Shell .....	16
2.6 Programme schreiben oder schnell mal der Welt “Hallo” sagen .....	17

<b>3</b>	<b>Bytecode und Maschinencode .....</b>	<b>21</b>
3.1	Einführung.....	21
3.2	Unterschied zwischen Programmier- und Skriptsprachen .....	21
3.3	Interpreter- oder Compilersprache.....	21
<b>4</b>	<b>Datentypen und Variablen .....</b>	<b>25</b>
4.1	Einführung.....	25
4.2	Variablennamen .....	28
4.2.1	Gültige Variablennamen .....	28
4.2.2	Konventionen für Variablennamen .....	29
4.3	Datentypen .....	29
4.3.1	Ganze Zahlen .....	29
4.3.2	Fließkommazahlen.....	31
4.3.3	Zeichenketten.....	31
4.3.4	Boolesche Werte .....	31
4.3.5	Komplexe Zahlen .....	32
4.3.6	Operatoren .....	32
4.4	Statische und dynamische Typdeklaration .....	34
4.5	Typumwandlung .....	35
4.6	Datentyp ermitteln .....	36
<b>5</b>	<b>Sequentielle Datentypen .....</b>	<b>39</b>
5.1	Übersicht.....	39
5.1.1	Zeichenketten oder Strings .....	40
5.1.2	Listen.....	42
5.1.3	Tupel .....	42
5.2	Indizierung von sequentiellen Datentypen .....	43
5.3	Teilbereichsoperator.....	44
5.4	Die len-Funktion .....	47
5.5	Aufgaben .....	47
<b>6</b>	<b>Listen und Tupel im Detail.....</b>	<b>49</b>
6.1	Virtueller Einkaufsbummel.....	49
6.2	Stapelspeicher/Stacks .....	51
6.3	Stapelverarbeitung in Python: pop und append.....	52
6.4	extend .....	52
6.5	Module importieren .....	53
6.6	,+'-Operator oder append .....	55
6.7	Entfernen eines Wertes.....	56

6.8	Prüfen, ob ein Element in Liste enthalten ist .....	57
6.9	Finden der Position eines Elementes .....	57
6.10	Einfügen von Elementen.....	57
6.11	Besonderheiten bei Tupel .....	58
6.11.1	Leere Tupel.....	58
6.11.2	1-Tupel.....	58
6.11.3	Mehrfachzuweisungen, Packing und Unpacking .....	59
6.12	Die veränderliche Unveränderliche .....	60
6.13	Aufgaben .....	60
<b>7</b>	<b>Verzweigungen .....</b>	<b>63</b>
7.1	Anweisungsblöcke und Einrückungen.....	63
7.2	Bedingte Anweisungen in Python .....	66
7.2.1	Einfachste if-Anweisung.....	66
7.2.2	if-Anweisung mit else-Zweig.....	67
7.2.3	elif-Zweige.....	67
7.3	Vergleichsoperatoren.....	68
7.4	Zusammengesetzte Bedingungen .....	68
7.5	Wahr oder falsch: Bedingungen in Verzweigungen .....	69
7.6	Aufgaben .....	70
<b>8</b>	<b>Schleifen .....</b>	<b>71</b>
8.1	Übersicht.....	71
8.2	while-Schleife .....	72
8.3	break und continue.....	73
8.4	Die Alternative im Erfolgsfall: else .....	74
8.5	Iterierbare Objekte (iterables).....	76
8.6	For-Schleife .....	77
8.7	Aufgaben .....	80
<b>9</b>	<b>Dictionaries .....</b>	<b>83</b>
9.1	Definition und Benutzung.....	83
9.2	Fehlerfreie Zugriffe auf Dictionaries .....	86
9.3	Einfachere Definition von Dictionaries .....	88
9.4	Zulässige Typen für Schlüssel und Werte .....	88
9.5	Verschachtelte Dictionaries .....	89
9.6	Dictionaries in Listen wandeln .....	89
9.7	Weitere Methoden auf Dictionaries .....	90
9.8	Operatoren.....	92

