

# Inhaltsverzeichnis

<b>Einleitung</b>	15
<b>I Begegnung mit Python</b>	19
I.I Was ist Python?	19
I.2 Python-Versionen	20
I.3 IDLE	21
I.3.I Die Python-Shell	21
I.3.2 Hotkeys	23
I.4 Die Python-Shell als Taschenrechner	24
I.4.I Operatoren und Terme	24
I.4.2 Zahlen	25
I.4.3 Mathematische Funktionen	29
I.5 Hilfe	34
I.6 Namen und Zuweisungen	35
I.6.I Zuweisungen für mehrere Variablen	37
I.6.2 Rechnen mit Variablen in der Shell	37
I.6.3 Syntaxregeln für Bezeichner	38
I.6.4 Neue Namen für Funktionen und andere Objekte	39
I.6.5 Erweiterte Zuweisungen	39
I.7 Mit Python-Befehlen Geräte steuern	40
I.7.I Projekt: Eine LED ein- und ausschalten	40
I.7.2 Das Modul RPI.GPIO	42
I.7.3 Steuern mit Relais	43
I.7.4 Projekt: Eine Taschenlampe an- und ausschalten	44
I.8 Aufgaben	46
I.9 Lösungen	49
<b>2 Python-Skripte</b>	51
2.I Ein Skript mit IDLE erstellen	51
2.I.I Ein neues Projekt starten	51
2.I.2 Programmtext eingeben	52
2.I.3 Das Skript ausführen	52
2.I.4 Shortcuts	53

2.2	Programme ausführen . . . . .	53
2.2.1	Programm in der Konsole starten . . . . .	53
2.2.2	Anklicken des Programmicons im File-Manager . . . . .	55
2.3	Interaktive Programme – das EVA-Prinzip . . . . .	57
2.3.1	Format mit Bedeutung – Aufbau eines Python-Programmtextes . . . . .	58
2.3.2	Eingabe – die input()-Funktion . . . . .	59
2.3.3	Verarbeitung – Umwandeln von Datentypen und Rechnen . . . . .	59
2.3.4	Ausgabe – die print()-Funktion . . . . .	60
2.4	Programmverzweigungen . . . . .	61
2.4.1	Einfache Bedingungen . . . . .	62
2.4.2	Wie erkennt man eine gute Melone? Zusammengesetzte Bedingungen . . . . .	64
2.4.3	Einseitige Verzweigungen und Programmblöcke . . . . .	65
2.4.4	Haben Sie Ihr Idealgewicht? . . . . .	66
2.4.5	Eine Besonderheit von Python: Wahrheitswerte für Objekte . . . . .	69
2.5	Bedingte Wiederholung – die while-Anweisung . . . . .	70
2.5.1	Projekt: Zahlenraten . . . . .	71
2.5.2	Have a break! Abbruch einer Schleife . . . . .	72
2.6	Projekte mit dem GPIO . . . . .	72
2.6.1	Blinklicht . . . . .	73
2.6.2	Schalter . . . . .	73
2.6.3	Zähler . . . . .	75
2.6.4	Grundzustände festlegen und Flankensteuerung . . . . .	76
2.7	Projekt: Eine Alarmanlage . . . . .	77
2.7.1	Aufbau und Arbeitsweise der Alarmanlage . . . . .	77
2.7.2	Programmierung . . . . .	78
2.8	Aufgaben . . . . .	80
2.9	Lösungen . . . . .	83
3	<b>Kollektionen: Mengen, Listen, Tupel und Dictionaries</b> . . . . .	87
3.1	Die Typihierarchie . . . . .	87
3.2	Gemeinsame Operationen für Kollektionen . . . . .	89
3.3	Kollektionen in Bedingungen . . . . .	90
3.3.1	Projekt: Kundenberatung . . . . .	91
3.3.2	Projekt: Sichere Kommunikation . . . . .	91

