Table of Contents

Foreword xiii

Preface xvii

1. Python Shortcuts 1

1.1 Swapping Values Without Using a Temporary Variable 3
1.2 Constructing a Dictionary Without Excessive Quoting 4
1.3 Getting a Value from a Dictionary 6
1.4 Adding an Entry to a Dictionary 7
1.5 Associating Multiple Values with Each Key in a Dictionary 8
1.6 Dispatching Using a Dictionary 11
1.7 Collecting a Bunch of Named Items 13
1.8 Finding the Intersection of Two Dictionaries 15
1.9 Assigning and Testing with One Statement 18
1.10 Using List Comprehensions Instead of map and filter 20
1.11 Unzipping Simple List-Like Objects 21
1.12 Flattening a Nested Sequence 23
1.13 Looping in Parallel over Index and Sequence Items 25
1.14 Looping Through Multiple Lists 27
1.15 Spanning a Range Defined by Floats 28
1.16 Transposing Two-Dimensional Arrays 30
1.17 Creating Lists of Lists Without Sharing References 31

2. Searching and Sorting 34

2.1 Sorting a Dictionary 38
2.2 Processing Selected Pairs of Structured Data Efficiently 40
2.3 Sorting While Guaranteeing Sort Stability 42
2.4 *Sorting by One Field, Then by Another* 43

2.5 Looking for Items in a Sorted Sequence Using Binary Search 46
2.6 Sorting a List of Objects by an Attribute of the Objects 47
2.7 Sorting by Item or by Attribute 49
2.8 Selecting Random Elements from a List Without Repetition 52
2.9 Performing Frequent Membership Tests on a Sequence 54
2.10 Finding the Deep Index of an Item in an Embedded Sequence 57
2.11 Showing Off Quicksort in Three Lines 59
2.12 Sorting Objects Using SQL's ORDER BY Syntax 61

3. Text 63
3.1 Processing a String One Character at a Time 69
3.2 Testing if an Object Is String-Like 70
3.3 Aligning Strings 71
3.4 Trimming Space from the Ends of a String 71
3.5 Combining Strings 72
3.6 Checking Whether a String Contains a Set of Characters 74
3.7 Filtering a String for a Set of Characters 76
3.8 Controlling Case 78
3.9 Reversing a String by Words or Characters 79
3.10 Accessing Substrings 81
3.11 Changing the Indentation of a Multiline String 83
3.12 Testing Whether a String Represents an Integer 85
3.13 Expanding and Compressing Tabs 86
3.14 Replacing Multiple Patterns in a Single Pass 88
3.15 Converting Between Different Naming Conventions 91
3.16 Converting Between Characters and Values 93
3.17 Converting Between Unicode and Plain Strings 94
3.18 Printing Unicode Characters to Standard Output 96
3.19 Dispatching Based on Pattern Matches 97
3.20 Evaluating Code Inside Strings 99
3.21 Replacing Python Code with the Results of Executing That Code 100
3.22 Module: Yet Another Python Templating Utility (YAPTU) 104
3.23 Module: Roman Numerals 109

4. Files 113
4.1 Reading from a File 117
4.2 Writing to a File 119
4.3 Searching and Replacing Text in a File 121
4.4 Reading a Particular Line from a File 122

4.5 Retrieving a Line at Random from a File of Unknown Size 123
4.6 Counting Lines in a File 125
4.7 Processing Every Word in a File 127
4.8 Reading a Text File by Paragraphs 129
4.9 Reading Lines with Continuation Characters 132
4.10 Reading Data from ZIP Files 134
4.11 Reading INI Configuration Files 135
4.12 Sending Binary Data to Standard Output Under Windows 137
4.13 Using Random-Access Input/Output 137
4.14 Updating a Random-Access File 138
4.15 Splitting a Path into All of Its Parts 140
4.16 Treating Pathnames as Objects 141
4.17 Creating Directories Including Necessary Parent Directories 143
4.18 Walking Directory Trees 144
4.19 Swapping One File Extension for Another Throughout a Directory Tree 145
4.20 Finding a File Given an Arbitrary Search Path 146
4.21 Finding a File on the Python Search Path 147
4.22 Dynamically Changing the Python Search Path 148
4.23 Computing Directory Sizes in a Cross-Platform Way 150
4.24 File Locking Using a Cross-Platform API 152
4.25 Versioning Filenames 154
4.26 Module: Versioned Backups 156

5. Object-Oriented Programming 160
5.1 Overriding a Built-In Method 165
5.2 Getting All Members of a Class Hierarchy 167
5.3 Calling a Superclass __init__ Method if It Exists 169
5.4 Calling a Superclass Implementation of a Method 172
5.5 Implementing Properties 174
5.6 Implementing Static Methods 176
5.7 Implementing Class Methods 177
5.8 Delegating Automatically as an Alternative to Inheritance 179
5.9 Decorating an Object with Print-Like Methods 182
5.10 Checking if an Object Has Necessary Attributes 183
5.11 Making a Fast Copy of an Object 186
5.12 Adding Methods to a Class at Runtime 188
5.13 Modifying the Class Hierarchy of an Instance 189

