

THIRD EDITION

Python Cookbook

David Beazley and Brian K. Jones

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

Table of Contents

| | |
|---|-----------|
| Preface..... | xii |
| 1. Data Structures & Algorithms..... | 1 |
| 1.1. Unpacking a Sequence Into Separate Variables | 1 |
| 1.2. Unpacking Elements From Iterables of Arbitrary Length | 3 |
| 1.3. Keeping the Last N Items | 5 |
| 1.4. Finding the Largest or Smallest N Items | 7 |
| 1.5. Implementing a Priority Queue | 8 |
| 1.6. How to Map Keys to Multiple Values in a Dictionary | 11 |
| 1.7. Keeping Dictionaries in Order | 12 |
| 1.8. Calculating With Dictionaries | 13 |
| 1.9. Finding Out What Two Dictionaries Have in Common | 15 |
| 1.10. Removing Duplicates From A Sequence While Maintaining Order | 17 |
| 1.11. Naming a Slice | 18 |
| 1.12. Determining the Most Frequently Occurring Items in a Sequence | 20 |
| 1.13. Sort a List of Dictionaries by a Common Key | 21 |
| 1.14. Sort Objects Without Native Comparison Support | 23 |
| 1.15. Grouping Records Together Based on a Field | 24 |
| 1.16. Filtering Sequence Elements | 26 |
| 1.17. Extracting a Subset of a Dictionary | 28 |
| 1.18. Mapping Names to Sequence Elements | 29 |
| 1.19. Transforming and Reducing Data at the Same Time | 32 |
| 1.20. Working with Multiple Mappings as a Single Mapping | 33 |
| 2. Strings and Text..... | 37 |
| 2.1. Splitting Strings On Any of Multiple Delimiters | 37 |
| 2.2. Matching Text at the Start or End of String | 38 |
| 2.3. Matching Strings Using Shell Wildcard Patterns | 40 |
| 2.4. Matching and Searching for Text Patterns | 42 |

| | |
|---|------------|
| 2.5. Searching and Replacing Text | 45 |
| 2.6. Case-Insensitive Search and Replace | 47 |
| 2.7. Specifying a Regular Expression for the Shortest Match | 48 |
| 2.8. Writing a Regular Expression For Multiline Patterns | 49 |
| 2.9. Normalizing Unicode Text to a Standard Representation | 50 |
| 2.10. Working with Unicode Characters in Regular Expressions | 52 |
| 2.11. Stripping Unwanted Characters From Strings | 53 |
| 2.12. Sanitizing and Cleaning up Text | 54 |
| 2.13. Aligning Text Strings | 57 |
| 2.14. Combining and Concatenating Strings | 58 |
| 2.15. Variable Interpolation in Strings | 61 |
| 2.16. Reformatting Text to Fixed Number of Columns | 64 |
| 2.17. Handling HTML and XML Entities in Text | 65 |
| 2.18. Tokenizing Text | 66 |
| 2.19. Writing a Simple Recursive Descent Parser | 69 |
| 2.20. Performing Text Operations on Byte Strings | 78 |
| 3. Numbers, Dates and Times..... | 83 |
| 3.1. Rounding Numerical Values | 83 |
| 3.2. Accurate Decimal Calculations | 84 |
| 3.3. Formatting Numbers for Output | 87 |
| 3.4. Binary, Octal, and Hexadecimal Integers | 88 |
| 3.5. Packing and Unpacking Large Integers From Bytes | 90 |
| 3.6. Complex Valued Math | 92 |
| 3.7. Infinity and NaNs | 93 |
| 3.8. Calculating with Fractions | 95 |
| 3.9. Calculating with Large Numerical Arrays | 96 |
| 3.10. Matrices and Linear Algebra | 100 |
| 3.11. Picking Things at Random | 101 |
| 3.12. Convert Days to Seconds, and Other Basic Time Conversions | 103 |
| 3.13. Determining Last Friday's Date | 105 |
| 3.14. Finding the Date Range For the Current Month | 107 |
| 3.15. Converting Strings Into Datetimes | 109 |
| 3.16. Date Manipulation Involving Timezones | 110 |
| 4. Iterators and Generators..... | 113 |
| 4.1. Manually Consuming an Iterator | 113 |
| 4.2. Delegating Iteration | 114 |
| 4.3. Creating New Iteration Patterns With Generators | 115 |
| 4.4. Easy Implementation of the Iterator Protocol | 117 |
| 4.5. Iterating in Reverse | 119 |
| 4.6. Generator Functions With Extra State | 120 |

