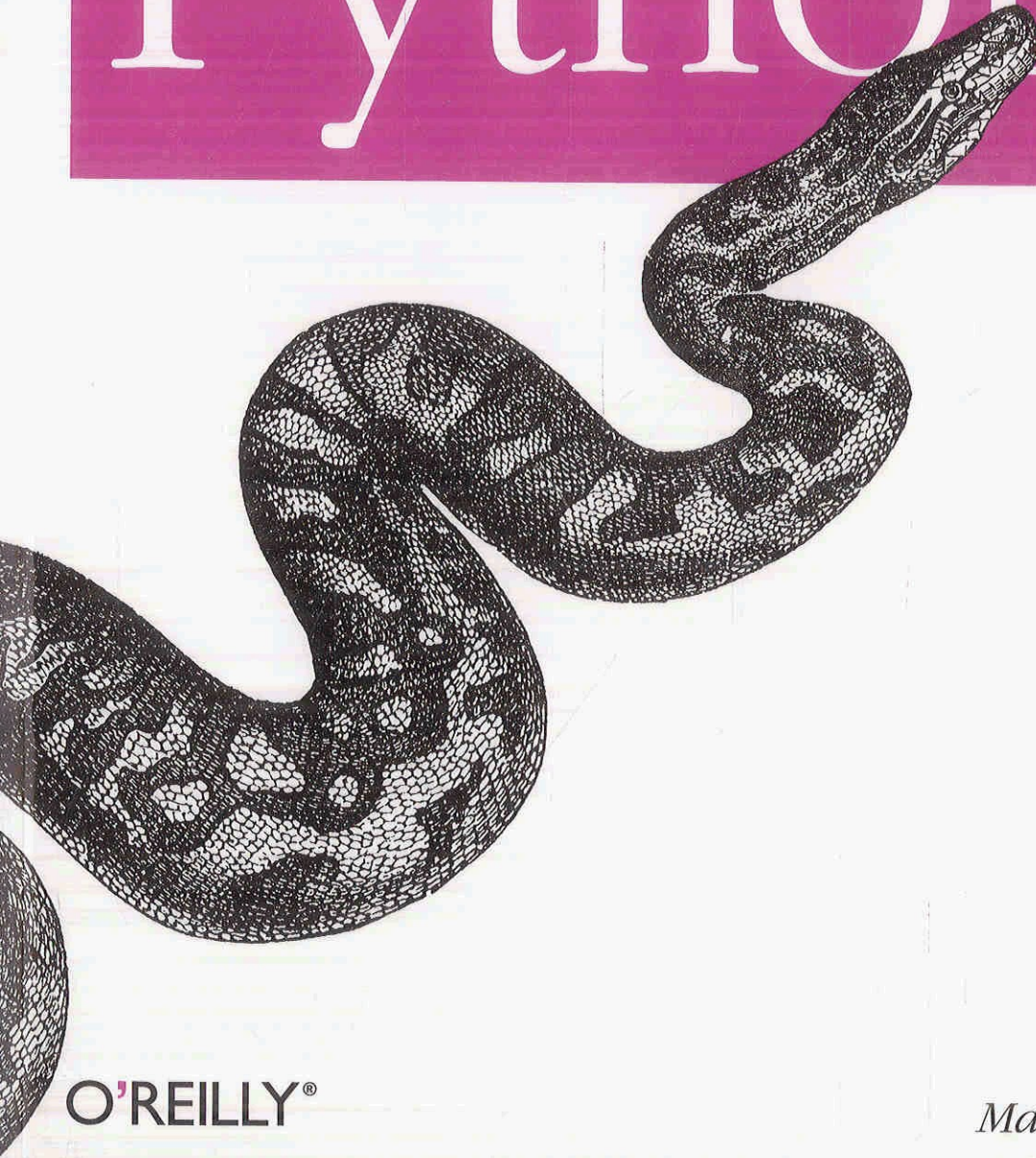


Powerful Object-Oriented Programming

4th Edition
Covers Python 3.x

Programming

Python



O'REILLY®

Mark Lutz

Table of Contents

Preface	xxiii
---------------	-------

Part I. The Beginning

1. A Sneak Preview	3
“Programming Python: The Short Story”	3
The Task	4
Step 1: Representing Records	4
Using Lists	4
Using Dictionaries	9
Step 2: Storing Records Persistently	14
Using Formatted Files	14
Using Pickle Files	19
Using Per-Record Pickle Files	22
Using Shelves	23
Step 3: Stepping Up to OOP	26
Using Classes	27
Adding Behavior	29
Adding Inheritance	29
Refactoring Code	31
Adding Persistence	34
Other Database Options	36
Step 4: Adding Console Interaction	37
A Console Shelf Interface	37
Step 5: Adding a GUI	40
GUI Basics	40
Using OOP for GUIs	42
Getting Input from a User	44
A GUI Shelf Interface	46
Step 6: Adding a Web Interface	52
CGI Basics	52

Running a Web Server	55
Using Query Strings and urllib	57
Formatting Reply Text	59
A Web-Based Shelf Interface	60
The End of the Demo	69

Part II. System Programming

2. System Tools	73
“The os.path to Knowledge”	73
Why Python Here?	73
The Next Five Chapters	74
System Scripting Overview	75
Python System Modules	76
Module Documentation Sources	77
Paging Documentation Strings	78
A Custom Paging Script	79
String Method Basics	80
Other String Concepts in Python 3.X: Unicode and bytes	82
File Operation Basics	83
Using Programs in Two Ways	84
Python Library Manuals	85
Commercially Published References	86
Introducing the sys Module	86
Platforms and Versions	86
The Module Search Path	87
The Loaded Modules Table	88
Exception Details	89
Other sys Module Exports	90
Introducing the os Module	90
Tools in the os Module	90
Administrative Tools	91
Portability Constants	92
Common os.path Tools	92
Running Shell Commands from Scripts	94
Other os Module Exports	100
3. Script Execution Context	103
“I’d Like to Have an Argument, Please”	103
Current Working Directory	104
