

# Inhaltsverzeichnis

	<b>Einleitung</b> .....	15
<b>I</b>	<b>Begegnung mit Python</b> .....	19
I.1	Was ist Python? .....	19
I.2	Python-Versionen .....	20
I.3	IDLE .....	21
I.3.1	Die Python-Shell .....	22
I.3.2	Hotkeys .....	23
I.4	Die Python-Shell als Taschenrechner .....	24
I.4.1	Operatoren und Terme .....	24
I.4.2	Zahlen .....	26
I.4.3	Mathematische Funktionen .....	30
I.5	Hilfe .....	35
I.6	Namen und Zuweisungen .....	36
I.6.1	Zuweisungen für mehrere Variablen .....	37
I.6.2	Rechnen mit Variablen in der Shell .....	38
I.6.3	Syntaxregeln für Bezeichner .....	39
I.6.4	Neue Namen für Funktionen und andere Objekte .....	40
I.6.5	Erweiterte Zuweisungen .....	40
I.7	Mit Python-Befehlen Geräte steuern .....	41
I.7.1	Projekt: Eine LED ein- und ausschalten .....	41
I.7.2	Das Modul RPI.GPIO .....	43
I.7.3	Steuern mit Relais .....	44
I.7.4	Projekt: Eine Taschenlampe an- und ausschalten .....	45
I.8	Aufgaben .....	47
I.9	Lösungen .....	50
<b>2</b>	<b>Python-Skripte</b> .....	53
2.1	Ein Skript mit IDLE erstellen .....	53
2.1.1	Ein neues Projekt starten .....	53
2.1.2	Programmtext eingeben .....	54
2.1.3	Das Skript ausführen .....	54
2.1.4	Shortcuts .....	55

2.2	Programme ausführen .....	55
2.2.1	Programm in der Konsole starten .....	55
2.2.2	Anklicken des Programmicons im File-Manager.....	57
2.3	Interaktive Programme – das EVA-Prinzip .....	59
2.3.1	Format mit Bedeutung – Aufbau eines Python-Programmtextes .....	60
2.3.2	Eingabe – die input()-Funktion .....	61
2.3.3	Verarbeitung – Umwandeln von Datentypen und Rechnen.....	61
2.3.4	Ausgabe – die print()-Funktion .....	62
2.4	Programmverzweigungen .....	63
2.4.1	Einfache Bedingungen.....	64
2.4.2	Wie erkennt man eine gute Melone? Zusammengesetzte Bedingungen .....	66
2.4.3	Einseitige Verzweigungen und Programmblöcke .....	67
2.4.4	Haben Sie Ihr Idealgewicht? .....	68
2.4.5	Eine Besonderheit von Python: Wahrheitswerte für Objekte	71
2.5	Bedingte Wiederholung – die while-Anweisung.....	72
2.5.1	Projekt: Zahlenraten.....	73
2.5.2	Have a break! Abbruch einer Schleife.....	74
2.6	Projekte mit dem GPIO.....	74
2.6.1	Blinklicht.....	75
2.6.2	Schalter .....	75
2.6.3	Zähler .....	77
2.6.4	Grundzustände festlegen und Flankensteuerung .....	78
2.7	Projekt: Eine Alarmanlage.....	79
2.7.1	Aufbau und Arbeitsweise der Alarmanlage .....	79
2.7.2	Programmierung .....	80
2.8	Aufgaben .....	83
2.9	Lösungen .....	85
3	<b>Kollektionen: Mengen, Listen, Tupel und Dictionaries .....</b>	<b>89</b>
3.1	Die Typhierarchie .....	89
3.2	Gemeinsame Operationen für Kollektionen .....	91
3.3	Kollektionen in Bedingungen.....	92
3.3.1	Projekt: Kundenberatung.....	93
3.3.2	Projekt: Sichere Kommunikation .....	93
3.4	Iteration – die for-Anweisung.....	94
3.4.1	Verwendung von break .....	95

