
Table of Contents

Preface	xvii
1. Text	1
1.1 Processing a String One Character at a Time	7
1.2 Converting Between Characters and Numeric Codes	8
1.3 Testing Whether an Object Is String-like	9
1.4 Aligning Strings	11
1.5 Trimming Space from the Ends of a String	12
1.6 Combining Strings	12
1.7 Reversing a String by Words or Characters	15
1.8 Checking Whether a String Contains a Set of Characters	16
1.9 Simplifying Usage of Strings' translate Method	20
1.10 Filtering a String for a Set of Characters	22
1.11 Checking Whether a String Is Text or Binary	25
1.12 Controlling Case	26
1.13 Accessing Substrings	28
1.14 Changing the Indentation of a Multiline String	31
1.15 Expanding and Compressing Tabs	32
1.16 Interpolating Variables in a String	35
1.17 Interpolating Variables in a String in Python 2.4	36
1.18 Replacing Multiple Patterns in a Single Pass	38
1.19 Checking a String for Any of Multiple Endings	41
1.20 Handling International Text with Unicode	43
1.21 Converting Between Unicode and Plain Strings	45
1.22 Printing Unicode Characters to Standard Output	48
1.23 Encoding Unicode Data for XML and HTML	49
1.24 Making Some Strings Case-Insensitive	52
1.25 Converting HTML Documents to Text on a Unix Terminal	55

2. Files	58
2.1 Reading from a File	62
2.2 Writing to a File	66
2.3 Searching and Replacing Text in a File	67
2.4 Reading a Specific Line from a File	68
2.5 Counting Lines in a File	69
2.6 Processing Every Word in a File	72
2.7 Using Random-Access Input/Output	74
2.8 Updating a Random-Access File	75
2.9 Reading Data from zip Files	77
2.10 Handling a zip File Inside a String	79
2.11 Archiving a Tree of Files into a Compressed tar File	80
2.12 Sending Binary Data to Standard Output Under Windows	82
2.13 Using a C++-like iostream Syntax	83
2.14 Rewinding an Input File to the Beginning	84
2.15 Adapting a File-like Object to a True File Object	87
2.16 Walking Directory Trees	88
2.17 Swapping One File Extension for Another Throughout a Directory Tree	90
2.18 Finding a File Given a Search Path	91
2.19 Finding Files Given a Search Path and a Pattern	92
2.20 Finding a File on the Python Search Path	93
2.21 Dynamically Changing the Python Search Path	94
2.22 Computing the Relative Path from One Directory to Another	96
2.23 Reading an Unbuffered Character in a Cross-Platform Way	98
2.24 Counting Pages of PDF Documents on Mac OS X	99
2.25 Changing File Attributes on Windows	100
2.26 Extracting Text from OpenOffice.org Documents	101
2.27 Extracting Text from Microsoft Word Documents	102
2.28 File Locking Using a Cross-Platform API	103
2.29 Versioning Filenames	105
2.30 Calculating CRC-64 Cyclic Redundancy Checks	107
3. Time and Money	110
3.1 Calculating Yesterday and Tomorrow	116
3.2 Finding Last Friday	118
3.3 Calculating Time Periods in a Date Range	120
3.4 Summing Durations of Songs	121