CWD, Files, and Import Paths	104
CWD and Command Lines	106

Command-Line Arguments	106
Parsing Command-Line Arguments	107
Shell Environment Variables	109
Fetching Shell Variables	110
Changing Shell Variables	111
Shell Variable Fine Points: Parents, putenv, and getenv	112
Standard Streams	113
Redirecting Streams to Files and Programs	114
Redirected Streams and User Interaction	119
Redirecting Streams to Python Objects	123
The io.StringIO and io.BytesIO Utility Classes	126
Capturing the stderr Stream	127
Redirection Syntax in Print Calls	127
Other Redirection Options: os.popen and subprocess Revisited	128
4. File and Directory Tools	135
“Erase Your Hard Drive in Five Easy Steps!”	135
File Tools	135
The File Object Model in Python 3.X	136
Using Built-in File Objects	137
Binary and Text Files	146
Lower-Level File Tools in the os Module	155
File Scanners	160
Directory Tools	163
Walking One Directory	164
Walking Directory Trees	168
Handling Unicode Filenames in 3.X: listdir, walk, glob	172
5. Parallel System Tools	177
“Telling the Monkeys What to Do”	177
Forking Processes	179
The fork/exec Combination	182
Threads	186
The _thread Module	189
The threading Module	199
The queue Module	204
Preview: GUIs and Threads	208
More on the Global Interpreter Lock	211
Program Exits	213
sys Module Exits	214
os Module Exits	215
Shell Command Exit Status Codes	216
Process Exit Status and Shared State	219

Thread Exits and Shared State	220
Interprocess Communication	222
Anonymous Pipes	224
Named Pipes (Fifos)	234
Sockets: A First Look	236
Signals	240
The multiprocessing Module	243
Why multiprocessing?	243
The Basics: Processes and Locks	245
IPC Tools: Pipes, Shared Memory, and Queues	248
<i>Starting Independent Programs</i>	254
And Much More	256
Why multiprocessing? The Conclusion	257
Other Ways to Start Programs	258
The <code>os.spawn</code> Calls	258
The <code>os.startfile</code> call on Windows	261
A Portable Program-Launch Framework	263
Other System Tools Coverage	268