3.5	Sequenzen .....	96
3.5.1	Konkatenation und Vervielfältigung .....	96
3.5.2	Direkter Zugriff auf Elemente – Indizierung .....	97
3.5.3	Slicing .....	97
3.5.4	Projekt: Lesbare Zufallspasswörter .....	98
3.6	Tupel .....	100
3.7	Zeichenketten (Strings) .....	101
3.7.1	Strings durch Bytestrings codieren .....	102
3.7.2	Formatieren .....	103
3.8	Listen .....	105
3.8.1	Listen sind Objekte und empfangen Botschaften .....	105
3.8.2	Klasse, Typ und Instanz .....	106
3.8.3	Kopie oder Alias? .....	107
3.8.4	Listenoperationen .....	108
3.8.5	Projekt: Zufallsnamen .....	109
3.8.6	Projekt: Telefonliste .....	110
3.8.7	Listen durch Comprehensions erzeugen .....	111
3.9	Zahlen in einer Folge – range()-Funktion .....	112
3.10	Projekt: Klopfschritte .....	114
3.11	Mengen .....	117
3.11.1	Projekt: Häufigkeit von Buchstaben in einem Text .....	118
3.12	Projekt: Zufallssounds .....	119
3.12.1	Wie kommen Töne aus dem Raspberry Pi? .....	119
3.12.2	Sounds mit PyGame .....	121
3.12.3	Programmierung .....	122
3.13	Dictionaries .....	123
3.13.1	Operationen für Dictionaries .....	125
3.13.2	Projekt: Morsen .....	126
3.14	Projekt: Der kürzeste Weg zum Ziel .....	127
3.15	Aufgaben .....	130
3.16	Lösungen .....	132
4	<b>Funktionen</b> .....	135
4.1	Aufruf von Funktionen .....	135
4.1.1	Unterschiedliche Anzahl von Argumenten .....	136
4.1.2	Positionsargumente und Schlüsselwort-Argumente .....	136
4.1.3	Für Experten: Funktionen als Argumente .....	137
4.2	Definition von Funktionen .....	138
4.3	Funktionen in der IDLE-Shell testen .....	140

4.4	Docstrings . . . . .	140
4.5	Veränderliche und unveränderliche Objekte als Parameter . . . . .	141
4.6	Voreingestellte Parameterwerte . . . . .	143
4.7	Beliebige Anzahl von Parametern . . . . .	144
4.8	Die return-Anweisung unter der Lupe . . . . .	145
4.9	Mehr Sicherheit! Vorbedingungen testen . . . . .	147
4.10	Namensräume: Global und lokal . . . . .	149
4.11	Rekursive Funktionen – die Hohe Schule der Algorithmik . . . . .	151
4.11.1	Projekt: Rekursive Summe . . . . .	151
4.11.2	Projekt: Quicksort . . . . .	152
4.12	Experimente zur Rekursion mit der Turtle-Grafik . . . . .	153
4.12.1	Turtle-Befehle im interaktiven Modus . . . . .	154
4.12.2	Projekt: Eine rekursive Spirale aus Quadraten . . . . .	155
4.12.3	Projekt: Pythagorasbaum . . . . .	157
4.12.4	Projekt: Eine Koch-Schneeflocke . . . . .	159
4.13	Projekt: Der Sierpinski-Teppich . . . . .	161
4.14	Funktionen per Knopfdruck aufrufen: Callback-Funktionen . . . . .	163
4.14.1	Projekt: Digitaler Türgong . . . . .	164
4.14.2	Projekt: Verkehrszählungen – Zählen mit mehreren Knöpfen . . . . .	165
4.15	Aufgaben . . . . .	168
4.16	Lösungen . . . . .	171
5	<b>Fenster für den RPi – Grafische Benutzungsoberflächen . . . . .</b>	<b>175</b>
5.1	Wie macht man eine Benutzungsoberfläche? . . . . .	175
5.2	Projekt: Die digitale Lostrommel . . . . .	176
5.2.1	Die Gestaltung der Widgets . . . . .	178
5.2.2	Das Layout-Management . . . . .	179
5.3	Bilder auf Widgets . . . . .	181
5.3.1	Projekt: Ein visueller Zufallsgenerator . . . . .	182
5.3.2	Bilder verarbeiten . . . . .	183
5.3.3	Projekt: Schwarz-Weiß-Malerei . . . . .	185
5.4	Projekt: Der Krimiautomat . . . . .	186
5.4.1	Texteingabe . . . . .	187
5.4.2	Programmierung . . . . .	188
5.5	Wer die Wahl hat, hat die Qual: Checkbutton und Radiobutton . . . . .	189
5.5.1	Projekt: Automatische Urlaubsgrüße . . . . .	190
5.5.2	Projekt: Digitaler Glückskeks . . . . .	192

