

# Inhaltsverzeichnis

<b>Einleitung</b> .....	<b>15</b>
<b>I Begegnung mit Python</b> .....	<b>19</b>
<b>I.1 Was ist Python?</b> .....	<b>19</b>
<b>I.2 Python-Versionen</b> .....	<b>20</b>
<b>I.3 IDLE</b> .....	<b>21</b>
<b>I.3.1 Die Python-Shell</b> .....	<b>21</b>
<b>I.3.2 Hotkeys</b> .....	<b>23</b>
<b>I.4 Die Python-Shell als Taschenrechner</b> .....	<b>24</b>
<b>I.4.1 Operatoren und Terme</b> .....	<b>24</b>
<b>I.4.2 Zahlen</b> .....	<b>25</b>
<b>I.4.3 Mathematische Funktionen</b> .....	<b>29</b>
<b>I.5 Hilfe</b> .....	<b>34</b>
<b>I.6 Namen und Zuweisungen</b> .....	<b>35</b>
<b>I.6.1 Zuweisungen für mehrere Variablen</b> .....	<b>37</b>
<b>I.6.2 Rechnen mit Variablen in der Shell</b> .....	<b>37</b>
<b>I.6.3 Syntaxregeln für Bezeichner</b> .....	<b>38</b>
<b>I.6.4 Neue Namen für Funktionen und andere Objekte</b> .....	<b>39</b>
<b>I.6.5 Erweiterte Zuweisungen</b> .....	<b>39</b>
<b>I.7 Mit Python-Befehlen Geräte steuern</b> .....	<b>40</b>
<b>I.7.1 Projekt: Eine LED ein- und ausschalten</b> .....	<b>40</b>
<b>I.7.2 Das Modul RPI.GPIO</b> .....	<b>42</b>
<b>I.7.3 Das Interface PiFace Digital</b> .....	<b>43</b>
<b>I.7.4 Projekt: Eine Taschenlampe an- und ausschalten</b> .....	<b>45</b>
<b>I.8 Aufgaben</b> .....	<b>48</b>
<b>I.8.1 Aufgabe 1: Formeln</b> .....	<b>48</b>
<b>I.8.2 Aufgabe 2: Gebäude</b> .....	<b>48</b>
<b>I.8.3 Aufgabe 3: Zylinder</b> .....	<b>48</b>
<b>I.8.4 Aufgabe 4: Anweisungen</b> .....	<b>49</b>
<b>I.8.5 Aufgabe 5: Visualisierungen interpretieren</b> .....	<b>49</b>
<b>I.8.6 Aufgabe 6</b> .....	<b>50</b>

1.9	Lösungen .....	50
1.9.1	Lösung 1 .....	50
1.9.2	Lösung 2 .....	50
1.9.3	Lösung 3 .....	50
1.9.4	Lösung 4 .....	50
1.9.5	Lösung 5 .....	51
1.9.6	Lösung 6 .....	51
2	Python-Skripte .....	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

2.7	Projekt: Eine Alarmanlage .....	78
2.7.1	Die digitalen Eingänge des PiFace .....	78
2.7.2	Aufbau und Arbeitsweise der Alarmanlage .....	80
2.7.3	Programmierung .....	81
2.8	Aufgaben .....	82
2.8.1	Aufgabe 1: Anpeilen .....	82
2.8.2	Aufgabe 2: Boolesche Ausdrücke .....	83
2.8.3	Aufgabe 3: Quiz .....	83
2.8.4	Aufgabe 4: Wiederholte Berechnung .....	84
2.8.5	Aufgabe 5: Gesteuertes Blinken .....	84
2.9	Lösungen .....	84
2.9.1	Lösung 1 .....	84
2.9.2	Lösung 2 .....	85
2.9.3	Lösung 3 .....	85
2.9.4	Lösung 4 .....	87
2.9.5	Lösung 5 .....	87
3	Kollektionen: Mengen, Listen, Tupel und Dictionaries .....	89
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	Der Formatierungsoperator % .....	103
3.8	Listen .....	104
3.8.1	Listen sind Objekte und empfangen Botschaften .....	104
3.8.2	Klasse, Typ und Instanz .....	106
3.8.3	Kopie oder Alias? .....	106
3.8.4	Listenoperationen .....	107