6. Complete System Programs	271
“The Greps of Wrath”	271
A Quick Game of “Find the Biggest Python File”	272
Scanning the Standard Library Directory	272
Scanning the Standard Library Tree	273
Scanning the Module Search Path	274
Scanning the Entire Machine	276
Printing Unicode Filenames	279
Splitting and Joining Files	282
Splitting Files Portably	283
Joining Files Portably	286
Usage Variations	289
Generating Redirection Web Pages	292
Page Template File	293
Page Generator Script	294
A Regression Test Script	297
<i>Running the Test Driver</i>	299
Copying Directory Trees	304
Comparing Directory Trees	308
Finding Directory Differences	309
Finding Tree Differences	311
Running the Script	314
Verifying Backups	316
Reporting Differences and Other Ideas	317

Searching Directory Trees	319
Greps and Globs and Finds	320
Rolling Your Own find Module	321
Cleaning Up Bytecode Files	324
A Python Tree Searcher	327
Visitor: Walking Directories “++”	330
Editing Files in Directory Trees (Visitor)	334
Global Replacements in Directory Trees (Visitor)	336
Counting Source Code Lines (Visitor)	338
Recoding Copies with Classes (Visitor)	339
Other Visitor Examples (External)	341
Playing Media Files	343
The Python webbrowser Module	347
The Python mimetypes Module	348
Running the Script	350
Automated Program Launchers (External)	351

Part III. GUI Programming

7. Graphical User Interfaces	355
“Here’s Looking at You, Kid”	355
GUI Programming Topics	355
Running the Examples	357
Python GUI Development Options	358
tkinter Overview	363
tkinter Pragmatics	363
tkinter Documentation	364
tkinter Extensions	364
tkinter Structure	366
Climbing the GUI Learning Curve	368
“Hello World” in Four Lines (or Less)	368
tkinter Coding Basics	369
Making Widgets	370
Geometry Managers	370
Running GUI Programs	371
tkinter Coding Alternatives	372
Widget Resizing Basics	373
Configuring Widget Options and Window Titles	375
One More for Old Times’ Sake	376
Packing Widgets Without Saving Them	377
Adding Buttons and Callbacks	379
Widget Resizing Revisited: Expansion	380

Adding User-Defined Callback Handlers	382
Lambda Callback Handlers	383
Deferring Calls with Lambdas and Object References	384
Callback Scope Issues	385
Bound Method Callback Handlers	391
Callable Class Object Callback Handlers	392
Other tkinter Callback Protocols	393
Binding Events	394
Adding Multiple Widgets	395
Widget Resizing Revisited: Clipping	396
Attaching Widgets to Frames	397
Layout: Packing Order and Side Attachments	397
The Packer's Expand and Fill Revisited	398
Using Anchor to Position Instead of Stretch	399
Customizing Widgets with Classes	400
Standardizing Behavior and Appearance	401
Reusable GUI Components with Classes	403
Attaching Class Components	405
Extending Class Components	407
Standalone Container Classes	408
The End of the Tutorial	410
Python/tkinter for Tcl/Tk Converts	412
8. A tkinter Tour, Part 1	415
"Widgets and Gadgets and GUIs, Oh My!"	415
This Chapter's Topics	415
Configuring Widget Appearance	416
Top-Level Windows	419
Toplevel and Tk Widgets	421
Top-Level Window Protocols	422
Dialogs	426
Standard (Common) Dialogs	426
The Old-Style Dialog Module	438
Custom Dialogs	439
Binding Events	443
Other bind Events	447
Message and Entry	448
Message	448
Entry	449
Laying Out Input Forms	451
tkinter "Variables" and Form Layout Alternatives	454
Checkbutton, Radiobutton, and Scale	457
Checkbuttons	457