5.6	Viele Widgets schnell platziert: Das Grid-Layout .....	194
5.6.1	Projekt: Rechenquiz .....	195
5.7	Projekt: Farbmixer .....	198
5.8	Projekt: Editor mit Pulldown-Menüs .....	200
5.8.1	Aufbau einer Menüstruktur .....	201
5.8.2	Programmierung .....	202
5.9	Aufgaben .....	204
5.10	Lösungen .....	206
<b>6</b>	<b>Daten finden, laden und speichern .....</b>	<b>211</b>
6.1	Dateien .....	211
6.1.1	Daten speichern .....	211
6.1.2	Daten laden .....	212
6.2	Ein Blick hinter die Kulissen: Die SD-Karte .....	212
6.3	Datenstrukturen haltbar machen mit pickle .....	215
6.4	Versuch und Irrtum – Mehr Zuverlässigkeit durch try-Anweisungen .....	216
6.5	Projekt: Karteikasten .....	216
6.5.1	Der Editor .....	217
6.5.2	Der Presenter .....	220
6.6	Benutzungsoberfläche zum Laden und Speichern .....	223
6.6.1	Dialogboxen .....	223
6.6.2	Erweiterung des Editors für Karteikarten .....	225
6.6.3	Erweiterung des Presenters .....	228
6.7	Daten aus dem Internet .....	230
6.8	Projekt: Goethe oder Schiller? .....	231
6.8.1	Methoden der String-Objekte .....	232
6.8.2	Programmierung .....	234
6.9	Daten finden mit regulären Ausdrücken .....	237
6.9.1	Reguläre Ausdrücke .....	237
6.9.2	Die Funktion findall() .....	239
6.9.3	Projekt: Staumelder .....	239
6.9.4	Programmierung .....	240
6.10	Aufgaben .....	243
6.11	Lösungen .....	244
<b>7</b>	<b>Projekte mit Zeitfunktionen .....</b>	<b>247</b>
7.1	Projekt: Fünf Sekunden stoppen und gewinnen .....	247
7.2	Datum und Zeit im Überblick .....	249

7.3	Projekt: Digitaluhr .....	250
7.3.1	Woher bekommt der RPi die Zeit? .....	250
7.3.2	Was ist ein Prozess? .....	251
7.3.3	Vollbildmodus. ....	253
7.3.4	Event-Verarbeitung. ....	256
7.3.5	Autostart .....	257
7.4	Projekt: Ein digitaler Bilderrahmen .....	257
7.4.1	Zugriff auf das Dateisystem: Das Modul os .....	258
7.4.2	Python Imaging Library (PIL) .....	259
7.4.3	Die Programmierung. ....	261
7.5	Projekt: Wahrnehmungstest .....	263
7.5.1	Die Programmierung. ....	264
7.6	Projekt: Stoppuhr mit Gong .....	267
7.7	Aufgaben .....	270
7.8	Lösungen .....	271
8	<b>Objektorientierte Programmierung</b> .....	277
8.1	Überall Objekte .....	277
8.2	Klassen und Vererbung bei Python .....	279
8.2.1	Einführendes Beispiel: Alphabet .....	280
8.2.2	Qualitätsmerkmal Änderbarkeit .....	283
8.2.3	Vererbung .....	284
8.3	Pong revisited. ....	286
8.3.1	Bau eines Fußschalters .....	287
8.3.2	Die Klasse Canvas. ....	289
8.3.3	Die Programmierung. ....	293
8.4	Renn, Lola renn!. ....	297
8.4.1	Vorbereitung .....	298
8.4.2	Struktur des Programms .....	298
8.4.3	Background .....	300
8.4.4	Switch .....	300
8.4.5	Display .....	302
8.4.6	Clock .....	302
8.4.7	Die Klasse Runner .....	303
8.4.8	Controller .....	304
8.4.9	Module. ....	306
8.5	Aufgaben .....	308
8.6	Lösungen .....	311

