Table of Contents

***A_Learners_Guide_to_Programming_Using_the_Python_Language_2009_442p.......................7
***Apress_Pro_Django_2nd_edition_2013_290p..7
***A_Primer_on_Scientific_Programming_with_Python_2009_726p...7
***Beginning_Game_Development_with_Python_and_Pygame_From_Novice_to_Professional_2007_342p..21
***Beginning_Python_2005_679p..22
***Beginning_Python_From_Novice_to_Professional_Second_Edition_2008_666p.....................34
***Beginning_Python_Using_Python_2.6_and_Python_3.1_2010_628p.......................................35
***Beginning_Python_Visualization_Crafting_Visual_Transformation_Scripts_2009_378p..........36
***Bioinformatics_Programming_Using_Python_First_Edition_2009_524p.................................36
***BuildingSkillsinPython_2010_574p..36
***byte_of_python_2013_129p...43
***CherryPy_Essentials_Rapid_Python_Web_Application_Development_2007_270p..................47
***Computer_Games_with_PYTHON_2010_436p..51
***Dive_Into_Python_5_4_2004_327p...56
***Expert_Python_Programming_2008_372p...60
***Foundations_of_Agile_Python_Development_2008_417p...66
***Game_Programming_with_Python_Lua_and_Ruby_2003_457p..71
***Getting_Started_with_Pyparsing_2007_65p...71
***Gray_Hat_Python_Python_Programming_for_Hackers_and_Reverse_Engineers_2009_220p. 72
***Guide_to_NumPy_for_Python_2006_378p...74
***Hello_World_Computer_Programming_for_Kids_and_Other_Beginners_Fourth_printing_2009_432p...80
***IronPython_in_Action_2009_494p..81
***Learning_Python_2.5_3rd.Edition_2007_748p...81
***learning_python_fourth_edition_2009_1213p...95
***learning_the_vi_and_vim_editors_seventh_edition_2008_494p..114
***Learning_to_Program_Python_2006_299p...117
***Learn_Python_The_Hard_Way_2010_165p...118
***making_games_PYTHON_2012_365p...121
***Making_Use_of_Python_2002_416p...127
***Matplotlib_for_Python_Developers_2009_307p..131
***Mobile_Python_Rapid_Prototyping_of_Applications_on_the_Mobile_Platform_2007_329p.134
***Natural_Language_Processing_with_Python_2009_491p..136
***Numerical_Methods_in_Engineering_with_Python_2005_433p...138
***Oreilly_Python_Cookbook_2nd_2005_974p...139
***Practical_Programming_An_Introduction_to_Computer_Science_Using_Python_2009_369p
....146
***Prentice_Hall_Core_Python_Programming_2nd_Edition_2001_703p......................................149
***Professional_Python_Frameworks_Web_2.0_Programming_with_Django_and_TurboGears_2007_459p...150
***Programming_in_Python_3_A_Complete_Introduction_to_the_Python_Language_SecondEdition_2010_644p...150
***Programming_Python_3rdEd_2006_1740p..151
***Pro_IronPython_2009_298p...155
***Pro_Python_System_Administration_Apress_2010_417p..156
***Python_2_1_Bible_2001_769p..156
***Python_2_6_Quick_Reference_Letter_2009_50p..157
***Python_3_for_Absolute_Beginners_2009_314p..158
***Python_Create_Modify_Reuse_2008_291p..158

***PythonDevelopersHandbook_2000_1481p....158
***Python_Essential_Reference_2nd_Edition_2001_538p....159
***Python_Essential_Reference_Fourth_Edition_2009_742p....159
***Python_for_Dummies_2006_351p....160
***Python_for_Software_Design_How_to_Think_Like_a_Computer_Scientist_2009_264p....161
***Python_for_Unix_and_Linux_System_Administration_2008_458p....161
***Python_How_to_Program_1e_2002_1103p....164
***Python_Pocket_Reference_4th_Edition_2009_210p....165
***Python_Power_The_Comprehensive_Guide_2008_529p....167
***Python_Programming_An_Introduction_to_Computer_Science_v1_0rc2_2002_261p....173
***Python_Programming_Tutorial_2000_195p....177
***Python_Scripting_for_Computational_Science_ThirdEdition_2008_767p....178
***Python_Testing_Beginners_Guide_2010_255p....189
***Python_Tutorial_Learning_To_Program_2006_342p....194
***python_tutorial_tutorialspoint_2014_431p....194
***Python_v1_3_13_S60_3rd_Edition_Docs_2006_86p....194
***Rapid_GUI_Programming_with_Python_and_Qt_The_Definitive_Guide_to_PyQt_Programming_2008_643p....196
***Snake_Wrangling_for_Kids_Learning_to_Program_with_Python_3_linux_v0_7_7_2007_160p....200
***Starting_Out_with_Python_2009_502p....201
***The_Definitive_Guide_to_Jython_Python_for_the_Java_Platform_2010_545p....203
***The_Quick_Python_Book_Second_Edition_2010_362p....204
***Thinking_in_Python_Bruce_Eckel_Rev_0_1_2_2001_177p....204
***Thinking_in_Tkinter_2005_53p....206
***think_python_2012_240p....206
***Tkinter_8_4_reference_a_GUI_for_Python_2010_118p....207
***wxPython_in_action_2006_583p....209
***XML_Processing_with_Perl_Python_and_PHP_2002_447p....209
***XML_Processing_with_Python_2002_447p....210
***Beginning_Django_E-Commerce_2009_398p....213
***Django_1.0_Template_Development_2008_272p....218
***Python_Web_Development_with_Django_2009_405....225
***Test-Driven_Development_with_Python_2014_478....231
***beautiful-soup-4_2015_66p....237
***Getting_Started_with_Beautiful_Soup_2014_130p....238
***HeadFirstPython_2011_494p....240
***IntroductiontoTornado_2012_136p....240
***Kivy__InteractiveApplicationsinPython_2013_138p....242
***Learning_Python_5thEdition_2013_1594p....243
***LearnPythontheHardWay3rdEdition_2014_306p....264
***MasteringPythonRegularExpressions-LopezFelixRomeroVictor_2014_110p....271
***MongoDB_and_Python_2011_66p....273
***MySQL_for_Python_2010_440p....273
***openpyxl_2015_133p....283
***Parallel_Programming_with_Python_2014_122p....284
***Pro_Android_Python_with_SL4A_2011_296p....286
***Programming_Python_MLutz_4th_ed_2011_1628p....286
***Python_3_Web_Development_2011_336....299
***Python_and_AWS_Cookbook_2012_74p....303
***Python_Cookbook_3rd_Edition_2013_706p....304
***python-excel_2009_56p....309

***python-pocket-reference-5th-edition-MarkLutz-2014-264p......309
***SciPy_and_NumPy_2013_66p......312
***Testing_Python_2014_243p......313
***The_Python_Standard_Library_by_Example_2011_1343p......317
***Think_Complexity_2012_146p......332
***Twisted_Network_Programming_Essentials_2nd_ed_2013_193p......335
***Using_Google_App_Engine_2009_264p......337
***Building_Web_Applications_with_Flask_2015_160p......341
***Flask_Framework_Cookbook_2014_258p......342
***Flask_Web_Development_2014_258p......344

***A_Learners_Guide_to_Programming_Using_the_Python_Language_2009_442p

Intro xxiii
1 Starting to Code: Finding Your Way--1
2 Textual Data: Every String Has Its Place--37
3 Functions: Let's Get Organized---77
4 Data Files and Arrays: Sort It Out---113
5 Hashes and Databases: Putting Data in Its Place---145
6 Modular Programming: Keeping Things Straight--177
7 Building a Graphical User Interface: Going All Gooey--215
8 GUIs and Data: Data Entry Widgets---257
81/2 Exceptions and Message Boxes: Get the Message?--293
9 Graphical Interface Elements: Selecting the Right Tool--313
10 Custom Widgets and Classes: With an Object in Mind---349
i Leftovers: The Top Ten Things (We Didn't Cover)--385

***Apress_Pro_Django_2nd_edition_2013_290p

About the Author--xvii
About the Technical Reviewers---xix
Acknowledgments---xxi
Preface---xxiii
Introduction--xxv
Chapter 1: Understanding Django---1
Chapter 2: Django Is Python--11
Chapter 3: Models--41
Chapter 4: URLs and Views--83
Chapter 5: Forms--107
Chapter 6: Templates---131
Chapter 7: Handling HTTP---157
Chapter 8: Backend Protocols--175
Chapter 9: Common Tools---201
Chapter 10: Coordinating Applications--225
Chapter 11: Enhancing Applications---247
Index---267

***A_Primer_on_Scientific_Programming_with_Python_2009_726p

1 Computing with Formulas--1
1.1 The First Programming Encounter: A Formula--1
1.1.1 Using a Program as a Calculator--2
1.1.2 About Programs and Programming---2
1.1.3 Tools for Writing Programs--3
1.1.4 Using Idle to Write the Program---4
1.1.5 How to Run the Program--7
1.1.6 Verifying the Result--8
1.1.7 Using Variables---8
1.1.8 Names of Variables--9

1.1.9 Reserved Words in Python--10
1.1.10 Comments---10
1.1.11 Formatting Text and Numbers--11
1.2 Computer Science Glossary---13
1.3 Another Formula: Celsius-Fahrenheit Conversion---18
1.3.1 Potential Error: Integer Division---19
1.3.2 Objects in Python---20
1.3.3 Avoiding Integer Division---21
1.3.4 Arithmetic Operators and Precedence--21
1.4 Evaluating Standard Mathematical Functions---22
1.4.1 Example: Using the Square Root Function--22
1.4.2 Example: Using More Mathematical Functions---25
1.4.3 A First Glimpse of Round-Off Errors---25
1.5 Interactive Computing---26
1.5.1 Calculating with Formulas in the Interactive Shell---------------------------------------27
1.5.2 Type Conversion---28
1.5.3 IPython---29
1.6 Complex Numbers---31
1.6.1 Complex Arithmetics in Python---32
1.6.2 Complex Functions in Python---32
1.6.3 Unified Treatment of Complex and Real Functions--------------------------------------33
1.7 Summary--35
1.7.1 Chapter Topics--35
1.7.2 Summarizing Example: Trajectory of a Ball---38
1.7.3 About Typesetting Conventions in This Book---39
1.8 Exercises--40
2 Basic Constructions---51
2.1 Loops and Lists for Tabular Data--51
2.1.1 A Naive Solution---51
2.1.2 While Loops---52
2.1.3 Boolean Expressions---54
2.1.4 Lists---56
2.1.5 For Loops--58
2.1.6 Alternative Implementations with Lists and Loops---------------------------------------60
2.1.7 Nested Lists--64
2.1.8 Printing Objects--65
2.1.9 Extracting Sublists---66
2.1.10 Traversing Nested Lists--68
2.1.11 Tuples---70
2.2 Functions---71
2.2.1 Functions of One Variable--71
2.2.2 Local and Global Variables---73
2.2.3 Multiple Arguments--75
2.2.4 Multiple Return Values--77
2.2.5 Functions with No Return Values---79
2.2.6 Keyword Arguments---80
2.2.7 Doc Strings--83
2.2.8 Function Input and Output---84
2.2.9 Functions as Arguments to Functions---84
2.2.10 The Main Program--86
2.2.11 Lambda Functions--87
2.3 If Tests---88

2.4 Summary--91
2.4.1 Chapter Topics---91
2.4.2 Summarizing Example: Tabulate a Function--94
2.4.3 How to Find More Python Information---98
2.5 Exercises---99
3 Input Data and Error Handling---119
3.1 Asking Questions and Reading Answers--120
3.1.1 Reading Keyboard Input--120
3.1.2 The Magic “eval” Function--121
3.1.3 The Magic “exec” Function--125
3.1.4 Turning String Expressions into Functions---126
3.2 Reading from the Command Line--127
3.2.1 Providing Input on the Command Line---127
3.2.2 A Variable Number of Command-Line Arguments------------------------------------128
3.2.3 More on Command-Line Arguments--129
3.2.4 Option–Value Pairs on the Command Line---130
3.3 Handling Errors---132
3.3.1 Exception Handling---133
3.3.2 Raising Exceptions--136
3.4 A Glimpse of Graphical User Interfaces--139
3.5 Making Modules--141
3.5.1 Example: Compund Interest Formulas--142
3.5.2 Collecting Functions in a Module File--143
3.5.3 Using Modules---148
3.6 Summary---150
3.6.1 Chapter Topics---150
3.6.2 Summarizing Example: Bisection Root Finding--------------------------------------152
3.7 Exercises---160
4 Array Computing and Curve Plotting---169
4.1 Vectors--170
4.1.1 The Vector Concept---170
4.1.2 Mathematical Operations on Vectors--171
4.1.3 Vector Arithmetics and Vector Functions---173
4.2 Arrays in Python Programs--175
4.2.1 Using Lists for Collecting Function Data--175
4.2.2 Basics of Numerical Python Arrays--176
4.2.3 Computing Coordinates and Function Values--177
4.2.4 Vectorization---178
4.3 Curve Plotting---179
4.3.1 The SciTools and Easyviz Packages--180
4.3.2 Plotting a Single Curve---181
4.3.3 Decorating the Plot---183
4.3.4 Plotting Multiple Curves--183
4.3.5 Controlling Line Styles---185
4.3.6 Interactive Plotting Sessions--189
4.3.7 Making Animations--190
4.3.8 Advanced Easyviz Topics---193
4.3.9 Curves in Pure Text---198
4.4 Plotting Difficulties---199
4.4.1 Piecewisely Defined Functions--199
4.4.2 Rapidly Varying Functions---205
4.4.3 Vectorizing StringFunction Objects--206

4.5 More on Numerical Python Arrays---207
4.5.1 Copying Arrays--207
4.5.2 In-Place Arithmetics---207
4.5.3 Allocating Arrays---208
4.5.4 Generalized Indexing---209
4.5.5 Testing for the Array Type--210
4.5.6 Equally Spaced Numbers---211
4.5.7 Compact Syntax for Array Generation---212
4.5.8 Shape Manipulation---212
4.6 Higher-Dimensional Arrays--213
4.6.1 Matrices and Arrays--213
4.6.2 Two-Dimensional Numerical Python Arrays---214
4.6.3 Array Computing--216
4.6.4 Two-Dimensional Arrays and Functions of Two Variables----------------------------------217
4.6.5 Matrix Objects---217
4.7 Summary--219
4.7.1 Chapter Topics---219
4.7.2 Summarizing Example: Animating a Function---220
4.8 Exercises---225
5 Sequences and Difference Equations---235
5.1 Mathematical Models Based on Difference Equations--236
5.1.1 Interest Rates---237
5.1.2 The Factorial as a Difference Equation--239
5.1.3 Fibonacci Numbers--240
5.1.4 Growth of a Population--241
5.1.5 Logistic Growth--242
5.1.6 Payback of a Loan--244
5.1.7 Taylor Series as a Difference Equation---245
5.1.8 Making a Living from a Fortune---246
5.1.9 Newton's Method--247
5.1.10 The Inverse of a Function---251
5.2 Programming with Sound--253
5.2.1 Writing Sound to File--253
5.2.2 Reading Sound from File---254
5.2.3 Playing Many Notes--255
5.3 Summary--256
5.3.1 Chapter Topics---256
5.3.2 Summarizing Example: Music of a Sequence---257
5.4 Exercises---260
6 Files, Strings, and Dictionaries---269
6.1 Reading Data from File--269
6.1.1 Reading a File Line by Line---270
6.1.2 Reading a Mixture of Text and Numbers---273
6.1.3 What Is a File, Really?---274
6.2 Dictionaries---278
6.2.1 Making Dictionaries---278
6.2.2 Dictionary Operations---279
6.2.3 Example: Polynomials as Dictionaries---280
6.2.4 Example: File Data in Dictionaries--282
6.2.5 Example: File Data in Nested Dictionaries--283
6.2.6 Example: Comparing Stock Prices---287
6.3 Strings--291

6.3.1 Common Operations on Strings--292
6.3.2 Example: Reading Pairs of Numbers---295
6.3.3 Example: Reading Coordinates--298
6.4 Reading Data from Web Pages--300
6.4.1 About Web Pages---300
6.4.2 How to Access Web Pages in Programs---302
6.4.3 Example: Reading Pure Text Files---302
6.4.4 Example: Extracting Data from an HTML Page--304
6.5 Writing Data to File--308
6.5.1 Example: Writing a Table to File--309
6.5.2 Standard Input and Output as File Objects--310
6.5.3 Reading and Writing Spreadsheet Files--312
6.6 Summary--317
6.6.1 Chapter Topics--317
6.6.2 Summarizing Example: A File Database---319
6.7 Exercises--323
7 Introduction to Classes---337
7.1 Simple Function Classes---338
7.1.1 Problem: Functions with Parameters---338
7.1.2 Representing a Function as a Class---340
7.1.3 Another Function Class Example---346
7.1.4 Alternative Function Class Implementations--347
7.1.5 Making Classes Without the Class Construct---349
7.2 More Examples on Classes---352
7.2.1 Bank Accounts---352
7.2.2 Phone Book--354
7.2.3 A Circle---355
7.3 Special Methods--356
7.3.1 The Call Special Method---357
7.3.2 Example: Automagic Differentiation--357
7.3.3 Example: Automagic Integration---360
7.3.4 Turning an Instance into a String--362
7.3.5 Example: Phone Book with Special Methods--363
7.3.6 Adding Objects--365
7.3.7 Example: Class for Polynomials--365
7.3.8 Arithmetic Operations and Other Special Methods--369
7.3.9 More on Special Methods for String Conversion---370
7.4 Example: Solution of Differential Equations--372
7.4.1 A Function for Solving ODEs---373
7.4.2 A Class for Solving ODEs--374
7.4.3 Verifying the Implementation--376
7.4.4 Example: Logistic Growth--377
7.5 Example: Class for Vectors in the Plane--378
7.5.1 Some Mathematical Operations on Vectors--378
7.5.2 Implementation--378
7.5.3 Usage---380
7.6 Example: Class for Complex Numbers--382
7.6.1 Implementation--382
7.6.2 Illegal Operations--383
7.6.3 Mixing Complex and Real Numbers--384
7.6.4 Special Methods for “Right” Operands--387
7.6.5 Inspecting Instances--388

7.7 Static Methods and Attributes--389
7.8 Summary---391
7.8.1 Chapter Topics---391
7.8.2 Summarizing Example: Interval Arithmetics---392
7.9 Exercises--397
8 Random Numbers and Simple Games---417
8.1 Drawing Random Numbers---418
8.1.1 The Seed---418
8.1.2 Uniformly Distributed Random Numbers---419
8.1.3 Visualizing the Distribution---420
8.1.4 Vectorized Drawing of Random Numbers--421
8.1.5 Computing the Mean and Standard Deviation---422
8.1.6 The Gaussian or Normal Distribution---423
8.2 Drawing Integers--424
8.2.1 Random Integer Functions--425
8.2.2 Example: Throwing a Die---426
8.2.3 Drawing a Random Element from a List---427
8.2.4 Example: Drawing Cards from a Deck--427
8.2.5 Example: Class Implementation of a Deck---429
8.3 Computing Probabilities--432
8.3.1 Principles of Monte Carlo Simulation---432
8.3.2 Example: Throwing Dice---433
8.3.3 Example: Drawing Balls from a Hat---435
8.3.4 Example: Policies for Limiting Population Growth---437
8.4 Simple Games--440
8.4.1 Guessing a Number--440
8.4.2 Rolling Two Dice---440
8.5 Monte Carlo Integration--443
8.5.1 Standard Monte Carlo Integration--443
8.5.2 Computing Areas by Throwing Random Points---446
8.6 Random Walk in One Space Dimension---447
8.6.1 Basic Implementation--448
8.6.2 Visualization---449
8.6.3 Random Walk as a Difference Equation---449
8.6.4 Computing Statistics of the Particle Positions---450
8.6.5 Vectorized Implementation---451
8.7 Random Walk in Two Space Dimensions---453
8.7.1 Basic Implementation--453
8.7.2 Vectorized Implementation---455
8.8 Summary--456
8.8.1 Chapter Topics--456
8.8.2 Summarizing Example: Random Growth--457
8.9 Exercises---463
9 Object-Oriented Programming--479
9.1 Inheritance and Class Hierarchies---479
9.1.1 A Class for Straight Lines---480
9.1.2 A First Try on a Class for Parabolas--481
9.1.3 A Class for Parabolas Using Inheritance---481
9.1.4 Checking the Class Type---483
9.1.5 Attribute versus Inheritance---484
9.1.6 Extending versus Restricting Functionality--485
9.1.7 Superclass for Defining an Interface---486

9.2 Class Hierarchy for Numerical Differentiation--488
9.2.1 Classes for Differentiation--488
9.2.2 A Flexible Main Program---491
9.2.3 Extensions--492
9.2.4 Alternative Implementation via Functions---495
9.2.5 Alternative Implementation via Functional Programming-----------------------------------496
9.2.6 Alternative Implementation via a Single Class---497
9.3 Class Hierarchy for Numerical Integration--499
9.3.1 Numerical Integration Methods---499
9.3.2 Classes for Integration--501
9.3.3 Using the Class Hierarchy--504
9.3.4 About Object-Oriented Programming--507
9.4 Class Hierarchy for Numerical Methods for ODEs--508
9.4.1 Mathematical Problem--508
9.4.2 Numerical Methods---510
9.4.3 The ODE Solver Class Hierarchy---511
9.4.4 The Backward Euler Method--515
9.4.5 Verification---518
9.4.6 Application 1: u′ = u--518
9.4.7 Application 2: The Logistic Equation---519
9.4.8 Application 3: An Oscillating System---521
9.4.9 Application 4: The Trajectory of a Ball---523
9.5 Class Hierarchy for Geometric Shapes---525
9.5.1 Using the Class Hierarchy--526
9.5.2 Overall Design of the Class Hierarchy---527
9.5.3 The Drawing Tool---529
9.5.4 Implementation of Shape Classes--530
9.5.5 Scaling, Translating, and Rotating a Figure---534
9.6 Summary--538
9.6.1 Chapter Topics---538
9.6.2 Summarizing Example: Input Data Reader--540
9.7 Exercises--546
A Discrete Calculus--573
A.1 Discrete Functions--573
A.1.1 The Sine Function---574
A.1.2 Interpolation--576
A.1.3 Evaluating the Approximation---576
A.1.4 Generalization--577
A.2 Differentiation Becomes Finite Differences--579
A.2.1 Differentiating the Sine Function--580
A.2.2 Differences on a Mesh---580
A.2.3 Generalization--582
A.3 Integration Becomes Summation---583
A.3.1 Dividing into Subintervals--584
A.3.2 Integration on Subintervals---585
A.3.3 Adding the Subintervals--586
A.3.4 Generalization--587
A.4 Taylor Series---589
A.4.1 Approximating Functions Close to One Point--589
A.4.2 Approximating the Exponential Function--589
A.4.3 More Accurate Expansions---590
A.4.4 Accuracy of the Approximation--592

A.4.5 Derivatives Revisited--594
A.4.6 More Accurate Difference Approximations---595
A.4.7 Second-Order Derivatives---597
A.5 Exercises--599
B Differential Equations---605
B.1 The Simplest Case---606
B.2 Exponential Growth---608
B.3 Logistic Growth--612
B.4 A General Ordinary Differential Equation--614
B.5 A Simple Pendulum---615
B.6 A Model for the Spread of a Disease---619
B.7 Exercises--621
C A Complete Project---625
C.1 About the Problem: Motion and Forces in Physics--626
C.1.1 The Physical Problem--626
C.1.2 The Computational Algorithm---628
C.1.3 Derivation of the Mathematical Model--628
C.1.4 Derivation of the Algorithm---631
C.2 Program Development and Testing---632
C.2.1 Implementation---632
C.2.2 Callback Functionality--635
C.2.3 Making a Module---636
C.2.4 Verification---637
C.3 Visualization---639
C.3.1 Simultaneous Computation and Plotting---639
C.3.2 Some Applications--642
C.3.3 Remark on Choosing Δt---643
C.3.4 Comparing Several Quantities in Subplots--644
C.3.5 Comparing Approximate and Exact Solutions---645
C.3.6 Evolution of the Error as Δt Decreases---646
C.4 Exercises--649
D Debugging--651
D.1 Using a Debugger---651
D.2 How to Debug--653
D.2.1 A Recipe for Program Writing and Debugging--654
D.2.2 Application of the Recipe---656
E Technical Topics---669
E.1 Different Ways of Running Python Programs---669
E.1.1 Executing Python Programs in IPython--669
E.1.2 Executing Python Programs on Unix---669
E.1.3 Executing Python Programs on Windows--671
E.1.4 Executing Python Programs on Macintosh--673
E.1.5 Making a Complete Stand-Alone Executable---673
E.2 Integer and Float Division---673
E.3 Visualizing a Program with Lumpy--674
E.4 Doing Operating System Tasks in Python---675
E.5 Variable Number of Function Arguments---678
E.5.1 Variable Number of Positional Arguments---679
E.5.2 Variable Number of Keyword Arguments---681
E.6 Evaluating Program Efficiency--683
E.6.1 Making Time Measurements---683
E.6.2 Profiling Python Programs--685

Bibliography 687
Index 689

List of Exercises

Exercise 1.1 Compute 1+1 42
Exercise 1.2 Write a “Hello, World!” program 43
Exercise 1.3 Convert from meters to British length units 43
Exercise 1.4 Compute the mass of various substances 43
Exercise 1.5 Compute the growth of money in a bank 43
Exercise 1.6 Find error(s) in a program 43
Exercise 1.7 Type in program text 43
Exercise 1.8 Type in programs and debug them 44
Exercise 1.9 Evaluate a Gaussian function 44
Exercise 1.10 Compute the air resistance on a football 45
Exercise 1.11 Define objects in IPython 45
Exercise 1.12 How to cook the perfect egg 46
Exercise 1.13 Evaluate a function defined by a sum 46
Exercise 1.14 Derive the trajectory of a ball 47
Exercise 1.15 Find errors in the coding of formulas 48
Exercise 1.16 Find errors in Python statements 48
Exercise 1.17 Find errors in the coding of a formula 49
Exercise 2.1 Make a Fahrenheit–Celsius conversion table 99
Exercise 2.2 Generate odd numbers 99
Exercise 2.3 Store odd numbers in a list 100
Exercise 2.4 Generate odd numbers by the range function 100
Exercise 2.5 Simulate operations on lists by hand 100
Exercise 2.6 Make a table of values from formula (1.1) 100
Exercise 2.7 Store values from formula (1.1) in lists 100
Exercise 2.8 Work with a list 100
Exercise 2.9 Generate equally spaced coordinates 100
Exercise 2.10 Use a list comprehension to solve Exer. 2.9 101
Exercise 2.11 Store data from Exer. 2.7 in a nested list 101
Exercise 2.12 Compute a mathematical sum 101
Exercise 2.13 Simulate a program by hand 101
Exercise 2.14 Use a for loop in Exer. 2.12 102
Exercise 2.15 Index a nested lists 102
Exercise 2.16 Construct a double for loop over a nested list 102
Exercise 2.17 Compute the area of an arbitrary triangle 102
Exercise 2.18 Compute the length of a path 102
Exercise 2.19 Approximate pi 103
Exercise 2.20 Write a Fahrenheit-Celsius conversion table 103
Exercise 2.21 Convert nested list comprehensions to nested standard loops 103
Exercise 2.22 Write a Fahrenheit–Celsius conversion function 104
Exercise 2.23 Write some simple functions 104
Exercise 2.24 Write the program in Exer. 2.12 as a function 104
Exercise 2.25 Implement a Gaussian function 104
Exercise 2.26 Find the max and min values of a function 104
Exercise 2.27 Explore the Python Library Reference 105
Exercise 2.28 Make a function of the formula in Exer. 1.12 105
Exercise 2.29 Write a function for numerical differentiation 105
Exercise 2.30 Write a function for numerical integration 105
Exercise 2.31 Improve the formula in Exer. 2.30 106
Exercise 2.32 Compute a polynomial via a product 106

Exercise 2.33 Implement the factorial function--106
Exercise 2.34 Compute velocity and acceleration from position data; one dimension----------107
Exercise 2.35 Compute velocity and acceleration from position data; two dimensions---------107
Exercise 2.36 Express a step function as a Python function---107
Exercise 2.37 Rewrite a mathematical function---108
Exercise 2.38 Make a table for approximations of cos x--108
Exercise 2.39 Implement Exer. 1.13 with a loop--109
Exercise 2.40 Determine the type of objects--109
Exercise 2.41 Implement the sum function--109
Exercise 2.42 Find the max/min elements in a list---110
Exercise 2.43 Demonstrate list functionality--110
Exercise 2.44 Write a sort function for a list of 4-tuples---110
Exercise 2.45 Find prime numbers--111
Exercise 2.46 Condense the program in Exer. 2.14---111
Exercise 2.47 Values of boolean expressions--112
Exercise 2.48 Explore round-off errors from a large number of inverse operations-------------112
Exercise 2.49 Explore what zero can be on a computer---112
Exercise 2.50 Resolve a problem with a function---113
Exercise 2.51 Compare two real numbers on a computer---113
Exercise 2.52 Use None in keyword arguments---114
Exercise 2.53 Improve the program from Ch. 2.4.2--114
Exercise 2.54 Interpret a code--114
Exercise 2.55 Explore problems with inaccurate indentation---------------------------------------115
Exercise 2.56 Find an error in a program---115
Exercise 2.57 Find programming errors--116
Exercise 2.58 Simulate nested loops by hand--117
Exercise 2.59 Explore punctuation in Python programs--117
Exercise 2.60 Investigate a for loop over a changing list---117
Exercise 3.1 Make an interactive program--160
Exercise 3.2 Read from the command line in Exer. 3.1--160
Exercise 3.3 Use exceptions in Exer. 3.2--160
Exercise 3.4 Read input from the keyboard--161
Exercise 3.5 Read input from the command line---161
Exercise 3.6 Prompt the user for input to the formula (1.1)---161
Exercise 3.7 Read command line input for the formula (1.1)--161
Exercise 3.8 Make the program from Exer. 3.7 safer--161
Exercise 3.9 Test more in the program from Exer. 3.7--161
Exercise 3.10 Raise an exception in Exer. 3.9---161
Exercise 3.11 Look up calendar functionality---162
Exercise 3.12 Use the StringFunction tool---162
Exercise 3.13 Extend a program from Ch. 3.2.1--162
Exercise 3.14 Why we test for specific exception types---162
Exercise 3.15 Make a simple module---163
Exercise 3.16 Make a useful main program for Exer. 3.15--163
Exercise 3.17 Make a module in Exer. 2.39--163
Exercise 3.18 Extend the module from Exer. 3.17--163
Exercise 3.19 Use options and values in Exer. 3.18---163
Exercise 3.20 Use optparse in the program from Ch. 3.2.4---163
Exercise 3.21 Compute the distance it takes to stop a car--163
Exercise 3.22 Check if mathematical rules hold on a computer-------------------------------------164
Exercise 3.23 Improve input to the program in Exer. 3.22--164
Exercise 3.24 Apply the program from Exer. 3.23--165

Exercise 3.25 Compute the binomial distribution 165
Exercise 3.26 Apply the binomial distribution 166
Exercise 3.27 Compute probabilities with the Poisson distribution 166
Exercise 4.1 Fill lists with function values 225
Exercise 4.2 Fill arrays; loop version 226
Exercise 4.3 Fill arrays; vectorized version 226
Exercise 4.4 Apply a function to a vector 226
Exercise 4.5 Simulate by hand a vectorized expression 226
Exercise 4.6 Demonstrate array slicing 227
Exercise 4.7 Plot the formula (1.1) 227
Exercise 4.8 Plot the formula (1.1) for several v0 values 227
Exercise 4.9 Plot exact and inexact Fahrenheit–Celsius formulas 227
Exercise 4.10 Plot the trajectory of a ball 227
Exercise 4.11 Plot a wave packet 227
Exercise 4.12 Use pyreport in Exer. 4.11 227
Exercise 4.13 Judge a plot 228
Exercise 4.14 Plot the viscosity of water 228
Exercise 4.15 Explore a function graphically 228
Exercise 4.16 Plot Taylor polynomial approximations to sin x 228
Exercise 4.17 Animate a wave packet 229
Exercise 4.18 Animate the evolution of Taylor polynomials 229
Exercise 4.19 Plot the velocity profile for pipeflow 230
Exercise 4.20 Plot the approximate function from Exer. 1.13 230
Exercise 4.21 Plot functions from the command line 231
Exercise 4.22 Improve the program from Exericse 4.21 231
Exercise 4.23 Demonstrate energy concepts from physics 231
Exercise 4.24 Plot a w-like function 231
Exercise 4.25 Plot a smoothed “hat” function 232
Exercise 4.26 Experience overflow in a function 232
Exercise 4.27 Experience less overflow in a function 233
Exercise 4.28 Extend Exer. 4.4 to a rank 2 array 233
Exercise 4.29 Explain why array computations fail 233
Exercise 5.1 Determine the limit of a sequence 260
Exercise 5.2 Determine the limit of a sequence 260
Exercise 5.3 Experience convergence problems 260
Exercise 5.4 Convergence of sequences with π as limit 261
Exercise 5.5 Reduce memory usage of difference equations 261
Exercise 5.6 Development of a loan over N months 261
Exercise 5.7 Solve a system of difference equations 261
Exercise 5.8 Extend the model (5.27)–(5.28) 261
Exercise 5.9 Experiment with the program from Exer. 5.8 262
Exercise 5.10 Change index in a difference equation 262
Exercise 5.11 Construct time points from dates 262
Exercise 5.12 Solve nonlinear equations by Newton’s method 263
Exercise 5.13 Visualize the convergence of Newton’s method 263
Exercise 5.14 Implement the Secant method 264
Exercise 5.15 Test different methods for root finding 264
Exercise 5.16 Difference equations for computing sin x 264
Exercise 5.17 Difference equations for computing cos x 265
Exercise 5.18 Make a guitar-like sound 265
Exercise 5.19 Damp the bass in a sound file 265
Exercise 5.20 Damp the treble in a sound file 266

Exercise 5.21 Demonstrate oscillatory solutions of (5.13)--266
Exercise 5.22 Make the program from Exer. 5.21 more flexible------------------------------------267
Exercise 5.23 Simulate the price of wheat---267
Exercise 6.1 Read a two-column data file---323
Exercise 6.2 Read a data file--323
Exercise 6.3 Simplify the implementation of Exer. 6.1--323
Exercise 6.4 Fit a polynomial to data--323
Exercise 6.5 Read acceleration data and find velocities---324
Exercise 6.6 Read acceleration data and plot velocities--325
Exercise 6.7 Find velocity from GPS coordinates---325
Exercise 6.8 Make a dictionary from a table--325
Exercise 6.9 Explore syntax differences: lists vs. dictionaries-------------------------------------326
Exercise 6.10 Improve the program from Ch. 6.2.4---326
Exercise 6.11 Interpret output from a program---326
Exercise 6.12 Make a dictionary---327
Exercise 6.13 Make a nested dictionary--327
Exercise 6.14 Make a nested dictionary from a file---327
Exercise 6.15 Compute the area of a triangle--327
Exercise 6.16 Compare data structures for polynomials--327
Exercise 6.17 Compute the derivative of a polynomial---327
Exercise 6.18 Generalize the program from Ch. 6.2.6--328
Exercise 6.19 Write function data to file--328
Exercise 6.20 Specify functions on the command line---328
Exercise 6.21 Interpret function specifications--329
Exercise 6.22 Compare average temperatures in two cities---330
Exercise 6.23 Compare average temperatures in many cities---------------------------------------330
Exercise 6.24 Plot the temperature in a city, 1995-today--331
Exercise 6.25 Plot temperatures in several cities---332
Exercise 6.26 Try Word or OpenOffice to write a program---332
Exercise 6.27 Evaluate objects in a boolean context---332
Exercise 6.28 Generate an HTML report---333
Exercise 6.29 Fit a polynomial to experimental data---333
Exercise 6.30 Interpret an HTML file with rainfall data--334
Exercise 6.31 Generate an HTML report with figures--334
Exercise 7.1 Make a function class---397
Exercise 7.2 Make a very simple class---398
Exercise 7.3 Extend the class from Ch. 7.2.1--398
Exercise 7.4 Make classes for a rectangle and a triangle---398
Exercise 7.5 Make a class for straight lines--398
Exercise 7.6 Improve the constructor in Exer. 7.5---398
Exercise 7.7 Make a class for quadratic functions--399
Exercise 7.8 Make a class for linear springs--399
Exercise 7.9 Implement Lagrange’s interpolation formula--399
Exercise 7.10 A very simple “Hello, World!” class--400
Exercise 7.11 Use special methods in Exer. 7.1---400
Exercise 7.12 Modify a class for numerical differentiation--400
Exercise 7.13 Make a class for nonlinear springs---401
Exercise 7.14 Extend the class from Ch. 7.2.1--401
Exercise 7.15 Implement a class for numerical differentation--401
Exercise 7.16 Verify a program--402
Exercise 7.17 Test methods for numerical differentation--402
Exercise 7.18 Make a class for summation of series---403

Exercise 7.19 Apply the differentiation class from Ch. 7.3.2---------------------------------------403
Exercise 7.20 Use classes for computing inverse functions---404
Exercise 7.21 Vectorize a class for numerical integration---404
Exercise 7.22 Speed up repeated integral calculations---404
Exercise 7.23 Solve a simple ODE in two ways---405
Exercise 7.24 Solve the ODE (B.36)--405
Exercise 7.25 Simulate a falling or rising body in a fluid--405
Exercise 7.26 Check the solution’s limit in Exer. 7.25--407
Exercise 7.27 Implement the modified Euler method; function-------------------------------------407
Exercise 7.28 Implement the modified Euler method; class---408
Exercise 7.29 Increase the flexibility in Exer. 7.28--408
Exercise 7.30 Solve an ODE specified on the command line---------------------------------------408
Exercise 7.31 Apply a polynomial class---409
Exercise 7.32 Find a bug in a class for polynomials--409
Exercise 7.33 Subtraction of polynomials---409
Exercise 7.34 Represent a polynomial by an array--409
Exercise 7.35 Vectorize a class for polynomials---409
Exercise 7.36 Use a dict to hold polynomial coefficients; add--------------------------------------410
Exercise 7.37 Use a dict to hold polynomial coefficients; mul--------------------------------------410
Exercise 7.38 Extend class Vec2D to work with lists/tuples---410
Exercise 7.39 Use NumPy arrays in class Vec2D--411
Exercise 7.40 Use classes in the program from Ch. 6.6.2--411
Exercise 7.41 Use a class in Exer. 6.28---412
Exercise 7.42 Apply the class from Exer. 7.41 interactively---413
Exercise 7.43 Find the optimal production for a company---413
Exercise 7.44 Extend the program from Exer. 7.43---415
Exercise 7.45 Model the economy of fishing---415
Exercise 8.1 Flip a coin N times---463
Exercise 8.2 Compute a probability---463
Exercise 8.3 Choose random colors---463
Exercise 8.4 Draw balls from a hat--464
Exercise 8.5 Probabilities of rolling dice---464
Exercise 8.6 Estimate the probability in a dice game--464
Exercise 8.7 Decide if a dice game is fair--464
Exercise 8.8 Adjust the game in Exer. 8.7--464
Exercise 8.9 Probabilities of throwing two dice---465
Exercise 8.10 Compute the probability of drawing balls--465
Exercise 8.11 Compute the probability of hands of cards--465
Exercise 8.12 Play with vectorized boolean expressions--466
Exercise 8.13 Vectorize the program from Exer. 8.1--466
Exercise 8.14 Vectorize the code in Exer. 8.2---466
Exercise 8.15 Throw dice and compute a small probability--466
Exercise 8.16 Difference equation for random numbers--466
Exercise 8.17 Make a class for drawing balls from a hat--467
Exercise 8.18 Independent vs. dependent random numbers--467
Exercise 8.19 Compute the probability of flipping a coin--467
Exercise 8.20 Extend Exer. 8.19---468
Exercise 8.21 Simulate the problems in Exer. 3.26--468
Exercise 8.22 Simulate a poker game--468
Exercise 8.23 Write a non-vectorized version of a code--469
Exercise 8.24 Estimate growth in a simulation model--469
Exercise 8.25 Investigate guessing strategies for Ch. 8.4.1---469

Exercise 8.26 Make a vectorized solution to Exer. 8.7---469
Exercise 8.27 Compare two playing strategies--470
Exercise 8.28 Solve Exercise 8.27 with different no. of dice---470
Exercise 8.29 Extend Exercise 8.28---470
Exercise 8.30 Compute π by a Monte Carlo method--470
Exercise 8.31 Do a variant of Exer. 8.30--470
Exercise 8.32 Compute π by a random sum---470
Exercise 8.33 1D random walk with drift---470
Exercise 8.34 1D random walk until a point is hit---471
Exercise 8.35 Make a class for 2D random walk---471
Exercise 8.36 Vectorize the class code from Exer. 8.35--471
Exercise 8.37 2D random walk with walls; scalar version---472
Exercise 8.38 2D random walk with walls; vectorized version-------------------------------------472
Exercise 8.39 Simulate the mixture of gas molecules--472
Exercise 8.40 Simulate the mixture of gas molecules--473
Exercise 8.41 Guess beer brands--473
Exercise 8.42 Simulate stock prices--473
Exercise 8.43 Compute with option prices in finance--474
Exercise 8.44 Compute velocity and acceleration---475
Exercise 8.45 Numerical differentiation of noisy signals--475
Exercise 8.46 Model the noise in the data in Exer. 8.44--476
Exercise 8.47 Reduce the noise in Exer. 8.44--477
Exercise 8.48 Find the expected waiting time in traffic lights----------------------------------477
Exercise 9.1 Demonstrate the magic of inheritance--546
Exercise 9.2 Inherit from classes in Ch. 9.1--546
Exercise 9.3 Inherit more from classes in Ch. 9.1--547
Exercise 9.4 Reverse the class hierarchy from Ch. 9.1--547
Exercise 9.5 Super- and subclass for a point---547
Exercise 9.6 Modify a function class by subclassing---547
Exercise 9.7 Explore the accuracy of difference formulas---------------------------------------548
Exercise 9.8 Implement a subclass---548
Exercise 9.9 Make classes for numerical differentiation--548
Exercise 9.10 Implement a new subclass for differentiation--------------------------------------548
Exercise 9.11 Understand if a class can be used recursively-------------------------------------549
Exercise 9.12 Represent people by a class hierarchy--549
Exercise 9.13 Add a new class in a class hierarchy---550
Exercise 9.14 Change the user interface of a class hierarchy---------------------------------------550
Exercise 9.15 Compute convergence rates of numerical integration methods-------------------550
Exercise 9.16 Add common functionality in a class hierarchy--------------------------------------551
Exercise 9.17 Make a class hierarchy for root finding--552
Exercise 9.18 Use the ODESolver hierarchy to solve a simple ODE------------------------------552
Exercise 9.19 Use the 4th-order Runge-Kutta on (B.34)---552
Exercise 9.20 Solve an ODE until constant solution--552
Exercise 9.21 Use classes in Exer. 9.20---553
Exercise 9.22 Scale away parameters in Exer. 9.20---553
Exercise 9.23 Compare ODE methods--553
Exercise 9.24 Solve two coupled ODEs for radioactive decay-------------------------------------554
Exercise 9.25 Compare methods for solving the ODE (B.36)-------------------------------------554
Exercise 9.26 Code a 2nd-order Runge-Kutta method; function-----------------------------------554
Exercise 9.27 Code a 2nd-order Runge-Kutta method; class---------------------------------------555
Exercise 9.28 Implement a midpoint method for ODEs--555
Exercise 9.29 Implement a modified Euler method for ODEs-----------------------------------555

Exercise 9.30 Improve the implementation in Exer. 7.25--555
Exercise 9.31 Visualize the different forces in Exer. 9.30--556
Exercise 9.32 Find the body’s position in Exer. 9.30---556
Exercise 9.33 Compare methods for solving (B.37)–(B.38)--556
Exercise 9.34 Add the effect of air resistance on a ball--557
Exercise 9.35 Make a class for drawing an arrow--557
Exercise 9.36 Make a class for drawing a person--557
Exercise 9.37 Animate a person with waving hands---558
Exercise 9.38 Make a class for drawing a car--558
Exercise 9.39 Make a car roll---558
Exercise 9.40 Make a class for differentiating noisy data--558
Exercise 9.41 Find local and global extrema of a function---559
Exercise 9.42 Improve the accuracy in Exer. 9.41---560
Exercise 9.43 Make a calculus calculator class--561
Exercise 9.44 Extend Exer. 9.43---561
Exercise 9.45 Formulate a 2nd-order ODE as a system---562
Exercise 9.46 Solve the system in Exer. 9.45 in a special case------------------------------------563
Exercise 9.47 Enhance the code from Exer. 9.46---563
Exercise 9.48 Make a tool for analyzing oscillatory solutions-------------------------------------565
Exercise 9.49 Replace functions by class in Exer. 9.46--566
Exercise 9.50 Allow flexible choice of functions in Exer. 9.49------------------------------------569
Exercise 9.51 Use the modules from Exer. 9.49 and 9.50--570
Exercise 9.52 Use the modules from Exer. 9.49 and 9.50--571
Exercise A.1 Interpolate a discrete function---599
Exercise A.2 Study a function for different parameter values--------------------------------------599
Exercise A.3 Study a function and its derivative--600
Exercise A.4 Use the Trapezoidal method---600
Exercise A.5 Compute a sequence of integrals--601
Exercise A.6 Use the Trapezoidal method---601
Exercise A.7 Trigonometric integrals--602
Exercise A.8 Plot functions and their derivatives---602
Exercise A.9 Use the Trapezoidal method---603
Exercise B.1 Solve a nonhomogeneous linear ODE--621
Exercise B.2 Solve a nonlinear ODE--621
Exercise B.3 Solve an ODE for y(x)--621
Exercise B.4 Experience instability of an ODE---622
Exercise B.5 Solve an ODE for the arc length--622
Exercise B.6 Solve an ODE with time-varying growth---622
Exercise B.7 Solve an ODE for emptying a tank---623
Exercise B.8 Solve an ODE system for an electric circuit---623
Exercise C.1 Use a w function with a step--649
Exercise C.2 Make a callback function in Exercise C.1--649
Exercise C.3 Improve input to the simulation program---650

***Beginning_Game_Development_with_Python_and_Pygame_From_Novice_to_Professional_2007_342p

About the Author--xv
About the Technical Reviewer--.xvii
Acknowledgments---xix
Introduction--xxi

CHAPTER 1 Introducing Python--1
CHAPTER 2 Exploring Python--19
CHAPTER 3 Introducing Pygame--41
CHAPTER 4 Creating Visuals--67
CHAPTER 5 Making Things Move--91
CHAPTER 6 Accepting User Input---111
CHAPTER 7 Take Me to Your Leader--139
CHAPTER 8 Moving into the Third Dimension--165
CHAPTER 9 Exploring the Third Dimension--181
CHAPTER 10 Making Things Go Boom--211
CHAPTER 11 Lights, Camera, Action!---235
CHAPTER 12 Setting the Scene with OpenGL---263
APPENDIX A Game Object Reference---285
APPENDIX B Packaging Your Game--293
INDEX--297

***Beginning_Python_2005_679p

Acknowledgments---xxix
Introduction---xxxi
Chapter 1: Programming Basics and Strings--1
How Programming Is Different from Using a Computer---------------------------------------1
Programming Is Consistency---2
Programming Is Control---2
Programming Copes with Change--2
What All That Means Together---3
The First Steps--3
Starting codeEditor---3
Using codeEditor’s Python Shell---4
Try It Out: Starting the Python Shell--4
Beginning to Use Python — Strings--5
What Is a String?--5
Why the Quotes?--6
Try It Out: Entering Strings with Different Quotes---6
Understanding Different Quotes---6
Putting Two Strings Together---8
Try It Out: Using + to Combine Strings---8
Putting Strings Together in Different Ways--9
Try It Out: Using a Format Specifier to Populate a String--9
Try It Out: More String Formatting--9
Displaying Strings with Print--10
Try It Out: Printing Text with Print---10
Summary--10
Exercises---11
Chapter 2: Numbers and Operators---13
Different Kinds of Numbers--13
Numbers in Python---14
Try It Out: Using Type with Different Numbers--14
Try It Out: Creating an Imaginary Number--15
Program Files--15
Try It Out: Using the Shell with the Editor---16
Using the Different Types---17

Try It Out Including Different Numbers in Strings--18
Try It Out: Escaping the % Sign in Strings--18
Basic Math--19
Try It Out Doing Basic Math---19
Try It Out: Using the Modulus Operation--20
Some Surprises--20
Try It Out: Printing the Results--21
Using Numbers---21
Order of Evaluation--21
Try It Out: Using Math Operations---21
Number Formats---22
Try It Out: Using Number Formats---22
Mistakes Will Happen---23
Try It Out: Making Mistakes---23
Some Unusual Cases--24
Try It Out: Formatting Numbers as Octal and Hexadecimal---24
Summary---24
Exercises---25
Chapter 3: Variables — Names for Values---27
Referring to Data – Using Names for Data---27
Try It Out: Assigning Values to Names---28
Changing Data Through Names--28
Try It Out: Altering Named Values--29
Copying Data---29
Names You Can’t Use and Some Rules---29
Using More Built-in Types--30
Tuples — Unchanging Sequences of Data---30
Try It Out: Creating and Using a Tuple---30
Try It Out: Accessing a Tuple Through Another Tuple--31
Lists — Changeable Sequences of Data--33
Try It Out Viewing the Elements of a List---33
Dictionaries — Groupings of Data Indexed by Name---34
Try It Out: Making a Dictionary---34
Try It Out: Getting the Keys from a Dictionary---35
Treating a String Like a List--36
Special Types---38
Other Common Sequence Properties--38
Referencing the Last Elements---38
Ranges of Sequences---39
Try It Out: Slicing Sequences--39
Growing Lists by Appending Sequences--40
Using Lists to Temporarily Store Data--40
Try It Out: Popping Elements from a List--40
Summary---41
Exercises---42
Chapter 4: Making Decisions---43
Comparing Values — Are They the Same?--43
Try It Out: Comparing Values for Sameness---43
Doing the Opposite — Not Equal--45
Try It Out: Comparing Values for Difference--45
Comparing Values — Which One Is More?---45
Try It Out: Comparing Greater Than and Less Than---45

More Than or Equal, Less Than or Equal--47
Reversing True and False--47
Try It Out: Reversing the Outcome of a Test---47
Looking for the Results of More Than One Comparison---48
How to Get Decisions Made---48
Try It Out: Placing Tests within Tests---49
Repetition---51
How to Do Something — Again and Again--51
Try It Out: Using a while Loop---51
Stopping the Repetition---52
Try It Out: Using else While Repeating---54
Try It Out: Using continue to Keep Repeating--54
Handling Errors---55
Trying Things Out--55
Try It Out: Creating an Exception with Its Explanation--56
Summary--57
Exercises---58
Chapter 5: Functions---59
Putting Your Program into Its Own File---59
Try It Out: Run a Program with Python -i---61
Functions: Grouping Code under a Name---61
Try It Out: Defining a Function---61
Choosing a Name--62
Describing a Function in the Function--63
Try It Out: Displaying __doc__--63
The Same Name in Two Different Places--64
Making Notes to Yourself--65
Try It Out: Experimenting with Comments---65
Asking a Function to Use a Value You Provide---66
Try It Out Invoking a Function with Parameters--67
Checking Your Parameters---68
Try It Out: Determining More Types with the type Function---69
Try It Out: Using Strings to Compare Types---69
Setting a Default Value for a Parameter — Just in Case--70
Try It Out: Setting a Default Parameter--70
Calling Functions from within Other Functions--71
Try It Out: Invoking the Completed Function--72
Functions Inside of Functions--72
Flagging an Error on Your Own Terms--73
Layers of Functions---74
How to Read Deeper Errors--74
Summary--75
Exercises---76
Chapter 6: Classes and Objects---79
Thinking About Programming---79
Objects You Already Know---79
Looking Ahead: How You Want to Use Objects--81
Defining a Class--81
How Code Can Be Made into an Object--81
Try It Out: Defining a Class--82
Try It Out: Creating an Object from Your Class--82
Try It Out: Writing an Internal Method---84

Try It Out: Writing Interface Methods--85
Try It Out: Using More Methods--87
Objects and Their Scope---89
Try It Out: Creating Another Class--89
Summary---92
Exercises---93
Chapter 7: Organizing Programs---95
Modules--96
Importing a Module So That You Can Use It--96
Making a Module from Pre-existing Code--97
Try It Out: Creating a Module---97
Try It Out: Exploring Your New Module---98
Using Modules — Starting With the Command Line--99
Try It Out: Printing sys.argv--100
Changing How Import Works — Bringing in More--101
Packages--101
Try It Out: Making the Files in the Kitchen Class--102
Modules and Packages---103
Bringing Everything into the Current Scope---103
Try It Out: Exporting Modules from a Package---104
Re-importing Modules and Packages---104
Try It Out: Examining sys.modules---105
Basics of Testing Your Modules and Packages---106
Summary--106
Exercises--107
Chapter 8: Files and Directories--109
File Objects---109
Writing Text Files---110
Reading Text Files--111
Try It Out: Printing the Lengths of Lines in the Sample File---------------------------------------112
File Exceptions--113
Paths and Directories---113
Paths---114
Directory Contents--116
Try It Out: Getting the Contents of a Directory--116
Try It Out: Listing the Contents of Your Desktop or Home Directory------------------------------118
Obtaining Information about Files--118
Recursive Directory Listings---118
Renaming, Moving, Copying, and Removing Files---119
Example: Rotating Files--120
Creating and Removing Directories--121
Globbing---122
Pickles---123
Try It Out: Creating a Pickle File---123
Pickling Tips---124
Efficient Pickling--125
Summary--125
Exercises--125
Chapter 9: Other Features of the Language--127
Lambda and Filter: Short Anonymous Functions---127
Reduce---128
Try It Out: Working with Reduce--128

Map: Short-Circuiting Loops--129
Try It Out: Use Map--129
Decisions within Lists — List Comprehension--130
Generating Lists for Loops---131
Try It Out: Examining an xrange Object---132
Special String Substitution Using Dictionaries--133
Try It Out: String Formatting with Dictionaries--133
Featured Modules---134
Getopt — Getting Options from the Command Line--134
Using More Than One Process---137
Threads — Doing Many Things in the Same Process---139
Storing Passwords--140
Summary---141
Exercises---142
Chapter 10: Building a Module--143
Exploring Modules--143
Importing Modules--145
Finding Modules---145
Digging through Modules---146
Creating Modules and Packages--150
Try It Out: Creating a Module with Functions--150
Working with Classes---151
Defining Object-Oriented Programming---151
Creating Classes--151
Try It Out: Creating a Meal Class--152
Extending Existing Classes---153
Finishing Your Modules--154
Defining Module-Specific Errors---154
Choosing What to Export--155
Documenting Your Modules---156
Try It Out: Viewing Module Documentation---157
Testing Your Module---162
Running a Module as a Program---164
Try It Out: Running a Module---164
Creating a Whole Module---165
Try It Out: Finishing a Module--165
Try It Out: Smashing Imports---169
Installing Your Modules---170
Try It Out: Creating an Installable Package--171
Summary---174
Exercises---174
Chapter 11: Text Processing--175
Why Text Processing Is So Useful--175
Searching for Files---176
Clipping Logs--177
Sifting through Mail---178
Navigating the File System with the os Module--178
Try It Out: Listing Files and Playing with Paths---180
Try It Out: Searching for Files of a Particular Type---181
Try It Out: Refining a Search--183
Working with Regular Expressions and the re Module---184
Try It Out: Fun with Regular Expressions--186

Try It Out: Adding Tests--187
Summary--189
Exercises---189
Chapter 12: Testing---191
Assertions---191
Try It Out: Using Assert---192
Test Cases and Test Suites--193
Try It Out: Testing Addition---194
Try It Out: Testing Faulty Addition--195
Test Fixtures---196
Try It Out: Working with Test Fixtures--197
Putting It All Together with Extreme Programming--199
Implementing a Search Utility in Python--200
Try It Out: Writing a Test Suite First---201
Try It Out: A General-Purpose Search Framework--203
A More Powerful Python Search---205
Try It Out: Extending the Search Framework--206
Formal Testing in the Software Life Cycle---207
Summary--208
Chapter 13: Writing a GUI with Python--209
GUI Programming Toolkits for Python---209
PyGTK Introduction---210
pyGTK Resources--211
Creating GUI Widgets with pyGTK---213
Try It Out: Writing a Simple pyGTK Program---213
GUI Signals---214
GUI Helper Threads and the GUI Event Queue---216
Try It Out: Writing a Multithreaded pyGTK App--219
Widget Packing---222
Glade: a GUI Builder for pyGTK---223
GUI Builders for Other GUI Frameworks--224
Using libGlade with Python--225
A Glade Walkthrough--225
Starting Glade--226
Creating a Project--227
Using the Palette to Create a Window---227
Putting Widgets into the Window--228
Glade Creates an XML Representation of the GUI---230
Try It Out: Building a GUI from a Glade File--231
Creating a Real Glade Application---231
Advanced Widgets---238
Further Enhancing PyRAP--241
Summary--248
Exercises---248
Chapter 14: Accessing Databases--249
Working with DBM Persistent Dictionaries--250
Choosing a DBM Module---250
Creating Persistent Dictionaries--251
Try It Out: Creating a Persistent Dictionary---251
Accessing Persistent Dictionaries--252
Try It Out: Accessing Persistent Dictionaries---253
Deciding When to Use DBM and When to Use a Relational Database---------------------------255

Working with Relational Databases---255
Writing SQL Statements---257
Defining Tables--259
Setting Up a Database---260
Try It Out: Creating a Gadfly Database---261
Using the Python Database APIs---262
Downloading Modules---263
Creating Connections--263
Working with Cursors--264
Try It Out: Inserting Records--264
Try It Out: Writing a Simple Query---266
Try It Out: Writing a Complex Join---267
Try It Out: Updating an Employee’s Manager--269
Try It Out: Removing Employees---270
Working with Transactions and Committing the Results--271
Examining Module Capabilities and Metadata--272
Handling Errors---272
Summary--273
Exercises--274
Chapter 15: Using Python for XML--275
What Is XML?---275
A Hierarchical Markup Language--275
A Family of Standards---277
What Is a Schema/DTD?--278
What Are Document Models For?---278
Do You Need One?--278
Document Type Definitions---278
An Example DTD---278
DTDs Aren’t Exactly XML---280
Limitations of DTDs--280
Schemas---280
An Example Schema---280
Schemas Are Pure XML---281
Schemas Are Hierarchical--281
Other Advantages of Schemas--281
Schemas Are Less Widely Supported--281
XPath---282
HTML as a Subset of XML--282
The HTML DTDs--283
HTMLParser---283
Try It Out: Using HTMLParser--283
htmllib--284
Try It Out: Using htmllib---284
XML Libraries Available for Python---285
Validating XML Using Python---285
What Is Validation?---286
Well-Formedness versus Validation--286
Available Tools--286
Try It Out: Validation Using xmlproc---286
What Is SAX?--287
Stream-based---288
Event-driven---288

What Is DOM?--288
In-memory Access--288
Why Use SAX or DOM--289
Capability Trade-Offs--289
Memory Considerations--289
Speed Considerations--289
SAX and DOM Parsers Available for Python--289
PyXML--290
xml.sax--290
xml.dom.minidom--290
Try It Out: Working with XML Using DOM--290
Try It Out: Working with XML Using SAX--292
Intro to XSLT--293
XSLT Is XML--293
Transformation and Formatting Language--293
Functional, Template-Driven--293
Using Python to Transform XML Using XSLT--294
Try It Out: Transforming XML with XSLT--294
Putting It All Together: Working with RSS--296
RSS Overview and Vocabulary--296
Making Sense of It All--296
RSS Vocabulary--297
An RSS DTD--297
A Real-World Problem--297
Try It Out: Creating an RSS Feed--298
Creating the Document--300
Checking It Against the DTD--301
Another Real-World Problem--301
Try It Out: Creating An Aggregator--301
Summary--303
Exercises--303
Chapter 16: Network Programming--305
Try It Out: Sending Some E-mail--305
Understanding Protocols--307
Comparing Protocols and Programming Languages--307
The Internet Protocol Stack--308
A Little Bit About the Internet Protocol--309
Internet Addresses--309
Internet Ports--310
Sending Internet E-mail--311
The E-mail File Format...311
MIME Messages...313
MIME Encodings: Quoted-printable and Base64...313
MIME Content Types...314
Try It Out: Creating a MIME Message with an Attachment...315
MIME Multipart Messages...316
Try It Out: Building E-mail Messages with SmartMessage...320
Sending Mail with SMTP and smtplib...321
Try It Out: Sending Mail with MailServer...323
Retrieving Internet E-mail...323
Parsing a Local Mail Spool with mailbox...323
Try It Out: Printing a Summary of Your Mailbox...324

Fetching Mail from a POP3 Server with poplib....325
Try It Out: Printing a Summary of Your POP3 Mailbox....327
Fetching Mail from an IMAP Server with imaplib....327
Try It Out: Printing a Summary of Your IMAP Mailbox....329
IMAP’s Unique Message IDs....330
Try It Out: Fetching a Message by Unique ID....330
Secure POP3 and IMAP....331
Webmail Applications Are Not E-mail Applications....331
Socket Programming....331
Introduction to Sockets....332
Try It Out: Connecting to the SuperSimpleSocketServer with Telnet....333
Binding to an External Hostname....334
The Mirror Server....335
Try It Out: Mirroring Text with the MirrorServer....336
The Mirror Client....336
SocketServer....337
Multithreaded Servers....339
The Python Chat Server....340
Design of the Python Chat Server....340
The Python Chat Server Protocol....341
Our Hypothetical Protocol in Action....341
Initial Connection....342
Chat Text....342
Server Commands....342
General Guidelines....343
The Python Chat Client....346
Single-Threaded Multitasking with select....348
Other Topics....350
Miscellaneous Considerations for Protocol Design....350
Trusted Servers....350
Terse Protocols....350
The Twisted Framework....351
Deferred Objects....351
The Peer-to-Peer Architecture....354
Summary....354
Exercises....354
Chapter 17: Extension Programming with C....355
Extension Module Outline....356
Building and Installing Extension Modules....358
Passing Parameters from Python to C....360
Returning Values from C to Python....363
The LAME Project....364
The LAME Extension Module....368
Using Python Objects from C Code....380
Summary....383
Exercises....383
Chapter 18: Writing Shareware and Commercial Programs....385
A Case Study: Background....385
How Much Python Should You Use?....386
Pure Python Licensing....387
Web Services Are Your Friend....388
Pricing Strategies....389

Watermarking..390
Other Models..394
Selling as a Platform, Rather Than a Product..395
Your Development Environment...395
Finding Python Programmers...396
Training non-Python Programmers..397
Python Employment Resources..397
Python Problems..397
Porting to Other Versions of Python...397
Porting to Other Operating Systems..398
Debugging Threads..399
Common Gotchas...399
Portable Distribution...400
Essential Libraries...401
Timeoutsocket...401
PyGTK...402
GEOip...402
Summary...403
Chapter 19: Numerical Programming..405
Numbers in Python...405
Integers...406
Long Integers..406
Floating-point Numbers..407
Formatting Numbers..408
Characters as Numbers...410
Mathematics...412
Arithmetic..412
Built-in Math Functions...414
The math Module...415
Complex Numbers...416
Arrays...418
The array Module..420
The numarray Package...422
Using Arrays...422
Computing the Standard Deviation...423
Summary...424
Exercises...425
Chapter 20: Python in the Enterprise...427
Enterprise Applications..428
Document Management...428
The Evolution of Document Management Systems...429
What You Want in a Document Management System...430
People in Directories..431
Taking Action with Workflow..432
Auditing, Sarbanes-Oxley, and What You Need to Know..433
Auditing and Document Management..434
Working with Actual Enterprise Systems...435
Introducing the wftk Workflow Toolkit...435
Try It Out: Very Simple Record Retrieval..436
Try It Out: Very Simple Record Storage..438
Try It Out: Data Storage in MySQL...439
Try It Out: Storing and Retrieving Documents...441

Try It Out: A Document Retention Framework....446
The python-ldap Module....448
Try It Out: Using Basic OpenLDAP Tools....449
Try It Out: Simple LDAP Search....451
More LDAP....453
Back to the wftk....453
Try It Out: Simple Workflow Trigger....454
Try It Out: Action Queue Handler....456
Summary....458
Exercises....458
Chapter 21: Web Applications and Web Services....459
REST: The Architecture of the Web....460
Characteristics of REST....460
A Distributed Network of Interlinked Documents....461
A Client-Server Architecture....461
Servers Are Stateless....461
Resources....461
Representations....462
REST Operations....462
HTTP: Real-World REST....463
Try It Out: Python's Three-Line Web Server....463
The Visible Web Server....464
Try It Out: Seeing an HTTP Request and Response....465
The HTTP Request....466
The HTTP Response....467
CGI: Turning Scripts into Web Applications....468
Try It Out: Running a CGI Script....469
The Web Server Makes a Deal with the CGI Script....470
CGI's Special Environment Variables....471
Accepting User Input through HTML Forms....473
The cgi Module: Parsing HTML Forms....474
Try It Out: Printing Any HTML Form Submission....478
Building a Wiki....480
The BittyWiki Core Library....481
Back-end Storage....481
WikiWords....481
Writing the BittyWiki Core....481
Try It Out: Creating Wiki Pages from an Interactive Python Session....483
The BittyWiki Web Interface....484
Resources....484
Request Structure....484
But Wait — There's More (Resources)....485
Wiki Markup....486
Web Services....493
How Web Services Work....494
REST Web Services....494
REST Quick Start: Finding Bargains on Amazon.com....495
Try It Out: Peeking at an Amazon Web Services Response....496
Introducing WishListBargainFinder....497
Giving BittyWiki a REST API....500
Wiki Search-and-Replace Using the REST Web Service....503
Try It Out: Wiki Searching and Replacing....507

XML-RPC....508
XML-RPC Quick Start: Get Tech News from Meerkat....509
The XML-RPC Request....511
Representation of Data in XML-RPC....512
The XML-RPC Response....513
If Something Goes Wrong....513
Exposing the BittyWiki API through XML-RPC....514
Try It Out: Manipulating BittyWiki through XML-RPC....517
Wiki Search-and-Replace Using the XML-RPC Web Service....518
SOAP....520
SOAP Quick Start: Surfing the Google API....520
The SOAP Request....522
The SOAP Response....524
If Something Goes Wrong....524
Exposing a SOAP Interface to BittyWiki....525
Try It Out: Manipulating BittyWiki through SOAP....526
Wiki Search-and-Replace Using the SOAP Web Service....527
Documenting Your Web Service API....529
Human-Readable API Documentation....529
The BittyWiki REST API Document....529
The BittyWiki XML-RPC API Document....529
The BittyWiki SOAP API Document....530
The XML-RPC Introspection API....530
Try It Out: Using the XML-RPC Introspection API....530
WSDL....531
Try It Out: Manipulating BittyWiki through a WSDL Proxy....533
Choosing a Web Service Standard....534
Web Service Etiquette....535
For Consumers of Web Services....535
For Producers of Web Services....535
Using Web Applications as Web Services....536
A Sampling of Publicly Available Web Services....536
Summary....538
Exercises....538
Chapter 22: Integrating Java with Python....539
Scripting within Java Applications....540
Comparing Python Implementations....541
Installing Jython....541
Running Jython....542
Running Jython Interactively....542
Try It Out: Running the Jython Interpreter....542
Running Jython Scripts....543
Try It Out Running a Python Script....543
Controlling the jython Script....544
Making Executable Commands....545
Try It Out: Making an Executable Script....546
Running Jython on Your Own....546
Packaging Jython-Based Applications....547
Integrating Java and Jython....547
Using Java Classes in Jython....548
Try It Out: Calling on Java Classes....548
Try It Out: Creating a User Interface from Jython....550

Accessing Databases from Jython..552
Working with the Python DB API..553
Setting Up a Database..554
Try It Out: Create Tables...555
Writing J2EE Servlets in Jython...558
Setting Up an Application Server...559
Adding the PyServlet to an Application Server...560
Extending HttpServlet..561
Try It Out: Writing a Python Servlet..562
Choosing Tools for Jython...564
Testing from Jython...565
Try It Out: Exploring Your Environment with Jython...565
Embedding the Jython Interpreter..566
Calling Jython Scripts from Java..566
Try It Out: Embedding Jython---567
Compiling Python Code to Java--568
Handling Differences between C Python and Jython---569
Summary--570
Exercises--571
Appendix A: Answers to Exercises---573
Appendix B: Online Resources---605
Appendix C: What’s New in Python 2.4--609
Glossary---613
Index---623

***Beginning_Python_From_Novice_to_Professional_Second_Edition_2008_666p

About the Author---xxiii
About the Technical Reviewer--xxv
Preface---xxvii
Introduction---xxix
CHAPTER 1 Instant Hacking: The Basics---1
CHAPTER 2 Lists and Tuples--31
CHAPTER 3 Working with Strings--53
CHAPTER 4 Dictionaries: When Indices Won’t Do---69
CHAPTER 5 Conditionals, Loops, and Some Other Statements--------------------------------------83
CHAPTER 6 Abstraction--113
CHAPTER 7 More Abstraction---141
CHAPTER 8 Exceptions--161
CHAPTER 9 Magic Methods, Properties, and Iterators---175
CHAPTER 10 Batteries Included---209
CHAPTER 11 Files and Stuff--261
CHAPTER 12 Graphical User Interfaces---277
CHAPTER 13 Database Support---293
CHAPTER 14 Network Programming---305
CHAPTER 15 Python and the Web--321
CHAPTER 16 Testing, 1-2-3---349
CHAPTER 17 Extending Python---365
CHAPTER 18 Packaging Your Programs---383
CHAPTER 19 Playful Programming---393

CHAPTER 20 Project 1: Instant Markup--403
CHAPTER 21 Project 2: Painting a Pretty Picture--425
CHAPTER 22 Project 3: XML for All Occasions---435
CHAPTER 23 Project 4: In the News---453
CHAPTER 24 Project 5: A Virtual Tea Party--469
CHAPTER 25 Project 6: Remote Editing with CGI--489
CHAPTER 26 Project 7: Your Own Bulletin Board---499
CHAPTER 27 Project 8: File Sharing with XML-RPC---517
CHAPTER 28 Project 9: File Sharing II—Now with GUI!--537
CHAPTER 29 Project 10: Do-It-Yourself Arcade Game--547
APPENDIX A The Short Version--569
APPENDIX B Python Reference--579
APPENDIX C Online Resources---595
APPENDIX D Python 3.0---599
INDEX---607

***Beginning_Python_Using_Python_2.6_and_Python_3.1_2010_628p

Introduction---xxvii
Part I: Dipping Your Toe into Python
Chapter 1: Programming Basics and Strings---3
Chapter 2: Numbers and Operators--15
Chapter 3: Variables — Names for Values--31
Part II: Python Language and the Standard Library
Chapter 4: Making Decisions--51
Chapter 5: Functions---71
Chapter 6: Classes and Objects---93
Chapter 7: Organizing Programs--111
Chapter 8: Files and Directories---127
Chapter 9: Other Features of the Language--143
Chapter 10: Building a Module---157
Chapter 11: Text Processing---189
Part III: Putting Python to Work
Chapter 12: Testing---207
Chapter 13: Writing a GUI with Python---227
Chapter 14: Accessing Databases--239
Chapter 15: Using Python for XML---265
Chapter 16: Network Programming---287
Chapter 17: Extension Programming with C---337
Chapter 18: Numerical Programming--367
Chapter 19: An Introduction to Django---387
Chapter 20: Web Applications and Web Services---407
Chapter 21: Integrating Java with Python--481
Part IV: Appendices
Appendix A: Answers to the Exercises--515
Appendix B: Online Resources--549
Appendix C: What's New in Python 3 1---553
Appendix D: Glossary--559
Index--569

***Beginning_Python_Visualization_Crafting_Visual_Transformation_Scripts_2009_378p

About the Author.---xv
About the Technical Reviewer---xvi
Acknowledgments---xvii
Introduction--xviii
CHAPTER 1 Navigating the World of Data Visualization--1
CHAPTER 2 The Environment--31
CHAPTER 3 Python for Programmers.--53
CHAPTER 4 Data Organization---101
CHAPTER 5 Processing Text Files--135
CHAPTER 6 Graphs and Plots--183
CHAPTER 7 Math Games--221
CHAPTER 8 Science and Visualization.---249
CHAPTER 9 Image Processing---285
CHAPTER 10 Advanced File Processing---319
APPENDIX Additional Source Listing--343
INDEX---349

