1 Einleitung und Grundlagen – Bevor es richtig losgeht 1
1.1 Was behandeln wir in dem einleitenden Kapitel? 1
1.2 Das Ziel des Buchs 1
1.3 Was sollten Sie bereits können? 2
1.4 Was ist Python? 2
1.4.1 Das Ziel von Python 3
1.4.2 Was umfasst Python? 3
1.4.3 Die verschiedenen Python-Paradigma 4
1.5 Was benötigen Sie zum Arbeiten mit dem Buch? 4
1.5.1 Hardware und Betriebssystem 4
1.5.2 Die Python-Version 4
1.5.3 Python laden und installieren 5

2 Erste Beispiele – Der Sprung ins kalte Wasser 19
2.1 Was behandeln wir in diesem Kapitel? 19
2.2 Der Interaktivmodus – die Kommandozeile von Python 19
2.2.1 Das Prompt 21
2.2.2 Der Hilfemodus in der Kommandozeile 25
2.3 Anweisungen in (echten) Quelltext auslagern 27
2.4 IDLE & Co 28
2.4.1 Weitere IDEs und Editoren für Python 31

3 Built-in-Functions – Modularisierung durch Unterprogramme 33
3.1 Was behandeln wir in diesem Kapitel? 33
3.2 Was sind Funktionen im Allgemeinen? 33
3.3 Built-in Functions 34
3.3.1 Hilfe zu Built-in-Functions im Hilfemodus 34
3.3.2 Hilfe zu Built-in-Functions im Editormodus 35
3.3.3 Die print()-Funktion 36
3.3.4 Die input()-Funktion 41
3.3.5 Eine kurze Übersicht aller Built-in Functions 42

4 Grundlegende Begriffe – Kommentare, SheBang und Strukturanalysen ... 47
4.1 Was behandeln wir in diesem Kapitel? ... 47
4.2 Token und Parser ... 47
4.2.1 Zerlegen von Quelltext ... 48
4.3 Kommentare ... 49
4.3.1 Kommentare in Python ... 50
4.4 SheBang und eine Python-Datei direkt ausführen ... 52
4.4.1 SheBang als besonderer Kommentar ... 52

5 Anweisungen – Dem Computer Befehle geben ... 53
5.1 Was behandeln wir in diesem Kapitel? ... 53
5.2 Was sind Anweisungen? ... 53
5.2.1 Eine Frage der Reihenfolge ... 53
5.3 Anweisungsarten ... 54
5.3.1 Blockanweisung ... 54
5.3.2 Kontrollflussanweisungen ... 55
5.3.3 Deklarationsanweisung ... 55
5.3.4 Ausdrucksanweisung ... 55
5.3.5 Die leere Anweisung *pass* ... 56

6 Datentypen, Variablen und Literale – Die Art der Information ... 57
6.1 Was behandeln wir in diesem Kapitel? ... 57
6.2 Variablen ... 57
6.2.1 Variablen deklarieren ... 57
6.2.2 Variablen im Quellcode verwenden ... 59
6.3 Die Datentypen in Python ... 59
6.3.1 Lose Typisierung und Typumwandlung in Python ... 59
6.3.2 Die Python-Datentypen ... 61
6.3.3 Zahlen – *int*, *float* und *complex* ... 63
6.3.4 Zeichenliterale ... 67
6.4 Den Datentyp bestimmen und umwandeln ... 68
6.4.1 Den Datentyp mit type() dynamisch bestimmen ... 68
6.4.2 Implizite und explizite Typumwandlung ... 69

7 Ausdrücke, Operatoren und Operanden – Die Verarbeitung von Daten ... 71
7.1 Was behandeln wir in diesem Kapitel? ... 71
7.2 Ausdrücke ... 71
7.3 Operationen mit Operatoren und Operanden ... 72
7.3.1 Arithmetische Operatoren ... 72
7.3.2 Der String-Verkettungsoperator ... 75
7.3.3 Zuweisungsoperatoren ... 75
7.3.4 Boolesche Operatoren (Vergleichsoperatoren) ... 77
7.3.5 Logische Operatoren ... 78

7.3.6 Die Membership-Operatoren ... 79
7.3.7 Identitätsoperatoren ... 80
7.3.8 Bitweise Operatoren ... 80
7.4 Operatorvorrang und Ausdrucksbewertung ... 85
7.4.1 Die Priorität der Python-Operatoren ... 85
7.4.2 Bewertung von Ausdrücken ... 85

8 Kontrollstrukturen – Die Steuerung des Programmflusses ... 87
8.1 Was behandeln wir in diesem Kapitel? ... 87
8.2 Was sind Kontrollstrukturen? ... 87
8.3 Die Kontrollstrukturen in Python ... 88
8.3.1 Entscheidungsanweisungen ... 88
8.3.2 Iterationsanweisungen ... 94
8.3.3 Sprunganweisungen ... 96