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: Klopfszeichen. . . . .	113
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. . . . .	120
3.12.3	Programmierung . . . . .	121
3.13	Dictionaries . . . . .	122
3.13.1	Operationen für Dictionaries . . . . .	124
3.13.2	Projekt: Morsen. . . . .	124
3.14	Projekt: Der kürzeste Weg zum Ziel . . . . .	126
3.15	Aufgaben . . . . .	129
3.15.1	Aufgabe 1: Länge von Sequenzen . . . . .	129
3.15.2	Aufgabe 2: Lottozahlen. . . . .	130
3.15.3	Aufgabe 3: Visualisierung der range()-Funktion . . . . .	130
3.16	Lösungen . . . . .	131
3.16.1	Lösung 1. . . . .	131
3.16.2	Lösung 2 . . . . .	131
3.16.3	Lösung 3 . . . . .	132
4	<b>Funktionen</b> . . . . .	133
4.1	Aufruf von Funktionen . . . . .	133
4.1.1	Unterschiedliche Anzahl von Argumenten . . . . .	134
4.1.2	Positionsargumente und Schlüsselwort-Argumente . . . . .	134
4.1.3	Für Experten: Funktionen als Argumente . . . . .	135
4.2	Definition von Funktionen . . . . .	136
4.3	Funktionen in der IDLE-Shell testen . . . . .	138
4.4	Docstrings. . . . .	138
4.5	Veränderliche und unveränderliche Objekte als Parameter . . . . .	139
4.6	Voreingestellte Parameterwerte . . . . .	141
4.7	Beliebige Anzahl von Parametern . . . . .	142
4.8	Die return-Anweisung unter der Lupe . . . . .	143
4.9	Mehr Sicherheit! Vorbedingungen testen . . . . .	145
4.10	Namensräume: Global und lokal . . . . .	147

4.II	Rekursive Funktionen – die Hohe Schule der Algorithmik . . . . .	149
4.II.1	Projekt: Rekursive Summe . . . . .	149
4.II.2	Projekt: Quicksort . . . . .	150
4.III	Experimente zur Rekursion mit der Turtle-Grafik . . . . .	151
4.III.1	Turtle-Befehle im interaktiven Modus . . . . .	151
4.III.2	Projekt: Eine rekursive Spirale aus Quadraten . . . . .	153
4.III.3	Projekt: Pythagorasbaum . . . . .	155
4.III.4	Projekt: Eine Koch-Schneeflocke . . . . .	157
4.IV	Projekt: Der Sierpinski-Teppich . . . . .	159
4.V	Aufgaben . . . . .	161
4.V.1	Aufgabe 1: Morsen . . . . .	161
4.V.2	Aufgabe 2: Rekursive Funktionen – Puzzle . . . . .	162
4.V.3	Aufgabe 3: Ein Pythagorasbaum mit Zufallselementen . . . . .	164
4.VI	Lösungen . . . . .	164
4.VI.1	Lösung 1 . . . . .	164
4.VI.2	Lösung 2 . . . . .	165
4.VI.3	Lösung 3 . . . . .	166
5	Fenster für den RPi – Grafische Benutzungsoberflächen . . . . .	169
5.I	Wie macht man eine Benutzungsoberfläche? . . . . .	169
5.II	Projekt: Die digitale Lostrommel . . . . .	170
5.II.1	Die Gestaltung der Widgets . . . . .	172
5.II.2	Das Layout-Management . . . . .	173
5.III	Bilder auf Widgets . . . . .	175
5.III.1	Projekt: Ein visueller Zufallsgenerator . . . . .	176
5.III.2	Bilder verarbeiten . . . . .	177
5.III.3	Projekt: Schwarzweißmalerei . . . . .	179
5.IV	Projekt: Der Krimiautomat . . . . .	180
5.IV.1	Texteingabe . . . . .	180
5.IV.2	Programmierung . . . . .	182
5.V	Wer die Wahl hat, hat die Qual: Checkbutton und Radiobutton . . . . .	183
5.V.1	Projekt: Automatische Urlaubsgrüße . . . . .	184
5.V.2	Projekt: Digitaler Glückskeks . . . . .	186
5.VI	Viele Widgets schnell platziert: Das Grid-Layout . . . . .	188
5.VI.1	Projekt: Rechenquiz . . . . .	189
5.VII	Projekt: Farbmixer . . . . .	192
5.VIII	Projekt: Editor mit Pulldown-Menüs . . . . .	194
5.VIII.1	Aufbau einer Menüstruktur . . . . .	195
5.VIII.2	Programmierung . . . . .	196