9.9	Die zip-Funktion .....	93
9.10	Dictionaries aus Listen erzeugen .....	95
9.11	Aufgaben .....	95
<b>10</b>	<b>Mengen .....</b>	<b>99</b>
10.1	Übersicht .....	99
10.2	Mengen in Python .....	99
10.2.1	Sets erzeugen .....	100
10.2.2	Mengen von unveränderlichen Elementen .....	100
10.3	Frozensets .....	101
10.4	Operationen auf „set“-Objekten .....	101
10.4.1	add(element) .....	101
10.4.2	clear() .....	101
10.4.3	copy .....	102
10.4.4	difference() .....	102
10.4.5	difference_update() .....	102
10.4.6	discard(el) .....	103
10.4.7	remove(el) .....	103
10.4.8	intersection(s) .....	103
10.4.9	union(s) .....	104
10.4.10	isdisjoint() .....	104
10.4.11	issubset() .....	104
10.4.12	issuperset() .....	105
10.4.13	pop() .....	105
10.5	Erweiterte Zuweisungsoperatoren für Mengen .....	106
<b>11</b>	<b>Eingaben .....</b>	<b>107</b>
11.1	Eingabe mittels input .....	107
<b>12</b>	<b>Dateien lesen und schreiben .....</b>	<b>109</b>
12.1	Dateien .....	109
12.2	Text aus einer Datei lesen .....	109
12.3	Schreiben in eine Datei .....	111
12.4	In einem Rutsch lesen: readlines und read .....	111
12.5	with-Anweisung .....	112
12.6	Aufgaben .....	113

<b>13</b>	<b>Formatierte Ausgabe und Strings formatieren</b>	<b>115</b>
13.1	Wege, die Ausgabe zu formatieren	115
13.2	print-Funktion	116
13.3	Notwendigkeit	118
13.4	Formatierte Stringlitterale / f-Strings	118
13.5	Die String-Methode „format“	123
13.6	Stringmodulo-Operator oder Formatierung à la C	125
13.7	Benutzung von Dictionaries beim Aufruf der „format“-Methode	129
13.8	Benutzung von lokalen Variablen in „format“	130
13.9	Weitere String-Methoden zum Formatieren	131
<b>14</b>	<b>Referenzen, flaches und tiefes Kopieren</b>	<b>133</b>
14.1	Einführung	133
14.2	Swen und Sarah	133
14.3	Variablen sind Referenzen	134
14.4	Kopieren einer Liste	135
14.5	Flache Kopien	136
14.6	Kopieren mit deepcopy	138
14.7	Problemverhinderung	139
14.8	Deepcopy für Dictionaries	139
<b>15</b>	<b>Funktionen</b>	<b>141</b>
15.1	Allgemein	141
15.2	Funktionen	141
15.3	Docstring	143
15.4	Standardwerte für Funktionen	145
15.5	Schlüsselwortparameter	146
15.6	Funktionen ohne oder mit leerer return-Anweisung	146
15.7	Mehrere Rückgabewerte	147
15.8	Parameterübergabe im Detail	148
15.9	Effekte bei veränderlichen Objekten	150
15.10	Kommandozeilenparameter	151
15.11	Variable Anzahl von Parametern / Variadische Funktionen	152
15.12	* in Funktionsaufrufen	154
15.13	Beliebige Schlüsselwortparameter	155
15.14	Doppeltes Sternchen im Funktionsaufruf	155
15.15	Aufgaben	155