<b>9</b>	<b>Sensortechnik</b>	<b>319</b>
9.1	Was ist ein digitaler Temperatursensor?	319
9.2	Den DS1820 anschließen	320
9.3	Temperaturdaten lesen	321
9.3.1	Temperaturdaten eines Sensors automatisch auswerten	323
9.4	Projekt: Ein digitales Thermometer mit mehreren Sensoren	324
9.4.1	Ein Modul für die Messwerterfassung	325
9.4.2	Die grafische Oberfläche	327
9.4.3	Temperaturdaten per E-Mail senden	328
9.5	Projekt: Ein Temperaturplotter	330
9.5.1	Temperatur-Zeitdiagramme	330
9.5.2	Programmierung	331
9.6	Projekt: Mobile Datenerfassung	334
9.6.1	Experimente mit mobiler Temperaturerfassung	336
9.6.2	Programmierung	336
9.6.3	Wiedergabe der Daten	338
9.7	Spannung messen	338
9.7.1	Das SPI-Protokoll	340
9.7.2	Programmierung	342
9.8	Alkoholsensor	344
9.8.1	Projekt: Achtung! Alkoholisches Getränk!	345
9.8.2	Projekt: Den Alkoholgehalt der Luft und von Flüssigkeiten messen	348
9.8.3	Wie kann man den Alkoholgehalt von Flüssigkeiten messen?	349
9.8.4	Messen und interpolieren	351
9.9	Projekte mit einem digitalen Lichtsensor	354
9.9.1	Das Modul tsl2561	355
9.9.2	Das Modul smbus	356
9.9.3	Mit dem Raspberry Pi Farben messen – Absorptionsspektrometer	360
9.10	Kohlendioxid-Sensor	366
9.10.1	Projekt: Datenlogger	367
9.10.2	Das Sensormodul kalibrieren	368
9.10.3	Projekt: Ein Experiment zur Diffusion von Gasen	369
9.11	Aufgaben	372
9.12	Lösungen	374

<b>10</b>	<b>Projekte mit der Kamera</b>	<b>383</b>
10.1	Das Kameramodul anschließen	383
10.2	Die Kamerasoftware	384
10.2.1	Einzelbilder	386
10.3	Projekt: Kameraoptionen testen	387
10.4	Projekt: Überwachungskamera – Livebild auf dem Bildschirm.	389
10.5	Projekt: Bewegung erfassen	390
10.6	Projekt: Gerichtete Bewegungen erfassen	394
10.6.1	Files verarbeiten mit subprocess und io	395
10.6.2	Die Programmierung	396
10.7	Projekt: Birnen oder Tomaten?	401
10.7.1	Magische Methoden – das Überladen von Operatoren	402
10.7.2	Programmierung	405
10.7.3	Weiterentwicklungen	408
10.8	Das Modul picamera	408
10.8.1	Die Klasse PiCamera	409
10.8.2	Projekt: Einen Film aufnehmen	410
10.8.3	Projekt: Fotos per E-Mail verschicken	411
10.9	Was ist los am Autobahnkreuz? Bilder einer Webcam auswerten.	413
10.9.1	Webcams im Internet anzapfen	413
10.9.2	Auf einem Foto zeichnen – das Modul ImageDraw	414
10.9.3	Projekt: Verkehrsdichte auf der Autobahn	417
10.9.4	Wie findet man den URL eines Webcam-Bildes?	421
10.10	Randbemerkung: Was darf man? Was soll man?	422
10.11	Aufgabe: Wie lang? Wie breit?	422
10.12	Lösung	423
<b>11</b>	<b>Webserver</b>	<b>427</b>
11.1	Der RPi im lokalen Netz	427
11.1.1	WLAN	427
11.1.2	Eine dauerhafte IP-Adresse für den RPi.	428
11.1.3	Über SSH auf dem RPi arbeiten	429
11.1.4	Virtual Network Computing (VNC).	429
11.2	Webserver	431
11.2.1	Der Apache-Webserver.	431
11.2.2	Ein Webserver mit Python	433
11.2.3	Die Startseite	434
11.2.4	Den Server testen	435