***Bioinformatics_Programming_Using_Python_First_Edition_2009_524p

Preface--xi
1. Primitives--1
2. Names, Functions, and Modules--21
3. Collections---47
4. Control Statements---99
5. Classes---165
6. Utilities---209
7. Pattern Matching---257
8. Structured Text---287
9. Web Programming---325
10. Relational Databases---359
11. Structured Graphics---399
A. Python Language Summary---449
B. Collection Type Summary---459
Index---473

***BuildingSkillsinPython_2010_574p

I Front Matter---3
1 Preface---5
1.1 Why Read This Book?--5
1.2 Audience---6
1.3 Organization of This Book---7
1.4 Limitations--8
1.5 Programming Style--9
1.6 Conventions Used in This Book--9
1.7 Acknowledgements---10
II Language Basics--11

2 Background and History--15
2.1 History--15
2.2 Features of Python---15
2.3 Comparisons--16
2.4 Some Jargon---19
3 Python Installation--21
3.1 Windows Installation---21
3.2 Macintosh Installation---24
3.3 GNU/Linux and UNIX Overview---25
3.4 “Build from Scratch” Installation---28
4 Getting Started---31
4.1 Command-Line Interaction--31
4.2 The IDLE Development Environment---34
4.3 Script Mode---36
4.4 Getting Help---40
4.5 Syntax Formalities---41
4.6 Exercises---42
4.7 Other Tools---44
4.8 Style Notes: Wise Choice of File Names--45
5 Simple Numeric Expressions and Output--47
5.1 Seeing Output with the print() Function (or print Statement)----------------------------------47
5.2 Numeric Types and Operators--50
5.3 Numeric Conversion (or “Factory”) Functions---53
5.4 Built-In Math Functions---54
5.5 Expression Exercises---56
5.6 Expression Style Notes---60
6 Advanced Expressions---61
6.1 Using Modules--61
6.2 The math Module---61
6.3 The random Module--63
6.4 Advanced Expression Exercises--64
6.5 Bit Manipulation Operators---66
6.6 Division Operators---68
7 Variables, Assignment and Input---71
7.1 Variables---71
7.2 The Assignment Statement---73
7.3 Input Functions---75
7.4 Multiple Assignment Statement---78
7.5 The del Statement--78
7.6 Interactive Mode Revisited---79
7.7 Variables, Assignment and Input Function Exercises--80
7.8 Variables and Assignment Style Notes---81
8 Truth, Comparison and Conditional Processing---83
8.1 Truth and Logic---83
8.2 Comparisons--85
8.3 Conditional Processing: the if Statement--88
8.4 The pass Statement--90
8.5 The assert Statement--91
8.6 The if-else Operator--92
8.7 Condition Exercises---93
8.8 Condition Style Notes---94
9 Loops and Iterative Processing---95

9.1 Iterative Processing: For All and There Exists--95
9.2 Iterative Processing: The for Statement--96
9.3 Iterative Processing: The while Statement---97
9.4 More Iteration Control: break and continue--98
9.5 Iteration Exercises---100
9.6 Condition and Loops Style Notes--103
9.7 A Digression--104
10 Functions--107
10.1 Semantics--107
10.2 Function Definition: The def and return Statements---109
10.3 Function Use---110
10.4 Function Varieties--111
10.5 Some Examples---112
10.6 Hacking Mode---113
10.7 More Function Definition Features---115
10.8 Function Exercises---118
10.9 Object Method Functions---121
10.10 Functions Style Notes---122
11 Additional Notes On Functions---125
11.1 Functions and Namespaces--125
11.2 The global Statement--127
11.3 Call By Value and Call By Reference--127
11.4 Function Objects---129
III Data Structures---131
12 Sequences: Strings, Tuples and Lists--135
12.1 Sequence Semantics---135
12.2 Overview of Sequences---136
12.3 Exercises---139
12.4 Style Notes---139
13 Strings---141
13.1 String Semantics---141
13.2 String Literal Values--141
13.3 String Operations---143
13.4 String Comparison Operations---146
13.5 String Statements---146
13.6 String Built-in Functions--147
13.7 String Methods--148
13.8 String Modules--151
13.9 String Exercises---152
13.10 Digression on Immutability of Strings---153
14 Tuples---155
14.1 Tuple Semantics--155
14.2 Tuple Literal Values--155
14.3 Tuple Operations--156
14.4 Tuple Comparison Operations---157
14.5 Tuple Statements--157
14.6 Tuple Built-in Functions--158
14.7 Tuple Exercises---160
14.8 Digression on The Sigma Operator---161
15 Lists---163
15.1 List Semantics--163
15.2 List Literal Values---163

15.3 List Operations--164
15.4 List Comparison Operations---164
15.5 List Statements--165
15.6 List Built-in Functions---166
15.7 List Methods--167
15.8 Using Lists as Function Parameter Defaults---169
15.9 List Exercises--170
16 Mappings and Dictionaries--175
16.1 Dictionary Semantics---175
16.2 Dictionary Literal Values---176
16.3 Dictionary Operations---176
16.4 Dictionary Comparison Operations--178
16.5 Dictionary Statements---178
16.6 Dictionary Built-in Functions--179
16.7 Dictionary Methods--180
16.8 Using Dictionaries as Function Parameter Defaults-------------------------------------181
16.9 Dictionary Exercises---182
16.10 Advanced Parameter Handling For Functions---184
17 Sets--187
17.1 Set Semantics---187
17.2 Set Literal Values--187
17.3 Set Operations--188
17.4 Set Comparison Operators---190
17.5 Set Statements--191
17.6 Set Built-in Functions---191
17.7 Set Methods---192
17.8 Using Sets as Function Parameter Defaults---194
17.9 Set Exercises--195
18 Exceptions---199
18.1 Exception Semantics--199
18.2 Basic Exception Handling--200
18.3 Raising Exceptions---203
18.4 An Exceptional Example---204
18.5 Complete Exception Handling and The finally Clause----------------------------------206
18.6 Exception Functions---206
18.7 Exception Attributes---207
18.8 Built-in Exceptions---208
18.9 Exception Exercises--210
18.10 Style Notes---211
18.11 A Digression--212
19 Iterators and Generators--213
19.1 Iterator Semantics--213
19.2 Generator Function Semantics--214
19.3 Defining a Generator Function--215
19.4 Generator Functions---216
19.5 Generator Statements--217
19.6 Iterators Everywhere---217
19.7 Generator Function Example--218
19.8 Generator Exercises--219
20 Files--221
20.1 File Semantics---221
20.2 File Organization and Structure---222

20.3 Additional Background---223
20.4 Built-in Functions---224
20.5 File Statements---226
20.6 File Methods--226
20.7 Several Examples---228
20.8 File Exercises---232
21 Functional Programming with Collections--235
21.1 Lists of Tuples--235
21.2 List Comprehensions--236
21.3 Sequence Processing Functions: map(), filter() and reduce()--------------------------------239
21.4 Advanced List Sorting---242
21.5 The Lambda---244
21.6 Multi-Dimensional Arrays or Matrices--246
21.7 Exercises--248
22 Advanced Mapping Techniques---251
22.1 Default Dictionaries---251
22.2 Inverting a Dictionary---252
22.3 Exercises--253
IV Data + Processing = Objects--255
23 Classes---259
23.1 Semantics--259
23.2 Class Definition: the class Statement---262
23.3 Creating and Using Objects--263
23.4 Special Method Names---264
23.5 Some Examples--266
23.6 Object Collaboration--269
23.7 Class Definition Exercises--271
24 Advanced Class Definition--287
24.1 Inheritance--287
24.2 Polymorphism--292
24.3 Built-in Functions--294
24.4 Collaborating with max(), min() and sort()---296
24.5 Initializer Techniques--296
24.6 Class Variables--297
24.7 Static Methods and Class Method---299
24.8 Design Approaches---299
24.9 Advanced Class Definition Exercises---301
24.10 Style Notes---303
25 Some Design Patterns--307
25.1 Factory---307
25.2 State--310
25.3 Strategy--313
25.4 Design Pattern Exercises--315
26 Creating or Extending Data Types---319
26.1 Semantics of Special Methods---320
26.2 Basic Special Methods--321
26.3 Special Attribute Names--322
26.4 Numeric Type Special Methods---322
26.5 Collection Special Method Names--327
26.6 Collection Special Method Names for Iterators and Iterable---------------------------------329
26.7 Collection Special Method Names for Sequences--330
26.8 Collection Special Method Names for Sets--331

26.9 Collection Special Method Names for Mappings--332
26.10 Mapping Example---333
26.11 Iterator Examples---334
26.12 Extending Built-In Classes---336
26.13 Special Method Name Exercises---336
27 Attributes, Properties and Descriptors---343
27.1 Semantics of Attributes--343
27.2 Properties--344
27.3 Descriptors---346
27.4 Attribute Handling Special Method Names--348
27.5 Attribute Access Exercises---349
28 Decorators---351
28.1 Semantics of Decorators--351
28.2 Built-in Decorators---352
28.3 Defining Decorators--354
28.4 Defining Complex Decorators---355
28.5 Decorator Exercises--356
29 Managing Contexts: the with Statement---357
29.1 Semantics of a Context--357
29.2 Using a Context--358
29.3 Defining a Context Manager Function---358
29.4 Defining a Context Manager Class--360
29.5 Context Manager Exercises--361
V Components, Modules and Packages--363
30 Modules---367
30.1 Module Semantics---367
30.2 Module Definition--368
30.3 Module Use: The import Statement--370
30.4 Finding Modules: The Path---372
30.5 Variations on An import Theme--373
30.6 The exec Statement---375
30.7 Module Exercises---375
30.8 Style Notes---377
31 Packages--379
31.1 Package Semantics--379
31.2 Package Definition---380
31.3 Package Use--381
31.4 Package Exercises--381
31.5 Style Notes---381
32 The Python Library---383
32.1 Overview of the Python Library---383
32.2 Most Useful Library Sections--385
32.3 Library Exercises--393
33 Complex Strings: the re Module---395
33.1 Semantics--395
33.2 Creating a Regular Expression---396
33.3 Using a Regular Expression---397
33.4 Regular Expression Exercises--399
34 Dates and Times: the time and datetime Modules---401
34.1 Semantics: What is Time?---401
34.2 Some Class Definitions---403
34.3 Creating a Date-Time---404

34.4 Date-Time Calculations and Manipulations--405
34.5 Presenting a Date-Time---407
34.6 Formatting Symbols--408
34.7 Time Exercises---409
34.8 Additional time Module Features--410
35 File Handling Modules--411
35.1 The os.path Module---413
35.2 The os Module--414
35.3 The fileinput Module---416
35.4 The glob and fnmatch Modules---417
35.5 The tempfile Module--418
35.6 The shutil Module--419
35.7 The File Archive Modules: tarfile and zipfile---419
35.8 The sys Module---423
35.9 Additional File-Processing Modules--424
35.10 File Module Exercises--425
36 File Formats: CSV, Tab, XML, Logs and Others--427
36.1 Overview--427
36.2 Comma-Separated Values: The csv Module---428
36.3 Tab Files: Nothing Special---431
36.4 Property Files and Configuration (orINI) Files: The ConfigParser Module----------------432
36.5 Fixed Format Files, A COBOL Legacy: The codecs Module---434
36.6 XML Files: The xml.etree and xml.sax Modules--436
36.7 Log Files: The logging Module--441
36.8 File Format Exercises--446
36.9 The DOM Class Hierarchy---446
37 Programs: Standing Alone--451
37.1 Kinds of Programs--451
37.2 Command-Line Programs: Servers and Batch Processing---453
37.3 The optparse Module--455
37.4 Command-Line Examples---458
37.5 Other Command-Line Features---459
37.6 Command-Line Exercises--461
37.7 The getopt Module--461
38 Architecture: Clients, Servers, the Internet and the World Wide Web---------------------------465
38.1 About TCP/IP--465
38.2 The World Wide Web and the HTTP protocol---466
38.3 Writing Web Clients: The urllib2 Module---467
38.4 Writing Web Applications--469
38.5 Sessions and State---477
38.6 Handling Form Inputs---478
38.7 Web Services--480
38.8 Client-Server Exercises---485
38.9 Socket Programming---491
VI Projects---499
39 Areas of the Flag---503
39.1 Basic Red, White and Blue---503
39.2 The Stars--504
40 Bowling Scores---507
41 Musical Pitches---509
41.1 Equal Temperament---510
41.2 Overtones--511

41.3 Circle of Fifths--511
41.4 Pythagorean Tuning---512
41.5 Five-Tone Tuning---513
42 What Can be Computed?--515
42.1 Background---515
42.2 The Turing Machine--517
42.3 Example Machine---518
42.4 Turing Machine Implementation--519
42.5 Exercise 1---521
42.6 Test Machines--521
42.7 Exercise 2---522
42.8 Better Implementations--523
42.9 Exercise 3---524
42.10 Consequences--525
42.11 Other Applications---525
42.12 Alternative Specifications--526
42.13 Exercise 4---528
43 Mah Jongg Hands---529
43.1 Tile Class Hierarchy--529
43.2 Wall Class---531
43.3 TileSet Class Hierarchy--532
43.4 Hand Class--534
43.5 Some Test Cases--535
43.6 Hand Scoring - Points--537
43.7 Hand Scoring - Doubles--539
43.8 Limit Hands---542
44 Chess Game Notation---545
44.1 Algebraic Notation---545
44.2 Algorithms for Resolving Moves--549
44.3 Descriptive Notation--552
44.4 Game State---552
44.5 PGN Processing Specifications--553
VII Back Matter---555
45 Bibliography---557
45.1 Use Cases---557
45.2 Computer Science---557
45.3 Design Patterns--557
45.4 Languages---557
45.5 Problem Domains---557
46 Indices and Tables---559
47 Production Notes--561
Bibliography---563

***byte_of_python_2013_129p

1 A Byte of Python--8
1.1 Who Reads A Byte of Python?---8
1.2 Academic Courses---11
1.3 License---12
1.4 Read Now--12
1.5 Buy the Book---12
1.6 Download---12

1.7 Read the book in your native language--13
2 Preface---13
2.1 Who This Book Is For---13
2.2 History Lesson--13
2.3 Status Of The Book--14
2.4 Official Website---14
2.5 Something To Think About---14
3 Introduction--15
3.1 Features of Python---15
3.1.1 Simple---15
3.1.2 Easy to Learn---15
3.1.3 Free and Open Source--15
3.1.4 High-level Language---16
3.1.5 Portable---16
3.1.6 Interpreted--16
3.1.7 Object Oriented--16
3.1.8 Extensible---17
3.1.9 Embeddable--17
3.1.10 Extensive Libraries--17
3.2 Python 2 versus 3--17
3.3 What Programmers Say---18
4 Installation--18
4.1 Installation on Windows--18
4.1.1 DOS Prompt--18
4.1.2 Running Python prompt on Windows---20
4.2 Installation on Mac OS X---20
4.3 Installation on Linux---20
4.4 Summary---21
5 First Steps--21
5.1 Using The Interpreter Prompt--21
5.2 Choosing An Editor--22
5.3 Using A Source File--23
5.3.1 Executable Python Programs---25
5.4 Getting Help---27
5.5 Summary---27
6 Basics--27
6.1 Comments---27
6.2 Literal Constants--28
6.3 Numbers--28
6.4 Strings---29
6.4.1 Single Quote--29
6.4.2 Double Quotes--29
6.4.3 Triple Quotes--29
6.4.4 Strings Are Immutable--29
6.4.5 The format method--30
6.5 Variable---31
6.6 Identifier Naming---31
6.7 Data Types---32
6.8 Object--32
6.9 How to write Python programs---32
6.10 Example: Using Variables And Literal Constants--32
6.10.1 Logical And Physical Line---33

6.10.2 Indentation--35
6.11 Summary--35
7 Operators and Expressions--36
7.1 Operators---36
7.1.1 Shortcut for math operation and assignment---38
7.2 Evaluation Order--39
7.3 Changing the Order Of Evaluation--40
7.4 Associativity--40
7.5 Expressions--40
7.6 Summary--41
8 Control Flow--41
8.1 The if statement--41
8.2 The while Statement--44
8.3 The for loop--45
8.4 The break Statement--46
8.4.1 Swaroop’s Poetic Python---47
8.5 The continue Statement--47
8.6 Summary--48
9 Functions--48
9.1 Function Parameters--49
9.2 Local Variables---50
9.3 Using The global Statement--51
9.4 Default Argument Values---52
9.5 Keyword Arguments---53
9.6 VarArgs parameters---54
9.7 Keyword-only Parameters---55
9.8 The return Statement---56
9.9 DocStrings--57
9.10 Summary---58
10 Modules---58
10.1 Byte-compiledpyc files---60
10.2 The from import statement--60
10.3 A module’s name--61
10.4 Making Your Own Modules---61
10.5 The dir function--63
10.6 Packages---64
10.7 Summary---65
11 Data Structures---65
11.1 List--65
11.1.1 Quick Introduction To Objects And Classes--66
11.2 Tuple--68
11.3 Dictionary---69
11.4 Sequence---71
11.5 Set---73
11.6 References--74
11.7 More About Strings--75
11.8 Summary--76
12 Problem Solving---76
12.1 The Problem--77
12.2 The Solution--77
12.3 Second Version---80
12.4 Third Version---81

12.5 Fourth Version--83
12.6 More Refinements---84
12.7 The Software Development Process--85
12.8 Summary---85
13 Object Oriented Programming--85
13.1 The self---86
13.2 Classes---87
13.3 Object Methods---88
13.4 The init method---88
13.5 Class And Object Variables--89
13.6 Inheritance---92
13.7 Summary--95
14 Input Output---95
14.1 Input from user---95
14.2 Files--97
14.3 Pickle--98
14.4 Summary--99
15 Exceptions--99
15.1 Errors--99
15.2 Exceptions--100
15.3 Handling Exceptions---100
15.4 Raising Exceptions---101
15.5 Try. Finally--102
15.6 The with statement---103
15.7 Summary--104
16 Standard Library---104
16.1 sys module---105
16.2 logging module--106
16.3 Module of the Week Series---107
16.4 Summary--107
17 More---107
17.1 Passing tuples around--108
17.2 Special Methods---108
17.3 Single Statement Blocks---109
17.4 Lambda Forms---109
17.5 List Comprehension--110
17.6 Receiving Tuples and Dictionaries in Functions-------------------------------------110
17.7 The assert statement--111
17.8 Escape Sequences---111
17.8.1 Raw String---112
17.9 Summary--112
18 What Next---113
18.1 Example Code--113
18.2 Questions and Answers--114
18.3 Tutorials---114
18.4 Videos--114
18.5 Discussion--114
18.6 News--114
18.7 Installing libraries---114
18.8 Graphical Software--115
18.8.1 Summary of GUI Tools---115
18.9 Various Implementations--116

18.10Functional Programming (for advanced readers)--116
18.11Summary---117
19 FLOSS---117
20 Colophon--118
20.1 Birth of the Book---119
20.2 Teenage Years---119
20.3 Now---119
20.4 About The Author--119
21 Revision History--119
22 Translations--122
22.1 Arabic--122
22.2 Brazilian Portuguese--122
22.3 Catalan---122
22.4 Chinese---123
22.5 Chinese Traditional---123
22.6 French--124
22.7 German--124
22.8 Greek---125
22.9 Indonesian--125
22.10 Italian---126
22.11 Japanese---126
22.12 Mongolian--126
22.13 Norwegian (bokmål)---126
22.14 Polish---127
22.15 Portuguese---127
22.16 Romanian---127
22.17 Russian and Ukranian---128
22.18 Slovak---128
22.19 Spanish--128
22.20 Swedish--129
22.21 Turkish--129

***CherryPy_Essentials_Rapid_Python_Web_Application_Development_2007_270p

Chapter 1: Introduction to CherryPy--7
Overview--7
History of CherryPy---8
The Community---9
CherryPy Project Strengths---10
Beyond CherryPy---11
Through the Book---11
Summary---12
Chapter 2: Download and Install CherryPy---13
Requirements--13
Overview--14
Installation from a Tarball---16
Installation through Easy Install---18
Installation from Subversion---20
Testing your Installation---23
Keeping CherryPy Up to Date---23

Summary---24
Chapter 3: Overview of CherryPy---25
Vocabulary--25
Basic Example--26
Built-In HTTP Server--32
Internal Engine--32
Configuration---33
Object Publisher Engine---36
Library--38
The Autoreload Feature--39
The Caching Module---39
The Coverage Module--39
The Encoding/Decoding Module---40
The HTTP Module--40
The Httpauth Module---40
The Profiler Module---40
The Sessions Module--41
The Static Module---42
The Tidy Module---42
The Wsgiapp Module---42
The XML-RPC Module--42
Tools--43
Error and Exception Handling--44
Summary---49
Chapter 4: CherryPy in Depth---51
HTTP Compliance---51
Multiple HTTP Servers---52
Multi-Threaded Application Server--54
URI Dispatching--55
HTTP Method Dispatcher--55
Routes Dispatcher---57
Virtual Host Dispatcher---58
Hook into CherryPy's Core Engine--59
CherryPy Toolbox--61
Basic Authentication Tool--62
Caching Tool--63
Decoding Tool--64
Digest Authentication Tool---65
Encode Tool---66
Error Redirect Tool---67
Etag Tool---67
Gzip Tool---69
Ignore Headers Tool---69
Log Headers Tool---70
Log Tracebacks Tool---71
Proxy Tool--72
Referer Tool--73
Response Headers Tool---74
Trailing Slash Tool--75
XML-RPC Tool--76
Toolbox--77
Creating a Tool---77

Static Resource Serving--81
Using the Staticfile Tool to Serve a Single File--81
Using the Staticdir Tool to Serve a Complete Directory--83
Bypassing Static Tools to Serve Static Content--85
WSGI Support---86
Hosting a WSGI Application within the CherryPy WSGI Server-----------------------------------87
Hosting a CherryPy WSGI Application within---a
Third-Party WSGI Server---89
Summary---90
Chapter 5: A Photoblog Application--91
A Photoblog Application--91
Photoblog Entities--92
Vocabulary---94
DBMSes Overview--95
Relational Database Management System (RDBMS)---95
Object-Oriented Database Management System (OODBMS)--------------------------------------96
XML Database Management System (XMLDBMS)---97
Object-Relational Mapping---97
Python Object-Relational Mappers---98
Photoblog Application Entity Modeling--108
Mapping Entities---109
Units and UnitProperties--111
Associating Units---112
The Sandbox Interface---112
Querying Units---113
Extending the Data Access Layer--114
Summary---117
Chapter 6: Web Services--119
Traditional Web Development--119
Separation of Concerns---121
REST---122
Uniform Resource Identifier--123
HTTP Methods--124
Putting it Together---128
REST Interface through CherryPy--130
Atom Publishing Protocol---131
Atom XML-Document Format--132
APP Implementation---134
Summary---136
Chapter 7: The Presentation Layer---137
HTML---137
XML---138
XHTML--138
CSS--139
DHTML--141
Templating--142
Kid—The Templating Engine---142
Overview--142
Kid's Attributes---144
XML-Based Templating Language---144
Variable Substitution--144
Conditional Statement---144

Looping Mechanism--145
Extensibility---146
Other Attributes---147
Photoblog Design Preparation--147
Targetting the User Agent---147
Tools--148
Global Design Goals---148
Design Directory Layout---149
CherryPy—Encapsulating the Template Rendering Process----------------------------------149
Photoblog Design in Detail---151
Basic Structure--151
Mochikit--156
Developing the Photoblog Design---157
HTML Code---157
Adding a Link---158
Handling the End-User Actions--158
Amending the Template--159
Amending the CSS--159
Let's be More Flexible...---160
Summary--161
Chapter 8: Ajax--163
Rise of the Rich-Client Applications--163
Ajax--164
Ajax—Advantages and Drawbacks---165
Behind the Scene: XMLHttpRequest---166
Performing a GET Request---167
Performing a Content-Negotiated GET Request ---168
Performing a POST Request---169
Performing PUT, HEAD, or DELETE Requests---170
Cookies---170
Authentication using Digest or Basic Schemes---170
JSON--176
Applying Ajax to our Application---178
Defining the Required Namespaces--178
Implementing Namespaces--179
Adding Methods to the Classes--179
Method to Create a New Album--183
Method to Update an Existing Album--190
Method to Delete an Existing Album---190
Summary--191
Chapter 9: Testing---193
Why Testing---193
Planning a Test--194
Common Testing Approach--195
Unit Testing--195
unittest--196
doctest--201
Unit Testing Web Applications--205
Performance and Load Testing--213
Functional Testing---218
Application under Test---219
Selenium Core--222

Selenium IDE--227
Selenium Remote Control--231
Summary---233
Chapter 10: Deployment---235
Configuration--235
CherryPy—Web and Engine Configuration System---235
Photoblog Application Configuration System--238
Deployment---240
Apache with mod_rewrite Module---241
Lighttpd with mod_proxy Module--243
Apache with mod_python Module---244
mod_python with WSGI Application--246
SSL---246
Creating a Certificate and a Private Key--247
Using the CherryPy SSL Support--248
Using the lighttpd SSL Support--250
Using the Apache mod_ssl Support--251
Summary---251
Index---253

***Computer_Games_with_PYTHON_2010_436p

Source Code Listing

hello.py---21
guess.py--30
jokes.py---51
dragon.py---58
buggy.py--83
coinFlips.py---87
hangman.py--103
tictactoe.py---150
truefalsefizz.py--172
bagels.py---184
sonar.py--213
cipher.py---244
reversi.py---261
aisim1.py---292
aisim2.py---294
aisim3.py---299
pygameHelloWorld.py--309
animation.py--324
collisionDetection.py--338
pygameInput.py--348
spritesAndSounds.py---360
dodger.py---371

1 Installing Python---1
Downloading and Installing Python--2
Starting Python---4
How to Use This Book---4
The Featured Programs--5
Line Numbers and Spaces--5

Summary--7

2 The Interactive Shell--8
Some Simple Math Stuff---8
Evaluating Expressions---11
Storing Values in Variables---12
Using More Than One Variable---15
Summary--16

3 Strings, and Your First Program--18
Strings---18
String Concatenation---19
Writing Programs in IDLE's File Editor---20
Hello World!---20
How the "Hello World" Program Works---23
Summary--26

4 Guess the Number---28
The "Guess the Number" Game--28
Sample Run of "Guess the Number"---29
Guess the Number's Source Code--29
The import Statement---31
The random.randint() Function---32
Passing Arguments to Functions---34
Blocks--36
The Boolean Data Type---37
Comparison Operators--37
Conditions---38
Experiment with Booleans, Comparison Operators, and Conditions-------------------------------38
Looping with While Statements--41
The Player Guesses---41
if Statements--44
Leaving Loops Early with the break Statement---45
Check if the Player Won---46
Summary: What Exactly is Programming?--47
A Web Page for Program Tracing---48

5 Jokes---50
Make the Most of print()---50
Sample Run of Jokes--50
Joke's Source Code---51
Escape Characters--52
Quotes and Double Quotes--53
The end Keyword Argument--54
Summary--55

6 Dragon Realm---56
Introducing Functions---56
Sample Run of Dragon Realm---57
Dragon Realm's Source Code---57
def Statements--60
Boolean Operators--61

Return Values--65
Variable Scope---65
Parameters---68
Where to Put Function Definitions--70
Displaying the Game Results--71
The Colon :--73
Where the Program Really Begins---73
Designing the Program--75
Summary--76

7 Using the Debugger--77
Bugs!---77
Starting the Debugger---78
Stepping--80
The Go and Quit Buttons--81
Stepping Over and Stepping Out--81
Find the Bug--83
Break Points--86
Summary---88

8 Flow Charts---89
How to Play "Hangman"--89
Sample Run of "Hangman"--89
ASCII Art---91
Designing a Program with a Flowchart--92
Creating the Flow Chart---93
Summary: The Importance of Planning Out the Game---100

9 Hangman---102
Hangman's Source Code--103
Multi-line Strings--107
Constant Variables---108
Lists--108
Changing the Values of List Items with Index Assignment--110
List Concatenation---110
The in Operator--111
Removing Items from Lists with del Statements---112
Lists of Lists--113
Methods--114
The reverse() and append() List Methods---115
The Difference Between Methods and Functions--116
The split() Function--116
The range() and list() Functions--120
for Loops--121
elif ("Else If") Statements---127
Review of the Functions We Defined--131
Making New Changes to the Hangman Program---132
Dictionaries---139
Sets of Words for Hangman---142
The random.choice() Function---143
Multiple Assignment---145
Summary--147

10 Tic Tac Toe--148
Sample Run of Tic Tac Toe--149
Source Code of Tic Tac Toe---150
Designing the Program--154
Game AI--156
List References--162
Short-Circuit Evaluation--170
The None Value--175
Summary: Creating Game-Playing Artificial Intelligences---182

11 Bagels---183
Sample Run--184
Bagel's Source Code---184
Designing the Program--186
The random.shuffle() Function---188
Augmented Assignment Operators---190
The sort() List Method---192
The join() String Method---192
String Interpolation--194
Summary: Getting Good at Bagels---198

12 Cartesian Coordinates--200
Grids and Cartesian Coordinates---201
Negative Numbers---202
Math Tricks--204
Absolute Values and the abs() Function--206
Coordinate System of a Computer Monitor--207
Summary: Using this Math in Games--208

13 Sonar Treasure Hunt--209
Sample Run--210
Sonar's Source Code---213
Designing the Program--218
The remove() List Method--229
Summary: Review of our Sonar Game---238

14 Caesar Cipher---239
About Cryptography--239
The Caesar Cipher--240
ASCII, and Using Numbers for Letters--241
The chr() and ord() Functions---242
Sample Run of Caesar Cipher---243
Caesar Cipher's Source Code--244
The isalpha() String Method---247
The isupper() and islower() String Methods--248
Brute Force---251
Summary: Reviewing Our Caesar Cipher Program---253

15 Reversi---256
How to Play Reversi---255
Sample Run--257

Reversi's Source Code--260
The bool() Function--276
Summary: Reviewing the Reversi Game---290

16 AI Simulation---291
"Computer vs. Computer" Games---291
AISim1.py Source Code---292
AISim2.py Source Code---294
Percentages---296
The round() Function---297
Comparing Different AI Algorithms---299
AISim3.py Source Code---299
Learning New Things by Running Simulation Experiments--305

17 Graphics and Animation--306
Installing Pygame---307
Hello World in Pygame--308
Hello World's Source Code--308
Importing the Pygame Module--311
Variables Store References to Objects---313
Colors in Pygame--313
Fonts, and the pygame.font.SysFont() Function--315
Attributes---316
Constructor Functions and the type() function.---317
The pygame.PixelArray Data Type---321
Events and the Game Loop---322
Animation--324
The Animation Program's Source Code---324
Some Small Modifications---335
Summary: Pygame Programming---335

18 Collision Detection and Input---337
The Collision Detection Program's Source Code---337
The Collision Detection Function---341
The pygame.time.Clock Object and tick() Method---344
The Keyboard Input Program's Source Code--348
The colliderect() Method--356
Summary: Collision Detection and Pygame Input---356

19 Sound and Images--358
Image and Sound Files---360
Sprites and Sounds Program---360
The Sprites and Sounds Program's Source Code---360
Setting Up the Window and the Data Structure---364
The pygame.transform.scale() Function--364
Summary: Games with Graphics and Sounds--368

20 Dodger--369
Review of the Basic Pygame Data Types--370
Dodger's Source Code--371
Implementing the Cheat Codes---392
Modifying the Dodger Game---397

Summary: Creating Your Own Games--397
Appendix A
Differences Between Python 2 and 3---399
Appendix B
Statements, Functions, and Methods Reference---403
Appendix C
Running Python Programs without Python Installed--404
Appendix D
Common Error Messages in Python--407
Glossary---411
About the Author--421

***Dive_Into_Python_5_4_2004_327p

Chapter 1. Installing Python---2
1.1. Which Python is right for you?---2
1.2. Python on Windows--2
1.3. Python on Mac OS X---3
1.4. Python on Mac OS 95
1.5. Python on RedHat Linux--5
1.6. Python on Debian GNU/Linux---6
1.7. Python Installation from Source--6
1.8. The Interactive Shell--7
1.9. Summary---8

Chapter 2. Your First Python Program---9
2.1. Diving in---9
2.2. Declaring Functions--9
2.3. Documenting Functions--10
2.4. Everything Is an Object---11
2.5. Indenting Code--13
2.6. Testing Modules---14

Chapter 3. Native Datatypes---15
3.1. Introducing Dictionaries--15
3.2. Introducing Lists---17
3.3. Introducing Tuples---22
3.4. Declaring variables--23
3.5. Formatting Strings---25
3.6. Mapping Lists--26
3.7. Joining Lists and Splitting Strings---28
3.8. Summary--29

Chapter 4. The Power Of Introspection---31
4.1. Diving In--31
4.2. Using Optional and Named Arguments--32
4.3. Using type, str, dir, and Other Built−In Functions---33
4.4. Getting Object References With getattr--36
4.5. Filtering Lists---38
4.6. The Peculiar Nature of and and or--39
4.7. Using lambda Functions--41
4.8. Putting It All Together---43

4.9. Summary---45

Chapter 5. Objects and Object−Orientation---47
5.1. Diving In---47
5.2. Importing Modules Using from module import---49
5.3. Defining Classes---50
5.4. Instantiating Classes---53
5.5. Exploring UserDict: A Wrapper Class--54
5.6. Special Class Methods---56
5.7. Advanced Special Class Methods---59
5.8. Introducing Class Attributes---60
5.9. Private Functions--62
5.10. Summary---63

Chapter 6. Exceptions and File Handling--64
6.1. Handling Exceptions---64
6.2. Working with File Objects--66
6.3. Iterating with for Loops---70
6.4. Using sys.modules--72
6.5. Working with Directories---74
6.6. Putting It All Together---77
6.7. Summary---78

Chapter 7. Regular Expressions--81
7.1. Diving In---81
7.2. Case Study: Street Addresses---81
7.3. Case Study: Roman Numerals---83
7.4. Using the {n,m} Syntax--85
7.5. Verbose Regular Expressions---88
7.6. Case study: Parsing Phone Numbers--89
7.7. Summary---93

Chapter 8. HTML Processing---94
8.1. Diving in---94
8.2. Introducing sgmllib.py---98
8.3. Extracting data from HTML documents--100
8.4. Introducing BaseHTMLProcessor.py--102
8.5. locals and globals--104
8.6. Dictionary−based string formatting--107
8.7. Quoting attribute values---108
8.8. Introducing dialect.py---109
8.9. Putting it all together--111
8.10. Summary---113

Chapter 9. XML Processing---115
9.1. Diving in---115
9.2. Packages---121
9.3. Parsing XML--123
9.4. Unicode--125
9.5. Searching for elements--129
9.6. Accessing element attributes--131
9.7. Segue---132

Chapter 10. Scripts and Streams--133
10.1. Abstracting input sources---133
10.2. Standard input, output, and error--136
10.3. Caching node lookups--140
10.4. Finding direct children of a node--141
10.5. Creating separate handlers by node type---141
10.6. Handling command−line arguments--143
10.7. Putting it all together--146
10.8. Summary---148

Chapter 11. HTTP Web Services---149
11.1. Diving in---149
11.2. How not to fetch data over HTTP--151
11.3. Features of HTTP---152
11.4. Debugging HTTP web services---153
11.5. Setting the User−Agent---155
11.6. Handling Last−Modified and ETag--156
11.7. Handling redirects--159
11.8. Handling compressed data---163
11.9. Putting it all together--165
11.10. Summary--167

Chapter 12. SOAP Web Services---168
12.1. Diving In---168
12.2. Installing the SOAP Libraries--169
12.3. First Steps with SOAP--171
12.4. Debugging SOAP Web Services---172
12.5. Introducing WSDL--173
12.6. Introspecting SOAP Web Services with WSDL--174
12.7. Searching Google---176
12.8. Troubleshooting SOAP Web Services---179
12.9. Summary---182

Chapter 13. Unit Testing--183
13.1. Introduction to Roman numerals---183
13.2. Diving in---184
13.3. Introducing romantest.py--184
13.4. Testing for success---187
13.5. Testing for failure--189
13.6. Testing for sanity---190

Chapter 14. Test−First Programming---193
14.1. roman.py, stage 1--193
14.2. roman.py, stage 2--196
14.3. roman.py, stage 3--199
14.4. roman.py, stage 4--202
14.5. roman.py, stage 5--205

Chapter 15. Refactoring---208
15.1. Handling bugs--208
15.2. Handling changing requirements--210

15.3. Refactoring--216
15.4. Postscript---219
15.5. Summary--221

Chapter 16. Functional Programming---223
16.1. Diving in--223
16.2. Finding the path---224
16.3. Filtering lists revisited---226
16.4. Mapping lists revisited--228
16.5. Data−centric programming---229
16.6. Dynamically importing modules---230
16.7. Putting it all together--231
16.8. Summary--234

Chapter 17. Dynamic functions--235
17.1. Diving in--235
17.2. plural.py, stage 1--235
17.3. plural.py, stage 2--237
17.4. plural.py, stage 3--239
17.5. plural.py, stage 4--240
17.6. plural.py, stage 5--242
17.7. plural.py, stage 6--243
17.8. Summary--246

Chapter 18. Performance Tuning---247
18.1. Diving in--247
18.2. Using the timeit Module--249
18.3. Optimizing Regular Expressions---250
18.4. Optimizing Dictionary Lookups---253
18.5. Optimizing List Operations---256
18.6. Optimizing String Manipulation---258
18.7. Summary--260

Appendix A. Further reading--261

Appendix B. A 5−minute review--268

Appendix C. Tips and tricks---282

Appendix D. List of examples--289

Appendix E. Revision history---302

Appendix F. About the book--314

Appendix G. GNU Free Documentation License--315
G.0. Preamble---315
G.1. Applicability and definitions--315
G.2. Verbatim copying---316
G.3. Copying in quantity--316
G.4. Modifications--317
G.5. Combining documents--318

G.6. Collections of documents--318
G.7. Aggregation with independent works--318

Appendix G. GNU Free Documentation License

G.8. Translation--318
G.9. Termination---319
G.10. Future revisions of this license--319
G.11. How to use this License for your documents---319

Appendix H. Python license---320
H.A. History of the software---320
H.B. Terms and conditions for accessing or otherwise using Python--------------------------320

***Expert_Python_Programming_2008_372p

Chapter 1: Getting started--9
Installing Python---10
Python Implementations--10
Jython--10
IronPython--11
PyPy--11
Other Implementations---11
Linux Installation --12
Package Installation--12
Compiling the Sources---13
Windows Installation--14
Installing Python --14
Installing MinGW ---15
Installing MSYS---16
Mac OS X Installation --17
Package Installation--17
Compiling the Source--18
The Python Prompt---18
Customizing the Interactive Prompt---19
iPython: An Advanced Prompt---20
Installing setuptools---21
Understanding How It Works---21
setuptools Installation Using EasyInstall---22
Hooking MinGW into distutils--23
Working Environment---24
Using an Editor and Complementary Tools--24
Code Editor---25
Installing and Configuring Vim ---25
Using Another Editor--27
Extra Binaries--28
Using an Integrated Development Environment ---28
Installing Eclipse with PyDev---29
Summary---32

Chapter 2: Syntax Best Practices—Below the Class Level---33
List Comprehensions---34
Iterators and Generators--36

Generators---37
Coroutines---41
Generator Expressions---43
The itertools Module---44
islice: The Window Iterator---44
tee: The Back and Forth Iterator---45
groupby: The uniq Iterator--45
Other Functions---46
Decorators---47
How to Write a Decorator---48
Argument checking--50
Caching--52
Proxy--54
Context Provider--55
with and contextlib--56
The contextlib Module--58
Context Example--59
Summary---61

Chapter 3: Syntax Best Practices—Above the Class Level---63
Subclassing Built-in Types---63
Accessing Methods from Superclasses---65
Understanding Python's Method Resolution Order (MRO)--66
super Pitfalls--70
Mixing super and classic Calls--70
Heterogeneous Arguments ---72
Best Practices--73
Descriptors and Properties---74
Descriptors--74
Introspection Descriptor---77
Meta-descriptor--79
Properties---81
Slots---83
Meta-programming---84
The__new__ Method---84
__metaclass__ Method--86
Summary---89

Chapter 4: Choosing Good Names---91
PEP 8 and Naming Best Practices---91
Naming Styles---92
Variables--92
Constants---92
Public and Private Variables--95
Functions and Methods--96
The Private Controversy---97
Special Methods---98
Arguments--98
Properties---99
Classes---99
Modules and Packages--99
Naming Guide---100

Use "has" or "is" Prefix for Boolean Elements--100
Use Plural for Elements That Are Sequences--100
Use Explicit Names for Dictionaries---101
Avoid Generic Names---101
Avoid Existing Names--101
Best Practices for Arguments--102
Build Arguments by Iterative Design---102
Trust the Arguments and Your Tests--103
Use args and kw Magic Arguments Carefully---104
Class Names---106
Module and Package Names---107
Working on APIs--107
Tracking Verbosity--108
Building the Namespace Tree--108
Splitting the Code---110
Using Eggs---111
Using a Deprecation Process--112
Useful Tools--113
Pylint--113
CloneDigger---115
Summary---116

Chapter 5: Writing a Package--117
A Common Pattern for All Packages--117
setup.py, the Script That Controls Everything--118
sdist---119
build and bdist---121
bdist_egg --122
install---123
How to Uninstall a Package--123
develop --124
test--124
register and upload ---125
Creating a New Command--128
setup.py Usage Summary---129
Other Important Metadata--129
The Template-Based Approach--131
Python Paste---131
Creating Templates---133
Creating the Package Template---133
Development Cycle---138
Summary---141

Chapter 6: Writing an Application--143
Atomisator: An Introduction---143
Overall Picture--144
Working Environment--146
Adding a Test Runner--148
Adding a packages Structure ---148
Writing the Packages---149
atomisator.parser---149
Creating the Initial Package---150

Creating the Initial doctest--151
Building the Test Environment---153
Writing the Code---153
atomisator.db---154
SQLAlchemy---154
Providing the APIs--158
atomisator.feed--159
atomisator.main---160
Distributing Atomisator--162
Dependencies between Packages--164
Summary---165

Chapter 7: Working with zc.buildout---167
zc.buildout Philosophy---168
Configuration File Structure---168
Minimum Configuration File---169
[buildout] Section Options---169
The buildout Command---170
Recipes---172
Notable Recipes--174
Creating Recipes---174
Atomisator buildout Environment--175
buildout Folder Structure--176
Going Further---177
Releasing and Distributing---178
Releasing the Packages---178
Adding a Release Configuration File--179
Building and Releasing the Application--180
Summary---181

Chapter 8: Managing Code---183
Version Control Systems---183
Centralized Systems---184
Distributed Systems---186
Distributed Strategies--188
Centralized or Distributed?---188
Mercurial---189
Project Management with Mercurial---193
Setting Up a Dedicated Folder---193
Configuring hgwebdir--194
Configuring Apache--195
Setting Up Authorizations---198
Setting Up the Client Side---199
Continuous Integration--200
Buildbot--201
Installing Buildbot---202
Hooking Buildbot and Mercurial---204
Hooking Apache and Buildbot---205
Summary---206

Chapter 9: Managing Life Cycle--207
Different Approaches---207

Waterfall Development Model--207
Spiral Development Model---208
Iterative Development Model--210
Defining a Life Cycle--210
Planning--212
Development---212
Global Debug --212
Release--213
Setting Up a Tracking System--213
Trac--213
Installation--215
Apache Settings---217
Permission Settings---218
Project Life Cycle with Trac---219
Planning--219
Development---221
Cleaning--221
Release---221
Summary---222

Chapter 10: Documenting Your Project---223
The Seven Rules of Technical Writing--223
Write in Two Steps--224
Target the Readership--225
Use a Simple Style---226
Limit the Scope of the Information--227
Use Realistic Code Examples---227
Use a Light but Sufficient Approach--228
Use Templates--228
A reStructuredText Primer---229
Section Structure---230
Lists---232
Inline Markup---232
Literal Block---232
Links---233
Building the Documentation---234
Building the Portfolio---234
Design --235
Usage ---238
Operations---242
Make Your Own Portfolio---242
Building the Landscape--243
Producer's Layout--243
Consumer's Layout---244
Summary--249

Chapter 11: Test-Driven Development---251
I Don't Test---251
Test-Driven Development Principles--251
Preventing Software Regression ---253
Improving Code Quality---254
Providing the Best Developer Documentation--254

Producing Robust Code Faster---255
What Kind of Tests?---255
Acceptance Tests--255
Unit Tests---256
Python Standard Test Tools--256
I Do Test---260
Unittest Pitfalls---260
Unittest Alternatives---261
nose---262
py.test--264
Fakes and Mocks---267
Building a Fake---268
Using Mocks--271
Document-Driven Development---273
Writing a Story---273
Summary--274

Chapter 12: Optimization: General Principles and Profiling Techniques-------------------------275
The Three Rules of Optimization--275
Make It Work First--275
Work from the User's Point of View---276
Keep the Code Readable(and thus maintainable)---277
Optimization Strategy--277
Find Another Culprit---278
Scale the Hardware--278
Write a Speed Test---279
Finding Bottlenecks---280
Profiling CPU Usage--280
Macro-Profiling--280
Micro-Profiling --284
Measuring Pystones---287
Profiling Memory Usage---288
How Python Deals with Memory---288
Profiling Memory--290
Profiling Network Usage---295
Summary--296

Chapter 13: Optimization: Solutions--297
Reducing the Complexity ---298
Measuring Cyclomatic Complexity ---298
Measuring the Big-O Notation--298
Simplifying ---301
Searching in a List---301
Using a Set Instead of a List---302
Cut the External Calls, Reduce the Workload---303
Using Collections ---303
Multithreading---306
What is Multithreading?---307
How Python Deals with Threads---307
When Should Threading Be Used?--309
Building Responsive Interfaces---309
Delegating Work --309

Multi-User Applications --310
Simple Example---310
Multiprocessing--314
Pyprocessing--315
Caching---317
Deterministic Caching--318
Non-Deterministic Caching--321
Pro-Active Caching--322
Memcached--322
Summary---323

Chapter 14: Useful Design Patterns---325
Creational Patterns---325
Singleton--326
Structural Patterns---328
Adapter--329
Interfaces--331
Proxy--332
Facade---333
Behavioral Patterns--334
Observer---334
Visitor---336
Template---339
Summary---341
Index--343

***Foundations_of_Agile_Python_Development_2008_417p

About the Author--xiii
About the Technical Reviewer.--xv
Acknowledgments---xvii
Introduction--xix

CHAPTER 1 What Is Agile Development?--1
Why More Methodologies?--1
A Little History--3
Planning and Agile Development---4
What Are Agile Methods?---4
Pair Programming--5
User Stories---7
The System Metaphor--8
On-Site Customers--8
Unit Tests--9
Test-Driven Development---10
Refactoring--11
Simple Design---12
Collective Code Ownership--12
Short Iterations--13
Continuous Reflection--15
Continuous Integration---16
Documentation---17
Summary.--18

CHAPTER 2 The IDE: Eclipsing the Command Line---------------------------------21
Installing Eclipse---23
Installing Plug-Ins---25
Installing and Configuring Pydev.---31
Your First Project---32
Looking Under the Hood---38
Paying for More Functionality--39
Summary 40 ---v

CHAPTER 3 Revision Control: Subverting Your Code.--41
Revision Control Phylum.---42
What Subversion Does for You---43
Getting Subverted--44
Working with Your Subverted Code--47
Examining Files---49
Adding Files---50
Copying and Moving Files---51
Deleting Files---52
Reverting Changes.--53
Modifying a File--53
Updating Your Working Copy--54
Conflicting Changes---55
Subverting Eclipse--59
Sharing Your Subverted Project---59
Importing from Subversion--60
Working with a Subverted Eclipse--64
The Team Repository View--65
Adding a File--68
Committing Changes---70
Editing a File---71
Reverting Changes.--72
Resolving Conflicts---73
Deleting Files---76
Moving Files--77
Renaming Files---77
Copying Files--78
Reverting Moves, Renames, and Copies--79
Summary.--79

CHAPTER 4 Setuptools: Harnessing Your Code---81
The Project: A Simple RSS Reader---81
Python Modules--82
The Old Way--83
The New Way: Cooking with Eggs---84
Some Notes About Building Multiple Versions---85
Installing Setuptools--86
Getting Started with Setuptools---87
Building the Project--88
Installing Executables---91
Dependencies---92
Think Globally, Install Locally--94

Removing an Existing Package: Undoing Your Hard Work------------------------95
Installing from the Local Copy--96
Fixing Options with setup.cfg---97
Bootstrapping Setuptools---97
Subverting Subversion: What Shouldn’t Be Versioned----------------------------------98
The Easy Way with Eclipse--100
Checking in Changes: Not Losing It---100
Working in Development Mode---100
Summary.---102

CHAPTER 5 A Build for Every Check-In---103
Buildbot Architecture---104
Installing Buildbot---104
Configuring the Build System---106
Mastering Buildbot--107
Enslaving Buildbot--112
Hooking Up Source Control---116
Using the Source---119
Subversion to Buildbot, Over--121
A Python for Every Builder---122
Finally, a Real Build Succeeds---124
Installing the Build--125
Supporting Python 2.4 Builds---128
Ensuring Local Dependency Processing--132
Keeping Up Appearances---134
Summary.---136

CHAPTER 6 Testing: The Horse and the Cart--139

Unit Testing---141
The Problems with Not Unit Testing---142
Pessimism---143
Test-Driven Development--146
Knowing Your Unit Tests---147
unittest and Nose---148
A Simple RSS Reader---149
The First Tests--151
Finding Tests with Nose---159
Skipping Slow Tests--160
Integrating the Tests into the Environment---162
Running Tests After Every Change--163
Running the Complete Test Suite in Development---------------------------------------167
Buildbot with Unit Tests---171
Summary.--173

CHAPTER 7 Test-Driven Development and Impostors------------------------------------175

Moving Beyond Acceptance Tests---175
Renaming---183
Overriding Existing Methods: Monkeypatching---185
Monkeypatching and Imports---186
The Changes Go Live.---188

Using Data Files--189
Isolation--190
Rolling Your Own--192
Python Quirks--193
Mocking Libraries---193
Aggregating Two Feeds---194
A Simple pMock Example--195
Implementing with pMock--196
Test: Defining combine_feeds--196
Test: Defining add_single_feed--197
Refactoring: Extracting AggregateFeed---198
Refactoring: Moving add_single_feed--199
Test: Defining create_entry--200
Test: Ensuring That AggregateFeed Creates a FeedEntry Factory---------------200
Test: Defining add---201
Test: AggregateFeed.entries Is Always Initialized to a Set-----------------------201
Test: Defining FeedEntry.from_parsed_feed---------------------------------------202
Test: Defining feed_entry_listing--202
Test: Defining feeds_from_urls--203
Test: AggregateFeed Initializes the FeedParser Factory---------------------------203
Test: Defining from_urls--204
Refactoring: Reimplementing from_urls---------------------------------------204
Refactoring: Condensing Some Tests--206
Test: Formatting Feed Entry Listings---207
Test: Defining print_entry_listings---208
Test: FeedWriter Initializes the stdout Attribute------------------------------------209
Test: Empty AggregateFeeds Generate No Output---------------------------------209
Test: Defining is_empty--210
Test: Defining new_main---210
Test: The Application Initializes Dependencies------------------------------------211
Refactoring: Making new_main the New main2------------------------------------212
A Simple PyMock Example---212
Monkeypatching.---214
Saying the Same Thing Differently--214
Implementing with PyMock---215
Test: from_urls and Mocking External Modules-----------------------------------216
Test: Defining add_single_feed--217
Refactoring: Moving Methods to a New Object-----------------------------------218
Refactoring: Moving add_single_feed---218
Refactoring: Moving from_urls()--219
Test: create_entry() and Mocking Class Constructors.-----------------------------220
Tests: Defining add and AggregateFeed.__init__----------------------------------221
Test: Defining FeedEntry.__init__---222
Test: Defining listing---222
Test: entry_listings Should Be Sorted---223
Test: Defining print_entry_listings--224
Test: print_entry_listings Should Do Nothing with Empty Feeds----------------225
Test: is_empty and the Unproven Test--226
Test: new_main, Hooking It All Together---226
Test: RSReader Initialization--227
Finishing Up: Activating the New Functionality-----------------------------------227
Other pMock and PyMock Features---228

Raising Exceptions with pMock--228
Raising Exceptions with PyMock--228
Playback Counts with pMock---229
Playback Counts with PyMock.---229
Mocking Attribute Setters with PyMock--229
Mocking Generators with PyMock--230
Using PyMock with unittest--230
Summary---231

CHAPTER 8 Everybody Needs Feedback--233
Measuring Software Quality---235
Measurements--236
Quantitative Measurements: How Much Is That Doggie in the Window?----------237
Code Coverage--237
Complexity Measurements---239
Velocity: When Are We Done?---242
Qualitative Measurements: It’s a Shih Tzu!---243
Coding Conventions---244
Welcome Back to Python---246
Never Try to Fix a Social Problem with a Technical Solution------------------------248
Code Reviews.---249
Renaming--250
Communication---250
Technological Feedback: Bad Programmer, No Cookie-------------------------------251
Coercion at the Keyboard--251
When Code Is Submitted--256
Buildbot and Coverage---258
Summary.--261

CHAPTER 9 Databases--263
A New Religion---263
Blurring the Boundaries--264
Concealing Data Access---265
Object-Relational Mappers--265
The Active Record Pattern--266
The Data Mapper Pattern---266
The Unit of Work Pattern---266
Python ORMs---267
SQLObject---267
SQLAlchemy---283
Building the Database.--296
Testing.---297
Refactorings---298
Migrations--298
The Instructions--299
Numbering Migrations and Playing Them Back-------------------------------299
Where to Put the Migration Mechanism.--300
DBMigrate: A Migration Mechanism---300
Summary.--306

CHAPTER 10 Web Testing--309
Really Simple Primer.--309

HTML--310
CSS--311
XML--311
URI and URL---311
HTTP.--312
JavaScript---312
Web Servers and Web Applications--312
WSGI.---314
Using the write Callback---315
WSGI Middleware---316
Testing Web Applications--316
Graphics and Images--317
Markup---317
Testing JavaScript--320
Using JsUnit---321
Running a Test---322
How It Works---326
Connoisseur of the Undefined---327
Adding a Little More Realism---328
Manipulating the DOM--328
Aggregating Tests---335
Running Tests by URL---336
Summary--337

CHAPTER 11 Functional Testing--339
Running Acceptance Tests--339
PyFit---340
Writing Requirements---341
A Simple PyFit Example--344
Giving the Acceptance Tests a Home---------------------------------346
Your First FIT.---346
FIT into Buildbot--353
Preparing the Slave---353
Run New Builder, Run!--354
Making the Reports Available---358
Getting Regular Builds--366
What’s Left?---367
Summary.---367

INDEX---369

***Game_Programming_with_Python_Lua_and_Ruby_2003_457p

///

***Getting_Started_with_Pyparsing_2007_65p

What Is Pyparsing?---3

Basic Form of a Pyparsing Program---5
"Hello, World!" on Steroids!---9
What Makes Pyparsing So Special?---14
Parsing Data from a Table—Using Parse Actions and ParseResults--------------------------------17
Extracting Data from a Web Page--26
A Simple S-Expression Parser--35
A Complete S-Expression Parser---38
Parsing a Search String---48
Search Engine in 100 Lines--of
Code--53
Conclusion--62
Index---63

***Gray_Hat_Python_Python_Programming_for_Hackers_and_Reverse_Engineers_2009_220p

FOREWORD by Dave Aitel---xiii
ACKNOWLEDGMENTS---xvii
INTRODUCTION---xix

1 SETTING UP YOUR DEVELOPMENT ENVIRONMENT---------------------------------------1
1.1   Operating System Requirements---2
1.2   Obtaining and Installing Python 2.5--2
1.2.1 Installing Python on Windows---2
1.2.2 Installing Python for Linux--3
1.3   Setting Up Eclipse and PyDev--4
1.3.1 The Hacker’s Best Friend: ctypes---5
1.3.2 Using Dynamic Libraries---6
1.3.3 Constructing C Datatypes---8
1.3.4 Passing Parameters by Reference---9
1.3.5 Defining Structures and Unions--9

2 DEBUGGERS AND DEBUGGERDESIGN--13
2.1   General-Purpose CPU Registers---14
2.2   The Stack---16
2.3   Debug Events--18
2.4   Breakpoints--18
2.4.1 Soft Breakpoints---19
2.4.2 Hardware Breakpoints---21
2.4.3 Memory Breakpoints---23

3 BUILDING A WINDOWS DEBUGGER---25
3.1   Debuggee, Where Art Thou?--25
3.2   Obtaining CPU Register State--33
3.2.1 Thread Enumeration---33
3.2.2 Putting It All Together---35
3.3   Implementing Debug Event Handlers--39
3.4   The Almighty Breakpoint---43
3.4.1 Soft Breakpoints---43
3.4.2 Hardware Breakpoints---47
3.4.3 Memory Breakpoints---52

3.5 Conclusion--55

4 PYDBG — A PURE PYTHON WINDOWS DEBUGGER--------------------------------------57
4.1 Extending Breakpoint Handlers---58
4.2 Access Violation Handlers---60
4.3 Process Snapshots---63
4.3.1 Obtaining Process Snapshots--63
4.3.2 Putting It All Together---65
5 IMMUNITY DEBUGGER—THE BEST OF BOTH WORLDS---------------------------------69
5.1 Installing Immunity Debugger--70
5.2 Immunity Debugger 101---70
5.2.1 PyCommands--71
5.2.2 PyHooks--71
5.3 Exploit Development---73
5.3.1 Finding Exploit-Friendly Instructions--73
5.3.2 Bad-Character Filtering---75
5.3.3 Bypassing DEP on Windows--77
5.4 Defeating Anti-Debugging Routines in Malware--81
5.4.1 IsDebuggerPresent---81
5.4.2 Defeating Process Iteration---82
6 HOOKING---85
6.1 Soft Hooking with PyDbg---86
6.2 Hard Hooking with Immunity Debugger--90

7 DLL AND CODE INJECTION--97
7.1 Remote Thread Creation---98
7.1.1 DLL Injection---99
7.1.2 Code Injection--101
7.2 Getting Evil--104
7.2.1 File Hiding---104
7.2.2 Coding the Backdoor--105
7.2.3 Compiling with py2exe--108
8 FUZZING--111
8.1 Bug Classes---112
8.1.1 Buffer Overflows---112
8.1.2 Integer Overflows--113
8.1.3 Format String Attacks--114
8.2 File Fuzzer---115
8.3 Future Considerations--122
8.3.1 Code Coverage---122
8.3.2 Automated Static Analysis---122

9 SULLEY---123
9.1 Sulley Installation---124
9.2 Sulley Primitives--125
9.2.1 Strings--125
9.2.2 Delimiters--125
9.2.3 Static and Random Primitives---126
9.2.4 Binary Data---126
9.2.5 Integers---126
9.2.6 Blocks and Groups--127
9.3 Slaying WarFTPD with Sulley---129

9.3.1 FTP 101--129
9.3.2 Creating the FTP Protocol Skeleton---130
9.3.3 Sulley Sessions--131
9.3.4 Network and Process Monitoring---132
9.3.5 Fuzzing and the Sulley Web Interface---133

10 FUZZING WINDOWS DRIVERS--137
10.1 Driver Communication--138
10.2 Driver Fuzzing with Immunity Debugger---139
10.3 Driverlib—The Static Analysis Tool for Drivers--------------------------------------142
10.3.1 Discovering Device Names---143
10.3.2 Finding the IOCTL Dispatch Routine---144
10.3.3 Determining Supported IOCTL Codes--145
10.4 Building a Driver Fuzzer--147

11 IDA PYTHON — SCRIPTING IDA PRO---153
11.1 IDAPython Installation--154
11.2 IDAPython Functions--155
11.2.1 Utility Functions---155
11.2.2 Segments---155
11.2.3 Functions--156
11.2.4 Cross-References--156
11.2.5 Debugger Hooks---157
11.3 Example Scripts---158
11.3.1 Finding Dangerous Function Cross-References--------------------------------------158
11.3.2 Function Code Coverage---160
11.3.3 Calculating Stack Size--161

12 PYEMU — THE SCRIPTABLE EMULATOR---163
12.1 Installing PyEmu--164
12.2 PyEmu Overview--164
12.2.1 PyCPU--164
12.2.2 PyMemory--165
12.2.3 PyEmu--165
12.2.4 Execution--165
12.2.5 Memory and Register Modifiers---165
12.2.6 Handlers---166
12.3 IDAPyEmu--171
12.3.1 Function Emulation---172
12.3.2 PEPyEmu--175
12.3.3 Executable Packers---176
12.3.4 UPX Packer--176
12.3.5 Unpacking UPX with PEPyEmu---177
INDEX---183

***Guide_to_NumPy_for_Python_2006_378p

I NumPy from Python--12
1 Origins of NumPy--13
2 Object Essentials---18
2.1 Data-Type Descriptors--20
2.2 Basic indexing (slicing)---23

2.3 Memory Layout of ndarray--26
2.3.1 Contiguous Memory Layout--26
2.3.2 Non-contiguous memory layout--28
2.4 Universal Functions for arrays---30
2.5 Summary of new features---32
2.6 Summary of differences with Numeric---34
2.6.1 First-step changes--34
2.6.2 Second-step changes---37
2.6.3 Updating code that uses Numeric using alter codeN-----------------------------------38
2.6.4 Changes to think about---39
2.7 Summary of differences with Numarray---40
2.7.1 First-step changes--41
2.7.1.1 Import changes---41
2.7.1.2 Attribute and method changes---42
2.7.2 Second-step changes---43
2.7.3 Additional Extension modules---43
3 The Array Object--45
3.1 ndarray Attributes---45
3.1.1 Memory Layout attributes--46
3.1.2 Data Type attributes--50
3.1.3 Other attributes---51
3.1.4 Array Interface attributes--52
3.2 ndarray Methods--55
3.2.1 Array conversion--55
3.2.2 Array shape manipulation---60
3.2.3 Array item selection and manipulation--62
3.2.4 Array calculation---66
3.3 Array Special Methods---72
3.3.1 Methods for standard library functions--72
3.3.2 Basic customization--73
3.3.3 Container customization---75
3.3.4 Arithmetic customization--76
3.3.4.1 Binary---76
3.3.4.2 In-place--78
3.3.4.3 Unary operations--79
3.4 Array indexing--80
3.4.1 Basic Slicing--80
3.4.2 Advanced selection--82
3.4.2.1 Integer---82
3.4.2.2 Boolean---84
3.4.3 Flat Iterator indexing--85
4 Basic Routines---86
4.1 Creating arrays---86
4.2 Operations on two or more arrays---91
4.3 Printing arrays---94
4.4 Functions redundant with methods---95
4.5 Dealing with data types---96
5 Additional Convenience Routines--98
5.1 Shape functions---98
5.2 Basic functions---102
5.3 Polynomial functions--110
5.4 Set Operations---113

5.5 Array construction using index tricks----114
5.6 Other indexing devices----117
5.7 Two-dimensional functions----118
5.8 More data type functions----120
5.9 Functions that behave like ufuncs----123
5.10 Miscellaneous Functions----123
5.11 Utility functions----126
6 Scalar objects----128
6.1 Attributes of array scalars----129
6.2 Methods of array scalars----131
6.3 Defining New Types----132
7 Data-type (dtype) Objects----133
7.1 Attributes----134
7.2 Construction----136
7.3 Methods----139
8 Standard Classes----141
8.1 Special attributes and methods recognized by NumPy----142
8.2 Matrix Objects----143
8.3 Memory-mapped-file arrays----145
8.4 Character arrays (numpy.char)----146
8.5 Record Arrays (numpy.rec)----147
8.6 Masked Arrays (numpy.ma)----151
8.7 Standard container class----152
8.8 Array Iterators----152
8.8.1 Default iteration----153
8.8.2 Flat iteration----153
8.8.3 N-dimensional enumeration----154
8.8.4 Iterator for broadcasting----154
9 Universal Functions----156
9.1 Description----156
9.1.1 Broadcasting----157
9.1.2 Output type determination----157
9.1.3 Use of internal buffers----158
9.1.4 Error handling----158
9.1.5 Optional keyword arguments----159
9.2 Attributes----160
9.3 Casting Rules----161
9.4 Methods----162
9.4.1 Reduce----164
9.4.2 Accumulate----164
9.4.3 Reduceat----165
9.4.4 Outer----166
9.5 Available ufuncs----167
9.5.1 Math operations----167
9.5.2 Trigonometric functions----170
9.5.3 Bit-twiddling functions----171
9.5.4 Comparison functions----172
9.5.5 Floating functions----174
10 Basic Modules----177
10.1 Linear Algebra (linalg)----177
10.2 Discrete Fourier Transforms (fft)----180
10.3 Random Numbers (random)----184

10.3.1 Discrete Distributions--185
10.3.2 Continuous Distributions---187
10.3.3 Miscellaneous utilities---194
10.4 Matrix-specific functions (matlib)---194
10.5 Ctypes utiltity functions (ctypeslib)--194
11 Testing and Packaging---196
11.1 Testing---196
11.2 NumPy Distutils---199
11.2.1 misc util--199
11.2.2 Other modules---206
11.3 Conversion ofsrc files---208
11.3.1 Fortran files--208
11.3.1.1 Named repeat rule---208
11.3.1.2 Short repeat rule--208
11.3.1.3 Pre-defined names---209
11.3.2 Other files--209
II C-API--211
12 New Python Types and C-Structures--212
12.1 New Python Types Defined---213
12.1.1 PyArray Type--214
12.1.2 PyArrayDescr Type---215
12.1.3 PyUFunc Type--223
12.1.4 PyArrayIter Type---226
12.1.5 PyArrayMultiIter Type--227
12.1.6 PyArrayFlags Type--228
12.1.7 ScalarArrayTypes---228
12.2 Other C-Structures--229
12.2.1 PyArray Dims--229
12.2.2 PyArray Chunk---230
12.2.3 PyArrayInterface--230
12.2.4 Internally used structures--232
12.2.4.1 PyUFuncLoopObject---232
12.2.4.2 PyUFuncReduceObject---232
12.2.4.3 PyUFunc Loop1d---232
12.2.4.4 PyArrayMapIter Type--232
13 Complete API--233
13.1 Configuration defines---233
13.1.1 Guaranteed to be defined--233
13.1.2 Possible defines---234
13.2 Array Data Types--235
13.2.1 Enumerated Types---235
13.2.2 Defines---236
13.2.2.1 Max and min values for integers---236
13.2.2.2 Number of bits in data types---236
13.2.2.3 Bit-width references to enumerated typenums----------------------------------237
13.2.2.4 Integer that can hold a pointer---237
13.2.3 C-type names--237
13.2.3.1 Boolean---237
13.2.3.2 (Un)Signed Integer--237
13.2.3.3 (Complex) Floating point--238
13.2.3.4 Bit-width names---238
13.2.4 Printf Formatting--238

13.3 Array API--239
13.3.1 Array structure and data access---239
13.3.1.1 Data access---240
13.3.2 Creating arrays--241
13.3.2.1 From scratch---241
13.3.2.2 From other objects---244
13.3.3 Dealing with types---249
13.3.3.1 General check of Python Type--249
13.3.3.2 Data-type checking---251
13.3.3.3 Converting data types--254
13.3.3.4 New data types--256
13.3.3.5 Special functions for PyArray OBJECT---257
13.3.4 Array flags--258
13.3.4.1 Basic Array Flags---258
13.3.4.2 Combinations of array flags--259
13.3.4.3 Flag-like constants--259
13.3.4.4 Flag checking--260
13.3.5 Array method alternative API---261
13.3.5.1 Conversion---261
13.3.5.2 Shape Manipulation--263
13.3.5.3 Item selection and manipulation--265
13.3.5.4 Calculation---268
13.3.6 Functions---270
13.3.6.1 Array Functions--270
13.3.6.2 Other functions--272
13.3.7 Array Iterators--273
13.3.8 Broadcasting (multi-iterators)---274
13.3.9 Array Scalars---276
13.3.10 Data-type descriptors---278
13.3.11 Conversion Utilities--280
13.3.11.1 For use with PyArg ParseTuple---280
13.3.11.2 Other conversions---282
13.3.12 Miscellaneous---283
13.3.12.1 Importing the API--283
13.3.12.2 Internal Flexibility---284
13.3.12.3 Memory management---285
13.3.12.4 Threading support--285
13.3.12.5 Priority--287
13.3.12.6 Default buffers--287
13.3.12.7 Other constants--287
13.3.12.8 Miscellaneous Macros--288
13.3.12.9 Enumerated Types---289
13.4 UFunc API---289
13.4.1 Constants---289
13.4.2 Macros--290
13.4.3 Functions---290
13.4.4 Generic functions--293
13.5 Importing the API---295
14 How to extend NumPy--297
14.1 Writing an extension module---297
14.2 Required subroutine--298
14.3 Defining functions---299

14.3.1 Functions without keyword arguments 300
14.3.2 Functions with keyword arguments 301
14.3.3 Reference counting 302
14.4 Dealing with array objects 303
14.4.1 Converting an arbitrary sequence object 304
14.4.2 Creating a brand-new ndarray 307
14.4.3 Getting at ndarray memory and accessing elements of the ndarray 308
14.5 Example 309
15 Beyond the Basics 311
15.1 Iterating over elements in the array 311
15.1.1 Basic Iteration 311
15.1.2 Iterating over all but one axis 313
15.1.3 Iterating over multiple arrays 313
15.1.4 Broadcasting over multiple arrays 314
15.2 Creating a new universal function 315
15.3 User-defined data-types 318
15.3.1 Adding the new data-type 319
15.3.2 Registering a casting function 319
15.3.3 Registering coercion rules 320
15.3.4 Registering a ufunc loop 321
15.4 Subtyping the ndarray in C 322
15.4.1 Creating sub-types 322
15.4.2 Specific features of ndarray sub-typing 323
15.4.2.1 The array finalize method 323
15.4.2.2 The array priority attribute 324
15.4.2.3 The array wrap method 324
16 Using Python as glue 325
16.1 Calling other compiled libraries from Python 326
16.2 Hand-generated wrappers 327
16.3 f2py 327
16.3.1 Creating source for a basic extension module 328
16.3.2 Creating a compiled extension module 328
16.3.3 Improving the basic interface 329
16.3.4 Inserting directives in Fortran source 330
16.3.5 A filtering example 331
16.3.6 Calling f2py from Python 332
16.3.7 Automatic extension module generation 333
16.3.8 Conclusion 333
16.4 weave 334
16.4.1 Speed up code involving arrays (also see scipy.numexpr) 334
16.4.2 Inline C-code 335
16.4.3 Simplify creation of an extension module 337
16.4.4 Conclusion 338
16.5 Pyrex 338
16.5.1 Pyrex-add 340
16.5.2 Pyrex-filter 341
16.5.3 Conclusion 342
16.6 ctypes 343
16.6.1 Having a shared library 344
16.6.2 Loading the shared library 345
16.6.3 Converting arguments 346
16.6.4 Calling the function 347

16.6.5 Complete example--348
16.6.6 Conclusion---352
16.7 Additional tools you may find useful---353
16.7.1 SWIG--353
16.7.2 SIP---354
16.7.3 Boost Python---354
16.7.4 Instant--355
16.7.5 PyInline--356
16.7.6 PyFort--356
17 Code Explanations---357
17.1 Memory model--357
17.2 Data-type encapsulation---358
17.3 N-D Iterators---359
17.4 Broadcasting---359
17.5 Array Scalars---360
17.6 Advanced (“Fancy”) Indexing---361
17.6.1 Fancy-indexing check---361
17.6.2 Fancy-indexing implementation---362
17.6.2.1 Creating the mapping object---362
17.6.2.2 Binding the mapping object--362
17.6.2.3 Getting (or Setting)---363
17.7 Universal Functions---363
17.7.1 Setup---364
17.7.2 Function call---365
17.7.2.1 One Loop---366
17.7.2.2 Strided Loop--366
17.7.2.3 Buffered Loop---366
17.7.3 Final output manipulation--367
17.7.4 Methods---367
17.7.4.1 Setup--367
17.7.4.2 Reduce--368
17.7.4.3 Accumulate--369
17.7.4.4 Reduceat---369

***Hello_World_Computer_Programming_for_Kids_and_Other_Beginners_Fourth_printing_2009_432p

Preface---xiii
Acknowledgments--xix
About this book---xxi
1 Getting Started--1
2 Remember This—Memory and Variables--14
3 Basic Math--26
4 Types of Data---38
5 Input---44
6 GUIs—Graphical User Interfaces--52
7 Decisions, Decisions---62
8 Loop the Loop---74
9 Just for You—Comments---89
10 Game Time---94
11 Nested and Variable Loops--99

12 Collecting Things Together—Lists--112
13 Functions---131
14 Objects--146
15 Modules---164
16 Graphics---174
17 Sprites and Collision Detection--202
18 A New Kind of Input—Events--217
19 Sound---239
20 More GUIs--254
21 Print Formatting and Strings--273
22 File Input and Output--290
23 Take a Chance—Randomness--313
24 Computer Simulations--336
25 What’s Next?--358
Appendix A Variable Naming Rules---363
Answers to Self-Test Questions---365
Index--393