3.4	Iteration – die for-Anweisung .....	92
3.4.1	Verwendung von break .....	93
3.5	Sequenzen .....	94
3.5.1	Konkatenation und Vervielfältigung .....	94
3.5.2	Direkter Zugriff auf Elemente – Indizierung .....	95
3.5.3	Slicing .....	95
3.5.4	Projekt: Lesbare Zufallspasswörter .....	96
3.6	Tupel .....	98
3.7	Zeichenketten (Strings) .....	99
3.7.1	Strings durch Bytestrings codieren .....	100
3.7.2	Der Formatierungsoperator % .....	101
3.8	Listen .....	102
3.8.1	Listen sind Objekte und empfangen Botschaften .....	102
3.8.2	Klasse, Typ und Instanz .....	104
3.8.3	Kopie oder Alias? .....	104
3.8.4	Listenoperationen .....	105
3.8.5	Projekt: Zufallsnamen .....	107
3.8.6	Projekt: Telefonliste .....	108
3.8.7	Listen durch Comprehensions erzeugen .....	109
3.9	Zahlen in einer Folge – range()-Funktion .....	110
3.10	Projekt: Klopfzeichen .....	111
3.11	Mengen .....	115
3.11.1	Projekt: Häufigkeit von Buchstaben in einem Text .....	116
3.12	Projekt: Zufallsounds .....	117
3.12.1	Wie kommen Töne aus dem Raspberry Pi? .....	117
3.12.2	Sounds mit PyGame .....	118
3.12.3	Programmierung .....	119
3.13	Dictionaries .....	120
3.13.1	Operationen für Dictionaries .....	122
3.13.2	Projekt: Morsen .....	123
3.14	Projekt: Der kürzeste Weg zum Ziel .....	124
3.15	Aufgaben .....	127
3.16	Lösungen .....	129
4	Funktionen .....	131
4.1	Aufruf von Funktionen .....	131
4.1.1	Unterschiedliche Anzahl von Argumenten .....	132
4.1.2	Positionsargumente und Schlüsselwort-Argumente .....	132
4.1.3	Für Experten: Funktionen als Argumente .....	133

4.2	Definition von Funktionen .....	134
4.3	Funktionen in der IDLE-Shell testen .....	136
4.4	Docstrings .....	136
4.5	Veränderliche und unveränderliche Objekte als Parameter .....	137
4.6	Voreingestellte Parameterwerte .....	139
4.7	Beliebige Anzahl von Parametern .....	140
4.8	Die return-Anweisung unter der Lupe .....	141
4.9	Mehr Sicherheit! Vorbedingungen testen .....	143
4.10	Namensräume: Global und lokal .....	145
4.11	Rekursive Funktionen – die Hohe Schule der Algorithmitik .....	147
4.11.1	Projekt: Rekursive Summe .....	147
4.11.2	Projekt: Quicksort .....	148
4.12	Experimente zur Rekursion mit der Turtle-Grafik .....	149
4.12.1	Turtle-Befehle im interaktiven Modus .....	149
4.12.2	Projekt: Eine rekursive Spirale aus Quadraten .....	151
4.12.3	Projekt: Pythagorasbaum .....	153
4.12.4	Projekt: Eine Koch-Schneeflocke .....	155
4.13	Projekt: Der Sierpinski-Tepich .....	157
4.14	Funktionen per Knopfdruck aufrufen: Callback-Funktionen .....	159
4.14.1	Projekt: Digitaler Türgong .....	160
4.14.2	Projekt: Verkehrszählungen – Zählen mit mehreren Knöpfen .....	161
4.15	Aufgaben .....	164
4.16	Lösungen .....	167
5	<b>Fenster für den RPi – Grafische Benutzungsoberflächen .....</b>	171
5.1	Wie macht man eine Benutzungsoberfläche? .....	171
5.2	Projekt: Die digitale Lostrommel .....	172
5.2.1	Die Gestaltung der Widgets .....	174
5.2.2	Das Layout-Management .....	175
5.3	Bilder auf Widgets .....	177
5.3.1	Projekt: Ein visueller Zufallsgenerator .....	178
5.3.2	Bilder verarbeiten .....	179
5.3.3	Projekt: Schwarzweißmalerei .....	181
5.4	Projekt: Der Krimiautomat .....	182
5.4.1	Texteingabe .....	182
5.4.2	Programmierung .....	184

