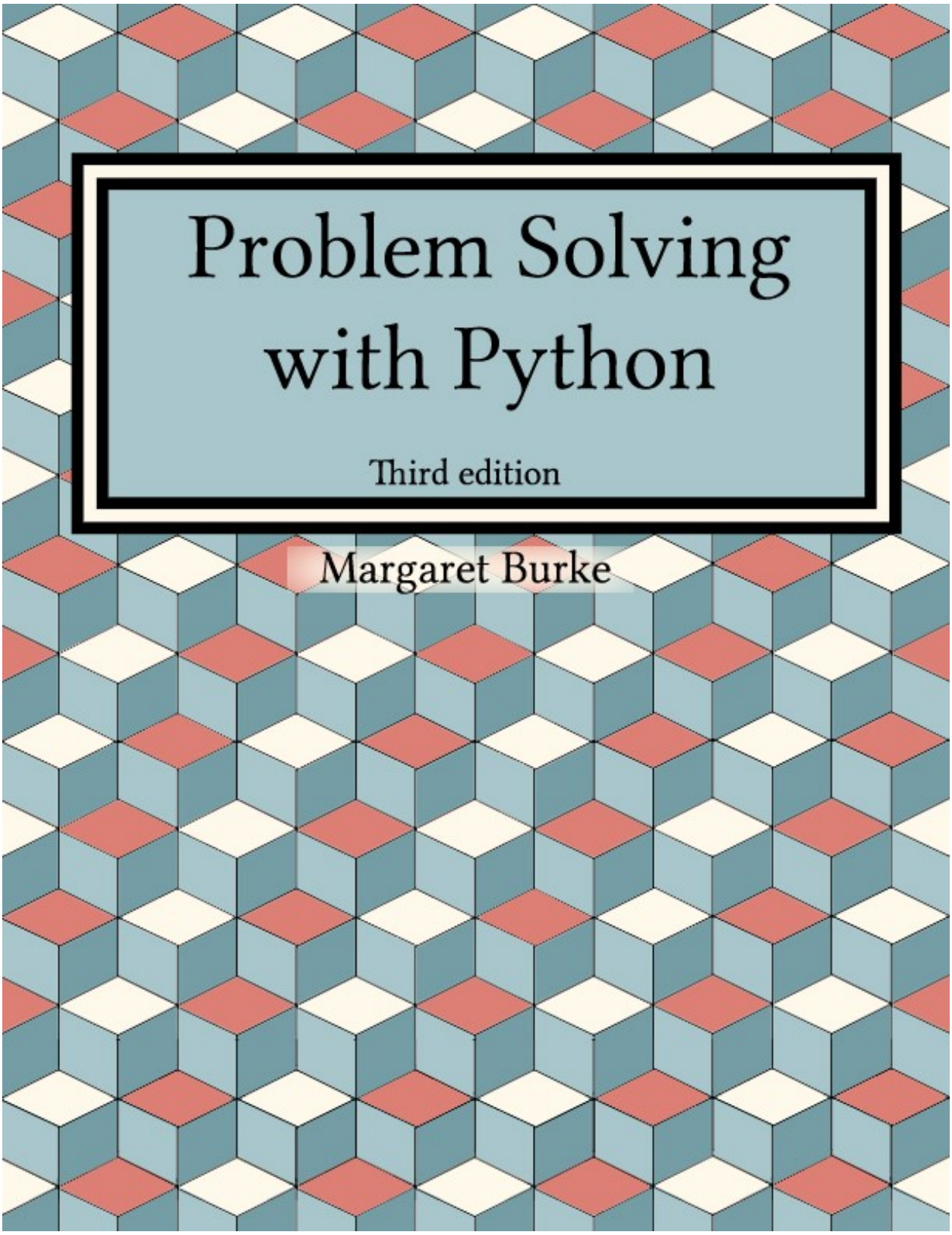
Problem Solving
with Python
Third edition
Margaret Burke

Copyright © 2018, 2021, 2022 by Margaret Burke

ISBN: 979-8-9861513-1-1

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the copyright owner.

Kendall-Hunt Publishing Company previously published this book.

Published in the United States of America

Table of Contents

Chapter 0 Number systems 1

0.0 What is a number base? 1

0.1 Binary numbers 2

0.2 Hexadecimal numbers 4

0.3 Conversions 6

0.3.1 Converting hexadecimal to decimal 6

0.3.2 Converting between hexadecimal and binary 7

0.3.3 Converting from decimal to binary 8

0.3.4 Converting from decimal to hexadecimal 10

0.4 Seven-bit ASCII, 8-bit ASCII, Unicode: How many digits do we need? 12

0.5 Challenge accepted! 15

Chapter 1 Getting started with Python 1

1.0 Problem solving 1

1.1 Let's get started! 3

1.2 Meet the turtle 4

1.3 Designing solutions 8

1.3.0 Defining the problem 9

1.3.1 Designing an algorithm 10

1.4 When things go wrong 11

1.5 Challenge accepted! 13

Chapter 2 Input, processing, output programs 15

2.0 Your programming tool kit 15

2.1 Literals and data types 16

2.2 Variables 18

2.3 Variable names and keywords 19

2.4 Constants 22

2.5 Evaluating expressions 23

2.6 Operators and operands 25

2.7 Order of operations 28

2.8 Math functions and mathematical expressions in Python 30
2.9 Input and type conversion functions 33
2.10 Formatting output 35
2.10.0 String concatenation 36
2.10.1 The format function 37
2.10.2 Escape characters 38
2.10.3 Formatting a table of output 39
2.11 Comments and variable annotations 40
2.12 Using your tool kit 42
2.12.0 Define the problem 42
2.12.1 Identify and clearly express solutions 44
2.12.2 Evaluate and select among those solutions 45
2.12.3 Implement a solution 45
2.12.4 Evaluate the result 49
2.13 Challenge accepted! 50
Chapter 3 Logic and conditional statements 52
3.0 Changing the flow 52
3.1 Relational operators and Boolean expressions 56
3.2 Boolean operators and short-circuiting 60
3.2.0 Boolean operators 60
3.2.1 Short-circuiting 61
3.2.2 The `not` operator 62
3.2.3 Order of operations 63
3.3 Checking ranges 64
3.4 Boolean variables 67
3.5 What else? 69
3.6 Complementing logical expressions 70
3.7 Testing assumptions with assert statements 73
3.8 Series if statements 75
3.9 Nested if statements 79
3.10 Desk checking and debugging conditional statements 80