***IronPython_in_Action_2009_494p

PART 1 GETTING STARTED WITH IRONPYTHON---1
1 A new language for .NET--3
2 Introduction to Python---29
3 .NET objects and IronPython---62

PART 2 CORE DEVELOPMENT TECHNIQUES---79
4 Writing an application and design patterns with IronPython----------------------81
5 First-class functions in action with XML---110
6 Properties, dialogs, and Visual Studio--133
7 Agile testing: where dynamic typing shines--157
8 Metaprogramming, protocols, and more---183

PART 3 IRONPYTHON AND ADVANCED .NET--215
9 WPF and IronPython---217
10 Windows system administration with IronPython-------------------------------244
11 IronPython and ASP.NET---273
12 Databases and web services--299
13 Silverlight: IronPython in the browser --329

PART 4 REACHING OUT WITH IRONPYTHON---357
14 Extending IronPython with C#/VB.NET---359
15 Embedding the IronPython engine---386

***Learning_Python_2.5_3rd.Edition_2007_748p

Preface---xxix
Part I. Getting Started
1. A Python Q&A Session---3
Why Do People Use Python?--3
Software Quality---5
Developer Productivity---5

Is Python a “Scripting Language”?--6
OK, but What’s the Downside?--7
Who Uses Python Today?---8
What Can I Do with Python?---9
Systems Programming---9
GUIs---9
Internet Scripting---10
Component Integration---10
Database Programming--11
Rapid Prototyping---11
Numeric and Scientific Programming--11
Gaming, Images, AI, XML, Robots, and More--12
What Are Python’s Technical Strengths?--12
It’s Object Oriented---12
It’s Free---13
It’s Portable--13
It’s Powerful--14
It’s Mixable---15
It’s Easy to Use--15
It’s Easy to Learn--17
It’s Named After Monty Python---17
How Does Python Stack Up to Language X?--18
Chapter Summary--19
Brain Builder---20
Chapter Quiz---20
Quiz Answers---20
2. How Python Runs Programs---22
Introducing the Python Interpreter--22
Program Execution---24
The Programmer’s View---24
Python’s View--25
Byte code compilation--25
The Python Virtual Machine (PVM)---26
Performance implications--26
Development implications---27
Execution Model Variations--27
Python Implementation Alternatives--28
CPython---28
Jython--28
IronPython--29
Execution Optimization Tools--29
The Psyco just-in-time compiler--29
The Shedskin C translator--30
Frozen Binaries---31
Future Possibilities?---32
Chapter Summary--32
Brain Builder---33
Chapter Quiz---33
Quiz Answers---33
3. How You Run Programs---34
Interactive Coding--34
Using the Interactive Prompt---37

System Command Lines and Files--37
Using Command Lines and Files--40
Unix Executable Scripts (#!)--41
Clicking File Icons---42
Clicking Icons on Windows---42
The raw_input Trick--44
Other Icon-Click Limitations---45
Module Imports and Reloads--45
The Grander Module Story: Attributes--47
Modules and namespaces--49
import and reload Usage Notes--49
The IDLE User Interface---50
IDLE Basics--51
Using IDLE---52
Advanced IDLE Tools--54
Other IDEs--54
Embedding Calls---56
Frozen Binary Executables---56
Text Editor Launch Options---57
Other Launch Options--57
Future Possibilities?--57
Which Option Should I Use?---58
Chapter Summary---58
Brain Builder---59
Chapter Quiz---59
Quiz Answers--59
Brain Builder: Part I Exercises--61

Part II. Types and Operations

4. Introducing Python Object Types---65
Why Use Built-in Types?---66
Python’s Core Data Types--67
Numbers--68
Strings---69
Sequence Operations--70
Immutability---71
Type-Specific Methods---72
Getting Help---73
Other Ways to Code Strings--74
Pattern Matching--75
Lists---75
Sequence Operations--76
Type-Specific Operations--76
Bounds Checking---77
Nesting--77
List Comprehensions---78
Dictionaries--79
Mapping Operations--79
Nesting Revisited---80
Sorting Keys: for Loops---81
Iteration and Optimization---83
Missing Keys: if Tests--84
Tuples---85

Why Tuples?--85
Files--85
Other File-Like Tools---86
Other Core Types---87
How to Break Your Code’s Flexibility---88
User-Defined Classes---88
And Everything Else--89
Chapter Summary--90
Brain Builder---91
Chapter Quiz---91
Quiz Answers---91
5. Numbers---93
Python Numeric Types---93
Numeric Literals--94
Built-in Numeric Tools and Extensions--95
Python Expression Operators--96
Mixed Operators Follow Operator Precedence---97
Parentheses Group Subexpressions--97
Mixed Types Are Converted Up--97
Preview: Operator Overloading---98
Numbers in Action--99
Variables and Basic Expressions--99
Numeric Display Formats---100
Division: Classic, Floor, and True---102
Bitwise Operations--103
Long Integers---103
Complex Numbers---104
Hexadecimal and Octal Notation--105
Other Built-in Numeric Tools--106
Other Numeric Types---107
Decimal Numbers--107
Sets--108
Booleans---109
Third-Party Extensions---110
Chapter Summary--110
Brain Builder---111
Chapter Quiz---111
Quiz Answers---111
6. The Dynamic Typing Interlude---112
The Case of the Missing Declaration Statements--112
Variables, Objects, and References--112
Types Live with Objects, Not Variables---114
Objects Are Garbage-Collected---115
Shared References--116
Shared References and In-Place Changes---118
Shared References and Equality---119
Dynamic Typing Is Everywhere--121
Chapter Summary---121
Brain Builder---122
Chapter Quiz--122
Quiz Answers--122
7. Strings---123

String Literals--124
Single- and Double-Quoted Strings Are the Same--125
Escape Sequences Represent Special Bytes--125
Raw Strings Suppress Escapes---127
Triple Quotes Code Multiline Block Strings---129
Unicode Strings Encode Larger Character Sets---------------------------------------130
Strings in Action---132
Basic Operations---132
Indexing and Slicing---133
Extended slicing: the third limit---135
String Conversion Tools--136
Character code conversions---138
Changing Strings---139
String Formatting--140
Advanced String Formatting--141
Dictionary-Based String Formatting---142
String Methods---143
String Method Examples: Changing Strings---144
String Method Examples: Parsing Text---146
Other Common String Methods in Action--147
The Original string Module---148
General Type Categories--149
Types Share Operation Sets by Categories---149
Mutable Types Can Be Changed In-Place---150
Chapter Summary--150
Brain Builder---151
Chapter Quiz--151
Quiz Answers--151
8. Lists and Dictionaries---152
Lists---152
Lists in Action--154
Basic List Operations--154
Indexing, Slicing, and Matrixes--155
Changing Lists In-Place---156
Index and slice assignments---156
List method calls--157
Other common list operations---159
Dictionaries---160
Dictionaries in Action--161
Basic Dictionary Operations--162
Changing Dictionaries In-Place--163
More Dictionary Methods---163
A Languages Table--165
Dictionary Usage Notes--166
Using dictionaries to simulate flexible lists---166
Using dictionaries for sparse data structures---167
Avoiding missing-key errors---167
Using dictionaries as “records”---168
Other ways to make dictionaries--169
Chapter Summary--170
Brain Builder---171
Chapter Quiz--171

Quiz Answers--171
9. Tuples, Files, and Everything Else---172
Tuples---172
Tuples in Action---173
Tuple syntax peculiarities: commas and parentheses--174
Conversions and immutability--174
Why Lists and Tuples?--175
Files---176
Opening Files---176
Using Files--177
Files in Action---178
Storing and parsing Python objects in files--178
Storing native Python objects with pickle--180
Storing and parsing packed binary data in files---181
Other File Tools---182
Type Categories Revisited---182
Object Flexibility--183
References Versus Copies---184
Comparisons, Equality, and Truth---186
The Meaning of True and False in Python---188
Python’s Type Hierarchies---189
Other Types in Python--191
Built-in Type Gotchas---191
Assignment Creates References, Not Copies---191
Repetition Adds One Level Deep--192
Beware of Cyclic Data Structures---193
Immutable Types Can’t Be Changed In-Place---193
Chapter Summary---193
Brain Builder---195
Chapter Quiz---195
Quiz Answers---195
Brain Builder: Part II Exercises--196

Part III. Statements and Syntax

10. Introducing Python Statements---201
Python Program Structure Revisited---201
Python’s Statements---202
A Tale of Two ifs---203
What Python Adds--204
What Python Removes--204
Parentheses are optional--204
End of line is end of statement---204
End of indentation is end of block--205
Why Indentation Syntax?---206
A Few Special Cases---208
Statement rule special cases---208
Block rule special case--209
A Quick Example: Interactive Loops---210
A Simple Interactive Loop--210
Doing Math on User Inputs--211
Handling Errors by Testing Inputs--212
Handling Errors with try Statements---213
Nesting Code Three Levels Deep--214

Chapter Summary--215
Brain Builder--216
Chapter Quiz---216
Quiz Answers--216
11. Assignment, Expressions, and print---217
Assignment Statements--217
Assignment Statement Forms--218
Sequence Assignments--219
Advanced sequence assignment patterns--220
Multiple-Target Assignments---222
Multiple-target assignment and shared references--222
Augmented Assignments--223
Augmented assignment and shared references--225
Variable Name Rules--225
Naming conventions---227
Names have no type, but objects do--227
Expression Statements---228
Expression Statements and In-Place Changes--229
print Statements---229
The Python “Hello World” Program--230
Redirecting the Output Stream--231
The print >> file Extension--232
Chapter Summary--234
Brain Builder---235
Chapter Quiz---235
Quiz Answers---235
12. if Tests--236
if Statements--236
General Format--236
Basic Examples--237
Multiway Branching--237
Python Syntax Rules--239
Block Delimiters---240
Statement Delimiters--241
A Few Special Cases--242
Truth Tests--243
The if/else Ternary Expression---244
Chapter Summary--246
Brain Builder---247
Chapter Quiz---247
Quiz Answers---247
13. while and for Loops---248
while Loops--248
General Format--249
Examples--249
break, continue, pass, and the Loop else---250
General Loop Format--250
Examples--251
pass---251
continue---251
break---252
else---252

More on the loop else clause--253
for Loops---254
General Format--254
Examples--256
Basic usage---256
Other data types--256
Tuple assignment in for--257
Nested for loops--257
Iterators: A First Look---258
File Iterators---260
Other Built-in Type Iterators--262
Other Iteration Contexts--263
User-Defined Iterators--264
Loop Coding Techniques--265
Counter Loops: while and range---265
Nonexhaustive Traversals: range--266
Changing Lists: range---267
Parallel Traversals: zip and map---268
Dictionary construction with zip---270
Generating Both Offsets and Items: enumerate-------------------------------------271
List Comprehensions: A First Look---272
List Comprehension Basics---272
Using List Comprehensions on Files---273
Extended List Comprehension Syntax---274
Chapter Summary--275
Brain Builder---276
Chapter Quiz---276
Quiz Answers--276
14. The Documentation Interlude---278
Python Documentation Sources---278
# Comments---279
The dir Function---279
Docstrings: _ _doc_ _--280
User-defined docstrings--281
Docstring standards---282
Built-in docstrings--282
PyDoc: The help Function--283
PyDoc: HTML Reports--285
Standard Manual Set---289
Web Resources--289
Published Books---290
Common Coding Gotchas--291
Chapter Summary---293
Brain Builder--294
Chapter Quiz--294
Quiz Answers---294
Brain Builder: Part III Exercises---295
Part IV Functions
15. Function Basics---299
Why Use Functions?---300
Coding Functions--300
def Statements--302

def Executes at Runtime--303
A First Example: Definitions and Calls--303
Definition--304
Calls--304
Polymorphism in Python--305
A Second Example: Intersecting Sequences--306
Definition--306
Calls--306
Polymorphism Revisited---307
Local Variables---308
Chapter Summary---308
Brain Builder--309
Chapter Quiz---309
Quiz Answers---309
16. Scopes and Arguments---310
Scope Rules---310
Python Scope Basics--311
Name Resolution: The LEGB Rule---312
Scope Example---314
The Built-in Scope--314
The global Statement--316
Minimize Global Variables---317
Minimize Cross-File Changes--318
Other Ways to Access Globals---319
Scopes and Nested Functions--320
Nested Scope Details---320
Nested Scope Examples---321
Factory functions--321
Retaining enclosing scopes’ state with defaults--323
Nested scopes and lambdas--324
Scopes versus defaults with loop variables---324
Arbitrary scope nesting--326
Passing Arguments---326
Arguments and Shared References---327
Avoiding Mutable Argument Changes--329
Simulating Output Parameters--329
Special Argument-Matching Modes--330
Keyword and Default Examples---332
Keywords--332
Defaults--333
Arbitrary Arguments Examples---333
Collecting arguments---334
Unpacking arguments--334
Combining Keywords and Defaults--335
The min Wakeup Call--336
Full credit--336
Bonus points--337
The punch line---338
A More Useful Example: General Set Functions---338
Argument Matching: The Gritty Details---339
Chapter Summary---340
Brain Builder--342

Chapter Quiz--342
Quiz Answers--343
17. Advanced Function Topics---344
Anonymous Functions: lambda---344
lambda Expressions--344
Why Use lambda?--346
How (Not) to Obfuscate Your Python Code--347
Nested lambdas and Scopes---348
Applying Functions to Arguments---350
The apply Built-in---350
Passing keyword arguments---351
apply-Like Call Syntax--351
Mapping Functions over Sequences: map---352
Functional Programming Tools: filter and reduce-----------------------------------353
List Comprehensions Revisited: Mappings--355
List Comprehension Basics--355
Adding Tests and Nested Loops---356
List Comprehensions and Matrixes--358
Comprehending List Comprehensions---360
Iterators Revisited: Generators--360
Generator Function Example---362
Extended Generator Function Protocol: send Versus next-----------------------364
Iterators and Built-in Types---364
Generator Expressions: Iterators Meet List Comprehensions----------------------365
Timing Iteration Alternatives---366
Function Design Concepts---369
Functions Are Objects: Indirect Calls---370
Function Gotchas--371
Local Names Are Detected Statically---372
Defaults and Mutable Objects--373
Functions Without returns---375
Enclosing Scope Loop Variables---375
Chapter Summary--375
Brain Builder--377
Chapter Quiz--377
Quiz Answers---377
Brain Builder: Part IV Exercises---379

Part V Modules

18. Modules: The Big Picture---385
Why Use Modules?--385
Python Program Architecture---386
How to Structure a Program--387
Imports and Attributes--387
Standard Library Modules--389
How Imports Work--389
1. Find It---390
The module search path--390
The sys.path list--392
Module file selection--393
Advanced module selection concepts---393
2. Compile It (Maybe)--394
3. Run It---394

Chapter Summary--395
Brain Builder---397
Chapter Quiz---397
Quiz Answers--397
19. Module Coding Basics---398
Module Creation--398
Module Usage--399
The import Statement---399
The from statement---400
The from * Statement---400
Imports Happen Only Once---400
import and from Are Assignments--401
Cross-File Name Changes--402
import and from Equivalence---402
Potential Pitfalls of the from Statement---403
When import is required---404
Module Namespaces--404
Files Generate Namespaces--405
Attribute Name Qualification---406
Imports Versus Scopes--407
Namespace Nesting--408
Reloading Modules---409
reload Basics---410
reload Example---411
Chapter Summary--412
Brain Builder---414
Chapter Quiz---414
Quiz Answers--414
20. Module Packages---415
Package Import Basics--415
Packages and Search Path Settings--416
Package _ _init_ _.py Files---416
Package Import Example---418
from Versus import with Packages--419
Why Use Package Imports?--420
A Tale of Three Systems---421
Chapter Summary--424
Brain Builder---425
Chapter Quiz---425
Quiz Answers--425
21. Advanced Module Topics---426
Data Hiding in Modules--426
Minimizing from * Damage: _X and _ _all_ _--426
Enabling Future Language Features--427
Mixed Usage Modes: _ _name_ _ and _ _main_ _--428
Unit Tests with _ _name_ _---429
Changing the Module Search Path---430
The import as Extension---431
Relative Import Syntax--431
Why Relative Imports?--432
Module Design Concepts--434
Modules Are Objects: Metaprograms--435

Module Gotchas--437
Statement Order Matters in Top-Level Code---437
Importing Modules by Name String--438
from Copies Names but Doesn’t Link---439
from * Can Obscure the Meaning of Variables---440
reload May Not Impact from Imports--440
reload, from, and Interactive Testing--441
reload Isn’t Applied Transitively---442
Recursive from Imports May Not Work---443
Chapter Summary--444
Brain Builder---445
Chapter Quiz--445
Quiz Answers---445
Brain Builder: Part V Exercises---446
Part VI. Classes and OOP
22. OOP: The Big Picture---451
Why Use Classes?---452
OOP from 30,000 Feet---453
Attribute Inheritance Search---453
Classes and Instances--455
Class Method Calls---456
Coding Class Trees---456
OOP Is About Code Reuse--459
Chapter Summary---462
Brain Builder---463
Chapter Quiz--463
Quiz Answers---463
23. Class Coding Basics--465
Classes Generate Multiple Instance Objects--465
Class Objects Provide Default Behavior--466
Instance Objects Are Concrete Items---466
A First Example--467
Classes Are Customized by Inheritance--469
A Second Example---470
Classes Are Attributes in Modules--471
Classes Can Intercept Python Operators---472
A Third Example---474
Why Use Operator Overloading?--475
The World’s Simplest Python Class--476
Chapter Summary---478
Brain Builder---479
Chapter Quiz--479
Quiz Answers---479
24. Class Coding Details--481
The class Statement--481
General Form---481
Example---482
Methods---484
Example---485
Calling Superclass Constructors--486
Other Method Call Possibilities---486
Inheritance--486

Attribute Tree Construction----------487
Specializing Inherited Methods----------488
Class Interface Techniques----------489
Abstract Superclasses----------490
Operator Overloading----------491
Common Operator Overloading Methods----------492
_ _getitem_ _ Intercepts Index References----------493
_ _getitem_ _ and _ _iter_ _ Implement Iteration----------493
User-Defined Iterators----------494
Multiple iterators on one object----------496
_ _getattr_ _ and _ _setattr_ _ Catch Attribute References----------498
Emulating Privacy for Instance Attributes----------499
_ _repr_ _ and _ _str_ _ Return String Representations----------500
_ _radd_ _ Handles Right-Side Addition----------502
_ _call_ _ Intercepts Calls----------502
Function Interfaces and Callback-Based Code----------503
_ _del_ _ Is a Destructor----------505
Namespaces: The Whole Story----------506
Simple Names: Global Unless Assigned----------506
Attribute Names: Object Namespaces----------506
The “Zen” of Python Namespaces: Assignments Classify Names----------506
Namespace Dictionaries----------508
Namespace Links----------511
A More Realistic Example----------512
Chapter Summary----------515
Brain Builder----------516
Chapter Quiz----------516
Quiz Answers----------516
25. Designing with Classes----------518
Python and OOP----------518
Overloading by Call Signatures (or Not)----------519
Classes As Records----------519
OOP and Inheritance: “Is-a” Relationships----------521
OOP and Composition: “Has-a” Relationships----------523
Stream Processors Revisited----------524
OOP and Delegation----------527
Multiple Inheritance----------529
Classes Are Objects: Generic Object Factories----------532
Why Factories?----------533
Methods Are Objects: Bound or Unbound----------534
Documentation Strings Revisited----------535
Classes Versus Modules----------537
Chapter Summary----------537
Brain Builder----------538
Chapter Quiz----------538
Quiz Answers----------538
26. Advanced Class Topics----------539
Extending Built-in Types----------539
Extending Types by Embedding----------540
Extending Types by Subclassing----------540
Pseudoprivate Class Attributes----------543
Name Mangling Overview----------543

Why Use Pseudoprivate Attributes?--544
New-Style Classes--545
Diamond Inheritance Change--546
Diamond inheritance example--547
Explicit conflict resolution--547
Other New-Style Class Extensions---549
Static and class methods---549
Instance slots---549
Class properties---550
New _ _getattribute_ _ overloading method--552
Static and Class Methods--552
Using Static and Class Methods--554
Function Decorators--556
Decorator Example--558
Class Gotchas--559
Changing Class Attributes Can Have Side Effects--559
Multiple Inheritance: Order Matters---560
Methods, Classes, and Nested Scopes---561
“Overwrapping-itis”--563
Chapter Summary--564
Brain Builder--565
Chapter Quiz--565
Quiz Answers---565
Brain Builder: Part VI Exercises---566
Part VII. Exceptions and Tools
27. Exception Basics--575
Why Use Exceptions?---576
Exception Roles--576
Exception Handling: The Short Story--577
The try/except/else Statement---581
try Statement Clauses---582
The try/else Clause--585
Example: Default Behavior--585
Example: Catching Built-in Exceptions--586
The try/finally Statement---587
Example: Coding Termination Actions with try/finally-------------------------------------588
Unified try/except/finally---589
Combining finally and except by Nesting--590
Unified try Example---591
The raise Statement--592
Example: Raising and Catching User-Defined Exceptions----------------------------------593
Example: Passing Extra Data with raise--593
Example: Propagating Exceptions with raise--594
The assert Statement--595
Example: Trapping Constraints (but Not Errors)---595
with/as Context Managers---596
Basic Usage---596
The Context Management Protocol---598
Chapter Summary--600
Brain Builder--601
Chapter Quiz--601
Quiz Answers---601

28. Exception Objects--602
String-Based Exceptions--603
String Exceptions Are Right Out!---603
Class-Based Exceptions--604
Class Exception Example--604
Why Class Exceptions?--606
Built-in Exception Classes--609
Specifying Exception Text--610
Sending Extra Data and Behavior in Instances--611
Example: Extra data with classes and strings--611
General raise Statement Forms--613
Chapter Summary---615
Brain Builder--616
Chapter Quiz--616
Quiz Answers--616
29. Designing with Exceptions--617
Nesting Exception Handlers--617
Example: Control-Flow Nesting--619
Example: Syntactic Nesting--619
Exception Idioms---621
Exceptions Aren't Always Errors---621
Functions Signal Conditions with raise--622
Debugging with Outer try Statements--622
Running In-Process Tests--623
More on sys.exc_info--624
Exception Design Tips---624
What Should Be Wrapped---624
Catching Too Much: Avoid Empty excepts--625
Catching Too Little: Use Class-Based Categories---627
Exception Gotchas--627
String Exceptions Match by Identity, Not by Value--628
Catching the Wrong Thing---629
Core Language Summary---629
The Python Toolset---630
Development Tools for Larger Projects--631
Chapter Summary---634
Brain Builder--635
Chapter Quiz--635
Quiz Answers--635
Brain Builder: Part VII Exercises---636
Part VIII Appendixes
A. Installation and Configuration---639
B. Solutions to End-of-Part Exercises---646
Index---681

***learning_python_fourth_edition_2009_1213p

Part I. Getting Started
1. A Python Q&A Session---3
Why Do People Use Python?---3
Software Quality--4
Developer Productivity--5

Is Python a “Scripting Language”?--5
OK, but What’s the Downside?---7
Who Uses Python Today?---7
What Can I Do with Python?--9
Systems Programming---9
GUIs--9
Internet Scripting---10
Component Integration--10
Database Programming--11
Rapid Prototyping---11
Numeric and Scientific Programming--11
Gaming, Images, Serial Ports, XML, Robots, and More--12
How Is Python Supported?---12
What Are Python’s Technical Strengths?---13
It’s Object-Oriented--13
It’s Free---13
It’s Portable--14
It’s Powerful---15
It’s Mixable--16
It’s Easy to Use--16
It’s Easy to Learn--17
It’s Named After Monty Python---17
How Does Python Stack Up to Language X?--17
Chapter Summary--18
Test Your Knowledge: Quiz--19
Test Your Knowledge: Answers---19

2. How Python Runs Programs---23
Introducing the Python Interpreter--23
Program Execution---24
The Programmer’s View---24
Python’s View--26
Execution Model Variations--29
Python Implementation Alternatives--29
Execution Optimization Tools---30
Frozen Binaries--32
Other Execution Options--33
Future Possibilities?---33
Chapter Summary--34
Test Your Knowledge: Quiz--34
Test Your Knowledge: Answers---34

3. How You Run Programs---35
The Interactive Prompt--35
Running Code Interactively---37
Why the Interactive Prompt?--38
Using the Interactive Prompt---39
System Command Lines and Files--41
A First Script---42
Running Files with Command Lines---43
Using Command Lines and Files--44
Unix Executable Scripts (#!)--46

Clicking File Icons--47
Clicking Icons on Windows---47
The input Trick---49
Other Icon-Click Limitations---50
Module Imports and Reloads--51
The Grander Module Story: Attributes---53
import and reload Usage Notes--56
Using exec to Run Module Files--57
The IDLE User Interface--58
IDLE Basics---58
Using IDLE--60
Advanced IDLE Tools---62
Other IDEs--63
Other Launch Options---64
Embedding Calls--64
Frozen Binary Executables--65
Text Editor Launch Options--65
Still Other Launch Options--66
Future Possibilities?---66
Which Option Should I Use?---66
Chapter Summary--68
Test Your Knowledge: Quiz--68
Test Your Knowledge: Answers--69
Test Your Knowledge: Part I Exercises--70

Part II. Types and Operations

4. Introducing Python Object Types--75
Why Use Built-in Types?--76
Python’s Core Data Types--77
Numbers---78
Strings--80
Sequence Operations--80
Immutability---82
Type-Specific Methods--82
Getting Help--84
Other Ways to Code Strings--85
Pattern Matching--85
Lists---86
Sequence Operations--86
Type-Specific Operations---87
Bounds Checking--87
Nesting--88
Comprehensions---88
Dictionaries--90
Mapping Operations---90
Nesting Revisited--91
Sorting Keys: for Loops---93
Iteration and Optimization--94
Missing Keys: if Tests---95
Tuples---96
Why Tuples?--97

Files--97
Other File-Like Tools--99
Other Core Types--99
How to Break Your Code’s Flexibility--100
User-Defined Classes--101
And Everything Else--102
Chapter Summary--103
Test Your Knowledge: Quiz--103
Test Your Knowledge: Answers--104

5. Numeric Types--105
Numeric Type Basics--105
Numeric Literals--106
Built-in Numeric Tools--108
Python Expression Operators--108
Numbers in Action--113
Variables and Basic Expressions--113
Numeric Display Formats--115
Comparisons: Normal and Chained--116
Division: Classic, Floor, and True--117
Integer Precision--121
Complex Numbers--122
Hexadecimal, Octal, and Binary Notation--122
Bitwise Operations--124
Other Built-in Numeric Tools--125
Other Numeric Types--127
Decimal Type--127
Fraction Type--129
Sets--133
Booleans--139
Numeric Extensions--140
Chapter Summary--141
Test Your Knowledge: Quiz--141
Test Your Knowledge: Answers--141

6. The Dynamic Typing Interlude--143
The Case of the Missing Declaration Statements--143
Variables, Objects, and References--144
Types Live with Objects, Not Variables--145
Objects Are Garbage-Collected--146
Shared References--148
Shared References and In-Place Changes--149
Shared References and Equality--151
Dynamic Typing Is Everywhere--152
Chapter Summary--153
Test Your Knowledge: Quiz--153
Test Your Knowledge: Answers--154
7. Strings--155
String Literals--157
Single- and Double-Quoted Strings Are the Same--158
Escape Sequences Represent Special Bytes--158
Raw Strings Suppress Escapes--161

Triple Quotes Code Multiline Block Strings---162
Strings in Action--163
Basic Operations---164
Indexing and Slicing---165
String Conversion Tools--169
Changing Strings--171
String Methods--172
String Method Examples: Changing Strings---174
String Method Examples: Parsing Text---176
Other Common String Methods in Action--177
The Original string Module (Gone in 3.0)---178
String Formatting Expressions---179
Advanced String Formatting Expressions--181
Dictionary-Based String Formatting Expressions--182
String Formatting Method Calls---183
The Basics---184
Adding Keys, Attributes, and Offsets--184
Adding Specific Formatting---185
Comparison to the % Formatting Expression--187
Why the New Format Method?--190
General Type Categories--193
Types Share Operation Sets by Categories---194
Mutable Types Can Be Changed In-Place--194
Chapter Summary--195
Test Your Knowledge: Quiz--195
Test Your Knowledge: Answers--196

8. Lists and Dictionaries--197
Lists--197
Lists in Action---200
Basic List Operations--200
List Iteration and Comprehensions---200
Indexing, Slicing, and Matrixes--201
Changing Lists In-Place---202
Dictionaries--207
Dictionaries in Action---209
Basic Dictionary Operations---209
Changing Dictionaries In-Place--210
More Dictionary Methods--211
A Languages Table---212
Dictionary Usage Notes---213
Other Ways to Make Dictionaries---216
Dictionary Changes in Python 3.0--217
Chapter Summary--223
Test Your Knowledge: Quiz--224
Test Your Knowledge: Answers---224

9. Tuples, Files, and Everything Else---225
Tuples---225
Tuples in Action---227
Why Lists and Tuples?---229
Files--229

Opening Files---230
Using Files--231
Files in Action---232
Other File Tools---238
Type Categories Revisited---239
Object Flexibility---241
References Versus Copies--241
Comparisons, Equality, and Truth---244
Python 3.0 Dictionary Comparisons---246
The Meaning of True and False in Python--246
Python's Type Hierarchies---248
Type Objects--249
Other Types in Python---250
Built-in Type Gotchas--251
Assignment Creates References, Not Copies---251
Repetition Adds One Level Deep---252
Beware of Cyclic Data Structures--252
Immutable Types Can't Be Changed In-Place---253
Chapter Summary---253
Test Your Knowledge: Quiz--254
Test Your Knowledge: Answers---254
Test Your Knowledge: Part II Exercises--255

Part III. Statements and Syntax

10. Introducing Python Statements---261
Python Program Structure Revisited--261
Python's Statements--262
A Tale of Two ifs--264
What Python Adds---264
What Python Removes--265
Why Indentation Syntax?--266
A Few Special Cases--269
A Quick Example: Interactive Loops---271
A Simple Interactive Loop--271
Doing Math on User Inputs---272
Handling Errors by Testing Inputs---273
Handling Errors with try Statements--274
Nesting Code Three Levels Deep--275
Chapter Summary--276
Test Your Knowledge: Quiz---276
Test Your Knowledge: Answers--277

11. Assignments, Expressions, and Prints--279
Assignment Statements--279
Assignment Statement Forms---280
Sequence Assignments--281
Extended Sequence Unpacking in Python 3.0---284
Multiple-Target Assignments--288
Augmented Assignments---289
Variable Name Rules--292
Expression Statements--295

Expression Statements and In-Place Changes--296
Print Operations--297
The Python 3.0 print Function---298
The Python 2.6 print Statement--300
Print Stream Redirection---302
Version-Neutral Printing---306
Chapter Summary--308
Test Your Knowledge: Quiz---308
Test Your Knowledge: Answers--308

12. if Tests and Syntax Rules---311
if Statements--311
General Format---311
Basic Examples---312
Multiway Branching--312
Python Syntax Rules---314
Block Delimiters: Indentation Rules---315
Statement Delimiters: Lines and Continuations---317
A Few Special Cases--318
Truth Tests---320
The if/else Ternary Expression---321
Chapter Summary---324
Test Your Knowledge: Quiz---324
Test Your Knowledge: Answers---324

13. while and for Loops--327
while Loops---327
General Format---328
Examples--328
break, continue, pass, and the Loop else--329
General Loop Format---329
pass---330
continue---331
break--331
Loop else---332
for Loops---334
General Format---334
Examples--335
Loop Coding Techniques--341
Counter Loops: while and range---342
Nonexhaustive Traversals: range and Slices---343
Changing Lists: range---344
Parallel Traversals: zip and map---345
Generating Both Offsets and Items: enumerate---348
Chapter Summary---349
Test Your Knowledge: Quiz---349
Test Your Knowledge: Answers---350

14. Iterations and Comprehensions, Part 1---351
Iterators: A First Look--351
The Iteration Protocol: File Iterators---352
Manual Iteration: iter and next---354

Other Built-in Type Iterators--356
List Comprehensions: A First Look---358
List Comprehension Basics---359
Using List Comprehensions on Files---359
Extended List Comprehension Syntax--361
Other Iteration Contexts--362
New Iterables in Python 3.0---366
The range Iterator---367
The map, zip, and filter Iterators--368
Multiple Versus Single Iterators---369
Dictionary View Iterators---370
Other Iterator Topics---372
Chapter Summary---372
Test Your Knowledge: Quiz--372
Test Your Knowledge: Answers--373

15. The Documentation Interlude---375
Python Documentation Sources--375
# Comments---376
The dir Function---376
Docstrings: __doc__---377
PyDoc: The help Function---380
PyDoc: HTML Reports--383
The Standard Manual Set---386
Web Resources--387
Published Books---387
Common Coding Gotchas--387
Chapter Summary--389
Test Your Knowledge: Quiz---389
Test Your Knowledge: Answers---390
Test Your Knowledge: Part III Exercises---390

Part IV Functions

16. Function Basics---395
Why Use Functions?---396
Coding Functions--396
def Statements---398
def Executes at Runtime---399
A First Example: Definitions and Calls---400
Definition--400
Calls---400
Polymorphism in Python---401
A Second Example: Intersecting Sequences---402
Definition--402
Calls---403
Polymorphism Revisited--403
Local Variables--404
Chapter Summary---404
Test Your Knowledge: Quiz---405
Test Your Knowledge: Answers---405
17. Scopes---407

Python Scope Basics--407
Scope Rules--408
Name Resolution: The LEGB Rule---410
Scope Example---411
The Built-in Scope--412
The global Statement---414
Minimize Global Variables--415
Minimize Cross-File Changes---416
Other Ways to Access Globals--418
Scopes and Nested Functions---419
Nested Scope Details---419
Nested Scope Examples---419
The nonlocal Statement--425
nonlocal Basics--425
nonlocal in Action--426
Why nonlocal?---429
Chapter Summary---432
Test Your Knowledge: Quiz---433
Test Your Knowledge: Answers--434

18. Arguments--435
Argument-Passing Basics---435
Arguments and Shared References---436
Avoiding Mutable Argument Changes--438
Simulating Output Parameters---439
Special Argument-Matching Modes---440
The Basics--441
Matching Syntax---442
The Gritty Details--443
Keyword and Default Examples--444
Arbitrary Arguments Examples---446
Python 3.0 Keyword-Only Arguments---450
The min Wakeup Call!--453
Full Credit---454
Bonus Points--455
The Punch Line...--456
Generalized Set Functions--456
Emulating the Python 3.0 print Function---457
Using Keyword-Only Arguments--459
Chapter Summary---460
Test Your Knowledge: Quiz---461
Test Your Knowledge: Answers--462
19. Advanced Function Topics--463
Function Design Concepts---463
Recursive Functions--465
Summation with Recursion---465
Coding Alternatives---466
Loop Statements Versus Recursion--467
Handling Arbitrary Structures---468
Function Objects: Attributes and Annotations---469
Indirect Function Calls---469
Function Introspection--470

Function Attributes--471
Function Annotations in 3.0--472
Anonymous Functions: lambda---474
lambda Basics---474
Why Use lambda?---475
How (Not) to Obfuscate Your Python Code---477
Nested lambdas and Scopes---478
Mapping Functions over Sequences: map--479
Functional Programming Tools: filter and reduce--481
Chapter Summary---483
Test Your Knowledge: Quiz--483
Test Your Knowledge: Answers--483

20. Iterations and Comprehensions, Part 2---485
List Comprehensions Revisited: Functional Tools---485
List Comprehensions Versus map---486
Adding Tests and Nested Loops: filter--487
List Comprehensions and Matrixes--489
Comprehending List Comprehensions---490
Iterators Revisited: Generators---492
Generator Functions: yield Versus return--492
Generator Expressions: Iterators Meet Comprehensions------------------------------------497
Generator Functions Versus Generator Expressions--498
Generators Are Single-Iterator Objects---499
Emulating zip and map with Iteration Tools--500
Value Generation in Built-in Types and Classes---506
3.0 Comprehension Syntax Summary---507
Comprehending Set and Dictionary Comprehensions--------------------------------------507
Extended Comprehension Syntax for Sets and Dictionaries--------------------------------508
Timing Iteration Alternatives---509
Timing Module--509
Timing Script---510
Timing Results---511
Timing Module Alternatives---513
Other Suggestions---517
Function Gotchas---518
Local Names Are Detected Statically--518
Defaults and Mutable Objects--520
Functions Without returns---522
Enclosing Scope Loop Variables--522
Chapter Summary---522
Test Your Knowledge: Quiz--523
Test Your Knowledge: Answers--523
Test Your Knowledge: Part IV Exercises--524

Part V Modules

21. Modules: The Big Picture--529
Why Use Modules?---529
Python Program Architecture---530
How to Structure a Program--531
Imports and Attributes---531

Standard Library Modules---533
How Imports Work--533
1. Find It---534
2. Compile It (Maybe)--534
3. Run It--535
The Module Search Path---535
Configuring the Search Path--537
Search Path Variations--538
The sys.path List---538
Module File Selection--539
Advanced Module Selection Concepts--540
Chapter Summary---541
Test Your Knowledge: Quiz---541
Test Your Knowledge: Answers---542

22. Module Coding Basics---543
Module Creation---543
Module Usage---544
The import Statement---544
The from Statement---545
The from * Statement---545
Imports Happen Only Once---546
import and from Are Assignments---546
Cross-File Name Changes--547
import and from Equivalence---548
Potential Pitfalls of the from Statement--548
Module Namespaces--550
Files Generate Namespaces---550
Attribute Name Qualification---552
Imports Versus Scopes---552
Namespace Nesting---553
Reloading Modules---554
reload Basics---555
reload Example---556
Chapter Summary---558
Test Your Knowledge: Quiz---558
Test Your Knowledge: Answers---558

23. Module Packages--561
Package Import Basics---561
Packages and Search Path Settings---562
Package __init__.py Files---563
Package Import Example---564
from Versus import with Packages--566
Why Use Package Imports?---566
A Tale of Three Systems---567
Package Relative Imports---569
Changes in Python 3.0---570
Relative Import Basics---570
Why Relative Imports?---572
The Scope of Relative Imports--574
Module Lookup Rules Summary---575

Relative Imports in Action--575
Chapter Summary---581
Test Your Knowledge: Quiz--582
Test Your Knowledge: Answers--582

24. Advanced Module Topics---583
Data Hiding in Modules--583
Minimizing from * Damage: _X and __all__---584
Enabling Future Language Features--584
Mixed Usage Modes: __name__ and __main__--585
Unit Tests with __name__---586
Using Command-Line Arguments with __name__--587
Changing the Module Search Path--590
The as Extension for import and from---591
Modules Are Objects: Metaprograms---591
Importing Modules by Name String--594
Transitive Module Reloads---595
Module Design Concepts--598
Module Gotchas--599
Statement Order Matters in Top-Level Code--599
from Copies Names but Doesn’t Link--600
from * Can Obscure the Meaning of Variables--601
reload May Not Impact from Imports---601
reload, from, and Interactive Testing---602
Recursive from Imports May Not Work--603
Chapter Summary---604
Test Your Knowledge: Quiz---604
Test Your Knowledge: Answers---605
Test Your Knowledge: Part V Exercises--605

Part VI. Classes and OOP

25. OOP: The Big Picture---611
Why Use Classes?--612
OOP from 30,000 Feet--613
Attribute Inheritance Search---613
Classes and Instances--615
Class Method Calls---616
Coding Class Trees--616
OOP Is About Code Reuse---619
Chapter Summary---622
Test Your Knowledge: Quiz--622
Test Your Knowledge: Answers---622

26. Class Coding Basics--625
Classes Generate Multiple Instance Objects--625
Class Objects Provide Default Behavior--626
Instance Objects Are Concrete Items--626
A First Example---627
Classes Are Customized by Inheritance---629
A Second Example---630
Classes Are Attributes in Modules--631

Classes Can Intercept Python Operators--633
A Third Example---634
Why Use Operator Overloading?---636
The World’s Simplest Python Class---636
Classes Versus Dictionaries---639
Chapter Summary---641
Test Your Knowledge: Quiz--641
Test Your Knowledge: Answers---641

27. A More Realistic Example--643
Step 1: Making Instances--644
Coding Constructors---644
Testing As You Go---645
Using Code Two Ways---646
Step 2: Adding Behavior Methods--648
Coding Methods--649
Step 3: Operator Overloading---651
Providing Print Displays--652
Step 4: Customizing Behavior by Subclassing--653
Coding Subclasses---653
Augmenting Methods: The Bad Way--654
Augmenting Methods: The Good Way--654
Polymorphism in Action--656
Inherit, Customize, and Extend---657
OOP: The Big Idea--658
Step 5: Customizing Constructors, Too---658
OOP Is Simpler Than You May Think--660
Other Ways to Combine Classes---660
Step 6: Using Introspection Tools--663
Special Class Attributes--664
A Generic Display Tool---665
Instance Versus Class Attributes---666
Name Considerations in Tool Classes---667
Our Classes’ Final Form--668
Step 7 (Final): Storing Objects in a Database--669
Pickles and Shelves---670
Storing Objects on a Shelve Database--671
Exploring Shelves Interactively---672
Updating Objects on a Shelve---674
Future Directions--675
Chapter Summary--677
Test Your Knowledge: Quiz---677
Test Your Knowledge: Answers---678

28. Class Coding Details--681
The class Statement---681
General Form---681
Example---682
Methods---684
Method Example--685
Calling Superclass Constructors---686
Other Method Call Possibilities--686

Inheritance--687
Attribute Tree Construction---687
Specializing Inherited Methods--687
Class Interface Techniques---689
Abstract Superclasses--690
Python 2.6 and 3.0 Abstract Superclasses--692
Namespaces: The Whole Story---693
Simple Names: Global Unless Assigned--693
Attribute Names: Object Namespaces---693
The “Zen” of Python Namespaces: Assignments Classify Names------------------------694
Namespace Dictionaries--696
Namespace Links---699
Documentation Strings Revisited--701
Classes Versus Modules--703
Chapter Summary---703
Test Your Knowledge: Quiz---703
Test Your Knowledge: Answers--704

29. Operator Overloading---705
The Basics---705
Constructors and Expressions: __init__ and __sub__----------------------------------706
Common Operator Overloading Methods---706
Indexing and Slicing: __getitem__ and __setitem__-----------------------------------708
Intercepting Slices--708
Index Iteration: __getitem__---710
Iterator Objects: __iter__ and __next__---711
User-Defined Iterators---712
Multiple Iterators on One Object--714
Membership: __contains__, __iter__, and __getitem__--------------------------------716
Attribute Reference: __getattr__ and __setattr__---718
Other Attribute Management Tools---719
Emulating Privacy for Instance Attributes: Part 1-------------------------------------720
String Representation: __repr__ and __str__--721
Right-Side and In-Place Addition: __radd__ and __iadd__----------------------------723
In-Place Addition--725
Call Expressions: __call__--725
Function Interfaces and Callback-Based Code--727
Comparisons: __lt__, __gt__, and Others--728
The 2.6 __cmp__ Method (Removed in 3.0)--729
Boolean Tests: __bool__ and __len__---730
Object Destruction: __del__--732
Chapter Summary---733
Test Your Knowledge: Quiz---734
Test Your Knowledge: Answers--734

30. Designing with Classes--737
Python and OOP---737
Overloading by Call Signatures (or Not)--738
OOP and Inheritance: “Is-a” Relationships---739
OOP and Composition: “Has-a” Relationships---740
Stream Processors Revisited--742
OOP and Delegation: “Wrapper” Objects---745

Pseudoprivate Class Attributes--747
Name Mangling Overview--748
Why Use Pseudoprivate Attributes?---748
Methods Are Objects: Bound or Unbound--750
Unbound Methods are Functions in 3.0---752
Bound Methods and Other Callable Objects--754
Multiple Inheritance: “Mix-in” Classes--756
Coding Mix-in Display Classes--757
Classes Are Objects: Generic Object Factories---768
Why Factories?--769
Other Design-Related Topics---770
Chapter Summary---770
Test Your Knowledge: Quiz---770
Test Your Knowledge: Answers---771

31. Advanced Class Topics--773
Extending Built-in Types---773
Extending Types by Embedding---774
Extending Types by Subclassing--775
The “New-Style” Class Model---777
New-Style Class Changes---778
Type Model Changes--779
Diamond Inheritance Change--783
New-Style Class Extensions--788
Instance Slots---788
Class Properties--792
__getattribute__ and Descriptors---794
Metaclasses---794
Static and Class Methods--795
Why the Special Methods?--795
Static Methods in 2.6 and 3.0---796
Static Method Alternatives--798
Using Static and Class Methods--799
Counting Instances with Static Methods---800
Counting Instances with Class Methods--802
Decorators and Metaclasses: Part 1---804
Function Decorator Basics--804
A First Function Decorator Example---805
Class Decorators and Metaclasses--807
For More Details---808
Class Gotchas---808
Changing Class Attributes Can Have Side Effects--808
Changing Mutable Class Attributes Can Have Side Effects, Too---------------------------810
Multiple Inheritance: Order Matters--811
Methods, Classes, and Nested Scopes--812
Delegation-Based Classes in 3.0: __getattr__ and built-ins---------------------------------814
“Overwrapping-itis”---814
Chapter Summary--815
Test Your Knowledge: Quiz---815
Test Your Knowledge: Answers--815
Test Your Knowledge: Part VI Exercises--816

Part VII. Exceptions and Tools

32. Exception Basics--825
Why Use Exceptions?--825
Exception Roles--826
Exceptions: The Short Story---827
Default Exception Handler--827
Catching Exceptions---828
Raising Exceptions---829
User-Defined Exceptions--830
Termination Actions---830
Chapter Summary---833
Test Your Knowledge: Quiz--833
Test Your Knowledge: Answers--833

33. Exception Coding Details--835
The try/except/else Statement--835
try Statement Clauses--837
The try else Clause--839
Example: Default Behavior--840
Example: Catching Built-in Exceptions--841
The try/finally Statement--842
Example: Coding Termination Actions with try/finally---------------------------------------843
Unified try/except/finally---844
Unified try Statement Syntax--845
Combining finally and except by Nesting--845
Unified try Example---846
The raise Statement--848
Propagating Exceptions with raise---849
Python 3.0 Exception Chaining: raise from---849
The assert Statement--850
Example: Trapping Constraints (but Not Errors!)--851
with/as Context Managers--851
Basic Usage---852
The Context Management Protocol---853
Chapter Summary--855
Test Your Knowledge: Quiz--856
Test Your Knowledge: Answers--856

34. Exception Objects---857
Exceptions: Back to the Future--858
String Exceptions Are Right Out!--858
Class-Based Exceptions--859
Coding Exceptions Classes---859
Why Exception Hierarchies?---861
Built-in Exception Classes---864
Built-in Exception Categories--865
Default Printing and State--866
Custom Print Displays--867
Custom Data and Behavior---868
Providing Exception Details--868
Providing Exception Methods---869

Chapter Summary--870
Test Your Knowledge: Quiz--871
Test Your Knowledge: Answers---871

35. Designing with Exceptions---873
Nesting Exception Handlers---873
Example: Control-Flow Nesting---875
Example: Syntactic Nesting---875
Exception Idioms---877
Exceptions Aren’t Always Errors--877
Functions Can Signal Conditions with raise--878
Closing Files and Server Connections--878
Debugging with Outer try Statements--879
Running In-Process Tests---880
More on sys.exc_info---881
Exception Design Tips and Gotchas--882
What Should Be Wrapped--882
Catching Too Much: Avoid Empty except and Exception------------------------------------883
Catching Too Little: Use Class-Based Categories--885
Core Language Summary--885
The Python Toolset--886
Development Tools for Larger Projects---887
Chapter Summary--890
Test Your Knowledge: Quiz---891
Test Your Knowledge: Answers--891
Test Your Knowledge: Part VII Exercises--891

Part VIII. Advanced Topics

36. Unicode and Byte Strings---895
String Changes in 3.0---896
String Basics---897
Character Encoding Schemes---897
Python’s String Types--899
Text and Binary Files---900
Python 3.0 Strings in Action---902
Literals and Basic Properties--902
Conversions--903
Coding Unicode Strings--904
Coding ASCII Text--905
Coding Non-ASCII Text--905
Encoding and Decoding Non-ASCII text--906
Other Unicode Coding Techniques---907
Converting Encodings---909
Coding Unicode Strings in Python 2.6---910
Source File Character Set Encoding Declarations---912
Using 3.0 Bytes Objects--913
Method Calls--913
Sequence Operations--914
Other Ways to Make bytes Objects---915
Mixing String Types---916
Using 3.0 (and 2.6) bytearray Objects--917

Using Text and Binary Files--920
Text File Basics--920
Text and Binary Modes in 3.0--921
Type and Content Mismatches---923
Using Unicode Files---924
Reading and Writing Unicode in 3.0---924
Handling the BOM in 3.0---926
Unicode Files in 2.6---928
Other String Tool Changes in 3.0--929
The re Pattern Matching Module---929
The struct Binary Data Module--930
The pickle Object Serialization Module---932
XML Parsing Tools---934
Chapter Summary--937
Test Your Knowledge: Quiz---937
Test Your Knowledge: Answers--937

37. Managed Attributes--941
Why Manage Attributes?--941
Inserting Code to Run on Attribute Access---942
Properties--943
The Basics--943
A First Example--944
Computed Attributes--945
Coding Properties with Decorators--946
Descriptors--947
The Basics--948
A First Example--950
Computed Attributes--952
Using State Information in Descriptors---953
How Properties and Descriptors Relate---955
__getattr__ and __getattribute__---956
The Basics--957
A First Example--959
Computed Attributes--961
__getattr__ and __getattribute__ Compared--962
Management Techniques Compared---963
Intercepting Built-in Operation Attributes---966
Delegation-Based Managers Revisited--970
Example: Attribute Validations--973
Using Properties to Validate--973
Using Descriptors to Validate---975
Using __getattr__ to Validate---977
Using __getattribute__ to Validate---978
Chapter Summary--979
Test Your Knowledge: Quiz---980
Test Your Knowledge: Answers--980

38. Decorators--983
What’s a Decorator?---983
Managing Calls and Instances--984
Managing Functions and Classes--984

Using and Defining Decorators--984
Why Decorators?---985
The Basics---986
Function Decorators---986
Class Decorators--990
Decorator Nesting---993
Decorator Arguments---994
Decorators Manage Functions and Classes, Too--995
Coding Function Decorators--996
Tracing Calls---996
State Information Retention Options---997
Class Blunders I: Decorating Class Methods---1001
Timing Calls---1006
Adding Decorator Arguments---1008
Coding Class Decorators---1011
Singleton Classes--1011
Tracing Object Interfaces---1013
Class Blunders II: Retaining Multiple Instances--1016
Decorators Versus Manager Functions--1018
Why Decorators? (Revisited)--1019
Managing Functions and Classes Directly---1021
Example: “Private” and “Public” Attributes---1023
Implementing Private Attributes---1023
Implementation Details I--1025
Generalizing for Public Declarations, Too--1026
Implementation Details II---1029
Open Issues--1030
Python Isn’t About Control---1034
Example: Validating Function Arguments---1034
The Goal---1034
A Basic Range-Testing Decorator for Positional Arguments-------------------------------1035
Generalizing for Keywords and Defaults, Too--1037
Implementation Details--1040
Open Issues--1042
Decorator Arguments Versus Function Annotations--------------------------------------1043
Other Applications: Type Testing (If You Insist!)--1045
Chapter Summary---1046
Test Your Knowledge: Quiz---1047
Test Your Knowledge: Answers---1047

39. Metaclasses--1051
To Metaclass or Not to Metaclass---1052
Increasing Levels of Magic--1052
The Downside of “Helper” Functions--1054
Metaclasses Versus Class Decorators: Round 1---1056
The Metaclass Model--1058
Classes Are Instances of type---1058
Metaclasses Are Subclasses of Type---1061
Class Statement Protocol--1061
Declaring Metaclasses---1062
Coding Metaclasses--1063
A Basic Metaclass--1064

Customizing Construction and Initialization--1065
Other Metaclass Coding Techniques--1065
Instances Versus Inheritance--1068
Example: Adding Methods to Classes---1070
Manual Augmentation---1070
Metaclass-Based Augmentation---1071
Metaclasses Versus Class Decorators: Round 2---1073
Example: Applying Decorators to Methods---1076
Tracing with Decoration Manually--1076
Tracing with Metaclasses and Decorators---1077
Applying Any Decorator to Methods--1079
Metaclasses Versus Class Decorators: Round 3---1080
Chapter Summary---1084
Test Your Knowledge: Quiz---1084
Test Your Knowledge: Answers---1085
Part IX Appendixes
A. Installation and Configuration--1089
B. Solutions to End-of-Part Exercises---1101
Index---1139

***learning_the_vi_and_vim_editors_seventh_edition_2008_494p

Preface--xiii
Part I. Basic and Advanced vi
1. The vi Text Editor--3
A Brief Historical Perspective---5
Opening and Closing Files---6
Quitting Without Saving Edits---10
2. Simple Editing---13
vi Commands--13
Moving the Cursor---14
Simple Edits--18
More Ways to Insert Text--30
Joining Two Lines with J--31
Review of Basic vi Commands---32
3. Moving Around in a Hurry--35
Movement by Screens--35
Movement by Text Blocks---38
Movement by Searches--39
Movement by Line Number--43
Review of vi Motion Commands--44
4. Beyond the Basics---47
More Command Combinations--47
Options When Starting vi---48
Making Use of Buffers---51
Marking Your Place--52
Other Advanced Edits--53
Review of vi Buffer and Marking Commands---53
5. Introducing the ex Editor--55
ex Commands--55

Editing with ex--58
Saving and Exiting Files--63
Copying a File into Another File--65
Editing Multiple Files---65
6. Global Replacement---71
Confirming Substitutions--72
Context-Sensitive Replacement--73
Pattern-Matching Rules--74
Pattern-Matching Examples--81
A Final Look at Pattern Matching---89
7. Advanced Editing---95
Customizing vi--95
Executing Unix Commands--99
Saving Commands---103
Using ex Scripts--114
Editing Program Source Code---120
8. Introduction to the vi Clones--125
And These Are My Brothers, Darrell, Darrell, and Darrell------------------------------------125
Multiwindow Editing---126
GUI Interfaces---127
Extended Regular Expressions--128
Enhanced Tags---129
Improved Facilities---134
Programming Assistance---138
Editor Comparison Summary--140
Nothing Like the Original--141
A Look Ahead--141
Part II. Vim
9. Vim (vi Improved): An Introduction---145
Overview--146
Where to Get Vim--150
Getting Vim for Unix and GNU/Linux--151
Getting Vim for Windows Environments---156
Getting Vim for the Macintosh Environment--157
Other Operating Systems--157
Aids and Easy Modes for New Users---157
Summary---158
10. Major Vim Improvements over vi---159
Built-in Help---159
Startup and Initialization Options--160
New Motion Commands--167
Extended Regular Expressions--169
Customizing the Executable--171
11. Multiple Windows in Vim---173
Initiating Multiwindow Editing---174
Opening Windows--177
Moving Around Windows (Getting Your Cursor from Here to There)----------------------180
Moving Windows Around---181
Resizing Windows--183
Buffers and Their Interaction with Windows--186
Playing Tag with Windows--190
Tabbed Editing---191

Closing and Quitting Windows--192
Summary---193
12. Vim Scripts--195
What’s Your Favorite Color (Scheme)?---195
Dynamic File Type Configuration Through Scripting---205
Some Additional Thoughts About Vim Scripting---213
Resources---218
13. Graphical Vim (gvim)---219
General Introduction to gvim--220
Customizing Scrollbars, Menus, and Toolbars---225
gvim in Microsoft Windows--236
gvim in the X Window System--237
GUI Options and Command Synopsis--237
14. Vim Enhancements for Programmers---239
Folding and Outlining (Outline Mode)--240
Auto and Smart Indenting--251
Keyword and Dictionary Word Completion---259
Tag Stacking---268
Syntax Highlighting--270
Compiling and Checking Errors with Vim---279
Some Final Thoughts on Vim for Writing Programs---284
15. Other Cool Stuff in Vim--285
Editing Binary Files--285
Digraphs: Non-ASCII Characters--287
Editing Files in Other Places--289
Navigating and Changing Directories---290
Backups with Vim---292
HTML Your Text---293
What’s the Difference?--294
Undoing Undos--296
Now, Where Was I?--297
What’s My Line (Size)?---300
Abbreviations of Vim Commands and Options---302
A Few Quickies (Not Necessarily Vim-Specific)--303
More Resources---304
Part III. Other vi Clones
16. nvi: New vi---307
Author and History--307
Important Command-Line Arguments---308
Online Help and Other Documentation--309
Initialization--309
Multiwindow Editing---310
GUI Interfaces--311
Extended Regular Expressions--311
Improvements for Editing--312
Programming Assistance---315
Interesting Features---315
Sources and Supported Operating Systems--315
17. Elvis---317
Author and History--317
Important Command-Line Arguments---317
Online Help and Other Documentation--319

Initialization--319
Multiwindow Editing---320
GUI Interfaces---323
Extended Regular Expressions--328
Improved Editing Facilities--328
Programming Assistance---332
Interesting Features---335
elvis Futures--340
Sources and Supported Operating Systems--340
18. vile: vi Like Emacs---343
Authors and History---343
Important Command-Line Arguments--344
Online Help and Other Documentation---345
Initialization--346
Multiwindow Editing---347
GUI Interfaces---349
Extended Regular Expressions--357
Improved Editing Facilities--359
Programming Assistance---365
Interesting Features---368
Sources and Supported Operating Systems--374
Part IV Appendixes
A. The vi, ex, and Vim Editors---377
B. Setting Options--415
C. Problem Checklists--431
D. vi and the Internet---435
Index---447

***Learning_to_Program_Python_2006_299p

Learning to program---1
What do I need to be a programmer?--7
What is programming?---9
Getting Started---16
Simple sequences--20
Data---26
More Sequences--56
Looping the loop--62
A Little Bit of Style---70
Input--77
Conditionals--86
Functions and Modules--95
File Handling--110
Text Handling---125
Error Handling--134
Namespaces--143
Regular Expressions--149
Classes--157
Event Driven Programs---178
Introduction to GUI Programming--184
Recursion---201
Introduction to Functional Programming--205

A Case Study--215
Python in Practice--233
Working with Databases---235
Working with the Operating System---258
Inter-Process Communications--284
References---292

***Learn_Python_The_Hard_Way_2010_165p

The Hard Way Is Easier---3
Reading and Writing--3
Attention to Detail--3
Spotting Differences--3
Do Not Copy-Paste---4
A Note On Practice And Persistence--4
License---4
Exercise 0: The Setup--5
Mac OSX--5
Windows--6
Linux---7
Warnings For Beginners---8
Exercise 1: A Good First Program---11
What You Should See--11
Extra Credit--12
Exercise 2: Comments And Pound Characters-------------------------------------13
What You Should See--13
Extra Credit--13
Exercise 3: Numbers And Math--15
What You Should See--16
Extra Credit--16
Exercise 4: Variables And Names---17
What You Should See--17
Extra Credit--18
Exercise 5: More Variables And Printing---19
What You Should See--19
Extra Credit--20
Exercise 6: Strings And Text--21
What You Should See--21
Extra Credit--22
Exercise 7: More Printing--23
What You Should See--23
Exercise 8: Printing, Printing--25
What You Should See--25
Extra Credit--25
Exercise 9: Printing, Printing, Printing--27
What You Should See--27
Extra Credit--27
Exercise 10: What Was That?---29
What You Should See--29
Extra Credit--30
Exercise 11: Asking Questions---31
What You Should See--31

Extra Credit--31
Exercise 12: Prompting People---33
What You Should See--33
Extra Credit--33
Exercise 13: Parameters, Unpacking, Variables--35
Hold Up! Features Have Another Name---35
What You Should See--36
Extra Credit--36
Exercise 14: Prompting And Passing---37
What You Should See--37
Extra Credit--38
Exercise 15: Reading Files---39
What You Should See--40
Extra Credit--40
Exercise 16: Reading And Writing Files--41
What You Should See--42
Extra Credit--42
Exercise 17: More Files--43
What You Should See--43
Extra Credit--44
Exercise 18: Names, Variables, Code, Functions---45
What You Should See--46
Extra Credit--46
Exercise 19: Functions And Variables--49
What You Should See--49
Extra Credit--50
Exercise 20: Functions And Files--51
What You Should See--51
Extra Credit--52
Exercise 21: Functions Can Return Something--53
What You Should See--54
Extra Credit--54
Exercise 22: What Do You Know So Far?---55
What You are Learning---55
Exercise 23: Read Some Code---57
Exercise 24: More Practice---59
What You Should See--60
Extra Credit--60
Exercise 25: Even More Practice---61
What You Should See--62
Extra Credit--63
Exercise 26: Congratulations, Take A Test!--65
Exercise 27: Memorizing Logic---67
The Truth Terms---67
The Truth Tables---68
Exercise 28: Boolean Practice--69
What You Should See--70
Extra Credit--71
Exercise 29: What If---73
What You Should See--73
Extra Credit--74
Exercise 30: Else And If--75

What You Should See--76
Extra Credit--76
Exercise 31: Making Decisions--77
What You Should See--78
Extra Credit--79
Exercise 32: Loops And Lists--81
What You Should See--82
Extra Credit--82
Exercise 33: While Loops---85
What You Should See--86
Extra Credit--86
Exercise 34: Accessing Elements Of Lists--87
Extra Credit--88
Exercise 35: Branches and Functions---89
What You Should See--90
Extra Credit--91
Exercise 36: Designing and Debugging--93
Rules For If-Statements---93
Rules For Loops--93
Tips For Debugging---93
Homework---94
Exercise 37: Symbol Review--95
Keywords---95
Data Types---96
String Escapes Sequences---96
String Formats---97
Operators--97
Exercise 38: Reading Code---99
Extra Credit--99
Exercise 39: Doing Things To Lists---101
What You Should See---102
Extra Credit---103
Exercise 40: Dictionaries, Oh Lovely Dictionaries--105
What You Should See---106
Extra Credit---107
Exercise 41: A Room With A View Of A Bear With A Broadsword----------------------------------109
What You Should See---112
Extra Credit---113
Exercise 42: Getting Classy---115
What You Should See---118
Extra Credit---119
Exercise 43: You Make A Game---121
Exercise 44: Evaluating Your Game---123
Function Style---123
Class Style--123
Code Style---124
Good Comments--124
Evaluate Your Game---124
Exercise 45: Is-A, Has-A, Objects, and Classes--127
How This Looks In Code---128
Exercise 46: A Project Skeleton---131
Skeleton Contents: Linux/OSX---131

Testing Your Setup--132
Using The Skeleton--133
Required Quiz--133
Exercise 47: Automated Testing---135
Writing A Test Case---135
Testing Guidelines---136
What You Should See--137
Extra Credit--137
Exercise 48: Advanced User Input--139
Our Game Lexicon---139
What You Should Test--141
Design Hints--142
Extra Credit--142
Exercise 49: Making Sentences---143
Match And Peek--143
The Sentence Grammar---144
A Word On Exceptions--146
What You Should Test--146
Extra Credit--146
Exercise 50: Your First Work Assignment--147
Review What You Know--147
Implementing A Feature List---147
The Feature List--148
Tips On Working The List--148
Exercise 51: Reviewing Your Game--149
How To Study A User---149
Implement The Changes---149
Exercise 52: Teach Someone Else What You Know---151
Extra Credit--151
Next Steps--153
Advice From An Old Programmer---155
Indices and tables--157

***making_games_PYTHON_2012_365p

Who is this book for?--i
About This Book--ii

Chapter 1 – Installing Python and Pygame--1
What You Should Know Before You Begin---1
Downloading and Installing Python--1
Windows Instructions--1
Mac OS X Instructions--2
Ubuntu and Linux Instructions---2
Starting Python--2
Installing Pygame--3
How to Use This Book---4
The Featured Programs--4
Downloading Graphics and Sound Files--4
Line Numbers and Spaces--4
Text Wrapping in This Book--5
Checking Your Code Online--6

More Info Links on http://invpy.com--6

Chapter 2 – Pygame Basics---7
GUI vs. CLI---7
Source Code for Hello World with Pygame---7
Setting Up a Pygame Program--8
Game Loops and Game States---10
pygame.event.Event Objects---11
The QUIT Event and pygame.quit() Function--12
Pixel Coordinates--13
A Reminder About Functions, Methods, Constructor Functions, and Functions in Modules (and the Difference Between Them)--14
Surface Objects and The Window--15
Colors--16
Transparent Colors---17
pygame.Color Objects--18
Rect Objects--18
Primitive Drawing Functions---20
pygame.PixelArray Objects---23
The pygame.display.update() Function--24
Animation---24
Frames Per Second and pygame.time.Clock Objects---27
Drawing Images with pygame.image.load() and blit()--28
Fonts--28
Anti-Aliasing--30
Playing Sounds---31
Summary--32

Chapter 3 – Memory Puzzle--33
How to Play Memory Puzzle---33
Nested for Loops---33
Source Code of Memory Puzzle--34
Credits and Imports--42
Magic Numbers are Bad---42
Sanity Checks with assert Statements--43
Telling If a Number is Even or Odd--44
Crash Early and Crash Often!---44
Making the Source Code Look Pretty---45
Using Constant Variables Instead of Strings--46
Making Sure We Have Enough Icons---47
Tuples vs. Lists, Immutable vs. Mutable--47
One Item Tuples Need a Trailing Comma---48
Converting Between Lists and Tuples---49
The global statement, and Why Global Variables are Evil--49
Data Structures and 2D Lists--51
The ―Start Game‖ Animation--52
The Game Loop---52
The Event Handling Loop--53
Checking Which Box The Mouse Cursor is Over---54
Handling the First Clicked Box---55
Handling a Mismatched Pair of Icons---56
Handling If the Player Won--56

Drawing the Game State to the Screen--57
Creating the ―Revealed Boxes‖ Data Structure---58
Creating the Board Data Structure: Step 1 – Get All Possible Icons--------------------------------58
Step 2 – Shuffling and Truncating the List of All Icons--59
Step 3 – Placing the Icons on the Board--59
Splitting a List into a List of Lists---60
Different Coordinate Systems---61
Converting from Pixel Coordinates to Box Coordinates--62
Drawing the Icon, and Syntactic Sugar---63
Syntactic Sugar with Getting a Board Space's Icon's Shape and Color----------------------------64
Drawing the Box Cover--64
Handling the Revealing and Covering Animation--65
Drawing the Entire Board--66
Drawing the Highlight--67
The ―Start Game‖ Animation---67
Revealing and Covering the Groups of Boxes--68
The ―Game Won‖ Animation---68
Telling if the Player Has Won---69
Why Bother Having a main() Function?---69
Why Bother With Readability?--70
Summary, and a Hacking Suggestion--74

Chapter 4 – Slide Puzzle---77
How to Play Slide Puzzle---77
Source Code to Slide Puzzle---77
Second Verse, Same as the First---85
Setting Up the Buttons---86
Being Smart By Using Stupid Code--87
The Main Game Loop---88
Clicking on the Buttons---89
Sliding Tiles with the Mouse--90
Sliding Tiles with the Keyboard---90
―Equal To One Of‖ Trick with the in Operator--91
WASD and Arrow Keys--91
Actually Performing the Tile Slide--92
IDLE and Terminating Pygame Programs---92
Checking for a Specific Event, and Posting Events to Pygame's Event Queue--------------------92
Creating the Board Data Structure---93
Not Tracking the Blank Position--94
Making a Move by Updating the Board Data Structure---94
When NOT to Use an Assertion--95
Getting a Not-So-Random Move---96
Converting Tile Coordinates to Pixel Coordinates---97
Converting from Pixel Coordinates to Board Coordinates--97
Drawing a Tile---97
The Making Text Appear on the Screen---98
Drawing the Board--99
Drawing the Border of the Board---99
Drawing the Buttons--100
Animating the Tile Slides---100
The copy() Surface Method--101
Creating a New Puzzle---103

Animating the Board Reset--104
Time vs. Memory Tradeoffs--105
Nobody Cares About a Few Bytes---106
Nobody Cares About a Few Million Nanoseconds--107
Summary---107

Chapter 5 – Simulate--108
How to Play Simulate---108
Source Code to Simulate--108
The Usual Starting Stuff---114
Setting Up the Buttons--115
The main() Function---115
Some Local Variables Used in This Program--116
Drawing the Board and Handling Input--117
Checking for Mouse Clicks---118
Checking for Keyboard Presses---118
The Two States of the Game Loop--119
Figuring Out if the Player Pressed the Right Buttons---119
Epoch Time---121
Drawing the Board to the Screen---122
Same Old terminate() Function---122
Reusing The Constant Variables--123
Animating the Button Flash--123
Drawing the Buttons---126
Animating the Background Change--126
The Game Over Animation--127
Converting from Pixel Coordinates to Buttons---129
Explicit is Better Than Implicit--129

Chapter 6 – Wormy---131
How to Play Wormy--131
Source Code to Wormy---131
The Grid--137
The Setup Code--137
The main() Function--138
A Separate runGame() Function--139
The Event Handling Loop---139
Collision Detection--140
Detecting Collisions with the Apple---141
Moving the Worm--142
The insert() List Method--142
Drawing the Screen---143
Drawing ―Press a key‖ Text to the Screen--143
The checkForKeyPress() Function---143
The Start Screen---144
Rotating the Start Screen Text---145
Rotations Are Not Perfect---146
Deciding Where the Apple Appears---147
Game Over Screens---147
Drawing Functions---148
Don’t Reuse Variable Names---151

Chapter 7 - Tetromino--153
How to Play Tetromino---153
Some Tetromino Nomenclature---153
Source Code to Tetromino--154
The Usual Setup Code--166
Setting up Timing Constants for Holding Down Keys---166
More Setup Code--166
Setting Up the Piece Templates--168
Splitting a ―Line of Code‖ Across Multiple Lines---171
The main() Function--172
The Start of a New Game--173
The Game Loop---174
The Event Handling Loop--174
Pausing the Game---174
Using Movement Variables to Handle User Input--175
Checking if a Slide or Rotation is Valid--175
Finding the Bottom--178
Moving by Holding Down the Key--179
Letting the Piece ―Naturally‖ Fall---182
Drawing Everything on the Screen--182
makeTextObjs(), A Shortcut Function for Making Text--183
The Same Old terminate() Function---183
Waiting for a Key Press Event with the checkForKeyPress() Function---------------------------183
showTextScreen(), A Generic Text Screen Function--184
The checkForQuit() Function--185
The calculateLevelAndFallFreq() Function---185
Generating Pieces with the getNewPiece() Function---188
Adding Pieces to the Board Data Structure---189
Creating a New Board Data Structure--189
The isOnBoard() and isValidPosition() Functions--190
Checking for, and Removing, Complete Lines--192
Convert from Board Coordinates to Pixel Coordinates--195
Drawing a Box on the Board or Elsewhere on the Screen--195
Drawing Everything to the Screen--196
Drawing the Score and Level Text--196
Drawing a Piece on the Board or Elsewhere on the Screen---197
Drawing the ―Next‖ Piece--197
Summary---198

Chapter 8 – Squirrel Eat Squirrel--200
How to Play Squirrel Eat Squirrel--200
The Design of Squirrel Eat Squirrel--200
Source Code to Squirrel Eat Squirrel--201
The Usual Setup Code--211
Describing the Data Structures---212
The main() Function---213
The pygame.transform.flip() Function--214
A More Detailed Game State than Usual---214
The Usual Text Creation Code---215
Cameras---215
The ―Active Area‖---217
Keeping Track of the Location of Things in the Game World-------------------------------------218

Starting Off with Some Grass--219
The Game Loop--219
Checking to Disable Invulnerability---219
Moving the Enemy Squirrels---219
Removing the Far Away Grass and Squirrel Objects---221
When Deleting Items in a List, Iterate Over the List in Reverse-----------------------------------221
Adding New Grass and Squirrel Objects---223
Camera Slack, and Moving the Camera View---223
Drawing the Background, Grass, Squirrels, and Health Meter---------------------------------224
The Event Handling Loop--226
Moving the Player, and Accounting for Bounce---228
Collision Detection: Eat or Be Eaten---229
The Game Over Screen--231
Winning---232
Drawing a Graphical Health Meter---232
The Same Old terminate() Function---232
The Mathematics of the Sine Function--233
Backwards Compatibility with Python Version 2--236
The getRandomVelocity() Function--237
Finding a Place to Add New Squirrels and Grass---237
Creating Enemy Squirrel Data Structures---238
Flipping the Squirrel Image--239
Creating Grass Data Structures--239
Checking if Outside the Active Area---240
Summary---241