5.9	Aufgaben .....	198
5.9.1	Aufgabe 1: Hangman mit Tastaturfeld .....	198
5.9.2	Aufgabe 2: Rasterbilder nach Meisenbach .....	199
5.10	Lösungen .....	200
5.10.1	Lösung 1 .....	200
5.10.2	Lösung 2 .....	202
6	Daten finden, laden und speichern .....	205
6.1	Dateien .....	205
6.1.1	Daten speichern .....	205
6.1.2	Daten laden .....	206
6.2	Ein Blick hinter die Kulissen: Die SD-Karte .....	206
6.3	Datenstrukturen haltbar machen mit pickle .....	209
6.4	Versuch und Irrtum – Mehr Zuverlässigkeit durch try-Anweisungen .....	210
6.5	Projekt: Karteikasten .....	210
6.5.1	Der Editor .....	211
6.5.2	Der Presenter .....	214
6.6	Benutzungsoberfläche zum Laden und Speichern .....	217
6.6.1	Dialogboxen .....	217
6.6.2	Erweiterung des Editors für Karteikarten .....	219
6.6.3	Erweiterung des Presenters .....	222
6.7	Daten aus dem Internet .....	224
6.8	Projekt: Goethe oder Schiller? .....	225
6.8.1	Methoden der String-Objekte .....	226
6.8.2	Programmierung .....	228
6.9	Daten finden mit regulären Ausdrücken .....	231
6.9.1	Reguläre Ausdrücke .....	231
6.9.2	Die Funktion findall() .....	233
6.9.3	Projekt: Staumelder .....	233
6.9.4	Programmierung .....	234
6.10	Aufgaben .....	237
6.10.1	Aufgabe 1: Reguläre Ausdrücke .....	237
6.10.2	Aufgabe 2: Geheime Botschaften .....	238
6.10.3	Aufgabe 3: Was reimt sich auf ...? .....	238
6.11	Lösungen .....	239
6.11.1	Lösung 1 .....	239
6.11.2	.....	239