5.5	Wer die Wahl hat, hat die Qual: Checkbutton und Radiobutton . . . . .	185
5.5.1	Projekt: Automatische Urlaubsgrüße . . . . .	186
5.5.2	Projekt: Digitaler Glückskekks . . . . .	188
5.6	Viele Widgets schnell platziert: Das Grid-Layout . . . . .	190
5.6.1	Projekt: Rechenquiz . . . . .	191
5.7	Projekt: Farbmixer . . . . .	194
5.8	Projekt: Editor mit Pulldown-Menüs . . . . .	196
5.8.1	Aufbau einer Menüstruktur . . . . .	197
5.8.2	Programmierung . . . . .	198
5.9	Aufgaben . . . . .	200
5.10	Lösungen . . . . .	202
6	<b>Daten finden, laden und speichern</b> . . . . .	207
6.1	Dateien . . . . .	207
6.1.1	Daten speichern . . . . .	207
6.1.2	Daten laden . . . . .	208
6.2	Ein Blick hinter die Kulissen: Die SD-Karte . . . . .	208
6.3	Datenstrukturen haltbar machen mit pickle . . . . .	211
6.4	Versuch und Irrtum – Mehr Zuverlässigkeit durch try-Anweisungen . . . . .	212
6.5	Projekt: Karteikasten . . . . .	212
6.5.1	Der Editor . . . . .	213
6.5.2	Der Presenter . . . . .	216
6.6	Benutzungsoberfläche zum Laden und Speichern . . . . .	219
6.6.1	Dialogboxen . . . . .	219
6.6.2	Erweiterung des Editors für Karteikarten . . . . .	221
6.6.3	Erweiterung des Presenters . . . . .	224
6.7	Daten aus dem Internet . . . . .	226
6.8	Projekt: Goethe oder Schiller? . . . . .	227
6.8.1	Methoden der String-Objekte . . . . .	228
6.8.2	Programmierung . . . . .	230
6.9	Daten finden mit regulären Ausdrücken . . . . .	233
6.9.1	Reguläre Ausdrücke . . . . .	233
6.9.2	Die Funktion findall() . . . . .	235
6.9.3	Projekt: Staumelder . . . . .	235
6.9.4	Programmierung . . . . .	236
6.10	Aufgaben . . . . .	239
6.11	Lösungen . . . . .	241

<b>7</b>	<b>Projekte mit Zeitfunktionen</b>	243
7.1	Projekt: Fünf Sekunden stoppen und gewinnen	243
7.2	Datum und Zeit im Überblick	245
7.3	Projekt: Digitaluhr	246
7.3.1	Woher bekommt der RPi die Zeit?	246
7.3.2	Was ist ein Prozess?	247
7.3.3	Vollbildmodus	249
7.3.4	Event-Verarbeitung	252
7.3.5	Autostart	253
7.4	Projekt: Ein digitaler Bilderrahmen	253
7.4.1	Zugriff auf das Dateisystem: Das Modul os	254
7.4.2	Python Imaging Library (PIL)	255
7.4.3	Die Programmierung	258
7.5	Projekt: Wahrnehmungstest	260
7.5.1	Die Programmierung	261
7.6	Projekt: Stoppuhr mit Gong	264
7.7	Aufgaben	267
7.8	Lösungen	268
<b>8</b>	<b>Objektorientierte Programmierung</b>	273
8.1	Überall Objekte	273
8.2	Klassen und Vererbung bei Python	275
8.2.1	Einführendes Beispiel: Alphabet	276
8.2.2	Qualitätsmerkmal Änderbarkeit	279
8.2.3	Vererbung	280
8.3	Pong revisited	282
8.3.1	Bau eines Fußschalters	283
8.3.2	Die Klasse Canvas	285
8.3.3	Die Programmierung	289
8.4	Renn, Lola renn!	293
8.4.1	Vorbereitung	294
8.4.2	Struktur des Programms	294
8.4.3	Background	296
8.4.4	Switch	296
8.4.5	Display	298
8.4.6	Clock	298
8.4.7	Die Klasse Runner	299
8.4.8	Controller	300
8.4.9	Module	302