Radio Buttons	462
Scales (Sliders)	467
Running GUI Code Three Ways	471
Attaching Frames	471
Independent Windows	476
Running Programs	478
Images	484
Fun with Buttons and Pictures	487
Viewing and Processing Images with PIL	491
PIL Basics	491
Displaying Other Image Types with PIL	493
Creating Image Thumbnails with PIL	496
9. A tkinter Tour, Part 2	507
“On Today’s Menu: Spam, Spam, and Spam”	507
Menus	507
Top-Level Window Menus	508
Frame- and Menubutton-Based Menus	512
Windows with Both Menus and Toolbars	517
Listboxes and Scrollbars	522
Programming Listboxes	524
Programming Scroll Bars	525
Packing Scroll Bars	526
Text	528
Programming the Text Widget	530
Adding Text-Editing Operations	533
Unicode and the Text Widget	538
Advanced Text and Tag Operations	548
Canvas	550
Basic Canvas Operations	550
Programming the Canvas Widget	551
Scrolling Canvases	554
Scrollable Canvases and Image Thumbnails	557
Using Canvas Events	560
Grids	564
Why Grids?	564
Grid Basics: Input Forms Revisited	565
Comparing grid and pack	566
Combining grid and pack	568
Making Gridded Widgets Expandable	570
Laying Out Larger Tables with grid	574
Time Tools, Threads, and Animation	582
Using Threads with tkinter GUIs	584

Using the after Method	585
Simple Animation Techniques	588
Other Animation Topics	593
The End of the Tour	595
Other Widgets and Options	595
10. GUI Coding Techniques	597
“Building a Better Mousetrap”	597
GuiMixin: Common Tool Mixin Classes	598
Widget Builder Functions	598
Mixin Utility Classes	599
GuiMaker: Automating Menus and Toolbars	603
Subclass Protocols	607
GuiMaker Classes	608
GuiMaker Self-Test	608
BigGui: A Client Demo Program	609
ShellGui: GUIs for Command-Line Tools	613
A Generic Shell-Tools Display	613
Application-Specific Tool Set Classes	615
Adding GUI Frontends to Command Lines	617
GuiStreams: Redirecting Streams to Widgets	623
Using Redirection for the Packing Scripts	627
Reloading Callback Handlers Dynamically	628
Wrapping Up Top-Level Window Interfaces	630
GUIs, Threads, and Queues	635
Placing Data on Queues	636
Placing Callbacks on Queues	640
More Ways to Add GUIs to Non-GUI Code	646
Popping Up GUI Windows on Demand	647
Adding a GUI As a Separate Program: Sockets (A Second Look)	649
Adding a GUI As a Separate Program: Command Pipes	654
The PyDemos and PyGadgets Launchers	662
PyDemos Launcher Bar (Mostly External)	662
PyGadgets Launcher Bar	667
11. Complete GUI Programs	671
“Python, Open Source, and Camaros”	671
Examples in Other Chapters	672
This Chapter’s Strategy	673
PyEdit: A Text Editor Program/Object	674
Running PyEdit	675
PyEdit Changes in Version 2.0 (Third Edition)	682
PyEdit Changes in Version 2.1 (Fourth Edition)	684

PyEdit Source Code	693
PyPhoto: An Image Viewer and Resizer	716
Running PyPhoto	717
PyPhoto Source Code	719
PyView: An Image and Notes Slideshow	727
Running PyView	727
PyView Source Code	732
PyDraw: Painting and Moving Graphics	738
Running PyDraw	738
PyDraw Source Code	738
PyClock: An Analog/Digital Clock Widget	747
A Quick Geometry Lesson	747
Running PyClock	751
PyClock Source Code	754
PyToe: A Tic-Tac-Toe Game Widget	762
Running PyToe	762
PyToe Source Code (External)	763
Where to Go from Here	766