<b>16</b>	<b>Wertebereich von Variablen</b> .....	<b>159</b>
16.1	Einführung .....	159
16.2	Globale und lokale Variablen in Funktionen .....	159
16.3	nonlocal-Variablen .....	162
<b>17</b>	<b>Rekursive Funktionen</b> .....	<b>167</b>
17.1	Definition und Herkunft des Begriffs .....	167
17.2	Definition der Rekursion .....	168
17.3	Rekursive Funktionen in Python .....	168
17.4	Die Tücken der Rekursion .....	169
17.5	Fibonacci-Folge in Python .....	170
17.6	Aufgaben .....	174
<b>18</b>	<b>Sortieren</b> .....	<b>177</b>
18.1	Sortieren von Listen .....	177
18.1.1	„sort“ und „sorted“ .....	177
18.1.2	Umkehrung der Sortierreihenfolge .....	178
18.1.3	Eigene Sortierfunktionen .....	178
18.2	Aufgaben .....	181
<b>19</b>	<b>Modularisierung</b> .....	<b>183</b>
19.1	Module .....	183
19.1.1	Namensräume von Modulen .....	184
19.1.2	Namensräume umbenennen .....	185
19.1.3	Modularten .....	185
19.1.4	Suchpfad für Module .....	186
19.1.5	Inhalt eines Moduls .....	187
19.1.6	Eigene Module .....	187
19.1.7	Dokumentation für eigene Module .....	188
19.2	Pakete .....	189
19.2.1	Einfaches Paket erzeugen .....	190
19.2.2	Komplexeres Paket .....	191
19.2.3	Komplettes Paket importieren .....	193
<b>20</b>	<b>Alles über Strings</b> .....	<b>197</b>
20.1	... fast alles .....	197
20.2	Aufspalten von Zeichenketten .....	198
20.2.1	split .....	199
20.2.2	rsplit .....	201

20.2.3	Folge von Trennzeichen .....	202
20.2.4	splitlines .....	203
20.2.5	partition und rpartition .....	204
20.3	Zusammenfügen von Stringlisten mit join .....	204
20.4	Suchen von Teilstrings .....	205
20.4.1	„in“ oder „not in“ .....	205
20.4.2	s.find(substring[, start[, end]]) .....	205
20.4.3	s.rfind(substring[, start[, end]]) .....	206
20.4.4	s.index(substring[, start[, end]]) .....	206
20.4.5	s.rindex(substring[, start[, end]]) .....	206
20.4.6	s.count(substring[, start[, end]]) .....	206
20.5	Suchen und Ersetzen .....	207
20.6	Nur noch Kleinbuchstaben oder Großbuchstaben .....	207
20.7	capitalize und title .....	208
20.8	Stripping Strings .....	208
20.9	Strings ausrichten .....	209
20.10	String-Tests .....	209
20.11	Aufgaben .....	211
<b>21</b>	<b>Ausnahmebehandlung .....</b>	<b>215</b>
21.1	Abfangen mehrerer Ausnahmen .....	217
21.2	except mit mehrfachen Ausnahmen .....	218
21.3	Die optionale else-Klausel .....	218
21.4	Fehlerinformationen über sys.exc_info .....	219
21.5	Exceptions generieren .....	220
21.6	Finalisierungsaktion .....	220
	<b>Teil III: Objektorientierte Programmierung .....</b>	<b>223</b>
<b>22</b>	<b>Grundlegende Aspekte .....</b>	<b>225</b>
22.1	Bibliotheksvergleich .....	225
22.2	Objekte und Instanzen einer Klasse .....	227
22.3	Kapselung von Daten und Methoden .....	228
22.4	Eine minimale Klasse in Python .....	228
22.5	Eigenschaften und Attribute .....	229
22.6	Methoden .....	232
22.7	Instanzvariablen .....	232
22.8	Die __init__-Methode .....	233