II.3	Was ist los im Gartenteich? .....	435
II.3.1	Projekt: Einfache Webcam mit statischer Webseite .....	435
II.3.2	CGI-Skripte .....	439
II.3.3	CGI-Skripte für den Apache-Server .....	442
II.3.4	Hilfe, mein CGI-Skript läuft nicht! .....	443
II.3.5	Interaktive Webseiten .....	444
II.3.6	Eingabekomponenten in einem HTML-Formular. ....	446
II.3.7	Verarbeitung von Eingaben in einem CGI-Skript .....	447
II.3.8	Zugriff aus der Ferne .....	449
II.4	Geräte über das Internet steuern .....	450
II.4.1	Privilegierte Rechte für ein CGI-Skript. ....	450
II.4.2	Programmierung .....	451
II.5	Datenbanken .....	453
II.5.1	Das Modul sqlite3 .....	453
II.5.2	Projekt: Freies Obst .....	456
II.5.3	Alternativen zu CGI .....	463
II.6	Aufgaben .....	464
II.7	Lösungen .....	465
I2	<b>Erweiterungen: OLED und HAT</b> .....	471
I2.1	OLED-Display .....	471
I2.2	Anschluss .....	471
I2.3	Installation der SSD1306-Bibliothek .....	472
I2.4	Auf dem Display Texte und Formen ausgeben. ....	472
I2.4.1	Projekt: Uhrzeit .....	473
I2.5	Mit ImageDraw zeichnen und Texte schreiben .....	474
I2.5.1	Grafiken zeichnen .....	475
I2.5.2	Schriftarten definieren. ....	477
I2.6	Projekt: Gespeicherte TrueType-Fonts darstellen. ....	478
I2.7	Projekt: Thermometer .....	480
I2.8	Sense HAT .....	481
I2.9	Die Klasse SenseHat .....	483
I2.10	Grafische Ausgabe über die LED-Matrix .....	485
I2.11	Die räumliche Orientierung des Sense HAT .....	486
I2.11.1	Projekt: Das Murrellabyrinth .....	487
I2.12	Der Joystick .....	490
I2.12.1	InputEvent .....	491
I2.12.2	Die Klasse JoyStick .....	492

- 12.12.3 Definition von Eventhandlern ..... 492
  - 12.12.4 Auf Events warten und Events abfragen ..... 494
- 12.13 Aufgaben ..... 496
- 12.14 Lösungen ..... 497
  
- A Den Raspberry Pi einrichten..... 501**
  - A.1 Hardware-Ausstattung..... 501
  - A.2 Das Betriebssystem installieren ..... 501
  - A.3 Raspberry Pi das erste Mal starten und konfigurieren ..... 502
  - A.4 Die grafische Oberfläche von Raspberry Pi OS ..... 503
  
- B Der GPIO ..... 507**
  - B.1 Pinbelegung und Funktionen..... 507
  - B.2 Ein Flachbandkabel mit Pfostenverbindern..... 510
  
- C Autostart ..... 513**
  
- D So entstand das Titelbild ..... 515**
  
- Stichwortverzeichnis ..... 519**