3.5	Calculating the Number of Weekdays Between Two Dates	122
3.6	Looking up Holidays Automatically	124
3.7	Fuzzy Parsing of Dates	127
3.8	Checking Whether Daylight Saving Time Is Currently in Effect	129
3.9	Converting Time Zones	130
3.10	Running a Command Repeatedly	131
3.11	Scheduling Commands	133
3.12	Doing Decimal Arithmetic	135
3.13	Formatting Decimals as Currency	137
3.14	Using Python as a Simple Adding Machine	140
3.15	Checking a Credit Card Checksum	143
3.16	Watching Foreign Exchange Rates	144
4.	Python Shortcuts	146
4.1	Copying an Object	148
4.2	Constructing Lists with List Comprehensions	151
4.3	Returning an Element of a List If It Exists	153
4.4	Looping over Items and Their Indices in a Sequence	154
4.5	Creating Lists of Lists Without Sharing References	155
4.6	Flattening a Nested Sequence	157
4.7	Removing or Reordering Columns in a List of Rows	160
4.8	Transposing Two-Dimensional Arrays	161
4.9	Getting a Value from a Dictionary	163
4.10	Adding an Entry to a Dictionary	165
4.11	Building a Dictionary Without Excessive Quoting	166
4.12	Building a Dict from a List of Alternating Keys and Values	168
4.13	Extracting a Subset of a Dictionary	170
4.14	Inverting a Dictionary	171
4.15	Associating Multiple Values with Each Key in a Dictionary	173
4.16	Using a Dictionary to Dispatch Methods or Functions	175
4.17	Finding Unions and Intersections of Dictionaries	176
4.18	Collecting a Bunch of Named Items	178
4.19	Assigning and Testing with One Statement	180
4.20	Using printf in Python	183
4.21	Randomly Picking Items with Given Probabilities	184
4.22	Handling Exceptions Within an Expression	185
4.23	Ensuring a Name Is Defined in a Given Module	187

5. Searching and Sorting	190
5.1 Sorting a Dictionary	195
5.2 Sorting a List of Strings Case-Insensitively	196
5.3 Sorting a List of Objects by an Attribute of the Objects	198
5.4 Sorting Keys or Indices Based on the Corresponding Values	200
5.5 Sorting Strings with Embedded Numbers	203
5.6 Processing All of a List's Items in Random Order	204
5.7 Keeping a Sequence Ordered as Items Are Added	206
5.8 Getting the First Few Smallest Items of a Sequence	208
5.9 Looking for Items in a Sorted Sequence	210
5.10 Selecting the nth Smallest Element of a Sequence	212
5.11 Showing off quicksort in Three Lines	215
5.12 Performing Frequent Membership Tests on a Sequence	217
5.13 Finding Subsequences	220
5.14 Enriching the Dictionary Type with Ratings Functionality	222
5.15 Sorting Names and Separating Them by Initials	226
6. Object-Oriented Programming	229
6.1 Converting Among Temperature Scales	235
6.2 Defining Constants	238
6.3 Restricting Attribute Setting	240
6.4 Chaining Dictionary Lookups	242
6.5 Delegating Automatically as an Alternative to Inheritance	244
6.6 Delegating Special Methods in Proxies	247
6.7 Implementing Tuples with Named Items	250
6.8 Avoiding Boilerplate Accessors for Properties	252
6.9 Making a Fast Copy of an Object	254
6.10 Keeping References to Bound Methods Without Inhibiting Garbage Collection	256
6.11 Implementing a Ring Buffer	259
6.12 Checking an Instance for Any State Changes	262
6.13 Checking Whether an Object Has Necessary Attributes	266
6.14 Implementing the State Design Pattern	269
6.15 Implementing the "Singleton" Design Pattern	271
6.16 Avoiding the "Singleton" Design Pattern with the Borg Idiom	273
6.17 Implementing the Null Object Design Pattern	277
6.18 Automatically Initializing Instance Variables from <code>__init__</code> Arguments	280