22.9	Destruktor .....	234
22.10	Datenkapselung, Datenabstraktion und Geheimnisprinzip .....	236
	22.10.1 Definitionen.....	236
	22.10.2 Zugriffsmethoden .....	238
	22.10.3 Properties.....	239
	22.10.4 Public-, Protected- und Private-Attribute.....	240
	22.10.5 Weitere Möglichkeiten der Properties .....	242
	22.10.6 Properties mit Dekoratoren .....	245
22.11	Stringausgaben mit str und repr .....	246
22.12	Klassenattribute .....	251
22.13	Statische Methoden .....	253
	22.13.1 Einleitendes Beispiel.....	254
	22.13.2 Getter und Setter für private Klassenattribute .....	255
22.14	Public-Attribute statt private Attribute .....	256
22.15	Magische Methoden und Operatorüberladung.....	258
	22.15.1 Einführung .....	258
	22.15.2 Übersicht magische Methoden.....	259
	22.15.3 Beispielklasse: Length .....	261
<b>23</b>	<b>Bruchklasse .....</b>	<b>265</b>
23.1	Brüche à la 1001 Nacht .....	265
23.2	Zurück in die Gegenwart.....	266
23.3	Rechenregeln .....	268
	23.3.1 Multiplikation von Brüchen .....	268
	23.3.2 Division von Brüchen.....	269
	23.3.3 Addition von Brüchen .....	270
	23.3.4 Subtraktion von Brüchen.....	270
	23.3.5 Vergleichsoperatoren .....	270
23.4	Integer plus Bruch .....	271
	23.4.1 Die Bruchklasse im Überblick .....	272
23.5	Fraction-Klasse.....	274
<b>24</b>	<b>Aufrufbare Objekte .....</b>	<b>275</b>
24.1	Einführung.....	275
	24.1.1 Die callable-Funktion.....	275
	24.1.2 Klassen statt Funktionen .....	276



<b>25</b>	<b>Vererbung</b> .....	<b>279</b>
25.1	Oberbegriffe und Oberklassen .....	279
25.2	Ein einfaches Beispiel .....	280
25.3	Überladen, Überschreiben und Polymorphie .....	281
25.4	Vererbung in Python.....	284
25.5	Klassenmethoden.....	287
25.6	Standardklassen als Basisklassen .....	289
<b>26</b>	<b>Mehrfachvererbung</b> .....	<b>291</b>
26.1	Einführung.....	291
26.2	Beispiel: KalenderUndUhr.....	292
26.3	Diamand-Problem oder „deadly diamond of death“ .....	300
26.4	super und MRO .....	302
26.4.1	Ein einfaches Beispiel.....	302
26.4.2	Ein umfangreicheres Beispiel.....	305
<b>27</b>	<b>Slots</b> .....	<b>307</b>
27.1	Erzeugung von dynamischen Attributen verhindern .....	307
<b>28</b>	<b>Dynamische Erzeugung von Klassen</b> .....	<b>309</b>
28.1	Beziehung zwischen „class“ und „type“ .....	309
<b>29</b>	<b>Metaklassen</b> .....	<b>313</b>
29.1	Motivation .....	313
29.2	Definition .....	318
29.3	Definition von Metaklassen in Python .....	318
29.4	Singletons mit Metaklassen erstellen .....	321
29.5	Beispiel: Methodenaufrufe zählen.....	322
29.5.1	Einführung .....	322
29.5.2	Vorbereitungen .....	322
29.5.3	Ein Dekorateur, um Funktionsaufrufe zu zählen .....	323
29.5.4	Die Metaklasse „Aufrufzähler“ .....	324
<b>30</b>	<b>Abstrakte Klassen</b> .....	<b>327</b>
<b>31</b>	<b>Aufgaben zur Objektorientierung</b> .....	<b>331</b>

