

# Inhalt

Vorwort .....	21
<b>TEIL I    Python lernen</b>	
<b>1    Hello, World!</b>	<b>25</b>
<hr/>	
<b>1.1   Python installieren</b> .....	<b>26</b>
Linux .....	26
Windows .....	27
macOS (Standardinstallation) .....	28
macOS (Manuelle »brew«-Installation) .....	29
<b>1.2   »Hello, World!« in der Python-Shell</b> .....	<b>31</b>
Python in der Shell kennenlernen .....	32
<b>1.3   »Hello, World!« als eigenständiges Script</b> .....	<b>34</b>
Das Hello-World-Script unter Windows ausführen .....	35
Das Hello-World-Script unter Linux und macOS ausführen	37
Was ist ein Script? .....	37
Windows versus Linux/macOS .....	38
Noch ein Beispiel .....	40
<b>1.4   Entwicklungsumgebungen (IDEs)</b> .....	<b>42</b>
Thonny .....	43
<b>1.5   Elementare Syntaxregeln</b> .....	<b>45</b>
Anweisungen .....	45
Blockelemente .....	46

»print« .....	47
»input« .....	49
Module und »import« .....	49
Kommentare .....	51
<b>1.6 Wiederholungsfragen .....</b>	<b>52</b>
<b>2 Variablen .....</b>	<b>53</b>
<hr/>	
<b>2.1 Grundregeln .....</b>	<b>53</b>
Variablenamen .....	54
<b>2.2 Datentypen .....</b>	<b>55</b>
Erkennung des Datentyps .....	56
Angabe des vorgesehen Datentyps (Type Annotation) .....	57
Mutable oder immutable .....	57
Veränderliche Daten kopieren .....	60
Typumwandlung .....	61
<b>2.3 Gültigkeitsbereich von Variablen .....</b>	<b>62</b>
<b>2.4 Wiederholungsfragen .....</b>	<b>64</b>
<b>3 Operatoren .....</b>	<b>65</b>
<hr/>	
<b>3.1 Überblick .....</b>	<b>65</b>
<b>3.2 Details und Sonderfälle .....</b>	<b>67</b>
Division .....	67
Zuweisung und Berechnung verbinden .....	67
»in«-Operator .....	68
Binäre versus logische Operatoren .....	68
Short-Circuit-Evaluation .....	69
Operatorhierarchie .....	69
Variablen und Daten vergleichen .....	71
Mehrfachvergleiche .....	72

	»operator«-Modul .....	72
	Operatoren selbst implementieren .....	73
3.3	Wiederholungsfragen .....	73
<b>4</b>	<b>Zahlen</b> .....	<b>74</b>
<hr/>		
4.1	Ganze Zahlen .....	75
	Division .....	75
	Binäre und hexadezimale Darstellung .....	76
	Zufallszahlen .....	77
4.2	Fließkommazahlen .....	77
	Rundungsfehler .....	77
	Zufallszahlen .....	78
4.3	Komplexe Zahlen, Brüche und Festkommazahlen .....	79
	Komplexe Zahlen .....	79
	Rationale Zahlen (Brüche) .....	80
	Dezimalzahlen .....	80
4.4	Boolesche Werte .....	81
4.5	Wiederholungsfragen .....	82
<b>5</b>	<b>Zeichenketten</b> .....	<b>83</b>
<hr/>		
5.1	Grundregeln .....	83
	Zeichenketten aneinanderfügen und vervielfältigen .....	84
	Sonderzeichen .....	84
	Raw-Zeichenketten .....	86
	»chr«- und »ord«-Funktion .....	86
5.2	Zugriff auf Teilzeichenketten .....	86
	»IndexError« .....	87
	Schrittweite (Stride) .....	87