| | |
|---|------------|
| 4.7. Taking a Slice of an Iterator | 122 |
| 4.8. Skipping the First Part of an Iterable | 123 |
| 4.9. Combinations and Permutations | 125 |
| 4.10. Iterate Over the Index-Value Pairs of a Sequence | 127 |
| 4.11. Iterating Over Multiple Sequences at Once | 129 |
| 4.12. Iterating on Items in Separate Containers | 131 |
| 4.13. Creating Data Processing Pipelines | 132 |
| 4.14. How to Flatten a Nested Sequence | 135 |
| 4.15. Iterating in Sorted Order Over Merged Sorted Iterables | 137 |
| 4.16. Replacing Infinite While-Loops with an Iterator | 138 |
| 5. Files and I/O..... | 141 |
| 5.1. Reading and Writing Text Data | 141 |
| 5.2. Printing to a File | 144 |
| 5.3. Printing With a Different Separator or Line Ending | 144 |
| 5.4. Reading and Writing Binary Data | 145 |
| 5.5. Writing to a File That Doesn't Already Exist | 147 |
| 5.6. Performing I/O Operations on a String | 148 |
| 5.7. Reading and Writing Compressed Data Files | 149 |
| 5.8. Iterating Over Fixed-Sized Records | 151 |
| 5.9. Reading Binary Data into a Mutable Buffer | 152 |
| 5.10. Memory Mapped Files | 153 |
| 5.11. Manipulating Pathnames | 156 |
| 5.12. Testing for the Existence of a File | 157 |
| 5.13. Getting a Directory Listing | 158 |
| 5.14. Bypassing Filename Encoding | 160 |
| 5.15. Printing Bad Filenames | 161 |
| 5.16. Adding or Changing the Encoding of an Already Open File | 163 |
| 5.17. Writing Bytes to a Text File | 165 |
| 5.18. Wrapping an Existing File Descriptor as a File Object | 166 |
| 5.19. Making Temporary Files and Directories | 167 |
| 5.20. Communicating with Serial Ports | 170 |
| 5.21. Serializing Python Objects | 171 |
| 6. Data Encoding and Processing..... | 175 |
| 6.1. Reading and Writing CSV Data | 175 |
| 6.2. Reading and Writing JSON Data | 179 |
| 6.3. Parsing Simple XML Data | 183 |
| 6.4. Incremental Parsing of Huge XML Files | 186 |
| 6.5. Turning a Dictionary into XML | 189 |
| 6.6. Parsing, Modifying, and Rewriting XML | 191 |
| 6.7. Parsing XML Documents with Namespaces | 193 |

| | |
|---|------------|
| 6.8. Interacting with a Relational Database | 195 |
| 6.9. Decoding and Encoding Hexadecimal Digits | 197 |
| 6.10. Decoding and Encoding Base64 | 199 |
| 6.11. Reading and Writing Binary Arrays of Structures | 199 |
| 6.12. Reading Nested and Variable Sized Binary Structures | 203 |
| 6.13. Summarizing Data and Performing Statistics | 214 |
| 7. Functions..... | 217 |
| 7.1. Functions That Accept Any Number of Arguments | 217 |
| 7.2. Functions That Only Accept Keyword Arguments | 218 |
| 7.3. Discussion | 219 |
| 7.4. Attaching Informational Metadata to Function Arguments | 220 |
| 7.5. Discussion | 220 |
| 7.6. Returning Multiple Values From a Function | 221 |
| 7.7. Functions With Default Arguments | 222 |
| 7.8. Defining Anonymous or Inline Functions | 224 |
| 7.9. Capturing Variables in Anonymous Functions | 225 |
| 7.10. Making an N-Argument Callable Work as a Callable With Fewer Arguments | 227 |
| 7.11. Replacing Single Method Classes with Functions | 231 |
| 7.12. Carrying Extra State with Callback Functions | 232 |
| 7.13. Inlining Callback Functions | 235 |
| 7.14. Accessing Variables Defined Inside a Closure | 238 |
| 8. Classes and Objects..... | 243 |
| 8.1. Changing the String Representation of Instances | 243 |
| 8.2. Customizing String Formatting | 245 |
| 8.3. Making Objects Support the Context-Management Protocol | 246 |
| 8.4. Saving Memory When Creating a Large Number of Instances | 248 |
| 8.5. How to Encapsulate Names in a Class | 249 |
| 8.6. Creating Managed Attributes | 251 |
| 8.7. Calling a Method on a Parent Class | 255 |
| 8.8. Extending a Property in a Subclass | 260 |
| 8.9. Creating a New Kind of Class or Instance Attribute | 264 |
| 8.10. Lazily Computed Properties | 267 |
| 8.11. Simplified Initialization of Data Structures | 270 |
| 8.12. How to Define an Interface or Abstract Base Class | 273 |
| 8.13. Implementing a Data Model or Type System | 276 |
| 8.14. Implementing Custom Containers | 282 |
| 8.15. Delegation and Proxies | 286 |
| 8.16. How to Define More Than One Constructor in a Class | 290 |
| 8.17. Creating an Instance Without Invoking <code>__init__</code> | 292 |