Part IV. Internet Programming

12. Network Scripting	771
“Tune In, Log On, and Drop Out”	771
Internet Scripting Topics	772
Running Examples in This Part of the Book	775
Python Internet Development Options	777
Plumbing the Internet	780
The Socket Layer	781
The Protocol Layer	782
Python’s Internet Library Modules	785
Socket Programming	787
Socket Basics	788
Running Socket Programs Locally	794
Running Socket Programs Remotely	795
Spawning Clients in Parallel	798
Talking to Reserved Ports	801
Handling Multiple Clients	802
Forking Servers	803
Threading Servers	815
Standard Library Server Classes	818
Multiplexing Servers with select	820
Summary: Choosing a Server Scheme	826

Making Sockets Look Like Files and Streams	827
A Stream Redirection Utility	828
A Simple Python File Server	840
Running the File Server and Clients	842
Adding a User-Interface Frontend	843
13. Client-Side Scripting	853
“Socket to Me!”	853
FTP: Transferring Files over the Net	854
Transferring Files with ftplib	854
Using urllib to Download Files	857
FTP get and put Utilities	860
Adding a User Interface	867
Transferring Directories with ftplib	874
Downloading Site Directories	874
Uploading Site Directories	880
Refactoring Uploads and Downloads for Reuse	884
Transferring Directory Trees with ftplib	892
Uploading Local Trees	893
Deleting Remote Trees	895
Downloading Remote Trees	899
Processing Internet Email	899
Unicode in Python 3.X and Email Tools	900
POP: Fetching Email	901
Mail Configuration Module	902
POP Mail Reader Script	905
Fetching Messages	906
Fetching Email at the Interactive Prompt	909
SMTP: Sending Email	910
SMTP Mail Sender Script	911
Sending Messages	913
Sending Email at the Interactive Prompt	919
email: Parsing and Composing Mail Content	921
Message Objects	922
Basic email Package Interfaces in Action	924
Unicode, Internationalization, and the Python 3.1 email Package	926
A Console-Based Email Client	947
Running the pypmail Console Client	952
The mailtools Utility Package	956
Initialization File	957
MailTool Class	958
MailSender Class	959
MailFetcher Class	967

MailParser Class	976
Self-Test Script	983
Updating the pymail Console Client	986
NNTP: Accessing Newsgroups	991
HTTP: Accessing Websites	994
The urllib Package Revisited	997
Other urllib Interfaces	999
Other Client-Side Scripting Options	1002

14. The PyMailGUI Client	1005
“Use the Source, Luke”	1005
Source Code Modules and Size	1006
Why PyMailGUI?	1008
Running PyMailGUI	1010
Presentation Strategy	1010
Major PyMailGUI Changes	1011
New in Version 2.1 and 2.0 (Third Edition)	1011
New in Version 3.0 (Fourth Edition)	1012
A PyMailGUI Demo	1019
Getting Started	1020
Loading Mail	1025
Threading Model	1027
Load Server Interface	1030
Offline Processing with Save and Open	1031
Sending Email and Attachments	1033
Viewing Email and Attachments	1037
Email Replies and Forwards and Recipient Options	1043
Deleting Email	1049
POP Message Numbers and Synchronization	1051
Handling HTML Content in Email	1053
Mail Content Internationalization Support	1055
Alternative Configurations and Accounts	1059
Multiple Windows and Status Messages	1060
PyMailGUI Implementation	1062
PyMailGUI: The Main Module	1063
SharedNames: Program-Wide Globals	1066
ListWindows: Message List Windows	1067
ViewWindows: Message View Windows	1085
messagecache: Message Cache Manager	1095
popuputil: General-Purpose GUI Pop Ups	1098
wraplines: Line Split Tools	1100
html2text: Extracting Text from HTML (Prototype, Preview)	1102
mailconfig: User Configurations	1105