<b>5.3</b>	<b>Zeichenkettenfunktionen</b> .....	88
	Eigenschaften von Zeichen(ketten) ermitteln .....	91
	Suchen und ersetzen .....	92
<b>5.4</b>	<b>Zeichenketten formatieren und konvertieren</b> .....	92
	Daten formatieren .....	93
	Formatierung mit dem Prozentoperator .....	94
	Formatierung mit der »format«-Methode .....	95
	»format«-Kurzschreibweise (Python 3.6) .....	96
<b>5.5</b>	<b>Lokalisierung</b> .....	97
	Lokalisierte Zeichenketten in Zahlen umwandeln .....	98
	Lokalisierungsbeispiel .....	99
<b>5.6</b>	<b>Reguläre Ausdrücke</b> .....	100
<b>5.7</b>	<b>Wiederholungsfragen und Übungen</b> .....	102
<b>6</b>	<b>Datum und Zeit</b> .....	103
<hr/>		
<b>6.1</b>	<b>Zeit ermitteln und darstellen</b> .....	103
	Datum aus Zeichenkette einlesen (»parsen«) .....	105
	Datum ohne Zeit .....	105
	Zeit ohne Datum .....	106
	Objekte für einen beliebigen Zeitpunkt .....	106
	Zeiten mit Zeitzone .....	107
<b>6.2</b>	<b>Mit Zeiten rechnen</b> .....	108
	Zeiten subtrahieren .....	109
	Laufzeit von Code messen .....	109
<b>6.3</b>	<b>Wiederholungsfragen und Übungen</b> .....	110
<b>7</b>	<b>Listen, Tupel, Sets und Dictionaries</b> .....	111
<hr/>		
<b>7.1</b>	<b>Listen</b> .....	112
	»range«-Funktion .....	113

	Listen und Zeichenketten .....	113
	List Comprehension .....	114
<b>7.2</b>	<b>Funktionen zur Verarbeitung von Listen</b> .....	<b>115</b>
	»map«-Funktion .....	117
	»reduce«-Funktion .....	119
	»filter«-Funktion .....	120
	Listen sortieren .....	121
	Zeichenketten mit deutschen Buchstaben sortieren .....	122
<b>7.3</b>	<b>Tupel (Sequenzen)</b> .....	<b>122</b>
	Anwendungsfälle .....	123
	»zip«-Funktion .....	124
<b>7.4</b>	<b>Sets (Mengen)</b> .....	<b>125</b>
	»set«-Methoden .....	126
	Mengenlehre .....	127
<b>7.5</b>	<b>Dictionaries</b> .....	<b>127</b>
	Dictionaries (assoziative Arrays) .....	127
	Verarbeitung von Dictionary-Elementen .....	129
	»zip«-Funktion .....	130
<b>7.6</b>	<b>Type Annotations</b> .....	<b>130</b>
<b>7.7</b>	<b>Arrays</b> .....	<b>131</b>
<b>7.8</b>	<b>Wiederholungsfragen und Übungen</b> .....	<b>133</b>
<b>8</b>	<b>Verzweigungen und Schleifen</b> .....	<b>134</b>
<hr/>		
<b>8.1</b>	<b>»if«-Verzweigung</b> .....	<b>134</b>
	»pass«-Schlüsselwort .....	136
	»if«-Kurzschreibweise .....	136
	Kein »switch« .....	137
<b>8.2</b>	<b>Beispiel: Schaltjahrtest</b> .....	<b>138</b>

<b>8.3</b>	<b>»for«-Schleife</b> .....	139
	Schleifen über Zahlenbereiche (»range«) .....	140
	Schleifen über die Zeichen einer Zeichenkette .....	141
	Schleifen über Listen, Tupel und Sets .....	141
	enumerate-Funktion .....	142
	List/Set/Tupel Comprehension .....	142
	Schleifen über Dictionaries, Dictionary Comprehension .....	143
	Generator Expressions .....	144
	»break«, »continue« und »else« .....	145
	»_« als Schleifenvariable .....	146
<b>8.4</b>	<b>»while«-Schleife</b> .....	147
<b>8.5</b>	<b>Beispiele für Schleifen</b> .....	148
	Summe der Zahlen von 1 bis 1.000 .....	149
	Einmaleins-Tabelle .....	150
	Parameter verarbeiten .....	151
<b>8.6</b>	<b>Wiederholungsfragen und Übungen</b> .....	153
<b>9</b>	<b>Funktionen</b> .....	155
<hr/>		
<b>9.1</b>	<b>Eigene Funktionen definieren</b> .....	156
	»Hello, Function!« .....	157
	Noch ein Beispiel .....	157
	Beispiel: Passwortgenerator .....	159
<b>9.2</b>	<b>Lokale und globale Variablen</b> .....	160
	Variablenverwaltung .....	161
	Globale Variablen .....	162
<b>9.3</b>	<b>Parameter</b> .....	164
	Parameter für unveränderliche Typen (immutable) .....	164
	Parameter für veränderliche Typen (mutable) .....	164
	Optionale Parameter .....	165