Chapter 9 – Star Pusher---242
How to Play Star Pusher---242
Source Code to Star Pusher---242
The Initial Setup--256
Data Structures in Star Pusher--271
The ―Game State‖ Data Structure---271
The ―Map‖ Data Structure---271
The ―Levels‖ Data Structure---272
Reading and Writing Text Files--272
Text Files and Binary Files---272
Writing to Files--273
Reading from Files---274
About the Star Pusher Map File Format---274
Recursive Functions---280
Stack Overflows--281
Preventing Stack Overflows with a Base Case--283
The Flood Fill Algorithm--284
Drawing the Map---285
Checking if the Level is Finished--287
Summary---288

Chapter 10 – Four Extra Games---289
Flippy, an ―Othello‖ Clone--290
Source Code for Flippy--292
Ink Spill, a ―Flood It‖ Clone--305
Source Code for Ink Spill---305

Four-In-A-Row, a ―Connect Four‖ Clone--317
Source Code for Four-In-A-Row--317
Gemgem, a ―Bejeweled‖ Clone--327
Source Code for Gemgem---327
Summary---340
Glossary--342
About the Author---347

***Making_Use_of_Python_2002_416p

Introduction--xi
Scenario--xxiii
Chapter 1 An Introduction to Python---1
Getting Started---1
Understanding Requirements--2
Determine Requirements of the University---2
Obtain Python and Its Documentation--3
Determine the System Requirements--4
Install Python---5
Start Python in Different Execution Modes---7
Summary---12

Chapter 2 Getting Started with Python--13
Getting Started---14
Writing Your First Python Program--14
Comments--15
Python as a Calculator--16
Using Variables in Python--16
Variables--17
Assigning Values to Variables--18
Standard Types---19
Identifiers and Keywords--39
Memory Management--40
Create a Sequence to Store All the Names of the Students-------------------------------------42
Write the Code to Display the Names of the Students---42
Declare a Dictionary of Student Purchases with the Names of the Students as the Key----------43
Write the Code to Display the Student Purchases---43
Save and Execute the Code---43
Verify the Details---44
Summary---44

Chapter 3 Intrinsic Operations and Input/Output--47
Getting Started---48
Using Input/Output Features and Intrinsic Operations
for Data Types in Python--48
Identify the Variables to Be Used--49
Accepting User Input--49
Formatting the Output---50
Introduction to Intrinsic Operations--55
Intrinsic Operations for Numeric Data Types--57
Intrinsic Operations for Strings---60
Intrinsic Operations for Lists and Tuples---66

Write the Code--71
Execute the Code---71
Summary---73

Chapter 4 Programming Basics---75
Getting Started---76
Conditional Operators---76
Order of Precedence of Operators---82
Using Programming Constructs---83
Identify the Control and Loop Statements to Be Used--84
Write the Code--94
Execute the Code---95
Summary---97

Chapter 5 Functions--99
Getting Started---100
Using Functions--100
Functions---101
Scope of Variables--118
Identify the Functions to Be Used---119
Write the Code--119
Execute the Code---121
Summary--122

Chapter 6 Modules---123
Getting Started---124
Using Modules---124
Modules---124
Packages--135
Identify the Modules to Be Used---136
Write the Code--137
Execute the Code---139
Summary--140

Chapter 7 Files--141
Getting Started---141
Using File Objects---142
Identify the Functions and Methods to Be Used--142
Write the Code to Store Course Details to the File---154
Execute the Code---155
Verify the Solution--155
Summary--156

Chapter 8 Object-Oriented Programming--157
Getting Started---158
Introducing OOP--158
Components of OOP--159
Benefits of OOP--160
Using Classes--161
Identify the Classes to Be Defined---162
Identifying the Class Objects---163
Identifying the Classes to Be Inherited and Their Objects-----------------------------------170

Identify the Methods to Be Overridden---173
Write the Code---182
Execute the Code--189
Summary---190

Chapter 9 Exception Handling---193
Getting Started---193
Handling Exceptions--194
Identify the Type of Error and Where the Error Occurs---196
Identify the Mechanism of Trapping the Exception--200
Identify the Location for the Code for Handling the Exception to Be Written-------------------209
Write the Code for Handling the Exception---209
Save and Execute the Code---210
Summary---210

Chapter 10 CGI Programming---213
Getting Started---213
Internet Basics---214
World Wide Web--217
Web Browsers--217
Hypertext Transfer Protocol (HTTP)--220
Revising HTML--221
Client-Side versus Server-Side Scripting--227
An Introduction to CGI--229
Writing CGI Applications--231
Write the Code for the HTML Form to Accept Data from the User-------------------------------231
Write the CGI Program in Python to Generate the Results Page----------------------------------232
Write the CGI Program to Generate Both the Form and Results Pages--------------------------236
Execute the Code--237
Summary---239

Chapter 11 Database Programming---241
Getting Started---241
Database Management---242
Introduction to MySQL--243
Working with MySQL--246
Accessing a Database from a Python Script--254
Identify the Elements of the Table That Stores Registration Details-------------------------------256
Identify the Steps for Connecting to the Database---256
Write the Code to Create a Table in the Database--259
Write the Code to Insert the Registration Details into the Table Created-------------------------260
Execute the Code to Create the Table in the Database--261
Execute the Code to Insert Data into the Table--261
Verify the Data in the Database--263
Summary---264

Chapter 12 Network Programming---267
Getting Started---267
Client/Server Architecture--268
Network Programming--269
Using Sockets--272
Identify the Sockets to Be Used--272

Write the Code to Run on the IT Department Computer--287
Write the Code to Run on the Admission Office Computer---288
Execute the Code Created for the IT Department Computer--289
Execute the Code Created for the Admission Office Computer------------------------------------290
Verify that Data Has Been Saved to a File in--the
IT Department Computer--292
Summary--292

Chapter 13 Multithreaded Programming--297
Getting Started--297
Single-Threaded Applications---298
Threading in Python---299
Creating Multithreaded Applications--300
Identify the Class and the Methods to Create a Multithreaded Application-----------------------300
Write Code for the Server--308
Write the Code for the Client--309
Execute the Code Created for the Server---310
Execute the Code Created for the Client--311
Summary--313

Chapter 14 Advanced Web Programming--315
Getting Started--316
Creating Web Servers---316
Accessing URLs--323
Creating Advanced CGI Applications---328
Identify the Elements of the Web Page for EnteringAssignment Details and Uploading the File-------328
Identify the Methodology for Uploading the File---329
Identify the Methodology for Storing User Information--330
Write the Code for the CGI Script---335
Execute the CGI Script---339
Summary--340

Chapter 15 GUI Programming with Tkinter---343
Getting Started--343
Introduction to Tkinter--344
Creating a GUI Application---347
Identify the Components of the User Interface---348
Identify the Tkinter Widgets to Design the User Interface---348
Write the Code for the User Interface---360
Execute the Code---362
Summary--364
Appendix A Distributing COM Objects--365
Basics of COM---365
The Binary Standard---367
COM Interfaces---369
Binding---370
Python and COM---371
Creating COM Clients---371
Creating COM Servers--373
Index--377

***Matplotlib_for_Python_Developers_2009_307p

Preface---1
Chapter 1: Introduction to Matplotlib---7
Merits of Matplotlib--8
Matplotlib web sites and online documentation---10
Output formats and backends--10
Output formats---11
Backends---12
About dependencies---13
Build dependencies--15
Installing Matplotlib--15
Installing Matplotlib on Linux---15
Installing Matplotlib on Windows--16
Installing Matplotlib on Mac OS X---16
Installing Matplotlib using packaged Python distributions-----------------------------------17
Installing Matplotlib from source code--17
Testing our installation---18
Summary---19
Chapter 2: Getting Started with Matplotlib---21
First plots with Matplotlib---21
Multiline plots--25
A brief introduction to NumPy arrays---27
Grid, axes, and labels---28
Adding a grid---28
Handling axes---29
Adding labels---31
Titles and legends--32
Adding a title---32
Adding a legend--33
A complete example--35
Saving plots to a file--36
Interactive navigation toolbar--38
IPython support--40
Controlling the interactive mode--42
Suppressing functions output--43
Configuring Matplotlib---43
Configuration files---44
Configuring through the Python code---45
Selecting backend from code---46
Summary---47

Chapter 3: Decorate Graphs with Plot Styles and Types---49
Markers and line styles---49
Control colors---50
Specifying styles in multiline plots---52
Control line styles--52
Control marker styles--53
Finer control with keyword arguments---56
Handling X and Y ticks--58
Plot types---59
Histogram charts---59
Error bar charts---61

Bar charts--63
Pie charts--67
Scatter plots--69
Polar charts--71
Navigation Toolbar with polar plots--73
Control radial and angular grids--73
Text inside figure, annotations, and arrows--74
Text inside figure--74
Annotations--75
Arrows--77
Summary--79

Chapter 4: Advanced Matplotlib---81
Object-oriented versus MATLAB styles---81
A brief introduction to Matplotlib objects---85
Our first (simple) example of OO Matplotlib--85
Subplots--86
Multiple figures--88
Additional Y (or X) axes--89
Logarithmic axes--91
Share axes--92
Plotting dates--94
Date formatting--95
Axes formatting with axes tick locators and formatters---96
Custom formatters and locators---99
Text properties, fonts, and LaTeX---99
Fonts--101
Using LaTeX formatting---102
Mathtext--103
External TeX renderer---104
Contour plots and image plotting--106
Contour plots--106
Image plotting--109
Summary--111

Chapter 5: Embedding Matplotlib in GTK--113
A brief introduction to GTK--113
Introduction to GTK signal system--115
Embedding a Matplotlib figure in a GTK window--116
Including a navigation toolbar--119
Real-time plots update--123
Embedding Matplotlib in a Glade application--132
Designing the GUI using Glade--132
Code to use Glade GUI---135
Summary--144

Chapter 6: Embedding Matplotlib in Qt 4---145
Brief introduction to Qt 4 and PyQt4--145
Embedding a Matplotlib figure in a Qt window--147
Including a navigation toolbar--151
Real-time update of a Matplotlib graph--156
Embedding Matplotlib in a GUI made with Qt Designer---165

Designing the GUI using Qt Designer--165
Code to use the Qt Designer GUI--168
Introduction to signals and slots---171
Returning to the example---172
Summary--179

Chapter 7: Embedding Matplotlib in wxWidgets---181
Brief introduction to wxWidgets and wxPython---181
Embedding a Matplotlib figure in a wxFrame---182
Including a navigation toolbar--186
Real-time plots update--192
Embedding Matplotlib in a GUI made with wxGlade--203
Summary--213

Chapter 8: Matplotlib for the Web--215
Matplotlib and CGI--216
What is CGI--216
Configuring Apache for CGI execution--216
Simple CGI example--218
Matplotlib in a CGI script---219
Passing parameters to a CGI script---220
Matplotlib and mod_python--223
What is mod_python--223
Apache configuration for mod_python---224
Matplotlib in a mod_python example---226
Matplotlib and mod_python's Python Server Pages--228
Web Frameworks and MVC---231
Matplotlib and Django--232
What is Django---232
Matplotlib in a Django application--233
Matplotlib and Pylons--237
What is Pylons---237
Matplotlib in a Pylons application--238
Summary--242

Chapter 9: Matplotlib in the Real World---243
Plotting data from a database---244
Plotting data from the Web---247
Plotting data by parsing an Apache log file--250
Plotting data from a CSV file---256
Plotting extrapolated data using curve fitting--261
Tools using Matplotlib--267
NetworkX---267
Mpmath--269
Plotting geographical data--271
First example---272
Using satellite background---274
Plot data over a map--275
Plotting shapefiles with Basemap---277
Summary--279
Index---281

***Mobile_Python_Rapid_Prototyping_of_Applications_on_the_Mobile_Platform_2007_329p

Contributors--ix
About the Authors--xi
Authors' Acknowledgments---xiii
Symbian Press Acknowledgments---xvii
Forewords--xix
1 Introduction and Basics--1
1.1 Why Does Python Make a Difference?---3
1.2 How to Use this Book---4
1.3 Who Is this Book For?---5
1.4 What Are Symbian OS, S60 and Python for S60?---------------------------------------6
1.5 Python Terminology in this Book--7
1.6 Democratizing Innovation on the Mobile Platform-------------------------------------8
1.7 The Process of Rapid Prototyping with Python S60-----------------------------------10
1.8 Summary--11

2 Getting Started---13
2.1 Installing Python for S60 on 3rd Edition Devices-------------------------------------14
2.2 Installing Python for S60 on 2nd Edition Devices-------------------------------------21
2.3 Writing a Program in Python for S60--27
2.4 White Space in Python Code---28
2.5 Troubleshooting--29
2.6 Summary--30

3 Graphical User Interface Basics---31
3.1 Using Modules---31
3.2 Native UI Elements – Dialogs, Menus and Selection Lists--------------------------------32
3.3 Messages--45
3.4 Summary--47

4 Application Building and SMS Inbox---49
4.1 Functions--49
4.2 Application Structure---52
4.3 String Handling--60
4.4 SMS Inbox---64
4.5 SMS Game Server--70
4.6 Summary--76

5 Sound, Interactive Graphics and Camera--77
5.1 Sound---77
5.2 Keyboard Keys--84
5.3 Graphics--92
5.4 Camera---100
5.5 Mobile Game: UFO Zapper---104
5.6 Summary---110

6 Data Handling---111
6.1 File Basics---112
6.2 Reading and Writing Text--117
6.3 Local Database---121

6.4 GSM and GPS Positioning--123
6.5 Vocabulector: A Language-Learning Tool---127
6.6 Summary---131

7 Bluetooth and Telephone Functionality--133
7.1 Bluetooth Pairing---134
7.2 OBEX and RFCOMM--134
7.3 Phone-to-Phone Communication--136
7.4 Phone-to-PC Communication---141
7.5 Communication with GPS and Other Devices--148
7.6 Telephone Functionality and Contacts---151
7.7 System Information--152
7.8 Summary---154

8 Mobile Networking---155
8.1 Simple Web Tasks--156
8.2 Setting up the Development Environment--158
8.3 Communication Protocols--166
8.4 Server Software---172
8.5 Pushing Data to a Phone--177
8.6 Peer-to-Peer Networking--183
8.7 Using a Phone as a Web Service---193
8.8 Summary---197

9 Web Services--199
9.1 Basic Principles---200
9.2 MopyMaps! Mobile Yahoo! Maps--201
9.3 EventFu: Finding Eventful Events---207
9.4 InstaFlickr: Shoot and Upload Photos to Flickr---215
9.5 Summary---224

10 Effective Python for S60---227
10.1 Powerful Language Constructs---227
10.2 Introspection--231
10.3 Custom Modules and Automatic Updating--234
10.4 Program Patterns---239
10.5 Summary--241

11 Combining Art and Engineering---245
11.1 MobiLenin--245
11.2 Manhattan Story Mashup--252
11.3 MobileArtBlog – Image-Composition Tool--256
11.4 ArduinoBT Micro-Controller Board--261
11.5 Controlling Max/MSP with a Phone--266
11.6 OpenSound Control---273
11.7 Robotics---274
11.8 Summary--277

Appendix A: Platform Security---279
Appendix B: Bluetooth Console---289
Appendix C: Debugging---295
Appendix D: How to Use the Emulator---301

References---303
Glossary---305
Examples---309
Python Language Lessons---315
Python for S60 Modules---317
Index--321

***Natural_Language_Processing_with_Python_2009_491p

Preface--ix
1. Language Processing and Python---1
1.1 Computing with Language: Texts and Words--1
1.2 A Closer Look at Python: Texts as Lists of Words----------------------------------10
1.3 Computing with Language: Simple Statistics--16
1.4 Back to Python: Making Decisions and Taking Control----------------------------22
1.5 Automatic Natural Language Understanding---27
1.6 Summary--33
1.7 Further Reading---34
1.8 Exercises--35

2. Accessing Text Corpora and Lexical Resources---39
2.1 Accessing Text Corpora---39
2.2 Conditional Frequency Distributions---52
2.3 More Python: Reusing Code---56
2.4 Lexical Resources--59
2.5 WordNet--67
2.6 Summary--73
2.7 Further Reading---73
2.8 Exercises--74

3. Processing Raw Text---79
3.1 Accessing Text from the Web and from Disk--80
3.2 Strings: Text Processing at the Lowest Level---87
3.3 Text Processing with Unicode---93
3.4 Regular Expressions for Detecting Word Patterns-----------------------------------97
3.5 Useful Applications of Regular Expressions--102
3.6 Normalizing Text---107
3.7 Regular Expressions for Tokenizing Text--109
3.8 Segmentation---112
3.9 Formatting: From Lists to Strings--116
3.10 Summary--121
3.11 Further Reading---122
3.12 Exercises--123

4. Writing Structured Programs---129
4.1 Back to the Basics---130
4.2 Sequences---133
4.3 Questions of Style---138
4.4 Functions: The Foundation of Structured Programming---------------------------142
4.5 Doing More with Functions---149
4.6 Program Development---154

4.7 Algorithm Design--160
4.8 A Sample of Python Libraries--167
4.9 Summary---172
4.10 Further Reading---173
4.11 Exercises---173

5. Categorizing and Tagging Words---179
5.1 Using a Tagger--179
5.2 Tagged Corpora--181
5.3 Mapping Words to Properties Using Python Dictionaries---------------------------189
5.4 Automatic Tagging---198
5.5 N-Gram Tagging---202
5.6 Transformation-Based Tagging---208
5.7 How to Determine the Category of a Word---210
5.8 Summary---213
5.9 Further Reading---214
5.10 Exercises---215

6. Learning to Classify Text--221
6.1 Supervised Classification---221
6.2 Further Examples of Supervised Classification--------------------------------------233
6.3 Evaluation--237
6.4 Decision Trees--242
6.5 Naive Bayes Classifiers---245
6.6 Maximum Entropy Classifiers---250
6.7 Modeling Linguistic Patterns---254
6.8 Summary---256
6.9 Further Reading---256
6.10 Exercises---257

7. Extracting Information from Text---261
7.1 Information Extraction--261
7.2 Chunking--264
7.3 Developing and Evaluating Chunkers---270
7.4 Recursion in Linguistic Structure--277
7.5 Named Entity Recognition---281
7.6 Relation Extraction---284
7.7 Summary---285
7.8 Further Reading---286
7.9 Exercises--286

8. Analyzing Sentence Structure---291
8.1 Some Grammatical Dilemmas---292
8.2 What’s the Use of Syntax?---295
8.3 Context-Free Grammar---298
8.4 Parsing with Context-Free Grammar---302
8.5 Dependencies and Dependency Grammar---310
8.6 Grammar Development---315
8.7 Summary---321
8.8 Further Reading---322
8.9 Exercises--322

9. Building Feature-Based Grammars---327
9.1 Grammatical Features---327
9.2 Processing Feature Structures--337
9.3 Extending a Feature-Based Grammar--344
9.4 Summary---356
9.5 Further Reading--357
9.6 Exercises---358

10. Analyzing the Meaning of Sentences--361
10.1 Natural Language Understanding--361
10.2 Propositional Logic--368
10.3 First-Order Logic--372
10.4 The Semantics of English Sentences--385
10.5 Discourse Semantics--397
10.6 Summary--402
10.7 Further Reading--403
10.8 Exercises---404

11. Managing Linguistic Data---407
11.1 Corpus Structure: A Case Study--407
11.2 The Life Cycle of a Corpus---412
11.3 Acquiring Data---416
11.4 Working with XML---425
11.5 Working with Toolbox Data---431
11.6 Describing Language Resources Using OLAC Metadata----------------------------435
11.7 Summary--437
11.8 Further Reading--437
11.9 Exercises---438

Afterword: The Language Challenge---441
Bibliography---449
NLTK Index---459
General Index--463

***Numerical_Methods_in_Engineering_with_Python_2005_433p

Preface--vii
1. Introduction to Python--1

2. Systems of Linear Algebraic Equations---27

3. Interpolation and Curve Fitting--103

4. Roots of Equations---142

5. Numerical Differentiation--181

6. Numerical Integration--198

7. Initial Value Problems---248

8. Two-Point Boundary Value Problems--295

9. Symmetric Matrix Eigenvalue Problems---324

10. Introduction to Optimization---381

Appendices---409

Index--419

***Oreilly_Python_Cookbook_2nd_2005_974p

Preface
The Design of the Book
The Implementation of the Book
Using the Code from This Book
Audience
Organization
Further Reading
Conventions Used in This Book
How to Contact Us
Safari® Enabled
Acknowledgments
Chapter 1. Text
Recipe 1.1. Processing a String One Character at a Time
Recipe 1.2. Converting Between Characters and Numeric Codes
Recipe 1.3. Testing Whether an Object Is String-like
Recipe 1.4. Aligning Strings
Recipe 1.5. Trimming Space from the Ends of a String
Recipe 1.6. Combining Strings
Recipe 1.7. Reversing a String by Words or Characters
Recipe 1.8. Checking Whether a String Contains a Set of Characters
Recipe 1.9. Simplifying Usage of Strings' translate Method
Recipe 1.10. Filtering a String for a Set of Characters
Recipe 1.11. Checking Whether a String Is Text or Binary
Recipe 1.12. Controlling Case
Recipe 1.13. Accessing Substrings
Recipe 1.14. Changing the Indentation of a Multiline String
Recipe 1.15. Expanding and Compressing Tabs
Recipe 1.16. Interpolating Variables in a String
Recipe 1.17. Interpolating Variables in a Stringin Python 2.4
Recipe 1.18. Replacing Multiple Patterns in a Single Pass
Recipe 1.19. Checking a String for Any of Multiple Endings
Recipe 1.20. Handling International Text with Unicode
Recipe 1.21. Converting Between Unicode and Plain Strings
Recipe 1.22. Printing Unicode Charactersto Standard Output
Recipe 1.23. Encoding Unicode Data for XML and HTML
Recipe 1.24. Making Some Strings Case-Insensitive
Recipe 1.25. Converting HTML Documents to Texton a Unix Terminal
Chapter 2. Files
Recipe 2.1. Reading from a File

Recipe 2.2. Writing to a File
Recipe 2.3. Searching and Replacing Text in a File
Recipe 2.4. Reading a Specific Line from a File
Recipe 2.5. Counting Lines in a File
Recipe 2.6. Processing Every Word in a File
Recipe 2.7. Using Random-Access Input/Output
Recipe 2.8. Updating a Random-Access File
Recipe 2.9. Reading Data from zip Files
Recipe 2.10. Handling a zip File Inside a String
Recipe 2.11. Archiving a Tree of Files into a Compressed tar File
Recipe 2.12. Sending Binary Data to Standard Output Under Windows
Recipe 2.13. Using a C++-like iostream Syntax
Recipe 2.14. Rewinding an Input File to the Beginning
Recipe 2.15. Adapting a File-like Object to a True File Object
Recipe 2.16. Walking Directory Trees
Recipe 2.17. Swapping One File Extension for Another Throughout a Directory Tree
Recipe 2.18. Finding a File Given a Search Path
Recipe 2.19. Finding Files Given a Search Path and a Pattern
Recipe 2.20. Finding a File on the Python Search Path
Recipe 2.21. Dynamically Changing the PythonSearch Path
Recipe 2.22. Computing the Relative Path from One Directory to Another
Recipe 2.23. Reading an Unbuffered Character in a Cross-Platform Way
Recipe 2.24. Counting Pages of PDF Documents on Mac OS X
Recipe 2.25. Changing File Attributes on Windows
Recipe 2.26. Extracting Text from OpenOffice.org Documents
Recipe 2.27. Extracting Text from Microsoft Word Documents
Recipe 2.28. File Locking Using a Cross-Platform API
Recipe 2.29. Versioning Filenames
Recipe 2.30. Calculating CRC-64 Cyclic Redundancy Checks
Chapter 3. Time and Money
Recipe 3.1. Calculating Yesterday and Tomorrow
Recipe 3.2. Finding Last Friday
Recipe 3.3. Calculating Time Periods in a Date Range
Recipe 3.4. Summing Durations of Songs
Recipe 3.5. Calculating the Number of Weekdays Between Two Dates
Recipe 3.6. Looking up Holidays Automatically
Recipe 3.7. Fuzzy Parsing of Dates
Recipe 3.8. Checking Whether Daylight Saving Time Is Currently in Effect
Recipe 3.9. Converting Time Zones
Recipe 3.10. Running a Command Repeatedly
Recipe 3.11. Scheduling Commands
Recipe 3.12. Doing Decimal Arithmetic
Recipe 3.13. Formatting Decimals as Currency
Recipe 3.14. Using Python as a Simple Adding Machine
Recipe 3.15. Checking a Credit Card Checksum
Recipe 3.16. Watching Foreign Exchange Rates
Chapter 4. Python Shortcuts
Recipe 4.1. Copying an Object
Recipe 4.2. Constructing Lists with List Comprehensions
Recipe 4.3. Returning an Element of a List If It Exists
Recipe 4.4. Looping over Items and Their Indices in a Sequence
Recipe 4.5. Creating Lists of Lists Without Sharing References

Recipe 4.6. Flattening a Nested Sequence
Recipe 4.7. Removing or Reordering Columnsin a List of Rows
Recipe 4.8. Transposing Two-Dimensional Arrays
Recipe 4.9. Getting a Value from a Dictionary
Recipe 4.10. Adding an Entry to a Dictionary
Recipe 4.11. Building a Dictionary Without Excessive Quoting
Recipe 4.12. Building a Dict from a List of Alternating Keys and Values
Recipe 4.13. Extracting a Subset of a Dictionary
Recipe 4.14. Inverting a Dictionary
Recipe 4.15. Associating Multiple Values with Each Key in a Dictionary
Recipe 4.16. Using a Dictionary to Dispatch Methods or Functions
Recipe 4.17. Finding Unions and Intersections of Dictionaries
Recipe 4.18. Collecting a Bunch of Named Items
Recipe 4.19. Assigning and Testing with One Statement
Recipe 4.20. Using printf in Python
Recipe 4.21. Randomly Picking Items with Given Probabilities
Recipe 4.22. Handling Exceptions Within an Expression
Recipe 4.23. Ensuring a Name Is Defined in a Given Module
Chapter 5. Searching and Sorting
Recipe 5.1. Sorting a Dictionary
Recipe 5.2. Sorting a List of Strings Case-Insensitively
Recipe 5.3. Sorting a List of Objects by an Attribute of the Objects
Recipe 5.4. Sorting Keys or Indices Basedon the Corresponding Values
Recipe 5.5. Sorting Strings with Embedded Numbers
Recipe 5.6. Processing All of a List's Items in Random Order
Recipe 5.7. Keeping a Sequence Ordered as Items Are Added
Recipe 5.8. Getting the First Few Smallest Items of a Sequence
Recipe 5.9. Looking for Items in a Sorted Sequence
Recipe 5.10. Selecting the nth Smallest Element of a Sequence
Recipe 5.11. Showing off quicksort in Three Lines
Recipe 5.12. Performing Frequent Membership Tests on a Sequence
Recipe 5.13. Finding Subsequences
Recipe 5.14. Enriching the Dictionary Type with Ratings Functionality
Recipe 5.15. Sorting Names and Separating Them by Initials
Chapter 6. Object-Oriented Programming
Recipe 6.1. Converting Among Temperature Scales
Recipe 6.2. Defining Constants
Recipe 6.3. Restricting Attribute Setting
Recipe 6.4. Chaining Dictionary Lookups
Recipe 6.5. Delegating Automatically as an Alternative to Inheritance
Recipe 6.6. Delegating Special Methods in Proxies
Recipe 6.7. Implementing Tuples with Named Items
Recipe 6.8. Avoiding Boilerplate Accessors for Properties
Recipe 6.9. Making a Fast Copy of an Object
Recipe 6.10. Keeping References to Bound Methods Without Inhibiting Garbage Collection
Recipe 6.11. Implementing a Ring Buffer
Recipe 6.12. Checking an Instance for Any State Changes
Recipe 6.13. Checking Whether an Object Has Necessary Attributes
Recipe 6.14. Implementing the State Design Pattern
Recipe 6.15. Implementing the "Singleton" Design Pattern
Recipe 6.16. Avoiding the "Singleton" Design Pattern with the Borg Idiom
Recipe 6.17. Implementing the Null Object Design Pattern

Recipe 6.18. Automatically Initializing Instance Variables from _ _init_ _ Arguments
Recipe 6.19. Calling a Superclass _ _init_ _ Method If It Exists
Recipe 6.20. Using Cooperative Supercalls Concisely and Safely
Chapter 7. Persistence and Databases
Recipe 7.1. Serializing Data Using the marshal Module
Recipe 7.2. Serializing Data Using the pickle and cPickle Modules
Recipe 7.3. Using Compression with Pickling
Recipe 7.4. Using the cPickle Module on Classes and Instances
Recipe 7.5. Holding Bound Methods in a Picklable Way
Recipe 7.6. Pickling Code Objects
Recipe 7.7. Mutating Objects with shelve
Recipe 7.8. Using the Berkeley DB Database
Recipe 7.9. Accesssing a MySQL Database
Recipe 7.10. Storing a BLOB in a MySQL Database
Recipe 7.11. Storing a BLOB in a PostgreSQL Database
Recipe 7.12. Storing a BLOB in a SQLite Database
Recipe 7.13. Generating a Dictionary Mapping Field Names to Column Numbers
Recipe 7.14. Using dtuple for Flexible Accessto Query Results
Recipe 7.15. Pretty-Printing the Contents of Database Cursors
Recipe 7.16. Using a Single Parameter-Passing Style Across Various DB API Modules
Recipe 7.17. Using Microsoft Jet via ADO
Recipe 7.18. Accessing a JDBC Database from a Jython Servlet
Recipe 7.19. Using ODBC to Get Excel Data with Jython
Chapter 8. Debugging and Testing
Recipe 8.1. Disabling Execution of Some Conditionals and Loops
Recipe 8.2. Measuring Memory Usage on Linux
Recipe 8.3. Debugging the Garbage-Collection Process
Recipe 8.4. Trapping and Recording Exceptions
Recipe 8.5. Tracing Expressions and Comments in Debug Mode
Recipe 8.6. Getting More Information from Tracebacks
Recipe 8.7. Starting the Debugger Automatically After an Uncaught Exception
Recipe 8.8. Running Unit Tests Most Simply
Recipe 8.9. Running Unit Tests Automatically
Recipe 8.10. Using doctest with unittest in Python 2.4
Recipe 8.11. Checking Values Against Intervals in Unit Testing
Chapter 9. Processes, Threads, and Synchronization
Recipe 9.1. Synchronizing All Methods in an Object
Recipe 9.2. Terminating a Thread
Recipe 9.3. Using a Queue.Queue as a Priority Queue
Recipe 9.4. Working with a Thread Pool
Recipe 9.5. Executing a Function in Parallel on Multiple Argument Sets
Recipe 9.6. Coordinating Threads by Simple Message Passing
Recipe 9.7. Storing Per-Thread Information
Recipe 9.8. Multitasking Cooperatively Without Threads
Recipe 9.9. Determining Whether Another Instanceof a Script Is Already Running in Windows
Recipe 9.10. Processing Windows Messages Using MsgWaitForMultipleObjects
Recipe 9.11. Driving an External Process with popen
Recipe 9.12. Capturing the Output and Error Streams from a Unix Shell Command
Recipe 9.13. Forking a Daemon Process on Unix
Chapter 10. System Administration
Recipe 10.1. Generating Random Passwords
Recipe 10.2. Generating Easily Remembered Somewhat-Random Passwords

Recipe 10.3. Authenticating Users by Means of a POP Server
Recipe 10.4. Calculating Apache Hits per IP Address
Recipe 10.5. Calculating the Rate of Client Cache Hits on Apache
Recipe 10.6. Spawning an Editor from a Script
Recipe 10.7. Backing Up Files
Recipe 10.8. Selectively Copying a Mailbox File
Recipe 10.9. Building a Whitelist of Email Addresses From a Mailbox
Recipe 10.10. Blocking Duplicate Mails
Recipe 10.11. Checking Your Windows Sound System
Recipe 10.12. Registering or Unregistering a DLL on Windows
Recipe 10.13. Checking and Modifying the Set of Tasks Windows Automatically Runs at Login
Recipe 10.14. Creating a Share on Windows
Recipe 10.15. Connecting to an Already Running Instance of Internet Explorer
Recipe 10.16. Reading Microsoft Outlook Contacts
Recipe 10.17. Gathering Detailed System Informationon Mac OS X
Chapter 11. User Interfaces
Recipe 11.1. Showing a Progress Indicator on a Text Console
Recipe 11.2. Avoiding lambda in Writing Callback Functions
Recipe 11.3. Using Default Values and Bounds with tkSimpleDialog Functions
Recipe 11.4. Adding Drag and Drop Reordering to a Tkinter Listbox
Recipe 11.5. Entering Accented Characters in Tkinter Widgets
Recipe 11.6. Embedding Inline GIFs Using Tkinter
Recipe 11.7. Converting Among Image Formats
Recipe 11.8. Implementing a Stopwatch in Tkinter
Recipe 11.9. Combining GUIs and Asynchronous I/Owith Threads
Recipe 11.10. Using IDLE's Tree Widget in Tkinter
Recipe 11.11. Supporting Multiple Values per Row in a Tkinter Listbox
Recipe 11.12. Copying Geometry Methods and Options Between Tkinter Widgets
Recipe 11.13. Implementing a Tabbed Notebook for Tkinter
Recipe 11.14. Using a wxPython Notebook with Panels
Recipe 11.15. Implementing an ImageJ Plug-in in Jython
Recipe 11.16. Viewing an Image from a URL with Swing and Jython
Recipe 11.17. Getting User Input on Mac OS
Recipe 11.18. Building a Python Cocoa GUI Programmatically
Recipe 11.19. Implementing Fade-in Windows with IronPython
Chapter 12. Processing XML
Recipe 12.1. Checking XML Well-Formedness
Recipe 12.2. Counting Tags in a Document
Recipe 12.3. Extracting Text from an XML Document
Recipe 12.4. Autodetecting XML Encoding
Recipe 12.5. Converting an XML Document into a Tree of Python Objects
Recipe 12.6. Removing Whitespace-only Text Nodes from an XML DOM Node's Subtree
Recipe 12.7. Parsing Microsoft Excel's XML
Recipe 12.8. Validating XML Documents
Recipe 12.9. Filtering Elements and Attributes Belonging to a Given Namespace
Recipe 12.10. Merging Continuous Text Events with a SAX Filter
Recipe 12.11. Using MSHTML to Parse XML or HTML
Chapter 13. Network Programming
Recipe 13.1. Passing Messages with Socket Datagrams
Recipe 13.2. Grabbing a Document from the Web
Recipe 13.3. Filtering a List of FTP Sites
Recipe 13.4. Getting Time from a Server via the SNTP Protocol

Recipe 13.5. Sending HTML Mail
Recipe 13.6. Bundling Files in a MIME Message
Recipe 13.7. Unpacking a Multipart MIME Message
Recipe 13.8. Removing Attachments from an Email Message
Recipe 13.9. Fixing Messages Parsed by Python 2.4 email.FeedParser
Recipe 13.10. Inspecting a POP3 Mailbox Interactively
Recipe 13.11. Detecting Inactive Computers
Recipe 13.12. Monitoring a Network with HTTP
Recipe 13.13. Forwarding and Redirecting Network Ports
Recipe 13.14. Tunneling SSL Through a Proxy
Recipe 13.15. Implementing the Dynamic IP Protocol
Recipe 13.16. Connecting to IRC and Logging Messages to Disk
Recipe 13.17. Accessing LDAP Servers
Chapter 14. Web Programming
Recipe 14.1. Testing Whether CGI Is Working
Recipe 14.2. Handling URLs Within a CGI Script
Recipe 14.3. Uploading Files with CGI
Recipe 14.4. Checking for a Web Page's Existence
Recipe 14.5. Checking Content Type via HTTP
Recipe 14.6. Resuming the HTTP Download of a File
Recipe 14.7. Handling Cookies While Fetching Web Pages
Recipe 14.8. Authenticating with a Proxy for HTTPS Navigation
Recipe 14.9. Running a Servlet with Jython
Recipe 14.10. Finding an Internet Explorer Cookie
Recipe 14.11. Generating OPML Files
Recipe 14.12. Aggregating RSS Feeds
Recipe 14.13. Turning Data into Web Pages Through Templates
Recipe 14.14. Rendering Arbitrary Objects with Nevow
Chapter 15. Distributed Programming
Recipe 15.1. Making an XML-RPC Method Call
Recipe 15.2. Serving XML-RPC Requests
Recipe 15.3. Using XML-RPC with Medusa
Recipe 15.4. Enabling an XML-RPC Server to Be Terminated Remotely
Recipe 15.5. Implementing SimpleXMLRPCServer Niceties
Recipe 15.6. Giving an XML-RPC Server a wxPython GUI
Recipe 15.7. Using Twisted Perspective Broker
Recipe 15.8. Implementing a CORBA Server and Client
Recipe 15.9. Performing Remote Logins Using telnetlib
Recipe 15.10. Performing Remote Logins with SSH
Recipe 15.11. Authenticating an SSL Client over HTTPS
Chapter 16. Programs About Programs
Recipe 16.1. Verifying Whether a String Represents a Valid Number
Recipe 16.2. Importing a Dynamically Generated Module
Recipe 16.3. Importing from a Module Whose Name Is Determined at Runtime
Recipe 16.4. Associating Parameters with a Function (Currying)
Recipe 16.5. Composing Functions
Recipe 16.6. Colorizing Python Source Using the Built-in Tokenizer
Recipe 16.7. Merging and Splitting Tokens
Recipe 16.8. Checking Whether a String Has Balanced Parentheses
Recipe 16.9. Simulating Enumerations in Python
Recipe 16.10. Referring to a List Comprehension While Building It
Recipe 16.11. Automating the py2exe Compilation of Scripts into Windows Executables

Recipe 16.12. Binding Main Script and Modules into One Executable on Unix
Chapter 17. Extending and Embedding
Recipe 17.1. Implementing a Simple Extension Type
Recipe 17.2. Implementing a Simple Extension Type with Pyrex
Recipe 17.3. Exposing a C++ Library to Python
Recipe 17.4. Calling Functions from a Windows DLL
Recipe 17.5. Using SWIG-Generated Modules in a Multithreaded Environment
Recipe 17.6. Translating a Python Sequence into a C Array with the PySequence_Fast Protocol
Recipe 17.7. Accessing a Python Sequence Item-by-Item with the Iterator Protocol
Recipe 17.8. Returning None from a Python-Callable C Function
Recipe 17.9. Debugging Dynamically Loaded C Extensions with gdb
Recipe 17.10. Debugging Memory Problems
Chapter 18. Algorithms
Recipe 18.1. Removing Duplicates from a Sequence
Recipe 18.2. Removing Duplicates from a Sequence While Maintaining Sequence Order
Recipe 18.3. Generating Random Samples with Replacement
Recipe 18.4. Generating Random Samples Without Replacement
Recipe 18.5. Memoizing (Caching) the Return Values of Functions
Recipe 18.6. Implementing a FIFO Container
Recipe 18.7. Caching Objects with a FIFO Pruning Strategy
Recipe 18.8. Implementing a Bag (Multiset) Collection Type
Recipe 18.9. Simulating the Ternary Operator in Python
Recipe 18.10. Computing Prime Numbers
Recipe 18.11. Formatting Integers as Binary Strings
Recipe 18.12. Formatting Integers as Strings in Arbitrary Bases
Recipe 18.13. Converting Numbers to Rationals via Farey Fractions
Recipe 18.14. Doing Arithmetic with Error Propagation
Recipe 18.15. Summing Numbers with Maximal Accuracy
Recipe 18.16. Simulating Floating Point
Recipe 18.17. Computing the Convex Hulls and Diameters of 2D Point Sets
Chapter 19. Iterators and Generators
Recipe 19.1. Writing a range-like Function with Float Increments
Recipe 19.2. Building a List from Any Iterable
Recipe 19.3. Generating the Fibonacci Sequence
Recipe 19.4. Unpacking a Few Items in a Multiple Assignment
Recipe 19.5. Automatically Unpacking the Needed Number of Items
Recipe 19.6. Dividing an Iterable into n Slices of Stride n
Recipe 19.7. Looping on a Sequence by Overlapping Windows
Recipe 19.8. Looping Through Multiple Iterables in Parallel
Recipe 19.9. Looping Through the Cross-Product of Multiple Iterables
Recipe 19.10. Reading a Text File by Paragraphs
Recipe 19.11. Reading Lines with Continuation Characters
Recipe 19.12. Iterating on a Stream of Data Blocks as a Stream of Lines
Recipe 19.13. Fetching Large Record Sets from a Database with a Generator
Recipe 19.14. Merging Sorted Sequences
Recipe 19.15. Generating Permutations, Combinations, and Selections
Recipe 19.16. Generating the Partitions of an Integer
Recipe 19.17. Duplicating an Iterator
Recipe 19.18. Looking Ahead into an Iterator
Recipe 19.19. Simplifying Queue-Consumer Threads
Recipe 19.20. Running an Iterator in Another Thread
Recipe 19.21. Computing a Summary Report with itertools.groupby

Chapter 20. Descriptors, Decorators,and Metaclasses
Recipe 20.1. Getting Fresh Default Values at Each Function Call
Recipe 20.2. Coding Properties as Nested Functions
Recipe 20.3. Aliasing Attribute Values
Recipe 20.4. Caching Attribute Values
Recipe 20.5. Using One Method as Accessorfor Multiple Attributes
Recipe 20.6. Adding Functionality to a Class by Wrapping a Method
Recipe 20.7. Adding Functionality to a Class by Enriching All Methods
Recipe 20.8. Adding a Method to a Class Instance at Runtime
Recipe 20.9. Checking Whether Interfaces Are Implemented
Recipe 20.10. Using _ _new_ _ and _ _init_ _ Appropriately in Custom Metaclasses
Recipe 20.11. Allowing Chaining of Mutating List Methods
Recipe 20.12. Using Cooperative Super calls with Terser Syntax
Recipe 20.13. Initializing Instance Attributes Without Using _ _init_ _
Recipe 20.14. Automatic Initialization of Instance Attributes
Recipe 20.15. Upgrading Class Instances Automatically on reload
Recipe 20.16. Binding Constants at Compile Time
Recipe 20.17. Solving Metaclass Conflicts
Colophon
Index

***Practical_Programming_An_Introduction_to_Computer_Science_Using_Python_2009_369p

1 Introduction--11
1.1 Programs and Programming---13
1.2 A Few Definitions---14
1.3 What to Install---14
1.4 For Instructors---15
1.5 Summary---16

2 Hello, Python--17
2.1 The Big Picture--17
2.2 Expressions---19
2.3 What Is a Type?--22
2.4 Variables and the Assignment Statement--25
2.5 When Things Go Wrong---29
2.6 Function Basics---30
2.7 Built-in Functions--33
2.8 Style Notes--34
2.9 Summary---35
2.10 Exercises--36

3 Strings---39
3.1 Strings---39
3.2 Escape Characters--42
3.3 Multiline Strings--43
3.4 Print--44
3.5 Formatted Printing--45
3.6 User Input--46
3.7 Summary--47

3.8 Exercises--48

4 Modules--50
4.1 Importing Modules---50
4.2 Defining Your Own Modules--54
4.3 Objects and Methods---60
4.4 Pixels and Colors---68
4.5 Testing--70
4.6 Style Notes--76
4.7 Summary---77
4.8 Exercises---78

5 Lists---81
5.1 Lists and Indices--81
5.2 Modifying Lists---85
5.3 Built-in Functions on Lists---87
5.4 Processing List Items---89
5.5 Slicing--92
5.6 Aliasing---94
5.7 List Methods---95
5.8 Nested Lists--97
5.9 Other Kinds of Sequences---99
5.10 Files as Lists---100
5.11 Comments---103
5.12 Summary--105
5.13 Exercises--105

6 Making Choices---108
6.1 Boolean Logic--108
6.2 if Statements--118
6.3 Storing Conditionals--125
6.4 Summary---126
6.5 Exercises---127

7 Repetition--131
7.1 Counted Loops---131
7.2 while Loops---140
7.3 User Input Loops---148
7.4 Controlling Loops--149
7.5 Style Notes--153
7.6 Summary---154
7.7 Exercises---155

8 File Processing---159
8.1 One Record per Line---160
8.2 Records with Multiple Fields--171
8.3 Positional Data---174
8.4 Multiline Records--177
8.5 Looking Ahead--179
8.6 Writing to Files---181
8.7 Summary---183
8.8 Exercises---183

9 Sets and Dictionaries------185
9.1 Sets------185
9.2 Dictionaries------190
9.3 Inverting a Dictionary------197
9.4 Summary------198
9.5 Exercises------199

10 Algorithms------203
10.1 Searching------204
10.2 Timing------211
10.3 Summary------211
10.4 Exercises------212

11 Searching and Sorting------214
11.1 Linear Search------214
11.2 Binary Search------218
11.3 Sorting------222
11.4 More Efficient Sorting Algorithms------228
11.5 Mergesort: An N log2 N Algorithm------229
11.6 Summary------233
11.7 Exercises------234

12 Construction------237
12.1 More on Functions------237
12.2 Exceptions------242
12.3 Testing------249
12.4 Debugging------254
12.5 Patterns------256
12.6 Summary------260
12.7 Exercises------261

13 Object-Oriented Programming------270
13.1 Class Color------271
13.2 Special Methods------276
13.3 More About dir and help------278
13.4 A Little Bit of OO Theory------280
13.5 A Longer Example------288
13.6 Summary------293
13.7 Exercises------293

14 Graphical User Interfaces------294
14.1 The Tkinter Module------295
14.2 Basic GUI Construction------296
14.3 Models, Views, and Controllers------301
14.4 Style------307
14.5 A Few More Widgets------312
14.6 Object-Oriented GUIs------316
14.7 Summary------317
14.8 Exercises------318

15 Databases------321

15.1 The Big Picture---321
15.2 First Steps--323
15.3 Retrieving Data---327
15.4 Updating and Deleting---330
15.5 Transactions--331
15.6 Using NULL for Missing Data--333
15.7 Using Joins to Combine Tables--334
15.8 Keys and Constraints--339
15.9 Advanced Features---341
15.10 Summary--346
15.11 Exercises--347
A Bibliography---351
Index--353

***Prentice_Hall_Core_Python_Programming_2nd_Edition_2001_703p

Welcome to Python!
Style:Technical, Yet Easy Reading
Author's Experience with Python
Book Contents
Part I : Core Python
Chapter 1 —Welcome to Python!
Chapter 2 —Getting Started
Chapter 3 —Syntax and Style
Chapter 4 —Python Objects
Chapter 5 —Numbers
Chapter 6 —Sequences: Strings, Lists, and Tuples
Chapter 7 —Dictionaries
Chapter 8 —Conditionals and Loops
Chapter 9 —Files and Input/Output
Chapter 10 —Errors and Exceptions
Chapter 11 —Functions
Chapter 12 —Modules
Chapter 13 —Classes and OOP
Chapter 14 —Execution Environment
Part II : Advanced Topics
Chapter 15 —Regular Expressions
Chapter 16 —Network Programming with Sockets
Chapter 17 —Multithreaded Programming
Chapter 18 —GUI Programming with Tkinter
Chapter 19 —Web Programming
Chapter 20 —Extending Python
Optional Sections
Conventions
Book Support

***Professional_Python_Frameworks_Web_2.0_Programming_with_Django_and_TurboGears_2007_459p

Acknowledgments--xiii
Introduction--xxi

Part I: Introduction to Python Frameworks
Chapter 1: Web 2 0, Python, and Frameworks--3
Chapter 2: Web 2 0 with Traditional Python---19
Chapter 3: Introducing the Frameworks--47

Part II: TurboGears
Chapter 4: Introduction to TurboGears---101
Chapter 5: User Identity and Visitor Tracking--137
Chapter 6: Widgets--169
Chapter 7: Advanced TurboGears Topics--203

Part III: Django
Chapter 8: Dblog: A Blog Implemented in Django---241
Chapter 9: Django Views and Users--275
Chapter 10: Advanced Django Topics: AJAX, RSS, and More-----------------------------305

Part IV: Advanced Client-Side Topics
Chapter 11: MochiKit — Pythonic JavaScripting--337
Chapter 12: Flash-Based Interfaces and TurboGears--------------------------------------359

Appendix A: Tools Used---395
Index--405

***Programming_in_Python_3_A_Complete_Introduction_to_the_Python_Language_SecondEdition_2010_644p

List of Tables---xv
Introduction--1
Chapter 1. Rapid Introduction to Procedural Programming--------------------------------------9
Chapter 2. Data Types---51
Chapter 3. Collection Data Types---107
Chapter 4. Control Structures and Functions---159
Chapter 5. Modules---195
Chapter 6. Object-Oriented Programming--233
Chapter 7. File Handling--287
Chapter 8. Advanced Programming Techniques---339
Chapter 9. Debugging, Testing, and Profiling---413
Chapter 10. Processes and Threading---439
Chapter 11. Networking---457
Chapter 12. Database Programming--475
Chapter 13. Regular Expressions---489
Chapter 14. Introduction to Parsing--513
Chapter 15. Introduction to GUI Programming---569
Epilogue--595
Selected Bibliography--597

Index---599

***Programming_Python_3rdEd_2006_1740p

Foreword
Preface
Part I: The Beginning
Chapter 1. Introducing Python
Section 1.1. "And Now for Something Completely Different"
Section 1.2. Python Philosophy 101
Section 1.3. The Life of Python
Section 1.4. Signs of the Python Times
Section 1.5. The Compulsory Features List
Section 1.6. What's Python Good For?
Section 1.7. What's Python Not Good For?
Section 1.8. Truth in Advertising
Chapter 2. A Sneak Preview
Section 2.1. "Programming Python: The Short Story"
Section 2.2. The Task
Section 2.3. Step 1: Representing Records
Section 2.4. Step 2: Storing Records Persistently
Section 2.5. Step 3: Stepping Up to OOP
Section 2.6. Step 4: Adding Console Interaction
Section 2.7. Step 5: Adding a GUI
Section 2.8. Step 6: Adding a Web Interface
Section 2.9. The End of the Demo
Part II: System Programming
Chapter 3. System Tools
Section 3.1. "The os.path to Knowledge"
Section 3.2. System Scripting Overview
Section 3.3. Introducing the sys Module
Section 3.4. Introducing the os Module
Section 3.5. Script Execution Context
Section 3.6. Current Working Directory
Section 3.7. Command-Line Arguments
Section 3.8. Shell Environment Variables
Section 3.9. Standard Streams
Chapter 4. File and Directory Tools
Section 4.1. "Erase Your Hard Drive in Five Easy Steps!"
Section 4.2. File Tools
Section 4.3. Directory Tools
Chapter 5. Parallel System Tools
Section 5.1. "Telling the Monkeys What to Do"
Section 5.2. Forking Processes
Section 5.3. Threads
Section 5.4. Program Exits
Section 5.5. Interprocess Communication
Section 5.6. Pipes
Section 5.7. Signals
Section 5.8. Other Ways to Start Programs
Section 5.9. A Portable Program-Launch Framework
Section 5.10. Other System Tools

Chapter 6. System Examples: Utilities
Section 6.1. "Splits and Joins and Alien Invasions"
Section 6.2. Splitting and Joining Files
Section 6.3. Generating Forward-Link Web Pages
Section 6.4. A Regression Test Script
Section 6.5. Packing and Unpacking Files
Section 6.6. Automated Program Launchers
Chapter 7. System Examples: Directories
Section 7.1. "The Greps of Wrath"
Section 7.2. Fixing DOS Line Ends
Section 7.3. Fixing DOS Filenames
Section 7.4. Searching Directory Trees
Section 7.5. Visitor: Walking Trees Generically
Section 7.6. Copying Directory Trees
Section 7.7. Deleting Directory Trees
Section 7.8. Comparing Directory Trees
Part III: GUI Programming
Chapter 8. Graphical User Interfaces
Section 8.1. "Here's Looking at You, Kid"
Section 8.2. Python GUI Development Options
Section 8.3. Tkinter Overview
Section 8.4. Climbing the GUI Learning Curve
Section 8.5. Tkinter Coding Basics
Section 8.6. Tkinter Coding Alternatives
Section 8.7. Adding Buttons and Callbacks
Section 8.8. Adding User-Defined Callback Handlers
Section 8.9. Adding Multiple Widgets
Section 8.10. Customizing Widgets with Classes
Section 8.11. Reusable GUI Components with Classes
Section 8.12. The End of the Tutorial
Section 8.13. Python/Tkinter for Tcl/Tk Converts
Chapter 9. A Tkinter Tour, Part 1
Section 9.1. "Widgets and Gadgets and GUIs, Oh My!"
Section 9.2. Configuring Widget Appearance
Section 9.3. Top-Level Windows
Section 9.4. Dialogs
Section 9.5. Binding Events
Section 9.6. Message and Entry
Section 9.7. Checkbutton, Radiobutton, and Scale
Section 9.8. Running GUI Code Three Ways
Section 9.9. Images
Section 9.10. Viewing and Processing Images with PIL
Chapter 10. A Tkinter Tour, Part 2
Section 10.1. "On Today's Menu: Spam, Spam, and Spam"
Section 10.2. Menus
Section 10.3. Listboxes and Scrollbars
Section 10.4. Text
Section 10.5. Canvas
Section 10.6. Grids
Section 10.7. Time Tools, Threads, and Animation
Section 10.8. The End of the Tour
Section 10.9. The PyDemos and PyGadgets Launchers

Chapter 11. GUI Coding Techniques
Section 11.1. "Building a Better Mouse Trap"
Section 11.2. GuiMixin: Common Tool Mixin Classes
Section 11.3. GuiMaker: Automating Menus and Toolbars
Section 11.4. ShellGui: GUIs for Command-Line Tools
Section 11.5. GuiStreams: Redirecting Streams to Widgets
Section 11.6. Reloading Callback Handlers Dynamically
Section 11.7. Wrapping Up Top-Level Window Interfaces
Section 11.8. GUIs, Threads, and Queues
Section 11.9. More Ways to Add GUIs to Non-GUI Code
Chapter 12. Complete GUI Programs
Section 12.1. "Python, Open Source, and Camaros"
Section 12.2. PyEdit: A Text Editor Program/Object
Section 12.3. PyPhoto: An Image Viewer and Resizer
Section 12.4. PyView: An Image and Notes Slideshow
Section 12.5. PyDraw: Painting and Moving Graphics
Section 12.6. PyClock: An Analog/Digital Clock Widget
Section 12.7. PyToe: A Tic-Tac-Toe Game Widget
Section 12.8. Where to Go from Here
Part IV: Internet Programming
Chapter 13. Network Scripting
Section 13.1. "Tune In, Log On, and Drop Out"
Section 13.2. Plumbing the Internet
Section 13.3. Socket Programming
Section 13.4. Handling Multiple Clients
Section 13.5. A Simple Python File Server
Chapter 14. Client-Side Scripting
Section 14.1. "Socket to Me!"
Section 14.2. FTP: Transferring Files over the Net
Section 14.3. Processing Internet Email
Section 14.4. POP: Fetching Email
Section 14.5. SMTP: Sending Email
Section 14.6. email: Parsing and Composing Mails
Section 14.7. pymail: A Console-Based Email Client
Section 14.8. The mailtools Utility Package
Section 14.9. NNTP: Accessing Newsgroups
Section 14.10. HTTP: Accessing Web Sites
Section 14.11. Module urllib Revisited
Section 14.12. Other Client-Side Scripting Options
Chapter 15. The PyMailGUI Client
Section 15.1. "Use the Source, Luke"
Section 15.2. A PyMailGUI Demo
Section 15.3. PyMailGUI Implementation
Chapter 16. Server-Side Scripting
Section 16.1. "Oh What a Tangled Web We Weave"
Section 16.2. What's a Server-Side CGI Script?
Section 16.3. Running Server-Side Examples
Section 16.4. Climbing the CGI Learning Curve
Section 16.5. Saving State Information in CGI Scripts
Section 16.6. The Hello World Selector
Section 16.7. Refactoring Code for Maintainability
Section 16.8. More on HTML and URL Escapes

Section 16.9. Transferring Files to Clients and Servers
Chapter 17. The PyMailCGI Server
Section 17.1. "Things to Do When Visiting Chicago"
Section 17.2. The PyMailCGI Web Site
Section 17.3. The Root Page
Section 17.4. Sending Mail by SMTP
Section 17.5. Reading POP Email
Section 17.6. Processing Fetched Mail
Section 17.7. Utility Modules
Section 17.8. CGI Script Trade-Offs
Chapter 18. Advanced Internet Topics
Section 18.1. "Surfing on the Shoulders of Giants"
Section 18.2. Zope: A Web Application Framework
Section 18.3. HTMLgen: Web Pages from Objects
Section 18.4. Jython: Python for Java
Section 18.5. Grail: A Python-Based Web Browser
Section 18.6. XML Processing Tools
Section 18.7. Windows Web Scripting Extensions
Section 18.8. Python Server Pages
Section 18.9. Rolling Your Own Servers in Python
Section 18.10. And Other Cool Stuff
Part V: Tools and Techniques
Chapter 19. Databases and Persistence
Section 19.1. "Give Me an Order of Persistence, but Hold the Pickles"
Section 19.2. Persistence Options in Python
Section 19.3. DBM Files
Section 19.4. Pickled Objects
Section 19.5. Shelve Files
Section 19.6. The ZODB Object-Oriented Database
Section 19.7. SQL Database Interfaces
Section 19.8. PyForm: A Persistent Object Viewer
Chapter 20. Data Structures
Section 20.1. "Roses Are Red, Violets Are Blue; Lists Are Mutable, and So Is Set Foo"
Section 20.2. Implementing Stacks
Section 20.3. Implementing Sets
Section 20.4. Subclassing Built-In Types
Section 20.5. Binary Search Trees
Section 20.6. Graph Searching
Section 20.7. Reversing Sequences
Section 20.8. Permuting Sequences
Section 20.9. Sorting Sequences
Section 20.10. Data Structures Versus Python Built-Ins
Section 20.11. PyTree: A Generic Tree Object Viewer
Chapter 21. Text and Language
Section 21.1. "See Jack Hack. Hack, Jack, Hack"
Section 21.2. Strategies for Parsing Text in Python
Section 21.3. String Method Utilities
Section 21.4. Regular Expression Pattern Matching
Section 21.5. Advanced Language Tools
Section 21.6. Handcoded Parsers
Section 21.7. PyCalc: A Calculator Program/Object
Part VI: Integration

Chapter 22. Extending Python
Section 22.1. "I Am Lost at C"
Section 22.2. Integration Modes
Section 22.3. C Extensions Overview
Section 22.4. A Simple C Extension Module
Section 22.5. Extension Module Details
Section 22.6. The SWIG Integration Code Generator
Section 22.7. Wrapping C Environment Calls
Section 22.8. A C Extension Module String Stack
Section 22.9. A C Extension Type String Stack
Section 22.10. Wrapping C++ Classes with SWIG
Section 22.11. Other Extending Tools
Chapter 23. Embedding Python
Section 23.1. "Add Python. Mix Well. Repeat."
Section 23.2. C Embedding API Overview
Section 23.3. Basic Embedding Techniques
Section 23.4. Registering Callback Handler Objects
Section 23.5. Using Python Classes in C
Section 23.6. A High-Level Embedding API: ppembed
Section 23.7. Other Integration Topics
Part VIII: The End
Chapter 24. Conclusion: Python and the Development Cycle
Section 24.1. "That's the End of the Book, Now Here's the Meaning of Life"
Section 24.2. "Something's Wrong with the Way We Program Computers"
Section 24.3. The "Gilligan Factor"
Section 24.4. Doing the Right Thing
Section 24.5. Enter Python
Section 24.6. But What About That Bottleneck?
Section 24.7. On Sinking the Titanic
Section 24.8. So What's "Python: The Sequel"?
Section 24.9. In the Final Analysis . . .
Section 24.10. Postscript to the Second Edition (2000)
Section 24.11. Postscript to the Third Edition (2006)
About the Author
Colophon
Index

***Pro_IronPython_2009_298p

About the Author--xiii
About the Technical Reviewer---xv
Acknowledgments--xvii
Introduction---xix
Chapter 1 Introduction to IronPython--1
Chapter 2 IronPython Syntax--15
Chapter 3 Advanced IronPython---39
Chapter 4 IronPython Studio---63
Chapter 5 Mixing and Mingling with the CLR---79
Chapter 6 Advanced Development---119
Chapter 7 Data Manipulation---163
Chapter 8 Caught in a Web--203
Chapter 9 IronPython Recipes---239

index---277

***Pro_Python_System_Administration_Apress_2010_417p

About the Author---xvi
About the Technical Reviewer---xvii
Acknowledgements--xviii
Introduction--xix

Chapter 1: Reading and Collecting Performance Data Using SNMP--------------------------------1
Chapter 2: Managing Devices Using the SOAP API--41
Chapter 3: Creating a Web Application for IP Address Accountancy -------------------------------75
Chapter 4: Integrating the IP Address Application with DHCP------------------------------------107
Chapter 5: Maintaining a List of Virtual Hosts in an Apache Configuration File ----------------137
Chapter 6: Gathering and Presenting Statistical Data from Apache Log Files--------------------159
Chapter 7: Performing Complex Searches and Reporting on Application Log Files-------------181
Chapter 8: A Web Site Availability Check Script for Nagios---------------------------------------211
Chapter 9: Management and Monitoring Subsystem---231
Chapter 10: Remote Monitoring Agents---269
Chapter 11: Statistics Gathering and Reporting---297
Chapter 12: Automatic MySQL Database Performance Tuning------------------------------------329
Chapter 13: Using Amazon EC2/S3 as a Data Warehouse Solution-------------------------------349
Index---373

***Python_2_1_Bible_2001_769p

Preface---ix
Acknowledgments--xiv

Part I: The Python Language--1
Chapter 1: Python in an Hour--3
Chapter 2: Identifiers, Variables, and Numeric Types---19
Chapter 3: Expressions and Strings---29
Chapter 4: Advanced Data Types---49
Chapter 5: Control Flow--73
Chapter 6: Program Organization---87
Chapter 7: Object-Oriented Python---99
Chapter 8: Input and Output---119

Part II: Files, Data Storage, and Operating System Services--------------------------------------131
Chapter 9: Processing Strings and Regular Expressions---133
Chapter 10: Working with Files and Directories--155
Chapter 11: Using Other Operating System Services--179
Chapter 12: Storing Data and Objects---195
Chapter 13: Accessing Date and Time---219
Chapter 14: Using Databases--229

Part III: Networking and the Internet--245
Chapter 15: Networking--247
Chapter 16: Speaking Internet Protocols---275
Chapter 17: Handling Internet Data--303

Chapter 18: Parsing XML and Other Markup Languages--325

Part IV: User Interfaces and Multimedia---345
Chapter 19: Tinkering with Tkinter--347
Chapter 20: Using Advanced Tkinter Widgets---371
Chapter 21: Building User Interfaces with wxPython--391
Chapter 22: Using Curses---415
Chapter 23: Building Simple Command Interpreters---433
Chapter 24: Playing Sound--453

Part V: Advanced Python Programming--465
Chapter 25: Processing Images--467
Chapter 26: Multithreading--481
Chapter 27: Debugging, Profiling, and Optimization---497
Chapter 28: Security and Encryption---515
Chapter 29: Writing Extension Modules---527
Chapter 30: Embedding the Python Interpreter--553
Chapter 31: Number Crunching---581
Chapter 32: Using NumPy--589
Chapter 33: Parsing and Interpreting Python Code---605

Part VI: Deploying Python Applications---617
Chapter 34: Creating Worldwide Applications--619
Chapter 35: Customizing Import Behavior--629
Chapter 36: Distributing Modules and Applications---643

Part VII: Platform-Specific Support---659
Chapter 37: Windows---661
Chapter 38: UNIX-Compatible Modules---671

Appendix A: Online Resources--685
Appendix B: Python Development Environments--689
Index--701

***Python_2_6_Quick_Reference_Letter_2009_50p

Front matter
Invocation Options
Environment variables
Lexical entities : keywords, identifiers, string literals, boolean constants, numbers, sequences, dictionaries,
operators
Basic types and their operations: None, bool, Numeric types, sequence types, list, dictionary, string, file, set,
named tuples, date/time
Advanced types
Statements: assignment, conditional expressions, control flow, exceptions, name space, function def, class def
Iterators; Generators; Descriptors; Decorators
Built-in Functions
Built-in Exceptions
Standard methods & operators redefinition in user-created Classes

Special informative state attributes for some types
Important modules : sys, os, posix, posixpath, shutil, time, string, re, math, getopt
List of modules in the base distribution
Workspace exploration and idiom hints
Python mode for Emacs

***Python_3_for_Absolute_Beginners_2009_314p

Contents--v
About the Author---xi
About the Technical Reviewer--xii
Chapter 1: Introducing Python---1
Chapter 2: Designing Software---9
Chapter 3: Variables and Data Types---27
Chapter 4: Making Choices---49
Chapter 5: Using Lists---75
Chapter 6: Functions--101
Chapter 7: Working with Text--125
Chapter 8: Executable Files, Organization, and Python on the Web-------------------------------161
Chapter 9: Classes---181
Chapter 10: Exceptions--221
Chapter 11: Reusing Code with Modules and Packages---241
Chapter 12: Simple Windowed Applications--261
Index--283

***Python_Create_Modify_Reuse_2008_291p

Introduction---xix
Chapter 1: A Python Primer---1
Part I: The Projects---17
Chapter 2: Directory/File Snapshot Program--19
Chapter 3: DVD Inventory System---43
Chapter 4: Web Performance Tester--81
Chapter 5: Customer Follow-Up System---103
Chapter 6: Test Management/Reporting System---125
Chapter 7: Version Management System--157
Chapter 8: Content Management System---177
Part II: Advanced Topics---197
Chapter 9: Interacting with the Operating System--199
Chapter 10: Debugging and Testing--221
Appendix A: Where to Go From Here — Resources That Can Help-----------------------------239
Appendix B: Installing Supplemental Programs--241
Index---253

***PythonDevelopersHandbook_2000_1481p

About the Author
Acknowledgments
Tell Us What You Think!
Introduction

I: Basic Programming
1. Introduction
2. Language Review
3. Python Libraries
4. Exception Handling
5. Object-Oriented Programming
II: Advanced Programming
6. Extending and Embedding Python
7. Objects Interfacing and Distribution
8. Working with Databases
9. Other Advanced Topics
III: Network Programming
10. Basic Network Background
11. Web Development
12. Scripting Programming
13. Data Manipulation
IV: Graphical Interfaces
14. Python and GUIs
15. Tkinter
16. Development Environment
17. Development Tools
VI: Python and Java
18. JPython
VII: Appendixes
A. Python/C API
B. Running Python on Specific Platforms
C. Python Copyright Notices
D. Migrating to Python 2.0

***Python_Essential_Reference_2nd_Edition_2001_538p

///

***Python_Essential_Reference_Fourth_Edition_2009_742p

Introduction--1
Part I: The Python Language
 1 A Tutorial Introduction--5
 2 Lexical Conventions and Syntax---25
 3 Types and Objects---33
 4 Operators and Expressions---65
 5 Program Structure and Control Flow--81
 6 Functions and Functional Programming---93
 7 Classes and Object-Oriented Programming---117
 8 Modules, Packages, and Distribution--143
 9 Input and Output---157
 10 Execution Environment---173
 11 Testing, Debugging, Profiling, and Tuning--181

Part II: The Python Library
 12 Built-In Functions---201

13 Python Runtime Services--219
14 Mathematics---243
15 Data Structures, Algorithms, and Code Simplification---257
16 String and Text Handling--277
17 Python Database Access---297
18 File and Directory Handling---313
19 Operating System Services--331
20 Threads and Concurrency---413
21 Network Programming and Sockets--449
22 Internet Application Programming---497
23 Web Programming---531
24 Internet Data Handling and Encoding--545
25 Miscellaneous Library Modules---585

Part III: Extending and Embedding

26 Extending and Embedding Python--591
Appendix: Python 3---621
Index---639

***Python_for_Dummies_2006_351p

Python for Dummies
Introduction
Part I - Getting Started
Chapter 1 - Introducing Python
Chapter 2 - Getting your Hands on the Keyboard—Using Help, Interactive Mode, and IDLE
Chapter 3 - Basic Elements and Syntax
Chapter 4 - Grand Tour of the Python Language
Chapter 5 - Working Like a Programmer
Part II - Building Blocks
Chapter 6 - So This String Walks into a Bar
Chapter 7 - Counting your Way to Fun and Profit
Chapter 8 - Processing Lists and Tuples
Chapter 9 - Diving into Dictionaries
Part III - Structures
Chapter 10 - Staying in Control
Chapter 11 - Fun with Functions
Chapter 12 - Building Applications with Modules and Packages
Chapter 13 - Getting Classy
Chapter 14 - Introducing New-Style Classes
Chapter 15 - Feeling Exceptional
Chapter 16 - Tackling Some Advanced Features
Part IV - Libraries
Chapter 17 - Using Python's Primary Services
Chapter 18 - Processing Text
Chapter 19 - Digging into Disk Data
Chapter 20 - Accessing the Internet
Part V - The Part of Tens
Chapter 21 - Ten Critical Python Idioms
Chapter 22 - Ten Great Resources 1
Part VI - Appendixes
Appendix A - Getting and Installing Python

Appendix B - Python Version Differences
Index
List of Figures
List of Tables
List of Listings
List of Sidebars

***Python_for_Software_Design_How_to_Think_Like_a_Computer_Scientist_2009_264p

1 The Way of the Program--1
2 Variables, Expressions, and Statements--10
3 Functions--21
4 Case Study: Interface Design--35
5 Conditionals and Recursion--46
6 Fruitful Functions---59
7 Iteration---73
8 Strings---82
9 Case Study: Word Play---95
10 Lists--103
11 Dictionaries--119
12 Tuples--133
13 Case Study: Data Structure Selection---147
14 Files---159
15 Classes and Objects--172
16 Classes and Functions--182
17 Classes and Methods---189
18 Inheritance---201
19 Case Study: Tkinter--214
Appendix---231
Index--241

***Python_for_Unix_and_Linux_System_Administration_2008_458p

Foreword---xi
Preface---xiii
1. Introduction---1
Why Python?--1
Motivation--6
The Basics--8
Executing Statements in Python---8
Using Functions in Python--12
Reusing Code with the Import Statement--16

2. IPython---21
Installing IPython--22
Basic Concepts---23
Help with Magic Functions---30
Unix Shell--34

Information Gathering--51
Automation and Shortcuts---64
Summary--69

3. Text---71
Python Built-ins and Modules--71
Log Parsing--110
ElementTree---116
Summary---120

4. Documentation and Reporting---123
Automated Information Gathering--123
Manual Information Gathering--126
Information Formatting---135
Information Distribution--141
Summary---145

5. Networking--147
Network Clients---147
Remote Procedure Call Facilities---158
SSH---164
Twisted---167
Scapy---173
Creating Scripts with Scapy--175

6. Data---177
Introduction--177
Using the OS Module to Interact with Data--178
Copying, Moving, Renaming, and Deleting Data---179
Working with Paths, Directories, and Files--181
Comparing Data---185
Merging Data--187
Pattern Matching Files and Directories--193
Wrapping Up rsync---195
Metadata: Data About Data---197
Archiving, Compressing, Imaging, and Restoring---199
Using tarfile Module to Create TAR Archives--199
Using a tarfile Module to Examine the Contents of TAR Files-------------------------------201

7. SNMP--205
Introduction--205
Brief Introduction to SNMP--205
IPython and Net-SNMP---208
Discovering a Data Center---211
Retrieving Multiple-Values with Net-SNMP--214
Creating Hybrid SNMP Tools--220
Extending Net-SNMP---222
SNMP Device Control--224
Enterprise SNMP Integration with Zenoss---225

8. OS Soup---227
Introduction---227

Cross-Platform Unix Programming in Python--228
PyInotify---238
OS X---240
Red Hat Linux Systems Administration--245
Ubuntu Administration---245
Solaris Systems Administration---245
Virtualization---246
Cloud Computing---247
Using Zenoss to Manage Windows Servers from Linux---253

9. Package Management---257
Introduction---257
Setuptools and Python Eggs--258
Using easy_install---258
easy_install Advanced Features--261
Creating Eggs---266
Entry Points and Console Scripts--270
Registering a Package with the Python Package Index--271
Distutils--273
Buildout---275
Using Buildout---276
Developing with Buildout---279
virtualenv---279
EPM Package Manager--283

10. Processes and Concurrency---289
Introduction---289
Subprocess--289
Using Supervisor to Manage Processes---298
Using Screen to Manage Processes--300
Threads in Python---301
Processes--313
Processing Module--313
Scheduling Python Processes---316
daemonizer---318
Summary--321

11. Building GUIs---323
GUI Building Theory---323
Building a Simple PyGTK App--324
Building an Apache Log Viewer Using PyGTK---326
Building an Apache Log Viewer Using Curses--330
Web Applications---334
Django---335
Conclusion---354

12. Data Persistence---357
Simple Serialization--357
Relational Serialization--376
Summary--385

13. Command Line---387

Introduction--387
Basic Standard Input Usage---388
Introduction to Optparse--389
Simple Optparse Usage Patterns--390
Unix Mashups: Integrating Shell Commands into Python Command-Line Tools---------397
Integrating Configuration Files--402
Summary---404

14. Pragmatic Examples--405
Managing DNS with Python---405
Using LDAP with OpenLDAP, Active Directory, and More with Python------------------406
Apache Log Reporting---408
FTP Mirror---415
Appendix: Callbacks---419
Index--423

***Python_How_to_Program_1e_2002_1103p

1. Introduction to Computers, Internet and the World Wide Web.
2. Introduction to Python Programming.
3. Control Structures.
4. Functions.
5. Tuples, Lists, and Dictionaries.
6. Introduction to the Common Gateway Interface (CGI).
7. Object-Based Programming: Classes and Data Abstraction.
8. Object-Oriented Programming: Inheritance and Polymorphism.
9. Operator Overloading.
10. Graphical User Interface Components: Part 1.
11. Graphical User Interface Components: Part 2.
12. Exception Handling.
13. Strings Manipulation and Regular Expressions.
14. File Processing and Serialization.
15. Extensible Markup Language (XML).
16. Python XML Processing.
17. Python Database Application Programming Interface (DB-API).
18. Process Management.
19. Multithreading.
20. Networking.
21. Security.
22. Data Structures.
23. Case Study: Multi-Tier Online Bookstore.
24. Multimedia.
25. Accessibility.
26. Bonus: Introduction to XHMTL: Part I.
27. Bonus: Introduction to XHTML: Part II.
28. Bonus: Cascading Style Sheets™ (CSS).
29. Bonus: Introduction to PHP.
Appendix A. Operator Precedence Chart.
Appendix B. ASCII Character Set.
Appendix C. Number Systems.
Appendix D. Python Development Environments.
Appendix E. Python 2.2 Resources.

