

SIXTH EDITION

Learning Python

Powerful Object-Oriented Programming

Mark Lutz

O'REILLY®

Table of Contents

Preface.....	xxix
--------------	------

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?	6
Who Uses Python Today?	7
What Can I Do with Python?	8
Systems Programming	8
GUIs and UIs	8
Internet and Web Scripting	8
Component Integration	9
Database Access	9
Rapid Prototyping	9
Numeric and Scientific Programming	10
And More: AI, Games, Images, QA, Excel, Apps...	10
What Are Python’s Technical Strengths?	10
It’s Object-Oriented and Functional	11
It’s Free and Open	11
It’s Portable	11
It’s Powerful	12
It’s Mixable	13
It’s Relatively Easy to Use	13
It’s Relatively Easy to Learn	13
Chapter Summary	14

Test Your Knowledge: Quiz	14
Test Your Knowledge: Answers	14
2. How Python Runs Programs.....	17
Introducing the Python Interpreter	17
Program Execution	18
The Programmer's View	18
Python's View	18
Execution-Model Variations	21
Python Implementation Alternatives	22
Standalone Executables	24
Future Possibilities	25
Chapter Summary	25
Test Your Knowledge: Quiz	25
Test Your Knowledge: Answers	26
3. How You Run Programs.....	27
Installing Python	27
Interactive Code	28
Starting an Interactive REPL	28
Where to Run: Code Folders	29
What Not to Type: Prompts and Comments	30
Other Python REPLs	30
Running Code Interactively	31
Why the Interactive Prompt?	32
Program Files	34
A First Script	34
Running Files with Command Lines	35
Command-Line Usage Variations	36
Other Ways to Run Files	37
Clicking and Tapping File Icons	37
The IDLE Graphical User Interface	38
Other IDEs for Python	39
Smartphone Apps	39
WebAssembly for Browsers	40
Jupyter Notebooks for Science	40
Ahead-of-Time Compilers for Speed	40
Running Code in Code	41
Other Launch Options	45
Which Option Should I Use?	46
Chapter Summary	46
Test Your Knowledge: Quiz	46
Test Your Knowledge: Answers	47
Test Your Knowledge: Part I Exercises	48

Part II. Objects and Operations

4. Introducing Python Objects.....	53
The Python Conceptual Hierarchy	53
Why Use Built-in Objects?	54
Python's Core Object Types	55
Numbers	56
Strings	57
Sequence Operations	58
Immutability	59
Type-Specific Methods	60
Getting Help	62
Other Ways to Code Strings	63
Unicode Strings	64
Lists	65
Sequence Operations	65
Type-Specific Operations	65
Bounds Checking	66
Nesting	66
Comprehensions	67
Dictionaries	69
Mapping Operations	69
Nesting Revisited	70
Missing Keys: if Tests	71
Item Iteration: for Loops	73
Tuples	74
Why Tuples?	75
Files	75
Unicode and Byte Files	76
Other File-Like Tools	77
Other Object Types	77
Sets	77
Booleans and None	78
Types	78
Type Hinting	79
User-Defined Objects	80
And Everything Else	80
Chapter Summary	81
Test Your Knowledge: Quiz	81
Test Your Knowledge: Answers	81
5. Numbers and Expressions.....	83
Numeric Object Basics	83
Numeric Literals	84

Built-in Numeric Tools	85
Python Expression Operators	85
Mixed Operators: Precedence	87
Parentheses Group Subexpressions	88
Mixed Types Are Converted Up	88
Preview: Operator Overloading and Polymorphism	89
Numbers in Action	89
Variables and Basic Expressions	89
Numeric Display Formats	91
Comparison Operators	92
Division Operators	94
Integer Precision	96
Complex Numbers	96
Hex, Octal, and Binary	96
Bitwise Operations	98
Underscore Separators in Numbers	99
Other Built-in Numeric Tools	101
Other Numeric Objects	103
Decimal Objects	103
Fraction Objects	104
Set Objects	106
Boolean Objects	112
Numeric Extensions	113
Chapter Summary	113
Test Your Knowledge: Quiz	113
Test Your Knowledge: Answers	114
6. The Dynamic Typing Interlude.....	115
The Case of the Missing Declaration Statements	115
Variables, Objects, and References	116
Types Live with Objects, Not Variables	118
Objects Are Garbage-Collected	118
Shared References	120
Shared References and In-Place Changes	121
Shared References and Equality	123
Dynamic Typing Is Everywhere	124
Type Hinting: Optional, Unused, and Why?	125
Chapter Summary	126
Test Your Knowledge: Quiz	126
Test Your Knowledge: Answers	127
7. String Fundamentals.....	129
String Object Basics	129
String Literals	131

Single and Double Quotes Are the Same	131
Escape Sequences Are Special Characters	132
Raw Strings Suppress Escapes	136
Triple Quotes and Multiline Strings	137
Strings in Action	138
Basic Operations	138
Indexing and Slicing	140
String Conversion Tools	143
“Changing” Strings Part 1: Sequence Operations	145
String Methods	146
Method Call Syntax	146
All String Methods (Today)	147
“Changing” Strings, Part 2: String Methods	148
More String Methods: Parsing Text	150
Other Common String Methods	151
String Formatting: The Triathlon	151
String-Formatting Options	152
The String-Formatting Expression	153
The String-Formatting Method	157
The F-String Formatting Literal	162
And the Winner Is...	167
General Type Categories	168
Types Share Operation Sets by Categories	168
Mutable Types Can Be Changed in Place	169
Chapter Summary	170
Test Your Knowledge: Quiz	170
Test Your Knowledge: Answers	170
8. Lists and Dictionaries.....	173
Lists	173
Lists in Action	175
Basic List Operations	175
Indexing and Slicing	176
Changing Lists in Place	177
More List Methods	180
Iteration, Comprehensions, and Unpacking	182
Other List Operations	184
Dictionaries	184
Dictionaries in Action	186
Basic Dictionary Operations	187
Changing Dictionaries in Place	187
More Dictionary Methods	188
Other Dictionary Makers	189
Dictionary Comprehensions	191

