Contents

About Programming

1	**What Is Programming?**	3
1.1	**The Mysterious Power—or: Consciousness Determines Being**	4
1.2	**Algorithms**	6
1.3	**Limits of Classical Algorithms: The Playing Field of Artificial Intelligence**	8
1.3.1	Only Apparently Intelligent	8
1.3.2	Cat or Not Cat: That Is the Question Here	10
2	**Why Learn Programming?**	13
2.1	**Many Good Reasons**	14
2.1.1	Make Your Everyday Life Easier!	14
2.1.2	Speak "IT"!	15
2.1.3	Make Money!	16
2.1.4	Understand What Holds the World Together At Its Core!	17
2.1.5	Train Your Logical, Problem-Solving Thinking!	17
2.2	**Clichés and Prejudices**	18
2.2.1	Programming is Only for Nerds	18
2.2.2	Software and Its Programming Are Just a Fashion, a Hype	19
2.2.3	Programming is Only for Men	19
3	**What Is a Programming Language?**	21
3.1	**Languages for Humans, Languages for Machines**	22
3.2	**Translation and Execution of Programming Languages**	24
3.3	**From Machine Language to High-Level Language**	26
4	**Why Are There So Many Programming Languages?**	29
4.1	**Different Fields of Application**	30
4.2	**Different Paradigms**	31
4.3	**Further Developments and Dialects**	32
5	**Which Programming Languages Should You Learn?**	35
5.1	**Subject of Your Project**	36
5.2	**Customer Preference and Employer Preference**	37
5.3	**Pedagogical Aspects**	38
6	**Some Tips**	41
6.1	**Learn the Basics First!**	42
6.2	**Play!**	42
6.3	**Don't Lose Heart!**	42
6.4	**Start Small, and Let Your Program Grow Incrementally!**	43

Bibliografische Informationen
https://d-nb.info/1299846769

digitalisiert durch DEUTSCHE NATIONAL BIBLIOTHEK

6.5 **Let Beauty Be Beauty!** ... 43
6.6 **Document!** ... 43

The Basic Concepts of Programming

7 **Nine Questions** ... 47

8 **What Do I Need for Programming?** ... 51
8.1 **Tools** ... 52
8.1.1 Compiler and Interpreter ... 52
8.1.2 Code Editors ... 53
8.1.3 Integrated Development Environments (IDEs) ... 54
8.1.4 Simple Online Development Environments ... 59
8.2 **Help and Information** ... 60
8.3 **Generative Artificial Intelligence Like ChatGPT** ... 62
8.4 **Your Roadmap to Learning a New Programming Language** ... 63

9 **What Do I Have to Do to Get a Program Running?** ... 65
9.1 **All Beginnings Are Easy** ... 66
9.2 **Hello, World!** ... 68
9.3 **Your Roadmap to Learning a New Programming Language** ... 69

10 **How Do I Make Sure That I (and Others) Still Understand My Program Later?** ... 71
10.1 **Understandable Program Code** ... 72
10.2 **Design of the Program Code and Naming of Program Elements** ... 73
10.3 **Comments** ... 76
10.3.1 Explain Your Own Program Code ... 76
10.3.2 What Else Comments Are Useful For ... 77
10.3.3 Documentation Outside the Program Code ... 78
10.4 **Your Roadmap to Learning a New Programming Language** ... 80

11 **How Do I Store Data to Work With?** ... 81
11.1 **Variables as Placeholders for Data** ... 82
11.2 **Data Types of Variables** ... 83
11.2.1 Different Types of Data Require Different Types of Variables ... 83
11.2.2 Important Data Types ... 84
11.2.3 Changing the Data Type: Conversion of Variables ... 88
11.3 **Creating and Initializing Variables** ... 89
11.4 **Not So Variable: Constants** ... 92
11.5 **Ordered Fields of Variables/Arrays** ... 92
11.6 **Associative Fields of Variables/Hashes** ... 96
11.7 **Objects** ... 97
11.7.1 The World Is Made of Objects ... 97
11.7.2 Classes ... 98

11.7.3 Inheritance 100
11.7.4 Methods 103
11.7.5 Polymorphism 105
11.7.6 Access Rights 107
11.8 Your Roadmap to Learning a New Programming Language 109
11.9 Solutions to the Exercises 109

