

Auf einen Blick

Teil I Python lernen

1	Hello, World!	25
2	Variablen	48
3	Operatoren	60
4	Zahlen	70
5	Zeichenketten	79
6	Datum und Zeit	99
7	Listen, Tupel, Sets und Dictionaries	107
8	Verzweigungen und Schleifen	131
9	Funktionen	153
10	Umgang mit Fehlern (Exceptions)	180
11	Objektorientierte Programmierung	192
12	Module	231
13	Interna	246

Teil II Python anwenden

14	Dateien lesen und schreiben	263
15	Netzwerkfunktionen	287
16	Systemadministration	301
17	Raspberry Pi	317
18	Grafische Benutzeroberflächen	337
19	Grafikprogrammierung	360
20	Wissenschaftliche Anwendung	384

Inhalt

Vorwort	21
---------------	----

TEIL I Python lernen

1	Hello, World!	25
<hr/>		
1.1	Python installieren	25
	Linux	25
	Windows	26
	macOS	28
1.2	»Hello, World!« in der Python-Shell	29
	Python in der Shell kennenlernen	29
1.3	»Hello, World!« als eigenständiges Programm	32
	VSCode samt Python-Erweiterung installieren	32
	Das minimalistische Hello-World-Programm	33
	Noch ein Beispiel	34
1.4	Thonny	37
	Thonny-Eigenheiten	38
	»Richtige« Entwicklungsumgebungen	39
1.5	Elementare Syntaxregeln	40
	Anweisungen	40
	Blockelemente	41
	»print«	42
	»input«	44

	Module und »import«	44
	Kommentare	46
1.6	Wiederholungsfragen	47
2	Variablen	48
2.1	Grundregeln	48
	Variablennamen	49
2.2	Datentypen	50
	Erkennung des Datentyps	51
	Angabe des vorgesehenen Datentyps (Type Annotation)	52
	Mutable oder immutable	52
	Veränderliche Daten kopieren	55
	Typumwandlung	56
2.3	Gültigkeitsbereich von Variablen	57
2.4	Wiederholungsfragen	59
3	Operatoren	60
3.1	Überblick	60
3.2	Details und Sonderfälle	62
	Division	62
	Zuweisung und Berechnung verbinden	62
	Zuweisung und Auswertung verbinden	63
	»in«-Operator	64
	Binäre versus logische Operatoren	64
	Short-Circuit-Evaluation	65
	Operatorhierarchie	66
	Variablen und Daten vergleichen	67
	Mehrfachvergleiche	69

»operator«-Modul	69
Operatoren selbst implementieren	69
3.3 Wiederholungsfragen	69
4 Zahlen	70
4.1 Ganze Zahlen	71
Division	71
Binäre und hexadezimale Darstellung	72
Ganze Zufallszahlen	73
4.2 Fließkommazahlen	73
Rundungsfehler	73
Zufällige Fließkommazahlen	74
4.3 Komplexe Zahlen, Brüche und Festkommazahlen	75
Komplexe Zahlen	75
Rationale Zahlen (Brüche)	76
Dezimalzahlen	76
4.4 Boolesche Werte	77
4.5 Wiederholungsfragen	78
5 Zeichenketten	79
5.1 Grundregeln	79
Zeichenketten aneinanderfügen und vervielfältigen	80
Sonderzeichen	80
Raw-Zeichenketten	82
»chr«- und »ord«-Funktion	82
5.2 Zugriff auf Teilzeichenketten	82
»IndexError«	83
Schrittweite (Stride)	83

5.3	Zeichenkettenfunktionen	84
	Eigenschaften von Zeichen(ketten) ermitteln	87
	Suchen und ersetzen	88
5.4	Zeichenketten formatieren und konvertieren	88
	Daten formatieren	89
	Formatierung mit dem Prozentoperator	89
	Formatierung mit der »format«-Methode	90
	»format«-Kurzschreibweise	92
5.5	Lokalisierung	93
	Lokalisierte Zeichenketten in Zahlen umwandeln	94
	Lokalisierungsbeispiel	95
5.6	Reguläre Ausdrücke	96
5.7	Wiederholungsfragen und Übungen	98
 6	 Datum und Zeit	 99
6.1	Zeit ermitteln und darstellen	99
	Datum aus Zeichenkette einlesen (»parsen«)	101
	Datum ohne Zeit	101
	Zeit ohne Datum	102
	Objekte für einen beliebigen Zeitpunkt	102
	Zeiten mit Zeitzone	103
6.2	Mit Zeiten rechnen	104
	Zeiten subtrahieren	105
	Laufzeit von Code messen	105
6.3	Wiederholungsfragen und Übungen	106