5.14 Keeping References to Bound Methods Without Inhibiting Garbage Collection 191
5.15 Defining Constants 193
5.16 Managing Options 194
5.17 Implementing a Set Class 198
5.18 Implementing a Ring Buffer 201
5.19 Implementing a Collection 202
5.20 Delegating Messages to Multiple Objects 203
5.21 Implementing the Singleton Design Pattern 206
5.22 Avoiding the Singleton Design Pattern with the Borg Idiom 208
5.23 Implementing the Null Object Design Pattern 211

6. Threads, Processes, and Synchronization 214
6.1 Storing Per-Thread Information 217
6.2 Terminating a Thread 218
6.3 Allowing Multithreaded Read Access While Maintaining a Write Lock 220
6.4 Running Functions in the Future 222
6.5 Synchronizing All Methods in an Object 225
6.6 Capturing the Output and Error Streams from a Unix Shell Command 227
6.7 Forking a Daemon Process on Unix 229
6.8 Determining if Another Instance of a Script Is Already Running in Windows 231
6.9 Processing Windows Messages Using MsgWaitForMultipleObjects 233

7. System Administration 236
7.1 Running a Command Repeatedly 237
7.2 Generating Random Passwords 238
7.3 Generating Non-Totally Random Passwords 240
7.4 Checking the Status of a Unix Network Interface 242
7.5 Calculating Apache Hits per IP Address 243
7.6 Calculating the Rate of Client Cache Hits on Apache 245
7.7 Manipulating the Environment on Windows NT/2000/XP 246
7.8 Checking and Modifying the Set of Tasks Windows Automatically Runs at Logon 248
7.9 Examining the Microsoft Windows Registry for a List of Name Server Addresses 250
7.10 Getting Information About the Current User on Windows NT/2000 253
7.11 Getting the Windows Service Name from Its Long Name 254

7.12 Manipulating Windows Services 255
7.13 Impersonating Principals on Windows 257
7.14 Changing a Windows NT Password Using ADSI 258
7.15 Working with Windows Scripting Host (WSH) from Python 260
7.16 Displaying Decoded Hotkeys for Shortcuts in Windows 262

8. Databases and Persistence 264
8.1 Serializing Data Using the marshal Module 267
8.2 Serializing Data Using the pickle and cPickle Modules 269
8.3 Using the cPickle Module on Classes and Instances 272
8.4 Mutating Objects with shelve 275
8.5 Accessing a MySQL Database 276
8.6 Storing a BLOB in a MySQL Database 278
8.7 Storing a BLOB in a PostgreSQL Database 280
8.8 Generating a Dictionary Mapping from Field Names to Column Numbers 281
8.9 Using dtuple for Flexible Access to Query Results 283
8.10 Pretty-Printing the Contents of Database Cursors 285
8.11 Establishing Database Connections Lazily 287
8.12 Accessing a JDBC Database from a Jython Servlet 290
8.13 Module: jet2sql—Creating a SQL DDL from an Access Database 293

9. User Interfaces 300
9.1 Avoiding lambda in Writing Callback Functions 302
9.2 Creating Menus with Tkinter 303
9.3 Creating Dialog Boxes with Tkinter 305
9.4 Supporting Multiple Values per Row in a Tkinter Listbox 306
9.5 Embedding Inline GIFs Using Tkinter 309
9.6 Combining Tkinter and Asynchronous I/O with Threads 310
9.7 Using a wxPython Notebook with Panels 313
9.8 Giving the User Unobtrusive Feedback During Data Entry with Qt 315
9.9 Building GUI Solutions Independent of the Specific GUI Toolkit 317
9.10 Creating Color Scales 319
9.11 Using Publish/Subscribe Broadcasting to Loosen the Coupling Between GUI and Business Logic Systems 320
9.12 Module: Building GTK GUIs Interactively 324

10. Network Programming 328
10.1 Writing a TCP Client 330
10.2 Writing a TCP Server 332

10.3 Passing Messages with Socket Datagrams 333
10.4 Finding Your Own Name and Address 335
10.5 Converting IP Addresses 336
10.6 Grabbing a Document from the Web 338
10.7 Being an FTP Client 339
10.8 Sending HTML Mail 340
10.9 Sending Multipart MIME Email 342
10.10 Bundling Files in a MIME Message 344
10.11 Unpacking a Multipart MIME Message 346
10.12 Module: PyHeartBeat—Detecting Inactive Computers 348
10.13 Module: Interactive POP3 Mailbox Inspector 352
10.14 Module: Watching for New IMAP Mail Using a GUI 355