6.11.3	Lösung 2 . . . . .	239
6.11.4	Lösung 3 . . . . .	239
7	<b>Projekte mit Zeitfunktionen . . . . .</b>	241
7.1	Projekt: Fünf Sekunden stoppen und gewinnen. . . . .	241
7.2	Datum und Zeit im Überblick . . . . .	243
7.3	Projekt: Digitaluhr . . . . .	244
7.3.1	Woher bekommt der RPi die Zeit? . . . . .	244
7.3.2	Was ist ein Prozess? . . . . .	245
7.3.3	Vollbildmodus . . . . .	247
7.3.4	Event-Verarbeitung . . . . .	250
7.3.5	Autostart . . . . .	251
7.4	Projekt: Ein digitaler Bilderrahmen . . . . .	251
7.4.1	Zugriff auf das Dateisystem: Das Modul os . . . . .	252
7.4.2	Python Imaging Library (PIL) . . . . .	253
7.4.3	Die Programmierung . . . . .	255
7.5	Projekt: Wahrnehmungstest . . . . .	257
7.5.1	Die Programmierung . . . . .	258
7.6	Projekt: Stoppuhr mit Gong . . . . .	261
7.7	Aufgaben . . . . .	264
7.7.1	Aufgabe 1: Zeiteinstellung . . . . .	264
7.7.2	Aufgabe 2: Zahlenschloss mit Tastaturfeld . . . . .	265
7.8	Lösungen . . . . .	265
7.8.1	Lösung 1 . . . . .	265
7.8.2	Lösung 2 . . . . .	267
8	<b>Objektorientierte Programmierung . . . . .</b>	271
8.1	Klassen und Vererbung bei Python . . . . .	271
8.1.1	Einführendes Beispiel: Alphabet . . . . .	272
8.1.2	Qualitätsmerkmal Änderbarkeit . . . . .	275
8.1.3	Vererbung . . . . .	276
8.2	Pong revisited . . . . .	278
8.2.1	Bau eines Fußschalters . . . . .	279
8.2.2	Die Klasse Canvas . . . . .	281
8.2.3	Die Programmierung . . . . .	285
8.3	Renn, Lola renn! . . . . .	289
8.3.1	Vorbereitung . . . . .	290
8.3.2	Struktur des Programms . . . . .	290
8.3.3	Background . . . . .	292

8.3.4	Switch .....	292
8.3.5	Display .....	294
8.3.6	Clock .....	294
8.3.7	Die Klasse Runner .....	295
8.3.8	Controller .....	296
8.3.9	Module .....	298
8.4	Aufgaben .....	300
8.4.1	Aufgabe 1: Buchstabensuppe .....	300
8.4.2	Aufgabe 2: Drumloops .....	301
8.4.3	Aufgabe 3: Beats mit Sound .....	302
8.5	Lösungen .....	303
8.5.1	Lösung 1 .....	303
8.5.2	Lösung 2 .....	305
8.5.3	Lösung 3 .....	308
9	Sensortechnik .....	311
9.1	Was ist ein digitaler Temperatursensor? .....	311
9.2	Den DS1820 anschließen .....	312
9.3	Temperaturdaten lesen .....	313
9.3.1	Temperaturdaten mehrerer Sensoren automatisch auswerten .....	314
9.4	Projekt: Ein digitales Thermometer mit mehreren Sensoren .....	316
9.4.1	Ein Modul für die Messwerterfassung .....	316
9.4.2	Die grafische Oberfläche .....	319
9.5	Projekt: Ein Temperaturplotter .....	320
9.5.1	Temperatur-Zeitdiagramme .....	320
9.5.2	Programmierung .....	321
9.6	Projekt: Mobile Datenerfassung .....	325
9.6.1	Experimente mit mobiler Temperaturerfassung .....	326
9.6.2	Programmierung .....	327
9.6.3	Wiedergabe der Daten .....	328
9.7	Spannung messen .....	329
9.7.1	Das SPI-Protokoll .....	331
9.7.2	Bitverarbeitung .....	333
9.7.3	Programmierung .....	336
9.8	Aufgaben .....	338
9.8.1	Aufgabe 1: Kalt, wärmer, heiß! .....	338
9.8.2	Aufgabe 2: Spannungsmesser .....	339
9.8.3	Aufgabe 3: Autosimulator .....	339