12 **How Do I Input and Output Data?** 113
12.1 Forms of Data Input and Output 114
12.2 Graphical or Non-Graphical: That Is the Question Here 115
12.2.1 Graphical User Interfaces 116
12.2.2 Console Applications 124
12.3 Working with Files 126
12.4 Working with Databases 131
12.5 Your Roadmap to Learning a New Programming Language 134
12.6 Solutions to the Exercises 134

13 **How Do I Work with Program Functions to Edit Data and Trigger Actions?** 137
13.1 Functions 138
13.2 Libraries 145
13.3 Frameworks 148
13.4 Application Programming Interfaces (APIs) 149
13.5 Your Roadmap to Learning a New Programming Language 150
13.6 Solutions to the Exercises 151

14 **How Do I Control the Program Flow and Make the Program React to User Actions and Other Events?** 153
14.1 Why Program Flow Control Is Necessary 154
14.2 Forms of Flow Control 155
14.3 If-Else Constructs 155
14.4 A Closer Look at Conditions 161
14.5 Complex Conditions with Logical Operators (AND, OR, NOT) 163
14.6 Checking Similar Conditions Efficiently with Switch-Case Constructs (Switch/Select...Case) 166
14.7 Events 168
14.8 Your Roadmap to Learning a New Programming Language 172
14.9 Solutions to the Exercises 173

15 **Loops: How Do I Repeat Program Instructions Efficiently?** 177
15.1 Loops and Their Variants 178
15.2 Counting Loops 179
15.3 Conditional Loops 185
15.4 Your Roadmap to Learning a New Programming Language 189
15.5 Solutions to the Exercises 189

16 **How Do I Search For and Fix Errors in a Structured Way?** ... 193
16.1 **Errors at Development Time** ... 194
16.2 **Errors at Runtime** ... 195
16.3 **Testing** ... 196
16.4 **Debugging Methods** ... 197
16.5 **Your Roadmap to Learning a New Programming Language** ... 199

Python

17 **Introduction** ... 203

18 **Tools & Help: What Do I Need for Programming?** ... 207
18.1 **Installing the Python Interpreter** ... 208
18.2 **Installing the *PyCharm* IDE** ... 210
18.3 **Getting Help with Python** ... 211
18.4 **Summary** ... 212

19 **Program Execution & Hello World: What Do I Have to Do to Get a Program Running?** ... 213
19.1 **Developing and Executing Programs in Python** ... 214
19.2 **The Python console: Python in interactive mode** ... 221
19.3 **Get to Know *PyCharm*** ... 222
19.4 **Some Alternatives to PyCharm for Python Code Execution** ... 223
19.5 **Summary** ... 224

20 **Syntax, Comments, Code Style & Documentation: How Do I Make Sure That I (And Others) Still Understand My Program Later?** ... 225
20.1 **Design of the Program Code and Naming Conventions** ... 226
20.1.1 Indentation and General Code Formatting ... 226
20.1.2 End of Statement Without Semicolon, Statements Over Several Lines ... 228
20.1.3 Case Sensitivity and Choice of Identifiers ... 229
20.2 **Comments** ... 230
20.3 **Documentation with Docstrings** ... 232
20.4 **Summary** ... 234

21 **Variables & Objects: How Do I Store Data to Work With?** ... 235
21.1 **Creating and Assigning Variables** ... 236
21.2 **Deleting Variables** ... 238
21.3 **Basic Types of Variables** ... 238
21.3.1 Numbers (int, float) ... 239
21.3.2 Strings (str) ... 239
21.3.3 Truth Values (bool) ... 241
21.3.4 None ... 242

21.3.5 Other Data Types ... 243
21.4 **Variables as Objects** ... 244
21.4.1 Attributes and Methods of Variables ... 244
21.4.2 Creating Variables with the Constructor Method ... 248
21.5 **Converting Variables** ... 249
21.6 **Complex Data Types** ... 251
21.6.1 Lists ... 251
21.6.2 Tuples ... 258
21.6.3 Dictionaries ... 259
21.6.4 Sets ... 262
21.7 **Self-Defined Classes** ... 264
21.7.1 Defining and Using Classes ... 264
21.7.2 Deriving Classes from Other Classes ... 266
21.7.3 Avoid Ambiguity: Name Mangling ... 268
21.8 **Summary** ... 268
21.9 **Solutions to the Exercises** ... 271