| | |
|---|------------|
| 8.18. Extending Classes with Mixins | 293 |
| 8.19. Implementing Stateful Objects or State Machines | 298 |
| 8.20. Calling a Method on an Object Given the Name as a String | 304 |
| 8.21. Implementing the Visitor Pattern | 305 |
| 8.22. Implementing the Visitor Pattern Without Recursion | 310 |
| 8.23. Managing Memory in Cyclic Data Structures | 316 |
| 8.24. Making Classes Support Comparison Operations | 320 |
| 8.25. Creating Cached Instances | 322 |
| 9. Metaprogramming..... | 327 |
| 9.1. Putting a Wrapper Around a Function | 327 |
| 9.2. Preserving Function Metadata When Writing Decorators | 329 |
| 9.3. Unwrapping a Decorator | 331 |
| 9.4. Defining a Decorator That Takes Arguments | 332 |
| 9.5. Defining a Decorator With User Adjustable Attributes | 334 |
| 9.6. Defining a Decorator That Takes an Optional Argument | 337 |
| 9.7. Enforcing Type Checking on a Function Using a Decorator | 339 |
| 9.8. Defining Decorators as Part of a Class | 343 |
| 9.9. Defining Decorators as Classes | 345 |
| 9.10. Applying Decorators to Class and Static Methods | 347 |
| 9.11. Decorators That Add Arguments to Wrapped Functions | 350 |
| 9.12. Using Decorators to Patch Class Definitions | 352 |
| 9.13. Using a Metaclass to Control Instance Creation | 354 |
| 9.14. Capturing Class Attribute Definition Order | 357 |
| 9.15. Defining a Metaclass That Takes Optional Arguments | 360 |
| 9.16. Enforcing an Argument Signature on <code>*args</code> and <code>**kwargs</code> | 362 |
| 9.17. Enforcing Coding Conventions in Classes | 365 |
| 9.18. Defining Classes Programmatically | 368 |
| 9.19. Initializing Class Members at Definition Time | 371 |
| 9.20. Multiple Dispatch with Function Annotations | 373 |
| 9.21. Avoiding Repetitive Property Methods | 380 |
| 9.22. Defining Context Managers the Easy Way | 382 |
| 9.23. Executing Code With Local Side Effects | 383 |
| 9.24. Parsing and Analyzing Python Source | 386 |
| 9.25. Disassembling Python Byte Code | 390 |
| 10. Modules and Packages..... | 395 |
| 10.1. Making a Hierarchical Package of Modules | 395 |
| 10.2. Controlling the Import of Everything | 397 |
| 10.3. Importing Package Submodules Using Relative Names | 397 |
| 10.4. Splitting a Module Into Multiple Files | 399 |

| | |
|---|------------|
| 10.5. Making Separate Directories of Code Import Under a Common Namespace | 402 |
| 10.6. Reloading Modules | 404 |
| 10.7. Making a Directory or Zip File Runnable as a Main Script | 406 |
| 10.8. Reading Data Files Within a Package | 407 |
| 10.9. Adding Directories to <code>sys.path</code> | 408 |
| 10.10. Importing Modules Using a Name Given in a String | 409 |
| 10.11. Loading Modules From a Remote Machine Using Import Hooks | 410 |
| 10.12. Patching Modules On Import | 427 |
| 10.13. Installing Packages Just For Yourself | 430 |
| 10.14. Creating a New Python Environment | 430 |
| 10.15. Distributing Packages | 432 |
| 11. Network & Web Programming..... | 435 |
| 11.1. Interacting with HTTP Services as a Client | 435 |
| 11.2. Creating a TCP Server | 439 |
| 11.3. Creating a UDP Server | 442 |
| 11.4. Generating a Range of IP Addresses From a CIDR Address | 444 |
| 11.5. Creating a Simple REST-based Interface | 446 |
| 11.6. Simple Remote Procedure Call with XML-RPC | 451 |
| 11.7. Simple Communication Between Interpreters | 454 |
| 11.8. Implementing Remote Procedure Call | 456 |
| 11.9. Simple Authentication of Clients | 459 |
| 11.10. Adding SSL to Network Services | 461 |
| 11.11. Passing a Socket File Descriptor Between Processes | 468 |
| 11.12. Event-Driven I/O Explained | 473 |
| 11.13. Zero-Copy Sending and Receiving of Large Arrays | 479 |
| 12. Concurrency..... | 481 |
| 12.1. How to Start and Stop Threads | 481 |
| 12.2. How to Determine if a Thread Has Started | 484 |
| 12.3. How to Communicate Between Threads | 487 |
| 12.4. How to Lock Critical Sections | 493 |
| 12.5. Locking With Deadlock Avoidance | 496 |
| 12.6. Storing Thread-Specific State | 500 |
| 12.7. How to Create a Thread Pool | 501 |
| 12.8. Simple Parallel Programming | 505 |
| 12.9. How to Stop Worrying About the GIL | 509 |
| 12.10. Defining an Actor Task | 512 |
| 12.11. Implementing Publish/Subscribe Messaging | 516 |
| 12.12. Using Generators as an Alternative to Threads | 520 |
| 12.13. Polling Multiple Thread Queues | 527 |

