

Research Software Engineering with Python

Building software that makes research
possible

Damien Irving

Kate Hertweck

Luke Johnston

Joel Ostblom

Charlotte Wickham

Greg Wilson



CRC Press

Taylor & Francis Group

Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an informa business
A CHAPMAN & HALL BOOK

Contents

Welcome	1
0.1 The Big Picture	1
0.2 Intended Audience	2
0.3 What You Will Learn	3
0.4 Using this Book	4
0.5 Contributing and Re-Use	4
0.6 Acknowledgments	5
1 Getting Started	7
1.1 Project Structure	7
1.2 Downloading the Data	9
1.3 Installing the Software	10
1.4 Summary	11
1.5 Exercises	12
1.6 Key Points	12
2 The Basics of the Unix Shell	13
2.1 Exploring Files and Directories	14
2.2 Moving Around	19
2.3 Creating New Files and Directories	24
2.4 Moving Files and Directories	28
2.5 Copying Files and Directories	29
2.6 Deleting Files and Directories	31
2.7 Wildcards	32
2.8 Reading the Manual	34

2.9	Summary	36
2.10	Exercises	37
2.11	Key Points	44
3	Building Tools with the Unix Shell	45
3.1	Combining Commands	45
3.2	How Pipes Work	49
3.3	Repeating Commands on Many Files	53
3.4	Variable Names	57
3.5	Redoing Things	57
3.6	Creating New Filenames Automatically	59
3.7	Summary	61
3.8	Exercises	61
3.9	Key Points	67
4	Going Further with the Unix Shell	69
4.1	Creating New Commands	70
4.2	Making Scripts More Versatile	72
4.3	Turning Interactive Work into a Script	74
4.4	Finding Things in Files	75
4.5	Finding Files	80
4.6	Configuring the Shell	85
4.7	Summary	89
4.8	Exercises	89
4.9	Key Points	93
5	Building Command-Line Tools with Python	95
5.1	Programs and Modules	96
5.2	Handling Command-Line Options	98
5.3	Documentation	100
5.4	Counting Words	102
5.5	Pipelining	106

5.6	Positional and Optional Arguments	108
5.7	Collating Results	108
5.8	Writing Our Own Modules	111
5.9	Plotting	114
5.10	Summary	116
5.11	Exercises	116
5.12	Key Points	120
6	Using Git at the Command Line	121
6.1	Setting Up	124
6.2	Creating a New Repository	125
6.3	Adding Existing Work	127
6.4	Describing Commits	129
6.5	Saving and Tracking Changes	130
6.6	Synchronizing with Other Repositories	138
6.7	Exploring History	143
6.8	Restoring Old Versions of Files	147
6.9	Ignoring Files	150
6.10	Summary	151
6.11	Exercises	151
6.12	Key Points	156
7	Going Further with Git	157
7.1	What's a Branch?	158
7.2	Creating a Branch	160
7.3	What Curve Should We Fit?	161
7.4	Verifying Zipf's Law	164
7.5	Merging	172
7.6	Handling Conflicts	174
7.7	A Branch-Based Workflow	178
7.8	Using Other People's Work	180

7.9	Pull Requests	184
7.10	Handling Conflicts in Pull Requests	193
7.11	Summary	194
7.12	Exercises	195
7.13	Key Points	196
8	Working in Teams	199
8.1	What Is a Project?	200
8.2	Include Everyone	201
8.3	Establish a Code of Conduct	202
8.4	Include a License	206
8.5	Planning	210
8.6	Bug Reports	212
8.7	Labeling Issues	214
8.8	Prioritizing	216
8.9	Meetings	218
8.10	Making Decisions	221
8.11	Make All This Obvious to Newcomers	223
8.12	Handling Conflict	223
8.13	Summary	226
8.14	Exercises	227
8.15	Key Points	229
9	Automating Analyses with Make	231
9.1	Updating a Single File	233
9.2	Managing Multiple Files	235
9.3	Updating Files When Programs Change	237
9.4	Reducing Repetition in a Makefile	238
9.5	Automatic Variables	240
9.6	Generic Rules	241
9.7	Defining Sets of Files	243