22 **User Interfaces: How Do I Input and Output Data?** ... 277
22.1 **Input and Output in the Console** ... 278
22.2 **Graphical User Interfaces with Tkinter** ... 280
22.2.1 Overview ... 280
22.2.2 Hello Tkinter! ... 281
22.2.3 Graphical Controls (Widgets) ... 282
22.2.4 Arrangement of the Controls (Geometry Managers) ... 299
22.2.5 Events ... 305
22.2.6 Example: Calculator Application ... 308
22.3 **Working with Files** ... 315
22.4 **Exercise: Development of a Simple Text Editor** ... 317
22.5 **Summary** ... 318
22.6 **Solutions to the Exercises** ... 319

23 **Functions & Methods: How Do I Work with Program Functions to Work with Data and Trigger Actions?** ... 327
23.1 **Working with Functions** ... 328
23.1.1 Definition of Functions ... 328
23.1.2 Function Arguments ... 329
23.1.3 Return Values ... 334
23.1.4 Local and Global Variables ... 335
23.2 **Using Functions as Class Methods of Objects** ... 339
23.3 **Working with Modules and Packages** ... 342
23.3.1 Modularizing Program Code ... 342
23.3.2 Importing Elements from Modules ... 344
23.3.3 Using the Community: The Python Package Index (PyPI) ... 345
23.4 **Summary** ... 348
23.5 **Solutions to the Exercises** ... 349

24 **Conditional Statements & Event Handling: How Do I Control the Program Flow and Make the Program React to User Actions and Other Events?** ... 353
24.1 **If-Else Constructs** ... 354
24.1.1 Simple If-Else Constructs ... 354
24.1.2 Nested if-else Constructs ... 356
24.1.3 If-Else Construct with Composite Conditions ... 358
24.1.4 If-Else Construct with Alternative Conditions (elif) ... 359
24.2 **Events** ... 361
24.3 **Summary** ... 362
24.4 **Solutions to the Exercises** ... 362

25 **Loops: How Do I Repeat Program Instructions Efficiently?** ... 365
25.1 **Counting Loops (for)** ... 366
25.1.1 Simple for-Loops ... 366
25.1.2 Nested for-Loops ... 370
25.1.3 List Comprehension Expressions ... 371
25.2 **Conditional Loops (while)** ... 372
25.3 **Exiting Loops Prematurely and Restarting Them** ... 375
25.4 **Summary** ... 377
25.5 **Solutions to the Exercises** ... 377

26 **Debugging & Error-Handling: How to Find and Fix Errors in a Structured Way** ... 381
26.1 **Error Handling at Runtime** ... 382
26.1.1 Catching Errors Through Targeted Checks ... 382
26.1.2 Try-Except Constructs ... 383
26.2 **Troubleshooting During Development** ... 386
26.2.1 Breakpoints ... 386
26.2.2 Display of Variable Content and Use of Watches ... 388
26.2.3 Step-by-Step Execution ... 389
26.3 **Summary** ... 390
26.4 **Solutions to the Exercises** ... 391

JavaScript

27 **Introduction** ... 395

28 **Tools & Help: What Do I Need for Programming?** ... 399
28.1 **Interpreter** ... 400
28.2 **Code Editors and Development Environments** ... 400
28.3 **Assistance and Documentation** ... 401
28.4 **Summary** ... 402

29 **Program Execution & Hello World: What Do I Have to Do to Get a Program Running?** ... 403
29.1 **Embedding JavaScript Code in Web Pages** ... 404
29.1.1 The Script Element in HTML ... 404
29.1.2 Safety Aspects ... 408
29.2 **"Hello World" in JavaScript** ... 409
29.2.1 Do-it-Yourself: The (Not so) Laborious Way ... 409
29.2.2 With a Little Help: Fast Implementation with a Web Service ... 411
29.3 **Summary** ... 412