9 Funktionen in Python – Modularisierung mit „Unterprogrammen" ... 99
9.1 Was behandeln wir in diesem Kapitel? ... 99
9.2 In Python eigene Funktionen deklarieren – das Schlüsselwort def ... 99
9.2.1 Übergabewerte ... 100
9.2.2 Rückgabewerte ... 101
9.3 Funktionen aufrufen ... 101
9.3.1 Stehen in Python global deklarierte Variablen in der Funktion zur Verfügung? ... 103
9.4 Rekursion ... 104
9.5 Innere Funktionen – Closures ... 107
9.6 Lambda-Ausdrücke und anonyme Funktionen ... 108
9.6.1 Lambda-Funktionen verwenden ... 109
9.7 Besondere Situationen bei Funktionen in Python ... 110
9.7.1 Lokale Variablen in Funktionen ... 110
9.7.2 Die Anzahl der Parameter passen nicht ... 111
9.7.3 Unerreichbarer Code ... 114

10 Sequenzielle Datenstrukturen – Mehrere Informationen gemeinsam verwalten ... 115
10.1 Was behandeln wir in diesem Kapitel? ... 115
10.2 Was sind sequenzielle Datenstrukturen? ... 115
10.2.1 Zeichenketten als sequenzielle Ansammlung von Zeichenliteralen ... 115
10.2.2 Arrays ... 116
10.3 Tupel ... 116
10.3.1 Verschachtelte Tupel ... 117
10.3.2 Tupel und der Membership-Operator ... 118
10.3.3 Einzelne Einträge in Tupel ansprechen ... 120
10.3.4 Die Anzahl der Elemente in einem Tupel bestimmen ... 124

10.4 Dynamische Listen ... 124
10.4.1 Warum Listen und Tupel? ... 125
10.5 Methoden für Listen ... 125
10.5.1 Verschiedene Listenmethoden in einem Beispiel ... 126
10.5.2 Einen Stack erzeugen ... 127
10.5.3 Eine Queue mit einer Liste erzeugen ... 128
10.6 Dictionaries ... 129
10.6.1 Spezielle Methoden für Dictionaries ... 130
10.6.2 Ein Beispiel zum allgemeinen Umgang mit Dictionaries ... 131
10.6.3 Ein Dictionary aktualisieren oder erweitern ... 132
10.6.4 Iteration über ein Dictionary ... 133
10.7 Mengen ... 134
10.7.1 Vereinfachte Notation ... 134
10.7.2 Operationen auf „set"-Objekten ... 135
10.8 Operatoren bei sequenziellen Datentypen ... 136
10.8.1 Der Plusoperator ... 137
10.8.2 Multiplikationen mit sequenziellen Datentypen ... 137
10.8.3 Inhalt überprüfen ... 138
10.9 Über sequenzielle Strukturen iterieren ... 140

11 Objektorientierte Programmierung in Python – Klassen, Objekte, Eigenschaften und Methoden ... 143
11.1 Was behandeln wir in diesem Kapitel? ... 143
11.2 Hintergründe der OOP ... 143
11.2.1 Ziele der OOP – Wiederverwendbarkeit und bessere Softwarequalität ... 144
11.2.2 Kernkonzepte der Objektorientierung ... 145
11.3 Klassen ... 146
11.3.1 Klassen als Baupläne, Konstruktoren und Destruktoren ... 147
11.3.2 Der konkrete Klassenaufbau in Python ... 147
11.3.3 Die konkrete Instanziierung ... 148
11.4 Details zu Objekten ... 150
11.4.1 OO-Philosophie als Abstraktion ... 151
11.4.2 Instanzelemente versus Klassenelemente ... 151
11.4.3 Der Aufbau von Objekten in Python ... 152
11.4.4 Zugriff auf Objektbestandteile ... 152
11.4.5 Von Grund auf objektorientiert ... 157
11.5 Klassenmethoden und statische Methoden ... 157
11.5.1 Klassenmethoden ... 158
11.5.2 Statische Methoden ... 159
11.6 Eine Frage der Sichtbarkeit ... 161
11.6.1 Ein Beispiel für den Zugriff auf ein öffentliches Element ... 161