	Variable Parameteranzahl .....	166
	Parametertyp überprüfen .....	168
	Type Annotations .....	169
<b>9.4</b>	<b>Rekursion</b> .....	169
	Fakultät rekursiv berechnen .....	169
	Unkontrollierte Rekursion .....	170
<b>9.5</b>	<b>Lambda-Funktionen</b> .....	171
<b>9.6</b>	<b>Funktionale Programmierung</b> .....	173
	Beispiel: Funktionsgenerator .....	173
	Beispiel: verschachtelter Funktionsgenerator .....	174
	Beispiel: Listengenerator .....	174
<b>9.7</b>	<b>Generatoren</b> .....	176
	Fibonacci-Zahlen .....	176
<b>9.8</b>	<b>Wiederholungsfragen und Übungen</b> .....	179
<b>10</b>	<b>Umgang mit Fehlern (Exceptions)</b> .....	182
<hr/>		
<b>10.1</b>	<b>Fehlerabsicherung mit »try« und »except«</b> .....	183
	Beispiel .....	184
	Catch-all-Absicherung .....	186
	Auswertung von Exception-Objekten .....	186
	Nichts tun mit »pass« .....	187
	Fehler in verschachteltem Code .....	187
<b>10.2</b>	<b>Selbst Exceptions auslösen (»raise«)</b> .....	189
<b>10.3</b>	<b>Programmunterbrechungen abfangen</b> .....	189
	Absicherung mit »try«/»except« .....	190
	Absicherung mit »signal« .....	190
	Probleme beim Programmabbruch .....	191
<b>10.4</b>	<b>Wiederholungsfragen und Übungen</b> .....	192

<b>11</b>	<b>Objektorientierte Programmierung</b>	194
<b>11.1</b>	<b>Crashkurs</b>	195
	Funktionen helfen, Teilaufgaben zu lösen	195
	Klassen bringen Daten und Methoden zusammen	195
	Objekte sind konkrete Ausformungen von Klassen	196
	Begriffe (Nomenklatur)	197
<b>11.2</b>	<b>»Hello, Class!«</b>	199
	Konstruktor	199
	Methoden	200
	Verwendung der Klasse (»Hello, Object!«)	201
<b>11.3</b>	<b>Klassen- und Instanzvariablen</b>	203
	Namensräume	204
	Nochmals Instanz- versus Klassenvariable	205
	Type Annotations	206
	Private Instanzvariablen	207
<b>11.4</b>	<b>Methoden</b>	208
	Statische Methoden versus Instanzmethoden	208
	Methoden innerhalb einer Klasse aufrufen	210
	Getter- und Setter-Methoden (Eigenschaften)	212
<b>11.5</b>	<b>Operator Overloading</b>	214
	Gleichheitstest (»eq«-Methode)	216
	Umwandlung in Zeichenkette (»str« und »repr«)	216
	Größenvergleich (»lt« und »le«)	217
	Hash-Funktion	218
<b>11.6</b>	<b>Unveränderliche Objekte</b>	219
	Hash-Funktionen für veränderliche Objekte	219
	Veränderungen von außen verhindern	220
<b>11.7</b>	<b>Datenklassen</b>	221
	Unveränderliche Datenklassen	222



<b>11.8</b>	<b>Vererbung</b> .....	223
	Klassenzuordnung und -hierarchie feststellen .....	224
	Beispiel: Schachfiguren .....	225
	Klasse für Pferd/Springer .....	227
	Klasse für Läufer .....	228
	Mehrfachvererbung .....	229
<b>11.9</b>	<b>Wiederholungsaufgaben und Übungen</b> .....	230
<b>12</b>	<b>Module</b> .....	234
<hr/>		
<b>12.1</b>	<b>Module nutzen (»import«)</b> .....	234
	»import as« .....	235
	»from import« .....	236
<b>12.2</b>	<b>Modul-Interna</b> .....	237
	Welche Module stehen zur Verfügung? .....	237
	Wo befinden sich Moduldateien? .....	238
	Welche Module sind geladen? .....	240
<b>12.3</b>	<b>Module installieren (»pip«)</b> .....	240
	»pip« installieren oder aktualisieren .....	242
	Pakete installieren und verwalten .....	242
<b>12.4</b>	<b>Eigene Module entwickeln</b> .....	244
	Variablen über mehrere Module teilen .....	245
	Packages .....	245
	»pycache«-Verzeichnis .....	247
<b>12.5</b>	<b>Wiederholungsfragen und Übungen</b> .....	247
<b>13</b>	<b>Interna</b> .....	249
<hr/>		
<b>13.1</b>	<b>Garbage Collection</b> .....	249
<b>13.2</b>	<b>Shared References</b> .....	250