Appendix F. Career Opportunities.
Appendix G. Unicode®.

***Python_Pocket_Reference_4th_Edition_2009_210p

Python Pocket Reference--1
Introduction--1
Conventions---2
Command-Line Options--4
Python Options---4
Program Specification--6
Environment Variables---7
Operational Variables--7
Command-Line Option Variables--8
Built-in Types and Operators---8
Operators and Precedence---8
Operator Usage Notes---10
Operations by Category---11
Sequence Operation Notes---15
Specific Built-in Types--16
Numbers---16
Strings--19
Unicode Strings--33
Lists--36
Dictionaries---41
Tuples---44
Files--45
Sets--49
Other Common Types---51
Type Conversions--52
Statements and Syntax---53
Syntax Rules--53
Name Rules---54
Specific Statements---56
The Assignment Statement--57
The Expression Statement---59
The print Statement---60
The if Statement---62
The while Statement--62
The for Statement--63
The pass Statement--63
The break Statement--63
The continue Statement--64
The del Statement--64
The def Statement--64
The return Statement--68
The yield Statement--68
The global Statement--70
The nonlocal Statement---70
The import Statement--71
The from Statement--72
The class Statement--73

The try Statement----75
The raise Statement----78
The assert Statement----80
The with Statement----80
Python 2.X Statements----82
Namespace and Scope Rules----82
Qualified Names: Object Namespaces----83
Unqualified Names: Lexical Scopes----83
Statically Nested Scopes----84
Object-Oriented Programming----85
Classes and Instances----85
Pseudoprivate Attributes----86
New Style Classes----87
Operator Overloading Methods----88
For All Types----88
For Collections (Sequences, Mappings)----93
For Numbers (Binary Operators)----94
For Numbers (Other Operations)----97
For Descriptors----98
For Context Managers----99
Python 2.X Operator Overloading Methods----99
Built-in Functions----102
Python 2.X Built-in Functions----119
Built-in Exceptions----124
Superclasses (Categories)----124
Specific Exceptions Raised----125
Warning Category Exceptions----129
Warnings Framework----130
Python 2.X Built-in Exceptions----131
Built-in Attributes----131
Standard Library Modules----132
The sys Module----133
The string Module----139
Module Functions and Classes----139
Constants----140
The os System Module----141
Administrative Tools----141
Portability Constants----142
Shell Commands----143
Environment Tools----144
File Descriptor Tools----145
File Pathname Tools----147
Process Control----150
The os.path Module----153
The re Pattern-Matching Module----155
Module Functions----155
Regular Expression Objects----157
Match Objects----158
Pattern Syntax----159
Object Persistence Modules----163
dbm and shelve Modules----164
pickle Module----166

The tkinter GUI Module and Tools---168
tkinter Example--168
tkinter Core Widgets--169
Common Dialog Calls--170
Additional tkinter Classes and Tools--171
Tcl/Tk-to-Python/tkinter Mappings---171
Internet Modules and Tools--173
Commonly Used Library Modules--173
Other Standard Library Modules---175
The math Module---176
The time Module---176
The datetime Module--177
Threading Modules--177
Binary Data Parsing---178
Python Portable SQL Database API---179
API Usage Example--179
Module Interface---180
Connection Objects--181
Cursor Objects---181
Type Objects and Constructors---182
Python Idioms and Hints---183
Core Language Hints---183
Environment Hints--184
Usage Hints--185
Assorted Hints---187
Index--189

***Python_Power_The_Comprehensive_Guide_2008_529p

CHAPTER 1 About Python--1
What Is Python?---1
A Brief History of Python--2
Interpreters Versus Compilers---5
When to Use (or Not Use) an Interpreted Language---------------------------------------8
Understanding Bytecodes--10
Why Use Python?---11
Object-Oriented---11
Cross Platform--11
Broad User Base--11
Well Supported in Third-Party Tools---12
Good Selection of Tools Available---12
Good Selection of Pre-built Libraries--12
Where Is Python Used?---13
How Is Python Licensed?---13
Where Do I Get Python?--14
Installing Python---14
Getting Information on Python---16
Python Communities---17
Other Software---18
And Now for Something Completely Different…---18

CHAPTER 2 Python Language Overview--19

Python Syntax--20
Comments---20
Indentation---20
Python Reserved Words---24
Decision Making and Iteration Keywords--25
Debugging Keywords---27
Package and Module Handling Keywords---27
Exception Handling Keywords--29
General Language Keywords--31
Other Keywords--32
Variable Usage--34
The Continuation Variable--36
Watching Out for Spelling Mistakes!--37
Predicates---38
Identifier Scope---39
Operators---42
Modulo Operator--44
Exponential Operator--46
Logical Operators---46
Comparative Operators--49
Bitwise Operators---51
Membership Operators and String Operators--53
Identity Operators--53
In Conclusion---53

CHAPTER 3 Tools--55
IDLE---55
File Menu--57
The Path Browser Dialog---62
Edit Menu---64
Shell Menu---70
Debug Menu---71
The Edit Window--79
Format Menu---80
Command Line Compiler---90
Creating Python Files---93
Documentation--95
In Conclusion--96

CHAPTER 4 Data Types---97
Numeric Types---98
Integers--98
Demonstrating Long Integers---99
Octal and Hexadecimal--100
Floating Point Numbers--101
Strings---103
String Variables--103
Concatenating Strings---106
Repeating Strings--107
Substrings---108
Slicing---110
String Functions---111

String Constants--112
Conversion Functions--114
Search Functions--118
Formatting Functions--120
Escape Sequences--121
Sequences---122
Lists--123
Shared References--128
Tuples--128
Dictionaries---132
Advanced Type---136
Classes and Objects--136
Complex Type--137
Generator Type---138
None Type---139
Unicode Type--140
In Conclusion---141

CHAPTER 5 Control Flow---143
Conditionals--144
The if Statement---144
The elif Statement--147
The else Statement---149
Wrapping Up the Conditionals: A Cool Example---------------------------------------150
Loops---153
The for Loop--153
The while Loop---161
In Conclusion---164

CHAPTER 6 Input and Output---165
User Input---165
The input Function--166
The raw_input Function--168
User Output--170
Formatting---172
File Input--175
File Output--177
Closing Files--179
Positioning in Files--180
Directories and Files--183
The stat Module: File Statistics--186
Command Line Arguments---190
Pickle---192
In Conclusion---195

CHAPTER 7 Functions and Modules--197
What Is a Function?--197
Defining Functions in Python---197
What Are Arguments?---200
How Do You Pass an Argument to a Function?--201
Default Arguments--203
Variable Default Arguments---205

Keyword Arguments--206
Returning Values from Functions---207
Returning Multiple Values from Functions--209
Recursive Functions--210
Passing Functions as Arguments--212
Lambda Functions--213
Variable Numbers of Arguments to a Function---215
Variable Scope in Functions---216
Using Modules---218
In Conclusion--219

CHAPTER 8 Exception Handling---221
Looking at Exceptions in Python--222
Traceback Example--223
Understanding Tracebacks--224
Exceptions--225
Catching Exceptions with try except--226
Multiple except Clauses---229
Blank except Clauses---231
The else Clauses---232
The finally Clause---234
Raising Your Own Exceptions---235
Exception Arguments---237
User-Defined Exceptions--238
Working with the Exception Information--239
exc_type--239
exc_value---240
Using the with Clause for Files--243
Re-throwing Exceptions--244
In Conclusion--246

CHAPTER 9 Object-Oriented Programming--247
A Brief History of OOP---247
What Is an Object?--248
Why Do We Use Objects?---249
Reuse---249
Ease in Debugging---250
Maintainability---250
The Attributes of Object-Oriented Development--251
Abstraction---251
Data Hiding---252
Inheritance---253
Polymorphism---255
Terminology---256
Class---256
Object--256
Attribute---257
Method--258
Message Passing---259
Event Handling--260
Derivation--260
Coupling--261

Cohesion--261
Constants---261
Other Concepts--262
In Conclusion--262

CHAPTER 10 Classes and Objects in Python--265
Python Classes--265
Properties--267
Attribute Modifying Functions---272
Private Attributes---274
Doc Strings---275
Properties--277
The self Object---279
Methods--281
Special Methods---283
Initialization---283
Termination--284
String Conversion---285
Inheritance--287
Multiple Inheritance---291
Using super---293
Polymorphism--295
Exception Classes--297
Iterators--299
Operator Overloading--301
In Conclusion---304

CHAPTER 11 The Python Library--305
Containers--305
Working with the deque Class---306
Math--312
Complex Math--313
Types---315
Strings--318
Regular Expressions--319
Patterns--320
Special Sequence Characters--323
Compiling Regular Expressions---323
Matching Strings--324
Meta Characters---326
Grouping--327
System--328
Random Number Generation---330
Dates and Times--331
Creating a New Time---332
Time Operations---332
Creating a New Date--333
Date Operations---333
Time Zone Information--335
Operating System Interface---336
System Information--336
Process Management--337

In Conclusion--341

CHAPTER 12 The GUI — TkInter--343
What Is TkInter?---343
Terms and Conditions---343
Event Handling--344
Callbacks---344
Widgets---345
Layout Managers--345
Working with TkInter--346
Creating a Label---347
Frame Widgets and Centering--349
An Application with a Button---351
Working with Entry Fields and Grid Layouts--353
Creating a Class to Handle User Interfaces--356
Working with List Boxes--358
Scrolling a List Box--361
Menus--363
Context Menus---366
Scale Widgets--367
RadioButtons and CheckButton--370
Text Widgets--373
In Conclusion---375

CHAPTER 13 The Web Server—Apache---377
Setting Up Apache--377
Testing Apache--378
Your First Python CGI Script: Hello Apache---379
Examining the Hello Python Script---380
The cgi-bin Directory--381
A Script for Displaying the Environment--382
Receiving Data from an HTML File---384
Sending Data to an HTML File--387
How It All Works---390
Dynamic HTML Displays Based on User Input--------------------------------------391
HTML Elements--396
Cookies--399
Uploading Files---402
Redirection--403
Error Handling--405
In Conclusion---406

CHAPTER 14 Working with Databases--407
What Is a Database?--407
Simple Database Terminology---408
What Is MySQL?--409
Downloading and Installing--409
Creating a New Database---410
Creating a New User---414
Opening an Existing Database---415
Writing to a Database--417
Reading from a Database---421

Updating a Database--424
Deleting from a Database---427
Searching a Database--430
In Conclusion--436

CHAPTER 15 Putting It All Together---437
Designing the Application--437
Program Flow--437
User Interface Design---438
Database Design---439
Implementing the Database Tables--440
Implementing the Forms---442
Adding Reviews--449
Adding the Review to the Database---452
Listing the Reviews---456
Deleting Books--459
In Conclusion--462

CHAPTER 16 Python and Graphics---463
The PIL Library---463
Downloading---464
Installing---464
Verifying Your Installation---465
Creating a New Image--465
Function Parameters---466
Drawing on the Image---467
Drawing the Image---468
Displaying the Image--470
Saving the Image--470
Loading an Existing Image--472
Displaying Text---474
Identifying an Image--476
Rotating an Image---478
Postscript Printing--480
Creating Thumbnails---481
In Conclusion--483
Index--485

***Python_Programming_An_Introduction_to_Computer_Science_v1_0rc2_2002_261p

1 Computers and Programs--1
1 1 The Universal Machine--1
1 2 Program Power--2
1 3 What is Computer Science?--2
1 4 Hardware Basics--3
1 5 Programming Languages---4
1 6 The Magic of Python--5
1 7 Inside a Python Program--8
1 8 Chaos and Computers---10
1 9 Exercises--11

2 Writing Simple Programs--13
2 1 The Software Development Process--13
2 2 Example Program: Temperature Converter--13
2 3 Elements of Programs---15
2 3 1 Names---15
2 3 2 Expressions--15
2 4 Output Statements--16
2 5 Assignment Statements--17
2 5 1 Simple Assignment--17
2 5 2 Assigning Input--18
2 5 3 Simultaneous Assignment--19
2 6 Definite Loops---20
2 7 Example Program: Future Value---22
2 8 Exercises--24

3 Computing with Numbers--25
3 1 Numeric Data Types--25
3 2 Using the Math Library---27
3 3 Accumulating Results: Factorial---28
3 4 The Limits of Int--31
3 5 Handling Large Numbers: Long Ints--32
3 6 Type Conversions---34
3 7 Exercises---35

4 Computing with Strings--39
4 1 The String Data Type--39
4 2 Simple String Processing--41
4 3 Strings and Secret Codes--43
4 3 1 String Representation--43
4 3 2 Programming an Encoder---44
4 3 3 Programming a Decoder---45
4 3 4 Other String Operations---48
4 3 5 From Encoding to Encryption---48
4 4 Output as String Manipulation---49
4 4 1 Converting Numbers to Strings---49
4 4 2 String Formatting---50
4 4 3 Better Change Counter---51
4 5 File Processing---52
4 5 1 Multi-Line Strings--52
4 5 2 File Processing--53
4 5 3 Example Program: Batch Usernames---55
4 5 4 Coming Attraction: Objects--56
4 6 Exercises---57

5 Objects and Graphics---61
5 1 The Object of Objects--61
5 2 Graphics Programming--62
5 3 Using Graphical Objects---64
5 4 Graphing Future Value---68
5 5 Choosing Coordinates--73
5 6 Interactive Graphics--75

5 6 1 Getting Mouse Clicks--75
5 6 2 Handling Textual Input---76
5 7 Graphics Module Reference---79
5 7 1 GraphWin Objects---79
5 7 2 Graphics Objects---79
5 7 3 Entry Objects--81
5 7 4 Displaying Images---81
5 7 5 Generating Colors--81
5 8 Exercises--82

6 Defining Functions--85
6 1 The Function of Functions---85
6 2 Functions, Informally---86
6 3 Future Value with a Function--89
6 4 Functions and Parameters: The Gory Details--90
6 5 Functions that Return Values--93
6 6 Functions and Program Structure---95
6 7 Exercises--97

7 Control Structures, Part 1---101
7 1 Simple Decisions---101
7 1 1 Example: Temperature Warnings---101
7 1 2 Forming Simple Conditions---103
7 1 3 Example: Conditional Program Execution---104
7 2 Two-Way Decisions--105
7 3 Multi-Way Decisions--107
7 4 Exception Handling--109
7 5 Study in Design: Max of Three---112
7 5 1 Strategy 1: Compare Each to All---112
7 5 2 Strategy 2: Decision Tree--113
7 5 3 Strategy 3: Sequential Processing--114
7 5 4 Strategy 4: Use Python--116
7 5 5 Some Lessons---116
7 6 Exercises---116
8 Control Structures, Part 2--119
8 1 For Loops: A Quick Review--119
8 2 Indefinite Loops--120
8 3 Common Loop Patterns--121
8 3 1 Interactive Loops--121
8 3 2 Sentinel Loops--123
8 3 3 File Loops--125
8 3 4 Nested Loops---126
8 4 Computing with Booleans---127
8 4 1 Boolean Operators--127
8 4 2 Boolean Algebra---129
8 5 Other Common Structures---130
8 5 1 Post-Test Loop---130
8 5 2 Loop and a Half--132
8 5 3 Boolean Expressions as Decisions---132
8 6 Exercises---134

9 Simulation and Design--137

9 1 Simulating Racquetball--137
9 1 1 A Simulation Problem--137
9 1 2 Program Specification--138
9 2 Random Numbers--138
9 3 Top-Down Design---140
9 3 1 Top-Level Design--140
9 3 2 Separation of Concerns--141
9 3 3 Second-Level Design---142
9 3 4 Designing simNGames---143
9 3 5 Third-Level Design--144
9 3 6 Finishing Up--146
9 3 7 Summary of the Design Process---148
9 4 Bottom-Up Implementation---148
9 4 1 Unit Testing--148
9 4 2 Simulation Results---149
9 5 Other Design Techniques--150
9 5 1 Prototyping and Spiral Development---150
9 5 2 The Art of Design--151
9 6 Exercises---152

10 Defining Classes--155
10 1 Quick Review of Objects---155
10 2 Example Program: Cannonball--156
10 2 1 Program Specification---156
10 2 2 Designing the Program---156
10 2 3 Modularizing the Program--159
10 3 Defining New Classes--159
10 3 1 Example: Multi-Sided Dice---160
10 3 2 Example: The Projectile Class--162
10 4 Objects and Encapsulation---164
10 4 1 Encapsulating Useful Abstractions--164
10 4 2 Putting Classes in Modules--164
10 5 Widget Objects--166
10 5 1 Example Program: Dice Roller--166
10 5 2 Building Buttons---166
10 5 3 Building Dice--169
10 5 4 The Main Program---172
10 6 Exercises---173

11 Data Collections--177
11 1 Example Problem: Simple Statistics---177
11 2 Applying Lists--178
11 2 1 Lists are Sequences--178
11 2 2 Lists vs Strings--179
11 2 3 List Operations--180
11 3 Statistics with Lists--181
11 4 Combining Lists and Classes---184
11 5 Case Study: Python Calculator---188
11 5 1 A Calculator as an Object--188
11 5 2 Constructing the Interface--188
11 5 3 Processing Buttons---190
11 6 Non-Sequential Collections--193

11 6 1 Dictionary Basics--193
11 6 2 Dictionary Operations---194
11 6 3 Example Program: Word Frequency---194
11 7 Exercises---198

12 Object-Oriented Design---201
12 1 The Process of OOD--201
12 2 Case Study: Racquetball Simulation--202
12 2 1 Candidate Objects and Methods---203
12 2 2 Implementing SimStats--203
12 2 3 Implementing RBallGame--205
12 2 4 Implementing Player--207
12 2 5 The Complete Program--207
12 3 Case Study: Dice Poker---210
12 3 1 Program Specification--210
12 3 2 Identifying Candidate Objects--210
12 3 3 Implementing the Model---211
12 3 4 A Text-Based UI--214
12 3 5 Developing a GUI--216
12 4 OO Concepts--221
12 4 1 Encapsulation--221
12 4 2 Polymorphism---222
12 4 3 Inheritance--222
12 5 Exercises--223

13 Algorithm Analysis and Design--225
13 1 Searching--225
13 1 1 A Simple Searching Problem---225
13 1 2 Strategy 1: Linear Search---226
13 1 3 Strategy 2: Binary Search---226
13 1 4 Comparing Algorithms--227
13 2 Recursive Problem-Solving--228
13 2 1 Recursive Definitions--229
13 2 2 Recursive Functions--230
13 2 3 Recursive Search--230
13 3 Sorting Algorithms---231
13 3 1 Naive Sorting: Selection Sort---231
13 3 2 Divide and Conquer: Merge Sort---232
13 3 3 Comparing Sorts--234
13 4 Hard Problems---235
13 4 1 Towers of Hanoi---236
13 4 2 The Halting Problem--239
13 4 3 Conclusion--241

***Python_Programming_Tutorial_2000_195p

0001 Lesson 1, Getting Started
0004 Let's Do Numbers
0008 Variables and Identifiers
0012 Strings, Part I
0016 Writing and Using Scripts
0020 Program Construction

0024 Strings, Part II
0028 Lists, Part I
0032 Lists, Part II
0036 Tuples, Index and Slice
0040 Nested Tuples
0044 Empty and Single-Item Tuples
0048 Unpacking Tuples
0052 Indexing Nested Tuples
0056 Slicing Nested Tuples
0060 Indirection
0064 Getting Started with Dictionaries
0068 Valid Keys, Key Lists, Iteration
0072 Using Tuples as Keys
0076 Nesting, Sorting, Deleting, and Membership Testing Dictionary Elements

***Python_Scripting_for_Computational_Science_ThirdEdition_2008_767p

1 Introduction---1
1 1 Scripting versus Traditional Programming--1
1 1 1 Why Scripting is Useful in Computational Science-------------------------------------2
1 1 2 Classification of Programming Languages---4
1 1 3 Productive Pairs of Programming Languages---5
1 1 4 Gluing Existing Applications--6
1 1 5 Scripting Yields Shorter Code--7
1 1 6 Efficiency---8
1 1 7 Type-Specification (Declaration) of Variables---9
1 1 8 Flexible Function Interfaces---11
1 1 9 Interactive Computing---12
1 1 10 Creating Code at Run Time--13
1 1 11 Nested Heterogeneous Data Structures--14
1 1 12 GUI Programming--16
1 1 13 Mixed Language Programming---17
1 1 14 When to Choose a Dynamically Typed Language-----------------------------------19
1 1 15 Why Python?---20
1 1 16 Script or Program?---21
1 2 Preparations for Working with This Book---22

2 Getting Started with Python Scripting---27
2 1 A Scientific Hello World Script---27
2 1 1 Executing Python Scripts--28
2 1 2 Dissection of the Scientific Hello World Script--29
2 2 Working with Files and Data--32
2 2 1 Problem Specification---32
2 2 2 The Complete Code---33
2 2 3 Dissection---33
2 2 4 Working with Files in Memory---36
2 2 5 Array Computing--37
2 2 6 Interactive Computing and Debugging---39
2 2 7 Efficiency Measurements--42
2 2 8 Exercises--43

2 3 Gluing Stand-Alone Applications---46
2 3 1 The Simulation Code--47
2 3 2 Using Gnuplot to Visualize Curves---49
2 3 3 Functionality of the Script---50
2 3 4 The Complete Code--51
2 3 5 Dissection--53
2 3 6 Exercises---55
2 4 Conducting Numerical Experiments--58
2 4 1 Wrapping a Loop Around Another Script---59
2 4 2 Generating an HTML Report--60
2 4 3 Making Animations---61
2 4 4 Varying Any Parameter---63
2 5 File Format Conversion--66
2 5 1 A Simple Read/Write Script--66
2 5 2 Storing Data in Dictionaries and Lists---68
2 5 3 Making a Module with Functions---69
2 5 4 Exercises---71

3 Basic Python---73
3 1 Introductory Topics---74
3 1 1 Recommended Python Documentation--74
3 1 2 Control Statements--75
3 1 3 Running Applications--76
3 1 4 File Reading and Writing--78
3 1 5 Output Formatting---79
3 2 Variables of Different Types--81
3 2 1 Boolean Types---81
3 2 2 The None Variable---82
3 2 3 Numbers and Numerical Expressions---82
3 2 4 Lists and Tuples--84
3 2 5 Dictionaries--90
3 2 6 Splitting and Joining Text--94
3 2 7 String Operations---95
3 2 8 Text Processing---96
3 2 9 The Basics of a Python Class--98
3 2 10 Copy and Assignment--100
3 2 11 Determining a Variable’s Type---104
3 2 12 Exercises---106
3 3 Functions---110
3 3 1 Keyword Arguments--111
3 3 2 Doc Strings--112
3 3 3 Variable Number of Arguments---112
3 3 4 Call by Reference---114
3 3 5 Treatment of Input and Output Arguments--115
3 3 6 Function Objects--116
3 4 Working with Files and Directories--117
3 4 1 Listing Files in a Directory---118
3 4 2 Testing File Types---118
3 4 3 Removing Files and Directories---119
3 4 4 Copying and Renaming Files---120
3 4 5 Splitting Pathnames---121
3 4 6 Creating and Moving to Directories---122

3 4 7 Traversing Directory Trees--122
3 4 8 Exercises---125
4 Numerical Computing in Python---131
4 1 A Quick NumPy Primer--132
4 1 1 Creating Arrays---132
4 1 2 Array Indexing--136
4 1 3 Loops over Arrays---138
4 1 4 Array Computations---139
4 1 5 More Array Functionality--142
4 1 6 Type Testing--144
4 1 7 Matrix Objects--145
4 1 8 Exercises---146
4 2 Vectorized Algorithms--147
4 2 1 From Scalar to Array in Function Arguments---147
4 2 2 Slicing---149
4 2 3 Exercises---150
4 3 More Advanced Array Computing--151
4 3 1 Random Numbers---152
4 3 2 Linear Algebra--153
4 3 3 Plotting--154
4 3 4 Example: Curve Fitting---157
4 3 5 Arrays on Structured Grids--159
4 3 6 File I/O with NumPy Arrays--163
4 3 7 Functionality in the Numpyutils Module---165
4 3 8 Exercises---168
4 4 Other Tools for Numerical Computations--173
4 4 1 The ScientificPython Package--173
4 4 2 The SciPy Package---178
4 4 3 The Python–Matlab Interface--183
4 4 4 Symbolic Computing in Python---184
4 4 5 Some Useful Python Modules---186

5 Combining Python with Fortran, C, and C---189
5 1 About Mixed Language Programming---189
5 1 1 Applications of Mixed Language Programming--------------------------------------190
5 1 2 Calling C from Python---190
5 1 3 Automatic Generation of Wrapper Code---192
5 2 Scientific Hello World Examples--194
5 2 1 Combining Python and Fortran---195
5 2 2 Combining Python and C---201
5 2 3 Combining Python and C Functions---208
5 2 4 Combining Python and C Classes---210
5 2 5 Exercises---214
5 3 A Simple Computational Steering Example--215
5 3 1 Modified Time Loop for Repeated Simulations--216
5 3 2 Creating a Python Interface---217
5 3 3 The Steering Python Script--218
5 3 4 Equipping the Steering Script with a GUI--222
5 4 Scripting Interfaces to Large Libraries---223
6 Introduction to GUI Programming---227
6 1 Scientific Hello World GUI---228
6 1 1 Introductory Topics---228

6 1 2 The First Python/Tkinter Encounter--230
6 1 3 Binding Events---233
6 1 4 Changing the Layout--234
6 1 5 The Final Scientific Hello World GUI---238
6 1 6 An Alternative to Tkinter Variables--240
6 1 7 About the Pack Command--241
6 1 8 An Introduction to the Grid Geometry Manager---243
6 1 9 Implementing a GUI as a Class--245
6 1 10 A Simple Graphical Function Evaluator--247
6 1 11 Exercises---248
6 2 Adding GUIs to Scripts---250
6 2 1 A Simulation and Visualization Script with a GUI------------------------------------250
6 2 2 Improving the Layout---253
6 2 3 Exercises---256
6 3 A List of Common Widget Operations--257
6 3 1 Frame--259
6 3 2 Label---260
6 3 3 Button--262
6 3 4 Text Entry--262
6 3 5 Balloon Help---264
6 3 6 Option Menu---265
6 3 7 Slider--265
6 3 8 Check Button--266
6 3 9 Making a Simple Megawidget---266
6 3 10 Menu Bar---267
6 3 11 List Data---269
6 3 12 Listbox---269
6 3 13 Radio Button---272
6 3 14 Combo Box--274
6 3 15 Message Box--275
6 3 16 User-Defined Dialogs--277
6 3 17 Color-Picker Dialogs--278
6 3 18 File Selection Dialogs--279
6 3 19 Toplevel---280
6 3 20 Some Other Types of Widgets--281
6 3 21 Adapting Widgets to the User’s Resize Actions-----------------------------------282
6 3 22 Customizing Fonts and Colors--284
6 3 23 Widget Overview--286
6 3 24 Exercises--289
7 Web Interfaces and CGI Programming--295
7 1 Introductory CGI Scripts--296
7 1 1 Web Forms and CGI Scripts--297
7 1 2 Generating Forms in CGI Scripts--299
7 1 3 Debugging CGI Scripts--301
7 1 4 A General Shell Script Wrapper for CGI Scripts------------------------------------302
7 1 5 Security Issues---304
7 2 Adding Web Interfaces to Scripts---306
7 2 1 A Class for Form Parameters--306
7 2 2 Calling Other Programs---308
7 2 3 Running Simulations--309
7 2 4 Getting a CGI Script to Work--311
7 2 5 Using Web Applications from Scripts---313

7 2 6 Exercises--316

8 Advanced Python---319
8 1 Miscellaneous Topics---319
8 1 1 Parsing Command-Line Arguments---319
8 1 2 Platform-Dependent Operations--322
8 1 3 Run-Time Generation of Code--323
8 1 4 Exercises--324
8 2 Regular Expressions and Text Processing--326
8 2 1 Motivation---326
8 2 2 Special Characters---329
8 2 3 Regular Expressions for Real Numbers---331
8 2 4 Using Groups to Extract Parts of a Text---334
8 2 5 Extracting Interval Limits---335
8 2 6 Extracting Multiple Matches---339
8 2 7 Splitting Text---344
8 2 8 Pattern-Matching Modifiers--345
8 2 9 Substitution and Backreferences---347
8 2 10 Example: Swapping Arguments in Function Calls--348
8 2 11 A General Substitution Script--351
8 2 12 Debugging Regular Expressions---353
8 2 13 Exercises--354
8 3 Tools for Handling Data in Files---362
8 3 1 Writing and Reading Python Data Structures--362
8 3 2 Pickling Objects---364
8 3 3 Shelving Objects---366
8 3 4 Writing and Reading Zip and Tar Archive Files--366
8 3 5 Downloading Internet Files--367
8 3 6 Binary Input/Output---368
8 3 7 Exercises--371
8 4 A Database for NumPy Arrays--371
8 4 1 The Structure of the Database---371
8 4 2 Pickling---374
8 4 3 Formatted ASCII Storage--375
8 4 4 Shelving---376
8 4 5 Comparing the Various Techniques--377
8 5 Scripts Involving Local and Remote Hosts---378
8 5 1 Secure Shell Commands--378
8 5 2 Distributed Simulation and Visualization---380
8 5 3 Client/Server Programming--382
8 5 4 Threads--382
8 6 Classes--384
8 6 1 Class Programming---384
8 6 2 Checking the Class Type--388
8 6 3 Private Data--389
8 6 4 Static Data---390
8 6 5 Special Attributes---390
8 6 6 Special Methods--391
8 6 7 Multiple Inheritance--392
8 6 8 Using a Class as a C-like Structure--393
8 6 9 Attribute Access via String Names--394
8 6 10 New-Style Classes---394

8 6 11 Implementing Get/Set Functions via Properties------------------------------------395
8 6 12 Subclassing Built-in Types--396
8 6 13 Building Class Interfaces at Run Time---399
8 6 14 Building Flexible Class Interfaces---403
8 6 15 Exercises--409
8 7 Scope of Variables--413
8 7 1 Global, Local, and Class Variables--413
8 7 2 Nested Functions--415
8 7 3 Dictionaries of Variables in Namespaces---416
8 8 Exceptions--418
8 8 1 Handling Exceptions---419
8 8 2 Raising Exceptions--420
8 9 Iterators---421
8 9 1 Constructing an Iterator--421
8 9 2 A Pointwise Grid Iterator---423
8 9 3 A Vectorized Grid Iterator--427
8 9 4 Generators--428
8 9 5 Some Aspects of Generic Programming--432
8 9 6 Exercises---436
8 10 Investigating Efficiency---437
8 10 1 CPU-Time Measurements---437
8 10 2 Profiling Python Scripts---441
8 10 3 Optimization of Python Code--442
8 10 4 Case Study on Numerical Efficiency---445
9 Fortran Programming with NumPy Arrays---451
9 1 Problem Definition--451
9 2 Filling an Array in Fortran--453
9 2 1 The Fortran Subroutine--454
9 2 2 Building and Inspecting the Extension Module---455
9 3 Array Storage Issues--457
9 3 1 Generating an Erroneous Interface---457
9 3 2 Array Storage in C and Fortran--459
9 3 3 Input and Output Arrays as Function Arguments--459
9 3 4 F2PY Interface Files--466
9 3 5 Hiding Work Arrays--470
9 4 Increasing Callback Efficiency--470
9 4 1 Callbacks to Vectorized Python Functions--471
9 4 2 Avoiding Callbacks to Python--473
9 4 3 Compiled Inline Callback Functions--474
9 5 Summary---478
9 6 Exercises---479

10 C and C Programming with NumPy Arrays---483
10 1 Automatic Interfacing of C/C Code---484
10 1 1 Using F2PY---485
10 1 2 Using Instant--486
10 1 3 Using Weave--487
10 2 C Programming with NumPy Arrays---488
10 2 1 The Basics of the NumPy C API--489
10 2 2 The Handwritten Extension Code---491
10 2 3 Sending Arguments from Python to C---492
10 2 4 Consistency Checks---493

10 2 5 Computing Array Values---494
10 2 6 Returning an Output Array---496
10 2 7 Convenient Macros--497
10 2 8 Module Initialization--499
10 2 9 Extension Module Template--500
10 2 10 Compiling, Linking, and Debugging the Module-----------------------------------502
10 2 11 Writing a Wrapper for a C Function--503
10 3 C Programming with NumPy Arrays---506
10 3 1 Wrapping a NumPy Array in a C Object--506
10 3 2 Using SCXX---508
10 3 3 NumPy–C Class Conversion--511
10 4 Comparison of the Implementations--519
10 4 1 Efficiency---519
10 4 2 Error Handling--523
10 4 3 Summary--524
10 5 Exercises---525
11 More Advanced GUI Programming--529
11 1 Adding Plot Areas in GUIs---529
11 1 1 The BLT Graph Widget---530
11 1 2 Animation of Functions in BLT Graph Widgets--536
11 1 3 Other Tools for Making GUIs with Plots--538
11 1 4 Exercises---539
11 2 Event Bindings--541
11 2 1 Binding Events to Functions with Arguments--542
11 2 2 A Text Widget with Tailored Keyboard Bindings------------------------------------544
11 2 3 A Fancy List Widget---547
11 3 Animated Graphics with Canvas Widgets--550
11 3 1 The First Canvas Encounter---551
11 3 2 Coordinate Systems---552
11 3 3 The Mathematical Model Class--556
11 3 4 The Planet Class---557
11 3 5 Drawing and Moving Planets--559
11 3 6 Dragging Planets to New Positions---560
11 3 7 Using Pmw's Scrolled Canvas Widget--564
11 4 Simulation and Visualization Scripts--566
11 4 1 Restructuring the Script---567
11 4 2 Representing a Parameter by a Class---569
11 4 3 Improved Command-Line Script---583
11 4 4 Improved GUI Script---584
11 4 5 Improved CGI Script---585
11 4 6 Parameters with Physical Dimensions---586
11 4 7 Adding a Curve Plot Area--588
11 4 8 Automatic Generation of Scripts--589
11 4 9 Applications of the Tools---590
11 4 10 Allowing Physical Units in Input Files---596
11 4 11 Converting Input Files to GUIs--601

12 Tools and Examples--605
12 1 Running Series of Computer Experiments---605
12 1 1 Multiple Values of Input Parameters--606
12 1 2 Implementation Details--609
12 1 3 Further Applications--614

12 2 Tools for Representing Functions--618
12 2 1 Functions Defined by String Formulas--618
12 2 2 A Unified Interface to Functions--623
12 2 3 Interactive Drawing of Functions--629
12 2 4 A Notebook for Selecting Functions---633
12 3 Solving Partial Differential Equations---640
12 3 1 Numerical Methods for 1D Wave Equations--641
12 3 2 Implementations of 1D Wave Equations---644
12 3 3 Classes for Solving 1D Wave Equations---651
12 3 4 A Problem Solving Environment---657
12 3 5 Numerical Methods for 2D Wave Equations--663
12 3 6 Implementations of 2D Wave Equations--666
12 3 7 Exercises--675

A Setting up the Required Software Environment--677
A 1 Installation on Unix Systems--677
A 1 1 A Suggested Directory Structure--677
A 1 2 Setting Some Environment Variables---678
A 1 3 Installing Tcl/Tk and Additional Modules---679
A 1 4 Installing Python---680
A 1 5 Installing Python Modules--681
A 1 6 Installing Gnuplot---683
A 1 7 Installing SWIG--684
A 1 8 Summary of Environment Variables---684
A 1 9 Testing the Installation of Scripting Utilities-------------------------------------685
A 2 Installation on Windows Systems---685

B Elements of Software Engineering---689
B 1 Building and Using Modules---689
B 1 1 Single-File Modules---689
B 1 2 Multi-File Modules--693
B 1 3 Debugging and Troubleshooting--694
B 2 Tools for Documenting Python Software--696
B 2 1 Doc Strings--696
B 2 2 Tools for Automatic Documentation--698
B 3 Coding Standards--702
B 3 1 Style Guide--702
B 3 2 Pythonic Programming---706
B 4 Verification of Scripts---711
B 4 1 Automating Regression Tests--711
B 4 2 Implementing a Tool for Regression Tests--715
B 4 3 Writing a Test Script---719
B 4 4 Verifying Output from Numerical Computations----------------------------------720
B 4 5 Automatic Doc String Testing--724
B 4 6 Unit Testing---726
B 5 Version Control Management--728
B 5 1 Mercurial--729
B 5 2 Subversion--732
B 6 Exercises--734

Bibliography--739
Index---741

List of--Exercises

Exercise 2 1 Become familiar with the electronic documentation---------------------------------31
Exercise 2 2 Extend Exercise 2 1 with a loop---43
Exercise 2 3 Find five errors in a script---43
Exercise 2 4 Basic use of control structures---43
Exercise 2 5 Use standard input/output instead of files---------------------------------------44
Exercise 2 6 Read streams of (x, y) pairs from the command line--------------------------------45
Exercise 2 7 Test for specific exceptions---45
Exercise 2 8 Sum columns in a file---45
Exercise 2 9 Estimate the chance of an event in a dice game-------------------------------------45
Exercise 2 10 Determine if you win or loose a hazard game---------------------------------------46
Exercise 2 11 Generate an HTML report from the simviz1 py script------------------------------55
Exercise 2 12 Generate a LATEX report from the simviz1 py script------------------------------56
Exercise 2 13 Compute time step values in the simviz1 py script--------------------------------57
Exercise 2 14 Use Matlab for curve plotting in the simviz1 py script-----------------------------57
Exercise 2 15 Combine curves from two simulations in one plot---------------------------------61
Exercise 2 16 Combine two-column data files to a multi-column file-----------------------------71
Exercise 2 17 Read/write Excel data files in Python--72
Exercise 3 1 Write format specifications in printf-style---106
Exercise 3 2 Write your own function for joining strings---106
Exercise 3 3 Write an improved function for joining strings-------------------------------------106
Exercise 3 4 Never modify a list you are iterating on---107
Exercise 3 5 Make a specialized sort function--107
Exercise 3 6 Check if your system has a specific program---------------------------------------108
Exercise 3 7 Find the paths to a collection of programs---108
Exercise 3 8 Use Exercise 3 7 to improve the simviz1 py script---------------------------------109
Exercise 3 9 Use Exercise 3 7 to improve the loop4simviz2 py script--------------------------109
Exercise 3 10 Find the version number of a utility--109
Exercise 3 11 Automate execution of a family of similar commands-----------------------------125
Exercise 3 12 Remove temporary files in a directory tree--125
Exercise 3 13 Find old and large files in a directory tree---126
Exercise 3 14 Remove redundant files in a directory tree--126
Exercise 3 15 Annotate a filename with the current date--127
Exercise 3 16 Automatic backup of recently modified files---------------------------------------127
Exercise 3 17 Search for a text in files with certain extensions-----------------------------------128
Exercise 3 18 Search directories for plots and make HTML report------------------------------128
Exercise 3 19 Fix Unix/Windows Line Ends---129
Exercise 4 1 Matrix-vector multiply with NumPy arrays---146
Exercise 4 2 Work with slicing and matrix multiplication---146
Exercise 4 3 Assignment and in-place NumPy array modifications-----------------------------147
Exercise 4 4 Vectorize a constant function---150
Exercise 4 5 Vectorize a numerical integration rule--150
Exercise 4 6 Vectorize a formula containing an if condition-------------------------------------151
Exercise 4 7 Slicing of two-dimensional arrays---151
Exercise 4 8 Implement Exercise 2 9 using NumPy arrays---------------------------------------168
Exercise 4 9 Implement Exercise 2 10 using NumPy arrays-------------------------------------169
Exercise 4 10 Replace lists by NumPy arrays in convert2 py------------------------------------169
Exercise 4 11 Use Easyviz in the simviz1 py script--169
Exercise 4 12 Extension of Exercise 2 8---169
Exercise 4 13 NumPy arrays and binary files--169

Exercise 4 14 One-dimensional Monte Carlo integration--169
Exercise 4 15 Higher-dimensional Monte Carlo integration---------------------------------------170
Exercise 4 16 Load data file into NumPy array and visualize-------------------------------------171
Exercise 4 17 Analyze trends in the data from Exercise 4 16--------------------------------------171
Exercise 4 18 Evaluate a function over a 3D grid--171
Exercise 4 19 Evaluate a function over a plane or line in a 3D grid------------------------------172
Exercise 5 1 Implement a numerical integration rule in F77--------------------------------------214
Exercise 5 2 Implement a numerical integration rule in C--214
Exercise 5 3 Implement a numerical integration rule in C--214
Exercise 6 1 Modify the Scientific Hello World GUI---248
Exercise 6 2 Change the layout of the GUI in Exercise 6 1---------------------------------------248
Exercise 6 3 Control a layout with the grid geometry manager-----------------------------------249
Exercise 6 4 Make a demo of Newton’s method---250
Exercise 6 5 Program with Pmw EntryField in hwGUI10 py--------------------------------------256
Exercise 6 6 Program with Pmw EntryField in simvizGUI2 py----------------------------------256
Exercise 6 7 Replace Tkinter variables by set/get-like functions---------------------------------256
Exercise 6 8 Use simviz1 py as a module in simvizGUI2 py-------------------------------------256
Exercise 6 9 Apply Matlab for visualization in simvizGUI2 py----------------------------------257
Exercise 6 10 Program with Pmw OptionMenu in simvizGUI2 py-------------------------------289
Exercise 6 11 Study the nonlinear motion of a pendulum---289
Exercise 6 12 Add error handling with an associated message box-------------------------------290
Exercise 6 13 Add a message bar to a balloon help--290
Exercise 6 14 Select a file from a list and perform an action---------------------------------------291
Exercise 6 15 Make a GUI for finding and selecting font names---------------------------------291
Exercise 6 16 Launch a GUI when command-line options are missing--------------------------292
Exercise 6 17 Write a GUI for Exercise 3 14---292
Exercise 6 18 Write a GUI for selecting files to be plotted--293
Exercise 6 19 Write an easy-to-use GUI generator--293
Exercise 7 1 Write a CGI debugging tool---316
Exercise 7 2 Make a web calculator--316
Exercise 7 3 Make a web application for registering participants--------------------------------317
Exercise 7 4 Make a web application for numerical experiments--------------------------------317
Exercise 7 5 Become a “nobody” user on a web server--317
Exercise 8 1 Use the getopt/optparse module in simviz1 py-------------------------------------324
Exercise 8 2 Store command-line options in a dictionary---------------------------------------325
Exercise 8 3 Turn files with commands into Python variables-----------------------------------325
Exercise 8 4 A grep script--354
Exercise 8 5 Experiment with a regex for real numbers---355
Exercise 8 6 Find errors in regular expressions--355
Exercise 8 7 Generate data from a user-supplied formula--356
Exercise 8 8 Explain the behavior of regular expressions---356
Exercise 8 9 Edit extensions in filenames--357
Exercise 8 10 Extract info from a program code--357
Exercise 8 11 Regex for splitting a pathname--357
Exercise 8 12 Rename a collection of files according to a pattern--------------------------------358
Exercise 8 13 Reimplement the re findall function--358
Exercise 8 14 Interpret a regex code and find programming errors-------------------------------358
Exercise 8 15 Automatic fine tuning of PostScript figures--------------------------------------359
Exercise 8 16 Transform a list of lines to a list of paragraphs-------------------------------------360
Exercise 8 17 Copy computer codes into documents---360
Exercise 8 18 A very useful script for all writers---361
Exercise 8 19 Read Fortran 90 files with namelists--361

Exercise 8 20 Automatic update of function calls in C files--------------------------------------361
Exercise 8 21 Read/write (x, y) pairs from/to binary files--371
Exercise 8 22 Use the XDR format in the script from Exercise 8 21-----------------------------371
Exercise 8 23 Archive all files needed in a LATEX document------------------------------------371
Exercise 8 24 Using a web site for distributed simulation---381
Exercise 8 25 Convert data structures to/from strings---409
Exercise 8 26 Implement a class for vectors in 3D---410
Exercise 8 27 Extend the class from Exericse 8 26--410
Exercise 8 28 Make a tuple with cyclic indices---411
Exercise 8 29 Make a dictionary type with ordered keys---411
Exercise 8 30 Make a smarter integration function---412
Exercise 8 31 Equip class Grid2D with subscripting---412
Exercise 8 32 Extend the functionality of class Grid2D--412
Exercise 8 33 Make a boundary iterator in a 2D grid--436
Exercise 8 34 Make a generator for odd numbers---436
Exercise 8 35 Make a class for sparse vectors--436
Exercise 9 1 Extend Exercise 5 1 with a callback to Python-------------------------------------479
Exercise 9 2 Compile callback functions in Exercise 9 1---479
Exercise 9 3 Smoothing of time series--480
Exercise 9 4 Smoothing of 3D data--480
Exercise 9 5 Type incompatibility between Python and Fortran--------------------------------481
Exercise 9 6 Problematic callbacks to Python from Fortran------------------------------------481
Exercise 9 7 Array look-up efficiency: Python vs Fortran---------------------------------------482
Exercise 10 1 Extend Exercise 5 2 or 5 3 with a callback to Python-----------------------------525
Exercise 10 2 Investigate the efficiency of vector operations-------------------------------------525
Exercise 10 3 Debug a C extension module---525
Exercise 10 4 Make callbacks to vectorized Python functions-----------------------------------526
Exercise 10 5 Avoid Python callbacks in extension modules-------------------------------------526
Exercise 10 6 Extend Exercise 9 4 with C and C code---526
Exercise 10 7 Apply SWIG to an array class in C---526
Exercise 10 8 Build a dictionary in C--526
Exercise 10 9 Make a C module for computing random numbers-------------------------------527
Exercise 10 10 Almost automatic generation of C extension modules-----------------------------527
Exercise 10 11 Introduce C array objects in Exercise 10 10--528
Exercise 10 12 Introduce SCXX in Exercise 10 11--528
Exercise 11 1 Incorporate a BLT graph widget in simviz1 py-------------------------------------539
Exercise 11 2 Plot a two-column datafile in a Pmw Blt widget------------------------------------539
Exercise 11 3 Use a BLT graph widget in simvizGUI2 py--539
Exercise 11 4 Extend Exercise 11 3 to handle multiple curves-----------------------------------539
Exercise 11 5 Use a BLT graph widget in Exercise 6 4---539
Exercise 11 6 Interactive dump of snapshot plots in an animation------------------------------540
Exercise 11 7 Extend the animate py GUI--540
Exercise 11 8 Animate a curve in a BLT graph widget--541
Exercise 11 9 Add animations to the GUI in Exercise 11 5---541
Exercise 11 10 Extend the GUI in Exercise 6 17 with a fancy list---------------------------------550
Exercise 11 11 Remove canvas items--566
Exercise 11 12 Introduce properties in class Parameters---580
Exercise 11 13 Convert command file into Python objects--------------------------------------600
Exercise 12 1 Allow multiple values of parameters in input files----------------------------------617
Exercise 12 2 Turn mathematical formulas into Fortran functions--------------------------------628
Exercise 12 3 Move a wave source during simulation--675
Exercise 12 4 Include damping in a 1D wave simulator---------------------------------------675

Exercise 12 5 Add a NumPy database to a PDE simulator---675
Exercise 12 6 Use iterators in finite difference schemes---675
Exercise 12 7 Set vectorized boundary conditions in 3D grids-----------------------------------675
Exercise B 1 Make a Python module of simviz1 py---734
Exercise B 2 Pack modules and packages using Distutils---735
Exercise B 3 Distribute mixed-language code using Distutils------------------------------------735
Exercise B 4 Use tools to document the script in Exercise 3 14----------------------------------735
Exercise B 5 Make a regression test for a trivial script--735
Exercise B 6 Repeat Exercise B 5 using the test script tools--------------------------------------735
Exercise B 7 Make a regression test for a script with I/O--735
Exercise B 8 Make a regression test for the script in Exercise 3 14-------------------------------736
Exercise B 9 Approximate floats in Exercise B 5---736
Exercise B 10 Make tests for grid iterators---736
Exercise B 11 Make a tar/zip archive of files associated with a script----------------------------736
Exercise B 12 Semi-automatic evaluation of a student project-------------------------------------737

***Python_Testing_Beginners_Guide_2010_255p

Preface 1
Chapter 1: Testing for Fun and Profit ---7
 How can testing help? --8
 Types of testing ---9
   Unit testing ---9
   Integration testing ---9
   System testing --9
 You've got Python, right? ---10
 Summary --10
Chapter 2: Doctest: The Easiest Testing Tool --11
 Basic doctest ---11
 Time for action – creating and running your first doctest --12
   The syntax of doctests ---13
 Time for action – writing a more complex test --14
   Expecting exceptions --15
 Time for action – expecting an exception --16
   Expecting blank lines in the output --17
   Using directives to control doctest ---17
     Ignoring part of the result ---17
 Time for action – using ellipsis in tests --17
     Ignoring whitespace ---18
 Time for action – normalizing whitespace ---19
     Skipping an example entirely ---19
 Time for action – skipping tests ---20
     Other doctest directives --21
   Execution scope ---21
 Embedding doctests in Python docstrings ---24
 Time for action – embedding a doctest in a docstring --24
   Doctest directives --25
   Execution scope ---26
 Putting it in practice: an AVL tree ---26
   English specification --27
   Node data ---28
   Constructor ---30

Recalculate height --30
Make deletable ---32
Rotation --33
Locating a node --34
Testing the rest of the specification ---34
Summary --35
Chapter 3: Unit Testing with Doctest --37
What is Unit testing and what it is not? ---37
Time for action – identifying units --38
Unit testing throughout the development process --40
Design phase --41
Time for action – unit testing during design --41
Development phase ---44
Time for action – unit testing during development ---44
Feedback phase --47
Time for action – unit testing during feedback ---47
Back to the development phase --51
Time for action – unit testing during development... again ---51
Maintenance phase --53
Time for action – unit testing during maintenance ---53
Reuse phase --55
Time for action – unit testing during reuse ---55
Summary --59
Chapter 4: Breaking Tight Coupling by using Mock Objects ---------------------------------------61
Installing Python Mocker --61
Time for action – installing Python Mocker ---62
The idea of a mock object ---62
Python Mocker --63
Time for action – exploring the basics of Mocker ---63
Mocking functions --67
Mocking containers --68
Parameter matching --69
ANY --69
ARGS ---70
KWARGS --70
IS --71
IN --71
CONTAINS --72
MATCH --72
Mocking complex expressions --73
Returning iterators --73
Raising exceptions --74
Calling functions via a mock --74
Specifying that an expectation should occur multiple times ---------------------------------------75
Replacing library objects with mocks ---77
Mocking self --80
Time for action – passing a mock object as self --80
Summary --82
Chapter 5: When Doctest isn't Enough: Unittest to the Rescue --------------------------------------83
Basic unittest ---83
Time for action – testing PID with unittest ---84
Assertions ---89

assertTrue --89
assertFalse ---90
assertEqual ---90
assertNotEqual ---90
assertAlmostEqual ---90
assertNotAlmostEqual --92
assertRaises ---92
fail --93
Test fixtures --94
Time for action – testing database-backed units ---95
Integrating with Python Mocker ---100
Summary ---100
Chapter 6: Running Your Tests: Follow Your Nose --101
What is Nose? ---101
Installing Nose --102
Organizing tests ---103
Time for action – organizing tests from previous chapters --104
Finding doctests ---108
Customizing Nose's search ---109
Nose and doctest --110
Time for action – creating a fixture for a doctest --111
Nose and unittest ---112
Time for action – creating a module fixture ---113
Time for action – creating a package fixture --114
Nose's own testing framework --116
Time for action – using Nose-specific tests ---116
Summary ---118
Chapter 7: Developing a Test-Driven Project ---119
Writing the specification --119
Time for action – what are you going to do? ---125
Writing initial unit tests --125
Time for action – nailing down the specification with unit tests ----------------------------------139
Coding planner.data ---139
Using the tests to get the code right ---143
Fixing the code --143
Time for action – writing and debugging code ---146
Writing persistence tests ---147
Writing persistence code --148
Finishing up ---151
Summary ---153
Chapter 8: Testing Web Application Frontends using Twill --155
Installing Twill --155
Exploring the Twill language --156
Time for action – browsing the web with Twill ---156
Time for action – Twill scripting --159
Twill commands --160
help ---160
setglobal ---160
setlocal --161
add_auth ---161
add_extra_header ---161
clear_extra_headers ---162

show_extra_headers ---162
agent --162
back ---162
clear_cookies ---162
code ---162
config ---163
debug --163
echo ---163
exit ---163
extend_with ---164
find ---164
notfind ---164
follow --164
formaction ---164
formclear ---165
formfile ---165
formvalue --165
getinput ---165
getpassword --165
go ---166
info --166
save_cookies --166
load_cookies --166
show_cookies --166
redirect_error ---166
redirect_output ---166
reset_error --166
reset_output ---167
reload --167
reset_browser --167
run ---167
runfile ---167
save_html --167
show --167
showforms ---167
showhistory ---168
showlinks --168
sleep --168
submit --168
tidy_ok ---168
title ---168
url --168
Calling Twill scripts from tests --169
Time for action – running Twill script files ---169
Time for action – running Twill script strings ---170
A nifty trick --171
Integrating Twill operations into unittest tests --172
Time for action – using Twill's browser object ---172
Browser methods --173
get_code --174
get_html --174
get_title ---174

get_url --174
find_link --174
follow_link ---175
get_all_forms ---175
get_form --175
get_form_field ---175
clicked --176
submit --176
Summary --176
Chapter 9: Integration Testing and System Testing ---177
Integration tests and system tests --177
Time for action – figuring out the order of integration --178
Automation with doctest, unittest, and Nose ---180
Time for action – writing integration tests for the time planner ----------------------------181
Summary --202
Chapter 10: Other Testing Tools and Techniques --203
Code coverage --203
coverage.py --204
Time for action – using coverage.py --205
Version control hooks --207
Bazaar --208
Time for action – installing Nose as a Bazaar post-commit hook ------------------------------208
Mercurial ---210
Time for action – installing Nose as a Mercurial ---210
post-commit hook --210
Git ---211
Time for action – installing Nose as a Git post-commit hook ------------------------------------212
Darcs ---213
Time for action – installing Nose as a Darcs post-record hook ----------------------------------213
Subversion --215
Time for action – installing Nose as a Subversion ---216
post-commit hook --216
Automated continuous integration --219
Buildbot --219
Time for action – using Buildbot with Bazaar --219
Summary --223
Appendix: Answers to Pop Quizes ---225
Chapter 2 ---225
Pop quiz – doctest syntax --225
Chapter 3 ---225
Pop quiz – understanding units --225
Pop quiz – unit testing during design --226
Pop quiz – unit testing --226
Chapter 4 ---226
Pop quiz – Mocker usage --226
Chapter 5 ---227
Pop quiz – basic unittest knowledge --227
Pop quiz – text fixtures --227
Chapter 6 ---227
Pop quiz – testing with Nose ---227
Chapter 7 ---227
Pop quiz – test-driven development --227

Chapter 8 --228
Pop quiz – the Twill language --228
Pop quiz – browser methods ---228
Chapter 9 --228
Pop quiz – diagramming integration --228
Pop quiz – writing integration tests --229
Chapter 10 --229
Pop quiz – code coverage ---229
Pop quiz – version control hooks --229
Index ---231

***Python_Tutorial_Learning_To_Program_2006_342p

Learning to program---1
What do I need to be a programmer?--8
What is programming?---10
Getting Started---19
Simple sequences---23
Data---30
More Sequences---64
Looping the loop---71
A Little Bit of Style--80
Input---88
Conditionals--98
Functions and Modules--109
File Handling--126
Text Handling---143
Error Handling--154
Namespaces---164
Regular Expressions--171
Classes--181
Event Driven Programs---206
Introduction to GUI Programming--213
Recursion---233
Introduction to Functional Programming--238
A Case Study---250
Python in Practice--272
Working with Databases--275
Working with the Operating System--302
References--334

***python_tutorial_tutorialspoint_2014_431p

///

***Python_v1_3_13_S60_3rd_Edition_Docs_2006_86p

1 Introduction---1
1.1 Scope---1
1.2 Audience---1

1.3 New in Release 1.3.12--2
1.4 Naming Conventions--2

2 API Summary--3
2.1 Python Standard Library--3
2.2 Python for S60 Extensions--3
2.3 Third-Party Extensions--4

3 Selected Issues on Python Programming for S60--5
3.1 Concurrency Aspects--5
3.2 Current S60 Python Script Execution Environment--5
3.3 Standard I/O Streams--6
3.4 Usage of Unicode--6
3.5 Date and Time--6
3.6 Sharing Native Resources between Threads--6
3.7 Scalable User Interface--7
3.8 Error Handling--7
3.9 Limitations and Areas of Development--7

4 Operating System Services and Information--9
4.1 e32 — A Symbian OS related services package---9
4.2 sysinfo — Access to system information---11

5 User Interface and Graphics--13
5.1 appuifw — Interface to the S60 GUI framework---13
5.2 graphics — A graphics related services package--24
5.3 camera — Interface for taking photographs---31
5.4 keycapture — Interface for global capturing of key events.-----------------------------------32
5.5 topwindow — Interface for creating windows that are shown on top of
other applications.--33

6 Audio and Communication Services--37
6.1 audio — An audio related services package--37
6.2 telephone — Telephone services--38
6.3 messaging — A messaging services package--39
6.4 inbox — Interface to device inbox--39
6.5 location — GSM location information--40

7 Data Management--41
7.1 contacts — A contacts related services package--41
7.2 calendar — Access to calendar related services--44
7.3 calendar for EKA2 — Access to calendar related services-------------------------------50
7.4 e32db — Interface to the Symbian native DB---55
7.5 e32dbm — DBM implemented using the Symbian native DBMS---------------------------57

8 Standard Library Support and Extensions---61
8.1 Support for Python Standard Library--61
8.2 Extensions to Standard Library Modules---62

9 Extending and Embedding---65
9.1 Python/C API Extensions--65
9.2 Extending Python for S60--66

10 Terms and Abbreviations--69
A Reporting Bugs--73
Module Index--75
Index---77

***Rapid_GUI_Programming_with_Python_and_Qt_The_Definitive_Guide_to_PyQt_Programming_2008_643p

Foreword--xiii
Introduction---1

Part I: Python Programming

Chapter 1. Data Types and Data Structures--9
Executing Python Code--10
Variables and Objects---12
Numbers and Strings--15
Integers and Long Integers---16
Floats and Decimals---17
Bytestrings, Unicode Strings, and QStrings--20
Collections--29
Tuples---29
Lists---31
Dictionaries---35
Sets--37
Built-in Functions---37
Summary---41
Exercises---42

Chapter 2. Control Structures---45
Conditional Branching--46
Looping---49
List Comprehensions and Generators--53
Functions--55
Generator Functions---58
Using Keyword Arguments--59
Lambda Functions---61
Dynamic Function Creation---62
Partial Function Application--63
Exception Handling--66
Summary--72
Exercises--72

Chapter 3. Classes and Modules---75
Creating Instances---77
Methods and Special Methods---79
Static Data, and Static Methods and Decorators---85
Example: The Length Class---86
Collection Classes---92
Example: The OrderedDict Class--92

Inheritance and Polymorphism--99
Modules and Multifile Applications---104
Using the doctest Module--105
Summary---107
Exercises---108

Part II: Basic GUI Programming

Chapter 4. Introduction to GUI Programming--111
A Pop-Up Alert in 25 Lines---112
An Expression Evaluator in 30 Lines--116
A Currency Converter in 70 Lines---121
Signals and Slots---127
Summary---136
Exercise--137

Chapter 5. Dialogs---139
Dumb Dialogs--141
Standard Dialogs---147
Modal OK/Cancel-Style Dialogs--148
Smart Dialogs--154
Modeless Apply/Close-Style Dialogs--155
Modeless “Live” Dialogs---159
Summary---162
Exercise--163

Chapter 6. Main Windows---165
Creating a Main Window---166
Actions and Key Sequences--171
Resource Files---172
Creating and Using Actions---174
Restoring and Saving the Main Window’s State---181
Handling User Actions---190
Handling File Actions---191
Handling Edit Actions---197
Handling Help Actions--200
Summary--201
Exercise---202

Chapter 7. Using Qt Designer---205
Designing User Interfaces---208
Implementing Dialogs--216
Testing Dialogs--221
Summary---223
Exercise--224

Chapter 8. Data Handling and Custom File Formats---227
Main Window Responsibilities---229
Data Container Responsibilities--235
Saving and Loading Binary Files---240
Writing and Reading Using QDataStream--240
Writing and Reading Using the pickle Module--246

Saving and Loading Text Files--249
Writing and Reading Using QTextStream---250
Writing and Reading Using the codecs Module--255
Saving and Loading XML Files--256
Writing XML--256
Reading and Parsing XML with PyQt’s DOM Classes---------------------------------------259
Reading and Parsing XML with PyQt’s SAX Classes---262
Summary---265
Exercise---266

Part III: Intermediate GUI Programming

Chapter 9. Layouts and Multiple Documents---269
Layout Policies--270
Tab Widgets and Stacked Widgets--272
Extension Dialogs--276
Splitters--280
Single Document Interface (SDI)---283
Multiple Document Interface (MDI)---290
Summary---300
Exercise--301

Chapter 10. Events, the Clipboard, and Drag and Drop---303
The Event-Handling Mechanism--303
Reimplementing Event Handlers--305
Using the Clipboard---310
Drag and Drop--312
Handling Custom Data--313
Summary---317
Exercise---318

Chapter 11. Custom Widgets---321
Using Widget Style Sheets--322
Creating Composite Widgets---325
Subclassing Built-in Widgets---326
Subclassing QWidget--328
Example: A Fraction Slider---331
Example: A Flow-Mixing Widget---339
Summary---345
Exercise---346

Chapter 12. Item-Based Graphics--349
Custom and Interactive Graphics Items--351
Animation and Complex Shapes---368
Summary---378
Exercise---379

Chapter 13. Rich Text and Printing---381
Rich Text Editing--382
Using QSyntaxHighlighter---382
A Rich Text Line Edit--389
Printing Documents---398

Printing Images--400
Printing Documents Using HTML and QTextDocument--------------------------------------401
Printing Documents Using QTextCursor and QTextDocument-------------------------------403
Printing Documents Using QPainter---407
Summary---411
Exercise--412

Chapter 14. Model/View Programming--413
Using the Convenience Item Widgets---415
Creating Custom Models--423
Implementing the View Logic--424
Implementing the Custom Model--427
Creating Custom Delegates---436
Summary---442
Exercise--443

Chapter 15. Databases---445
Connecting to the Database--446
Executing SQL Queries--446
Using Database Form Views--451
Using Database Table Views--457
Summary---470
Exercise--471

Part IV: Advanced GUI Programming

Chapter 16. Advanced Model/View Programming---475
Custom Views---476
Generic Delegates--483
Representing Tabular Data in Trees---492
Summary---505
Exercise--505

Chapter 17. Online Help and Internationalization--509
Online Help--510
Internationalization---512
Summary---519
Exercise--520

Chapter 18. Networking---521
Creating a TCP Client---523
Creating a TCP Server--529
Summary---534
Exercise--534

Chapter 19. Multithreading---537
Creating a Threaded Server--539
Creating and Managing Secondary Threads---544
Implementing a Secondary Thread---552
Summary---557
Exercise--558
This Is Not Quite the End--559

Appendix A. Installing---561
Installing on Windows--561
Installing on Mac OS X---566
Installing on Linux and Unix---570
Appendix B. Selected PyQt Widgets--575
Appendix C. Selected PyQt Class Hierarchies---581
Index---585

***Snake_Wrangling_for_Kids_Learning_to_Program_with_Python_3_linux_v0_7_7_2007_160p

1 Not all snakes will squish you--1
1.1 A Few Words About Language---2
1.2 The Order of Non-venomousConstricting Serpentes.---------------------------------------3
1.3 Your first Python program--4
1.4 Your Second Python program.the same again?---5

2 8 multiplied by 3.57 equals.---9
2.1 Use of brackets and “Order of Operations”---11
2.2 There’s nothing so fickle as a variable--13
2.3 Using Variable---15
2.4 A Piece of String?---16
2.5 Tricks with Strings--18
2.6 Not quite a shopping list--19
2.7 Tuples and Lists---22
2.8 Things to try--23

3 Turtles, and other slow moving creatures---25
3.1 Things to try--31

4 How to ask a question--33
4.1 Do this. or ELSE!!!---34
4.2 Do this. or do this. or do this. or ELSE!!!---35
4.3 Combining conditions--36
4.4 Emptiness--36
4.5 What’s the difference.?--38

5 Again and again---41
5.1 When is a block not square?---44
5.2 While we’re talking about looping.---50
5.3 Things to try--52

6 Sort of like recycling.--53
6.1 Bits and Pieces---57
6.2 Modules--58
6.3 Things to try--61

7 A short chapter about Files---63

8 Turtles galore---65

8.1 Colouring in--69
8.2 Darkness--72
8.3 Filling things---72
8.4 Things to try--78

9 A bit graphic--81
9.1 Quick Draw---83
9.2 Simple Drawing---85
9.3 Drawing Boxes--87
9.4 Drawing Arcs--91
9.5 Drawing Ovals---92
9.6 Drawing Polygons--95
9.7 Drawing Images---96
9.8 Basic Animation---98
9.9 Reacting to events.--99

10 Where to go from here--103

A Python Keywords---105
B Built-in Functions---119
C A Few Python Modules--129
D Answers to “Things to try”--139

***Starting_Out_with_Python_2009_502p

Part I: Programming Fundamentals
Chapter 1 Introduction to Computers and Programming
1.1 Introduction
1.2 Hardware and Software
1.3 How Computers Store Data
1.4 How a Program Works
1.5 Using Python

Chapter 2 Input, Processing, and Output
2.1 Designing a Program
2.2 Input, Processing, and Output
2.3 Displaying Output with the print Statement
2.4 Comments
2.5 Variables
2.6 Reading Input from the Keyboard
2.7 Performing Calculations
2.8 More About Data Output

chapter 3 Simple Functions
3.1 Introduction to Functions
3.2 Defining and Calling a Function
3.3 Designing a Program to Use Functions
3.4 Local Variables
3.5 Passing Arguments to Functions
3.6 Global Variables and Global Constants

Chapter 4 Decision Structures and Boolean Logic

4.1 The if Statement
4.2 The if - else Statement
4.3 Comparing Strings
4.4 Nested Decision Structures and the if-elif-else Statement
4.5 Logical Operators
4.6 Boolean Variables

Chapter 5 Repetition Structures
5.1 Introduction to Repetition Structures
5.2 The while Loop: a Condition-Controlled Loop
5.3 The for Loop: a Count-Controlled Loop
5.4 Calculating a Running Total
5.5 Sentinels
5.6 Input Validation Loops
5.7 Nested Loops

Chapter 6 Value-Returning Functions and Modules
6.1 Introduction to Value-Returning Functions:
Generating Random Numbers
Writing Your Own Functions
The math Module
Storing Functions in Modules

Part II: Using Objects to Perform Tasks
Chapter 7 Files and Exceptions
Introduction to File Input and Output
Using Loops to Process Files
Processing Records
Exceptions

Chapter 8 Working with Sequences: Strings and Lists
8.1 Sequences
8.2 Working with Strings
8.3 Lists

Part Ill: Object-Oriented Programming
Chapter 9 Classes and Object-Oriented Programming
9.1 Procedural and Object-Oriented Programming
9.2 Classes
9.3 Working with Instances
9.4 Techniques for Designing Classes

Chapter 10 Inheritance
10.1 Introduction to Inheritance
10.2 Polymorphism

Part IV: Advanced Topics
Chapter 11 Recursion
11.1 Introduction to Recursion
11.2 Problem Solving with Recursion
11.3 Examples of Recursive Algorithms

Chapter 12 GUI Programming
Graphical User Interfaces
Using the Tkinter Module
Display Text with Label Widgets
Organizing Widgets with Frames
Button Widgets and Info Dialog Boxes
Getting Input with the Entry Widget
Using Labels as Output Fields
Radio Buttons and Check Buttons

Appendix A Installing Python
Appendix B introduction to IDLE
Appendix C The ASCII Character Set
Index
Student CD The following appendix is on the accompanying Student CD.
Appendix D Answers to Checkpoints

***The_Definitive_Guide_to_Jython_Python_for_the_Java_Platform_2010_545p

Contents at a Glance--iii
Contents---v
Foreword---xix
About the Authors---xx
About the Technical Reviewers---xxii
Acknowledgments---xxiii
Introduction--xxvi
Part I: Jython Basics: Learning the Language--1
Chapter 1: Language and Syntax---3
Chapter 2: Data Types and Referencing---25
Chapter 3: Operators, Expressions, and Program Flow---59
Chapter 4: Defining Functions and Using Built-ins--81
Chapter 5: Input and Output---105
Chapter 6: Object-Oriented Jython--113
Chapter 7: Exception Handling and Debugging---133
Chapter 8: Modules and Packages for Code Reuse--151
Part II: Using the Language--163
Chapter 9: Scripting With Jython---165
Chapter 10: Jython and Java Integration---175
Chapter 11: Using Jython in an IDE--197
Chapter 12: Databases and Jython: Object Relational Mapping and Using JDBC---------------231
Part III: Developing Applications with Jython---263
Chapter 13: Simple Web Applications--265
Chapter 14: Web Applications With Django--281
Chapter 15: Introduction to Pylons---327
Chapter 16: GUI Applications--347
Chapter 17: Deployment Targets--359
Part IV: Strategy and Technique---377
Chapter 18: Testing and Continuous Integration---379
Chapter 19: Concurrency---413
Appendix A: Using Other Tools with Jython---437

Appendix B: Jython Cookbook--445
Appendix C: Built-in Functions--463
Index---485

***The_Quick_Python_Book_Second_Edition_2010_362p

PART 1 STARTING OUT---1
1 About Python--3
2 Getting started---10
3 The Quick Python overview--18

PART 2 THE ESSENTIALS--33
4 The absolute basics--35
5 Lists, tuples, and sets---45
6 Strings--63
7 Dictionaries--81
8 Control flow---90
9 Functions---103
10 Modules and scoping rules---115
11 Python programs---12
12 Using the filesystem--147
13 Reading and writing files--159
14 Exceptions--172
15 Classes and object-oriented programming--186
16 Graphical user interfaces--209

PART 3 ADVANCED LANGUAGE FEATURES--223
17 Regular expressions---225
18 Packages---234
19 Data types as objects---242
20 Advanced object-oriented features---247

PART 4 WHERE CAN YOU GO FROM HERE?--263
21 Testing your code made easy(-er)--265
22 Moving from Python 2 to Python 3--274
23 Using Python libraries--282
24 Network, web, and database programming--290

***Thinking_in_Python_Bruce_Eckel_Rev_0_1_2_2001_177p

Preface--5
Introduction---7
The Y2K syndrome---8
Context and composition---9
A quick course in Python for programmers---11
Python overview--11
Built-in containers---12
Functions---13
Strings--14
Classes--16
The pattern concept---21

What is a pattern?--21
Pattern taxonomy--23
Design Structures--24
Design principles---25
The Singleton---27
Classifying patterns--32
The development challenge---33
Exercises---34
2: Unit Testing---34
Write tests first--36
Simple Python testing--37
A very simple framework---38
Writing tests--39
White-box & black-box tests---42
Running tests---44
Automatically executing tests--47
Exercises---47
3: Building application frameworks---47
Template method--48
Exercises---49
4:Fronting for an implementation---49
Proxy--50
State--52
StateMachine--54
Table-Driven State Machine--61
The State class--63
Conditions for transition--63
Transition actions--64
The table--64
The basic machine---65
Simple vending machine--65
Testing the machine---70
Tools--70
Exercises---70
X: Decorators: dynamic type selection--72
Basic decorator structure---73
A coffee example--73
Class for each combination---73
The decorator approach---76
Compromise---79
Other considerations--82
Exercises---82
Y: Iterators: decoupling algorithms from containers---83
Type-safe iterators--84
5: Factories: encapsulating object creation--85
Simple Factory method---86
Polymorphic factories---88
Abstract factories---90
Exercises---93
6: Function objects--94
Command: choosing the operation at run-time--94
Strategy: choosing the algorithm at run-time---96

Chain of responsibility--97
Exercises--101
7: Changing the interface--101
Adapter--101
Façade---103
Exercises--104
8: Table-driven code: configuration flexibility---105
Table-driven code using anonymous inner classes--105
10: Callbacks---107
Observer---107
Observing flowers---109
A visual example of observers---116
Exercises--122
11: Multiple dispatching--122
Visitor, a type of multiple dispatching---127
Exercises--128
12: Pattern refactoring--130
Simulating the trash recycler---130
Improving the design---134
“Make more objects”---135
A pattern for prototyping creation--137
Trash subclasses--142
Parsing Trash from an external file---143
Recycling with prototyping--146
Abstracting usage--147
Multiple dispatching--151
Implementing the double dispatch--152
The Visitor pattern--158
A Reflective Decorator--161
More coupling?---166
RTTI considered harmful?--166
Summary---169
Exercises---171
13: Projects--171
Rats & Mazes---171
Other maze resources--176
XML Decorator---176

***Thinking_in_Tkinter_2005_53p

///

***think_python_2012_240p

1 The way of the program---1
2 Variables, expressions and statements---11
3 Functions---19
4 Case study: interface design---31
5 Conditionals and recursion--41
6 Fruitful functions---51
7 Iteration--63

8 Strings--71
9 Case study: word play---81
10 Lists---87
11 Dictionaries--101
12 Tuples---113
13 Case study: data structure selection--123
14 Files--133
15 Classes and objects---143
16 Classes and functions--151
17 Classes and methods--157
18 Inheritance---167
19 Case study: Tkinter---179
A Debugging---193
B Analysis of Algorithms---201
C Lumpy--211

***Tkinter_8_4_reference_a_GUI_for_Python_2010_118p

1. What is Tkinter?---3
2. A minimal application---3
3. Definitions---4
4. Layout management---5
4.1. Thegrid() method---5
4.2. Other grid management methods--6
4.3. Configuring column and row sizes---7
4.4. Making the root window resizeable--8
5. Standard attributes--8
5.1. Dimensions--9
5.2. The coordinate system--9
5.3. Colors---9
5.4. Type fonts---10
5.5. Anchors--11
5.6. Relief styles--12
5.7. Bitmaps--12
5.8. Cursors---12
5.9. Images--14
5.10. Geometry strings---14
5.11. Window names---15
5.12. Cap and join styles--15
5.13. Dash patterns---16
5.14. Matching stipple patterns---16
6. The Button widget---17
7. The Canvas widget---19
7.1. Canvas coordinates---20
7.2. The Canvas display list--20
7.3. Canvas object IDs--21
7.5. Canvas tagOrId arguments---21
7.6. Methods on Canvas widgets---21
7.7. Canvas arc objects--26
7.8. Canvas bitmap objects--28
7.9. Canvas image objects---29
7.10. Canvas line objects--29

7.11. Canvas oval objects--31
7.12. Canvas polygon objects--32
7.13. Canvas rectangle objects--34
7.14. Canvas text objects--35
7.15. Canvas window objects---36
8. The Checkbutton widget---37
9. The Entry widget---40
9.1. Scrolling an Entry widget--43
10. The Frame widget---43
11. The Label widget---44
12. The LabelFrame widget--46
13. The Listbox widget---48
13.1. Scrolling a Listbox widget--52
14. The Menu widget--52
14.1. Menu item creation (coption) options--55
15. The Menubutton widget--56
16. The Message widget---58
17. The OptionMenu widget---59
18. The PanedWindow widget---60
18.1. PanedWindow child configuration options--63
19. The Radiobutton widget---63
20. The Scale widget---66
21. The Scrollbar widget--69
21.1. The Scrollbar command callback---72
21.2. Connecting a Scrollbar to another widget---73
22. The Spinbox widget--73
23. The Text widget---77
23.1. Text widget indices---80
23.2. Text widget marks--81
23.3. Text widget images---82
23.4. Text widget windows---82
23.5. Text widget tags---82
23.6. Setting tabs in a Text widget---82
23.7. The Text widget undo/redo stack--83
23.8. Methods on Text widgets--83
24. Toplevel: Top-level window methods---90
25. Universal widget methods--93
26. Standardizing appearance---100
26.1. How to name a widget class--101
26.2. How to name a widget instance--101
26.3. Resource specification lines--102
26.4. Rules for resource matching--103
27. Connecting your application logic to the widgets---103
28. Control variables: the values behind the widgets---104
29. Focus: routing keyboard input--106
30. Events---107
30.1. Levels of binding.--107
30.2. Event sequences---108
30.3. Event types.---109
30.4. Event modifiers.--110
30.5. Key names--110
30.6. Writing your handler: The Event class.--113

30.7. The extra arguments trick.--114
30.8. Virtual events---115
31. Pop-up dialogs.---116
31.1. The tkMessageBox dialogs module.--116
31.2. The tkFileDialog module--117
31.3. The tkColorChooser module---118

***wxPython_in_action_2006_583p

PART 1 INTRODUCTION TO WXPYTHON--1
1 Welcome to wxPython---3
2 Giving your wxPython program a solid foundation----------------------------------29
3 Working in an event-driven environment---56
4 Making wxPython easier to handle with PyCrust-------------------------------------83
5 Creating your blueprint--116
6 Working with the basic building blocks---146

PART 2 ESSENTIAL WXPYTHON--183
7 Working with the basic controls---185
8 Putting widgets in frames--224
9 Giving users choices with dialogs---258
10 Creating and using wxPython menus--293
11 Placing widgets with sizers--323
12 Manipulating basic graphical images--356

PART 3 ADVANCED WXPYTHON--391
13 Building list controls and managing items--393
14 Coordinating the grid control--425
15 Climbing the tree control--460
16 Incorporating HTML into your application--485
17 The wxPython printing framework---504
18 Using other wxPython functionality---521

***XML_Processing_with_Perl_Python_and_PHP_2002_447p

Introduction---xxi
Part I Applying XML--1
Chapter 1: Introduction to XML--3
Chapter 2 Fundamentals of XML--11
Chapter 3 Data Type Definitions (DTDs)--23
Chapter 4 Applying XML with Scripting Languages---35
Chapter 5 Data Exchange and XML--47
Part II XML and Perl--61
Chapter 6 XML Solutions in Perl--63
Chapter 7 Perl and Unicode--87
Chapter 8 Generating and Parsing XML Documents with Perl-----------------------------------97
Chapter 9 Converting XML Documents Using Perl---119
Chapter 10 Applying SOAP/XML-RPC in Perl--147
Part III XML and Python---167
Chapter 11 XML Solutions in Python---169

Chapter 12 Python and Unicode--187
Chapter 13 Generating and Parsing XML Documents with Python------------------------------197
4021fm.qxd 11/2/01 4:27 PM Page--ix
Chapter 14 Converting XML Documents Using Python---209
Chapter 15 Applying SOAP/XML-RPC in Python--225
Chapter 16 Zope and XML Documents--243
Part IV XML and PHP--263
Chapter 17 XML and PHP---265
Chapter 18 Developing XML Applications with PHP---287
Chapter 19 PHP and XML-RPC--305
Part V XML and Other Languages---319
Chapter 20 XML and REBOL---321
Chapter 21 XML and Ruby---329
Chapter 22 XML and Tcl--341
Chapter 23 AppleScript and XML--359
Appendices
Appendix A Unicode Quick Reference--377
Appendix B Resource Guide--393
Index---403

***XML_Processing_with_Python_2002_447p

Preface
Audience
Organization
Conventions Used in This Book
How to Contact Us
Acknowledgments

1. Python and XML
1.1 Key Advantages of XML
1.2 The XML Specifications
1.3 The Power of Python and XML
1.4 What Can We Do with It?