11.6.2 Ein Beispiel für den versuchten Zugriff auf ein privates Element von außen ... 162
11.6.3 Getter und Setter ... 162
11.7 Ein Objekt löschen ... 165
11.7.1 Ein Beispiel für das Redefinieren des Destruktors ... 165
11.8 Ein paar besondere OO-Techniken ... 166
11.8.1 Eine To-String-Funktionalität bereitstellen – __str__ ... 166
11.8.2 Objekte dynamisch erweitern, das Dictionary __dict__ und Slots ... 167
11.8.3 Dynamische Erzeugung von Klassen, Metaklassen und die Klasse type ... 169
11.9 Vererbung ... 170
11.9.1 Grundlagentheorie zur Vererbung ... 170
11.9.2 Umsetzung von Vererbung in Python ... 172
11.9.3 Mehrfachvererbung in Python ... 172
11.9.4 Polymorphie über Überschreiben und Überladen ... 173
11.10 Was ist mit Schnittstellen und abstrakten Klassen in Python? ... 175
11.10.1 Abstrakte Superklassen ... 176
11.10.2 Was ist im Allgemeinen eine Schnittstelle? ... 176
11.11 Module und Pakete ... 177
11.11.1 Die import-Anweisung ... 177
11.11.2 Importieren mit from ... 178
11.11.3 Pakete ... 178
11.11.4 Das Python-API ... 180

12 Exceptionhandling – Ausnahmsweise ... 181
12.1 Was behandeln wir in diesem Kapitel? ... 181
12.2 Was sind Ausnahmen? ... 181
12.3 Warum ein Ausnahmekonzept? ... 182
12.4 Konkrete Ausnahmebehandlung in Python ... 183
12.4.1 Ein erstes Beispiel mit einfacher Ausnahmebehandlung ... 183
12.4.2 Mehrere Ausnahme-Blöcke ... 184
12.4.3 Die finally-Anweisung ... 184
12.4.4 Praktische Beispiele ... 185
12.5 Standard Exceptions ... 188
12.5.1 Die Reihenfolge bei mehreren Ausnahmetypen ... 189
12.6 Der else-Block ... 189
12.7 Ausnahmeobjekte auswerten ... 190
12.8 Werfen von Ausnahmen mit raise ... 191
12.9 Eigene Ausnahmeklassen definieren ... 193
12.10 Die assert-Anweisung ... 193

13 String-Verarbeitung in Python – Programmierte Textverarbeitung 195
13.1 Was behandeln wir in diesem Kapitel? 195
13.2 Typische String-Verarbeitungstechniken 196
13.3 Das konkrete Vorgehen in Python 196
13.3.1 String-Konstanten und die Format Specification Mini-Language 196
13.3.2 String-Funktionen 197
13.3.3 String-Methoden 197
13.4 Umgang mit regulären Ausdrücken 198
13.4.1 Was sind allgemein reguläre Ausdrücke? 198
13.4.2 Wo setzt man reguläre Ausdrücke ein? 199
13.4.3 Details zu Pattern 200
13.4.4 Optionen für die Häufigkeit 204
13.4.5 Die Umsetzung von regulären Ausdrücken in Python – das Modul re 205
13.4.6 Die Match-Objekte 206
13.4.7 Ein paar Beispiele mit regulären Ausdrücken 207

14 Datei-, Datenträger- und Datenbankzugriffe – Dauerhafte Daten 211
14.1 Was behandeln wir in diesem Kapitel? 211
14.2 Datenströme für die Ein- und Ausgabe 211
14.2.1 Das Öffnen und Schließen einer Datei 212
14.2.2 Schreiben in eine Datei 213
14.2.3 Auslesen aus einer Datei 213
14.2.4 Lese- und Schreibvorgänge absichern 215
14.3 Allgemeine Datei- und Verzeichnisoperationen 215
14.4 Objekte serialisieren und deserialisieren 217
14.4.1 Mit dump() den Objektzustand persistent machen 218
14.4.2 Mit load() den Objektzustand reproduzieren 218
14.5 Datenbankzugriffe 219
14.5.1 Was ist SQLite? 219
14.5.2 Zugriff auf SQLite in Python 219
14.5.3 Ein konkretes Datenbankbeispiel 221

15 Umgang mit Datum und Zeit – Terminsachen 225
15.1 Was behandeln wir in diesem Kapitel? 225
15.2 Allgemeines zum Umgang mit Datum und Zeit 225
15.3 Die Python-Module 226
15.4 Typische Beispiele für Operationen mit Zeit und Datum 226
15.4.1 Das aktuelle Systemdatum des Computers auslesen 227
15.4.2 Ein beliebiges Datumsobjekt erstellen 227

16 Grafische Oberflächen (GUI) mit Python – Das Modul tkinter als GUI-Framework 233
16.1 Was behandeln wir in diesem Kapitel? 233
16.2 Hintergrundinformationen zu modernen grafischen Oberflächen 233
16.3 Konkrete GUI-Konzepte in Python und das Modul tkinter 234
16.3.1 Ein Fenster vom Typ TK als Basis jeder GUI-Applikation 234
16.3.2 Der übliche OO-Ansatz 234
16.3.3 Die Geometry Manager 235
16.3.4 Wichtige GUI-Elemente 242
16.4 Die Ereignisbehandlung 242
16.4.1 Die konkrete Ereignisbehandlung in Python 243
16.5 Eine grafische Datenbankapplikation 245

Stichwortverzeichnis 249