<b>Teil IV: Funktionale Programmierung</b> .....	<b>335</b>
<b>32 Begriffsbestimmung</b> .....	<b>337</b>
<b>33 lambda, map, filter und reduce</b> .....	<b>339</b>
33.1 lambda .....	339
33.2 map .....	341
33.3 Filtern von sequentiellen Datentypen mittels „filter“ .....	344
33.4 reduce .....	344
33.5 Aufgaben .....	346
<b>34 Listen-Abstraktion/List Comprehension</b> .....	<b>347</b>
34.1 Die Alternative zu Lambda und Co. ....	347
34.2 Syntax .....	348
34.3 Weitere Beispiele .....	348
34.4 Die zugrunde liegende Idee .....	349
34.5 Anspruchsvolleres Beispiel .....	349
34.6 Mengen-Abstraktion .....	350
34.7 Rekursive Primzahlberechnung .....	351
34.8 Generatoren-Abstraktion .....	351
34.9 Aufgaben .....	352
<b>35 Generatoren und Iteratoren</b> .....	<b>353</b>
35.1 Einführung .....	353
35.2 Iteration in for-Schleifen .....	353
35.3 Generatoren .....	355
35.4 Endlos-Generatoren zähmen mit firstn und islice .....	359
35.5 Sinnvollere Beispiele .....	360
35.6 Beispiele aus der Kombinatorik .....	361
35.6.1 Permutationen .....	361
35.6.2 Variationen und Kombinationen .....	362
35.7 Generator-Ausdrücke .....	364
35.8 return-Anweisungen in Generatoren .....	365
35.9 send-Methode .....	366
35.10 Die close-Methode .....	370
35.11 Die throw-Methode .....	371
35.12 Dekoration von Generatoren .....	375
35.13 yield from .....	376
35.14 Aufgaben .....	378

<b>36</b>	<b>Dekoratore</b> .....	<b>381</b>
36.1	Einführung Dekoratore .....	381
36.1.1	Verschachtelte Funktionen .....	382
36.1.2	Funktionen als Parameter .....	384
36.1.3	Funktionen als Rückgabewert .....	385
36.1.4	Fabrikfunktionen .....	386
36.2	Ein einfacher Dekorateur .....	388
36.3	@-Syntax für Dekoratore .....	389
36.4	Anwendungsfälle für Dekoratore .....	392
36.4.1	Überprüfung von Argumenten durch Dekoratore .....	392
36.4.2	Funktionsaufrufe mit einem Dekorateur zählen .....	393
36.5	Dekoratore mit Parametern .....	395
36.6	Benutzung von Wraps aus functools .....	396
36.7	Eine Klasse als Dekorateur benutzen .....	398
36.8	Memoisation .....	399
36.8.1	Bedeutung und Herkunft des Begriffs .....	399
36.8.2	Memoisation mit Dekorateurfunktionen .....	399
36.8.3	Memoisation mit einer Klasse .....	400
36.8.4	Memoisation mit functools.lru_cache .....	401
<b>Teil V: Weiterführende Themen</b> .....		<b>405</b>
<b>37</b>	<b>Tests und Fehler</b> .....	<b>407</b>
37.1	Einführung .....	407
37.2	Modultests .....	409
37.3	Modultests unter Benutzung von <code>__name__</code> .....	410
37.4	doctest-Modul .....	412
37.5	Testgetriebene Entwicklung oder „Im Anfang war der Test“ .....	415
37.6	unittest .....	417
37.7	Methoden der Klasse TestCase .....	419
37.8	Aufgaben .....	420
<b>38</b>	<b>Daten konservieren</b> .....	<b>423</b>
38.1	Persistente Speicherung .....	423
38.2	Pickle-Modul .....	424
38.2.1	Daten „einpökeln“ mit <code>pickle.dump</code> .....	424
38.2.2	<code>pickle.load</code> .....	425
38.3	Ein persistentes Dictionary mit <code>shelve</code> .....	425

