

Table of Contents

Preface	1
Chapter 1: A Gentle Introduction to Python	9
A proper introduction	11
Enter the Python	13
About Python	13
Portability	13
Coherence	14
Developer productivity	14
An extensive library	14
Software quality	14
Software integration	15
Satisfaction and enjoyment	15
What are the drawbacks?	15
Who is using Python today?	16
Setting up the environment	16
Python 2 versus Python 3	16
Installing Python	18
Setting up the Python interpreter	18
About virtualenv	20
Your first virtual environment	21
Your friend, the console	24
How you can run a Python program	24
Running Python scripts	24
Running the Python interactive shell	25
Running Python as a service	26
Running Python as a GUI application	27
How is Python code organized?	27
How do we use modules and packages?	29
Python's execution model	32
Names and namespaces	32
Scopes	34
Objects and classes	37
Guidelines on how to write good code	40
The Python culture	41
A note on IDEs	42
Summary	43
Chapter 2: Built-in Data Types	45
Everything is an object	46

Mutable or immutable? That is the question	47
Numbers	48
Integers	48
Booleans	50
Real numbers	51
Complex numbers	52
Fractions and decimals	53
Immutable sequences	54
Strings and bytes	54
Encoding and decoding strings	55
Indexing and slicing strings	56
String formatting	57
Tuples	58
Mutable sequences	59
Lists	59
Byte arrays	63
Set types	64
Mapping types – dictionaries	66
The collections module	70
namedtuple	71
defaultdict	72
ChainMap	73
Enums	74
Final considerations	75
Small values caching	75
How to choose data structures	76
About indexing and slicing	78
About the names	79
Summary	80
Chapter 3: Iterating and Making Decisions	81
Conditional programming	82
A specialized else – elif	83
The ternary operator	85
Looping	86
The for loop	86
Iterating over a range	87
Iterating over a sequence	87
Iterators and iterables	89
Iterating over multiple sequences	90
The while loop	92
The break and continue statements	95
A special else clause	97
Putting all this together	99
A prime generator	99

Applying discounts	101
A quick peek at the itertools module	104
Infinite iterators	105
Iterators terminating on the shortest input sequence	106
Combinatoric generators	107
Summary	107
Chapter 4: Functions, the Building Blocks of Code	109
Why use functions?	110
Reducing code duplication	111
Splitting a complex task	111
Hiding implementation details	112
Improving readability	113
Improving traceability	114
Scopes and name resolution	115
The global and nonlocal statements	116
Input parameters	118
Argument-passing	118
Assignment to argument names doesn't affect the caller	120
Changing a mutable affects the caller	120
How to specify input parameters	121
Positional arguments	121
Keyword arguments and default values	122
Variable positional arguments	123
Variable keyword arguments	124
Keyword-only arguments	126
Combining input parameters	126
Additional unpacking generalizations	128
Avoid the trap! Mutable defaults	129
Return values	130
Returning multiple values	132
A few useful tips	133
Recursive functions	134
Anonymous functions	134
Function attributes	136
Built-in functions	137
One final example	137
Documenting your code	138
Importing objects	139
Relative imports	141
Summary	142
Chapter 5: Saving Time and Memory	143
The map, zip, and filter functions	145
map	145
zip	148

filter	149
Comprehensions	150
Nested comprehensions	151
Filtering a comprehension	152
dict comprehensions	154
set comprehensions	155
Generators	156
Generator functions	156
Going beyond next	159
The yield from expression	163
Generator expressions	164
Some performance considerations	166
Don't overdo comprehensions and generators	169
Name localization	173
Generation behavior in built-ins	175
One last example	175
Summary	177
Chapter 6: OOP, Decorators, and Iterators	179
Decorators	179
A decorator factory	185
Object-oriented programming (OOP)	187
The simplest Python class	188
Class and object namespaces	189
Attribute shadowing	190
Me, myself, and I – using the self variable	191
Initializing an instance	192
OOP is about code reuse	193
Inheritance and composition	193
Accessing a base class	198
Multiple inheritance	200
Method resolution order	203
Class and static methods	205
Static methods	205
Class methods	207
Private methods and name mangling	209
The property decorator	211
Operator overloading	213
Polymorphism – a brief overview	214
Data classes	215
Writing a custom iterator	216
Summary	217
Chapter 7: Files and Data Persistence	219
Working with files and directories	220
Opening files	220