9.8	Documenting a Makefile	246
9.9	Automating Entire Analyses	248
9.10	Summary	251
9.11	Exercises	251
9.12	Key Points	255
10	Configuring Programs	257
10.1	Configuration File Formats	258
10.2	Matplotlib Configuration	259
10.3	The Global Configuration File	260
10.4	The User Configuration File	262
10.5	Adding Command-Line Options	263
10.6	A Job Control File	264
10.7	Summary	267
10.8	Exercises	268
10.9	Key Points	270
11	Testing Software	271
11.1	Assertions	272
11.2	Unit Testing	274
11.3	Testing Frameworks	276
11.4	Testing Floating-Point Values	280
11.5	Integration Testing	283
11.6	Regression Testing	285
11.7	Test Coverage	286
11.8	Continuous Integration	289
11.9	When to Write Tests	293
11.10	Summary	294
11.11	Exercises	294
11.12	Key Points	298

12 Handling Errors	299
12.1 Exceptions	300
12.2 Writing Useful Error Messages	307
12.3 Testing Error Handling	309
12.4 Reporting Errors	310
12.5 Summary	314
12.6 Exercises	314
12.7 Key Points	319
13 Tracking Provenance	321
13.1 Data Provenance	323
13.2 Code Provenance	325
13.3 Summary	330
13.4 Exercises	330
13.5 Key Points	333
14 Creating Packages with Python	335
14.1 Creating a Python Package	336
14.2 Virtual Environments	339
14.3 Installing a Development Package	342
14.4 What Installation Does	349
14.5 Distributing Packages	350
14.6 Documenting Packages	355
14.7 Software Journals	365
14.8 Summary	366
14.9 Exercises	366
14.10 Key Points	368
15 Finale	369
15.1 Why We Wrote This Book	370
Appendix	370

A Solutions	371
B Learning Objectives	419
B.1 Getting Started	419
B.2 The Basics of the Unix Shell	419
B.3 Building Tools with the Unix Shell	420
B.4 Going Further with the Unix Shell	420
B.5 Building Command-Line Tools with Python	420
B.6 Using Git at the Command Line	421
B.7 Going Further with Git	421
B.8 Working in Teams	422
B.9 Automating Analyses with Make	422
B.10 Configuring Programs	422
B.11 Testing Software	423
B.12 Handling Errors	423
B.13 Tracking Provenance	423
B.14 Creating Packages with Python	424
C Key Points	425
C.1 Getting Started	425
C.2 The Basics of the Unix Shell	425
C.3 Building Tools with the Unix Shell	426
C.4 Going Further with the Unix Shell	427
C.5 Building Command-Line Programs in Python	427
C.6 Using Git at the Command Line	428
C.7 Going Further with Git	428
C.8 Working in Teams	429
C.9 Automating Analyses with Make	429
C.10 Configuring Programs	430
C.11 Testing Software	430
C.12 Handling Errors	430
C.13 Tracking Provenance	431
C.14 Creating Packages with Python	431

D Project Tree	433
E Working Remotely	437
E.1 Logging In	437
E.2 Copying Files	439
E.3 Running Commands	441
E.4 Creating Keys	441
E.5 Dependencies	443
F Writing Readable Code	447
F.1 Python Style	447
F.2 Order	450
F.3 Checking Style	452
F.4 Refactoring	454
F.5 Code Reviews	463
F.6 Python Features	467
F.7 Summary	472
G Documenting Programs	473
G.1 Writing Good Docstrings	473
G.2 Defining Your Audience	476
G.3 Creating an FAQ	478
H YAML	481
I Anaconda	485
I.1 Package Management with conda	485
I.2 Environment Management with conda	487
J Glossary	489
K References	503
Index	511