textConfig: Customizing Pop-Up PyEdit Windows	1110
PyMailGUIHelp: User Help Text and Display	1111
altconfigs: Configuring for Multiple Accounts	1114
Ideas for Improvement	1116
15. Server-Side Scripting	1125
“Oh, What a Tangled Web We Weave”	1125
What’s a Server-Side CGI Script?	1126
The Script Behind the Curtain	1126
Writing CGI Scripts in Python	1128
Running Server-Side Examples	1130
Web Server Options	1130
Running a Local Web Server	1131
The Server-Side Examples Root Page	1133
Viewing Server-Side Examples and Output	1134
Climbing the CGI Learning Curve	1135
A First Web Page	1135
A First CGI Script	1141
Adding Pictures and Generating Tables	1146
Adding User Interaction	1149
Using Tables to Lay Out Forms	1157
Adding Common Input Devices	1163
Changing Input Layouts	1166
Passing Parameters in Hardcoded URLs	1170
Passing Parameters in Hidden Form Fields	1172
Saving State Information in CGI Scripts	1174
URL Query Parameters	1176
Hidden Form Input Fields	1176
HTTP “Cookies”	1177
Server-Side Databases	1181
Extensions to the CGI Model	1182
Combining Techniques	1183
The Hello World Selector	1183
Checking for Missing and Invalid Inputs	1190
Refactoring Code for Maintainability	1192
Step 1: Sharing Objects Between Pages—A New Input Form	1193
Step 2: A Reusable Form Mock-Up Utility	1196
Step 3: Putting It All Together—A New Reply Script	1199
More on HTML and URL Escapes	1201
URL Escape Code Conventions	1202
Python HTML and URL Escape Tools	1203
Escaping HTML Code	1203
Escaping URLs	1204

Escaping URLs Embedded in HTML Code	1205
Transferring Files to Clients and Servers	1209
Displaying Arbitrary Server Files on the Client	1211
Uploading Client Files to the Server	1218
More Than One Way to Push Bits over the Net	1227
16. The PyMailCGI Server	1229
“Things to Do When Visiting Chicago”	1229
The PyMailCGI Website	1230
Implementation Overview	1230
New in This Fourth Edition (Version 3.0)	1233
New in the Prior Edition (Version 2.0)	1235
Presentation Overview	1236
Running This Chapter’s Examples	1237
The Root Page	1239
Configuring PyMailCGI	1240
Sending Mail by SMTP	1241
The Message Composition Page	1242
The Send Mail Script	1242
Error Pages	1246
Common Look-and-Feel	1246
Using the Send Mail Script Outside a Browser	1247
Reading POP Email	1249
The POP Password Page	1250
The Mail Selection List Page	1251
Passing State Information in URL Link Parameters	1254
Security Protocols	1257
The Message View Page	1259
Passing State Information in HTML Hidden Input Fields	1262
Escaping Mail Text and Passwords in HTML	1264
Processing Fetched Mail	1266
Reply and Forward	1267
Delete	1268
Deletions and POP Message Numbers	1272
Utility Modules	1276
External Components and Configuration	1276
POP Mail Interface	1277
POP Password Encryption	1278
Common Utilities Module	1286
Web Scripting Trade-Offs	1291
PyMailCGI Versus PyMailGUI	1292
The Web Versus the Desktop	1293
Other Approaches	1296

Part V. Tools and Techniques

17. Databases and Persistence	1303
“Give Me an Order of Persistence, but Hold the Pickles”	1303
Persistence Options in Python	1303
DBM Files	1305
Using DBM Files	1305
DBM Details: Files, Portability, and Close	1308
Pickled Objects	1309
Using Object Pickling	1310
Pickling in Action	1311
Pickle Details: Protocols, Binary Modes, and <code>_pickle</code>	1314
Shelve Files	1315
Using Shelves	1316
Storing Built-in Object Types in Shelves	1317
Storing Class Instances in Shelves	1318
Changing Classes of Objects Stored in Shelves	1320
Shelve Constraints	1321
Pickled Class Constraints	1323
Other Shelve Limitations	1324
The ZODB Object-Oriented Database	1325
The Mostly Missing ZODB Tutorial	1326
SQL Database Interfaces	1329
SQL Interface Overview	1330
An SQL Database API Tutorial with SQLite	1332
Building Record Dictionaries	1339
Tying the Pieces Together	1342
Loading Database Tables from Files	1344
SQL Utility Scripts	1347
SQL Resources	1354
ORMs: Object Relational Mappers	1354
PyForm: A Persistent Object Viewer (External)	1356
18. Data Structures	1359
“Roses Are Red, Violets Are Blue; Lists Are Mutable, and So Is Set Foo”	1359
Implementing Stacks	1360
Built-in Options	1360
A Stack Module	1362
A Stack Class	1364
Customization: Performance Monitors	1366
Optimization: Tuple Tree Stacks	1367