<b>13.3</b>	<b>Namensräume erforschen</b> .....	252
<b>13.4</b>	<b>Python-Compiler</b> .....	254
<b>13.5</b>	<b>Systemfunktionen</b> .....	255
	Programmparameter .....	255
	Andere Programme bzw. Kommandos aufrufen .....	256
	Warten («sleep») .....	256
	Asynchrone Programmierung .....	256
	Programm beenden .....	257
<b>13.6</b>	<b>Python 2</b> .....	258
	»print« .....	259
	»input« .....	260
	Ganze Zahlen .....	260
	Operatoren .....	260
	Zahlenbereiche («range«/»xrange«) .....	260
	Listen und andere Aufzählungen .....	261
	Unicode .....	261
	Exceptions (Fehler) .....	261
	»future«-Modul .....	261

## TEIL II Python anwenden

<b>14</b>	<b>Dateien lesen und schreiben</b> .....	265
<hr/>		
<b>14.1</b>	<b>Verzeichnisse</b> .....	265
	Das aktuelle Verzeichnis .....	266
	Das Codeverzeichnis .....	266
	Heimatverzeichnis ermitteln .....	267
	Temporäres Verzeichnis und temporäre Dateien .....	267
	Verzeichnis- und Dateinamen kombinieren und zerlegen ..	268

	Verzeichnisse erstellen, umbenennen und löschen .....	269
	Verzeichnisse durchsuchen .....	270
<b>14.2</b>	<b>Textdateien lesen und schreiben</b> .....	270
	»Hello, File!« .....	272
	Dateien mit »with/as« automatisch schließen .....	273
<b>14.3</b>	<b>Beispiel: Indexeinträge in Textdateien ändern</b> .....	275
<b>14.4</b>	<b>Standardeingabe und Standardausgabe</b> .....	277
	Beispiel .....	278
<b>14.5</b>	<b>JSON-Dateien verarbeiten</b> .....	279
	Hello, JSON! .....	279
	JSON-Dokumente erzeugen .....	280
	JSON für Objekte eigener Klassen .....	281
<b>14.6</b>	<b>XML-Dateien verarbeiten</b> .....	283
	Hello, XML! .....	284
	Beispiel: Newsfeed auswerten .....	286
<b>14.7</b>	<b>Wiederholungsfragen und Übungen</b> .....	288
<b>15</b>	<b>Netzwerkfunktionen</b> .....	289
<b>15.1</b>	<b>Download und Upload von Dateien</b> .....	289
	Zertifikatsprobleme unter macOS .....	290
	Große Dateien stückweise herunterladen .....	291
	FTP .....	292
<b>15.2</b>	<b>Mails versenden</b> .....	293
	Mail-Versand an einen externen SMTP-Server .....	294
	HTML-Mail .....	295
	Bild hinzufügen .....	295
	Weitere Mail-Varianten .....	296
<b>15.3</b>	<b>Dropbox</b> .....	296
	Dropbox-Zugang testen .....	297

	Upload .....	298
	Download .....	299
<b>15.4</b>	<b>Twitter</b> .....	300
	Tweet senden .....	300
<b>16</b>	<b>Systemadministration</b>	303
<hr/>		
<b>16.1</b>	<b>Linux-Kommandos aufrufen</b> .....	303
	Ergebnisse verarbeiten .....	304
	Kommando durch die »bash« ausführen .....	305
	Fehler beim Kommandoaufruf .....	306
<b>16.2</b>	<b>Linux-Accounts einrichten</b> .....	307
	Passwörter erzeugen .....	308
	Dateien lesen und schreiben .....	308
	Kommandos aufrufen .....	308
	Script ausführen .....	309
	Accounts wieder löschen .....	309
<b>16.3</b>	<b>MySQL/MariaDB-Datenbankzugriff</b> .....	310
	Voraussetzungen .....	311
	»Hello, Database!« .....	312
	Verbindungsaufbau .....	313
	SQL-Kommandos ausführen .....	313
	SELECT-Ergebnisse verarbeiten .....	314
	Weitere Informationen .....	314
<b>16.4</b>	<b>Mehrere gleichartige Datenbanken analysieren</b> .....	315
	SELECT-Kommando zusammenstellen .....	315
	Liste der Kundendatenbanken einlesen .....	316
	Datenbankverbindung herstellen, Abfragen durchführen ..	316
	Auswertung .....	317