7	Listen, Tupel, Sets und Dictionaries	107
7.1	Listen	108
	»range«-Funktion	109
	Listen und Zeichenketten	109
	List Comprehension	110
7.2	Funktionen zur Verarbeitung von Listen	111
	»map«-Funktion	113
	»reduce«-Funktion	115
	»filter«-Funktion	116
	»any«- und »all«-Funktion	117
	Listen sortieren	117
	Zeichenketten mit deutschen Buchstaben sortieren	118
7.3	Tupel (Sequenzen)	119
	Anwendungsfälle	119
	»zip«-Funktion	120
7.4	Sets (Mengen)	121
	»set«-Methoden	122
	Mengenlehre	123
7.5	Dictionaries	123
	Dictionaries (assoziative Arrays)	123
	Verarbeitung von Dictionary-Elementen	126
	Dictionaries zusammenführen (Vereinigung/Union)	126
	»zip«-Funktion	127
7.6	Type Annotations	127
7.7	Arrays	128
7.8	Wiederholungsfragen und Übungen	130

8	Verzweigungen und Schleifen	131
8.1	»if«-Verzweigung	131
	»pass«-Schlüsselwort	133
	»if«-Kurzschreibweise	133
	Kein »switch«	134
8.2	Beispiel: Schaltjahrtest	135
8.3	»for«-Schleife	136
	Schleifen über Zahlenbereiche (»range«)	137
	Schleifen über die Zeichen einer Zeichenkette	138
	Schleifen über Listen, Tupel und Sets	138
	»enumerate«-Funktion	139
	List/Tupel/Set Comprehension	139
	Schleifen über Dictionaries, Dictionary Comprehension	140
	Generator Expressions	141
	»break«, »continue« und »else«	142
	»_« als Schleifenvariable	143
8.4	»while«-Schleife	144
8.5	Beispiele für Schleifen	145
	Summe der Zahlen von 1 bis 1000	146
	Einmaleins-Tabelle	147
	Parameter verarbeiten	148
8.6	Pattern Matching	150
8.7	Wiederholungsfragen und Übungen	151
9	Funktionen	153
9.1	Eigene Funktionen definieren	154
	»Hello, Function!«	155
	Beispiel: zufällige Zeichenkette	155
	Beispiel: Passwortgenerator	157

9.2	Lokale und globale Variablen	158
	Variablenverwaltung	158
	Globale Variablen	160
9.3	Parameter	161
	Parameter für unveränderliche Typen (immutable)	162
	Parameter für veränderliche Typen (mutable)	162
	Optionale Parameter	163
	Variable Parameteranzahl	164
	Parameterreihenfolge erzwingen	166
	Parametertyp überprüfen	166
	Type Annotations	167
9.4	Rekursion	167
	Fakultät rekursiv berechnen	168
	Unkontrollierte Rekursion	169
9.5	Lambda-Funktionen	170
9.6	Funktionale Programmierung	171
	Beispiel: Funktionsgenerator	171
	Beispiel: verschachtelter Funktionsgenerator	172
	Beispiel: Listengenerator	173
9.7	Generatoren	174
	Beispiel: Fibonacci-Zahlen	175
9.8	Wiederholungsfragen und Übungen	178
10	Umgang mit Fehlern (Exceptions)	180
10.1	Fehlerabsicherung mit »try« und »except«	181
	Beispiel	182
	Catch-all-Absicherung	184
	Auswertung von Exception-Objekten	184
	Nichts tun mit »pass«	185
	Fehler in verschachteltem Code	185

10.2	Selbst Exceptions auslösen (»raise«)	187
10.3	Programmunterbrechungen abfangen	187
	Absicherung mit »try«/»except«	188
	Absicherung mit »signal«	188
	Probleme beim Programmabbruch	189
10.4	Wiederholungsfragen und Übungen	190
11	Objektorientierte Programmierung	192
11.1	Crashkurs	193
	Funktionen helfen, Teilaufgaben zu lösen	193
	Klassen bringen Daten und Methoden zusammen	193
	Objekte sind konkrete Ausformungen von Klassen	194
	Begriffe (Nomenklatur)	195
11.2	»Hello, Class!«	197
	Konstruktor	197
	Methoden	198
	Verwendung der Klasse (»Hello, Object!«)	199
11.3	Klassen- und Instanzvariablen	201
	Namensräume	202
	Nochmals Instanz- versus Klassenvariable	203
	Type Annotations	204
	Private Instanzvariablen	205
	Ad-hoc-Objekte	206
11.4	Methoden	207
	Statische Methoden versus Instanzmethoden	207
	Methoden innerhalb einer Klasse aufrufen	209
	Getter- und Setter-Methoden (Eigenschaften)	210

