

Inhalt

Einleitung	11
1. Python: eine einfach zu erlernende Programmiersprache	13
1.1 Die Ziele bei der Entwicklung von Python	14
1.2 Die Entwicklungsgeschichte	15
1.3 Eine interpretierte Programmiersprache	17
2. Die Vorbereitungsmaßnahmen	21
2.1 Den Python-Interpreter installieren	22
2.2 Ein Texteditor für die Erstellung des Codes	25
3. Interaktive Interpretation: ideal für den ersten Kontakt mit Python	29
3.1 Den Python-Prompt aufrufen	30
3.2 Erste Befehle ausprobieren	31
4. Ein Python-Programm in eine eigene Datei schreiben	35
4.1 Ein Programm für eine einfache Textausgabe	35
4.2 Die Ausführung im Python-Interpreter	37
4.3 Kommentare: hilfreich für das Verständnis des Programms	39
4.4 Übungsaufgabe: eigene Inhalte zum Programm hinzufügen	42

5. Variablen: unverzichtbar für die Programmierung mit Python	47
5.1 Die Aufgabe von Variablen in einem Computerprogramm	47
5.2 Variablen in Python verwenden	49
5.3 Den Wert einer Variablen durch eine Eingabe des Nutzers festlegen	52
5.4 Dynamische Typisierung: viele Freiheiten bei der Nutzung von Variablen	56
5.5 Datentypen sind auch in Python von Bedeutung	58
5.6 Übungsaufgabe: mit Variablen arbeiten	62
6. Datenstrukturen in Python	65
6.1 Listen: mehrere Informationen zusammenfassen	67
6.2 Dictionaries: Zugriff über einen Schlüsselbegriff	72
6.3 Tupel: unveränderliche Daten	75
6.4 Übungsaufgabe: mit Datenstrukturen arbeiten	78
7. Entscheidungen im Programm treffen	83
7.1 Der Schlüsselbegriff if	83
7.2 Vergleiche: wichtig für das Aufstellen der Bedingung	86
7.3 Die Verknüpfung mehrerer Bedingungen	90
7.4 Mit else und elif weitere Alternativen hinzufügen	92
7.5 Übungsaufgabe: eigene Abfragen erstellen	96
8. Schleifen für die Wiederholung bestimmter Programmteile	101
8.1 Die while-Schleife: der grundlegende Schleifentyp	102
8.2 Die for-Schleife: ein mächtiges Instrument in Python	105

8.3	break und continue: weitere Werkzeuge für die Steuerung von Schleifen	109
8.4	Übungsaufgabe: mit verschiedenen Schleifen arbeiten	112
9.	Funktionen in Python.	117
9.1	Die Vorteile einer Funktion	117
9.2	Eine Funktion selbst erstellen.	118
9.3	Argumente für Funktionen verwenden.	120
9.4	Einen Rückgabewert verwenden	125
9.5	Funktionen in einer eigenen Datei abspeichern.	130
9.6	Übungsaufgabe: Funktionen selbst gestalten.	133
10.	Mit Modulen aus der Standardbibliothek arbeiten	137
10.1	Was ist die Standardbibliothek und welche Module enthält sie?	137
10.2	Die Referenz für die Standardbibliothek	140
10.3	Beispiel für ein häufig verwendetes Modul: math	141
10.4	Übungsaufgabe: mit der Standardbibliothek arbeiten ..	145
11.	Objektorientierung in Python.	149
11.1	Objektorientierung: Was ist das?	149
11.2	Klassen: die Grundlage der objektorientierten Programmierung	152
11.3	Objekte: Instanzen der Klassen	156
11.4	Die Kapselung der Daten	159
11.5	Methoden: Funktionen für Objekte	162
11.6	Klassen- und Objektvariablen.	167

11.7	Vererbung: ein grundlegendes Prinzip der objektorientierten Programmierung.....	171
11.8	Übungsaufgabe: mit Objekten arbeiten.....	174

12. Die Behandlung von Fehlern und Ausnahmen in Python 181

12.1	Warum ist es wichtig, Fehler und Ausnahmen zu behandeln?	182
12.2	try und except: So werden Ausnahmen behandelt	183
12.3	finally: die Ausnahmebehandlung abschließen	189
12.4	Selbst definierte Ausnahmen festlegen	191
12.5	Übungsaufgabe: Programme mit Ausnahmebehandlung schreiben	195

13. Daten in Dateien dauerhaft abspeichern 199

13.1	Unterschiedliche Möglichkeiten für die dauerhafte Datenspeicherung.....	199
13.2	Daten in die Datei schreiben.....	201
13.3	Daten aus der Datei auslesen	205
13.4	XML-Dateien verwenden	207
13.5	Übungsaufgabe: mit Dateien für die Datenspeicherung arbeiten.....	217

14. Datenbanken für eine sichere und effiziente Datenspeicherung verwenden 221

14.1	Was ist eine Datenbank?	221
14.2	Ein passendes Datenbanksystem für Python-Programme auswählen	224
14.3	Eine SQLite-Datenbank erstellen und Tabellen anlegen	226

14.4	Daten in die Tabellen einfügen.....	231
14.5	Daten verändern und löschen	236
14.6	Informationen aus der Datenbank abrufen.....	240
14.7	Übungsaufgabe: Programme mit SQLite erstellen.....	244
15.	Grafische Benutzeroberflächen mit Tkinter erzeugen.....	251
15.1	Ein erstes einfaches Fenster erstellen.....	252
15.2	Buttons mit Funktionen hinzufügen	257
15.3	Das Layout der Fenster.....	262
15.4	Weitere Elemente für die Gestaltung der Fenster.....	267
15.5	Übungsaufgabe: Programme mit Fenstern selbst gestalten	272
16.	Anwendungsbeispiel: das Sortiment eines Baumarkts verwalten.....	277
16.1	Die grundlegenden Strukturen des Programms.....	278
16.2	Neue Produkte zum Sortiment hinzufügen.....	282
16.3	Die Anzeige des Sortiments.....	291
16.4	Ein Produkt verkaufen	298
16.5	Preise anpassen.....	307
16.6	Die Änderungen am Hauptprogramm im Überblick	312
16.7	Ausblick	313
	Index	315