<b>17</b>	<b>Raspberry Pi</b>	319
<hr/>		
17.1	<b>GPIO-Zugriff mit RPi.GPIO</b> .....	320
	GPIO-Setup .....	321
	LED ein- und ausschalten .....	323
	LED-Helligkeit steuern .....	323
	Zustand eines Tasters auswerten .....	324
	Taster entprellen .....	325
17.2	<b>LED-Ampel für die CPU-Temperatur</b> .....	327
	Code .....	328
	Test .....	330
17.3	<b>GPIO-Zugriff mit »gpiozero«</b> .....	330
	Nochmals die CPU-Temperatur visualisieren .....	331
17.4	<b>Sense HAT</b> .....	332
	»Hello, Sense HAT!« .....	333
	Display Pixel für Pixel steuern .....	333
	Pixel mit Joystick bewegen .....	335
	Pixel mit dem Gyroscope-Sensor bewegen .....	337
<b>18</b>	<b>Grafische Benutzeroberflächen</b>	339
<hr/>		
18.1	»Hello, Qt!« .....	340
18.2	<b>Buttons platzieren und nutzen</b> .....	343
	Gewöhnliche Buttons .....	343
	Auswahlhäkchen (Checkboxes) und Radio-Buttons .....	345
18.3	<b>Textfelder, Grid-Layout und Nachrichtenboxen</b> .....	347
	Grid-Layout .....	348
	Nachrichtenbox anzeigen .....	351
18.4	<b>Dateiauswahl</b> .....	352
	Beispiel .....	353

<b>18.5</b>	<b>Listenfelder</b> .....	355
	Beispiel: Länderauswahl .....	357
<b>18.6</b>	<b>Oberfläche zur GPIO-Steuerung</b> .....	359
	Programmaufbau .....	360
	Schieberegler (»QSlider«) .....	361
	Periodischer Aufruf einer Methode (»QTimer«) .....	361
<b>19</b>	<b>Grafikprogrammierung</b> .....	363
<hr/>		
<b>19.1</b>	<b>Grafische Grundfunktionen</b> .....	364
	Linien .....	367
	Rechtecke und Polygone .....	367
	Kreise und Ellipsen .....	368
	Text .....	369
	Hintergrundfarbe einstellen .....	370
<b>19.2</b>	<b>Grafik in einem Widget</b> .....	370
	Grafik in PNG-Datei speichern .....	372
<b>19.3</b>	<b>Spaß mit Sinus und Cosinus</b> .....	373
	Programmaufbau .....	374
	Einen Kreis zeichnen .....	376
	Spiralkurve zeichnen .....	376
	Linienmuster zeichnen .....	377
	Bunte, verdrehte Rechtecke auf einer Spiralkurve zeichnen	378
<b>19.4</b>	<b>Temperaturkurve anzeigen</b> .....	379
	Programmaufbau .....	380
	Daten einlesen .....	381
	Gitter zeichnen und beschriften .....	382
	Temperaturkurve zeichnen .....	384
	Verbesserungsmöglichkeiten .....	385

<b>20</b>	<b>Wissenschaftliche Anwendung</b>	386
<b>20.1</b>	<b>Anaconda, IPython und Jupyter-Notebooks</b>	387
	Installation	388
	Paketmanager conda	389
	IPython	390
	Jupyter-Notebooks	393
<b>20.2</b>	<b>NumPy</b>	395
	Matrizen erzeugen und speichern	396
	Slicing	397
	Mit Matrizen rechnen	398
<b>20.3</b>	<b>pandas</b>	400
	Serien (Series-Klasse)	401
	Tabellen (DataFrame-Klasse)	402
<b>20.4</b>	<b>SciPy</b>	406
	Beispiel 1: Minimum einer Funktion suchen	406
	Beispiel 2: Interpolationsfunktion	408
<b>20.5</b>	<b>Matplotlib</b>	409
	Hello, Matplotlib!	410
	vectorize-Funktion	412
	Diagramm und Achsen beschriften	413
	Mehrteilige Diagramme	414
	Temperaturkurve	415
	Andere Diagrammtypen	417