<b>39</b>	<b>Reguläre Ausdrücke</b> .....	<b>429</b>
39.1	Ursprünge und Verbreitung .....	429
39.2	Stringvergleiche .....	429
39.3	Überlappungen und Teilstrings .....	431
39.4	Das re-Modul .....	431
39.5	Matching-Problem .....	432
39.6	Syntax der regulären Ausdrücke .....	434
	39.6.1 Beliebiges Zeichen .....	434
39.7	Zeichenauswahl .....	435
39.8	Endliche Automaten .....	436
39.9	Anfang und Ende eines Strings .....	436
39.10	Vordefinierte Zeichenklassen .....	438
39.11	Optionale Teile .....	440
39.12	Quantoren .....	441
39.13	Gruppierungen und Rückwärtsreferenzen .....	443
	39.13.1 Match-Objekte .....	443
39.14	Iteration über Matches mit finditer .....	446
39.15	Umfangreiche Übung .....	446
39.16	Alles finden mit findall .....	448
39.17	Alternativen .....	449
39.18	Compilierung von regulären Ausdrücken .....	450
39.19	Aufspalten eines Strings mit oder ohne regulären Ausdruck .....	450
	39.19.1 split-Methode der String-Klasse .....	450
	39.19.2 split-Methode des re-Moduls .....	452
	39.19.3 Wörter filtern .....	454
39.20	Suchen und Ersetzen mit sub .....	455
39.21	Aufgaben .....	455
<b>40</b>	<b>Typanmerkungen</b> .....	<b>459</b>
40.1	Einführung .....	459
40.2	Einfaches Beispiel .....	460
40.3	Variablenanmerkungen .....	461
40.4	Listenbeispiele .....	462
40.5	Listen mit homogenem Typ .....	463
	40.5.1 Version 3.6 bis 3.8 .....	463
	40.5.2 Python 3.9 und später .....	464

<b>41</b>	<b>Systemprogrammierung</b>	<b>467</b>
41.1	Einleitung	467
41.2	Häufig falsch verstanden: Shell	467
41.3	os-Modul	468
41.3.1	Vorbemerkungen	468
41.3.2	Umgebungsvariablen	469
41.3.3	Dateiverarbeitung auf niedrigerer Ebene	471
41.3.4	Weitere Funktionen im Überblick	476
41.3.5	os.path – Arbeiten mit Pfaden	490
41.4	shutil-Modul	498
41.5	glob-Modul	503
<b>42</b>	<b>Forks</b>	<b>505</b>
42.1	Fork	505
42.2	Fork in Python	505
<b>Teil VI: Lösungen zu den Aufgaben</b>		<b>509</b>
<b>43</b>	<b>Lösungen zu den Aufgaben</b>	<b>511</b>
43.1	Lösungen zu Kapitel 5 (Sequentielle Datentypen)	511
43.2	Lösungen zu Kapitel 6 (Listen und Tupel im Detail)	514
43.3	Lösungen zu Kapitel 7 (Verzweigungen)	516
43.4	Lösungen zu Kapitel 8 (Schleifen)	518
43.5	Lösungen zu Kapitel 9 (Dictionaries)	521
43.6	Lösungen zu Kapitel 12 (Dateien lesen und schreiben)	523
43.7	Lösungen zu Kapitel 15 (Funktionen)	524
43.8	Lösungen zu Kapitel 17 (Rekursive Funktionen)	529
43.9	Lösungen zu Kapitel 18 (Sortieren)	534
43.10	Lösungen zu Kapitel 20 (Alles über Strings ...)	537
43.11	Lösungen zu den Kapiteln 22 bis 31 (Aufgaben zur Objektorientierung)	540
43.12	Lösungen zu Kapitel 33 (lambda, map, filter und reduce)	554
43.13	Lösungen zu Kapitel 34 (Listen-Abstraktion/List Comprehension)	555
43.14	Lösungen zu Kapitel 35 (Generatoren und Iteratoren)	556
43.15	Lösungen zu Kapitel 37 (Tests und Fehler)	560
43.16	Lösungen zu Kapitel 39 (Reguläre Ausdrücke)	560
<b>Stichwortverzeichnis</b>		<b>567</b>