2. XML Fundamentals
2.1 XML Structure in a Nutshell
2.2 Document Types and Schemas
2.3 Types of Conformance
2.4 Physical Structures
2.5 Constructing XML Documents
2.6 Document Type Definitions
2.7 Canonical XML
2.8 Going Beyond the XML Specification

3. The Simple API for XML
3.1 The Birth of SAX
3.2 Understanding SAX
3.3 Reading an Article
3.4 Searching File Information
3.5 Building an Image Index
3.6 Converting XML to HTML

3.7 Advanced Parser Factory Usage
3.8 Native Parser Interfaces

4. The Document Object Model
4.1 The DOM Specifications
4.2 Understanding the DOM
4.3 Python DOM Offerings
4.4 Retrieving Information
4.5 Changing Documents
4.6 Building a Web Application
4.7 Going Beyond SAX and DOM

5. Querying XML with XPath
5.1 XPath at a Glance
5.2 Where Is XPath Used?
5.3 Location Paths
5.4 XPath Arithmetic Operators
5.5 XPath Functions
5.6 Compiling XPath Expressions

6. Transforming XML with XSLT
6.1 The XSLT Specification
6.2 XSLT Processors
6.3 Defining Stylesheets
6.4 Using XSLT from the Command Line
6.5 XSLT Elements
6.6 A More Complex Example
6.7 Embedding XSLT Transformations in Python
6.8 Choosing a Technique

7. XML Validation and Dialects
7.1 Working with DTDs
7.2 Validation at Runtime
7.3 The BillSummary Example
7.4 Dialects, Frameworks, and Workflow
7.5 What Does ebXML Offer?

8. Python Internet APIs
8.1 Connecting Web Sites
8.2 Working with URLs
8.3 Opening URLs
8.4 Connecting with HTTP
8.5 Using the Server Classes

9. Python, Web Services, and SOAP
9.1 Python Web Services Support
9.2 The Emerging SOAP Standard
9.3 Python SOAP Options
9.4 Example SOAP Server and Client
9.5 What About XML-RPC?

10. Python and Distributed Systems Design

10.1 Sample Application and Flow Analysis
10.2 Understanding the Scope
10.3 Building the Database
10.4 Building the Profiles Access Class
10.5 Creating an XML Data Store
10.6 The XML Switch
10.7 Running the XML Switch
10.8 A Web Application

A. Installing Python and XML Tools
A.1 Installing Python
A.2 Installing PyXML
A.3 Installing 4Suite

B. XML Definitions
B.1 XML Definitions

C. Python SAX API

D. Python DOM API
D.1 4DOM Extensions

E. Working with MSXML3.0
E.1 Setting Up MSXML3.0
E.2 Basic DOM Operations
E.3 MSXML3.0 Support for XSLT
E.4 Handling Parsing Errors
E.5 MSXML3.0 Reference

F. Additional Python XML Tools
F.1 Pyxie
F.2 Python XML Tools
F.3 XML Schema Validator
F.4 Sab-pyth
F.5 Redfoot
F.6 XML Components for Zope
F.7 Online Resources

***Beginning_Django_E-Commerce_2009_398p

About the Author--xi
About the Technical Reviewer--xii
Acknowledgments---xiii
Introduction--xiv
Chapter 1: Best Laid Plans---1
Chapter 2: Creating a Django Site---17
Chapter 3: Models for Sale---39
Chapter 4: The Shopping Cart---79
Chapter 5: Site Checkout & Orders---109
Chapter 6: Creating User Accounts---153
Chapter 7: Product Images--173
Chapter 8: Implementing Product Search---179
Chapter 9: Intelligent Cross-Selling---193
Chapter 10: Adding in Ajax--205
Chapter 11: Search Engine Optimization---231
Chapter 12: Web Security Overview---255
Chapter 13: Improving Performance---279
Chapter 14: Django Testing--299
Chapter 15: Deployment---323
Chapter 16: Django on Google App Engine--341
Index---365
About the Author--xvi
About the Technical Reviewer---xvii
Acknowledgments---xviii
Introduction---xix
Chapter 1: Best Laid Plans---1
Selling Stuff Online--2
Why Django?--4
Straying From the Django Philosophy--5
A Quick Word on the Software---5
A Note on Version Control---6
Firefox---6
Installing the Software--7
Installing Python--7
Installing Django--8
Installing MySQL--9
Installing Eclipse and the PyDev Extensions--9
Things to Consider Before You Start---9
Security--10
Accessibility---10
PCI Compliance--11
Search Engine Optimization--11
Deployment---12
Business Requirements--12
Accounting & Auditing--12
Supply Chain Management--13
Marketing Decisions--14
Summary--16
Chapter 2: Creating a Django Site--17
A Django-istic Welcome---17

Creating the Project--17
What Django Creates--18
Creating the MySQL Database---19
Dealing with Django Exceptions--22
Template & View Basics--23
Advanced Templates with Inheritance---25
Greater Ease with render_to_response()--28
Adding in the CSS---29
Location, Location, Location---34
A Site Navigation Include--35
A Word (or Two) About URLs---37
Summary--38
Chapter 3: Models for Sale---39
Databases 101--40
An Introduction To SQL--40
What Makes a Relational Database--42
What Django Gives You – The ORM---44
Creating the Catalog App---45
Creating the Django Models---46
Model Field Data Types--48
Creating the Category Model--50
Creating the Product Model--54
The Django Admin Interface---56
Product and Category Admins---57
A Note on Model Validation---59
Syncing Up the Models---60
Playing with Model Structure and Data--64
Templates, URLs, and Views--65
Configuring Page Titles and Meta Tags--66
Coding Up the Catalog Views---68
Creating the Template Files--69
A Category Link List--73
Our Code in Review---73
So How Does It All Work?---75
File Not Found and Custom 404s---77
Summary--78
Chapter 4: The Shopping Cart---79
Shopping Cart Requirements--79
An Introduction to Sessions--80
The Shopping Cart Model--82
Django Sessions Overview---84
Enabling and Using Sessions--85
Using Django Forms--86
The Add To Cart Form--86
Processing the Form--88
Putting It All Together--89
Cart Form Code in Review---94
Creating the Shopping Cart Page--95
Django Template ‘if’ and ‘for’ Tags---96
Custom Template Filters---96
Creating the Cart Page--97
Adding Custom Template Tags--101

Re-creating the Category List Tag--103
Static Content with Flatpages--105
Navigation Tags--107
Summary--108
Chapter 5: Site Checkout & Orders---109
Google Checkout API---109
Signing up with Google Checkout---110
Submitting Orders to Google---111
Building XML Documents in Python--112
Making HTTP Requests in Python---114
Your Google Merchant ID and Key---115
The Python Property Decorator--115
Creating the Checkout App--116
Order Checkout Requirements--124
SSL Middleware---124
DRY Models and Forms---126
Secure HTTP Requests--128
Credit Card Transactions--129
Order Checkout by Django--130
Signing up for an Authorize.Net Test Account--130
Order Information Models---131
The Checkout Form--133
Authorization and Capture--137
Order Processing---139
Checkout Views and URLs--141
Checkout Template and Order Form---143
Order Administration--149
Summary--151
Chapter 6: Creating User Accounts--153
Making Friends with the Source--153
Hooking Into Django’s Authentication--154
Creating the Login & Registration Pages---156
The My Account Page--160
The Change Password Page--162
The Order Details Page---163
Django User Profiles---165
Abstract Base Classes--165
The Order Info Page--167
Updating the Checkout Page---169
Summary--171
Chapter 7: Product Images--173
Dealing with Images---173
Django Image Fields---174
Installing the Python Imaging Library---174
Database Changes---174
Editing the Model---175
Adding a New Image--176
Image Template Changes--177
Summary--178
Chapter 8: Implementing Product Search--179
Instant Search---179
Search Requirements--180

Model Managers---180
Complex Lookups with Q---182
Search Results Pagination--183
Implementing Search--184
The Search Module--185
Search Template Tags--187
Search View and Template--189
Third-Party Search Solutions---192
Summary---192
Chapter 9: Intelligent Cross-Selling---193
Product Page Recommendations--193
Order-Based Filtering--194
Customer-Based Order Filtering---195
A Hybrid Approach--195
Home Page Recommendations---196
Tracking Each User--197
Dealing with Searches---198
View-Based Recommendations---200
Building the Homepage--202
Summary---204
Chapter 10: Adding in Ajax---205
The Ajax Pros and Cons--205
How Ajax Works--207
jQuery for Ajax--208
Getting jQuery--208
jQuery Basics--209
JavaScript Object Notation--211
Making Ajax Requests--212
Product Reviews---213
Review Model and Form---213
Template and View Changes---214
The Ajax Part of this Equation---216
Adding a Product Review--219
Product Catalog Tagging---220
Getting Django-Tagging---220
Django Content Types---221
Enabling Product Tagging--222
Creating the Tag Cloud---224
JavaScript Finishing Touches---226
Summary---229
Chapter 11: Search Engine Optimization---231
The Importance of Inbound Links---232
Content is King--233
Title and Meta Tags---233
Keywords in URLs---234
Generating a Keyword List---235
The Duplicate Content Problem--236
Semantic Web - Microformats & RDFa---238
Launching the Site---240
Submit Your URL--240
robots.txt File--241
Sitemaps for Search Engines---242

Content Relocation--244
Google Webmasters---245
Google Analytics---246
The Data Warehouse Principle---247
Signing Up for Google Analytics--248
E-Commerce and Search Tracking--248
Google Base Product Feed--251
500 Server Errors--253
Summary--253
Chapter 12: Web Security Overview---255
Securing the Site From Within--255
Django Permissions--256
Applying Permissions to Users--257
Applying Permissions to Groups--257
Protecting Against External Attacks---258
The Evils of Debug Mode---258
Configuring Local Settings--259
Customer Registration Revisited---259
Cross-Site Scripting Attacks---262
What’s in a QueryString?---263
Cross-Site Request Forgery---263
SQL Injection--265
Moving the Admin Interface---266
Storing Secrets---266
Storing Customer Passwords---267
Storing Credit Card Data--269
Symmetric Cryptography--270
Google Keyczar---271
A Credit Card Model and Form--273
Summary--278
Chapter 13: Improving Performance--279
The Database--280
Searching your Models---280
Avoiding Expensive Joins--281
Creating Database Indexes--283
Deleting Old Data--284
Caching with Memcached---287
The Virtue of Stale Data---287
Template Caching---288
The Low-Level Cache API--289
Django Signals for Cache Invalidation--291
A Quick Word about Django Signals--293
Front-End Engineering---293
Move CSS and JavaScript Into Separate Files--295
Reduce the Number of External Components--295
Optimize External Components--296
Summary--297
Chapter 14: Django Testing--299
Why We Test--299
How to Test Code--300
Creation of the Test Database--301
Python & Django Test Methods--301

Anatomy of a Test Class--303
Testing the Product Catalog---304
Writing Functional Tests---304
Managing Test State with Fixtures---307
Category Testing---309
Testing the ActiveProductManager--312
Product Catalog Model Tests--313
Testing Forms & Shopping Cart---315
Testing the Checkout Form---318
Security Testing--319
Summary--321
Chapter 15: Deployment---323
The Django Philosophy--324
Finding a Hosting Plan--325
Phase One: Apache and mod_wsgi---326
Installing the Apache Web Server--326
Creating the mod_wsgi File and Apache Virtual Host---328
Phase Two: Nginx for Static Media--331
Installing and Configuring NginX---331
Updating the Apache Virtual Host--333
Phase Three: Configuring SSL--334
Transferring Data with Django--338
The Admin Interface Styles--338
Summary--339
Chapter 16: Django on Google App Engine---341
The Inconvenient Truth--342
Signing Up For An Account---342
The Django App Engine Patch--343
Getting the Test Page Running--346
Our Google App Engine Store--346
The Brubeck Shopping Cart App---353
Views and Templates---356
Managing Database Indexes---361
Error Logs, Remote Data API, and Network Programming---------------------------------------362
Summary--364
Index---365

***Django_1.0_Template_Development_2008_272p

Preface--1
Chapter 1: An Introduction to the Django Template System---7
What are templates?---7
Understanding the need for templates--8
Overview of the Django template system---8
Separating code from presentation--8
Helping designers and developers collaborate--9
Increasing maintainability--9
Template syntax---10
Modularity and reusability---10
Flexibility--10
Limitations--11

Critics of the system--11
Exploring how Django handles requests---12
Understanding the template system syntax---15
Context variable--15
Variables---16
Filters--16
Tags--16
Comments---17
Code note: Python dictionaries---18
How invalid variables are handled--19
Creating our demo application--19
Starting our application--21
Adding templates to our application---22
Adding variables to the view--23
Moving the logic into a separate template file--24
Using template filters---25
Using template tags to perform logical tests--26
Adding comments---27
Summary---28
Chapter 2: Views, URLs, and Generic Views---29
An overview---30
Creating the application--30
Create the data model---30
Create the admin file--31
Configure the URLs---31
Add data in the admin application--32
Mapping URLs to views--32
Handling unmatched URL patterns---34
Splitting up the URL configurations--35
Creating views---36
Accepting the request object---36
Responding with an HTTP response--37
Responding with an exception---37
Putting the views together--38
Building the basic view---38
Cleaning up the error handling--39
Adding the template files---39
Adding the template to the view--41
Creating the list view and template---42
Using generic views to shorten development time--44
Simple generic views--44
Loading a template directly--44
Redirecting URLs--45
List/detail generic views--46
Replacing the list view--46
Replacing the detail view---48
Using the other generic views--49
Comparing views and generic views--49
Summary---50
Chapter 3: Template Context---51
The context explained---51
Practicing working with the context--53

Using locals for prototyping--55
Adding, changing, and removing items in the context---56
Using the context values in your templates---57
Preventing method execution from templates--60
Handling invalid context variables---60
Cleaning up the view--61
Context rendering shortcuts--62
Using render_to_response()--62
Using render_to_string()--63
Context processors---63
Exploring the default context processors---63
Auth---63
Debug---64
Media--64
il8n--65
Configuring your project to use context processors---65
Configuring your views to use context processors---65
Using render_to_response with RequestContext--66
Using the context processors in our project---67
Writing your own context processor---68
Summary---70
Chapter 4: Using the Built-In Tags and Filters--71
Built-in filter reference---71
add--72
addslashes---72
capfirst--73
center--73
cut--74
date---74
default--75
default_if_none---75
dictsort---76
dictsortreversed--76
divisibleby---77
escape--77
escapejs--78
filesizeformat--78
first---79
fix_ampersands---79
floatformat---79
force_escape---80
get_digit--80
iriencode---81
join--81
last--82
length---82
length_is--83
linebreaks---83
linebreaksbr--84
linenumbers--84
ljust--84
lower--85

make_list---86
phone2numeric---86
pluralize---87
pprint---87
random---88
removetags---88
rjust---89
safe---89
slice---90
slugify---90
stringformat---90
striptags---91
time---92
timesince---92
timeuntil---92
title---92
truncatewords---93
truncatewords_html---94
unordered_list---94
upper---95
urlencode---95
urlize---96
urlizetrunc---96
wordcount---97
wordwrap---97
yesno---98
Built-in tag reference---98
autoescape---99
block---99
comment---100
cycle---100
debug---101
extends---101
filter---101
firstof---102
for---102
forloop---103
if---104
ifchanged---105
ifequal---106
ifnotequal---107
include---107
load---108
now---108
regroup---109
spaceless---110
ssi---111
templatetag---111
url---112
widthratio---113
with---114
Summary---115

Chapter 5: Loading and Inheriting Templates--117
Configuring the template system--117
Finding a home for the template files--120
Working with the template loaders--121
Loading templates manually---121
Choosing a template loader---122
Using the filesystem loader--122
Using the application directories loader---123
About the eggs template loader---124
Using the loaders together---124
Loading your template files---125
Setting up the error handling templates--125
Creating the error templates--126
Testing the error templates---126
Breaking templates into reusable pieces---127
Extending templates with inheritance--127
Using the block tag---128
Extending templates--129
Adding inheritance to the press application--131
Using multiple block tags--132
Adding template inheritance to our press release list---133
Inheriting from multiple child templates---134
Appending to blocks--136
Template strategy--137
Creating content placeholders---137
Extra JS---137
Extra style--138
Extra head content--139
Extra body tag attributes---139
Using include files--140
Using include---140
Using SSI---141
Summary---142
Chapter 6: Serving Multiple Templates--143
Considering the different approaches--143
Serving mobile devices---144
Adapting content---145
Setting up our example---145
Serving printable pages---147
Creating site themes--149
Testing the template overrides--150
Serving different templates by domain name---152
Serving different sites with the development web server--153
Redirecting users to the mobile site (optional)---155
Detecting mobile devices---155
Writing the middleware--156
Checking only once---158
Installing the middleware--159
Summary---161
Chapter 7: Custom Tags and Filters--163
Examining the built-in tags and filters--163
Template filters--164

Template tags--164
Writing your own template filters--165
Setting up a test application---165
Creating a home for our filter library--167
Template filter syntax--168
Loading template libraries---169
U.S. currency filter--169
Replace profanities filter--171
Filters that expect strings---173
In-list filter---174
Writing your own template tags--175
Creating another sample application---176
Adding the template library--177
Template tag syntax--177
A simple tag example--178
The compilation function---179
The template node subclass---179
Registering our custom tag--179
All work and no play tag---180
Passing a template variable to a tag--182
Modifying the context through a tag--184
Summary---186
Chapter 8: Pagination--187
An Overview--187
Verifying our application setup---188
Verifying the application---188
Verifying the configuration---188
Verifying the URL configuration--189
Verifying the model--189
Verifying the view---190
Adding test records--190
Exploring pagination using the Django shell---191
Examining database performance--193
Allowing for empty result sets--195
Preventing orphaned records--195
Using pagination in your views--196
Creating the view--196
Retrieving the current position from the URL--197
Putting navigation into the templates--197
Pagination with generic views--199
Setting up our generic list view--200
Generically calling the last page---201
Summary---201
Chapter 9: Customizing the Admin Look and Feel---203
Overriding the admin templates--203
Leveraging the template loader--204
Locating the admin template files---204
Exploring the admin template files--205
Inspecting the base.html template---205
Inspecting the base_site.html template--206
Inspecting the index.html template--206
Inspecting the change_list.html template--207

Inspecting the change_form.html template---208
Customizing the admin header---208
Replacing the page title---208
Changing the header text---209
Adding a new link box to the admin--209
Overriding the admin index file--210
Creating the include file---211
Customizing the admin color scheme---212
Identifying styles to change---213
Using the extrastyle block---213
Summary--216
Chapter 10: Caching Your Pages--217
An overview--217
Do you need caching?---218
How caching works--218
Exploring the available cache systems---218
Filesystem caching---219
Database caching---219
Memcached---219
Local memory caching--219
Dummy caching---220
Setting up your cache system--220
Configuring the cache backend--220
Database caching---220
Filesystem caching---220
Local memory caching--221
Dummy caching---221
Memcached---221
Adding additional backend arguments--221
Setting up for the examples---222
Caching individual views---222
Adding caching---224
Caching pieces of templates---224
Low-level caching--226
Caching your whole site--227
Preventing data from being cached--229
General caching strategies---229
Working with outside caches--229
Summary--230
Chapter 11: Internationalization--231
Exploring i18n--231
Creating an example application---232
Configuring your project for i18n---236
Installing libraries for i18n translation--236
Marking strings as translatable---237
Creating message files---237
Enabling automatic language preference---242
How Django determines language preference--244
Summary--244
Index---245

***Python_Web_Development_with_Django_2009_405

Introduction--1
Where Web Frameworks Come From---1
A Better Way--2
We’re Not in Kansas Anymore---2
Web Development Is Better with Python and Django--3
I: Getting Started
1 Practical Python for Django---7
Python Skills Are Django Skills---7
Getting Started: Python’s Interactive Interpreter---8
Python Basics--10
Comments--10
Variables and Assignment--10
Operators---11
Python Standard Types--11
Object Boolean Values--12
Numbers--12
Numeric Operators--13
Numeric Built-in and Factory Functions--14
Sequences and Iterables--14
Lists---17
Strings---19
Sequence Built-ins and Factory Functions--25
Mapping Type: Dictionaries---26
Standard Type Summary---28
Flow Control--28
Conditionals--29
Loops--29
Exception Handling---30
The finally Clause--31
Throwing Exceptions with raise---32
Files--33
Functions--34
Declaring and Calling Functions--34
Functions Are First-Class Objects--36
Anonymous Functions--38
*args and **kwargs---40
Decorators---42
Object-Oriented Programming---44
Class Definitions--44
Instantiation---45
Subclassing--46
Inner Classes--46
Regular Expressions--47
The re module---47
Searching Versus Matching--48
Common Gotchas---48
Single-Item Tuples--48
Modules--48
Mutability--50
Constructor Versus Initializer--52
Coding Style (PEP 8 and Beyond)--53

Indent Four Spaces--53
Use Spaces and Not Tabs--53
Don’t Write Single-Line Suites on the Same Line as the Header-----------------------------------54
Create Documentation Strings (aka “docstrings”)--54
Summary--55
2 Django for the Impatient: Building a Blog---57
Creating the Project--58
Running the Development Server--59
Creating the Blog Application---61
Designing Your Model---62
Setting Up the Database--62
Using a Database Server--63
Using SQLite--63
Creating the Tables---64
Setting Up the Automatic admin Application---65
Trying Out the admin---66
Making Your Blog’s Public Side---70
Creating a Template--70
Creating a View Function--71
Creating a URL Pattern--72
Finishing Touches---73
Template Niceties---73
Date-Based Ordering---74
Timestamp Formatting Via a Template Filter--75
Summary--75
3 Starting Out--77
Dynamic Web Site Basics--77
Communication: HTTP, URLs, Requests, Responses--78
Data Storage: SQL and Relational Databases--78
Presentation: Rendering Templates into HTML and Other Formats---------------------------------79
Putting It All Together--79
Understanding Models, Views, and Templates---79
Separating the Layers (MVC)---79
Models--80
Views--81
Templates---81
Overall Django Architecture--82
Core Philosophies of Django---82
Django Tries to Be Pythonic--84
Don’t Repeat Yourself (DRY)--84
Loose Coupling and Flexibility---84
Rapid Development--85
Summary--86
II: Django in Depth
4 Defining and Using Models--89
Defining Models---89
Why Use an ORM?---89
Django’s Rich Field Types---91
Relationships Between Models--93
Model Inheritance---97
Meta Inner Class--100
Admin Registration and Options---101

Using Models--102
Creating and Updating Your Database Using manage.py--103
Query Syntax--104
Utilizing SQL Features Django Doesn’t Provide--112
Summary--116
5 URLs, HTTP Mechanisms, and Views--117
URLs---117
Introduction to URLconfs---117
Replacing Tuples with url---119
Using Multiple patterns Objects--119
Including Other URL Files with include---120
Function Objects Versus Function-Name Strings--121
Modeling HTTP: Requests, Responses, and Middleware---------------------------------------122
Request Objects---123
Response Objects---125
Middleware--126
Views/Logic--127
Just Python Functions--128
Generic Views---128
Semi-generic Views--130
Custom Views---131
Summary--133
6 Templates and Form Processing--135
Templates--135
Understanding Contexts---135
Template Language Syntax---136
Forms---142
Defining Forms--142
Filling Out Forms---147
Validation and Cleaning---149
Form Display--150
Widgets---152
Summary--154

III: Django Applications by Example

7 Photo Gallery--159
The Model--160
Preparing for File Uploads---161
Installing PIL--162
Testing ImageField--163
Building Our Custom File Field--164
Initialization---166
Adding Attributes to the Field--167
Saving and Deleting the Thumbnail---168
Using ThumbnailImageField--169
Setting Up DRY URLs--169
The Item App’s URL Layout--172
Tying It All Together with Templates--173
Summary--179
8 Content Management System--181
What’s a CMS?--181
The Un-CMS: Flatpages--182
Enabling the Flatpages App---182

Flatpage Templates--184
Testing It Out---184
Beyond Flatpages: A Simple Custom CMS---185
Making the Model---186
Imports--188
Completing the Model---188
Controlling Which Stories Are Viewed---189
Working with Markdown---190
URL Patterns in urls.py--192
Admin Views---193
Displaying Content Via Generic Views---196
Template Layout---196
Displaying Stories---198
Adding Search--199
Managing Users---201
Supporting Workflow---202
Possible Enhancements---202
Summary--203
9 Liveblog---205
What Exactly Is Ajax?---205
Why Ajax Is Useful--206
Planning the Application---206
Choosing an Ajax Library---207
Laying Out the Application---208
Putting the Ajax In---211
The Basics--212
The “X” in Ajax (Or XML Versus JSON)--212
Installing the JavaScript Library--213
Setting Up and Testing jQuery--214
Creating the View Function---216
Using the View Function Via JavaScript--217
Summary--219
10 Pastebin---221
Defining the Model---222
Creating the Templates---223
Designing the URLs--225
Trying It Out---226
Limiting Number of Recent Pastes Displayed--229
Syntax Highlighting--230
Cleanup Via Cron Job--231
Summary--232
IV: Advanced Django Techniques and Features
11 Advanced Django Programming--235
Customizing the Admin--235
Changing Layout and Style Using Fieldsets--236
Extending the Base Templates---237
Adding New Views---238
Authentication Decorators---239
Using Syndication--240
The Feed Class---240
Giving the Feed a URL--242
Doing More with Feeds---242

Generating Downloadable Files--243
Nagios Configuration Files--243
vCard---244
Comma-Separated Value (CSV)---245
Charts and Graphs Using PyCha--246
Enhancing Django’s ORM with Custom Managers---248
Changing the Default Set of Objects--248
Adding New Manager Methods---249
Extending the Template System---250
Simple Custom Template Tags--250
Inclusion Tags--253
Custom Filters--256
More Complex Custom Template Tags---258
Alternative Templating--258
Summary--260
12 Advanced Django Deployment---261
Writing Utility Scripts---261
Cronjobs for Cleanup--262
Data Import/Export---263
Customizing the Django Codebase Itself--264
Caching--265
A Basic Caching Recipe---265
Caching Strategies---267
Caching Backend Types--272
Testing Django Applications--274
Doctest Basics--275
Unittest Basics--276
Running Tests---276
Testing Models---276
Testing Your Entire Web App---278
Testing the Django Codebase Itself--279
Summary--281
V: Appendices
A Command Line Basics---285
Putting the “Command” in “Command Line”---285
Options and Arguments--288
Pipes and Redirection--289
Environment Variables--291
The Path---293
Summary--294
B Installing and Running Django---295
Python---295
Mac OS X---296
Unix/Linux--296
Windows---296
Updating Your Path---296
Testing---299
Optional Add-ons---300
Django---301
Packaged Releases---302
Development Version--302
Installation--302

Testing--303
Web Server---303
The Built-In Server: Not for Production---303
The Standard Approach: Apache and
mod_python--304
The Flexible Alternative: WSGI---306
Another Approach: Flup and FastCGI--307
SQL Database---308
SQLite---308
PostgreSQL--309
MySQL--310
Oracle--311
Other Databases---311
Summary--311
C Tools for Practical Django Development---313
Version Control--313
The Trunk and Branches--314
Merging---314
Centralized Version Control---315
Decentralized Version Control--315
Version Control for Your Project---316
Project Management Software--318
Trac---319
Text Editors---319
Emacs--319
Vim--320
TextMate--320
Eclipse---320
D Finding, Evaluating, and Using Django
Applications---321
Where to Look for Applications--321
How to Evaluate Applications---322
How to Use Applications---323
Sharing Your Own Application---323
E Django on the Google App Engine--325
Why the App Engine Matters--325
Pure Google App Engine Applications---326
Limitations of the App Engine Framework--326
Google App Engine Helper for Django---327
Getting the SDK and the Helper--327
More on the Helper---327
Integrating the App Engine--328
Copying the App Engine Code to Your Project---328
Integrating the App Engine Helper--329
Porting Your Application to App Engine---330
Taking a Test Drive---330
Adding Data--331
Creating a New Django Application That Runs
on App Engine---333
Summary--334
Online Resources---334
F Getting Involved in the Django Project---337

Index--339
Colophon---375

***Test-Driven_Development_with_Python_2014_478

Preface--xv
Prerequisites and Assumptions--xxi
Acknowledgments---xxvii
Part I.---The Basics of TDD and Django
1 Getting Django Set Up Using a Functional Test---3
Obey the Testing Goat! Do Nothing Until You Have a Test---3
Getting Django Up and Running--6
Starting a Git Repository--8
2 Extending Our Functional Test Using the unittest Module-------------------------------------13
Using a Functional Test to Scope Out a Minimum Viable App-------------------------------------13
The Python Standard Library’s unittest Module--16
Implicit waits---18
Commit---18
3 Testing a Simple Home Page with Unit Tests---21
Our First Django App, and Our First Unit Test--22
Unit Tests, and How They Differ from Functional Tests---22
Unit Testing in Django--23
Django’s MVC, URLs, and View Functions---24
At Last! We Actually Write Some Application Code!--26
urls.py---27
Unit Testing a View---30
The Unit-Test/Code Cycle--31
4 What Are We Doing with All These Tests?--35
Programming Is like Pulling a Bucket of Water up from a Well--------------------------------------36
Using Selenium to Test User Interactions---37
The “Don’t Test Constants” Rule, and Templates to the Rescue--------------------------------------40
Refactoring to Use a Template---40
On Refactoring---44
A Little More of Our Front Page--45
Recap: The TDD Process--47
5 Saving User Input---51
Wiring Up Our Form to Send a POST Request---51
Processing a POST Request on the Server--54
Passing Python Variables to Be Rendered in the Template---55
Three Strikes and Refactor---59
The Django ORM and Our First Model---60
Our First Database Migration---62
The Test Gets Surprisingly Far--63
A New Field Means a New Migration---64
Saving the POST to the Database--65
Redirect After a POST---68
Better Unit Testing Practice: Each Test Should Test One Thing---68
Rendering Items in the Template--69
Creating Our Production Database with migrate--71
6 Getting to the Minimum Viable Site---77
Ensuring Test Isolation in Functional Tests--77

Running Just the Unit Tests--80
Small Design When Necessary--81
YAGNI!---82
REST---82
Implementing the New Design Using TDD--83
Iterating Towards the New Design--86
Testing Views, Templates, and URLs Together with the Django Test Client------------------------87
A New Test Class--88
A New URL--88
A New View Function--89
A Separate Template for Viewing Lists---90
Another URL and View for Adding List Items--92
A Test Class for New List Creation---93
A URL and View for New List Creation--94
Removing Now-Redundant Code and Tests--95
Pointing Our Forms at the New URL---96
Adjusting Our Models--97
A Foreign Key Relationship---99
Adjusting the Rest of the World to Our New Models--100
Each List Should Have Its Own URL---102
Capturing Parameters from URLs--103
Adjusting new_list to the New World---104
One More View to Handle Adding Items to an Existing List---------------------------------------105
Beware of Greedy Regular Expressions!---106
The Last New URL--106
The Last New View---107
But How to Use That URL in the Form?--108
A Final Refactor Using URL includes--110
Part II. Web Development Sine Qua Nons
7 Prettification: Layout and Styling, and What to Test About It------------------------------------115
What to Functionally Test About Layout and Style--115
Prettification: Using a CSS Framework--118
Django Template Inheritance---120
Integrating Bootstrap--121
Rows and Columns--122
Static Files in Django--122
Switching to StaticLiveServerCase--124
Using Bootstrap Components to Improve the Look of the Site------------------------------------125
Jumbotron!--125
Large Inputs---125
Table Styling--126
Using Our Own CSS---126
What We Glossed Over: collectstatic and Other Static Directories--------------------------------127
A Few Things That Didn’t Make It--130
8 Testing Deployment Using a Staging Site---131
TDD and the Danger Areas of Deployment--132
As Always, Start with a Test--133
Getting a Domain Name---135
Manually Provisioning a Server to Host Our Site---136
Choosing Where to Host Our Site--136
Spinning Up a Server---137
User Accounts, SSH, and Privileges---137

Installing Nginx--138
Configuring Domains for Staging and Live---139
Using the FT to Confirm the Domain Works and Nginx Is Running------------------------------139
Deploying Our Code Manually--140
Adjusting the Database Location--141
Creating a Virtualenv---142
Simple Nginx Configuration---144
Creating the Database with migrate--147
Getting to a Production-Ready Deployment--148
Switching to Gunicorn--148
Getting Nginx to Serve Static Files--149
Switching to Using Unix Sockets--150
Switching DEBUG to False and Setting ALLOWED_HOSTS------------------------------------151
Using Upstart to Make Sure Gunicorn Starts on Boot--151
Saving Our Changes: Adding Gunicorn to Our requirements.txt------------------------------152
Automating--152
“Saving Your Progress”---156
9 Automating Deployment with Fabric--157
Breakdown of a Fabric Script for Our Deployment---158
Trying It Out---162
Deploying to Live--163
Nginx and Gunicorn Config Using sed--165
Git Tag the Release--166
Further Reading--166
10 Input Validation and Test Organisation--169
Validation FT: Preventing Blank Items--169
Skipping a Test--170
Splitting Functional Tests out into Many Files---171
Running a Single Test File---174
Fleshing Out the FT---174
Using Model-Layer Validation--175
Refactoring Unit Tests into Several Files---175
Unit Testing Model Validation and the self.assertRaises Context Manager-----------------------177
A Django Quirk: Model Save Doesn’t Run Validation---178
Surfacing Model Validation Errors in the View---178
Checking Invalid Input Isn’t Saved to the Database--181
Django Pattern: Processing POST Requests in the Same View as Renders the Form-----------183
Refactor: Transferring the new_item Functionality into view_list---------------------------------184
Enforcing Model Validation in view_list--186
Refactor: Removing Hardcoded URLs---187
The {% url %} Template Tag--188
Using get_absolute_url for Redirects---188
11 A Simple Form---193
Moving Validation Logic into a Form---193
Exploring the Forms API with a Unit Test--194
Switching to a Django ModelForm--195
Testing and Customising Form Validation--196
Using the Form in Our Views--198
Using the Form in a View with a GET Request--198
A Big Find and Replace---201
Using the Form in a View That Takes POST Requests--203
Adapting the Unit Tests for the new_list View---203

Using the Form in the View--204
Using the Form to Display Errors in the Template---205
Using the Form in the Other View---205
A Helper Method for Several Short Tests--206
Using the Form's Own Save Method--208
12 More Advanced Forms---211
Another FT for Duplicate Items---211
Preventing Duplicates at the Model Layer---212
A Little Digression on Queryset Ordering and String Representations----------------------------214
Rewriting the Old Model Test--216
Some Integrity Errors Do Show Up on Save---217
Experimenting with Duplicate Item Validation at the Views Layer--------------------------------218
A More Complex Form to Handle Uniqueness Validation---219
Using the Existing List Item Form in the List View---221
13 Dipping Our Toes, Very Tentatively, into JavaScript--225
Starting with an FT--225
Setting Up a Basic JavaScript Test Runner--226
Using jQuery and the Fixtures Div---229
Building a JavaScript Unit Test for Our Desired Functionality------------------------------------232
Javascript Testing in the TDD Cycle--234
Columbo Says: Onload Boilerplate and Namespacing---234
A Few Things That Didn't Make It--235
14 Deploying Our New Code---237
Staging Deploy---237
Live Deploy--237
What to Do If You See a Database Error--238
Wrap-Up: git tag the New Release---238
Part III. More Advanced Topics
15 User Authentication, Integrating Third-Party Plugins, and Mocking with JavaScript--------241
Mozilla Persona (BrowserID)---242
Exploratory Coding, aka "Spiking"---242
Starting a Branch for the Spike---243
Frontend and JavaScript Code--243
The Browser-ID Protocol---244
The Server Side: Custom Authentication---245
De-spiking---251
A Common Selenium Technique: Explicit Waits---253
Reverting Our Spiked Code---255
JavaScript Unit Tests Involving External Components: Our First Mocks!-----------------------256
Housekeeping: A Site-Wide Static Files Folder--256
Mocking: Who, Why, What?---257
Namespacing--258
A Simple Mock to Unit Tests Our initialize Function--258
More Advanced Mocking---264
Checking Call Arguments---267
QUnit setup and teardown, Testing Ajax--268
More Nested Callbacks! Testing Asynchronous Code--272
16 Server-Side Authentication and Mocking in Python---277
A Look at Our Spiked Login View---277
Mocking in Python--278
Testing Our View by Mocking Out authenticate---278
Checking the View Actually Logs the User In---281

De-spiking Our Custom Authentication Backend: Mocking Out an Internet Request-----------285
1 if = 1 More Test--286
Patching at the Class Level---287
Beware of Mocks in Boolean Comparisons--290
Creating a User if Necessary--291
The get_user Method--291
A Minimal Custom User Model---293
A Slight Disappointment---295
Tests as Documentation--296
Users Are Authenticated--297
The Moment of Truth: Will the FT Pass?---298
Finishing Off Our FT, Testing Logout--299
17 Test Fixtures, Logging, and Server-Side Debugging---303
Skipping the Login Process by Pre-creating a Session---303
Checking It Works--305
The Proof Is in the Pudding: Using Staging to Catch Final Bugs----------------------------------306
Setting Up Logging---307
Fixing the Persona Bug---309
Managing the Test Database on Staging--311
A Django Management Command to Create Sessions---311
Getting the FT to Run the Management Command on the Server---------------------------------312
An Additional Hop via subprocess--314
Baking In Our Logging Code--317
Using Hierarchical Logging Config---318
Wrap-Up--320
18 Finishing “My Lists”: Outside-In TDD--323
The Alternative: “Inside Out”--323
Why Prefer “Outside-In”?--323
The FT for “My Lists”--324
The Outside Layer: Presentation and Templates---325
Moving Down One Layer to View Functions (the Controller)-------------------------------------326
Another Pass, Outside-In---327
A Quick Restructure of the Template Inheritance Hierarchy---------------------------------------327
Designing Our API Using the Template--328
Moving Down to the Next Layer: What the View Passes to the Template-----------------------329
The Next “Requirement” from the Views Layer: New Lists Should Record Owner------------330
A Decision Point: Whether to Proceed to the Next Layer with a Failing Test--------------------331
Moving Down to the Model Layer--331
Final Step: Feeding Through the name API from the Template------------------------------------333
19 Test Isolation, and “Listening to Your Tests”---337
Revisiting Our Decision Point: The Views Layer Depends on Unwritten Models Code--------337
A First Attempt at Using Mocks for Isolation--338
Using Mock side_effects to Check the Sequence of Events--339
Listen to Your Tests: Ugly Tests Signal a Need to Refactor---341
Rewriting Our Tests for the View to Be Fully Isolated--342
Keep the Old Integrated Test Suite Around as a Sanity Check--------------------------------------342
A New Test Suite with Full Isolation--343
Thinking in Terms of Collaborators---343
Moving Down to the Forms Layer--347
Keep Listening to Your Tests: Removing ORM Code from Our Application--------------------348
Finally, Moving Down to the Models Layer--351
Back to Views---353

The Moment of Truth (and the Risks of Mocking)--354
Thinking of Interactions Between Layers as “Contracts”---355
Identifying Implicit Contracts---356
Fixing the Oversight---357
One More Test---358
Tidy Up: What to Keep from Our Integrated Test Suite--359
Removing Redundant Code at the Forms Layer---359
Removing the Old Implementation of the View---360
Removing Redundant Code at the Forms Layer---361
Conclusions: When to Write Isolated Versus Integrated Tests-------------------------------------362
Let Complexity Be Your Guide---363
Should You Do Both?--363
Onwards!---363
20 Continuous Integration (CI)--365
Installing Jenkins---365
Configuring Jenkins Security---367
Adding Required Plugins---368
Setting Up Our Project---369
First Build!--371
Setting Up a Virtual Display so the FTs Can Run Headless--372
Taking Screenshots--374
A Common Selenium Problem: Race Conditions---378
Running Our QUnit JavaScript Tests in Jenkins with PhantomJS---------------------------------381
Installing node---382
Adding the Build Steps to Jenkins--383
More Things to Do with a CI Server--384
21 The Token Social Bit, the Page Pattern, and an Exercise for the Reader----------------------387
An FT with Multiple Users, and addCleanup--387
Implementing the Selenium Interact/Wait Pattern---389
The Page Pattern---390
Extend the FT to a Second User, and the “My Lists” Page--393
An Exercise for the Reader---395
22 Fast Tests, Slow Tests, and Hot Lava---397
Thesis: Unit Tests Are Superfast and Good Besides That---398
Faster Tests Mean Faster Development--398
xii |---Table of Contents
The Holy Flow State--399
Slow Tests Don’t Get Run as Often, Which Causes Bad Code------------------------------------399
We’re Fine Now, but Integrated Tests Get Slower Over Time-------------------------------------399
Don’t Take It from Me--399
And Unit Tests Drive Good Design--400
The Problems with “Pure” Unit Tests--400
Isolated Tests Can Be Harder to Read and Write---400
Isolated Tests Don’t Automatically Test Integration--400
Unit Tests Seldom Catch Unexpected Bugs---400
Mocky Tests Can Become Closely Tied to Implementation---------------------------------------400
But All These Problems Can Be Overcome---401
Synthesis: What Do We Want from Our Tests, Anyway?---401
Correctness---401
Clean, Maintainable Code---401
Productive Workflow--402
Evaluate Your Tests Against the Benefits You Want from Them----------------------------------402

Architectural Solutions--402
Ports and Adapters/Hexagonal/Clean Architecture---403
Functional Core, Imperative Shell--403
Conclusion---404
Obey the Testing Goat!---407
A PythonAnywhere--409
B Django Class-Based Views---413
C Provisioning with Ansible---423
D Testing Database Migrations--427
E What to Do Next--433
F Cheat Sheet---437
G Bibliography---441
Index---443

***beautiful-soup-4_2015_66p

1 Getting help--3
2 Quick Start--5
3 Installing Beautiful Soup--9
3.1 Problems after installation--9
3.2 Installing a parser--10
4 Making the soup 11
5 Kinds of objects---13
5.1 Tag---13
5.2 NavigableString--15
5.3 BeautifulSoup--15
5.4 Comments and other special strings--15
6 Navigating the tree 17
6.1 Going down---17
6.2 Going up---20
6.3 Going sideways---21
6.4 Going back and forth--23
7 Searching the tree 25
7.1 Kinds of filters--25
7.2 find_all()---27
7.3 Calling a tag is like calling find_all()---31
7.4 find()---31
7.5 find_parents() and find_parent()---32
7.6 find_next_siblings() and find_next_sibling()--32
7.7 find_previous_siblings() and find_previous_sibling()---33
7.8 find_all_next() and find_next()--33
7.9 find_all_previous() and find_previous()--34
7.10 CSS selectors--34
8 Modifying the tree 37
8.1 Changing tag names and attributes--37
8.2 Modifying string--37
8.3 append()---38
8.4 BeautifulSoup.new_string() and new_tag()--38
8.5--insert() 39
8.6 insert_before() and insert_after()---39
8.7 clear()--39

8.8 extract()--39
8.9 decompose()---40
8.10 replace_with()---40
8.11 wrap()--41
8.12 unwrap()---41
9 Output---43
9.1 Pretty-printing---43
9.2 Non-pretty printing---43
9.3 Output formatters---44
9.4 get_text()---46
10 Specifying the parser to use 47
10.1 Differences between parsers--47
11 Encodings 49
11.1 Output encoding---50
11.2 Unicode, Dammit--51
12 Parsing only part of a document 53
12.1 SoupStrainer---53
13 Troubleshooting 55
13.1 diagnose()--55
13.2 Errors when parsing a document--55
13.3 Version mismatch problems--56
13.4 Parsing XML--56
13.5 Other parser problems---56
13.6 Miscellaneous--56
13.7 Improving Performance--57
14 Beautiful Soup 3 59 Porting code to BS4---59

***Getting_Started_with_Beautiful_Soup_2014_130p

Preface---1
Chapter 1: Installing Beautiful Soup---7
Installing Beautiful Soup--7
Installing Beautiful Soup in Linux---7
Installing Beautiful Soup using package manager---8
Installing Beautiful Soup using pip or easy_install---9
Installing Beautiful Soup using pip---9
Installing Beautiful Soup using easy_install---9
Installing Beautiful Soup in Windows---10
Verifying Python path in Windows---10
Installing Beautiful Soup using setup.py---12
Using Beautiful Soup without installation---12
Verifying the installation---13
Quick reference--13
Summary--14
Chapter 2: Creating a BeautifulSoup Object---15
Creating a BeautifulSoup object---15
Creating a BeautifulSoup object from a string---16
Creating a BeautifulSoup object from a file-like object-----------------------------------16
Creating a BeautifulSoup object for XML parsing---18
Understanding the features argument---19
Tag--22
Accessing the Tag object from BeautifulSoup--22

Name of the Tag object--23
Attributes of a Tag object--23
The NavigableString object--24
Quick reference--24
Summary---25
Chapter 3: Search Using Beautiful Soup--27
Searching in Beautiful Soup---27
Searching with find()---28
Finding the first producer---29
Explaining find()--30
Searching with find_all()--37
Finding all tertiary consumers---37
Understanding parameters used with find_all()--38
Searching for Tags in relation---40
Searching for the parent tags--40
Searching for siblings---42
Searching for next--44
Searching for previous--45
Using search methods to scrape information from a web page--------------------------------46
Quick reference--51
Summary---52
Chapter 4: Navigation Using Beautiful Soup--53
Navigation using Beautiful Soup--53
Navigating down---55
Using the name of the child tag---55
Using predefined attributes--56
Special attributes for navigating down--59
Navigating up---60
The parent attribute--60
The parents attribute---61
Navigating sideways to the siblings--61
The next_sibling attribute--62
The previous_sibling attribute---62
Navigating to the previous and next objects parsed---63
Quick reference--63
Summary---64
Chapter 5: Modifying Content Using Beautiful Soup---65
Modifying Tag using Beautiful Soup--65
Modifying the name property of Tag--66
Modifying the attribute values of Tag---68
Updating the existing attribute value of Tag---68
Adding new attribute values to Tag--69
Deleting the tag attributes--70
Adding a new tag--71
Modifying string contents--73
Using string to modify the string content--74
Adding strings using append(), insert(), and new_string()---75
Deleting tags from the HTML document--77
Deleting the producer using decompose()---77
Deleting the producer using extract()--78
Deleting the contents of a tag using Beautiful Soup--79
Special functions to modify content--80

Quick reference--84
Summary---86
Chapter 6: Encoding Support in Beautiful Soup--87
Encoding in Beautiful Soup--88
Understanding the original encoding of the HTML document---------------------------------89
Specifying the encoding of the HTML document---89
Output encoding---90
Quick reference--92
Summary---92
Chapter 7: Output in Beautiful Soup---93
Formatted printing--93
Unformatted printing--94
Output formatters in Beautiful Soup---95
The minimal formatter--98
The html formatter--98
The None formatter---99
The function formatter--99
Using get_text()---100
Quick reference---101
Summary--102
Chapter 8: Creating a Web Scraper---103
Getting book details from PacktPub.com---103
Finding pages with a list of books--104
Finding book details---107
Getting selling prices from Amazon---109
Getting the selling price from Barnes and Noble---111
Summary--112
Index---113

***HeadFirstPython_2011_494p

Intro---xxiii
1 Meet Python: Everyone Loves Lists---1
2 Sharing Your Code: Modules of Functions--33
3 Files and Exceptions: Dealing with Errors--73
4 Persistence: Saving Data to Files---105
5 Comprehending Data: Work That Data!--139
6 Custom Data Objects: Bundling Code with Data---173
7 Web Development: Putting It All Together---213
8 Mobile App Development: Small Devices---255
9 Manage Your Data: Handling Input--293
10 Scaling Your Webapp: Getting Real--351
11 Dealing with Complexity: Data Wrangling--397
Appendix: Leftovers: The Top Ten Things (We Didn’t Cover)-------------------------------------435
Index---447

***IntroductiontoTornado_2012_136p

1 Introduction---1

What Is Tornado?--------------------2
Getting Started with Tornado--------------------3
Community and Support--------------------3
Simple Web Services--------------------4
Hello Tornado--------------------4
String Service--------------------7
More About RequestHandlers--------------------9
Next Steps--------------------11
2 Forms and Templates--------------------13
Simple Example: Poem Maker Pro--------------------13
Rendering Templates--------------------15
Interpolation--------------------16
Template Syntax--------------------17
Interpolating Expressions--------------------18
Control Flow Statements--------------------18
Using Functions Inside Templates--------------------19
Complete Example: The Alpha Munger--------------------20
How It Works--------------------23
Serving Static Files--------------------25
Next Steps with Templates--------------------26
3 Extending Templates--------------------27
Blocks and Substitutions--------------------27
Basics of Blocks--------------------27
Templates in Practice: Burt’s Books--------------------31
Autoescaping--------------------34
UI Modules--------------------37
Basic Module Usage--------------------38
Modules in Depth--------------------39
Embedding JavaScript and CSS--------------------42
Summing Up--------------------44
4 Databases--------------------47
Basic MongoDB Operations with PyMongo--------------------48
Establishing a Connection--------------------48
Dealing with Documents--------------------49
MongoDB Documents and JSON--------------------51
A Simple Persistent Web Service--------------------52
A Read-Only Dictionary--------------------52
Writing the Dictionary--------------------54
Burt’s Books--------------------56
Reading Books (From the Database)--------------------56
Editing and Adding Books--------------------59
MongoDB: Next Steps--------------------63
5 Asynchronous Web Services--------------------67
Asynchronous Web Requests--------------------67
Starting Synchronous--------------------68
The Trouble with Blocking--------------------70
Basic Asynchronous Calls--------------------72
The asynchronous Decorator and the finish Method--------------------73
Asynchronous Generators--------------------75
Summary of Asynchronous Operations--------------------78
Long Polling with Tornado--------------------78
The Benefits of Long Polling--------------------79

Example: Live Inventory Reporting--80
The Downsides of Long Polling---86
WebSockets with Tornado---87
Tornado’s WebSocket Module---88
Example: Live Inventory with WebSockets---88
The Future of WebSockets--92
6 Writing Secure Applications---93
Cookie Vulnerabilities---93
Cookie Forgery--93
Secure Cookies--93
Request Vulnerabilities--96
Anatomy of a Cross-Site Request Forgery--96
Defending Against Request Forgeries--96
Using Tornado’s XSRF protection---97
User Authentication--98
Example: Welcome Back--98
The authenticated Decorator---100
Summing up--101
7 Authenticating with External Services---103
The Tornado auth Module---103
The Authorization Workflow--103
Asynchronous Requests--104
Example: Sign in With Twitter--104
Example: Facebook Authentication and the Graph API---109
8 Deploying Tornado---115
Reasons for Running Multiple Tornado Instances--115
Using Nginx as a Reverse Proxy--116
Basic Nginx Configuration---116
SSL Decryption with Nginx--118
Using Supervisor to Manage Tornado Processes---119

***Kivy__InteractiveApplicationsinPython_2013_138p

Preface--1
Chapter 1: GUI Basics – Building an Interface---7
Hello World!--8
Basic widgets – labels and buttons---11
Layouts---14
Embedding layouts---18
Our Project – comic creator--22
Summary---28
Chapter 2: Graphics – The Canvas--31
Basic shapes---32
Images, colors, and backgrounds--38
Rotating, translating, and scaling--41
Comic creator – PushMatrix and PopMatrix--44
Summary---48
Chapter 3: Widget Events – Binding Actions--51
Attributes, id and root---52
Basic widget events – dragging the stickman---54
Localizing coordinates – adding stickmen---59

Binding and unbinding events – sizing limbs and heads--62
Binding events in the Kivy language--67
Creating your own events – the magical properties---69
Kivy and properties--72
Summary---75
Chapter 4: Improving the User Experience---77
Screen manager – selecting colors for the figures---78
Color Control on the canvas – coloring figures---81
StencilView – limiting the drawing space---84
Scatter – multitouching to drag, rotate, and scale--85
Recording gestures – line, circles, and cross---89
Simple gestures – drawing with the finger--91
Summary---95
Chapter 5: Invaders Revenge – An Interactive Multitouch Game------------------------------------97
Invaders Revenge – an animated multitouch game---98
Atlas – efficient management of images---99
Boom – simple sound effects---101
Ammo – simple animation--102
Invader – transitions for animations--103
Dock – automatic binding in the Kivy language--105
Fleet – infinite concatenation of animations--107
Scheduling events with the Clock---108
Shooter – multitouch control--110
Invasion – moving the shooter with the keyboard--113
Combining animations with '+' and '&'--115
Summary---117
Index---119

***Learning_Python_5thEdition_2013_1594p

To Vera.
You are my life.

Preface xxxiii

Part I Getting Started

1 A Python Q&A Session--3
Why Do People Use Python?---3
Software Quality---4
Developer Productivity---5
Is Python a “Scripting Language”?---5
OK, but What’s the Downside?--7
Who Uses Python Today?---9
What Can I Do with Python?--10
Systems Programming--11
GUIs--11
Internet Scripting--11
Component Integration--12
Database Programming--12
Rapid Prototyping---13
Numeric and Scientific Programming--13
And More: Gaming, Images, Data Mining, Robots, Excel---14
How Is Python Developed and Supported?---15

Open Source Tradeoffs--15
What Are Python’s Technical Strengths?---16
It’s Object-Oriented and Functional---16
It’s Free---17
It’s Portable--17
It’s Powerful--18
It’s Mixable--19
It’s Relatively Easy to Use--19
It’s Relatively Easy to Learn---20
It’s Named After Monty Python--20
How Does Python Stack Up to Language X?---21
Chapter Summary--22
Test Your Knowledge: Quiz---23
Test Your Knowledge: Answers---23
2 How Python Runs Programs---27
Introducing the Python Interpreter--27
Program Execution--28
The Programmer’s View---28
Python’s View---30
Execution Model Variations--33
Python Implementation Alternatives---33
Execution Optimization Tools---37
Frozen Binaries---39
Future Possibilities?--40
Chapter Summary--40
Test Your Knowledge: Quiz---41
Test Your Knowledge: Answers---41
3 How You Run Programs--43
The Interactive Prompt---43
Starting an Interactive Session---44
The System Path--45
New Windows Options in 3.3: PATH, Launcher--46
Where to Run: Code Directories---47
What Not to Type: Prompts and Comments--48
Running Code Interactively--49
Why the Interactive Prompt?---50
Usage Notes: The Interactive Prompt--52
System Command Lines and Files---54
A First Script---55
Running Files with Command Lines--56
Command-Line Usage Variations---57
Usage Notes: Command Lines and Files---58
Unix-Style Executable Scripts: #!--59
Unix Script Basics---59
The Unix env Lookup Trick---60
The Python 3.3 Windows Launcher: #! Comes to Windows--60
Clicking File Icons--62
Icon-Click Basics--62
Clicking Icons on Windows---63
The input Trick on Windows---63
Other Icon-Click Limitations---66
Module Imports and Reloads--66

Import and Reload Basics--66
The Grander Module Story: Attributes--68
Usage Notes: import and reload---71
Using exec to Run Module Files--72
The IDLE User Interface---73
IDLE Startup Details--74
IDLE Basic Usage---75
IDLE Usability Features--76
Advanced IDLE Tools--77
Usage Notes: IDLE---78
Other IDEs--79
Other Launch Options---81
Embedding Calls---81
Frozen Binary Executables---82
Text Editor Launch Options--82
Still Other Launch Options---82
Future Possibilities?--83
Which Option Should I Use?---83
Chapter Summary--85
Test Your Knowledge: Quiz---85
Test Your Knowledge: Answers---86
Test Your Knowledge: Part I Exercises---87
Part II Types and Operations
4 Introducing Python Object Types--93
The Python Conceptual Hierarchy---93
Why Use Built-in Types?--94
Python’s Core Data Types--95
Numbers--97
Strings--99
Sequence Operations---99
Immutability---101
Type-Specific Methods--102
Getting Help---104
Other Ways to Code Strings---105
Unicode Strings--106
Pattern Matching---108
Lists---109
Sequence Operations--109
Type-Specific Operations--109
Bounds Checking--110
Nesting---110
Comprehensions---111
Dictionaries--113
Mapping Operations---114
Nesting Revisited--115
Missing Keys: if Tests---116
Sorting Keys: for Loops---118
Iteration and Optimization---120
Tuples--121
Why Tuples?---122
Files--122
Binary Bytes Files--123

Unicode Text Files------124
Other File-Like Tools------126
Other Core Types------126
How to Break Your Code’s Flexibility------128
User-Defined Classes------129
And Everything Else------130
Chapter Summary------130
Test Your Knowledge: Quiz------131
Test Your Knowledge: Answers------131
5 Numeric Types------133
Numeric Type Basics------133
Numeric Literals------134
Built-in Numeric Tools------136
Python Expression Operators------136
Numbers in Action------141
Variables and Basic Expressions------141
Numeric Display Formats------143
Comparisons: Normal and Chained------144
Division: Classic, Floor, and True------146
Integer Precision------150
Complex Numbers------151
Hex, Octal, Binary: Literals and Conversions------151
Bitwise Operations------153
Other Built-in Numeric Tools------155
Other Numeric Types------157
Decimal Type------157
Fraction Type------160
Sets------163
Booleans------171
Numeric Extensions------172
Chapter Summary------172
Test Your Knowledge: Quiz------173
Test Your Knowledge: Answers------173
6 The Dynamic Typing Interlude------175
The Case of the Missing Declaration Statements------175
Variables, Objects, and References------176
Types Live with Objects, Not Variables------177
Objects Are Garbage-Collected------178
Shared References------180
Shared References and In-Place Changes------181
Shared References and Equality------183
Dynamic Typing Is Everywhere------185
Chapter Summary------186
Test Your Knowledge: Quiz------186
Test Your Knowledge: Answers------186
7 String Fundamentals------189
This Chapter’s Scope------189
Unicode: The Short Story------189
String Basics------190
String Literals------192
Single- and Double-Quoted Strings Are the Same------193
Escape Sequences Represent Special Characters------193

Raw Strings Suppress Escapes--196
Triple Quotes Code Multiline Block Strings---198
Strings in Action---200
Basic Operations--200
Indexing and Slicing--201
String Conversion Tools--205
Changing Strings I---208
String Methods---209
Method Call Syntax---209
Methods of Strings--210
String Method Examples: Changing Strings II---211
String Method Examples: Parsing Text--213
Other Common String Methods in Action--214
The Original string Module’s Functions (Gone in 3.X)---215
String Formatting Expressions--216
Formatting Expression Basics---217
Advanced Formatting Expression Syntax---218
Advanced Formatting Expression Examples---220
Dictionary-Based Formatting Expressions---221
String Formatting Method Calls--222
Formatting Method Basics--222
Adding Keys, Attributes, and Offsets---223
Advanced Formatting Method Syntax--224
Advanced Formatting Method Examples---225
Comparison to the % Formatting Expression--227
Why the Format Method?--230
General Type Categories---235
Types Share Operation Sets by Categories--235
Mutable Types Can Be Changed in Place--236
Chapter Summary---237
Test Your Knowledge: Quiz--237
Test Your Knowledge: Answers--237
8 Lists and Dictionaries---239
Lists--239
Lists in Action---242
Basic List Operations--242
List Iteration and Comprehensions--242
Indexing, Slicing, and Matrixes---243
Changing Lists in Place--244
Dictionaries--250
Dictionaries in Action---252
Basic Dictionary Operations---253
Changing Dictionaries in Place--254
More Dictionary Methods--254
Example: Movie Database--256
Dictionary Usage Notes--258
Other Ways to Make Dictionaries--262
Dictionary Changes in Python 3.X and 2.7--264
Chapter Summary---271
Test Your Knowledge: Quiz--272
Test Your Knowledge: Answers--272
9 Tuples, Files, and Everything Else---275

Tuples--276
Tuples in Action--277
Why Lists and Tuples?--279
Records Revisited: Named Tuples---280
Files--282
Opening Files---283
Using Files--284
Files in Action---285
Text and Binary Files: The Short Story---287
Storing Python Objects in Files: Conversions--288
Storing Native Python Objects: pickle--290
Storing Python Objects in JSON Format--291
Storing Packed Binary Data: struct---293
File Context Managers---294
Other File Tools---294
Core Types Review and Summary---295
Object Flexibility--297
References Versus Copies---297
Comparisons, Equality, and Truth---300
The Meaning of True and False in Python--304
Python’s Type Hierarchies---306
Type Objects--306
Other Types in Python--308
Built-in Type Gotchas---308
Assignment Creates References, Not Copies--308
Repetition Adds One Level Deep--309
Beware of Cyclic Data Structures---310
Immutable Types Can’t Be Changed in Place---311
Chapter Summary---311
Test Your Knowledge: Quiz---311
Test Your Knowledge: Answers--312
Test Your Knowledge: Part II Exercises---313
Part III Statements and Syntax
10 Introducing Python Statements---319
The Python Conceptual Hierarchy Revisited--319
Python’s Statements--320
A Tale of Two ifs---322
What Python Adds--322
What Python Removes--323
Why Indentation Syntax?---324
A Few Special Cases--327
A Quick Example: Interactive Loops---329
A Simple Interactive Loop--329
Doing Math on User Inputs---331
Handling Errors by Testing Inputs---332
Handling Errors with try Statements---333
Nesting Code Three Levels Deep--335
Chapter Summary---336
Test Your Knowledge: Quiz---336
Test Your Knowledge: Answers--336
11 Assignments, Expressions, and Prints--339
Assignment Statements--339

Assignment Statement Forms--340
Sequence Assignments---341
Extended Sequence Unpacking in Python 3.X--344
Multiple-Target Assignments---348
Augmented Assignments--350
Variable Name Rules--352
Expression Statements--356
Expression Statements and In-Place Changes--357
Print Operations--358
The Python 3.X print Function---359
The Python 2.X print Statement--361
Print Stream Redirection--363
Version-Neutral Printing--366
Chapter Summary--369
Test Your Knowledge: Quiz--370
Test Your Knowledge: Answers--370
12 if Tests and Syntax Rules--371
if Statements--371
General Format---371
Basic Examples---372
Multiway Branching---372
Python Syntax Revisited--375
Block Delimiters: Indentation Rules--376
Statement Delimiters: Lines and Continuations---378
A Few Special Cases---379
Truth Values and Boolean Tests---380
The if/else Ternary Expression--382
Chapter Summary--385
Test Your Knowledge: Quiz--385
Test Your Knowledge: Answers--386
13 while and for Loops--387
while Loops---387
General Format---388
Examples---388
break, continue, pass, and the Loop else---389
General Loop Format--389
pass--390
continue--391
break---391
Loop else---392
for Loops---395
General Format---395
Examples---395
Loop Coding Techniques--402
Counter Loops: range--402
Sequence Scans: while and range Versus for--403
Sequence Shufflers: range and len---404
Nonexhaustive Traversals: range Versus Slices--405
Changing Lists: range Versus Comprehensions---406
Parallel Traversals: zip and map---407
Generating Both Offsets and Items: enumerate--410
Chapter Summary--413

Test Your Knowledge: Quiz----------414
Test Your Knowledge: Answers----------414
14 Iterations and Comprehensions----------415
Iterations: A First Look----------416
The Iteration Protocol: File Iterators----------416
Manual Iteration: iter and next----------419
Other Built-in Type Iterables----------422
List Comprehensions: A First Detailed Look----------424
List Comprehension Basics----------425
Using List Comprehensions on Files----------426
Extended List Comprehension Syntax----------427
Other Iteration Contexts----------429
New Iterables in Python 3.X----------434
Impacts on 2.X Code: Pros and Cons----------434
The range Iterable----------435
The map, zip, and filter Iterables----------436
Multiple Versus Single Pass Iterators----------438
Dictionary View Iterables----------439
Other Iteration Topics----------440
Chapter Summary----------441
Test Your Knowledge: Quiz----------441
Test Your Knowledge: Answers----------441
15 The Documentation Interlude----------443
Python Documentation Sources----------443
# Comments----------444
The dir Function----------444
Docstrings: __doc__----------446
PyDoc: The help Function----------449
PyDoc: HTML Reports----------452
Beyond docstrings: Sphinx----------461
The Standard Manual Set----------461
Web Resources----------462
Published Books----------463
Common Coding Gotchas----------463
Chapter Summary----------465
Test Your Knowledge: Quiz----------466
Test Your Knowledge: Answers----------466
Test Your Knowledge: Part III Exercises----------467
Part IV Functions and Generators
16 Function Basics----------473
Why Use Functions?----------474
Coding Functions----------475
def Statements----------476
def Executes at Runtime----------477
A First Example: Definitions and Calls----------478
Definition----------478
Calls----------478
Polymorphism in Python----------479
A Second Example: Intersecting Sequences----------480
Definition----------481
Calls----------481
Polymorphism Revisited----------482

