

# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>8</b>
1.1 Python: eine einfach zu erlernende Programmiersprache.....	8
1.2 Die Ziele bei der Entwicklung von Python .....	10
1.3 Die Entwicklungsgeschichte .....	11
1.4 Eine interpretierte Programmiersprache .....	12
<b>2. Die Vorbereitungsmaßnahmen</b>	<b>15</b>
2.1 Den Python-Interpreter installieren .....	16
2.2 Einen Texteditor für die Erstellung des Codes .....	19
<b>3. Interaktive Interpretation: ideal für den ersten Kontakt mit Python</b>	<b>22</b>
3.1 Den Python-Prompt aufrufen.....	23
3.2 Erste Befehle ausprobieren .....	24
<b>4. Ein Python-Programm in eine eigene Datei schreiben</b>	<b>28</b>
4.1 Ein Programm für eine einfache Textausgabe .....	28
4.2 Die Ausführung im Python-Interpreter .....	30
4.3 Kommentare: hilfreich für das Verständnis des Programms.....	32
4.4 Übungsaufgabe: eigene Inhalte zum Programm hinzufügen .....	34
<b>5. Variablen: unverzichtbar für die Programmierung mit Python</b>	<b>39</b>
5.1 Die Aufgabe von Variablen in einem Computerprogramm.....	39
5.2 Variablen in Python verwenden.....	41
5.3 Den Wert einer Variablen durch eine Eingabe des Nutzers festlegen.....	44
5.4 Dynamische Typisierung: viele Freiheiten bei der Nutzung von Variablen .....	48
5.5 Datentypen sind auch in Python von Bedeutung .....	50
5.6 Übungsaufgabe: Mit Variablen arbeiten .....	54
<b>6. Datenstrukturen in Python</b>	<b>57</b>
6.1 Listen: Mehrere Informationen zusammenfassen.....	59
6.2 Dictionaries: Zugriff über einen Schlüsselbegriff.....	63
6.3 Tupel: unveränderliche Daten.....	66
6.4 Übungsaufgabe: Mit Datenstrukturen arbeiten .....	70

<b>7. Entscheidungen im Programm treffen</b>	<b>75</b>
7.1 Der Schlüsselbegriff if.....	75
7.2 Vergleiche: wichtig für das Aufstellen der Bedingung.....	77
7.3 Die Verknüpfung mehrerer Bedingungen.....	82
7.4 Mit else und elif weitere Alternativen hinzufügen.....	85
7.5 Übungsaufgabe: eigene Abfragen erstellen.....	89
<b>8. Schleifen für die Wiederholung bestimmter Programmteile</b>	<b>96</b>
8.1 Die while-Schleife: der grundlegende Schleifentyp.....	97
8.2 Die for-Schleife: ein mächtiges Instrument in Python.....	100
8.3 break und continue: weitere Werkzeuge für die Steuerung von Schleifen.....	105
8.4 Übungsaufgabe: mit verschiedenen Schleifen arbeiten.....	107
<b>9. Funktionen in Python</b>	<b>113</b>
9.1 Die Vorteile einer Funktion.....	113
9.2 Eine Funktion selbst erstellen.....	114
9.3 Argumente für Funktionen verwenden.....	116
9.4 Einen Rückgabewert verwenden.....	121
9.5 Funktionen in einer eigenen Datei abspeichern.....	125
9.6 Übungsaufgabe: Funktionen selbst gestalten.....	127
<b>10. Mit Modulen aus der Standardbibliothek arbeiten</b>	<b>132</b>
10.1 Was ist die Standardbibliothek und welche Module enthält sie?.....	132
10.2 Die Referenz für die Standardbibliothek.....	134
10.3 Beispiel für ein häufig verwendetes Modul: math.....	136
10.4 Übungsaufgabe: mit der Standardbibliothek arbeiten.....	139
<b>11. Objektorientierung in Python</b>	<b>143</b>
11.1 Objektorientierung: Was ist das?.....	143
11.2 Klassen: die Grundlage der objektorientierten Programmierung.....	146
11.3 Objekte: Instanzen der Klassen.....	149
11.4 Die Kapselung der Daten.....	152
11.5 Methoden: Funktionen für Objekte.....	154
11.6 Klassen- und Objektvariablen.....	159
11.7 Vererbung: ein grundlegendes Prinzip der objektorientierten Programmierung...	163
11.8 Übungsaufgabe: mit Objekten arbeiten.....	166

<b>12. Die Behandlung von Fehlern und Ausnahmen in Python</b>	<b>173</b>
12.1 Warum ist es wichtig, Fehler und Ausnahmen zu behandeln? .....	173
12.2 try und except: So werden Ausnahmen behandelt .....	175
12.3 finally: die Ausnahmebehandlung abschließen .....	181
12.4 Selbst definierte Ausnahmen festlegen .....	184
12.5 Übungsaufgabe: Programme mit Ausnahmebehandlung schreiben .....	187
<b>13. Dateien für die Datenspeicherung verwenden</b>	<b>192</b>
13.1 Daten dauerhaft abspeichern: verschiedene Möglichkeiten .....	192
13.2 Daten in die Datei schreiben .....	194
13.3 Daten aus der Datei auslesen .....	196
13.4 Übungsaufgabe: mit Dateien für die Datenspeicherung arbeiten .....	203
<b>14. Grafische Benutzeroberflächen mit Tkinter erzeugen</b>	<b>208</b>
14.1 Ein erstes einfaches Fenster erstellen .....	209
14.2 Buttons mit Funktionen hinzufügen .....	214
14.3 Das Layout der Fenster .....	219
14.4 Weitere Elemente für die Gestaltung der Fenster .....	224
14.5 Übungsaufgabe: Programme mit Fenstern selbst gestalten .....	228
<b>15. Anwendungsbeispiel: Verwaltungsprogramm für Gebrauchtwagenhändler</b>	<b>234</b>
15.1 Die Struktur des Programms festlegen .....	235
15.2 Das Sortiment anzeigen .....	238
15.3 Ein neues Auto zum Sortiment hinzufügen .....	243
15.4 Ein Auto verkaufen .....	248
15.5 Den Preis eines Fahrzeugs anpassen .....	253
15.6 Letzte Anpassungen am Hauptprogramm .....	258
15.7 Ausblick .....	259
<b>16. Anhang: Index</b>	<b>262</b>