

Modern Python Cookbook

Second Edition

133 recipes to develop flawless and expressive programs
in Python 3.8

Steven F. Lott



BIRMINGHAM - MUMBAI

Table of Contents

Preface	v
Chapter 1: Numbers, Strings, and Tuples	1
Working with large and small integers	2
Choosing between float, decimal, and fraction	4
Choosing between true division and floor division	11
Rewriting an immutable string	14
String parsing with regular expressions	20
Building complex strings with f-strings	24
Building complicated strings from lists of characters	28
Using the Unicode characters that aren't on our keyboards	30
Encoding strings – creating ASCII and UTF-8 bytes	33
Decoding bytes – how to get proper characters from some bytes	35
Using tuples of items	38
Using NamedTuples to simplify item access in tuples	42
Chapter 2: Statements and Syntax	45
Writing Python script and module files – syntax basics	46
Writing long lines of code	51
Including descriptions and documentation	56
Writing better RST markup in docstrings	61
Designing complex if...elif chains	67
Saving intermediate results with the := "walrus"	71
Avoiding a potential problem with break statements	74
Leveraging exception matching rules	79
Avoiding a potential problem with an except: clause	82
Concealing an exception root cause	84
Managing a context using the with statement	87
Chapter 3: Function Definitions	91
Function parameters and type hints	92
Designing functions with optional parameters	97
Designing type hints for optional parameters	102
Using super flexible keyword parameters	105
Forcing keyword-only arguments with the * separator	109
Defining position-only parameters with the / separator	114
Writing hints for more complex types	116
Picking an order for parameters based on partial functions	121
Writing clear documentation strings with RST markup	126

Designing recursive functions around Python's stack limits	131
Writing testable scripts with the script-library switch	136
Chapter 4: Built-In Data Structures Part 1: Lists and Sets	141
Choosing a data structure	142
Building lists – literals, appending, and comprehensions	146
Slicing and dicing a list	153
Deleting from a list – deleting, removing, popping, and filtering	158
Writing list-related type hints	164
Reversing a copy of a list	168
Building sets – literals, adding, comprehensions, and operators	171
Removing items from a set – remove(), pop(), and difference	178
Writing set-related type hints	181
Chapter 5: Built-In Data Structures Part 2: Dictionaries	187
Creating dictionaries – inserting and updating	188
Removing from dictionaries – the pop() method and the del statement	193
Controlling the order of dictionary keys	198
Writing dictionary-related type hints	202
Understanding variables, references, and assignment	206
Making shallow and deep copies of objects	209
Avoiding mutable default values for function parameters	213
Chapter 6: User Inputs and Outputs	219
Using the features of the print() function	220
Using input() and getpass() for user input	225
Debugging with f'{value=}' strings	232
Using argparse to get command-line input	234
Using cmd to create command-line applications	241
Using the OS environment settings	246
Chapter 7: Basics of Classes and Objects	251
Using a class to encapsulate data and processing	252
Essential type hints for class definitions	257
Designing classes with lots of processing	261
Using typing.NamedTuple for immutable objects	268
Using dataclasses for mutable objects	271
Using frozen dataclasses for immutable objects	275
Optimizing small objects with __slots__	278
Using more sophisticated collections	282
Extending a built-in collection – a list that does statistics	286
Using properties for lazy attributes	290
Creating contexts and context managers	296
Managing multiple contexts with multiple resources	301

Chapter 8: More Advanced Class Design	307
Choosing between inheritance and composition – the "is-a" question	308
Separating concerns via multiple inheritance	317
Leveraging Python's duck typing	324
Managing global and singleton objects	328
Using more complex structures – maps of lists	334
Creating a class that has orderable objects	339
Improving performance with an ordered collection	345
Deleting from a list of complicated objects	352
Chapter 9: Functional Programming Features	359
Introduction	359
Writing generator functions with the yield statement	361
Applying transformations to a collection	369
Using stacked generator expressions	375
Picking a subset – three ways to filter	386
Summarizing a collection – how to reduce	392
Combining the map and reduce transformations	397
Implementing "there exists" processing	404
Creating a partial function	409
Simplifying complex algorithms with immutable data structures	415
Writing recursive generator functions with the yield from statement	420
Chapter 10: Input/Output, Physical Format, and Logical Layout	427
Using pathlib to work with filenames	429
Replacing a file while preserving the previous version	438
Reading delimited files with the CSV module	443
Using dataclasses to simplify working with CSV files	449
Reading complex formats using regular expressions	453
Reading JSON and YAML documents	460
Reading XML documents	467
Reading HTML documents	475
Refactoring a .csv DictReader as a dataclass reader	483
Chapter 11: Testing	491
Test tool setup	492
Using docstrings for testing	494
Testing functions that raise exceptions	501
Handling common doctest issues	506
Unit testing with the unittest module	513
Combining unittest and doctest tests	522
Unit testing with the pytest module	527
Combining pytest and doctest tests	532

Testing things that involve dates or times	537
Testing things that involve randomness	543
Mocking external resources	550
Chapter 12: Web Services	561
Defining the card model	563
Using the Flask framework for RESTful APIs	569
Parsing the query string in a request	576
Making REST requests with urllib	583
Parsing the URL path	592
Parsing a JSON request	607
Implementing authentication for web services	620
Chapter 13: Application Integration: Configuration	641
Finding configuration files	642
Using YAML for configuration files	649
Using Python for configuration files	656
Using class-as-namespace for configuration	660
Designing scripts for composition	668
Using logging for control and audit output	677
Chapter 14: Application Integration: Combination	689
Combining two applications into one	690
Combining many applications using the Command Design Pattern	697
Managing arguments and configuration in composite applications	702
Wrapping and combining CLI applications	709
Wrapping a program and checking the output	719
Controlling complex sequences of steps	726
Chapter 15: Statistical Programming and Linear Regression	733
Using the built-in statistics library	734
Average of values in a counter	742
Computing the coefficient of a correlation	747
Computing regression parameters	753
Computing an autocorrelation	758
Confirming that the data is random – the null hypothesis	766
Locating outliers	774
Analyzing many variables in one pass	782
Other Books You May Enjoy	791
Index	795