## Anhang

<b>A</b>	<b>Lösungen</b>	419
A.1	Kapitel 1, »Hello, World!«	419
A.2	Kapitel 2, Variablen	421
A.3	Kapitel 3, Operatoren	423
A.4	Kapitel 4, Zahlen	424
A.5	Kapitel 5, Zeichenketten	425
A.6	Kapitel 6, Datum und Uhrzeit	426
A.7	Kapitel 7, Listen, Tupel, Sets und Dictionaries	428
A.8	Kapitel 8, Verzweigungen und Schleifen	432
A.9	Kapitel 9, Funktionen	436
A.10	Kapitel 10, Umgang mit Fehlern	441
A.11	Kapitel 11, objektorientierte Programmierung	442
A.12	Kapitel 12: Module	447
A.13	Kapitel 14: Dateien lesen und schreiben	448
	Index	452



## Vorwort

Obwohl die Entwicklung von Python schon 1991 begann, stand die Sprache lange im Schatten anderer Programmiersprachen. So richtig im Mainstream angekommen ist Python erst in den vergangenen Jahren. Dabei gibt es viele Gründe für die Popularität von Python:

- ▶ Die Syntax von Python ist wesentlich einfacher als die vieler anderer Sprachen. Aus diesem Grund wird Python in Schulen und Universitäten immer öfter als erste Programmiersprache unterrichtet, noch vor Java, C# oder C++.
- ▶ Ein Designprinzip bei der Entwicklung von Python ist es, dass Programmcode möglichst gut lesbar sein soll. Unter anderem deswegen verzichtet Python auf geschwungene Klammern zur Strukturierung von Code. Stattdessen *muss* Code in Schleifen, Verzweigungen oder Funktionen eingerückt werden. Das macht unübersichtlichen »Spagetti-Code« fast unmöglich.
- ▶ Python ist eine schlanke Programmiersprache mit wenig Syntaxvarianten. Dass Python dennoch so universell für alle erdenklichen Aufgaben geeignet ist, liegt an seinem Modulkonzept: Im Internet finden Sie Tausende kostenlose Erweiterungen. Die Installation solcher Module und ihre Integration in eigenen Code sind ausgesprochen einfach.
- ▶ Python war im (natur)wissenschaftlichen Sektor schon immer beliebt – aktuell ganz besonders im Bereich der künstlichen Intelligenz. Mit keiner anderen Sprache kommen Sie mit derart wenig Code ans Ziel. Python ist ideal zum Experimentieren geeignet, wenn konkrete Ergebnisse und nicht der perfekte Code im Vordergrund stehen.
- ▶ Zur Popularität von Python hat natürlich auch der Raspberry Pi beigetragen. Trotz vieler Alternativen ist Python die bevorzugte Programmiersprache der Maker-Gemeinde.

Schon eher muss man sich darüber wundern, dass der Erfolg von Python so lange auf sich hat warten lassen. Vielleicht liegt es daran, dass Python ein echtes Open-Source-Projekt ist, hinter dem keine Firmeninteressen stehen.

### Über dieses Buch

Dieses Buch vermittelt einen kompakten Einstieg in die Programmiersprache Python:

- ▶ **Teil I** erläutert die wichtigsten Sprachkonstrukte. Hier lernen Sie Variablen, Schleifen, Funktionen und die Grundzüge objektorientierter Programmierung kennen. Übungen und Wiederholungsaufgaben (mit Lösungen im Anhang) helfen Ihnen, Ihr neu erworbenes Wissen zu festigen.
- ▶ **Teil II** zeigt in mehreren kurzen Kapiteln Anwendungsmöglichkeiten auf. Zu den Themenschwerpunkten zählen unter anderem der Umgang mit Dateien, die Systemadministration, Netzwerk-, Datenbank- und Grafikfunktionen, die Steuerung des Raspberry Pi und die Gestaltung grafischer Benutzeroberflächen.

### Viel Erfolg!

Programmieren ist eine faszinierende Beschäftigung, durchaus vergleichbar mit dem Lösen von Denksporträtseln. Die erfolgreiche Lösung einer kniffligen Programmierübung kann Ihnen Erfolgserlebnisse wie im Sport bescheren.

Naturgemäß verläuft der Lernprozess angenehmer, wenn Sie parallel zu diesem Buch jemanden haben, dem Sie Fragen stellen können – sei es ein Freund, eine Bekannte, ein Lehrer oder eine Kursvortragende.

Lassen Sie sich von meiner Begeisterung für das Programmieren anstecken, und fangen Sie an – jetzt!

Michael Kofler (<https://kofler.info>)