9.9	Lösungen .....	340
9.9.1	Lösung 1 .....	340
9.9.2	Lösung 2 .....	341
9.9.3	Lösung 3 .....	344
9.9.4	Lösung zum Rätsel aus Abschnitt 9.5 .....	347
10	Projekte mit der Kamera .....	349
10.1	Das Kameramodul anschließen .....	349
10.2	Die Kamerasoftware .....	351
10.2.1	Einzelbilder .....	352
10.3	Projekt: Kameraoptionen testen .....	353
10.4	Projekt: Überwachungskamera – Livebild auf dem Bildschirm .....	355
10.5	Projekt: Bewegung erfassen .....	357
10.6	Projekt: Gerichtete Bewegungen erfassen .....	360
10.6.1	Files verarbeiten mit subprocess und StringIO .....	361
10.6.2	Die Programmierung .....	362
10.7	Projekt: Birnen oder Tomaten? .....	367
10.7.1	Magische Methoden – das Überladen von Operatoren .....	368
10.7.2	Programmierung .....	371
10.7.3	Weiterentwicklungen .....	374
10.8	Randbemerkung: Was darf man? Was soll man? .....	374
10.9	Aufgabe .....	375
10.9.1	Aufgabe 1: Wie lang? Wie breit? .....	375
10.10	Lösung .....	376
10.10.1	Lösung 1 .....	376
II	Webserver .....	379
II.1	Der RPi im lokalen Netz .....	379
II.1.1	WLAN .....	379
II.1.2	Eine dauerhafte IP-Adresse für den RPi .....	380
II.1.3	Über SSH auf dem RPi arbeiten .....	381
II.2	Ein Webserver .....	381
II.2.1	Den Server starten .....	382
II.2.2	Die Startseite .....	383
II.2.3	Den Server testen .....	383
II.3	Was ist los im Gartenteich? .....	384
II.3.1	Projekt: Einfache Webcam mit statischer Webseite .....	384
II.3.2	CGI-Skripte .....	388
II.3.3	Hilfe, mein CGI-Skript läuft nicht! .....	391

II.3.4	Interaktive Webseiten . . . . .	393
II.3.5	Eingabekomponenten in einem HTML-Formular . . . . .	395
II.3.6	Verarbeitung von Eingaben in einem CGI-Skript . . . . .	396
II.3.7	Zugriff aus der Ferne . . . . .	398
II.4	Geräte über das Internet steuern . . . . .	399
II.4.1	Privilegierte Rechte für ein CGI-Skript . . . . .	399
II.4.2	Programmierung . . . . .	400
II.5	Datenbanken . . . . .	402
II.5.1	Das Modul <code>sqlite3</code> . . . . .	402
II.5.2	Projekt: Freies Obst . . . . .	405
II.6	Aufgaben . . . . .	412
II.6.1	Aufgabe 1: Sind Sie ein Optimist? . . . . .	412
II.6.2	Aufgabe 2: Eine interaktive Webcam . . . . .	413
II.7	Lösungen . . . . .	414
II.7.1	Lösung 1 . . . . .	414
II.7.2	Lösung 2 . . . . .	415
II.7.3	Lösung zur Zwischenfrage 1 . . . . .	417
II.7.4	Lösung zur Zwischenfrage 2 . . . . .	418
A	Den Raspberry Pi einrichten . . . . .	419
A.1	Hardware-Ausstattung . . . . .	419
A.2	Verpackung und Gehäuse . . . . .	419
A.3	Das Betriebssystem installieren . . . . .	420
A.3.1	Download der Software und Vorbereitung . . . . .	420
A.3.2	Betriebssystem auf die SD-Karte übertragen . . . . .	421
A.4	Den Raspberry Pi das erste Mal starten und konfigurieren . . . . .	421
A.5	Die grafische Oberfläche von Wheezy . . . . .	422
B	Wie verbindet man eine Steckplatine mit dem GPIO? . . . . .	425
B.1	Der GPIO . . . . .	425
B.2	Ein Flachbandkabel mit Pfostenverbindern . . . . .	428
B.3	Anschluss einer Steckplatine über ein Breakout-Board . . . . .	429
C	Autostart . . . . .	433
D	So entstand das Titelbild . . . . .	435
	Stichwortverzeichnis . . . . .	439