8.5	Aufgaben .....	304
8.6	Lösungen .....	307
9	<b>Sensortechnik .....</b>	315
9.1	Was ist ein digitaler Temperatursensor? .....	315
9.2	Den DS1820 anschließen .....	316
9.3	Temperaturdaten lesen .....	317
9.3.1	Temperaturdaten eines Sensors automatisch auswerten ..	318
9.4	Projekt: Ein digitales Thermometer mit mehreren Sensoren .....	320
9.4.1	Ein Modul für die Messwerterfassung .....	321
9.4.2	Die grafische Oberfläche .....	323
9.4.3	Temperaturdaten per E-Mail senden .....	324
9.5	Projekt: Ein Temperaturplotter .....	326
9.5.1	Temperatur-Zeitdiagramme .....	326
9.5.2	Programmierung .....	327
9.6	Projekt: Mobile Datenerfassung .....	330
9.6.1	Experimente mit mobiler Temperaturerfassung .....	332
9.6.2	Programmierung .....	332
9.6.3	Wiedergabe der Daten .....	334
9.7	Spannung messen .....	334
9.7.1	Das SPI-Protokoll .....	336
9.7.2	Bitverarbeitung .....	338
9.7.3	Programmierung .....	341
9.8	Aufgaben .....	343
9.9	Lösungen .....	345
10	<b>Projekte mit der Kamera .....</b>	355
10.1	Das Kameramodul anschließen .....	355
10.2	Die Kamerasoftware .....	357
10.2.1	Einzelbilder .....	358
10.3	Projekt: Kameraoptionen testen .....	359
10.4	Projekt: Überwachungskamera – Livebild auf dem Bildschirm ..	361
10.5	Projekt: Bewegung erfassen .....	363
10.6	Projekt: Gerichtete Bewegungen erfassen .....	366
10.6.1	Files verarbeiten mit subprocess und StringIO .....	367
10.6.2	Die Programmierung .....	368
10.7	Projekt: Birnen oder Tomaten? .....	373
10.7.1	Magische Methoden – das Überladen von Operatoren ..	374
10.7.2	Programmierung .....	377
10.7.3	Weiterentwicklungen .....	380

10.8	Das Modul picamera .....	380
10.8.1	Die Klasse PiCamera .....	381
10.8.2	Projekt: Einen Film aufnehmen .....	382
10.8.3	Projekt: Fotos per E-Mail verschicken.....	383
10.9	Was ist los am Autobahnkreuz? Bilder einer Webcam auswerten...	385
10.9.1	Webcams im Internet anzapfen .....	385
10.9.2	Auf einem Foto zeichnen – das Modul ImageDraw .....	386
10.9.3	Projekt: Verkehrsdichte auf der Autobahn .....	388
10.10	Randbemerkung: Was darf man? Was soll man? .....	392
10.11	Aufgabe.....	393
10.12	Lösung .....	394
II	<b>Webserver.....</b>	397
II.1	Der RPi im lokalen Netz .....	397
II.1.1	WLAN .....	397
II.1.2	Eine dauerhafte IP-Adresse für den RPi.....	398
II.1.3	Über SSH auf dem RPi arbeiten .....	399
II.2	Ein Webserver .....	399
II.2.1	Den Server starten .....	400
II.2.2	Die Startseite .....	401
II.2.3	Den Server testen .....	401
II.3	Was ist los im Gartenteich? .....	402
II.3.1	Projekt: Einfache Webcam mit statischer Webseite.....	402
II.3.2	CGI-Skripte .....	406
II.3.3	Hilfe, mein CGI-Skript läuft nicht! .....	409
II.3.4	Interaktive Webseiten .....	411
II.3.5	Eingabekomponenten in einem HTML-Formular.....	413
II.3.6	Verarbeitung von Eingaben in einem CGI-Skript .....	414
II.3.7	Zugriff aus der Ferne .....	416
II.4	Geräte über das Internet steuern .....	417
II.4.1	Privilegierte Rechte für ein CGI-Skript.....	417
II.4.2	Programmierung .....	418
II.5	Datenbanken .....	420
II.5.1	Das Modul sqlite3 .....	420
II.5.2	Projekt: Freies Obst .....	423
II.6	Aufgaben .....	430
II.7	Lösungen .....	432

<b>A</b>	<b>Den Raspberry Pi einrichten . . . . .</b>	437
A.1	Hardware-Ausstattung . . . . .	437
A.2	Das Betriebssystem installieren . . . . .	437
	A.2.1 Download der Software . . . . .	437
	A.2.2 Die SD-Karte formatieren . . . . .	438
A.3	Den Raspberry Pi das erste Mal starten und konfigurieren . . . . .	439
A.4	Die grafische Oberfläche von Raspbian . . . . .	440
<b>B</b>	<b>Der GPIO . . . . .</b>	443
B.1	Der GPIO . . . . .	443
B.2	Ein Flachbandkabel mit Pfostenverbindern . . . . .	446
<b>C</b>	<b>Autostart . . . . .</b>	449
<b>D</b>	<b>So entstand das Titelbild . . . . .</b>	451
	<b>Stichwortverzeichnis . . . . .</b>	455