

Inhalt

1	Los geht's – dein smarter Einstieg ins Programmieren	11
1.1	Warum KI das Programmieren revolutioniert	12
1.1.1	Von der Maschinensprache zur natürlichen Sprache	12
1.1.2	Der KI-Quantensprung	14
1.2	Dein KI-Code-Baukasten – die wichtigsten KI-Assistenten	17
1.3	Dein erstes Programm	20
1.3.1	Schritt-für-Schritt zu deinem ersten Programm	20
1.3.2	Ist eine KI für »Hallo Welt« überqualifiziert?	24
1.3.3	Grüße aus Babylon – »Hallo Welt« in anderen Programmiersprachen ...	26
1.3.4	Codeverstehere – die KI erklärt dir den Code	27
1.4	Der »Beipackzettel« zum Buch	28
1.4.1	Das pädagogische Konzept	28
1.4.2	Der menschliche Vorteil: Kreativität	30
1.4.3	Die Inhalte – ein kurzer Überblick	31
2	Die perfekte Basis – deine Entwicklungsumgebung	33
2.1	Visual Studio Code	34
2.1.1	Installation unter Windows, macOS und Linux	34
2.1.2	VSC auf Deutsch umstellen und Python einrichten	36
2.1.3	Ein erstes Python-Projekt in VSC erstellen	38
2.1.4	Integration von KI-Tools in Visual Studio Code	40
2.2	Jupyter Notebook und Google Colab	44
2.2.1	Jupyter Notebook – der Standard für Python-Projekte	44
2.2.2	Google Colab – Jupyter Notebook in der Cloud	45
2.2.3	Anmeldung und Einrichtung von Google Colab	46
2.2.4	Dein erstes Colab Notebook	47
2.2.5	Installation und Verwaltung von Python-Bibliotheken	50

2.2.6	Gemini-KI in Google Colab	51
2.2.7	Jupyter Notebook lokal installieren	52

3 Einfach(e) Programme mit KI schreiben (lassen) 57

3.1	Taschenrechner & Co.	58
3.1.1	Vom Umgang mit Variablen	58
3.1.2	Was sind Variablentypen?	61
3.1.3	Wie verarbeitet man Zahlen und Texte?	62
3.1.4	Lineare Algorithmen und Benutzerinteraktion	64
3.1.5	Einfache Verzweigungen	67
3.1.6	Einen einfachen Taschenrechner programmieren	69
3.1.7	Ein professioneller Taschenrechner mit echten Tasten	71
3.2	Das Abenteuer der Verzweigung	74
3.2.1	Verzweigungen verstehen – der Währungsrechner	74
3.2.2	Ein Mini-Textabenteuer – der virtuelle Escape-Room	76
3.2.3	Im nächtlichen Dschungel der Mathematik	79
3.3	Jetzt geht's rund – Schleifen	81
3.3.1	Das Ziel: Wiederholungen automatisieren	82
3.3.2	Schleifentypen im Überblick	83
3.3.3	Der Passwort-Checker	84
3.4	Der KI-Coach – von der Idee zum Programm	87
3.4.1	Der Chatbot als Programmier-Tutor	87
3.4.2	Vom Konzept zum Code – einen Countdown-Timer entwickeln	88
3.4.3	Bibliotheken für spezielle Aufgaben entdecken	93
3.4.4	Von der Idee über das Struktogramm bis zum Programm	95
3.5	Die KI als Coding-Assistentin	100
3.5.1	Pair Programming – Definition und Einsatzbereiche	100
3.5.2	Explizit anfängertauglichen Code generieren	105
3.5.3	Pair Programming in verschiedenen Umgebungen	108
3.5.4	ChatGPT »Work with Apps«	114
3.5.5	Ein eigenes GPT oder Projekt für die didaktische Programmierung bauen	116

4 Programmieren mit System 121

4.1	Datenverarbeitung	121
4.1.1	Listen und Tabellen als Datenstruktur	122
4.1.2	Listen verarbeiten – die wichtigsten Operationen	125
4.1.3	Praktische Beispiele zum Einsatz von Listen	126
4.1.4	Komplexere Datenstrukturen – Stacks, Queues, Bäume und Graphen	129
4.1.5	Daten einlesen, verarbeiten und speichern	135
4.1.6	Der Vokabeltrainer	140
4.1.7	Arrays und Matrizen	141
4.2	Funktionen definieren	146
4.2.1	Funktionen als Strukturierungselement	146
4.2.2	Parameter und Rückgabewerte	147
4.2.3	Praxisbeispiel: Das Mehrspielerquiz	149
4.3	Bibliotheken verwenden	152
4.3.1	Bibliotheken importieren	152
4.3.2	Lösung linearer Gleichungssysteme mit NumPy	155
4.3.3	Web Scraping mit BeautifulSoup	157
4.4	Grafische Benutzeroberflächen	159
4.4.1	Was ist ein GUI?	159
4.4.2	GUI-Frameworks für Python	160
4.4.3	Fenster und Schaltflächen mit Tkinter erstellen	161
4.4.4	Ereignisse verarbeiten	163
4.4.5	Projekt: Grafikfähiger Taschenrechner mit Tkinter	165
4.4.6	Projekt: To-do-Liste mit PyQt5	167
4.5	Computergrafik	169
4.5.1	Erste Schritte mit Turtle-Grafik	169
4.5.2	Projekt: Das Haus vom Nikolaus	174
4.5.3	3D-Grafik	177
4.5.4	Projekt: Die Mandelbrotmenge (Apfelmännchen)	180
4.5.5	Spieledesign für Einsteiger*innen	182