Optimization: In-Place List Modifications	1369
Timing the Improvements	1371
Implementing Sets	1373
Built-in Options	1374
Set Functions	1375
Set Classes	1377
Optimization: Moving Sets to Dictionaries	1378
Adding Relational Algebra to Sets (External)	1382
Subclassing Built-in Types	1383
Binary Search Trees	1385
Built-in Options	1385
Implementing Binary Trees	1386
Trees with Both Keys and Values	1388
Graph Searching	1390
Implementing Graph Search	1390
Moving Graphs to Classes	1393
Permuting Sequences	1395
Reversing and Sorting Sequences	1397
Implementing Reversals	1398
Implementing Sorts	1399
Data Structures Versus Built-ins: The Conclusion	1400
PyTree: A Generic Tree Object Viewer	1402

19. Text and Language	1405
“See Jack Hack. Hack, Jack, Hack”	1405
Strategies for Processing Text in Python	1405
String Method Utilities	1406
Templating with Replacements and Formats	1408
Parsing with Splits and Joins	1409
Summing Columns in a File	1410
Parsing and Unparsing Rule Strings	1412
Regular Expression Pattern Matching	1415
The re Module	1416
First Examples	1416
String Operations Versus Patterns	1418
Using the re Module	1421
More Pattern Examples	1425
Scanning C Header Files for Patterns	1427
XML and HTML Parsing	1429
XML Parsing in Action	1430
HTML Parsing in Action	1435
Advanced Language Tools	1438
Custom Language Parsers	1440

The Expression Grammar	1440
The Parser's Code	1441
Adding a Parse Tree Interpreter	1449
Parse Tree Structure	1454
Exploring Parse Trees with the PyTree GUI	1456
Parsers Versus Python	1457
PyCalc: A Calculator Program/Object	1457
A Simple Calculator GUI	1458
PyCalc—A “Real” Calculator GUI	1463
20. Python/C Integration	1483
“I Am Lost at C”	1483
Extending and Embedding	1484
Extending Python in C: Overview	1486
A Simple C Extension Module	1487
The SWIG Integration Code Generator	1491
A Simple SWIG Example	1491
Wrapping C Environment Calls	1495
Adding Wrapper Classes to Flat Libraries	1499
Wrapping C Environment Calls with SWIG	1500
Wrapping C++ Classes with SWIG	1502
A Simple C++ Extension Class	1503
Wrapping the C++ Class with SWIG	1505
Using the C++ Class in Python	1507
Other Extending Tools	1511
Embedding Python in C: Overview	1514
The C Embedding API	1515
What Is Embedded Code?	1516
Basic Embedding Techniques	1518
Running Simple Code Strings	1519
Running Code Strings with Results and Namespaces	1522
Calling Python Objects	1524
Running Strings in Dictionaries	1526
Precompiling Strings to Bytecode	1528
Registering Callback Handler Objects	1530
Registration Implementation	1531
Using Python Classes in C	1535
Other Integration Topics	1538

Part VI. The End

21. Conclusion: Python and the Development Cycle	1543
“That’s the End of the Book, Now Here’s the Meaning of Life”	1544
“Something’s Wrong with the Way We Program Computers”	1544
The “Gilligan Factor”	1544
Doing the Right Thing	1545
The Static Language Build Cycle	1546
Artificial Complexities	1546
One Language Does Not Fit All	1546
Enter Python	1547
But What About That Bottleneck?	1548
Python Provides Immediate Turnaround	1549
Python Is “Executable Pseudocode”	1550
Python Is OOP Done Right	1550
Python Fosters Hybrid Applications	1551
On Sinking the Titanic	1552
So What’s “Python: The Sequel”?	1555
In the Final Analysis...	1555
Index	1557