30 **Syntax, Comments, Code Style & Documentation: How Do I Make Sure That I (And Others) Still Understand My Program Later?** ... 413
30.1 **Design of the Program Code and Naming Conventions** ... 414
30.2 **Comments** ... 416
30.3 **Summary** ... 417

31 **Variables & Objects: How Do I Store Data to Work With?** ... 419
31.1 **Declaration of Variables** ... 420
31.2 **Elementary Data Types** ... 421
31.2.1 Numbers (number) ... 421
31.2.2 Text (string) ... 423
31.2.3 Logical Values (boolean) ... 427
31.2.4 Special Types and Values (null, undefined, NaN) ... 428
31.3 **Converting Variables** ... 430
31.3.1 Implicit Conversion ... 430
31.4 **Arrays** ... 433
31.5 **Objects** ... 441
31.5.1 Object Orientation in JavaScript ... 441
31.5.2 Creating Objects Directly ... 442
31.5.3 Accessing Properties of Objects ... 443
31.5.4 Creating Objects Using the Object Constructor ... 444
31.5.5 Creating Objects Using Constructor Functions ... 444
31.5.6 JSON ... 445
31.6 **Summary** ... 446
31.7 **Solutions to the Exercises** ... 448

32 **User Interfaces: How Do I Input and Output Data?** ... 451
32.1 **Overview of Input and Output in JavaScript** ... 453
32.2 **Output Via the Console** ... 454
32.3 **Inputs and Outputs Via Dialog Boxes** ... 457
32.4 **Output to an HTML Document/Web Page** ... 458
32.4.1 Writing HTML Code into the Web Page ... 458

32.4.2 The Document Object Model (DOM) ... 460
32.4.3 Selecting DOM Nodes by Their Properties ... 461
32.4.4 Selecting DOM Nodes via the Hierarchical Structure of the Document ... 463
32.4.5 Changing HTML Elements ... 466
32.4.6 Adding and Deleting HTML Elements ... 468
32.5 **Input with Forms** ... 470
32.5.1 Forms in HTML ... 470
32.5.2 Accessing Forms from JavaScript ... 472
32.6 **Example: Simple Calculator** ... 476
32.6.1 The Web Interface ... 477
32.6.2 The CSS Design Instructions ... 479
32.6.3 The JavaScript Code ... 482
32.7 **Example: Color Picker** ... 484
32.7.1 The Web Interface ... 484
32.7.2 The JavaScript Code ... 486
32.8 **Summary** ... 488
32.9 **Solutions to the Exercises** ... 490

33 **Functions & Methods: How Do I Work with Program Functions to Work with Data and Trigger Actions?** ... 493
33.1 **Working with Functions** ... 494
33.1.1 Definition of Functions ... 494
33.1.2 Return Values ... 499
33.1.3 Arguments and Parameters of Functions ... 500
33.1.4 Scope of Variables in Functions ... 504
33.2 **Working with Modules/Libraries** ... 506
33.2.1 Developing and Using Your Own Modules ... 506
33.2.2 Finding and Integrating External Modules/Libraries ... 508
33.3 **Frameworks** ... 509
33.4 **Summary** ... 510
33.5 **Solutions to the Exercises** ... 511

34 **Conditional Statements & Event Handling: How Do I Control the Program Flow and Make the Program React to User Actions and Other Events?** ... 515
34.1 **If-Else Constructs** ... 516
34.2 **Switch-Case Constructs** ... 519
34.3 **Events** ... 520
34.4 **Summary** ... 524
34.5 **Solutions to the Exercises** ... 525

35 **Loops: How Do I Repeat Program Instructions Efficiently?** ... 527
35.1 **Counting Loops (for and for-of)** ... 528
35.1.1 for-Loops with Numerical Run Variables ... 528
35.1.2 for-Loop with Object Run Variable (for...of) ... 534

35.2 Conditional Loops (while and do-while) 536
35.3 Summary 537
35.4 Solutions to the Exercises 538

36 **Debugging & Error-Handling: How Do I Search for and Fix Errors in a Structured Way?** 543
36.1 Error Handling at Runtime 544
36.2 Troubleshooting During Development 546
36.3 Summary 550

Index **553**