5 Daten und ihre Visualisierung 189

5.1	KI und Excel	189
5.1.1	KI als Excel-Tutor	191
5.1.2	KI-Integration in Excel & Co.	194
5.1.3	Daten in das Excel-Format konvertieren	198
5.1.4	Excel-Daten per KI verarbeiten	200
5.1.5	Excel um VBA erweitern	205
5.2	Diagramme, Plots und Grafiken erstellen	209
5.2.1	Datensätze aus freien Quellen beschaffen	209
5.2.2	Datenbereinigung und -vorbereitung	212
5.2.3	Die KI als Ideengeber zur Datenanalyse nutzen	214
5.2.4	Visualisierungsbibliotheken im Überblick	218
5.2.5	Projekt: Das Solar-Dashboard	221
5.3	Webdaten nutzen	228
5.3.1	Am Anfang war die API	228
5.3.2	Wie wird das Wetter?	230
5.3.3	Der Börsenticker	234
5.3.4	Projekt: Ein Web Scraper für Medien	238
5.3.5	Einen KI-Chatbot per API nutzen	241
5.3.6	Projekt: Der KI-gestützte Webseitenkompressor	242
5.3.7	Projekt: KI-Bilder via API generieren	245

6 Hardware und Sensoren erforschen 247

6.1	Erste Schritte	247
6.1.1	Der Calliope – ein Mikrocontroller für alle	248
6.1.2	Den Calliope mini einrichten und programmieren	249
6.1.3	Die Alternative – der Funduino Cube	256
6.2	Ich fühle was, was du nicht fühlst – Sensoren	264
6.2.1	Sensoren – ein Überblick	265
6.2.2	Die Umgebungstemperatur messen	266
6.2.3	Die elektronische Wasserwaage	271
6.2.4	Der elektronische Kompass	276

6.3	... und Action! Aktoren nutzen	282
6.3.1	Erste Schritte: LEDs und Sound steuern	282
6.3.2	Motoren steuern	284
6.3.3	Ein Fahrzeug steuern	287
6.4	Komplexe Projekte realisieren	292
6.4.1	Dein persönlicher Hardwareassistent	292
6.4.2	Projekt: Die intelligente Wetterstation auf dem Cube	299
6.4.3	Projekt: Ein Navigationssystem für den Funduino Calle	304
6.4.4	It's Game Time – »Snake« auf dem Calliope	306

7 Künstliche Intelligenz selbst programmieren 313

7.1	Grundlagen des maschinellen Lernens	313
7.1.1	KI, maschinelles Lernen und neuronale Netze	313
7.1.2	Projekt: Deine erste Bildklassifikation	320
7.1.3	Die Grenzen maschinellen Lernens	329
7.2	Neuronale Netze programmieren	330
7.2.1	Grundlagen neuronaler Netze	330
7.2.2	TensorFlow kennenlernen	334
7.2.3	Projekt: Handschrifterkennung	336

8 Fortgeschrittene Techniken 345

8.1	Objektorientierte Programmierung	345
8.1.1	Warum objektorientierte Programmierung?	346
8.1.2	Grundkonzepte der objektorientierten Programmierung	347
8.1.3	Praxisbeispiel: Tierbestand im Zoo verwalten	350
8.1.4	Tipps zur objektorientierten Programmierung	357
8.2	Datenbanken	358
8.2.1	Das Prinzip einer relationalen Datenbank	358
8.2.2	SQLite – der einfache Einstieg	360
8.2.3	MySQL und Docker	367

8.3	App-Entwicklung für Smartphones	379
8.3.1	Apps für alle entwickeln	379
8.3.2	Von der Desktop-App zur mobilen Anwendung	381
8.3.3	Eine plattformübergreifende App mit Flutter bauen	384
8.4	Tools für Profis – Agenten und MCP	391
8.4.1	Agenten	392
8.4.2	Das Model Context Protocol (MCP)	401
8.5	Versionsverwaltung mit GitHub	405
8.5.1	Was ist GitHub und wozu braucht man es?	406
8.5.2	Dein erstes GitHub Repository einrichten	407
8.5.3	GitHub und Visual Studio Code – das perfekte Team	409
9	Tipps und Tricks	415
9.1	Troubleshooting mit KI	415
9.1.1	Wenn nichts funktioniert – Probleme aus der Praxis	416
9.1.2	Systematisches Debugging mit KI	419
9.1.3	Codeoptimierung	422
9.2	Refaktorisierung und Dokumentation	426
9.2.1	Was ist Refactoring?	426
9.2.2	Code zwischen Programmiersprachen übersetzen	430
9.2.3	Dokumentation	431
9.3	Weiterführende Ressourcen	435
9.3.1	Prompt-Sammlungen nicht nur für Programmierer*innen	435
9.3.2	YouTube-Kanäle und Onlinere Ressourcen	435
9.3.3	Bücher	436
9.3.4	Herausforderungen auf Coding-Plattformen	437
9.4	Fazit und Ausblick	438
	Glossar	441
	Index	449