Local Variables 483

Chapter Summary 483

Test Your Knowledge: Quiz 483

Test Your Knowledge: Answers 484

17 Scopes 485

Python Scope Basics 485

Scope Details 486

Name Resolution: The LEGB Rule 488

Scope Example 490

The Built-in Scope 491

The global Statement 494

Program Design: Minimize Global Variables 495

Program Design: Minimize Cross-File Changes 497

Other Ways to Access Globals 498

Scopes and Nested Functions 499

Nested Scope Details 500

Nested Scope Examples 500

Factory Functions: Closures 501

Retaining Enclosing Scope State with Defaults 504

The nonlocal Statement in 3.X 508

nonlocal Basics 508

nonlocal in Action 509

Why nonlocal? State Retention Options 512

State with nonlocal: 3.X only 512

State with Globals: A Single Copy Only 513

State with Classes: Explicit Attributes (Preview) 513

State with Function Attributes: 3.X and 2.X 515

Chapter Summary 519

Test Your Knowledge: Quiz 519

Test Your Knowledge: Answers 520

18 Arguments 523

Argument-Passing Basics 523

Arguments and Shared References 524

Avoiding Mutable Argument Changes 526

Simulating Output Parameters and Multiple Results 527

Special Argument-Matching Modes 528

Argument Matching Basics 529

Argument Matching Syntax 530

The Gritty Details 531

Keyword and Default Examples 532

Arbitrary Arguments Examples 534

Python 3.X Keyword-Only Arguments 539

The min Wakeup Call! 542

Full Credit 542

Bonus Points 544

The Punch Line 544

Generalized Set Functions 545

Emulating the Python 3.X print Function 547

Using Keyword-Only Arguments 548

Chapter Summary 550

Test Your Knowledge: Quiz 551

Test Your Knowledge: Answers 552

19 Advanced Function Topics---553
Function Design Concepts---553
Recursive Functions--555
Summation with Recursion---555
Coding Alternatives---556
Loop Statements Versus Recursion---557
Handling Arbitrary Structures--558
Function Objects: Attributes and Annotations---562
Indirect Function Calls: “First Class” Objects---562
Function Introspection--563
Function Attributes---564
Function Annotations in 3.X---565
Anonymous Functions: lambda---567
lambda Basics--568
Why Use lambda?---569
How (Not) to Obfuscate Your Python Code--571
Scopes: lambdas Can Be Nested Too---572
Functional Programming Tools---574
Mapping Functions over Iterables: map--574
Selecting Items in Iterables: filter--576
Combining Items in Iterables: reduce---576
Chapter Summary--578
Test Your Knowledge: Quiz---578
Test Your Knowledge: Answers---578
20 Comprehensions and Generations---581
List Comprehensions and Functional Tools---581
List Comprehensions Versus map---582
Adding Tests and Nested Loops: filter--583
Example: List Comprehensions and Matrixes---586
Don’t Abuse List Comprehensions: KISS---588
Generator Functions and Expressions--591
Generator Functions: yield Versus return--592
Generator Expressions: Iterables Meet Comprehensions--597
Generator Functions Versus Generator Expressions--602
Generators Are Single-Iteration Objects---604
Generation in Built-in Types, Tools, and Classes--606
Example: Generating Scrambled Sequences--609
Don’t Abuse Generators: EIBTI---614
Example: Emulating zip and map with Iteration Tools---617
Comprehension Syntax Summary--622
Scopes and Comprehension Variables--623
Comprehending Set and Dictionary Comprehensions---624
Extended Comprehension Syntax for Sets and Dictionaries---------------------------------------625
Chapter Summary--626
Test Your Knowledge: Quiz---626
Test Your Knowledge: Answers---626
21 The Benchmarking Interlude---629
Timing Iteration Alternatives--629
Timing Module: Homegrown---630
Timing Script---634
Timing Results--635
Timing Module Alternatives---638

Other Suggestions--642
Timing Iterations and Pythons with timeit---642
Basic timeit Usage---643
Benchmark Module and Script: timeit---647
Benchmark Script Results---649
More Fun with Benchmarks--651
Other Benchmarking Topics: pystones--656
Function Gotchas---656
Local Names Are Detected Statically--657
Defaults and Mutable Objects---658
Functions Without returns--660
Miscellaneous Function Gotchas---661
Chapter Summary--661
Test Your Knowledge: Quiz--662
Test Your Knowledge: Answers---662
Test Your Knowledge: Part IV Exercises--663
Part V Modules and Packages
22 Modules: The Big Picture---669
Why Use Modules?---669
Python Program Architecture--670
How to Structure a Program---671
Imports and Attributes--671
Standard Library Modules---673
How Imports Work--674
1 Find It--674
2 Compile It (Maybe)--675
3 Run It--675
Byte Code Files: __pycache__ in Python 3.2+--676
Byte Code File Models in Action---677
The Module Search Path--678
Configuring the Search Path--681
Search Path Variations---681
The sys.path List---681
Module File Selection--682
Chapter Summary--685
Test Your Knowledge: Quiz--685
Test Your Knowledge: Answers---685
23 Module Coding Basics--687
Module Creation---687
Module Filenames---687
Other Kinds of Modules---688
Module Usage---688
The import Statement--689
The from Statement---689
The from * Statement--689
Imports Happen Only Once--690
import and from Are Assignments---691
import and from Equivalence--692
Potential Pitfalls of the from Statement--693
Module Namespaces---694
Files Generate Namespaces--695
Namespace Dictionaries: __dict__--696

Attribute Name Qualification--697
Imports Versus Scopes---698
Namespace Nesting---699
Reloading Modules---700
reload Basics---701
reload Example--702
Chapter Summary---703
Test Your Knowledge: Quiz--704
Test Your Knowledge: Answers---704
24 Module Packages---707
Package Import Basics--708
Packages and Search Path Settings---708
Package __init__.py Files--709
Package Import Example---711
from Versus import with Packages--713
Why Use Package Imports?---713
A Tale of Three Systems--714
Package Relative Imports---717
Changes in Python 3.X--718
Relative Import Basics---718
Why Relative Imports?--720
The Scope of Relative Imports---722
Module Lookup Rules Summary--723
Relative Imports in Action--723
Pitfalls of Package-Relative Imports: Mixed Use---729
Python 3.3 Namespace Packages---734
Namespace Package Semantics---735
Impacts on Regular Packages: Optional __init__.py--736
Namespace Packages in Action---737
Namespace Package Nesting---738
Files Still Have Precedence over Directories--740
Chapter Summary---742
Test Your Knowledge: Quiz--742
Test Your Knowledge: Answers---742
25 Advanced Module Topics---745
Module Design Concepts---745
Data Hiding in Modules---747
Minimizing from * Damage: _X and __all__--747
Enabling Future Language Features: __future__--748
Mixed Usage Modes: __name__ and __main__---749
Unit Tests with __name__---750
Example: Dual Mode Code---751
Currency Symbols: Unicode in Action--754
Docstrings: Module Documentation at Work---756
Changing the Module Search Path---756
The as Extension for import and from---758
Example: Modules Are Objects---759
Importing Modules by Name String---761
Running Code Strings---762
Direct Calls: Two Options--762
Example: Transitive Module Reloads--763
A Recursive Reloader---764

Alternative Codings---767
Module Gotchas---770
Module Name Clashes: Package and Package-Relative Imports-----------------------------------771
Statement Order Matters in Top-Level Code--771
from Copies Names but Doesn’t Link--772
from * Can Obscure the Meaning of Variables--773
reload May Not Impact from Imports--773
reload, from, and Interactive Testing--774
Recursive from Imports May Not Work---775
Chapter Summary---776
Test Your Knowledge: Quiz--777
Test Your Knowledge: Answers--777
Test Your Knowledge: Part V Exercises---778
Part VI Classes and OOP
26 OOP: The Big Picture--783
Why Use Classes?---784
OOP from 30,000 Feet---785
Attribute Inheritance Search---785
Classes and Instances--788
Method Calls---788
Coding Class Trees---789
Operator Overloading---791
OOP Is About Code Reuse---792
Chapter Summary--795
Test Your Knowledge: Quiz--795
Test Your Knowledge: Answers--795
27 Class Coding Basics---797
Classes Generate Multiple Instance Objects--797
Class Objects Provide Default Behavior--798
Instance Objects Are Concrete Items---798
A First Example---799
Classes Are Customized by Inheritance--801
A Second Example--802
Classes Are Attributes in Modules---804
Classes Can Intercept Python Operators---805
A Third Example--806
Why Use Operator Overloading?--808
The World’s Simplest Python Class---809
Records Revisited: Classes Versus Dictionaries--812
Chapter Summary--814
Test Your Knowledge: Quiz--815
Test Your Knowledge: Answers--815
28 A More Realistic Example---817
Step 1: Making Instances--818
Coding Constructors--818
Testing As You Go--819
Using Code Two Ways--820
Step 2: Adding Behavior Methods---822
Coding Methods--824
Step 3: Operator Overloading--826
Providing Print Displays--826
Step 4: Customizing Behavior by Subclassing--828

Coding Subclasses--828
Augmenting Methods: The Bad Way---829
Augmenting Methods: The Good Way---829
Polymorphism in Action---832
Inherit, Customize, and Extend---833
OOP: The Big Idea--833
Step 5: Customizing Constructors, Too---834
OOP Is Simpler Than You May Think--836
Other Ways to Combine Classes---836
Step 6: Using Introspection Tools---840
Special Class Attributes---840
A Generic Display Tool--842
Instance Versus Class Attributes--843
Name Considerations in Tool Classes---844
Our Classes’ Final Form---845
Step 7 (Final): Storing Objects in a Database---847
Pickles and Shelves---847
Storing Objects on a Shelve Database---848
Exploring Shelves Interactively--849
Updating Objects on a Shelve--851
Future Directions--853
Chapter Summary---855
Test Your Knowledge: Quiz---855
Test Your Knowledge: Answers--856
29 Class Coding Details--859
The class Statement---859
General Form--860
Example--860
Methods--862
Method Example--863
Calling Superclass Constructors---864
Other Method Call Possibilities--864
Inheritance---865
Attribute Tree Construction--865
Specializing Inherited Methods--866
Class Interface Techniques---867
Abstract Superclasses--869
Namespaces: The Conclusion--872
Simple Names: Global Unless Assigned---872
Attribute Names: Object Namespaces--872
The “Zen” of Namespaces: Assignments Classify Names--873
Nested Classes: The LEGB Scopes Rule Revisited--875
Namespace Dictionaries: Review---878
Namespace Links: A Tree Climber--880
Documentation Strings Revisited---882
Classes Versus Modules--884
Chapter Summary---884
Test Your Knowledge: Quiz---884
Test Your Knowledge: Answers--885
30 Operator Overloading---887
The Basics---887
Constructors and Expressions: __init__ and __sub__--888

Common Operator Overloading Methods--888
Indexing and Slicing: __getitem__ and __setitem__---890
Intercepting Slices---891
Slicing and Indexing in Python 2.X--893
But 3.X’s __index__ Is Not Indexing!--894
Index Iteration: __getitem__--894
Iterable Objects: __iter__ and __next__---895
User-Defined Iterables--896
Multiple Iterators on One Object--899
Coding Alternative: __iter__ plus yield---902
Membership: __contains__, __iter__, and __getitem__--906
Attribute Access: __getattr__ and __setattr__---909
Attribute Reference---909
Attribute Assignment and Deletion---910
Other Attribute Management Tools--912
Emulating Privacy for Instance Attributes: Part 1---912
String Representation: __repr__ and __str__---913
Why Two Display Methods?--914
Display Usage Notes---916
Right-Side and In-Place Uses: __radd__ and __iadd__--917
Right-Side Addition---917
In-Place Addition---920
Call Expressions: __call__--921
Function Interfaces and Callback-Based Code---923
Comparisons: __lt__, __gt__, and Others---925
The __cmp__ Method in Python 2.X--926
Boolean Tests: __bool__ and __len__---927
Boolean Methods in Python 2.X---928
Object Destruction: __del__---929
Destructor Usage Notes--930
Chapter Summary--931
Test Your Knowledge: Quiz---931
Test Your Knowledge: Answers--931
31 Designing with Classes---933
Python and OOP--933
Polymorphism Means Interfaces, Not Call Signatures--934
OOP and Inheritance: “Is-a” Relationships---935
OOP and Composition: “Has-a” Relationships--937
Stream Processors Revisited---938
OOP and Delegation: “Wrapper” Proxy Objects---942
Pseudoprivate Class Attributes--944
Name Mangling Overview--945
Why Use Pseudoprivate Attributes?---945
Methods Are Objects: Bound or Unbound---948
Unbound Methods Are Functions in 3.X--950
Bound Methods and Other Callable Objects---951
Classes Are Objects: Generic Object Factories--954
Why Factories?--955
Multiple Inheritance: “Mix-in” Classes--956
Coding Mix-in Display Classes---957
Other Design-Related Topics---977
Chapter Summary--977

Test Your Knowledge: Quiz--978
Test Your Knowledge: Answers---978
32 Advanced Class Topics--979
Extending Built-in Types---980
Extending Types by Embedding---980
Extending Types by Subclassing---981
The “New Style” Class Model---983
Just How New Is New-Style?--984
New-Style Class Changes---985
Attribute Fetch for Built-ins Skips Instances--987
Type Model Changes---992
All Classes Derive from “object”--995
Diamond Inheritance Change---997
More on the MRO: Method Resolution Order--1001
Example: Mapping Attributes to Inheritance Sources---1004
New-Style Class Extensions--1010
Slots: Attribute Declarations--1010
Properties: Attribute Accessors--1020
__getattribute__ and Descriptors: Attribute Tools---1023
Other Class Changes and Extensions---1023
Static and Class Methods---1024
Why the Special Methods?--1024
Static Methods in 2.X and 3.X--1025
Static Method Alternatives--1027
Using Static and Class Methods--1028
Counting Instances with Static Methods--1030
Counting Instances with Class Methods--1031
Decorators and Metaclasses: Part 1---1034
Function Decorator Basics---1035
A First Look at User-Defined Function Decorators--1037
A First Look at Class Decorators and Metaclasses--1038
For More Details--1040
The super Built-in Function: For Better or Worse?---1041
The Great super Debate---1041
Traditional Superclass Call Form: Portable, General--1042
Basic super Usage and Its Tradeoffs---1043
The super Upsides: Tree Changes and Dispatch---1049
Runtime Class Changes and super---1049
Cooperative Multiple Inheritance Method Dispatch--1050
The super Summary---1062
Class Gotchas--1064
Changing Class Attributes Can Have Side Effects--1064
Changing Mutable Class Attributes Can Have Side Effects, Too-------------------------------1065
Multiple Inheritance: Order Matters--1066
Scopes in Methods and Classes--1068
Miscellaneous Class Gotchas---1069
KISS Revisited: “Overwrapping-itis”---1070
Chapter Summary--1070
Test Your Knowledge: Quiz---1071
Test Your Knowledge: Answers---1071
Test Your Knowledge: Part VI Exercises---1072
Part VII Exceptions and Tools

33 Exception Basics--1081
Why Use Exceptions?--1081
Exception Roles---1082
Exceptions: The Short Story---1083
Default Exception Handler---1083
Catching Exceptions--1084
Raising Exceptions---1085
User-Defined Exceptions---1086
Termination Actions---1087
Chapter Summary---1089
Test Your Knowledge: Quiz--1090
Test Your Knowledge: Answers--1090
34 Exception Coding Details---1093
The try/except/else Statement---1093
How try Statements Work---1094
try Statement Clauses--1095
The try else Clause---1098
Example: Default Behavior--1098
Example: Catching Built-in Exceptions---1100
The try/finally Statement--1100
Example: Coding Termination Actions with try/finally--1101
Unified try/except/finally---1102
Unified try Statement Syntax--1104
Combining finally and except by Nesting---1104
Unified try Example---1105
The raise Statement---1106
Raising Exceptions--1107
Scopes and try except Variables--1108
Propagating Exceptions with raise--1110
Python 3.X Exception Chaining: raise from---1110
The assert Statement---1112
Example: Trapping Constraints (but Not Errors!)---1113
with/as Context Managers--1114
Basic Usage--1114
The Context Management Protocol--1116
Multiple Context Managers in 3.1, 2.7, and Later---1118
Chapter Summary---1119
Test Your Knowledge: Quiz--1120
Test Your Knowledge: Answers--1120
35 Exception Objects---1123
Exceptions: Back to the Future--1124
String Exceptions Are Right Out!--1124
Class-Based Exceptions--1125
Coding Exceptions Classes--1126
Why Exception Hierarchies?---1128
Built-in Exception Classes---1131
Built-in Exception Categories--1132
Default Printing and State--1133
Custom Print Displays--1135
Custom Data and Behavior--1136
Providing Exception Details---1136
Providing Exception Methods--1137

Chapter Summary--1139
Test Your Knowledge: Quiz---1139
Test Your Knowledge: Answers---1139
36 Designing with Exceptions--1141
Nesting Exception Handlers---1141
Example: Control-Flow Nesting--1143
Example: Syntactic Nesting--1143
Exception Idioms---1145
Breaking Out of Multiple Nested Loops: “go to”--1145
Exceptions Aren’t Always Errors--1146
Functions Can Signal Conditions with raise--1147
Closing Files and Server Connections--1148
Debugging with Outer try Statements--1149
Running In-Process Tests---1149
More on sys.exc_info---1150
Displaying Errors and Tracebacks---1151
Exception Design Tips and Gotchas---1152
What Should Be Wrapped---1152
Catching Too Much: Avoid Empty except and Exception--------------------------------1153
Catching Too Little: Use Class-Based Categories--1155
Core Language Summary--1155
The Python Toolset---1156
Development Tools for Larger Projects---1157
Chapter Summary--1160
Test Your Knowledge: Quiz---1161
Test Your Knowledge: Answers---1161
Test Your Knowledge: Part VII Exercises--1161
Part VIII Advanced Topics
37 Unicode and Byte Strings--1165
String Changes in 3.X--1166
String Basics---1167
Character Encoding Schemes---1167
How Python Stores Strings in Memory---1170
Python’s String Types--1171
Text and Binary Files---1173
Coding Basic Strings 1174
Python 3.X String Literals---1175
Python 2.X String Literals---1176
String Type Conversions---1177
Coding Unicode Strings 1178
Coding ASCII Text--1178
Coding Non-ASCII Text---1179
Encoding and Decoding Non-ASCII text--1180
Other Encoding Schemes--1181
Byte String Literals: Encoded Text---1183
Converting Encodings--1184
Coding Unicode Strings in Python 2.X---1185
Source File Character Set Encoding Declarations--1187
Using 3.X bytes Objects 1189
Method Calls---1189
Sequence Operations---1190
Other Ways to Make bytes Objects---1191

Mixing String Types--1192
Using 3.X/2.6+ bytearray Objects 1192
bytearrays in Action--1193
Python 3.X String Types Summary---1195
Using Text and Binary Files 1195
Text File Basics--1196
Text and Binary Modes in 2.X and 3.X--1197
Type and Content Mismatches in 3.X---1198
Using Unicode Files 1199
Reading and Writing Unicode in 3.X--1199
Handling the BOM in 3.X--1201
Unicode Files in 2.X---1204
Unicode Filenames and Streams--1205
Other String Tool Changes in 3.X 1206
The re Pattern-Matching Module--1206
The struct Binary Data Module---1207
The pickle Object Serialization Module--1209
XML Parsing Tools--1211
Chapter Summary 1215
Test Your Knowledge: Quiz 1215
Test Your Knowledge: Answers 1216
38 Managed Attributes--1219
Why Manage Attributes?--1219
Inserting Code to Run on Attribute Access--1220
Properties--1221
The Basics--1222
A First Example---1222
Computed Attributes---1224
Coding Properties with Decorators--1224
Descriptors---1226
The Basics--1227
A First Example---1229
Computed Attributes---1231
Using State Information in Descriptors--1232
How Properties and Descriptors Relate--1236
__getattr__ and __getattribute__--1237
The Basics--1238
A First Example---1241
Computed Attributes---1243
__getattr__ and __getattribute__ Compared---1245
Management Techniques Compared--1246
Intercepting Built-in Operation Attributes---1249
Example: Attribute Validations---1256
Using Properties to Validate--1256
Using Descriptors to Validate---1259
Using __getattr__ to Validate--1263
Using __getattribute__ to Validate--1265
Chapter Summary--1266
Test Your Knowledge: Quiz---1266
Test Your Knowledge: Answers--1267
39 Decorators--1269
What’s a Decorator?---1269

Managing Calls and Instances--1270
Managing Functions and Classes--1270
Using and Defining Decorators---1271
Why Decorators?---1271
The Basics---1273
Function Decorators--1273
Class Decorators---1277
Decorator Nesting--1279
Decorator Arguments---1281
Decorators Manage Functions and Classes, Too---1282
Coding Function Decorators--1283
Tracing Calls--1283
Decorator State Retention Options--1285
Class Blunders I: Decorating Methods---1289
Timing Calls---1295
Adding Decorator Arguments---1298
Coding Class Decorators---1301
Singleton Classes---1301
Tracing Object Interfaces---1303
Class Blunders II: Retaining Multiple Instances---1308
Decorators Versus Manager Functions--1309
Why Decorators? (Revisited)--1310
Managing Functions and Classes Directly--1312
Example: “Private” and “Public” Attributes---1314
Implementing Private Attributes---1314
Implementation Details I--1317
Generalizing for Public Declarations, Too--1318
Implementation Details II---1320
Open Issues---1321
Python Isn’t About Control---1329
Example: Validating Function Arguments--1330
The Goal---1330
A Basic Range-Testing Decorator for Positional Arguments--------------------------------1331
Generalizing for Keywords and Defaults, Too--1333
Implementation Details--1336
Open Issues---1338
Decorator Arguments Versus Function Annotations---1340
Other Applications: Type Testing (If You Insist!)---1342
Chapter Summary--1343
Test Your Knowledge: Quiz---1344
Test Your Knowledge: Answers---1345
40 Metaclasses---1355
To Metaclass or Not to Metaclass---1356
Increasing Levels of “Magic”---1357
A Language of Hooks---1358
The Downside of “Helper” Functions---1359
Metaclasses Versus Class Decorators: Round 1--1361
The Metaclass Model---1364
Classes Are Instances of type---1364
Metaclasses Are Subclasses of Type--1366
Class Statement Protocol---1367
Declaring Metaclasses--1368

Declaration in 3.X--1369
Declaration in 2.X--1369
Metaclass Dispatch in Both 3.X and 2.X---1370
Coding Metaclasses---1370
A Basic Metaclass---1371
Customizing Construction and Initialization--1372
Other Metaclass Coding Techniques--1373
Inheritance and Instance---1378
Metaclass Versus Superclass---1381
Inheritance: The Full Story---1382
Metaclass Methods--1388
Metaclass Methods Versus Class Methods--1389
Operator Overloading in Metaclass Methods---1390
Example: Adding Methods to Classes---1391
Manual Augmentation---1391
Metaclass-Based Augmentation--1393
Metaclasses Versus Class Decorators: Round 2---1394
Example: Applying Decorators to Methods---1400
Tracing with Decoration Manually---1400
Tracing with Metaclasses and Decorators---1401
Applying Any Decorator to Methods---1403
Metaclasses Versus Class Decorators: Round 3 (and Last)-----------------------------------1404
Chapter Summary--1407
Test Your Knowledge: Quiz---1407
Test Your Knowledge: Answers--1408
41 All Good Things---1409
The Python Paradox---1409
On “Optional” Language Features--1410
Against Disquieting Improvements--1411
Complexity Versus Power--1412
Simplicity Versus Elitism---1412
Closing Thoughts---1413
Where to Go From Here---1414
Encore: Print Your Own Completion Certificate!--1414
Part IX Appendixes
A Installation and Configuration--1421
Installing the Python Interpreter--1421
Is Python Already Present?--1421
Where to Get Python---1422
Installation Steps--1423
Configuring Python--1427
Python Environment Variables--1427
How to Set Configuration Options--1429
Python Command-Line Arguments--1432
Python 3.3 Windows Launcher Command Lines--1435
For More Help--1436
B The Python 3.3 Windows Launcher--1437
The Unix Legacy--1437
The Windows Legacy---1438
Introducing the New Windows Launcher---1439
A Windows Launcher Tutorial--1441
Step 1: Using Version Directives in Files---1441

Step 2: Using Command-Line Version Switches---1444
Step 3: Using and Changing Defaults--1445
Pitfalls of the New Windows Launcher---1447
Pitfall 1: Unrecognized Unix !# Lines Fail--1447
Pitfall 2: The Launcher Defaults to 2.X--1448
Pitfall 3: The New PATH Extension Option---1449
Conclusions: A Net Win for Windows---1450
C Python Changes and This Book--1451
Major 2.X/3.X Differences--1451
3.X Differences--1452
3.X-Only Extensions---1453
General Remarks: 3.X Changes--1454
Changes in Libraries and Tools--1454
Migrating to 3.X--1455
Fifth Edition Python Changes: 2.7, 3.2, 3.3---1456
Changes in Python 2.7--1456
Changes in Python 3.3--1457
Changes in Python 3.2--1458
Fourth Edition Python Changes: 2.6, 3.0, 3.1---1458
Changes in Python 3.1--1458
Changes in Python 3.0 and 2.6---1459
Specific Language Removals in 3.0---1460
Third Edition Python Changes: 2.3, 2.4, 2.5--1462
Earlier and Later Python Changes--1463
D Solutions to End-of-Part Exercises---1465
Part I, Getting Started---1465
Part II, Types and Operations--1467
Part III, Statements and Syntax--1473
Part IV, Functions and Generators---1475
Part V, Modules and Packages--1485
Part VI, Classes and OOP---1489
Part VII, Exceptions and Tools--1497
Index---1507

***LearnPythontheHardWay3rdEdition_2014_306p

Preface---1
Acknowledgments---1
The Hard Way Is Easier--1
Reading and Writing--2
Attention to Detail--2
Spotting Differences---2
Do Not Copy-Paste--2
A Note on Practice and Persistence---3
A Warning for the Smarties--3
Exercise 0 The Setup---6
Mac OSX---6
OSX: What You Should See---7
Windows--7
Windows: What You Should See---8
Linux--9

Linux: What You Should See--10
Warnings for Beginners---10
Exercise 1 A Good First Program---12
What You Should See--14
Study Drills--15
Common Student Questions---16
Exercise 2 Comments and Pound Characters---18
What You Should See--18
Study Drills--18
Common Student Questions---19
Exercise 3 Numbers and Math---20
What You Should See--21
Study Drills--21
Common Student Questions---22
vi CONTENTS
Exercise 4 Variables and Names---24
What You Should See--25
Study Drills--25
Common Student Questions---25
Exercise 5 More Variables and Printing--28
What You Should See--28
Study Drills--29
Common Student Questions---29
Exercise 6 Strings and Text---30
What You Should See--31
Study Drills--31
Common Student Questions---31
Exercise 7 More Printing 32
What You Should See--32
Study Drills--32
Common Student Questions---33
Exercise 8 Printing, Printing---34
What You Should See--34
Study Drills--34
Common Student Questions---34
Exercise 9 Printing, Printing, Printing---36
What You Should See--36
Study Drills--36
Common Student Questions---37
Exercise 10 What Was That?---38
What You Should See--39
Escape Sequences--39
Study Drills--40
Common Student Questions---40
Exercise 11 Asking Questions--42
What You Should See--42
Study Drills--43
Common Student Questions---43
CONTENTS--vii
Exercise 12 Prompting People 44
What You Should See--44
Study Drills--44

Common Student Questions--45
Exercise 13 Parameters, Unpacking, Variables 46
Hold Up! Features Have Another Name---46
What You Should See--47
Study Drills--48
Common Student Questions--48
Exercise 14 Prompting and Passing 50
What You Should See--50
Study Drills--51
Common Student Questions--51
Exercise 15 Reading Files 54
What You Should See--55
Study Drills--55
Common Student Questions--56
Exercise 16 Reading and Writing Files 58
What You Should See--59
Study Drills--59
Common Student Questions--60
Exercise 17 More Files 62
What You Should See--63
Study Drills--63
Common Student Questions--63
Exercise 18 Names, Variables, Code, Functions 66
What You Should See--67
Study Drills--68
Common Student Questions--68
Exercise 19 Functions and Variables 70
What You Should See--71
Study Drills--71
Common Student Questions--71
viii CONTENTS
Exercise 20 Functions and Files--74
What You Should See--75
Study Drills--75
Common Student Questions--75
Exercise 21 Functions Can Return Something---78
What You Should See--79
Study Drills--79
Common Student Questions--80
Exercise 22 What Do You Know So Far?--81
What You Are Learning--81
Exercise 23 Read Some Code---82
Exercise 24 More Practice---84
What You Should See--85
Study Drills--85
Common Student Questions--85
Exercise 25 Even More Practice--86
What You Should See--87
Study Drills--88
Common Student Questions--89
Exercise 26 Congratulations, Take a Test!---90
Common Student Questions--90

Exercise 27 Memorizing Logic---92
The Truth Terms--92
The Truth Tables---93
Common Student Questions---94
Exercise 28 Boolean Practice---96
What You Should See---98
Study Drills---98
Common Student Questions---98
Exercise 29 What If---100
What You Should See---100
Study Drills--101
Common Student Questions---101
CONTENTS---ix
Exercise 30 Else and If 102
What You Should See---103
Study Drills--103
Common Student Questions---103
Exercise 31 Making Decisions 104
What You Should See---105
Study Drills--105
Common Student Questions---105
Exercise 32 Loops and Lists 106
What You Should See---107
Study Drills--108
Common Student Questions---108
Exercise 33 While-Loops 110
What You Should See---111
Study Drills--111
Common Student Questions---112
Exercise 34 Accessing Elements of Lists 114
Study Drills--115
Exercise 35 Branches and Functions 116
What You Should See---117
Study Drills--118
Common Student Questions---118
Exercise 36 Designing and Debugging 120
Rules for If-Statements--120
Rules for Loops--120
Tips for Debugging---121
Homework 121
Exercise 37 Symbol Review 122
Keywords--122
Data Types---123
String Escape Sequences 124
String Formats--124
Operators--125
x CONTENTS
Reading Code---126
Study Drills--127
Common Student Questions---127
Exercise 38 Doing Things to Lists---128
What You Should See---129

Study Drills--130
Common Student Questions--130
Exercise 39 Dictionaries, Oh Lovely Dictionaries--132
What You Should See--134
Study Drills--135
Common Student Questions--135
Exercise 40 Modules, Classes, and Objects--138
Modules Are Like Dictionaries---138
Classes Are Like Modules---139
Objects Are Like Mini-Imports--140
Getting Things from Things---141
A First-Class Example---141
What You Should See--142
Study Drills--142
Common Student Questions--143
Exercise 41 Learning to Speak Object Oriented---144
Word Drills--144
Phrase Drills--144
Combined Drills 145
A Reading Test--145
Practice English to Code--147
Reading More Code--148
Common Student Questions--148
Exercise 42 Is-A, Has-A, Objects, and Classes--150
How This Looks in Code--151
About class Name(object) 153
Study Drills--153
Common Student Questions--154
CONTENTS--xi
Exercise 43 Basic Object-Oriented Analysis and Design 156
The Analysis of a Simple Game Engine--157
Write or Draw about the Problem--157
Extract Key Concepts and Research Them---158
Create a Class Hierarchy and Object Map for the Concepts---158
Code the Classes and a Test to Run Them---159
Repeat and Refine--161
Top Down vs Bottom Up--161
The Code for “Gothons from Planet Percal #25”--162
What You Should See--167
Study Drills--168
Common Student Questions--168
Exercise 44 Inheritance vs Composition---170
What is Inheritance?--170
Implicit Inheritance--171
Override Explicitly--172
Alter Before or After---172
All Three Combined--174
The Reason for super()--175
Using super() with __init__--175
Composition--176
When to Use Inheritance or Composition---177
Study Drills--177

Common Student Questions--178
Exercise 45 You Make a Game 180
Evaluating Your Game---180
Function Style--181
Class Style--181
Code Style--182
Good Comments---182
Evaluate Your Game--183
Exercise 46 A Project Skeleton 184
Installing Python Packages--184
Creating the Skeleton Project Directory--185
xii CONTENTS
Final Directory Structure---186
Testing Your Setup--187
Using the Skeleton--188
Required Quiz--188
Common Student Questions--189
Exercise 47 Automated Testing--190
Writing a Test Case---190
Testing Guidelines 192
What You Should See--192
Study Drills---193
Common Student Questions--193
Exercise 48 Advanced User Input--194
Our Game Lexicon--194
Breaking Up a Sentence---195
Lexicon Tuples---195
Scanning Input---195
Exceptions and Numbers--196
What You Should Test--196
Design Hints--198
Study Drills---198
Common Student Questions--198
Exercise 49 Making Sentences---200
Match and Peek--200
The Sentence Grammar--201
A Word on Exceptions--203
What You Should Test--204
Study Drills---204
Common Student Questions--204
Exercise 50 Your First Website---206
Installing lpthw.web---206
Make a Simple “Hello World” Project---207
What’s Going On?---208
Fixing Errors--209
CONTENTS--xiii
Create Basic Templates--209
Study Drills---211
Common Student Questions--211
Exercise 51 Getting Input from a Browser 214
How the Web Works 214
How Forms Work--216

Creating HTML Forms--218
Creating a Layout Template--220
Writing Automated Tests for Forms---221
Study Drills---223
Common Student Questions---224
Exercise 52 The Start of Your Web Game 226
Refactoring the Exercise 43 Game---226
Sessions and Tracking Users--231
Creating an Engine---232
Your Final Exam--235
Common Student Questions---236
Next Steps 237
How to Learn Any Programming Language--238
Advice from an Old Programmer 241
Appendix Command Line Crash Course--243
Introduction: Shut Up and Shell--243
How to Use This Appendix--243
You Will Be Memorizing Things---244
Exercise 1: The Setup--245
Do This---245
You Learned This---246
Do More---246
Exercise 2: Paths, Folders, Directories (pwd)---248
Do This---248
You Learned This---249
Do More---249
Exercise 3: If You Get Lost--250

Do This---250
You Learned This---250
Exercise 4: Make a Directory (mkdir)--250
Do This---250
You Learned This---252
Do More---252
Exercise 5: Change Directory (cd) 252
Do This---252
You Learned This---255
Do More---255
Exercise 6: List Directory (ls)--256
Do This---256
You Learned This---259
Do More---260
Exercise 7: Remove Directory (rmdir) 260
Do This---260
You Learned This---262
Do More---262
Exercise 8: Move Around (pushd, popd)--262
Do This---263
You Learned This---264
Do More---265
Exercise 9: Make Empty Files (Touch, New-Item)--265
Do This---265

You Learned This--266
Do More--266
Exercise 10: Copy a File (cp)---266
Do This---266
You Learned This--268
Do More--269
Exercise 11: Move a File (mv)---269
Do This---269
You Learned This--271
Do More--271
CONTENTS---xv
Exercise 12: View a File (less, MORE)---271
Do This---271
You Learned This--272
Do More--272
Exercise 13: Stream a File (cat)---272
Do This---272
You Learned This--273
Do More--273
Exercise 14: Remove a File (rm)---273
Do This---273
You Learned This--275
Do More--275
Exercise 15: Exit Your Terminal (exit)--275
Do This---275
You Learned This--276
Do More--276
Command Line Next Steps---276
Unix Bash References--276
PowerShell References--277
Index--279

***MasteringPythonRegularExpressions-LopezFelixRomeroVictor_2014_110p

Preface---1
Chapter 1: Introducing Regular Expressions---5
History, relevance, and purpose---6
The regular expression syntax---8
Literals---9
Character classes--11
Predefined character classes--12
Alternation--14
Quantifiers--16
Greedy and reluctant quantifiers--19
Boundary Matchers---20
Summary--23
Chapter 2: Regular Expressions with Python--25
A brief introduction--25
Backslash in string literals---27

String Python 2.x--27
Building blocks for Python regex--28
RegexObject--28
Searching--30
Modifying a string--35
MatchObject--39
group([group1, …])--39
groups([default])---40
groupdict([default])--41
start([group])--41
end([group])---42
span([group])--42
expand(template)---42
Module operations--42
escape()---43
purge()--43
Compilation flags---43
re.IGNORECASE or re.I--45
re.MULTILINE or re.M---45
re.DOTALL or re.S--45
re.LOCALE or re.L--46
re.UNICODE or re.U--46
re.VERBOSE or re.X--47
re.DEBUG--47
Python and regex special considerations--47
Differences between Python and other flavors---47
Unicode--48
What's new in Python 3---49
Summary--51
Chapter 3: Grouping---53
Introduction---53
Backreferences---56
Named groups--57
Non-capturing groups---58
Atomic groups--59
Special cases with groups--60
Flags per group---60
yes-pattern|no-pattern---61
Overlapping groups---62
Summary--64
Chapter 4: Look Around--65
Look ahead--66
Negative look ahead---68
Look around and substitutions--69
Look behind---71
Negative look behind--74
Look around and groups--75
Summary--76
Chapter 5: Performance of Regular Expressions--77
Benchmarking regular expressions with Python--78
The RegexBuddy tool---80
Understanding the Python regex engine---81

Backtracking--82
Optimization recommendations--84
Reuse compiled patterns---85
Extract common parts in alternation--86
Shortcut to alternation---87
Use non-capturing groups when appropriate--88
Be specific--88
Don't be greedy--88
Summary---89
Index---91

***MongoDB_and_Python_2011_66p

1 Getting Started---1
Introduction---1
Finding Reference Documentation---2
Installing MongoDB---3
Running MongoDB--5
Setting up a Python Environment with MongoDB--6
2 Reading and Writing to MongoDB with Python--9
Connecting to MongoDB with Python---10
Getting a Database Handle--11
Inserting a Document into a Collection--12
Write to a Collection Safely and Synchronously--13
Guaranteeing Writes to Multiple Database Nodes---14
Introduction to MongoDB Query Language---15
Reading, Counting, and Sorting Documents in a Collection------------------------------15
Updating Documents in a Collection---18
Deleting Documents from a Collection---20
MongoDB Query Operators--21
MongoDB Update Modifiers--22
3 Common MongoDB and Python Patterns---23
A Uniquely Document-Oriented Pattern: Embedding--23
Fast Lookups: Using Indexes with MongoDB---29
Location-based Apps with MongoDB: GeoSpatial Indexing---------------------------------33
Code Defensively to Avoid KeyErrors and Other Bugs--37
Update-or-Insert: Upserts in MongoDB--39
Atomic Read-Write-Modify: MongoDB’s findAndModify------------------------------------40
Fast Accounting Pattern---41
4 MongoDB with Web Frameworks--45
Pylons 1.x and MongoDB---45
Pyramid and MongoDB---49
Django and MongoDB---51
Going Further--53

***MySQL_for_Python_2010_440p

Preface--1
Chapter 1: Getting Up and Running with MySQL for Python-----------------------------------7

Getting MySQL for Python--7
Using a package manager (only on Linux)--8
Using RPMs and yum---9
Using RPMs and urpm--9
Using apt tools on Debian-like systems--9
Using an installer for Windows---10
Using an egg file---10
Using a tarball (tar.gz file)--14
Importing MySQL for Python--17
Accessing online help when you need it---18
MySQLdb--18
_mysql---19
Connecting with a database--20
Creating a connection object--20
Creating a cursor object---22
Interacting with the database---22
Closing the connection---23
Multiple database connections--23
Summary---24
Chapter 2: Simple Querying--25
A brief introduction to CRUD--25
Forming a query in MySQL--26
SELECT--27
* (asterisk)---27
FROM---28
staff---28
; (semicolon)--29
Other helpful quantifiers---29
WHERE---30
GROUP BY---30
HAVING---32
ORDER BY--33
LIMIT--35
INTO OUTFILE---37
Passing a query to MySQL---37
A simple SELECT statement--38
Modifying the results--39
Using user-defined variables--40
Determining characteristics of a database and its tables--41
Determining what tables exist---42
Assigning each table a number--43
Offering the options to the user--43
Allowing the user to detail a search query---44
Changing queries dynamically---45
Pattern matching in MySQL queries---45
Putting it into practice---46
Project: A command-line search utility--48
Preparing a database for searching--49
Planning your work, then working your plan--50
Develop a well-abstracted search functionality--50
Specifying the search term from the command-line---52
Implementing and incorporating the other functions: -t, -f, and -o----------------------------------55

Including an option for an output file--57
Room to grow---57
Summary--58
Chapter 3: Simple Insertion---59
Forming a MySQL insertion statement--60
INSERT--60
INTO--61
Table name--61
Column names--61
VALUES---63
<some values>--64
; (semicolon)---66
Helpful ways to nuance an INSERT statement---66
INSERT SELECT---66
INSERT DELAYED…--70
INSERT ON DUPLICATE KEY UPDATE--71
Passing an insertion through MySQL for Python---72
Setting up the preliminaries---72
A simple INSERT statement---73
More complex INSERT commands---75
Using user-defined variables---75
Using metadata--77
Querying the database for its structure---78
Retrieving the table structure---80
Changing insertion values dynamically---82
Validating the value of name---83
Validating the value of price--83
Querying the user for a correction--84
Passing fish and price for validation--84
Essentials: close and commit---85
In need of some closure---85
What happened to commit?---85
Why are these essentials non-essential?---85
Project: A command-line insertion utility--86
The necessary modules--86
The main() thing---87
Coding the flag system---88
Testing the values passed by the user--88
Try to establish a database connection---89
Showing the tables---90
Showing the table structure, if desired---90
Accepting user input for the INSERT statement--91
Building the INSERT statement from the user input and executing it---92
Committing changes and closing the connection--93
Coding the other functions--93
valid_digit() and valid_string()---93
valid_table()--94
query()---94
Calling main()--95
Room to grow--99
Summary---100
Chapter 4: Exception Handling---101

Why errors and warnings are good for you--101
Errors versus warnings: There's a big difference---104
The two main errors in MySQLdb---104
DatabaseError---105
InterfaceError--105
Warnings in MySQL for Python---105
Handling exceptions passed from MySQL--105
Python exception-handling--105
Catching an exception from MySQLdb---106
Raising an error or a warning--107
Making exceptions less intimidating--108
Catching different types of exceptions---109
Types of errors--109
DataError---110
IntegrityError---110
InternalError--111
NotSupportedError---111
OperationalError---111
ProgrammingError---112
Customizing for catching--113
Catching one type of exception--113
Catching different exceptions--114
Combined catching of exceptions--115
Raising different exceptions--115
Creating a feedback loop---116
Project: Bad apples---117
The preamble---118
Making the connection---119
Sending error messages--119
The statement class--121
The main() thing---125
Try, try again---126
If all else fails--126
Room to grow--127
Summary---128
Chapter 5: Results Record-by-Record---129
The problem--129
Why?--131
Computing resources---131
Local resources---132
Web applications---133
Network latency--134
Server-client communications---134
Apparent responsiveness---134
Pareto's Principle---134
How?--135
The fetchone() method---135
The fetchmany() method---136
Iteration: What is it?---137
Generating loops---138
while if loops---138
The for loop--139

Iterators--140
Illustrative iteration--141
Iteration and MySQL for Python---141
Generators---142
Using fetchone() in a generator--142
Using fetchmany() in a generator--143
Project: A movie database---144
Getting Sakila--145
Creating the Sakila database--145
The structure of Sakila---146
Planning it out--148
The SQL statements to be used--148
Returning the films of an actor--148
Returning the actors of a film--149
Accepting user data---150
A MySQL query with class--150
The __init__ method: The consciousness of the class---151
Setting the query's type---151
Creating the cursor---152
Forming the query---153
Executing the query--154
Formatting the results--155
Formatting a sample---155
Formatting a larger set of results--156
The main() thing--157
Calling main()---158
Running it---159
Room to grow--159
Summary--160
Chapter 6: Inserting Multiple Entries---161
The problem--161
Why not a MySQL script?---162
Lack of automation---162
Debugging the process---162
Why not iterate?---163
A test sample: Generating primes--163
Comparing execution speeds--166
Introducing the executemany() method--166
executemany(): Basic syntax--167
executemany(): Multiple INSERT statements--168
executemany(): Multiple SELECT statements--170
executemany(): Behind the scenes---170
MySQL server has gone away--173
Command-line option configuration--173
Using a configuration file--174
More than 16 MB is often unnecessary--174
Project: Converting a CSV file to a MySQL table---175
The preamble--175
The options---176
Defining the connection---177
Creating convert---177
The main() function---178

Calling main()--181
Room to grow---181
Summary--182
Chapter 7: Creating and Dropping--183
Creating databases---183
Test first, create second---184
CREATE specifications---185
Specifying the default character set---185
Specifying the collation for a database--186
Declaring collation--186
Finding available character sets and collations---187
Removing or deleting databases--187
Avoiding errors--188
Preventing (illegal) access after a DROP---188
Creating tables--189
Covering our bases---190
Avoiding errors--191
Creating temporary tables---191
Dropping tables---192
Playing it safe---192
Avoiding errors--193
Removing user privileges---193
Doing it in Python--193
Creating databases with MySQLdb--194
Testing the output--194
Dynamically configuring the CREATE statement--195
Dropping databases with MySQLdb---195
Creating tables in Python---195
Verifying the creation of a table---196
Another way to verify table creation---197
Dropping tables with MySQLdb--198
Project: Web-based administration of MySQL---198
CGI vs PHP: What is the difference?--199
Basic CGI---200
Using PHP as a substitute for CGI---202
CGI versus PHP: When to use which?---203
Some general considerations for this program--203
Program flow--203
The basic menu--204
Authorization details---206
Three operational sections of the dialogue--206
The variables--206
Planning the functions---207
Code of each function--207
Connecting without a database---207
Connecting with a database---208
Database action---208
Table action--209
Query action---210
execute()---211
The HTML output---212
Basic definition--212

The message attribute--213
Defining header()---213
Defining footer()--213
Defining body()---214
Defining page()---214
Getting the data--214
Using CGI--214
Using PHP--215
Defining main()---217
Room to grow--218
Summary--218
Chapter 8: Creating Users and Granting Access---219
A word on security--219
Creating users in MySQL---220
Forcing the use of a password--221
Restricting the client's host---221
Creating users from Python---223
Removing users in MySQL---224
DROPping users in Python---225
GRANT access in MySQL---225
Important dynamics of GRANTing access--226
The GRANT statement in MySQL--226
Using REQUIREments of access---229
Using a WITH clause---230
Granting access in Python---231
Removing privileges in MySQL--233
Basic syntax--233
After using REVOKE, the user still has access!?--233
Using REVOKE in Python--235
Project: Web-based user administration--236
New options in the code---236
Adding the functions: CREATE and DROP--239
Adding CREATE and DROP to main()---240
Adding the functions: GRANT and REVOKE---241
Adding GRANT and REVOKE to main()--241
Test the program--243
New options on the page--244
Room to grow--244
Summary--245
Chapter 9: Date and Time Values--247
Date and time data types in MySQL--247
DATETIME--248
Output format---248
Input formats---248
Input range---249
Using DATETIME in a CREATE statement--249
DATE--249
Output and Input formats--249
Input range---250
TIMESTAMP---250
Input of values---250
Range--251

Defaults, initialization, and updating--251
YEAR---252
Two-digit YEAR values---252
Four-digit YEAR values--252
Valid input---253
TIME--253
Format---254
Invalid values--255
Date and time types in Python---256
Date and time functions--257
NOW()---260
CURDATE()--260
CURTIME()---261
DATE()---261
DATE_SUB() and DATE_ADD()---262
DATEDIFF()--266
DATE_FORMAT()---267
EXTRACT()---269
TIME()--270
Project: Logging user activity--270
The log framework---272
The logger() function---273
Creating the database---273
Using the database--274
Creating the table---274
Forming the INSERT statement--274
Ensure logging occurs--275
Room to grow---276
Summary--277
Chapter 10: Aggregate Functions and Clauses--279
Calculations in MySQL--280
COUNT()--281
SUM()--282
MAX()--283
MIN()---284
AVG()--284
The different kinds of average--285
Trimming results--287
DISTINCT---287
GROUP_CONCAT()---289
Specifying the delimiter---290
Customizing the maximum length---290
Using GROUP_CONCAT() with DISTINCT---291
Server-side sorting in MySQL---292
GROUP BY---293
ORDER BY---294
Using a universal quantifier--294
Sorting alphabetically or from low-to-high--295
Reversing the alphabet or sorting high-to-low--296
Sorting with multiple keys--298
Putting it in Python---298
Project: Incorporating aggregate functions--300

Adding to qaction()--300
New variables---301
New statement formation--302
Revising main()---305
Setting up the options---308
Changing the HTML form---309
Summary--310
Chapter 11: SELECT Alternatives---311
HAVING clause---312
WHERE versus HAVING: Syntax---312
WHERE versus HAVING: Aggregate functions---312
WHERE versus HAVING: Application--314
Subqueries---317
Unions---319
Joins--321
LEFT and RIGHT joins---321
OUTER joins---323
INNER joins--324
NATURAL joins--326
CROSS joins--327
Doing it in Python---327
Subqueries---328
Unions---329
Joins--329
Project: Implement HAVING---330
Revising the Python backend--331
Revising qaction()--331
Revising main()---333
Revising the options--336
Revising the HTML interface--337
Room to grow---338
Summary--339
Chapter 12: String Functions---341
Preparing results before their return--341
CONCAT() function---342
SUBSTRING() or MID()--343
TRIM()---344
Basic syntax---344
Options---345
Alternatives---346
REPLACE()--347
INSERT()---348
REGEXP---350
Accessing and using index data---354
LENGTH()---354
INSTR() or LOCATE()--355
INSTR()--356
LOCATE()--356
Nuancing data---357
ROUND()---357
FORMAT()---359
UPPER()---360

LOWER()--360
Project: Creating your own functions--360
Hello()--361
Capitalise()---362
DELIMITER---362
The function definition---362
Calling the function--364
Defining the function in Python---365
Defining the function as a Python value--365
Sourcing the MySQL function as a Python module--366
Sourcing the function as MySQL code--366
Room to grow--367
Summary---367
Chapter 13: Showing MySQL Metadata---369
MySQL's system environment--370
ENGINE---371
The most popular engines--372
Transactions--372
Specifying the engine--373
ENGINE status--373
SHOW ENGINES--374
Profiling--375
SHOW PROFILE--375
SHOW PROFILES---376
SHOW system variables---376
Accessing database metadata--377
DATABASES---377
Using the USE command--378
Accessing metadata about tables--378
SHOW TABLES--378
SHOW TABLE STATUS---379
Showing columns from a table---379
FUNCTION STATUS--380
Accessing user metadata--383
SHOW GRANTS---383
PRIVILEGES---384
Project: Building a database class---384
Writing the class---384
Defining fetchquery() and some core methods--385
Retrieving table status and structure--386
Retrieving the CREATE statements--386
Define main()—part 1--387
Writing resproc()---388
Define main()—part 2--389
The preamble---389
Modules and variables---390
Login and USE--390
Closing out the program--390
Room to grow--391
Summary---391
Chapter 14: Disaster Recovery---393
Every database needs a backup plan---394

Offline backups--394
Live backups---395
Choosing a backup method---395
Copying the table files--396
Locking and flushing--397
Unlocking the tables--398
Restoring the data---398
Delimited backups within MySQL--398
Using SELECT INTO OUTFILE to export data---398
Using LOAD DATA INFILE to import data--399
Archiving from the command line--400
mysqldump---400
mysqlhotcopy---403
Backing up a database with Python---405
Summary--406
Index--407

***openpyxl_2015_133p

1 Introduction--3
1.1 Sample code:---3
2 User List 5
3 How to Contribute Code--7
4 Other ways to help---9
5 Installation---11
6 Getting the source---13
7 Usage examples--15
7.1 Tutorial--15
7.2 Cookbook---19
7.3 Charts--23
7.4 Comments---23
7.5 Read/write large files---24
7.6 Working with styles--25
7.7 Conditional Formatting--29
7.8 Data Validation---30
8 Information for Developers 33
8.1 Development--33
8.2 Testing on Windows---35
9 API Documentation 39
9.1 openpyxl package---39
10 Indices and tables 95
11 Release Notes 97
11.1 2.2.4 (2015-06-17)---97
11.2 2.2.3 (2015-05-26)---97
11.3 2.2.2 (2015-04-28)---97
11.4 2.2.1 (2015-03-31)---98
11.5 2.2.0 (2015-03-11)---98
11.6 2.2.0-b1 (2015-02-18)---98
11.7 2.1.5 (2015-02-18) 99
11.8 2.1.4 (2014-12-16)---99
11.9 2.1.3 (2014-12-09)--100

11.10 2.1.2 (2014-10-23)--100
11.11 2.1.1 (2014-10-08)--100
11.12 2.1.0 (2014-09-21)--101
11.13 2.0.5 (2014-08-08)--102
11.14 2.0.4 (2014-06-25)--102
11.15 2.0.3 (2014-05-22)--102
11.16 2.0.2 (2014-05-13)--102
11.17 2.0.1 (2014-05-13) brown bag--102
11.18 2.0.0 (2014-05-13) brown bag--102
11.19 1.8.6 (2014-05-05)--104
11.20 1.8.5 (2014-03-25)--104
11.21 1.8.4 (2014-02-25)--104
11.22 1.8.3 (2014-02-09)--104
11.23 1.8.2 (2014-01-17)--105
11.24 1.8.1 (2014-01-14)--105
11.25 1.8.0 (2014-01-08)--105
11.26 1.7.0 (2013-10-31)--106

***Parallel_Programming_with_Python_2014_122p

Preface---1
Chapter 1: Contextualizing Parallel, Concurrent,
and Distributed Programming--7
Why use parallel programming?--9
Exploring common forms of parallelization--9
Communicating in parallel programming--11
Understanding shared state--12
Understanding message passing--12
Identifying parallel programming problems---13
Deadlock--13
Starvation---13
Race conditions--14
Discovering Python's parallel programming tools--15
The Python threading module---15
The Python multiprocessing module---15
The parallel Python module--16
Celery – a distributed task queue---16
Taking care of Python GIL---16
Summary---17
Chapter 2: Designing Parallel Algorithms--19
The divide and conquer technique--19
Using data decomposition---20
Decomposing tasks with pipeline---21
Processing and mapping---22
Identifying independent tasks--22
Identifying the tasks that require data exchange---22
Load balance--23
Summary---23
Chapter 3: Identifying a Parallelizable Problem---25
Obtaining the highest Fibonacci value for multiple inputs--25

Crawling the Web--27
Summary---28
Chapter 4: Using the threading and concurrent.futures Modules------------------------------------29
Defining threads---29
Advantages and disadvantages of using threads--30
Understanding different kinds of threads--30
Defining the states of a thread---31
Choosing between threading and _thread---32
Using threading to obtain the Fibonacci series term with
multiple inputs---32
Crawling the Web using the concurrent.futures module--36
Summary---40
Chapter 5: Using Multiprocessing and ProcessPoolExecutor---------------------------------------41
Understanding the concept of a process---41
Understanding the process model--42
Defining the states of a process---42
Implementing multiprocessing communication--42
Using multiprocessing.Pipe--43
Understanding multiprocessing.Queue---45
Using multiprocessing to compute Fibonacci series terms
with multiple inputs--45
Crawling the Web using ProcessPoolExecutor---48
Summary---51
Chapter 6: Utilizing Parallel Python--53
Understanding interprocess communication--53
Exploring named pipes---54
Using named pipes with Python---54
Writing in a named pipe--55
Reading named pipes---56
Discovering PP--57
Using PP to calculate the Fibonacci series term on SMP architecture--------------------------------59
Using PP to make a distributed Web crawler---61
Summary---66
Chapter 7: Distributing Tasks with Celery--67
Understanding Celery---67
Why use Celery?--68
Understanding Celery's architecture--68
Working with tasks--69
Discovering message transport (broker)--70
Understanding workers---70
Understanding result backends---71
Setting up the environment---71
Setting up the client machine--71
Setting up the server machine---73
Dispatching a simple task--73
Using Celery to obtain a Fibonacci series term--76
Defining queues by task types---79
Using Celery to make a distributed Web crawler--81
Summary---84
Chapter 8: Doing Things Asynchronously--85
Understanding blocking, nonblocking, and asynchronous operations-------------------------------85
Understanding blocking operations--86

Understanding nonblocking operations--86
Understanding asynchronous operations--86
Understanding event loop---87
Polling functions---87
Using event loops--89
Using asyncio---89
Understanding coroutines and futures---90
Using coroutine and asyncio.Future--90
Using asyncio.Task---92
Using an incompatible library with asyncio---93
Summary--96
Index--99

***Pro_Android_Python_with_SL4A_2011_296p

About the Author--xi
About the Technical Reviewer---xii
Acknowledgments---xiii
Preface--xiv
Chapter 1: Introduction--1
Chapter 2: Getting Started--27
Chapter 3: Navigating the Android SDK--57
Chapter 4: Developing with Eclipse---83
Chapter 5: Exploring the Android API--113
Chapter 6: Background Scripting with Python--139
Chapter 7: Python Scripting Utilities---165
Chapter 8: Python Dialog Box–based GUIs--195
Chapter 9: Python GUIs with HTML--221
Chapter 10: Packaging and Distributing--249
Index---273

***Programming_Python_MLutz_4th_ed_2011_1628p

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 Shelve Interface---37
Step 5: Adding a GUI--40
GUI Basics--40
Using OOP for GUIs---42
Getting Input from a User---44
A GUI Shelve 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 Shelve 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
Starting Independent Programs--254

And Much More----------256
Why multiprocessing? The Conclusion----------257
Other Ways to Start Programs----------258
The os.spawn Calls----------258
The os.startfile 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
Running the Test Driver----------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 pymail 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 _pickle---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

***Python_3_Web_Development_2011_336

Preface---1
Chapter 1: Choosing Your Tools---7
Identifying the components of a web application---7
Time for action – getting an overview of a web application---8
Choosing suitable tools--10
Time for action – choosing a delivery framework, also known as web server---------------------11
Time for action – choosing a server-side scripting language--12
Time for action – choosing a database engine--14
Time for action – deciding on object relational mappers---15
Time for action – choosing a presentation framework---17
Designing for maintainability and usability---18
Testing--18
Time for action – choosing a test framework---19
Version management---19
Usability--20
Good looking – adhering to common GUI paradigms--20
Themable---21
Cross-browser compatible---21
Cross-platform compatible--22
Maintainability--22
Standards compliant--22
Security---23
Reliable---23
Robust--23
Access control and authentication--24
Confidentiality--24

Integrity--25
A final word on security--25
Help, I am confused!--25
Time for action – maintaining overview--26
Summary---28
Chapter 2: Creating a Simple Spreadsheet--29
Python 3---30
Time for action – installing Python 3 CherryPy---30
Time for action – installing CherryPy--31
Installing jQuery and jQuery UI--31
Serving an application---32
Time for action – serving a dummy application---33
Time for action – serving HTML as dynamic content---34
Who serves what: an overview--36
HTML: separating form and content--37
Time for action – a unit convertor---38
HTML: form-based interaction--39
JavaScript: using jQuery UI widgets--40
Time for action – conversion using unitconverter.js--40
jQuery selectors---42
CSS: applying a jQuery UI theme to other elements--43
Time for action – converting a unit convertor into a plugin--45
JavaScript: creating a jQuery UI plugin---46
Designing a spreadsheet application---51
Time for action – serving a spreadsheet application--51
HTML: keeping it simple--52
JavaScript: creating a spreadsheet plugin---52
The missing parts---58
Summary---58
Chapter 3: Tasklist I: Persistence--59
Designing a tasklist application--59
Time for action – creating a logon screen--62
Serving a logon screen---69
Setting up a session--70
Expiring a session--71
Designing a task list--72
Time for action – running tasklist.py--72
Python: the task module---75
Time for action – implementing the task module--76
Adding new tasks---80
Deleting a task--81
JavaScript: tasklist.js---83
Time for action – styling the buttons--83
JavaScript: tooltip.js--85
Time for action – implementing inline labels--86
CSS: tasklist.css--87
Summary---90
Chapter 4: Tasklist II: Databases and AJAX---91
The advantages of a database compared to a filesystem---92
Choosing a database engine---92
Database-driven authentication---93
Time for action – authentication using a database---94

Tasklist II – storing tasks in a database--99
Improving interactivity with AJAX---99
Time for action – getting the time with AJAX---100
Redesigning the Tasklist application--102
Database design---103
Time for action – creating the task database--103
Time for action – retrieving information with select statements--------------------------------------105
TaskDB – interfacing with the database--106
Time for action – connecting to the database--106
Time for action – storing and retrieving information--107
Time for action – updating and deleting information--109
Testing--111
Time for action – testing factorial.py--112
Now what have we gained?--113
Time for action – writing unit tests for tasklistdb.py---114
Designing for AJAX--116
Click handlers--120
The application---121
Time for action – putting it all together---123
Have a go hero – refreshing the itemlist on a regular basis---125
Summary---126
Chapter 5: Entities and Relations---127
Designing a book database--127
The Entity class---128
Time for action – using the Entity class---129
Time for action – creating instances--132
The Relation class--138
Time for action – using the Relation class--138
Relation instances--141
Time for action – defining the Books database---144
The delivery layer---150
Time for action – designing the delivery layer--151
Time for action – adding a new book--162
Auto completion---165
Time for action – using input fields with auto completion--166
The presentation layer--168
Time for action – using an enhanced presentation layer--168
Summary---170
Chapter 6: Building a Wiki---171
The data layer---172
Time for action – designing the wiki data model--172
The delivery layer--175
Time for action – implementing the opening screen--176
The structural components---177
The application methods---179
Time for action – implementing a wiki topic screen--180
Time for action – editing wiki topics---182
Additional functionality---185
Time for action – selecting an image---185
Time for action – implementing a tag cloud--190
Time for action – searching for words--192
The importance of input validation--195

Time for action – scrubbing your content--196
Time for action – rendering content--200
Summary--201
Chapter 7: Refactoring Code for Reuse--203
Time for action – taking a critical look---203
Refactoring--205
Time for action – defining new entities: how it should look---205
Metaclasses--206
Time for action – using metaclasses--207
MetaEntity and AbstractEntity classes--208
Time for action – implementing the MetaEntity and AbstractEntity classes-----------------------209
Relations--217
Time for action – defining new relations: how it should look--217
Implementing the MetaRelation and AbstractRelation classes--219
Adding new methods to existing classes--222
Browsing lists of entities---224
Time for action – using a table-based Entity browser---224
Time for action – examining the HTML markup---229
Caching--232
The books application revisited--236
Time for action – creating a books application, take two--236
Summary--242
Chapter 8: Managing Customer Relations---243
A critical review--243
Designing a Customer Relationship Management application---------------------------------------244
Time for action – implementing a basic CRM---244
Adding and editing values---248
Time for action – adding an instance--249
Time for action – editing an instance--251
Adding relations---257
Picklists---259
Time for action – implementing picklists--259
Summary--262
Chapter 9: Creating Full-Fledged Webapps: Implementing Instances---------------------------------263
Even more relations--263
Time for action – showing one-to-many relationships---264
Time for action – adapting MetaRelation--266
Time for action – enhancing Display--270
Time for action – enhancing Browse---271
Access control--274
Time for action – implementing access control---275
Role-based access control---278
Time for action – implementing role-based access control--279
Summary--283
Chapter 10: Customizing the CRM Application--285
Time for action – sorting---285
Time for action – filtering---290
Customization--292
Time for action – customizing entity displays--292
Time for action – customizing entity lists--298
Time for action – adding a delete button---301
Summary--302

Appendix A: References to Resources---303
Good old offline reference books---303
Additional websites, wikis, and blogs---304
Appendix B: Pop Quiz Answers---307
Chapter 2, Creating a Simple Spreadsheet--307
Chapter 3, Tasklist I: Persistence---308
Chapter 4, Tasklist II: Databases and AJAX---309
Chapter 5, Entities and Relations---310
Chapter 6, Building a Wiki--310
Index---311

***Python_and_AWS_Cookbook_2012_74p

Preface v
1 General Info---1
1.1 A Quick Note About Python--1
1.2 Installing boto---1
1.3 Getting Started with Amazon Web Services--4
1.4 Using boto with Eucalyptus--7
1.5 Using boto with Google Cloud Storage--8
1.6 Finding Available Regions for AWS--9
1.7 Enabling Debug Output with boto---10
1.8 Controlling Socket Timeouts in boto---10
2 EC2 Recipes---11
2.1 Launching an Instance---11
2.2 Keeping Track of Instances with Tags---15
2.3 Accessing the Console Log--17
2.4 Uploading Your Own SSH Keypair---17
2.5 Synchronizing SSH Keypairs Across EC2 Regions--18
2.6 Associate an Elastic IP Address with an Instance--19
2.7 Attach a Persistent EBS Volume to an Instance---20
2.8 Back Up Your EBS Volumes--22
2.9 Restore a Volume from a Snapshot---23
2.10 Clone an Existing Instance--24
2.11 Find All Running EC2 Instances--26
2.12 Monitoring the Performance of Your Instance--27
2.13 Getting Notifications--31
2.14 Storing Custom Data in CloudWatch--36
2.15 Executing Custom Scripts upon Instance Startup---37
3 S3 Recipes--47
3.1 Create a Bucket--47
3.2 Create a Bucket in a Specific Location--48
3.3 Store Private Data---49
3.4 Store Metadata with an Object--50
3.5 Computing Total Storage Used by a Bucket---52
3.6 Copy an Existing Object to Another Bucket---53
3.7 Modify the Metadata of an Existing Object--54
3.8 Find Out Who Is Accessing Your Data---54
3.9 Reduce the Cost of Storing Noncritical Data--56
3.10 Generating Expiring URLs for S3 Objects---57
3.11 Preventing Accidental Deletion of Data from S3--58

3.12 Hosting Static Websites on S3--60
3.13 Uploading Large Objects to S3--61

***Python_Cookbook_3rd_Edition_2013_706p

Preface xi
1 Data Structures and 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 Mapping Keys to Multiple Values in a Dictionary--11
1.7 Keeping Dictionaries in Order---12
1.8 Calculating with Dictionaries--13
1.9 Finding Commonalities in Two Dictionaries---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 Sorting a List of Dictionaries by a Common Key---21
1.14 Sorting 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 Combining Multiple Mappings into 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 a 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 Searching and Replacing Case-Insensitive Text---46
2.7 Specifying a Regular Expression for the Shortest Match---47
2.8 Writing a Regular Expression for Multiline Patterns--48
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 Interpolating Variables in Strings---61
2.16 Reformatting Text to a 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 Performing Accurate Decimal Calculations--84

3.3 Formatting Numbers for Output--87
3.4 Working with Binary, Octal, and Hexadecimal Integers--89
3.5 Packing and Unpacking Large Integers from Bytes--90
3.6 Performing Complex-Valued Math---92
3.7 Working with Infinity and NaNs--94
3.8 Calculating with Fractions---96
3.9 Calculating with Large Numerical Arrays---97
3.10 Performing Matrix and Linear Algebra Calculations--100
3.11 Picking Things at Random--102
3.12 Converting Days to Seconds, and Other Basic Time Conversions----------------------------104
3.13 Determining Last Friday's Date---106
3.14 Finding the Date Range for the Current Month---107
3.15 Converting Strings into Datetimes--109
3.16 Manipulating Dates Involving Time Zones--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 Implementing the Iterator Protocol--117
4.5 Iterating in Reverse---119
4.6 Defining 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 Iterating Over All Possible Combinations or Permutations--------------------------------------125
4.10 Iterating Over the Index-Value Pairs of a Sequence--127
4.11 Iterating Over Multiple Sequences Simultaneously---129
4.12 Iterating on Items in Separate Containers---131
4.13 Creating Data Processing Pipelines---132
4.14 Flattening a Nested Sequence---135
4.15 Iterating in Sorted Order Over Merged Sorted Iterables--136
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 Datafiles---149
5.8 Iterating Over Fixed-Sized Records--151
5.9 Reading Binary Data into a Mutable Buffer---152
5.10 Memory Mapping Binary 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 Parsing Huge XML Files Incrementally--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 Writing Functions That Accept Any Number of Arguments-------------------------------217
7.2 Writing Functions That Only Accept Keyword Arguments--------------------------------219
7.3 Attaching Informational Metadata to Function Arguments--------------------------------220
7.4 Returning Multiple Values from a Function---221
7.5 Defining Functions with Default Arguments--222
7.6 Defining Anonymous or Inline Functions--224
7.7 Capturing Variables in Anonymous Functions--225
7.8 Making an N-Argument Callable Work As a Callable with Fewer Arguments----------------227
7.9 Replacing Single Method Classes with Functions--231
7.10 Carrying Extra State with Callback Functions---232
7.11 Inlining Callback Functions--235
7.12 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 Encapsulating Names in a Class---250
8.6 Creating Managed Attributes--251
8.7 Calling a Method on a Parent Class--256
8.8 Extending a Property in a Subclass---260
8.9 Creating a New Kind of Class or Instance Attribute--264
8.10 Using Lazily Computed Properties--267
8.11 Simplifying the Initialization of Data Structures---270
8.12 Defining an Interface or Abstract Base Class--274
8.13 Implementing a Data Model or Type System--277
8.14 Implementing Custom Containers--283
8.15 Delegating Attribute Access--287
8.16 Defining More Than One Constructor in a Class--291
8.17 Creating an Instance Without Invoking init--293
8.18 Extending Classes with Mixins--294
8.19 Implementing Stateful Objects or State Machines--299
8.20 Calling a Method on an Object Given the Name As a String-------------------------------305
8.21 Implementing the Visitor Pattern---306
8.22 Implementing the Visitor Pattern Without Recursion---------------------------------------311
8.23 Managing Memory in Cyclic Data Structures---317

8.24 Making Classes Support Comparison Operations--321
8.25 Creating Cached Instances--323
9 Metaprogramming 329
9.1 Putting a Wrapper Around a Function--329
9.2 Preserving Function Metadata When Writing Decorators---------------------------------------331
9.3 Unwrapping a Decorator--333
9.4 Defining a Decorator That Takes Arguments---334
9.5 Defining a Decorator with User Adjustable Attributes---336
9.6 Defining a Decorator That Takes an Optional Argument--339
9.7 Enforcing Type Checking on a Function Using a Decorator-------------------------------------341
9.8 Defining Decorators As Part of a Class---345
9.9 Defining Decorators As Classes---347
9.10 Applying Decorators to Class and Static Methods--350
9.11 Writing Decorators That Add Arguments to Wrapped Functions----------------------------352
9.12 Using Decorators to Patch Class Definitions---355
9.13 Using a Metaclass to Control Instance Creation---356
9.14 Capturing Class Attribute Definition Order---359
9.15 Defining a Metaclass That Takes Optional Arguments-------------------------------------362
9.16 Enforcing an Argument Signature on *args and **kwargs----------------------------------364
9.17 Enforcing Coding Conventions in Classes--367
9.18 Defining Classes Programmatically---370
9.19 Initializing Class Members at Definition Time---374
9.20 Implementing Multiple Dispatch with Function Annotations-------------------------------376
9.21 Avoiding Repetitive Property Methods---382
9.22 Defining Context Managers the Easy Way--384
9.23 Executing Code with Local Side Effects--386
9.24 Parsing and Analyzing Python Source---388
9.25 Disassembling Python Byte Code--392
10 Modules and Packages 397
10.1 Making a Hierarchical Package of Modules---397
10.2 Controlling the Import of Everything--398
10.3 Importing Package Submodules Using Relative Names-------------------------------------399
10.4 Splitting a Module into Multiple Files--401
10.5 Making Separate Directories of Code Import Under a Common
Namespace---404
10.6 Reloading Modules---406
10.7 Making a Directory or Zip File Runnable As a Main Script---------------------------------407
10.8 Reading Datafiles Within a Package--408
10.9 Adding Directories to sys.path---409
10.10 Importing Modules Using a Name Given in a String---------------------------------------411
10.11 Loading Modules from a Remote Machine Using Import Hooks----------------------------412
10.12 Patching Modules on Import--428
10.13 Installing Packages Just for Yourself--431
10.14 Creating a New Python Environment---432
10.15 Distributing Packages---433
11 Network and Web Programming 437
11.1 Interacting with HTTP Services As a Client---437
11.2 Creating a TCP Server---441
11.3 Creating a UDP Server--445
11.4 Generating a Range of IP Addresses from a CIDR Address---------------------------------447
11.5 Creating a Simple REST-Based Interface--449
11.6 Implementing a Simple Remote Procedure Call with XML-RPC-----------------------------454