6.19	Calling a Superclass <code>__init__</code> Method If It Exists	282
6.20	Using Cooperative Supercalls Concisely and Safely	285
7.	Persistence and Databases	288
7.1	Serializing Data Using the <code>marshal</code> Module	291
7.2	Serializing Data Using the <code>pickle</code> and <code>cPickle</code> Modules	293
7.3	Using Compression with Pickling	296
7.4	Using the <code>cPickle</code> Module on Classes and Instances	297
7.5	Holding Bound Methods in a Picklable Way	300
7.6	Pickling Code Objects	302
7.7	Mutating Objects with <code>shelve</code>	305
7.8	Using the Berkeley DB Database	307
7.9	Accesssing a MySQL Database	310
7.10	Storing a BLOB in a MySQL Database	312
7.11	Storing a BLOB in a PostgreSQL Database	313
7.12	Storing a BLOB in a SQLite Database	315
7.13	Generating a Dictionary Mapping Field Names to Column Numbers	316
7.14	Using <code>dtuple</code> for Flexible Access to Query Results	318
7.15	Pretty-Printing the Contents of Database Cursors	320
7.16	Using a Single Parameter-Passing Style Across Various DB API Modules	323
7.17	Using Microsoft Jet via ADO	325
7.18	Accessing a JDBC Database from a Jython Servlet	327
7.19	Using ODBC to Get Excel Data with Jython	330
8.	Debugging and Testing	332
8.1	Disabling Execution of Some Conditionals and Loops	333
8.2	Measuring Memory Usage on Linux	334
8.3	Debugging the Garbage-Collection Process	336
8.4	Trapping and Recording Exceptions	337
8.5	Tracing Expressions and Comments in Debug Mode	339
8.6	Getting More Information from Tracebacks	342
8.7	Starting the Debugger Automatically After an Uncaught Exception	345
8.8	Running Unit Tests Most Simply	346
8.9	Running Unit Tests Automatically	348
8.10	Using <code>doctest</code> with <code>unittest</code> in Python 2.4	350
8.11	Checking Values Against Intervals in Unit Testing	352

9. Processes, Threads, and Synchronization	355
9.1 Synchronizing All Methods in an Object	359
9.2 Terminating a Thread	362
9.3 Using a Queue.Queue as a Priority Queue	364
9.4 Working with a Thread Pool	366
9.5 Executing a Function in Parallel on Multiple Argument Sets	369
9.6 Coordinating Threads by Simple Message Passing	372
9.7 Storing Per-Thread Information	374
9.8 Multitasking Cooperatively Without Threads	378
9.9 Determining Whether Another Instance of a Script Is Already Running in Windows	380
9.10 Processing Windows Messages Using MsgWaitForMultipleObjects	381
9.11 Driving an External Process with popen	384
9.12 Capturing the Output and Error Streams from a Unix Shell Command	386
9.13 Forking a Daemon Process on Unix	388
10. System Administration	391
10.1 Generating Random Passwords	393
10.2 Generating Easily Remembered Somewhat-Random Passwords	394
10.3 Authenticating Users by Means of a POP Server	397
10.4 Calculating Apache Hits per IP Address	398
10.5 Calculating the Rate of Client Cache Hits on Apache	400
10.6 Spawning an Editor from a Script	401
10.7 Backing Up Files	403
10.8 Selectively Copying a Mailbox File	405
10.9 Building a Whitelist of Email Addresses From a Mailbox	406
10.10 Blocking Duplicate Mails	408
10.11 Checking Your Windows Sound System	410
10.12 Registering or Unregistering a DLL on Windows	411
10.13 Checking and Modifying the Set of Tasks Windows Automatically Runs at Login	412
10.14 Creating a Share on Windows	414
10.15 Connecting to an Already Running Instance of Internet Explorer	415
10.16 Reading Microsoft Outlook Contacts	416
10.17 Gathering Detailed System Information on Mac OS X	418

11. User Interfaces	422
11.1 Showing a Progress Indicator on a Text Console	424
11.2 Avoiding lambda in Writing Callback Functions	426
11.3 Using Default Values and Bounds with tkSimpleDialog Functions	427
11.4 Adding Drag and Drop Reordering to a Tkinter Listbox	428
11.5 Entering Accented Characters in Tkinter Widgets	430
11.6 Embedding Inline GIFs Using Tkinter	432
11.7 Converting Among Image Formats	434
11.8 Implementing a Stopwatch in Tkinter	437
11.9 Combining GUIs and Asynchronous I/O with Threads	439
11.10 Using IDLE's Tree Widget in Tkinter	443
11.11 Supporting Multiple Values per Row in a Tkinter Listbox	445
11.12 Copying Geometry Methods and Options Between Tkinter Widgets	448
11.13 Implementing a Tabbed Notebook for Tkinter	451
11.14 Using a wxPython Notebook with Panels	453
11.15 Implementing an ImageJ Plug-in in Jython	455
11.16 Viewing an Image from a URL with Swing and Jython	456
11.17 Getting User Input on Mac OS	456
11.18 Building a Python Cocoa GUI Programmatically	459
11.19 Implementing Fade-in Windows with IronPython	461
12. Processing XML	463
12.1 Checking XML Well-Formedness	465
12.2 Counting Tags in a Document	467
12.3 Extracting Text from an XML Document	468
12.4 Autodetecting XML Encoding	469
12.5 Converting an XML Document into a Tree of Python Objects	471
12.6 Removing Whitespace-only Text Nodes from an XML DOM Node's Subtree	474
12.7 Parsing Microsoft Excel's XML	475
12.8 Validating XML Documents	477
12.9 Filtering Elements and Attributes Belonging to a Given Namespace	478
12.10 Merging Continuous Text Events with a SAX Filter	480
12.11 Using MSHTML to Parse XML or HTML	483
13. Network Programming	485
13.1 Passing Messages with Socket Datagrams	487
13.2 Grabbing a Document from the Web	489
13.3 Filtering a List of FTP Sites	490