11.5	Operator Overloading	213
	Gleichheitstest (»eq«-Methode)	214
	Umwandlung in Zeichenkette (»str« und »repr«)	215
	Größenvergleich (»lt« und »le«)	216
	Hash-Funktion	217
11.6	Datenklassen	217
	Unveränderliche Datenklassen	218
11.7	Vererbung	219
	Klassenzuordnung und -hierarchie feststellen	221
	Beispiel: Schachfiguren	221
	Klasse für Pferd/Springer	223
	Klasse für Läufer	225
	Mehrfachvererbung	226
11.8	Wiederholungsaufgaben und Übungen	227
12	Module	231
12.1	Module nutzen (»import«)	231
	»import as«	232
	»from import«	233
12.2	Modul-Interna	234
	Welche Module stehen zur Verfügung?	234
	Wo befinden sich Moduldateien?	235
	Welche Module sind geladen?	237
12.3	Module installieren (»pip«)	237
	»pip« installieren oder aktualisieren	238
	Pakete installieren und verwalten	239
	»pip« versus Linux-Pakete versus Virtual Environments	240

12.4	Eigene Module entwickeln	241
	Variablen über mehrere Module teilen	241
	Die »name«-Variable	242
	Packages	242
	»pycache«-Verzeichnis	244
12.5	Wiederholungsfragen und Übungen	244
 13	 Interna	 246
13.1	Python-Scripts ausführen	246
	Was ist ein Script?	246
	Scripts ausführen	247
	Scripts unter Windows ausführen	248
	Scripts unter Linux und macOS ausführen	249
	Windows versus Linux/macOS	249
13.2	Garbage Collection	252
13.3	Shared References	253
13.4	Namensräume erforschen	255
13.5	Python-Compiler	256
13.6	Systemfunktionen	257
	Programmparameter	257
	Andere Programme bzw. Kommandos aufrufen	258
	Warten (»sleep«)	258
	Asynchrone Programmierung	259
	Programm beenden	260
13.7	Wiederholungsfragen	260

TEIL II Python anwenden

14	Dateien lesen und schreiben	263
14.1	Verzeichnisse	263
	Das aktuelle Verzeichnis	264
	Das Codeverzeichnis	264
	Heimatverzeichnis ermitteln	265
	Temporäres Verzeichnis und temporäre Dateien	265
	Verzeichnis- und Dateinamen kombinieren und zerlegen	266
	Verzeichnisse erstellen, umbenennen und löschen	267
	Verzeichnisse durchsuchen	268
14.2	Textdateien lesen und schreiben	269
	»Hello, File!«	270
	Dateien mit »with/as« automatisch schließen	271
14.3	Beispiel: Indexeinträge in Textdateien ändern	273
14.4	Standardeingabe und Standardausgabe	275
	Beispiel	276
14.5	JSON-Dateien verarbeiten	277
	»Hello, JSON!«	277
	JSON-Dokumente erzeugen	278
	JSON für Objekte eigener Klassen	279
14.6	XML-Dateien verarbeiten	281
	»Hello, XML!«	282
	Beispiel: Newsfeed auswerten	284
14.7	Wiederholungsfragen und Übungen	286

15	Netzwerkfunktionen	287
<hr/>		
15.1	Download und Upload von Dateien	287
	Zertifikatsprobleme unter macOS	288
	Große Dateien stückweise herunterladen	289
	FTP	290
15.2	Mails versenden	291
	Mail-Versand an einen externen SMTP-Server	292
	HTML-Mail	293
	Bild hinzufügen	294
	Weitere Mail-Varianten	294
15.3	Dropbox	295
	Dropbox-Zugang testen	295
	Upload	296
	Download	297
15.4	Twitter	298
	Tweet senden	298
16	Systemadministration	301
<hr/>		
16.1	Linux-Kommandos aufrufen	301
	Ergebnisse verarbeiten	302
	Kommando durch die »bash« ausführen	303
	Fehler beim Kommandoaufruf	304
16.2	Linux-Accounts einrichten	304
	Passwörter erzeugen	306
	Dateien lesen und schreiben	306
	Kommandos aufrufen	306
	Script ausführen	307
	Accounts wieder löschen	307