3.10.0 Desk checking 80
3.10.1 Debugging 82
3.10.2 Debugging with print statements 87
3.10.3 Debugging is problem-solving 88
3.11 Data representation 92
3.11.0 Representing continuous data in a digital world 92
3.11.1 Numeric types 92
3.11.2 Strings 95
3.12 The `random` library 97
3.13 Challenge accepted! 98
Chapter 4 Loops and files 102
4.0 Finding repetition in a problem 102
4.1 Desk-checking loops 107
4.2 The loop control variable 111
4.3 Loop design errors 112
4.4 Augmented assignment operators 114
4.5 Loop patterns 115
4.5.0 Stepper variables 115
4.5.1 Most-recent holder variables 117
4.5.2 Accumulator variables 118
4.5.3 Most-wanted holder variables 119
4.5.4 The haversine code 123
4.6 Input validation loops 124
4.7 Sentinel-controlled loops 125
4.8 File processing 128
4.8.0 Writing to files 128
4.8.1 Reading from files 131
4.8.2 Testing if a file exists 136
4.9 Data analysis with loops and files 137
4.10 Counting loops and ranges 142
4.10.0 Counting (for) loops 142

4.10.1 Ranges 143
4.10.2 When to choose counting loops 145
4.10.3 Solving a counting loop problem 147
4.10.4 Desk-checking counting loops 148
4.11 Nested loops 150
4.12 Finding repetition in a problem (again) 154
4.13 Challenge accepted! 158
Chapter 5 Functions 162
5.0 Defining functions 162
5.1 Functional decomposition 164
5.2 Flow of control 167
5.3 Parameters 171
5.4 Returning values 177
5.5 Input–processing–output (IPO) charts and structure charts 183
5.5.0 Input–processing–output charts 183
5.5.1 Structure charts 184
5.6 Code blocks, namespaces, and scope 185
5.6.0 Code blocks and namespaces 185
5.6.1 Shadowing 187
5.6.2 Best practices 188
5.6.3 Module namespace 188
5.7 The math module, revisited 190
5.8 Input validation functions 193
5.9 Event handling and functions 195
5.10 Data analysis with functions 199
5.11 Problem-solving with functions 203
5.12 Challenge accepted! 208
Chapter 6 Lists 213
6.0 Collections and indexing 213
6.1 Loops and lists 215
6.1.0 Stepper-controlled loops 215

6.1.1 For loops 218
6.1.2 Annotating list variables 220
6.1.3 Common loop algorithms 220
6.2 Slicing, concatenating, and copying 225
6.2.0 Variables and memory 225
6.2.1 Concatenation 227
6.2.2 Slicing 228
6.2.3 The `copy` method 229
6.3 Operations on lists 230
6.3.0 Adding to a list 230
6.3.1 Finding an element in a list 231
6.3.2 Removing from a list 232
6.3.3 Processing a list 233
6.4 Lists and functions 234
6.5 Lists and files 238
6.5.0 Splitting strings 238
6.5.1 Reading data from a file 239
6.6 Two-dimensional lists 243
6.7 Parallel lists and records 247
6.7.0 Parallel lists 248
6.7.1 Using lists as records 249
6.8 Strings, revisited 250
6.9 Lists for data analysis 252
6.10 Problem-solving with lists 256
6.10.0 Define the problem 256
6.10.1 Identify and clearly express solutions 257
6.10.2 Evaluate and select among those solutions 259
6.10.3 Implement a solution 259
6.10.4 Evaluate the result 261
6.11 Challenge accepted! 262
Chapter 7 Recursion 265

7.0 Recursive problems 265
7.1 The call stack 271
7.2 Designing recursive solutions 277
7.2.0 Solving a problem that is defined recursively 277
7.2.1 Solving a collection processing problem recursively 278
7.3 Challenge accepted! 282
Chapter 8 More collection types: dictionaries and sets 285
8.0 Python's dict type 285
8.1 Dictionaries with collection values 287
8.1.0 A dictionary with list values 287
8.1.1 A dictionary with dictionary values 291
8.2 Iterating over a dictionary 294
8.3 Default dictionaries and anonymous functions 296
8.3.0 Creating and using default dictionaries 296
8.3.1 Writing your own `default_factory` function 298
8.3.2 Anonymous functions 300
8.4 Dictionaries for data analysis 303
8.5 List comprehensions 308
8.6 Sets 311
8.6.0 Set operations 312
8.6.1 Sets of sets: `frozensets`, combinations 318
8.6.2 Order of operations 321
8.7 Problem-solving with sets 322
8.8 Challenge accepted! 328
Chapter 9 Analytics Libraries 333
9.0 Installing 333
9.1 NumPy 333
9.2 pandas 342
9.3 matplotlib 355
9.4 Comparing illness death rate to healthcare by state 362
9.5 Challenge accepted! 364

References **368**
Index **369**