13.4	Getting Time from a Server via the SNTP Protocol	491
13.5	Sending HTML Mail	492
13.6	Bundling Files in a MIME Message	495
13.7	Unpacking a Multipart MIME Message	497
13.8	Removing Attachments from an Email Message	499
13.9	Fixing Messages Parsed by Python 2.4 email.FeedParser	501
13.10	Inspecting a POP3 Mailbox Interactively	503
13.11	Detecting Inactive Computers	506
13.12	Monitoring a Network with HTTP	511
13.13	Forwarding and Redirecting Network Ports	513
13.14	Tunneling SSL Through a Proxy	516
13.15	Implementing the Dynamic IP Protocol	519
13.16	Connecting to IRC and Logging Messages to Disk	522
13.17	Accessing LDAP Servers	524
14.	Web Programming	526
14.1	Testing Whether CGI Is Working	527
14.2	Handling URLs Within a CGI Script	530
14.3	Uploading Files with CGI	532
14.4	Checking for a Web Page's Existence	533
14.5	Checking Content Type via HTTP	535
14.6	Resuming the HTTP Download of a File	536
14.7	Handling Cookies While Fetching Web Pages	538
14.8	Authenticating with a Proxy for HTTPS Navigation	541
14.9	Running a Servlet with Jython	542
14.10	Finding an Internet Explorer Cookie	543
14.11	Generating OPML Files	545
14.12	Aggregating RSS Feeds	548
14.13	Turning Data into Web Pages Through Templates	552
14.14	Rendering Arbitrary Objects with Nevow	554
15.	Distributed Programming	558
15.1	Making an XML-RPC Method Call	561
15.2	Serving XML-RPC Requests	562
15.3	Using XML-RPC with Medusa	564
15.4	Enabling an XML-RPC Server to Be Terminated Remotely	566
15.5	Implementing SimpleXMLRPCServer Niceties	567
15.6	Giving an XML-RPC Server a wxPython GUI	569
15.7	Using Twisted Perspective Broker	571

15.8	Implementing a CORBA Server and Client	574
15.9	Performing Remote Logins Using telnetlib	576
15.10	Performing Remote Logins with SSH	579
15.11	Authenticating an SSL Client over HTTPS	582
16.	Programs About Programs	584
16.1	Verifying Whether a String Represents a Valid Number	590
16.2	Importing a Dynamically Generated Module	591
16.3	Importing from a Module Whose Name Is Determined at Runtime	592
16.4	Associating Parameters with a Function (Currying)	594
16.5	Composing Functions	597
16.6	Colorizing Python Source Using the Built-in Tokenizer	598
16.7	Merging and Splitting Tokens	602
16.8	Checking Whether a String Has Balanced Parentheses	604
16.9	Simulating Enumerations in Python	606
16.10	Referring to a List Comprehension While Building It	609
16.11	Automating the py2exe Compilation of Scripts into Windows Executables	611
16.12	Binding Main Script and Modules into One Executable on Unix	613
17.	Extending and Embedding	616
17.1	Implementing a Simple Extension Type	619
17.2	Implementing a Simple Extension Type with Pyrex	623
17.3	Exposing a C++ Library to Python	625
17.4	Calling Functions from a Windows DLL	627
17.5	Using SWIG-Generated Modules in a Multithreaded Environment	630
17.6	Translating a Python Sequence into a C Array with the PySequence_Fast Protocol	631
17.7	Accessing a Python Sequence Item-by-Item with the Iterator Protocol	635
17.8	Returning None from a Python-Callable C Function	638
17.9	Debugging Dynamically Loaded C Extensions with gdb	639
17.10	Debugging Memory Problems	641
18.	Algorithms	643
18.1	Removing Duplicates from a Sequence	647
18.2	Removing Duplicates from a Sequence While Maintaining Sequence Order	649
18.3	Generating Random Samples with Replacement	653
18.4	Generating Random Samples Without Replacement	654