11. Web Programming 358
11.1 Testing Whether CGI Is Working 359
11.2 Writing a CGI Script 361
11.3 Using a Simple Dictionary for CGI Parameters 363
11.4 Handling URLs Within a CGI Script 364
11.5 Resuming the HTTP Download of a File 366
11.6 Stripping Dangerous Tags and Javascript from HTML 368
11.7 Running a Servlet with Jython 370
11.8 Accessing Netscape Cookie Information 371
11.9 Finding an Internet Explorer Cookie 374
11.10 Module: Fetching Latitude/Longitude Data from the Web 376

12. Processing XML 379
12.1 Checking XML Well-Formedness 381
12.2 Counting Tags in a Document 382
12.3 Extracting Text from an XML Document 384
12.4 Transforming an XML Document Using XSLT 385
12.5 Transforming an XML Document Using Python 386
12.6 Parsing an XML File with xml.parsers.expat 387
12.7 Converting Ad-Hoc Text into XML Markup 389
12.8 Normalizing an XML Document 391
12.9 Controlling XSLT Stylesheet Loading 392
12.10 Autodetecting XML Encoding 395
12.11 Module: XML Lexing (Shallow Parsing) 397
12.12 Module: Converting a List of Equal-Length Lists into XML 400

C

13. Distributed Programming 404
13.1 Making an XML-RPC Method Call 406
13.2 Serving XML-RPC Requests 407
13.3 Using XML-RPC with Medusa 409
13.4 Writing a Web Service That Supports Both XML-RPC and SOAP 411
13.5 Implementing a CORBA Client and Server 414
13.6 Performing Remote Logins Using telnetlib 416
13.7 Using Publish/Subscribe in a Distributed Middleware Architecture 419
13.8 Using Request/Reply in a Distributed Middleware Architecture 422

14. Debugging and Testing 425
14.1 Reloading All Loaded Modules 426
14.2 Tracing Expressions and Comments in Debug Mode 427
14.3 Wrapping Tracebacks in HTML 430
14.4 Getting More Information from Tracebacks 431
14.5 Starting the Debugger Automatically After an Uncaught Exception 434
14.6 Logging and Tracing Across Platforms 436
14.7 Determining the Name of the Current Function 439
14.8 Introspecting the Call Stack with Older Versions of Python 441
14.9 Debugging the Garbage-Collection Process 442
14.10 Tracking Instances of Particular Classes 443

15. Programs About Programs 446
15.1 Colorizing Python Source Using the Built-in Tokenizer 451
15.2 Importing a Dynamically Generated Module 454
15.3 Importing from a Module Whose Name Is Determined at Runtime 456
15.4 Importing Modules with Automatic End-of-Line Conversions 457
15.5 Simulating Enumerations in Python 459
15.6 Modifying Methods in Place 462
15.7 Associating Parameters with a Function (Currying) 463
15.8 Composing Functions 465
15.9 Adding Functionality to a Class 467
15.10 Adding a Method to a Class Instance at Runtime 469
15.11 Defining a Custom Metaclass to Control Class Behavior 471
15.12 Module: Allowing the Python Profiler to Profile C Modules 473

16. Extending and Embedding 477
16.1 Implementing a Simple Extension Type 479
16.2 Translating a Python Sequence into a C Array with the PySequence_Fast Protocol 483

16.3 Accessing a Python Sequence Item-by-Item with the Iterator Protocol 487
16.4 Returning None from a Python-Callable C Function 490
16.5 Coding the Methods of a Python Class in C 491
16.6 Implementing C Function Callbacks to a Python Function 493
16.7 Debugging Dynamically Loaded C Extensions with gdb 496
16.8 Debugging Memory Problems 498
16.9 Using SWIG-Generated Modules in a Multithreaded Environment 499

17. Algorithms 501
17.1 Testing if a Variable Is Defined 505
17.2 Evaluating Predicate Tests Across Sequences 508
17.3 Removing Duplicates from a Sequence 509
17.4 Removing Duplicates from a Sequence While Maintaining Sequence Order 511
17.5 Simulating the Ternary Operator in Python 515
17.6 Counting Items and Sorting by Incidence (Histograms) 517
17.7 Memoizing (Caching) the Return Values of Functions 519
17.8 Looking Up Words by Sound Similarity 521
17.9 Computing Factorials with lambda 523
17.10 Generating the Fibonacci Sequence 524
17.11 Wrapping an Unbounded Iterator to Restrict Its Output 526
17.12 Operating on Iterators 528
17.13 Rolling Dice 531
17.14 Implementing a First-In First-Out Container 532
17.15 Modeling a Priority Queue 534
17.16 Converting Numbers to Rationals via Farey Fractions 535
17.17 Evaluating a Polynomial 537
17.18 Module: Finding the Convex Hull of a Set of 2D Points 538
17.19 Module: Parsing a String into a Date/Time Object Portably 541

List of Contributors 549

Index 553