| | |
|--|------------|
| 12.14. Launching a Daemon Process on Unix | 530 |
| 13. Utility Scripting and System Administration..... | 535 |
| 13.1. Accept Script Input Via Redirection, Pipes, or Input Files | 535 |
| 13.2. Terminating a Program With an Error Message | 536 |
| 13.3. Parsing Command Line Options | 537 |
| 13.4. Prompting For A Password At Runtime | 540 |
| 13.5. Getting the Terminal Size | 541 |
| 13.6. Executing an External Command and Getting Its Output | 541 |
| 13.7. Copying or Moving Files and Directories | 543 |
| 13.8. Creating and Unpacking Archives | 545 |
| 13.9. Finding Files By Name | 546 |
| 13.10. Reading Configuration Files | 548 |
| 13.11. Simple Logging for Scripts | 551 |
| 13.12. Adding Logging to Libraries | 554 |
| 13.13. Making a Stopwatch Timer | 555 |
| 13.14. Putting Limits on Memory and CPU Usage | 557 |
| 13.15. Launching a Web Browser | 558 |
| 14. Testing, Debugging, and Exceptions..... | 561 |
| 14.1. Testing Output Sent to Stdout | 561 |
| 14.2. Patching Objects in Unit Tests | 563 |
| 14.3. Testing For Exceptional Conditions in Unit Tests | 566 |
| 14.4. Logging Test Output to a File | 568 |
| 14.5. Skipping or Anticipating Test Failures | 569 |
| 14.6. Handling Multiple Exceptions | 570 |
| 14.7. Catching All Exceptions | 572 |
| 14.8. Creating Custom Exceptions | 574 |
| 14.9. Raising an Exception in Response to Another Exception | 576 |
| 14.10. Reraising the Last Exception | 578 |
| 14.11. Issuing Warning Messages | 579 |
| 14.12. Basic Debugging | 581 |
| 14.13. Profiling and Timing Your Program | 583 |
| 14.14. Making Your Programs Run Faster | 586 |
| 15. C Extensions..... | 591 |
| 15.1. Accessing C Code Using ctypes | 593 |
| 15.2. Writing a Simple C Extension Module | 599 |
| 15.3. Writing an Extension Function That Operates on Arrays | 603 |
| 15.4. Managing Opaque Pointers in C Extension Modules | 606 |
| 15.5. Defining and Exporting C APIs From Extension Modules | 608 |
| 15.6. Calling Python From C | 613 |

| | |
|--|------------|
| 15.7. Releasing the GIL in C Extensions | 618 |
| 15.8. Mixing Threads from C and Python | 619 |
| 15.9. Wrapping C code with Swig | 620 |
| 15.10. Wrapping Existing C Code with Cython | 625 |
| 15.11. Using Cython to Write High Performance Array Operations | 632 |
| 15.12. Turning a Function Pointer Into a Callable | 636 |
| 15.13. Passing NULL-Terminated Strings to C Libraries | 638 |
| 15.14. Passing Unicode Strings to C Libraries | 642 |
| 15.15. Converting C Strings to Python | 646 |
| 15.16. Working with C Strings of Dubious Encoding | 648 |
| 15.17. Passing Filenames to C Extensions | 651 |
| 15.18. Passing Open Files to C Extensions | 652 |
| 15.19. Reading File-Like Objects From C | 653 |
| 15.20. Consuming an Iterable from C | 656 |
| 15.21. Diagnosing Segmentation Faults | 657 |
| Index..... | 661 |