11.7 Communicating Simply Between Interpreters---456
11.8 Implementing Remote Procedure Calls---458
11.9 Authenticating Clients Simply---461
11.10 Adding SSL to Network Services---464
11.11 Passing a Socket File Descriptor Between Processes---470
11.12 Understanding Event-Driven I/O---475
11.13 Sending and Receiving Large Arrays---481
12 Concurrency 485
12.1 Starting and Stopping Threads---485
12.2 Determining If a Thread Has Started---488
12.3 Communicating Between Threads---491
12.4 Locking Critical Sections---497
12.5 Locking with Deadlock Avoidance--500
12.6 Storing Thread-Specific State---504
12.7 Creating a Thread Pool--505
12.8 Performing Simple Parallel Programming---509
12.9 Dealing with the GIL (and How to Stop Worrying About It)----------------------------------513
12.10 Defining an Actor Task--516
12.11 Implementing Publish/Subscribe Messaging--520
12.12 Using Generators As an Alternative to Threads---524
12.13 Polling Multiple Thread Queues--531
12.14 Launching a Daemon Process on Unix---534
13 Utility Scripting and System Administration 539
13.1 Accepting Script Input via Redirection, Pipes, or Input Files--------------------------------539
13.2 Terminating a Program with an Error Message---540
13.3 Parsing Command-Line Options---541
13.4 Prompting for a Password at Runtime--544
13.5 Getting the Terminal Size--545
13.6 Executing an External Command and Getting Its Output---------------------------------------545
13.7 Copying or Moving Files and Directories--547
13.8 Creating and Unpacking Archives---549
13.9 Finding Files by Name---550
13.10 Reading Configuration Files--552
13.11 Adding Logging to Simple Scripts---555
13.12 Adding Logging to Libraries---558
13.13 Making a Stopwatch Timer---559
13.14 Putting Limits on Memory and CPU Usage--561
13.15 Launching a Web Browser---563
14 Testing, Debugging, and Exceptions 565
14.1 Testing Output Sent to stdout--565
14.2 Patching Objects in Unit Tests---567
14.3 Testing for Exceptional Conditions in Unit Tests---570
14.4 Logging Test Output to a File---572
14.5 Skipping or Anticipating Test Failures--573
14.6 Handling Multiple Exceptions--574
14.7 Catching All Exceptions---576
14.8 Creating Custom Exceptions---578
14.9 Raising an Exception in Response to Another Exception---------------------------------------580
14.10 Reraising the Last Exception---582
14.11 Issuing Warning Messages--583
14.12 Debugging Basic Program Crashes--585
14.13 Profiling and Timing Your Program---587

14.14 Making Your Programs Run Faster--590
15 C Extensions 597
15.1 Accessing C Code Using ctypes--599
15.2 Writing a Simple C Extension Module--605
15.3 Writing an Extension Function That Operates on Arrays---------------------------------------609
15.4 Managing Opaque Pointers in C Extension Modules---612
15.5 Defining and Exporting C APIs from Extension Modules--------------------------------------614
15.6 Calling Python from C---619
15.7 Releasing the GIL in C Extensions--625
15.8 Mixing Threads from C and Python--625
15.9 Wrapping C Code with Swig--627
15.10 Wrapping Existing C Code with Cython--632
15.11 Using Cython to Write High-Performance Array Operations---------------------------------638
15.12 Turning a Function Pointer into a Callable--643
15.13 Passing NULL-Terminated Strings to C Libraries---644
15.14 Passing Unicode Strings to C Libraries---648
15.15 Converting C Strings to Python--653
15.16 Working with C Strings of Dubious Encoding--654
15.17 Passing Filenames to C Extensions--657
15.18 Passing Open Files to C Extensions--658
15.19 Reading File-Like Objects from C---659
15.20 Consuming an Iterable from C---662
15.21 Diagnosing Segmentation Faults---663
A Further Reading---665
Index--667

***python-excel_2009_56p

Introduction
xlrd
xlwt
xlutils
There are still reasons why automating an Excel instance via COM is necessary

***python-pocket-reference-5th-edition-MarkLutz-2014-264p

Introduction 1
Book Conventions 2
Python Command-Line Usage 3
Python Command Options---4
Command-Line Program Specification--5
Python 2.X Command Options---7
Python Environment Variables 7
Operational Variables--8
Python Command Option Variables--9
Python Windows Launcher Usage 10
Launcher File Directives---10
Launcher Command Lines---11
Launcher Environment Variables--11

Built-in Types and Operators 12

Operators and Precedence---12
Operator Usage Notes--14
Operations by Category---16
Sequence Operation Notes--20

Specific Built-in Types 21

Numbers--22
Strings---24
Unicode Strings---42
Lists---46
Dictionaries--53
Tuples--57
Files---58
Sets---63
Other Types and Conversions--65
Statements and Syntax---67
Syntax Rules--67
Name Rules---69
Specific Statements--71
The Assignment Statement--72
The Expression Statement--76
The print Statement--77
The if Statement---80
The while Statement---80
The for Statement---80
The pass Statement---81
The break Statement--81
The continue Statement---81
The del Statement---81
The def Statement---82
The return Statement--86
The yield Statement---87
The global Statement--88
The nonlocal Statement---89
The import Statement---89
The from Statement---93
The class Statement---95
The try Statement---97
The raise Statement---99
The assert Statement---101
The with Statement---102
Python 2.X Statements---104

Namespace and Scope Rules 105

Qualified Names: Object Namespaces---105
Unqualified Names: Lexical Scopes---105
Nested Scopes and Closures---107

Object-Oriented Programming 108

Classes and Instances---109
Pseudoprivate Attributes--110
New-Style Classes--111
Formal Inheritance Rules--112

Operator Overloading Methods 117

Methods for All Types--118
Methods for Collections (Sequences, Mappings)--123
Methods for Numbers (Binary Operators)---125
Methods for Numbers (Other Operations)---128
Methods for Descriptors---129
Methods for Context Managers---130
Python 2.X Operator Overloading Methods---131
Built-in Functions 134
Python 2.X Built-in Functions---155
Built-in Exceptions 161
Superclasses: Categories---162
Specific Exceptions--163
Specific OSError Exceptions---167
Warning Category Exceptions---169
Warnings Framework--170
Python 3.2 Built-in Exceptions---171
Python 2.X Built-in Exceptions---172
Built-in Attributes 172
Standard Library Modules 173
The sys Module--174
The string Module---182
Functions and Classes---182
Constants--183
The os System Module--184
Administrative Tools--185
Portability Constants---186
Shell Commands---187
Environment Tools---189
File Descriptor Tools---190
File Pathname Tools--193
Process Control---197
The os.path Module---200
The re Pattern-Matching Module---202
Module Functions---202
Regular Expression Objects--205
Match Objects--205
Pattern Syntax--207
Object Persistence Modules--210
The shelve and dbm Modules--211
The pickle Module---213
The tkinter GUI Module and Tools---216
tkinter Example--216
tkinter Core Widgets--217
Common Dialog Calls---218
Additional tkinter Classes and Tools--219
Tcl/Tk-to-Python/tkinter Mappings---220
Internet Modules and Tools--221
Other Standard Library Modules---224
The math Module---224
The time Module--225
The timeit Module--226
The datetime Module---227

The random Module---228
The json Module--228
The subprocess Module--229
The enum Module---229
The struct Module---230
Threading Modules---231
Python SQL Database API---232
API Usage Example--233
Module Interface---234
Connection Objects--234
Cursor Objects---235
Type Objects and Constructors--236
More Hints and Idioms---236
Core Language Hints---237
Environment Hints---238
Usage Hints---240
Assorted Hints---242
Index--243

***SciPy_and_NumPy_2013_66p

Preface---v
1 Introduction---1
1.1 Why SciPy and NumPy?---1
1.2 Getting NumPy and SciPy---2
1.3 Working with SciPy and NumPy---3
2 NumPy---5
2.1 NumPy Arrays---5
2.2 Boolean Statements and NumPy Arrays--10
2.3 Read and Write--12
2.4 Math--14
3 SciPy--17
3.1 Optimization and Minimization---17
3.2 Interpolation--22
3.3 Integration--26
3.4 Statistics--28
3.5 Spatial and Clustering Analysis---32
3.6 Signal and Image Processing--38
3.7 Sparse Matrices--40
3.8 Reading and Writing Files Beyond NumPy---41
4 SciKit: Taking SciPy One Step Further--43
4.1 Scikit-Image---43
4.2 Scikit-Learn--48
5 Conclusion---55
5.1 Summary---55
5.2 What’s Next?---55

***Testing_Python_2014_243p

Introduction--1
CHAPTER 1
A History of Testing--5
You Do Test, Don't You? 7
Fundamentals and Best Practices 7
Python Installation 8
Linux--8
Mac 8
Windows 8
Pip--9
Virtualenv 9
Source Control (SVN, Git)--10
Interactive Development Environment (IDE)---11
Summary---12
CHAPTER 2
Writing Unit Tests--15
What Is Unit Testing?---15
What Should You Test?---17
Writing Your First Unit Test--17
Checking Values with the assertEquals Method--18
Checking Exception Handling with assertRaises--20
Following the PEP-8 Standard--22
Unit Test Structure--23
Additional Unit Test Examples--24
Getting Clever with assertRaises--24
Making Your Life Easier with setUp---25
Useful Methods in Unit Testing--27
assertEqual(x, y, msg=None)---27
assertAlmostEqual(x, y, places=None, msg=None, delta=None)-------------------------------------27
assertRaises(exception, method, arguments, msg=None)---28
assertDictContainsSubset(expected, actual, msg=None)---28
assertDictEqual(d1, d2, msg=None)---28
x TESTING PYTHON
assertTrue(expr, msg=None)---28
assertFalse(expr, msg=None)--29
assertGreater(a, b, msg=None)--29
assertGreaterEqual(a, b, msg=None)---29
assertIn(member, container, msg=None)--30
assertIs(expr1, expr2)---30
assertIsInstance(obj, class, msg=None)--30
assertNotIsInstance(obj, class, msg=None)--30
assertIsNone(obj, msg=None)--30
assertIsNot(expr1, expr2, msg=None)---31
assertIsNotNone(obj, msg=None)--31
assertLess(a, b, msg=None)---31
assertLessEqual(a, b, msg=None)--31
assertItemsEqual(a, b, msg=None)---31
assertRaises(excClass, callableObj, *args, **kwargs, msg=None)------------------------------------32
Summary---32
CHAPTER 3
Utilizing Unit Test Tools---33

Using Python’s Nose--33
Installing Nose---34
Using Nose’s Best Features--35
Running Specific Test Files--35
Getting More Detail with Verbose---35
Debugging Support with PDB 36
Checking Your Coverage---38
Coloring your tests with Rednose---39
PyTest: An Alternative Test Runner---40
Installing PyTest---40
PyTest’s Best Features 41
Running Specific Tests---41
Viewing Detail with Verbose and Summary---42
Debugging with PDB---43
Checking Your Coverage with PyTest--45
Choosing Between Nose and PyTest---46
Mock and Patch Tricky Situations--46
TABLE OF CONTENTS---xi
Installing the Mock Library--47
Mocking a Class and Method Response---47
When Mock Won’t Do, Patch!--50
The Requests Library---50
Patch in Action--50
Advanced Mocking---52
Summary---53
CHAPTER 4
Writing Testable Documentation--55
Writing Your First Doctest---56
The Python Shell--56
Adding Doctests to a Method--57
Running Your Doctests---58
Handling Error Cases---59
Advanced Doctest Usage---61
Improving Doctests with Nose Integration---62
Summary---65
Resources--65
CHAPTER 5
Driving Your Development with Tests 67
Agile Development---67
Adopting the Agile Process Now---68
Ethos of Test Driven Development---70
Advantages of Test Driven Development--72
Ping-Pong Programming---72
Test Driving Your Problem--73
Writing Your Failing Test--74
Making Your Test Pass---75
Driving More Features with Tests--75
Wrapping Up the Task--77
Summary---82
Resources--83
xii TESTING PYTHON
CHAPTER 6

Writing Acceptance Tests--85
What Is Acceptance Testing?--85
Anatomy of an Acceptance Test--87
Using Gherkin Syntax---87
The Magic Is in the Step File--88
Goals of Acceptance Testing--89
Implementing Developer and QA Collaboration--90
Letting Behavior Drive Your Problem--90
Writing Your Failing Acceptance Test--90
Defining Your Steps--92
Implementing Your Code---94
Developing More of the Feature---95
bank_app py--96
index html---97
Delivering the Finished Article--98
Advanced Acceptance Test Techniques---102
Scenario Outline---102
Tables of Data in Scenarios--103
Summary--104
Resources---105
C HAP TER 7
Utilizing Acceptance Test Tools--107
Cucumber: The Acceptance Test Standard---107
Lettuce in Detail---108
Tagging--108
Fail Fast---112
Nosetest Integration---114
Robot: An Alternative Test Framework---115
Installing Robot--116
Writing a Test Case--116
Implementing Keywords--117
Running Robot Tests 119
Summary--123
Resources---123
TABLE OF CONTENTS--xiii
C HAP T E R 8
Maximizing Your Code's Performance--125
Understanding the Importance of Performance Testing--126
JMeter and Python---126
Installation--127
Configuring Your Test Plans---128
Utilizing Your Test Plans Effectively---135
Code Profiling with cProfile---135
Run a cProfile Session--136
Analyzing the cProfile Output---142
Summary--144
Resources---144
C HAP T E R 9
Looking After Your Lint--145
Coming to Grips with Pylint---146
Installing Pylint--146
Using Pylint---146

Understanding the Pylint Report--149
The Module Block---149
The Messages by Category Section---149
The Messages Section---150
The Code Evaluation Score--150
The Raw Metrics Section---150
The Statistics by Type Section---150
Customizing Pylint’s Output--150
Telling Pylint to Ignore Errors--153
Covering All Your Code with Unit Tests--154
Installing Coverage--155
Using Coverage 155
Advanced Coverage Options--157
Producing an HTML/XML Report--157
Setting a Minimum Coverage Threshold---159
Restricting Coverage to a Specific Package---159
Ignoring Coverage--160
Summary---161
Resources--162

C HAP TER 1 0
Automating Your Processes---163
Build Paver Tasks---164
Installing Paver--164
Creating a Paver Task--164
Executing Paver Tasks---165
Defining a Default Build--166
Setting Up Automated Builds--168
Installing Jenkins---169
Adding Coverage and PyLint Reports--175
Generating a PyLint Report---175
Generating a Coverage Report--176
Making Your Build Status Highly Visible---176
Summary---181
Resources--181
CHAPTER 11
Deploying Your Application--183
Deploying Your Application to Production---184
Creating a Deployable Artifact---185
Defining the Paver Tasks---185
Incorporating Packaging into the Build---187
Enabling Archiving on Jenkins--188
QA Environment--189
Implementing Stage and Production Environments--190
Implementing a Cloud Deployment--191
Creating a Heroku Account---192
Creating a Small Application 193
Setting up Git for Heroku--193
Deploying the Application to Heroku---194
Smoke Testing a Deployed Application--195
Example Application Stack--196
Smoke Test Scenarios--197

Implementing Smoke Tests--198
Summary--200
Resources--201
TABLE OF CONTENTS---xv
CHAPTER 12
The Future of Testing Python--203
Stub the Solution--203
Making Deployment Natural---205
Automating (Nearly) Everything---206
Working in Public--207
Collaborating on Step Definitions---208
Final Thoughts---209
Resources---210
Index--211

***The_Python_Standard_Library_by_Example_2011_1343p

CONTENTS AT A GLANCE

Tables--xxxi
Foreword--xxxiii
Acknowledgments--xxxvii
About the Author---xxxix
INTRODUCTION--1
1 TEXT--3
2 DATA STRUCTURES--69
3 ALGORITHMS--129
4 DATES AND TIMES---173
5 MATHEMATICS---197
6 THE FILE SYSTEM--247
7 DATA PERSISTENCE AND EXCHANGE---333
8 DATA COMPRESSION AND ARCHIVING--421
9 CRYPTOGRAPHY--469
10 PROCESSES AND THREADS--481
11 NETWORKING---561
12 THE INTERNET--637
13 EMAIL---727
14 APPLICATION BUILDING BLOCKS---769
15 INTERNATIONALIZATION AND LOCALIZATION------------------------------------899
16 DEVELOPER TOOLS--919
17 RUNTIME FEATURES---1045
18 LANGUAGE TOOLS---1169
19 MODULES AND PACKAGES---1235
Index of Python Modules--1259
Index--1261

CONTENTS

Tables--xxxi
Foreword--xxxiii
Acknowledgments--xxxvii
About the Author---xxxix
INTRODUCTION--1
1 TEXT--3

1.1 string—Text Constants and Templates--4
1.1.1 Functions---4
1.1.2 Templates--5
1.1.3 Advanced Templates---7
1.2 textwrap—Formatting Text Paragraphs--9
1.2.1 Example Data--9
1.2.2 Filling Paragraphs--10
1.2.3 Removing Existing Indentation--10
1.2.4 Combining Dedent and Fill---11
1.2.5 Hanging Indents---12
1.3 re—Regular Expressions---13
1.3.1 Finding Patterns in Text---14
1.3.2 Compiling Expressions---14
1.3.3 Multiple Matches--15
1.3.4 Pattern Syntax---16
1.3.5 Constraining the Search---28
1.3.6 Dissecting Matches with Groups---30
1.3.7 Search Options--37
1.3.8 Looking Ahead or Behind---45
1.3.9 Self-Referencing Expressions--50
1.3.10 Modifying Strings with Patterns--56
1.3.11 Splitting with Patterns--58
1.4 difflib—Compare Sequences---61
1.4.1 Comparing Bodies of Text---62
1.4.2 Junk Data---65
1.4.3 Comparing Arbitrary Types---66
2 DATA STRUCTURES---69
2.1 collections—Container Data Types--70
2.1.1 Counter---70
2.1.2 defaultdict--74
2.1.3 Deque---75
2.1.4 namedtuple---79
2.1.5 OrderedDict--82
2.2 array—Sequence of Fixed-Type Data--84
2.2.1 Initialization--84
2.2.2 Manipulating Arrays--85
2.2.3 Arrays and Files---85
2.2.4 Alternate Byte Ordering--86
2.3 heapq—Heap Sort Algorithm---87
2.3.1 Example Data--88
2.3.2 Creating a Heap---89
2.3.3 Accessing Contents of a Heap--90
2.3.4 Data Extremes from a Heap---92
2.4 bisect—Maintain Lists in Sorted Order--93
2.4.1 Inserting in Sorted Order--93
2.4.2 Handling Duplicates--95
2.5 Queue—Thread-Safe FIFO Implementation--96
2.5.1 Basic FIFO Queue---96
2.5.2 LIFO Queue--97
2.5.3 Priority Queue---98
2.5.4 Building a Threaded Podcast Client--99
2.6 struct—Binary Data Structures---102

2.6.1 Functions vs Struct Class--102
2.6.2 Packing and Unpacking--102
2.6.3 Endianness---103
2.6.4 Buffers---105
2.7 weakref—Impermanent References to Objects--106
2.7.1 References---107
2.7.2 Reference Callbacks---108
2.7.3 Proxies--108
2.7.4 Cyclic References--109
2.7.5 Caching Objects--114
2.8 copy—Duplicate Objects---117
2.8.1 Shallow Copies--118
2.8.2 Deep Copies---118
2.8.3 Customizing Copy Behavior---119
2.8.4 Recursion in Deep Copy--120
2.9 pprint—Pretty-Print Data Structures---123
2.9.1 Printing---123
2.9.2 Formatting---124
2.9.3 Arbitrary Classes---125
2.9.4 Recursion--125
2.9.5 Limiting Nested Output---126
2.9.6 Controlling Output Width--126
3 ALGORITHMS--129
3.1 functools—Tools for Manipulating Functions--129
3.1.1 Decorators---130
3.1.2 Comparison--138
3.2 itertools—Iterator Functions---141
3.2.1 Merging and Splitting Iterators---142
3.2.2 Converting Inputs--145
3.2.3 Producing New Values--146
3.2.4 Filtering--148
3.2.5 Grouping Data--151
3.3 operator—Functional Interface to Built-in Operators---153
3.3.1 Logical Operations---154
3.3.2 Comparison Operators---154
3.3.3 Arithmetic Operators--155
3.3.4 Sequence Operators--157
3.3.5 In-Place Operators---158
3.3.6 Attribute and Item “Getters”---159
3.3.7 Combining Operators and Custom Classes--161
3.3.8 Type Checking--162
3.4 contextlib—Context Manager Utilities---163
3.4.1 Context Manager API--164
3.4.2 From Generator to Context Manager---167
3.4.3 Nesting Contexts--168
3.4.4 Closing Open Handles---169
4 DATES AND TIMES---173
4.1 time—Clock Time--173
4.1.1 Wall Clock Time--174
4.1.2 Processor Clock Time--174
4.1.3 Time Components--176
4.1.4 Working with Time Zones--177

4.1.5 Parsing and Formatting Times--179
4.2 datetime—Date and Time Value Manipulation--180
4.2.1 Times--181
4.2.2 Dates--182
4.2.3 timedeltas---185
4.2.4 Date Arithmetic--186
4.2.5 Comparing Values---187
4.2.6 Combining Dates and Times--188
4.2.7 Formatting and Parsing--189
4.2.8 Time Zones---190
4.3 calendar—Work with Dates---191
4.3.1 Formatting Examples--191
4.3.2 Calculating Dates--194
5 MATHEMATICS---197
5.1 decimal—Fixed and Floating-Point Math--197
5.1.1 Decimal---198
5.1.2 Arithmetic--199
5.1.3 Special Values---200
5.1.4 Context---201
5.2 fractions—Rational Numbers--207
5.2.1 Creating Fraction Instances---207
5.2.2 Arithmetic--210
5.2.3 Approximating Values---210
5.3 random—Pseudorandom Number Generators---211
5.3.1 Generating Random Numbers--211
5.3.2 Seeding--212
5.3.3 Saving State--213
5.3.4 Random Integers--214
5.3.5 Picking Random Items--215
5.3.6 Permutations--216
5.3.7 Sampling--218
5.3.8 Multiple Simultaneous Generators---219
5.3.9 SystemRandom--221
5.3.10 Nonuniform Distributions--222
5.4 math—Mathematical Functions--223
5.4.1 Special Constants---223
5.4.2 Testing for Exceptional Values--224
5.4.3 Converting to Integers--226
5.4.4 Alternate Representations---227
5.4.5 Positive and Negative Signs---229
5.4.6 Commonly Used Calculations--230
5.4.7 Exponents and Logarithms--234
5.4.8 Angles--238
5.4.9 Trigonometry---240
5.4.10 Hyperbolic Functions--243
5.4.11 Special Functions--244
6 THE FILE SYSTEM---247
6.1 os.path—Platform-Independent Manipulation of Filenames-------------------------------248
6.1.1 Parsing Paths---248
6.1.2 Building Paths--252
6.1.3 Normalizing Paths--253
6.1.4 File Times--254

6.1.5 Testing Files--255
6.1.6 Traversing a Directory Tree--256
6.2 glob—Filename Pattern Matching--257
6.2.1 Example Data---258
6.2.2 Wildcards--258
6.2.3 Single Character Wildcard--259
6.2.4 Character Ranges---260
6.3 linecache—Read Text Files Efficiently---261
6.3.1 Test Data--261
6.3.2 Reading Specific Lines---262
6.3.3 Handling Blank Lines---263
6.3.4 Error Handling---263
6.3.5 Reading Python Source Files--264
6.4 tempfile—Temporary File System Objects--265
6.4.1 Temporary Files--265
6.4.2 Named Files--268
6.4.3 Temporary Directories--268
6.4.4 Predicting Names---269
6.4.5 Temporary File Location--270
6.5 shutil—High-Level File Operations---271
6.5.1 Copying Files--271
6.5.2 Copying File Metadata--274
6.5.3 Working with Directory Trees---276
6.6 mmap—Memory-Map Files---279
6.6.1 Reading--279
6.6.2 Writing--280
6.6.3 Regular Expressions--283
6.7 codecs—String Encoding and Decoding--284
6.7.1 Unicode Primer---284
6.7.2 Working with Files---287
6.7.3 Byte Order---289
6.7.4 Error Handling---291
6.7.5 Standard Input and Output Streams---295
6.7.6 Encoding Translation---298
6.7.7 Non-Unicode Encodings---300
6.7.8 Incremental Encoding---301
6.7.9 Unicode Data and Network Communication---303
6.7.10 Defining a Custom Encoding---307
6.8 StringIO—Text Buffers with a File-like API---314
6.8.1 Examples---314
6.9 fnmatch—UNIX-Style Glob Pattern Matching--315
6.9.1 Simple Matching--315
6.9.2 Filtering--317
6.9.3 Translating Patterns---318
6.10 dircache—Cache Directory Listings---319
6.10.1 Listing Directory Contents---319
6.10.2 Annotated Listings---321
6.11 filecmp—Compare Files---322
6.11.1 Example Data--323
6.11.2 Comparing Files--325
6.11.3 Comparing Directories--327
6.11.4 Using Differences in a Program---328

7 DATA PERSISTENCE AND EXCHANGE--333
7.1 pickle—Object Serialization---334
7.1.1 Importing--335
7.1.2 Encoding and Decoding Data in Strings--335
7.1.3 Working with Streams--336
7.1.4 Problems Reconstructing Objects---338
7.1.5 Unpicklable Objects--340
7.1.6 Circular References---340
7.2 shelve—Persistent Storage of Objects--343
7.2.1 Creating a New Shelf---343
7.2.2 Writeback--344
7.2.3 Specific Shelf Types--346
7.3 anydbm—DBM-Style Databases---347
7.3.1 Database Types---347
7.3.2 Creating a New Database---348
7.3.3 Opening an Existing Database---349
7.3.4 Error Cases--349
7.4 whichdb—Identify DBM-Style Database Formats--350
7.5 sqlite3—Embedded Relational Database---351
7.5.1 Creating a Database---352
7.5.2 Retrieving Data--355
7.5.3 Query Metadata--357
7.5.4 Row Objects--358
7.5.5 Using Variables with Queries--359
7.5.6 Bulk Loading---362
7.5.7 Defining New Column Types---363
7.5.8 Determining Types for Columns---366
7.5.9 Transactions--368
7.5.10 Isolation Levels---372
7.5.11 In-Memory Databases--376
7.5.12 Exporting the Contents of a Database--376
7.5.13 Using Python Functions in SQL---378
7.5.14 Custom Aggregation--380
7.5.15 Custom Sorting--381
7.5.16 Threading and Connection Sharing---383
7.5.17 Restricting Access to Data---384
7.6 xml.etree.ElementTree—XML Manipulation API---387
7.6.1 Parsing an XML Document---387
7.6.2 Traversing the Parsed Tree--388
7.6.3 Finding Nodes in a Document---390
7.6.4 Parsed Node Attributes---391
7.6.5 Watching Events While Parsing---393
7.6.6 Creating a Custom Tree Builder---396
7.6.7 Parsing Strings--398
7.6.8 Building Documents with Element Nodes--400
7.6.9 Pretty-Printing XML---401
7.6.10 Setting Element Properties--403
7.6.11 Building Trees from Lists of Nodes---405
7.6.12 Serializing XML to a Stream---408
7.7 csv—Comma-Separated Value Files--411
7.7.1 Reading--411
7.7.2 Writing---412

7.7.3 Dialects--413
7.7.4 Using Field Names--418
8 DATA COMPRESSION AND ARCHIVING---421
8.1 zlib—GNU zlib Compression---421
8.1.1 Working with Data in Memory--422
8.1.2 Incremental Compression and Decompression---------------------------------------423
8.1.3 Mixed Content Streams--424
8.1.4 Checksums--425
8.1.5 Compressing Network Data---426
8.2 gzip—Read and Write GNU Zip Files---430
8.2.1 Writing Compressed Files---431
8.2.2 Reading Compressed Data--433
8.2.3 Working with Streams--434
8.3 bz2—bzip2 Compression---436
8.3.1 One-Shot Operations in Memory---436
8.3.2 Incremental Compression and Decompression---------------------------------------438
8.3.3 Mixed Content Streams--439
8.3.4 Writing Compressed Files---440
8.3.5 Reading Compressed Files--442
8.3.6 Compressing Network Data---443
8.4 tarfile—Tar Archive Access--448
8.4.1 Testing Tar Files--448
8.4.2 Reading Metadata from an Archive---449
8.4.3 Extracting Files from an Archive---450
8.4.4 Creating New Archives--453
8.4.5 Using Alternate Archive Member Names--453
8.4.6 Writing Data from Sources Other than Files--454
8.4.7 Appending to Archives--455
8.4.8 Working with Compressed Archives---456
8.5 zipfile—ZIP Archive Access--457
8.5.1 Testing ZIP Files--457
8.5.2 Reading Metadata from an Archive---457
8.5.3 Extracting Archived Files from an Archive---459
8.5.4 Creating New Archives--460
8.5.5 Using Alternate Archive Member Names--462
8.5.6 Writing Data from Sources Other than Files--462
8.5.7 Writing with a ZipInfo Instance---463
8.5.8 Appending to Files--464
8.5.9 Python ZIP Archives---466
8.5.10 Limitations--467
9 CRYPTOGRAPHY---469
9.1 hashlib—Cryptographic Hashing---469
9.1.1 Sample Data--470
9.1.2 MD5 Example--470
9.1.3 SHA-1 Example--470
9.1.4 Creating a Hash by Name---471
9.1.5 Incremental Updates---472
9.2 hmac—Cryptographic Message Signing and Verification---------------------------------473
9.2.1 Signing Messages--474
9.2.2 SHA vs MD5--474
9.2.3 Binary Digests---475
9.2.4 Applications of Message Signatures---476

10 PROCESSES AND THREADS--481
10.1 subprocess—Spawning Additional Processes--481
10.1.1 Running External Commands---482
10.1.2 Working with Pipes Directly---486
10.1.3 Connecting Segments of a Pipe--489
10.1.4 Interacting with Another Command--490
10.1.5 Signaling between Processes---492
10.2 signal—Asynchronous System Events--497
10.2.1 Receiving Signals--498
10.2.2 Retrieving Registered Handlers---499
10.2.3 Sending Signals---501
10.2.4 Alarms---501
10.2.5 Ignoring Signals--502
10.2.6 Signals and Threads--502
10.3 threading—Manage Concurrent Operations---505
10.3.1 Thread Objects--505
10.3.2 Determining the Current Thread--507
10.3.3 Daemon vs Non-Daemon Threads---509
10.3.4 Enumerating All Threads--512
10.3.5 Subclassing Thread---513
10.3.6 Timer Threads---515
10.3.7 Signaling between Threads--516
10.3.8 Controlling Access to Resources---517
10.3.9 Synchronizing Threads---523
10.3.10 Limiting Concurrent Access to Resources--524
10.3.11 Thread-Specific Data--526
10.4 multiprocessing—Manage Processes like Threads--529
10.4.1 Multiprocessing Basics---529
10.4.2 Importable Target Functions--530
10.4.3 Determining the Current Process---531
10.4.4 Daemon Processes--532
10.4.5 Waiting for Processes---534
10.4.6 Terminating Processes--536
10.4.7 Process Exit Status--537
10.4.8 Logging--539
10.4.9 Subclassing Process--540
10.4.10 Passing Messages to Processes--541
10.4.11 Signaling between Processes--545
10.4.12 Controlling Access to Resources--546
10.4.13 Synchronizing Operations--547
10.4.14 Controlling Concurrent Access to Resources---548
10.4.15 Managing Shared State---550
10.4.16 Shared Namespaces---551
10.4.17 Process Pools---553
10.4.18 Implementing MapReduce--555
11 NETWORKING--561
11.1 socket—Network Communication--561
11.1.1 Addressing, Protocol Families, and Socket Types--562
11.1.2 TCP/IP Client and Server---572
11.1.3 User Datagram Client and Server---580
11.1.4 UNIX Domain Sockets--583
11.1.5 Multicast---587

11.1.6 Sending Binary Data--591
11.1.7 Nonblocking Communication and Timeouts---593
11.2 select—Wait for I/O Efficiently---594
11.2.1 Using select()---595
11.2.2 Nonblocking I/O with Timeouts---601
11.2.3 Using poll()--603
11.2.4 Platform-Specific Options---608
11.3 SocketServer—Creating Network Servers--609
11.3.1 Server Types---609
11.3.2 Server Objects---609
11.3.3 Implementing a Server---610
11.3.4 Request Handlers---610
11.3.5 Echo Example---610
11.3.6 Threading and Forking--616
11.4 asyncore—Asynchronous I/O---619
11.4.1 Servers--619
11.4.2 Clients---621
11.4.3 The Event Loop---623
11.4.4 Working with Other Event Loops--625
11.4.5 Working with Files---628
11.5 asynchat—Asynchronous Protocol Handler--629
11.5.1 Message Terminators--629
11.5.2 Server and Handler---630
11.5.3 Client---632
11.5.4 Putting It All Together--634
12 THE INTERNET---637
12.1 urlparse—Split URLs into Components--638
12.1.1 Parsing---638
12.1.2 Unparsing--641
12.1.3 Joining---642
12.2 BaseHTTPServer—Base Classes for Implementing Web Servers-----------------------------644
12.2.1 HTTP GET--644
12.2.2 HTTP POST--646
12.2.3 Threading and Forking---648
12.2.4 Handling Errors---649
12.2.5 Setting Headers---650
12.3 urllib—Network Resource Access---651
12.3.1 Simple Retrieval with Cache--651
12.3.2 Encoding Arguments--653
12.3.3 Paths vs URLs---655
12.4 urllib2—Network Resource Access---657
12.4.1 HTTP GET--657
12.4.2 Encoding Arguments--660
12.4.3 HTTP POST--661
12.4.4 Adding Outgoing Headers--661
12.4.5 Posting Form Data from a Request--663
12.4.6 Uploading Files--664
12.4.7 Creating Custom Protocol Handlers--667
12.5 base64—Encode Binary Data with ASCII---670
12.5.1 Base64 Encoding--670
12.5.2 Base64 Decoding--671
12.5.3 URL-Safe Variations--672

12.5.4 Other Encodings--673
12.6 robotparser—Internet Spider Access Control--674
12.6.1 robots.txt---674
12.6.2 Testing Access Permissions--675
12.6.3 Long-Lived Spiders--676
12.7 Cookie—HTTP Cookies--677
12.7.1 Creating and Setting a Cookie---678
12.7.2 Morsels---678
12.7.3 Encoded Values--680
12.7.4 Receiving and Parsing Cookie Headers---681
12.7.5 Alternative Output Formats--682
12.7.6 Deprecated Classes--683
12.8 uuid—Universally Unique Identifiers---684
12.8.1 UUID 1—IEEE 802 MAC Address---684
12.8.2 UUID 3 and 5—Name-Based Values---686
12.8.3 UUID 4—Random Values---688
12.8.4 Working with UUID Objects--689
12.9 json—JavaScript Object Notation---690
12.9.1 Encoding and Decoding Simple Data Types---690
12.9.2 Human-Consumable vs Compact Output--692
12.9.3 Encoding Dictionaries---694
12.9.4 Working with Custom Types--695
12.9.5 Encoder and Decoder Classes---697
12.9.6 Working with Streams and Files---700
12.9.7 Mixed Data Streams--701
12.10 xmlrpclib—Client Library for XML-RPC--702
12.10.1 Connecting to a Server---704
12.10.2 Data Types---706
12.10.3 Passing Objects--709
12.10.4 Binary Data--710
12.10.5 Exception Handling---712
12.10.6 Combining Calls into One Message--712
12.11 SimpleXMLRPCServer—An XML-RPC Server--714
12.11.1 A Simple Server--714
12.11.2 Alternate API Names---716
12.11.3 Dotted API Names--718
12.11.4 Arbitrary API Names---719
12.11.5 Exposing Methods of Objects---720
12.11.6 Dispatching Calls--722
12.11.7 Introspection API--724
13 EMAIL--727
13.1 smtplib—Simple Mail Transfer Protocol Client--727
13.1.1 Sending an Email Message--728
13.1.2 Authentication and Encryption---730
13.1.3 Verifying an Email Address---732
13.2 smtpd—Sample Mail Servers---734
13.2.1 Mail Server Base Class---734
13.2.2 Debugging Server--737
13.2.3 Proxy Server---737
13.3 imaplib—IMAP4 Client Library--738
13.3.1 Variations--739
13.3.2 Connecting to a Server--739

13.3.3 Example Configuration--741
13.3.4 Listing Mailboxes--741
13.3.5 Mailbox Status--744
13.3.6 Selecting a Mailbox---745
13.3.7 Searching for Messages---746
13.3.8 Search Criteria--747
13.3.9 Fetching Messages---749
13.3.10 Whole Messages--752
13.3.11 Uploading Messages--753
13.3.12 Moving and Copying Messages--755
13.3.13 Deleting Messages--756
13.4 mailbox—Manipulate Email Archives---758
13.4.1 mbox--759
13.4.2 Maildir--762
13.4.3 Other Formats---768
14 APPLICATION BUILDING BLOCKS---769
14.1 getopt—Command-Line Option Parsing---770
14.1.1 Function Arguments--771
14.1.2 Short-Form Options--771
14.1.3 Long-Form Options---772
14.1.4 A Complete Example---772
14.1.5 Abbreviating Long-Form Options---775
14.1.6 GNU-Style Option Parsing--775
14.1.7 Ending Argument Processing--777
14.2 optparse—Command-Line Option Parser--777
14.2.1 Creating an OptionParser--777
14.2.2 Short- and Long-Form Options---778
14.2.3 Comparing with getopt--779
14.2.4 Option Values---781
14.2.5 Option Actions--784
14.2.6 Help Messages---790
14.3 argparse—Command-Line Option and Argument Parsing------------------------------------795
14.3.1 Comparing with optparse--796
14.3.2 Setting Up a Parser--796
14.3.3 Defining Arguments--796
14.3.4 Parsing a Command Line---796
14.3.5 Simple Examples---797
14.3.6 Automatically Generated Options---805
14.3.7 Parser Organization--807
14.3.8 Advanced Argument Processing--815
14.4 readline—The GNU Readline Library--823
14.4.1 Configuring---823
14.4.2 Completing Text---824
14.4.3 Accessing the Completion Buffer---828
14.4.4 Input History---832
14.4.5 Hooks---834
14.5 getpass—Secure Password Prompt---836
14.5.1 Example---836
14.5.2 Using getpass without a Terminal--837
14.6 cmd—Line-Oriented Command Processors--839
14.6.1 Processing Commands---839
14.6.2 Command Arguments--840

14.6.3 Live Help--842
14.6.4 Auto-Completion---843
14.6.5 Overriding Base Class Methods--845
14.6.6 Configuring Cmd through Attributes--847
14.6.7 Running Shell Commands--848
14.6.8 Alternative Inputs---849
14.6.9 Commands from sys.argv--851
14.7 shlex—Parse Shell-Style Syntaxes---852
14.7.1 Quoted Strings--852
14.7.2 Embedded Comments--854
14.7.3 Split---855
14.7.4 Including Other Sources of Tokens---855
14.7.5 Controlling the Parser---856
14.7.6 Error Handling--858
14.7.7 POSIX vs Non-POSIX Parsing--859
14.8 ConfigParser—Work with Configuration Files---861
14.8.1 Configuration File Format---862
14.8.2 Reading Configuration Files---862
14.8.3 Accessing Configuration Settings---864
14.8.4 Modifying Settings---869
14.8.5 Saving Configuration Files--871
14.8.6 Option Search Path--872
14.8.7 Combining Values with Interpolation---875
14.9 logging—Report Status, Error, and Informational Messages--878
14.9.1 Logging in Applications vs Libraries--878
14.9.2 Logging to a File--879
14.9.3 Rotating Log Files---879
14.9.4 Verbosity Levels---880
14.9.5 Naming Logger Instances---882
14.10 fileinput—Command-Line Filter Framework---883
14.10.1 Converting M3U Files to RSS---883
14.10.2 Progress Metadata--886
14.10.3 In-Place Filtering---887
14.11 atexit—Program Shutdown Callbacks--890
14.11.1 Examples--890
14.11.2 When Are atexit Functions Not Called?---891
14.11.3 Handling Exceptions--893
14.12 sched—Timed Event Scheduler---894
14.12.1 Running Events with a Delay--895
14.12.2 Overlapping Events---896
14.12.3 Event Priorities---897
14.12.4 Canceling Events---897
15 INTERNATIONALIZATION AND LOCALIZATION---899
15.1 gettext—Message Catalogs--899
15.1.1 Translation Workflow Overview---900
15.1.2 Creating Message Catalogs from Source Code---900
15.1.3 Finding Message Catalogs at Runtime--903
15.1.4 Plural Values--905
15.1.5 Application vs Module Localization--907
15.1.6 Switching Translations--908
15.2 locale—Cultural Localization API---909
15.2.1 Probing the Current Locale--909

15.2.2 Currency--915
15.2.3 Formatting Numbers--916
15.2.4 Parsing Numbers---917
15.2.5 Dates and Times--917
16 DEVELOPER TOOLS---919
16.1 pydoc—Online Help for Modules---920
16.1.1 Plain-Text Help--920
16.1.2 HTML Help--920
16.1.3 Interactive Help---921
16.2 doctest—Testing through Documentation---921
16.2.1 Getting Started--922
16.2.2 Handling Unpredictable Output--924
16.2.3 Tracebacks---928
16.2.4 Working around Whitespace--930
16.2.5 Test Locations---936
16.2.6 External Documentation---939
16.2.7 Running Tests--942
16.2.8 Test Context--945
16.3 unittest—Automated Testing Framework--949
16.3.1 Basic Test Structure---949
16.3.2 Running Tests--949
16.3.3 Test Outcomes---950
16.3.4 Asserting Truth--952
16.3.5 Testing Equality--953
16.3.6 Almost Equal?--954
16.3.7 Testing for Exceptions---955
16.3.8 Test Fixtures--956
16.3.9 Test Suites--957
16.4 traceback—Exceptions and Stack Traces--958
16.4.1 Supporting Functions--958
16.4.2 Working with Exceptions--959
16.4.3 Working with the Stack--963
16.5 cgitb—Detailed Traceback Reports--965
16.5.1 Standard Traceback Dumps--966
16.5.2 Enabling Detailed Tracebacks---966
16.5.3 Local Variables in Tracebacks---968
16.5.4 Exception Properties---971
16.5.5 HTML Output--972
16.5.6 Logging Tracebacks--972
16.6 pdb—Interactive Debugger---975
16.6.1 Starting the Debugger---976
16.6.2 Controlling the Debugger---979
16.6.3 Breakpoints---990
16.6.4 Changing Execution Flow---1002
16.6.5 Customizing the Debugger with Aliases---1009
16.6.6 Saving Configuration Settings--1011
16.7 trace—Follow Program Flow---1012
16.7.1 Example Program---1013
16.7.2 Tracing Execution---1013
16.7.3 Code Coverage--1014
16.7.4 Calling Relationships---1017
16.7.5 Programming Interface--1018

16.7.6 Saving Result Data--1020
16.7.7 Options---1022
16.8 profile and pstats—Performance Analysis---1022
16.8.1 Running the Profiler--1023
16.8.2 Running in a Context--1026
16.8.3 pstats: Saving and Working with Statistics---1027
16.8.4 Limiting Report Contents---1028
16.8.5 Caller / Callee Graphs---1029
16.9 timeit—Time the Execution of Small Bits of Python Code----------------------------------1031
16.9.1 Module Contents--1031
16.9.2 Basic Example---1032
16.9.3 Storing Values in a Dictionary--1033
16.9.4 From the Command Line--1035
16.10 compileall—Byte-Compile Source Files--1037
16.10.1 Compiling One Directory---1037
16.10.2 Compiling sys.path--1038
16.10.3 From the Command Line---1039
16.11 pyclbr—Class Browser--1039
16.11.1 Scanning for Classes--1041
16.11.2 Scanning for Functions---1042
17 RUNTIME FEATURES--1045
17.1 site—Site-Wide Configuration---1046
17.1.1 Import Path---1046
17.1.2 User Directories---1047
17.1.3 Path Configuration Files---1049
17.1.4 Customizing Site Configuration--1051
17.1.5 Customizing User Configuration---1053
17.1.6 Disabling the site Module---1054
17.2 sys—System-Specific Configuration--1055
17.2.1 Interpreter Settings--1055
17.2.2 Runtime Environment--1062
17.2.3 Memory Management and Limits--1065
17.2.4 Exception Handling---1071
17.2.5 Low-Level Thread Support--1074
17.2.6 Modules and Imports--1080
17.2.7 Tracing a Program as It Runs---1101
17.3 os—Portable Access to Operating System Specific Features-------------------------------1108
17.3.1 Process Owner---1108
17.3.2 Process Environment---1111
17.3.3 Process Working Directory--1112
17.3.4 Pipes--1112
17.3.5 File Descriptors--1116
17.3.6 File System Permissions--1116
17.3.7 Directories--1118
17.3.8 Symbolic Links--1119
17.3.9 Walking a Directory Tree--1120
17.3.10 Running External Commands---1121
17.3.11 Creating Processes with os.fork()--1122
17.3.12 Waiting for a Child---1125
17.3.13 Spawn--1127
17.3.14 File System Permissions--1127
17.4 platform—System Version Information--1129

17.4.1 Interpreter--1129
17.4.2 Platform---1130
17.4.3 Operating System and Hardware Info--1131
17.4.4 Executable Architecture---1133
17.5 resource—System Resource Management---1134
17.5.1 Current Usage---1134
17.5.2 Resource Limits---1135
17.6 gc—Garbage Collector--1138
17.6.1 Tracing References---1138
17.6.2 Forcing Garbage Collection---1141
17.6.3 Finding References to Objects that Cannot Be Collected---------------------------------1146
17.6.4 Collection Thresholds and Generations--1148
17.6.5 Debugging--1151
17.7 sysconfig—Interpreter Compile-Time Configuration--1160
17.7.1 Configuration Variables--1160
17.7.2 Installation Paths--1163
17.7.3 Python Version and Platform--1167
18 LANGUAGE TOOLS---1169
18.1 warnings—Nonfatal Alerts--1170
18.1.1 Categories and Filtering--1170
18.1.2 Generating Warnings---1171
18.1.3 Filtering with Patterns--1172
18.1.4 Repeated Warnings---1174
18.1.5 Alternate Message Delivery Functions---1175
18.1.6 Formatting--1176
18.1.7 Stack Level in Warnings--1177
18.2 abc—Abstract Base Classes--1178
18.2.1 Why Use Abstract Base Classes?---1178
18.2.2 How Abstract Base Classes Work--1178
18.2.3 Registering a Concrete Class---1179
18.2.4 Implementation through Subclassing---1179
18.2.5 Concrete Methods in ABCs---1181
18.2.6 Abstract Properties---1182
18.3 dis—Python Bytecode Disassembler---1186
18.3.1 Basic Disassembly---1187
18.3.2 Disassembling Functions---1187
18.3.3 Classes--1189
18.3.4 Using Disassembly to Debug---1190
18.3.5 Performance Analysis of Loops--1192
18.3.6 Compiler Optimizations--1198
18.4 inspect—Inspect Live Objects--1200
18.4.1 Example Module---1200
18.4.2 Module Information---1201
18.4.3 Inspecting Modules--1203
18.4.4 Inspecting Classes---1204
18.4.5 Documentation Strings---1206
18.4.6 Retrieving Source--1207
18.4.7 Method and Function Arguments--1209
18.4.8 Class Hierarchies---1210
18.4.9 Method Resolution Order---1212
18.4.10 The Stack and Frames---1213
18.5 exceptions—Built-in Exception Classes---1216

18.5.1 Base Classes--1216
18.5.2 Raised Exceptions---1217
18.5.3 Warning Categories--1233
19 MODULES AND PACKAGES---1235
19.1 imp—Python’s Import Mechanism--1235
19.1.1 Example Package--1236
19.1.2 Module Types---1236
19.1.3 Finding Modules---1237
19.1.4 Loading Modules---1238
19.2 zipimport—Load Python Code from ZIP Archives---1240
19.2.1 Example---1240
19.2.2 Finding a Module---1241
19.2.3 Accessing Code---1242
19.2.4 Source---1243
19.2.5 Packages--1244
19.2.6 Data---1244
19.3 pkgutil—Package Utilities---1247
19.3.1 Package Import Paths--1247
19.3.2 Development Versions of Packages---1249
19.3.3 Managing Paths with PKG Files---1251
19.3.4 Nested Packages---1253
19.3.5 Package Data---1255
Index of Python Modules 1259
Index 1261
TABLES
1.1 Regular Expression Escape Codes 24
1.2 Regular Expression Anchoring Codes 27
1.3 Regular Expression Flag Abbreviations 45
2.1 Byte Order Specifiers for struct 104
6.1 Codec Error Handling Modes 292
7.1 The “project” Table 353
7.2 The “task” Table 353
7.3 CSV Dialect Parameters 415
10.1 Multiprocessing Exit Codes 537
11.1 Event Flags for poll() 604
13.1 IMAP 4 Mailbox Status Conditions 744
14.1 Flags for Variable Argument Definitions in argparse 815
14.2 Logging Levels 881
16.1 Test Case Outcomes 950
17.1 CPython Command-Line Option Flags--1057
17.2 Event Hooks for settrace()--1101
17.3 Platform Information Functions---1132
17.4 Path Names Used in sysconfig---1164
18.1 Warning Filter Actions--1171

***Think_Complexity_2012_146p

Preface v
0.1 Why I wrote this book--v
0.2 Suggestions for teachers--v
0.3 Suggestions for autodidacts---vi

1 Complexity Science--------------------1
1.1 What is this book about?--------------------1
1.2 A new kind of science--------------------2
1.3 Paradigm shift?--------------------3
1.4 The axes of scientific models--------------------4
1.5 A new kind of model--------------------5
1.6 A new kind of engineering--------------------6
1.7 A new kind of thinking--------------------7
2 Graphs--------------------9
2.1 What’s a graph?--------------------9
2.2 Representing graphs--------------------11
2.3 Random graphs--------------------14
2.4 Connected graphs--------------------15
2.5 Paul Erd˝os: peripatetic mathematician, speed freak--------------------15
2.6 Iterators--------------------16
2.7 Generators--------------------17
3 Analysis of algorithms--------------------19
3.1 Order of growth--------------------20
3.2 Analysis of basic operations--------------------22
3.3 Analysis of search algorithms--------------------23
3.4 Hashtables--------------------24
3.5 Summing lists--------------------28
3.6 pyplot--------------------30
3.7 List comprehensions--------------------31
4 Small world graphs--------------------33
4.1 Analysis of graph algorithms--------------------33
4.2 FIFO implementation--------------------34
4.3 Stanley Milgram--------------------35
4.4 Watts and Strogatz--------------------36
4.5 Dijkstra--------------------37
4.6 What kind of explanation is that?--------------------38
5 Scale-free networks--------------------41
5.1 Zipf’s Law--------------------41
5.2 Cumulative distributions--------------------42
5.3 Continuous distributions--------------------43
5.4 Pareto distributions--------------------44
5.5 Barabási and Albert--------------------46
5.6 Zipf, Pareto and power laws--------------------47
5.7 Explanatory models--------------------49
6 Cellular Automata--------------------51
6.1 Stephen Wolfram--------------------51
6.2 Implementing CAs--------------------53
6.3 CADrawer--------------------54
6.4 Classifying CAs--------------------56
Contents xi
6.5 Randomness--------------------56
6.6 Determinism--------------------58
6.7 Structures--------------------59
6.8 Universality--------------------61
6.9 Falsifiability--------------------62
6.10 What is this a model of?--------------------64
7 Game of Life--------------------65

7.1 Implementing Life--66
7.2 Life patterns--68
7.3 Conway’s conjecture--69
7.4 Realism---69
7.5 Instrumentalism---70
7.6 Turmites--71
8 Fractals---73
8.1 Fractal CAs--74
8.2 Percolation---76
9 Self-organized criticality---79
9.1 Sand piles---79
9.2 Spectral density--80
9.3 Fast Fourier Transform---82
9.4 Pink noise---83
9.5 Reductionism and Holism--84
9.6 SOC, causation and prediction---86
10 Agent-based models 89
10.1 Thomas Schelling---89
10.2 Agent-based models---90
10.3 Traffic jams--91
xii Contents
10.4 Boids--91
10.5 Prisoner’s Dilemma---93
10.6 Emergence--95
10.7 Free will--96
11 Case study: Sugarscape 99
11.1 The Original Sugarscape---99
11.2 The Occupy movement---99
11.3 A New Take on Sugarscape---100
11.4 Taxation and the Leave Behind---101
11.5 The Gini coefficient---101
11.6 Results With Taxation--102
11.7 Conclusion---103
12 Case study: Ant trails 105
12.1 Introduction--105
12.2 Model Overview--105
12.3 API design---106
12.4 Sparse matrices--108
12.5 wx---108
12.6 Applications---109
13 Case study: Directed graphs and knots 111
13.1 Directed Graphs---111
13.2 Implementation--112
13.3 Detecting knots--112
13.4 Knots in Wikipedia---114
14 Case study: The Volunteer’s Dilemma--115
14.1 The prairie dog’s dilemma---115
14.2 Analysis--116
14.3 The Norms Game---117
14.4 Results--118
14.5 Improving the chances---119
A Call for submissions---121

B Reading list---123

***Twisted_Network_Programming_Essentials_2nd_ed_2013_193p

Foreword to the First Edition ix
Preface xiii
Part I An Introduction to Twisted
1 Getting Started 3
Installing Twisted---3
Installation on Linux---3
Installation on Windows--4
Installation on OS X---5
Installing from Source---5
Required Dependencies---5
Installing Twisted from a Release Tarball--6
Installing Twisted from a Source Checkout--6
Installing Optional Dependencies from Source---6
Testing Your Installation--7
Using the Twisted Documentation---8
API Documentation--8
Subproject Documentation---8
Finding Answers to Your Questions---8
Mailing Lists---9
IRC Channels--9
Stack Overflow--9
Twisted Blogs--9
2 Building Basic Clients and Servers 11
A TCP Echo Server and Client---11
Event-Driven Programming---12
The Reactor---14
Transports---14
Protocols---15
Protocol Factories--16
Decoupling Transports and Protocols---16
A TCP Quote Server and Client---16
Protocol State Machines--19
More Practice and Next Steps---22
3 Writing Asynchronous Code with Deferreds 25
What Deferreds Do and Don’t Do---25
The Structure of a Deferred Object--26
Callback Chains and Using Deferreds in the Reactor------------------------------------28
Practice: What Do These Deferred Chains Do?--30
Exercise 1 31
Exercise 2 31
Exercise 3 31
Exercise 4 32
Exercise 5 32
Exercise 6 33
The Truth About addCallbacks---33

Exercise 7 34
Exercise 8 35
Key Facts About Deferreds--35
Summary of the Deferred API---36
More Practice and Next Steps---36
4 Web Servers 39
Responding to HTTP Requests: A Low-Level Review--39
The Structure of an HTTP Request--40
Parsing HTTP Requests--42
Handling GET Requests--43
Serving Static Content--43
Serving Dynamic Content--45
Dynamic Dispatch---46
Handling POST Requests--48
A Minimal POST Example---48
Asynchronous Responses--49
More Practice and Next Steps---51
5 Web Clients 53
Basic HTTP Resource Retrieval---53
Printing a Web Resource---53
Downloading a Web Resource--54
Agent--55
Requesting Resources with Agent--55
Retrieving Response Metadata--57
POSTing Data with Agent---58
More Practice and Next Steps---60
Part II Building Production-Grade Twisted Services
6 Deploying Twisted Applications 63
The Twisted Application Infrastructure---63
Services--64
Applications---64
TAC Files--64
twistd---65
Plugins---66
More twistd Examples---68
More Practice and Next Steps---68
Suggested Exercises--69
7 Logging 71
Basic In-Application Logging--71
twistd Logging--73
Custom Loggers--73
Key Facts and Caveats About Logging---75
8 Databases 77
Nonblocking Database Queries---77
More Practice and Next Steps---80
9 Authentication 81
The Components of Twisted Cred--81
Twisted Cred: An Example---82
Credentials Checkers--86
Authentication in Twisted Applications---89
More Practice and Next Steps---91
10 Threads and Subprocesses 93

Threads---93
Subprocesses--96
Running a Subprocess and Getting the Result---96
Custom Process Protocols---97
More Practice and Next Steps--100
11 Testing 103
Writing and Running Twisted Unit Tests with Trial---103
Testing Protocols---104
Tests and the Reactor--108
Testing Deferreds---109
Testing the Passage of Time--112
More Practice and Next Steps--115
Part III More Protocols and More Practice
12 Twisted Words 119
IRC Clients---119
IRC Servers--121
More Practice and Next Steps--124
13 Twisted Mail 125
SMTP Clients and Servers--126
The SMTP Protocol--126
Sending Emails Using SMTP--127
SMTP Servers---128
Storing Mail---130
IMAP Clients and Servers--132
IMAP Servers--133
IMAP Clients---137
POP3 Clients and Servers---139
POP3 Servers--139
More Practice and Next Steps--143
14 SSH 145
SSH Servers---145
A Basic SSH Server--145
Using Public Keys for Authentication---151
Providing an Administrative Python Shell--153
Running Commands on a Remote Server--156
SSH Clients--156
More Practice and Next Steps--159
15 The End 161
Contributing to Twisted--161
Index---163

***Using_Google_App_Engine_2009_264p

Preface xi
1 Programming on the Web---1
The Request/Response Cycle---3
What Is Google App Engine?--5
What Is a “Cloud”?---6

Why Did Google Build App Engine and Give It Away for Free?-------------------------------------7
What Is the Google Infrastructure Cloud?---8
Enter the Application Engine--9
Your Application Must Be a Good Citizen in the Google Cloud-------------------------------------10
How the Cloud Runs Your Application--11
Why You Really Want to Run in the Cloud--12
The Simplest App Engine Application---13
Summary--14
Exercises---15
2 HTML and CSS--17
A Brief History of HTML and CSS--18
HyperText Markup Language (HTML)--18
A Well-Formed HTML Document--20
Validating Your HTML---22
HyperText Links (Anchor Tags)--23
Multiple Files---25
Lists in HTML--26
Specifying Styles Using CSS--27
Styling Tags with CSS---27
Exerting More Control over Markup--30
Validating CSS--32
Tools to Help You Work with CSS---33
Building the Navigation Menu--34
The CSS Box Model---37
Adding Navigation Hints for Users--39
Summary--40
Exercises---42
3 Python--43
What Is Programming?--43
About Python---44
Installing Python--45
The Essence of Programming--46
Input, Processing, and Output--47
Conditional Steps: A Number-Guessing Web Application---48
Variables and Expressions---52
Naming Variables--53
Constants--54
Handling Text Data Using Strings---55
Using the String Library---56
Types and Conversion---57
Variables with Many Values at the Same Time: Lists---59
Repeated Code: Loops---60
Python’s Backpack: Dictionaries---63
Looping Through a Dictionary--64
Stored and Reused Code: Python Functions--65
Turning Traceback to Our Advantage Using Try and Except--68
Object-Oriented Python---70
Comments in Python--71
The Tao of Programming--71
Summary--72
Exercises---72
4 Sending Data to Your Application--75

Making an HTTP Request--75
Hacking the HTTP Protocol---77
The HTTP Protocol Standards---79
Watching HTTP in Action--79
HTML: Building a Form---81
Sending Form Data to the Server Using POST--82
Handling Form Data in the App Engine Server---83
Sending Form Data to the Server Using GET--87
App Engine Log--88
Looking at All the Data Available on an HTTP Request--89
Advanced: Multipart POST Data with File Upload--93
Summary--96
Exercises--96
5 The App Engine webapp Framework---99
A Trivial App Engine Application---99
An Equivalent Trivial webapp Application---100
The Handler Callback Pattern---102
Looking at the Handler Code--102
What Is “self” and Why Is It There?---103
Number Guessing Using the webapp Framework---105
Web Server Logs--109
Summary---112
Exercises---112
6 Templates--113
Template Syntax---113
Using the Templates from Python--114
The Number-Guessing Game Using Templates---116
Abstraction and Separation of Concerns: “Model View Controller”-------------------------118
Building a Multiscreen Application--118
Static Files in App Engine--120
Referencing Static Files---121
Generalizing Template Lookup with Multiple Templates--121
Extending Base Templates---123
Conditional Processing Templates--125
Replacing More Than One Block in the Base Template-------------------------------------129
Extending Our Application--131
Syntax Errors--139
More on Templates---141
Summary---141
Exercises---141
7 Cookies and Sessions--143
HTTP Cookies---143
Logging In and Logging Out Using Sessions---145
Installing the Simple Session Utility---146
Using the Session to Mark Login and Logout---146
Changing the User Interface for a Logged-In User--148
Summary---150
Exercises---150
8 App Engine Datastore--153
The Model-View-Controller Pattern--153
Defining Models---154
Adding a New Account Form--155

Adding a User Account--156
Looking Through the Datastore--159
Logging In and Checking Accounts and Passwords--160
Retrieving and Displaying Many Objects--161
Terminology: Different Types of Keys---164
References Between Data Models---165
Interactions Between Models--166
Putting the Primary Key into Session--168
Adding the Chat Feature---169
Summary--174
Exercises---174
9 JavaScript, jQuery, and AJAX--177
jQuery---178
Create a View of the Chat Messages in HTML---178
Asynchronously Updating a <div> Using AJAX---181
Summary--183
Exercises---183
10 Running Your Application on the Google Infrastructure--185
Application Engine Accounts---185
Creating an Application on App Engine--186
Uploading Your Application--187
Testing Your Application--187
Logs, Statistics, and Other Features---188
Uploading New Versions--190
Collaboration on Applications---191
Summary--192
Exercises---192
11 Memory Cache---193
The Rules of Memory Cache--194
Using the Memory Cache--195
Using the Application Console to Examine Memory Cache--196
Using a Memory Cache in Your Application---197
Building a Session Store Using Memory Cache---198
Sessions and Memory Cache---202
Summary--204
Exercises---204
A Installing and Running Google App Engine on Windows XP------------------------------------205
B Installing and Running Google App Engine on Windows Vista---------------------------------213
C Installing and Running Google App Engine on a Macintosh System----------------------------219
D Installing and Running Google App Engine on a Linux System---------------------------------227
Index---235

***Building_Web_Applications_with_Flask_2015_160p

Preface---v
Chapter 1: Flask in a Flask, I Mean, Book--1
An introduction to Flask and its features---1
Summary---3
Chapter 2: First App, How Hard Could it Be?---5
Hello World---5
Prerequisites and tools---5
Setting up a virtual environment---6
Understanding the "Hello World" app---6
Serving HTML pages--9
Summary--10
Chapter 3: Man, Do I Like Templates!---11
What is Jinja2 and how is it coupled with Flask?--11
What can you do with Jinja2?--12
Control structures---15
Macros---20
Extensions---22
Filters--23
Messing with the template context--25
Summary--26
Chapter 4: Please Fill in This Form, Madam---27
HTML forms for the faint of heart--27
Handling forms---29
WTForms and you--30
Flask-WTF--34
Integration with WTForms--34
Securing forms with a CSRF token--36
Challenges---37
Summary--37
Chapter 5: Where Do You Store Your Stuff?--39
SQLAlchemy--40
Concepts---40
Hands on---41
Flask-SQLAlchemy--48
MongoDB---50
MongoEngine---52
Flask-MongoEngine---54
Relational versus NoSQL--56
Summary--56
Chapter 6: But I Wanna REST Mom, Now!---57
Beyond GET--63
Flask-Restless--66
Summary--68
Chapter 7: If Ain't Tested, It Ain't Game, Bro!--69
What kinds of test are there?--70
Unit testing---70
Behavior testing--74
Flask-testing--76
LiveServer--77
Extra assertions---80

JSON handle--80
Fixtures--81
Extra – integration testing--84
Summary---84
Chapter 8: Tips and Tricks or Flask Wizardry 101---85
Overengineering--86
Premature optimization---86
Blueprints 101---86
Oh God, please tell me you have the logs…---94
Debugging, DebugToolbar, and happiness---96
Flask-DebugToolbar--97
Sessions or storing user data between requests--99
Exercise--101
Summary---102
Chapter 9: Extensions, How I Love Thee--103
How to configure extensions--103
Flask-Principal and Flask-Login (aka Batman and Robin)---105
Admin like a boss--114
Custom pages---119
Summary---120
Chapter 10: What Now?--121
You deploy better than my ex--121
Placing your code in a server---122
Setting up your database--122
Setting up the web server---125
StackOverflow---129
Structuring your projects---129
Summary---130
Postscript---130
Index---131

***Flask_Framework_Cookbook_2014_258p

Preface---1
Chapter 1: Flask Configurations--- 7
Introduction---7
Environment setup with virtualenv--8
Handling basic configurations--11
Class-based settings--12
Organization of static files---14
Being deployment specific with instance folders---15
Composition of views and models--16
Creating a modular web app with blueprints---19
Making a Flask app installable using setuptools---21
Chapter 2: Templating with Jinja2--23
Introduction---23
Bootstrap layout---24
Block composition and layout inheritance---26
Creating a custom context processor--32
Creating a custom Jinja2 filter--33
Creating a custom macro for forms---35

Advanced date and time formatting--36
Chapter 3: Data Modeling in Flask--39
Introduction--39
Creating a SQLAlchemy DB instance--40
Creating a basic product model--42
Creating a relational category model--46
Database migration using Alembic and Flask-Migrate--49
Model data indexing with Redis--52
Opting the NoSQL way with MongoDB--53
Chapter 4: Working with Views--57
Introduction--57
Writing function-based views and URL routes--58
Class-based views--60
URL routing and product-based pagination--62
Rendering to templates--64
Dealing with XHR requests--68
Decorator to handle requests beautifully--71
Creating custom 404 and 500 handlers--72
Flashing messages for better user feedback--74
SQL-based searching--77
Chapter 5: Webforms with WTForms--79
Introduction--79
SQLAlchemy model data as form representation--80
Validating fields on the server side--83
Creating a common forms set--86
Creating custom fields and validation--88
Creating a custom widget--90
Uploading files via forms--92
Cross-site Request Forgery protection--96
Chapter 6: Authenticating in Flask--99
Introduction--99
Simple session-based authentication--100
Authenticating using the Flask-Login extension--107
Using OpenID for authentication--110
Using Facebook for authentication--115
Using Google for authentication--119
Using Twitter for authentication--122
Chapter 7: RESTful API Building--125
Introduction--126
Creating a class-based REST interface--126
Creating an extension-based REST interface--128
Creating a SQLAlchemy-independent REST API--131
A complete REST API example--133
Chapter 8: Admin Interface for Flask Apps--137
Introduction--137
Creating a simple CRUD interface--138
Using the Flask-Admin extension--143
Registering models with Flask-Admin--146
Creating custom forms and actions--148
WYSIWYG for textarea integration--151
Creating user roles--153
Chapter 9: Internationalization and Localization--159

Introduction--159
Adding a new language---160
Lazy evaluation and the gettext/ngettext functions--164
Global language-switching action--166
Chapter 10: Debugging, Error Handling, and Testing--169
Introduction--170
Setting up basic file logging--170
Sending e-mails on the occurrence of errors--173
Using Sentry to monitor exceptions--174
Debugging with pdb--178
Creating our first simple test--179
Writing more tests for views and logic--181
Nose library integration--185
Using mocking to avoid real API access--187
Determining test coverage---190
Using profiling to find bottlenecks--192
Chapter 11: Deployment and Post Deployment--195
Introduction--196
Deploying with Apache--196
Deploying with uWSGI and Nginx---199
Deploying with Gunicorn and Supervisor---202
Deploying with Tornado--204
Using Fabric for deployment---205
S3 storage for file uploads--207
Deploying with Heroku--209
Deploying with AWS Elastic Beanstalk---212
Application monitoring with Pingdom---215
Application performance management and monitoring with New Relic--------------------------217
Chapter 12: Other Tips and Tricks---221
Introduction--221
Full-text search with Whoosh---222
Full-text search with Elasticsearch--224
Working with signals---226
Using caching with your application--228
E-mail support for Flask applications---230
Understanding asynchronous operations---233
Working with Celery---234
Index--237

***Flask_Web_Development_2014_258p

Preface. xi
Part I. Introduction to Flask
1. Installation--3
Using VirtualEnvironments---4
Installing Python Packages with pip---6

2. Basic Application Structure. 7
Initialization--7

Routes and View Functions--8
Server Startup---9
A Complete Application---9
The Request-Response Cycle--12
Application and Request Contexts---12
Request Dispatching--14
Request Hooks---14
Responses--15
Flask Extensions--16
Command-Line Options with Flask-Script--17

3. Templates. 21
The Jinja2 Template Engine---22
Rendering Templates---22
Variables--23
Control Structures---24
Twitter Bootstrap Integration with Flask-Bootstrap----------------------------------26
Custom Error Pages---29
Links---31
Static Files---32
Localization of Dates and Times with Flask-Moment---------------------------------33

4. Web Forms37
Cross-Site Request Forgery (CSRF) Protection-------------------------------------37
Form Classes--38
HTML Rendering of Forms--40
Form Handling in View Functions---41
Redirects and User Sessions--44
Message Flashing---46

5. Databases. 49
SQL Databases---49
NoSQL Databases---50
SQL or NoSQL?--51
Python Database Frameworks---51
Database Management with Flask-SQLAlchemy---------------------------------------52
Model Definition--54
Relationships---56
Database Operations--57
Creating the Tables---58
Inserting Rows---58
Modifying Rows--60
Deleting Rows--60
Querying Rows---60
Database Use in View Functions--62
Integration with the Python Shell---63
Database Migrations with Flask-Migrate--64
Creating a Migration Repository---64
Creating a Migration Script--65
Upgrading the Database--66

6. Email. 69

Email Support with Flask-Mail--69
Sending Email from the Python Shell--70
Integrating Emails with the Application---71
Sending Asynchronous Email---72

7. Large Application Structure75
Project Structure---75
Configuration Options---76
Application Package---78
Using an Application Factory--78
Implementing Application Functionality in a Blueprint-------------------------------------79
Launch Script--81
Requirements File--82
Unit Tests---83
Database Setup---85

Part II. Example: A Social Blogging Application
8. User Authentication89
Authentication Extensions for Flask--89
Password Security--90
Hashing Passwords with Werkzeug--90
Creating an Authentication Blueprint---92
User Authentication with Flask-Login--94
Preparing the User Model for Logins--94
Protecting Routes---95
Adding a Login Form---96
Signing Users In---97
Signing Users Out--99
Testing Logins---99
New User Registration---100
Adding a User Registration Form--100
Registering New Users---102
Account Confirmation--103
Generating Confirmation Tokens with itsdangerous--------------------------------------103
Sending Confirmation Emails---105
Account Management--109

9. User Roles111
Database Representation of Roles---111
Role Assignment---113
Role Verification---114

10. User Profiles119
Profile Information--119
User Profile Page--120
Profile Editor--122
User-Level Profile Editor---122
Administrator-Level Profile Editor---124

User Avatars---127

11. Blog Posts131

Blog Post Submission and Display--131
Blog Posts on Profile Pages---134
Paginating Long Blog Post Lists---135
Creating Fake Blog Post Data---135
Rendering Data on Pages--137
Adding a Pagination Widget---138
Rich-Text Posts with Markdown and Flask-PageDown---------------------------------------141
Using Flask-PageDown--141
Handling Rich Text on the Server--143
Permanent Links to Blog Posts---145
Blog Post Editor---146

12. Followers149
Database Relationships Revisited--149
Many-to-Many Relationships---150
Self-Referential Relationships--151
Advanced Many-to-Many Relationships--152
Followers on the Profile Page---155
Query Followed Posts Using a Database Join---158
Show Followed Posts on the Home Page---160

13. User Comments. 165
Database Representation of Comments---165
Comment Submission and Display---167
Comment Moderation--169

14. Application Programming Interfaces 175
Introduction to REST---175
Resources Are Everything--176
Request Methods--177
Request and Response Bodies--177
Versioning--178
RESTful Web Services with Flask--179
Creating an API Blueprint--179
Error Handling---180
User Authentication with Flask-HTTPAuth--181
Token-Based Authentication--184
Serializing Resources to and from JSON---186
Implementing Resource Endpoints---188

Pagination of Large Resource Collections---191
Testing Web Services with HTTPie--192

Part III. The Last Mile
15. Testing197
Obtaining Code Coverage Reports---197
The Flask Test Client---200
Testing Web Applications---200
Testing Web Services---204
End-to-End Testing with Selenium---205
Is It Worth It?---209

16. Performance211

Logging Slow Database Performance--211

Source Code Profiling--213

17. Deployment215

Deployment Workflow--215

Logging of Errors During Production---216

Cloud Deployment--217

The Heroku Platform--218

Preparing the Application---218

Testing with Foreman--222

Enabling Secure HTTP with Flask-SSLify--223

Deploying with git push---225

Reviewing Logs--226

Deploying an Upgrade--227

Traditional Hosting---227

Server Setup--227

Importing Environment Variables--228

Setting Up Logging--228

18. Additional Resources231

Using an Integrated Development Environment (IDE)--231

Finding Flask Extensions---232

Getting Involved with Flask---232

Index. 235