

Bevor Sie beginnen ...	4	5 Grundlegende Sprachelemente	63
1. Grundlagen zu Computing, Programmen und Programmiersprachen	6	5.1 Syntax und Semantik	63
1.1 Soft- und Hardware	6	5.2 Grundlegende Elemente einer Sprache	64
1.2 Grundlagen zu Programmen	7	5.3 Standarddatentypen (elementare Datentypen)	66
1.3 Computing und Computational Thinking	8	5.4 Literale für primitive Datentypen	69
1.4 Qualitätskriterien und Dokumentation	10	5.5 Variablen und Konstanten	70
1.5 Warum Programmieren?	15	5.6 Operatoren	76
1.6 Klassifizierung von Programmiersprachen	15	5.7 Ausdrücke und Operatorenrangfolgen	82
1.7 Die Klassifizierung nach Generationen	16	5.8 Übungen	85
1.8 Prozedurale und funktionale Programmiersprachen	19	6 Kontrollstrukturen	86
1.9 Objektorientierte Programmiersprachen	21	6.1 Anweisungen und Folgen	86
1.10 Hybride Programmiersprachen und Skriptsprachen	23	6.2 Bedingungen und Kontrollstrukturen	89
1.11 Logische Programmiersprachen	26	6.3 Grundlagen zu Verzweigungen	90
1.12 Erziehungorientierte Programmiersprachen und Minisprachen	26	6.4 Bedingte Anweisung	90
1.13 Webprogrammierung und mobile Apps	28	6.5 Verzweigung	91
1.14 Übungen	30	6.6 Geschachtelte Verzweigung	92
2 Programmlogik und Darstellungsmittel für Programmabläufe	32	6.7 Mehrfache Verzweigung (Fallauswahl)	93
2.1 Abstraktion der Wirklichkeit	32	6.8 Schleifen	97
2.2 Programmlogik und Programmablauf	32	6.9 Zählergesteuerte Schleife (Iteration)	98
2.3 Programmabläufe visualisieren	34	6.10 Kopfgesteuerte bedingte Schleife	100
2.4 Programmablaufplan	34	6.11 Fußgesteuerte bedingte Schleife	101
2.5 Datenflussdiagramm	36	6.12 Kontrollstrukturen – Übersicht	104
2.6 Struktogramme	37	6.13 Sprunganweisungen	106
2.7 Pseudocode	39	6.14 Endlosschleifen	107
2.8 Entscheidungstabellen	39	6.15 Übungen	108
2.9 Übung	41	7 Elementare Datenstrukturen	110
3 Werkzeuge der Softwareentwicklung	42	7.1 Warum werden Datenstrukturen benötigt?	110
3.1 Programme erstellen	42	7.2 Arrays	111
3.2 Konzepte zur Übersetzung	43	7.3 Eindimensionale Arrays	112
3.3 Entwicklungsumgebungen	46	7.4 Records	113
3.4 Standardbibliotheken und Wiederverwendung	46	7.5 Zeichenketten	114
3.5 Grundaufbau eines Programms am Beispiel Python	47	7.6 Tupel und Listen	115
3.6 Übungen	52	7.7 Dictionaries	119
4 Zahlensysteme und Zeichencodes	53	7.8 Mengen	121
4.1 Zahlensysteme unterscheiden	53	7.9 Besondere Datenstrukturen anhand von Stapel (Stack) und Schlangen (Queue)	122
4.2 Programme basieren auf Daten	56	7.10 Übungen	126
4.3 Digitales Rechnen	58	8 Methoden, Prozeduren und Funktionen	128
4.4 Zeichencodes	60	8.1 Unterprogramme	128
4.5 Übung	62	8.2 Parameterübergabe	131
		8.3 Parameterübergabe als Wert	132
		8.4 Parameterübergabe über Referenzen	132
		8.5 Rückgabewerte von Funktionen oder Methoden	133
		8.6 Innere Funktionen – Closures	134
		8.7 Lambda-Ausdrücke und Comprehensions	135

8.8	Standardbibliotheken und Built-in-Funktionalitäten	139	11 Grundlagen der Softwareentwicklung	161	
8.9	Übungen	144	11.1	Software entwickeln	161
9	Algorithmen	145	11.2	Methoden	163
9.1	Eigenschaften eines Algorithmus	145	11.3	Der Software-Lebenszyklus	164
9.2	Iterativer Algorithmus	145	11.4	Vorgehensmodelle im Überblick	169
9.3	Rekursiver Algorithmus	147	11.5	Computergestützte Softwareentwicklung (CASE)	175
9.4	Iterativ oder rekursiv?	148	11.6	Fehler finden und identifizieren	175
9.5	Generischer Algorithmus	149	11.7	Begriffsübersicht	184
9.6	Übung	150	11.8	Übung	185
10	Reaktion auf Ereignisse	151	A	Anhang A: PAP, Struktogramm und Pseudocode	186
10.1	Grundlagen zu Ereignissen und Eventhandlern	151	A.1	Beispiel Zinsberechnung	186
10.2	Timer und Scheduler	152	A.2	Beispiel Geldautomat	187
10.3	Ereignisbehandlung bei Programmen mit grafischen Oberflächen	154	B	Anhang B: Installation und Quellangaben	189
10.4	Verschiedene Techniken der Ereignisbehandlung	158	B.1	Python laden und installieren	189
10.5	Begriffsübersicht	160	B.2	Quellangaben im Internet	191
10.6	Übung	160		Stichwortverzeichnis	192