Advanced Python Development

Using Powerful Language Features in Real-World Applications

Matthew Wilkes

Apress®

Table of Contents

About the Author xi

About the Technical Reviewers xiii

Acknowledgments xv

Introduction xvii

Chapter 1: Prototyping and environments 1

Prototyping in Python 1

Prototyping with the REPL 2

Prototyping with a Python script 6

Prototyping with scripts and pdb 7

Prototyping with Jupyter 11

Prototyping in this chapter 15

Environment setup 18

Setting up a new project 19

Prototyping our scripts 20

Installing dependencies 23

Exporting to a .py file 27

Building a command-line interface 29

The sys module and argv 30

argparse 32

click 34

Pushing the boundaries 37

Remote kernels 38

Developing code that cannot be run locally 42

The completed script 46
Summary 48
Additional resources 49

Chapter 2: Testing, checking, linting 51

Testing 54
When to write tests 57
Creating formatting functions for improved testability 59
pytest 63
Type checking 77
Installing mypy 78
Adding type hints 79
Subclasses and inheritance 82
Generic types 85
Debugging and overuse of typing 87
When to use typing and when to avoid it 89
Keeping type hints separate from code 90
Linting 92
Installing flake8 and black 94
Fixing existing code 94
Running automatically 96
Running on pull requests 98
Summary 99
Additional resources 100

Chapter 3: Packaging scripts 103

Terminology 104
Directory structure 105
Setup scripts and metadata 108
Dependencies 109
Declarative configurations 110
Things to avoid in setup.py 111
Using setup.cfg 117

Custom index servers 119
Setting up pypiserver 121
Durability 123
Confidentiality 123
Integrity 124
Wheel formats and executing code on installation 125
Installing the console script using entrypoints 129
README, DEVELOP, and CHANGES 130
Markdown format 131
reStructured text format 133
README 135
CHANGES.md and versioning 136
Upstream dependency version pins 138
Loose pins 139
Strict pins 140
Which pinning scheme to use 141
Uploading a version 141
Configuring twine 143
Summary 144
Additional resources 144

Chapter 4: From script to framework 147

Writing a sensor plugin 148
Developing the plugin 149
Adding a new command option 152
Subcommands 153
Command options 156
Error handling 157
Off-loading parsing to Click with argument types 162
Custom click argument types 163
Canned options 166

Allowing third-party sensor plugins 167
Plugin detection using fixed names 169
Plugin detection using entrypoints 170
Configuration files 174
Environment variables 178
Approach for apd.sensors vs. similar programs 179
Summary 180
Additional resources 181

Chapter 5: Alternative interfaces 183

Web microservices 183
WSGI 184
API design 190
Flask 192
Python decorators 196
Testing the view function 210
Deployment 213
Extending software as a third party 214
Agreeing on an ad hoc signature with peers 221
Abstract base classes 223
Fallback strategies 227
Bringing it all together 233
Fixing the serialization problem in our code 235
Tidying up 239
Versioning APIs 240
Testability 242
Summary 244
Additional resources 245

Chapter 6: Aggregation process 247

Cookiecutter 247
Creating a new template 249
Creating the aggregation package 252

Database types 254
Our example 257
Object-relational mappers 258
Versioning the database 263
Loading data 270
New technologies 279
Databases 279
Custom attribute behavior 279
Generators 280
Summary 280
Additional resources 280

Chapter 7: Parallelization and async 283

Nonblocking IO 284
Making our code nonblocking 289
Multithreading and multiprocessing 291
Low-level threads 292
Bytecode 296
Locks and deadlocks 300
Avoiding global state 306
Other synchronization primitives 312
ProcessPoolExecutors 321
Making our code multithreaded 321
AsyncIO 322
async def 323
await 324
async for 327
async with 331
Async locking primitives 332
Working with synchronous libraries 334
Making our code asynchronous 335

Comparison 339
Making a choice 341
Summary 343
Additional resources 343

Chapter 8: Advanced asyncio 345

Testing async code 345
Testing our code 347
Mocking objects for easier unit testing 356
Asynchronous databases 368
Classic SQLAlchemy style 369
Using run_in_executor 373
Querying data 376
Avoiding complex queries 378
Alternatives 391
Global variables in asynchronous code 392
Summary 395
Additional resources 395

Chapter 9: Viewing the data 397

Query functions 397
Filtering data 404
Multilevel iterators 408
Additional filters 415
Testing our query functions 417
Displaying multiple sensors 421
Processing data 425
Interactivity with Jupyter widgets 430
Multiply nested synchronous and asynchronous code 431
Tidying up 437
Persistent endpoints 439

Charting maps and geographic data 440
New plot types 442
Supporting map type charts in apd.aggregation 445
Drawing a custom map using the new configs 448
Summary 451
Additional resources 452

Chapter 10: Speeding things up 453

Optimizing a function 453
Profiling and threads 455
Interpreting the profile report 459
Other profilers 462
Optimizing control flow 468
Visualizing profiling data 473
Caching 477
Summary 489
Additional resources 489

Chapter 11: Fault tolerance 491

Error handling 491
Getting items from a container 492
Custom exceptions 498
Tracebacks involving multiple exceptions 502
Testing for exception handling 507
Warnings 514
Warning filters 518
Logging 520
Nested loggers 522
Custom actions 523
Logging configuration 530
Other handlers 532

Designing around problems 533
Scheduling sensor lookups 533
Summary 539
Additional resources 540

Chapter 12: Callbacks and data analysis 541

Generator data flow 541
Generators that consume their own output 543
Enhanced generators 548
Queues 556
Choosing a control flow 559
Structure for our actions 560
Analysis coroutines 561
Ingesting data 567
Running the analysis process 571
Process status 574
Callbacks 578
Extending the actions available 581
Summary 584
Additional resources 584
Epilogue 585

Index 589