16.3	MySQL/MariaDB-Datenbankzugriff	308
	Voraussetzungen	309
	»Hello, Database!«	310
	Verbindungsaufbau	311
	SQL-Kommandos ausführen	311
	SELECT-Ergebnisse verarbeiten	312
	Weitere Informationen	312
16.4	Mehrere gleichartige Datenbanken analysieren	313
	SELECT-Kommando zusammenstellen	313
	Liste der Kundendatenbanken einlesen	314
	Datenbankverbindung herstellen, Abfragen durchführen ..	314
	Auswertung	315
17	Raspberry Pi	317

17.1	GPIO-Zugriff mit RPi.GPIO	318
	GPIO-Setup	319
	LED ein- und ausschalten	321
	LED-Helligkeit steuern	321
	Zustand eines Tasters auswerten	322
	Taster entprellen	323
17.2	LED-Ampel für die CPU-Temperatur	325
	Code	326
	Test	328
17.3	GPIO-Zugriff mit »gpiozero«	328
	Nochmals die CPU-Temperatur visualisieren	329
17.4	Sense HAT	330
	»Hello, Sense HAT!«	331
	Das Display Pixel für Pixel steuern	332
	Pixel mit Joystick bewegen	333
	Pixel mit dem Gyroskop-Sensor bewegen	335

18	Grafische Benutzeroberflächen	337
18.1	»Hello, Qt!«	338
18.2	Buttons platzieren und nutzen	340
	Gewöhnliche Buttons	340
	Auswahlhäkchen (Checkboxes) und Radio-Buttons	342
18.3	Textfelder, Grid-Layout und Nachrichtenboxen	345
	Grid-Layout	346
	Nachrichtenbox anzeigen	348
18.4	Dateiauswahl	349
	Beispiel	350
18.5	Listenfelder	352
	Beispiel: Länderauswahl	353
18.6	Oberfläche zur GPIO-Steuerung	356
	Programmaufbau	356
	Schiebereglern (»QSlider«)	357
	Periodischer Aufruf einer Methode (»QTimer«)	358
19	Grafikprogrammierung	360
19.1	Grafische Grundfunktionen	361
	Linien	364
	Rechtecke und Polygone	364
	Kreise und Ellipsen	365
	Text	366
	Hintergrundfarbe einstellen	367
19.2	Grafik in einem Widget	367
	Grafik in PNG-Datei speichern	369
19.3	Spaß mit Sinus und Cosinus	370
	Programmaufbau	371

Einen Kreis zeichnen	373
Spiralkurve zeichnen	374
Linienmuster zeichnen	374
Bunte, verdrehte Rechtecke auf einer Spiralkurve zeichnen	375
19.4 Temperaturkurve anzeigen	377
Programmaufbau	377
Daten einlesen	378
Gitter zeichnen und beschriften	379
Temperaturkurve zeichnen	382
Verbesserungsmöglichkeiten	383
 20 Wissenschaftliche Anwendung	 384
20.1 Anaconda, IPython und Jupyter-Notebooks	384
Installation	385
Paketmanager »conda«	387
IPython	388
Jupyter-Notebooks	391
Traditionelle Python-Scripts	393
20.2 NumPy	394
Matrizen erzeugen und speichern	394
Slicing	396
Mit Matrizen rechnen	397
20.3 pandas	399
Serien (»Series«-Klasse)	400
Tabellen (»DataFrame«-Klasse)	401
20.4 SciPy	404
Beispiel 1: Minimum einer Funktion suchen	405
Beispiel 2: Interpolationsfunktion	406

20.5	Matplotlib	408
	»Hello, Matplotlib!«	408
	»vectorize«-Funktion	411
	Diagramm und Achsen beschriften	412
	Mehrteilige Diagramme	413
	Temperaturkurve	414
	Andere Diagrammtypen	416
 A	 Lösungen	 417
A.1	Kapitel 1: »Hello, World!«	417
A.2	Kapitel 2: Variablen	418
A.3	Kapitel 3: Operatoren	420
A.4	Kapitel 4: Zahlen	421
A.5	Kapitel 5: Zeichenketten	422
A.6	Kapitel 6: Datum und Uhrzeit	424
A.7	Kapitel 7: Listen, Tupel, Sets und Dictionaries	425
A.8	Kapitel 8: Verzweigungen und Schleifen	429
A.9	Kapitel 9: Funktionen	433
A.10	Kapitel 10: Umgang mit Fehlern	439
A.11	Kapitel 11: Objektorientierte Programmierung	440
A.12	Kapitel 12: Module	445
A.13	Kapitel 13: Interna	446
A.14	Kapitel 14: Dateien lesen und schreiben	447
	Index	451