18.5 Memoizing (Caching) the Return Values of Functions	656
18.6 Implementing a FIFO Container	658
18.7 Caching Objects with a FIFO Pruning Strategy	660
18.8 Implementing a Bag (Multiset) Collection Type	662
18.9 Simulating the Ternary Operator in Python	666
18.10 Computing Prime Numbers	669
18.11 Formatting Integers as Binary Strings	671
18.12 Formatting Integers as Strings in Arbitrary Bases	673
18.13 Converting Numbers to Rationals via Farey Fractions	675
18.14 Doing Arithmetic with Error Propagation	677
18.15 Summing Numbers with Maximal Accuracy	680
18.16 Simulating Floating Point	682
18.17 Computing the Convex Hulls and Diameters of 2D Point Sets	685
19. Iterators and Generators	689
19.1 Writing a range-like Function with Float Increments	693
19.2 Building a List from Any Iterable	695
19.3 Generating the Fibonacci Sequence	697
19.4 Unpacking a Few Items in a Multiple Assignment	698
19.5 Automatically Unpacking the Needed Number of Items	700
19.6 Dividing an Iterable into n Slices of Stride n	702
19.7 Looping on a Sequence by Overlapping Windows	704
19.8 Looping Through Multiple Iterables in Parallel	708
19.9 Looping Through the Cross-Product of Multiple Iterables	710
19.10 Reading a Text File by Paragraphs	713
19.11 Reading Lines with Continuation Characters	715
19.12 Iterating on a Stream of Data Blocks as a Stream of Lines	717
19.13 Fetching Large Record Sets from a Database with a Generator	719
19.14 Merging Sorted Sequences	721
19.15 Generating Permutations, Combinations, and Selections	724
19.16 Generating the Partitions of an Integer	726
19.17 Duplicating an Iterator	728
19.18 Looking Ahead into an Iterator	731
19.19 Simplifying Queue-Consumer Threads	734
19.20 Running an Iterator in Another Thread	735
19.21 Computing a Summary Report with <code>itertools.groupby</code>	737

20. Descriptors, Decorators, and Metaclasses	740
20.1 Getting Fresh Default Values at Each Function Call	742
20.2 Coding Properties as Nested Functions	744
20.3 Aliasing Attribute Values	747
20.4 Caching Attribute Values	750
20.5 Using One Method as Accessor for Multiple Attributes	752
20.6 Adding Functionality to a Class by Wrapping a Method	754
20.7 Adding Functionality to a Class by Enriching All Methods	757
20.8 Adding a Method to a Class Instance at Runtime	759
20.9 Checking Whether Interfaces Are Implemented	761
20.10 Using <code>__new__</code> and <code>__init__</code> Appropriately in Custom Metaclasses	763
20.11 Allowing Chaining of Mutating List Methods	765
20.12 Using Cooperative Supercalls with Terser Syntax	767
20.13 Initializing Instance Attributes Without Using <code>__init__</code>	769
20.14 Automatic Initialization of Instance Attributes	771
20.15 Upgrading Class Instances Automatically on reload	774
20.16 Binding Constants at Compile Time	778
20.17 Solving Metaclass Conflicts	783
Index	789