Using a context manager to open a file	222
Reading and writing to a file	222
Reading and writing in binary mode	223
Protecting against overriding an existing file	224
Checking for file and directory existence	224
Manipulating files and directories	225
Manipulating pathnames	227
Temporary files and directories	228
Directory content	229
File and directory compression	230
Data interchange formats	230
Working with JSON	231
Custom encoding/decoding with JSON	234
IO, streams, and requests	238
Using an in-memory stream	238
Making HTTP requests	239
Persisting data on disk	242
Serializing data with pickle	242
Saving data with shelve	244
Saving data to a database	245
Summary	252
Chapter 8: Testing, Profiling, and Dealing with Exceptions	253
Testing your application	254
The anatomy of a test	256
Testing guidelines	257
Unit testing	259
Writing a unit test	259
Mock objects and patching	261
Assertions	261
Testing a CSV generator	262
Boundaries and granularity	271
Testing the export function	271
Final considerations	274
Test-driven development	276
Exceptions	278
Profiling Python	284
When to profile?	286
Summary	288
Chapter 9: Cryptography and Tokens	289
The need for cryptography	289
Useful guidelines	290
Hashlib	291
Secrets	293
Random numbers	293

Token generation	294
Digest comparison	296
HMAC	297
JSON Web Tokens	297
Registered claims	300
Time-related claims	300
Auth-related claims	302
Using asymmetric (public-key) algorithms	303
Useful references	304
Summary	305
Chapter 10: Concurrent Execution	307
Concurrency versus parallelism	308
Threads and processes – an overview	308
Quick anatomy of a thread	309
Killing threads	309
Context-switching	310
The Global Interpreter Lock	311
Race conditions and deadlocks	311
Race conditions	312
Scenario A – race condition not happening	312
Scenario B – race condition happening	312
Locks to the rescue	313
Scenario C – using a lock	313
Deadlocks	313
Quick anatomy of a process	314
Properties of a process	315
Multithreading or multiprocessing?	315
Concurrent execution in Python	316
Starting a thread	317
Starting a process	319
Stopping threads and processes	319
Stopping a process	321
Spawning multiple threads	321
Dealing with race conditions	322
A thread's local data	324
Thread and process communication	325
Thread communication	326
Sending events	327
Inter-process communication with queues	328
Thread and process pools	329
Using a process to add a timeout to a function	332
Case examples	334
Example one – concurrent mergesort	334
Single-thread mergesort	335
Single-thread multipart mergesort	336
Multithreaded mergesort	337

Multiprocess mergesort	338
Example two – batch sudoku-solver	340
What is Sudoku?	340
Implementing a sudoku-solver in Python	341
Solving sudoku with multiprocessing	346
Example three – downloading random pictures	349
Downloading random pictures with asyncio	351
Summary	355
Chapter 11: Debugging and Troubleshooting	357
Debugging techniques	359
Debugging with print	359
Debugging with a custom function	359
Inspecting the traceback	361
Using the Python debugger	364
Inspecting log files	367
Other techniques	369
Profiling	370
Assertions	370
Where to find information	370
Troubleshooting guidelines	371
Using console editors	371
Where to inspect	371
Using tests to debug	372
Monitoring	372
Summary	372
Chapter 12: GUIs and Scripts	375
First approach – scripting	377
The imports	378
Parsing arguments	379
The business logic	381
Second approach – a GUI application	385
The imports	388
The layout logic	388
The business logic	392
Fetching the web page	393
Saving the images	395
Alerting the user	398
How can we improve the application?	399
Where do we go from here?	400
The turtle module	401
wxPython, PyQt, and PyGTK	401
The principle of least astonishment	402
Threading considerations	402
Summary	403

Chapter 13: Data Science	405
IPython and Jupyter Notebook	406
Installing the required libraries	409
Using Anaconda	409
Starting a Notebook	409
Dealing with data	410
Setting up the Notebook	410
Preparing the data	411
Cleaning the data	415
Creating the DataFrame	417
Unpacking the campaign name	420
Unpacking the user data	422
Cleaning everything up	426
Saving the DataFrame to a file	427
Visualizing the results	428
Where do we go from here?	435
Summary	437
Chapter 14: Web Development	439
What is the web?	439
How does the web work?	440
The Django web framework	441
Django design philosophy	441
The model layer	442
The view layer	443
The template layer	443
The Django URL dispatcher	444
Regular expressions	444
A regex website	445
Setting up Django	445
Starting the project	445
Creating users	447
Adding the Entry model	448
Customizing the admin panel	450
Creating the form	452
Writing the views	454
The home view	454
The entry list view	455
The form view	457
Tying up URLs and views	459
Writing the templates	461
The future of web development	467
Writing a Flask view	468
Building a JSON quote server in Falcon	470
Summary	472
Farewell	473

Other Books You May Enjoy	475
Index	479