Key Insertion Ordering	192
Dictionary “Union” Operator	193
Intermission: Books Database	194
Dictionary Usage Tips	196
Chapter Summary	203
Test Your Knowledge: Quiz	203
Test Your Knowledge: Answers	203
9. Tuples, Files, and Everything Else.....	207
Tuples	207
Tuples in Action	209
Why Lists and Tuples?	211
Records Revisited: Named Tuples	212
Files	213
Opening Files	214
Using Files	214
Files in Action	216
Text and Binary Files: The Short Story	217
Storing Objects with Conversions	218
Storing Objects with pickle	220
Storing Objects with JSON	221
Storing Objects with Other Tools	222
File Context Managers	223
Other File Tools	223
Core Types Review and Summary	224
Object Flexibility	225
References Versus Copies	226
Comparisons, Equality, and Truth	228
The Meaning of True and False in Python	231
Python’s Type Hierarchies	233
Type Objects	233
Other Types in Python	234
Built-in Type Gotchas	235
Assignment Creates References, Not Copies	235
Repetition Adds One Level Deep	236
Beware of Cyclic Data Structures	237
Immutable Types Can’t Be Changed in Place	237
Chapter Summary	237
Test Your Knowledge: Quiz	238
Test Your Knowledge: Answers	238
Test Your Knowledge: Part II Exercises	239

Part III. Statements and Syntax

10. Introducing Python Statements.....	245
The Python Conceptual Hierarchy Revisited	245
Python's Statements	246
A Tale of Two ifs	248
What Python Adds	248
What Python Removes	248
Why Indentation Syntax?	250
A Few Special Cases	252
A Quick Example: Interactive Loops	254
A Simple Interactive Loop	254
Doing Math on User Inputs	255
Handling Errors by Testing Inputs	256
Handling Errors with try Statements	257
Supporting Floating-Point Numbers	258
Nesting Code Three Levels Deep	259
Chapter Summary	259
Test Your Knowledge: Quiz	260
Test Your Knowledge: Answers	260
11. Assignments, Expressions, and Prints.....	261
Assignments	261
Assignment Syntax Forms	262
Basic Assignments	263
Sequence Assignments	264
Extended-Unpacking Assignments	266
Multiple-Target Assignments	271
Augmented Assignments	272
Named Assignment Expressions	274
Variable Name Rules	277
Expression Statements	280
Expression Statements and In-Place Changes	281
Print Operations	282
The print Function	282
Print Stream Redirection	285
Chapter Summary	288
Test Your Knowledge: Quiz	288
Test Your Knowledge: Answers	288
12. if and match Selections.....	291
if Statements	291
General Format	291
Basic Examples	292

Multiple-Choice Selections	293
match Statements	295
Basic match Usage	295
Advanced match Usage	297
Python Syntax Revisited	299
Block Delimiters: Indentation Rules	300
Statement Delimiters: Lines and Continuations	302
Special Syntax Cases in Action	303
Truth Values Revisited	304
The if/else Ternary Expression	306
Chapter Summary	307
Test Your Knowledge: Quiz	307
Test Your Knowledge: Answers	308
13. while and for Loops.....	311
while Loops	311
General Format	311
Examples	312
break, continue, pass, and the Loop else	313
General Loop Format	313
pass	313
continue	314
break	315
Loop else	316
for Loops	318
General Format	318
Examples	318
Loop Coding Techniques	323
Counter Loops: range	324
Sequence Scans: while, range, and for	325
Sequence Shufflers: range and len	326
Skipping Items: range and Slices	327
Changing Lists: range and Comprehensions	327
Parallel Traversals: zip	328
Offsets and Items: enumerate	331
Chapter Summary	332
Test Your Knowledge: Quiz	332
Test Your Knowledge: Answers	332
14. Iterations and Comprehensions.....	335
Iterations	336
The Iteration Protocol	337
Other Built-in Iterables	342
Comprehensions	347

List Comprehension Basics	347
List Comprehensions and Files	348
Extended List Comprehension Syntax	349
Comprehensions Cliff-Hanger	351
Iteration Tools	351
Other Iteration Topics	355
Chapter Summary	355
Test Your Knowledge: Quiz	356
Test Your Knowledge: Answers	356
15. The Documentation Interlude.....	357
Python Documentation Sources	357
# Comments	358
The dir Function	358
Docstrings and <code>__doc__</code>	360
Pydoc: The help Function	363
Pydoc: HTML Reports	366
Beyond Docstrings: Sphinx	370
The Standard Manuals	370
Web Resources	371
Common Coding Gotchas	372
Chapter Summary	373
Test Your Knowledge: Quiz	373
Test Your Knowledge: Answers	374
Test Your Knowledge: Part III Exercises	374

Part IV. Functions and Generators

16. Function Basics.....	379
Why Use Functions?	380
Function Coding Overview	381
Basic Function Tools	381
Advanced Function Tools	382
General Function Concepts	382
def Statements	383
return Statements	383
def Executes at Runtime	384
lambda Makes Anonymous Functions	384
A First Example: Definitions and Calls	385
Definition	385
Calls	386
Polymorphism in Python	386
A Second Example: Intersecting Sequences	387

Definition	387
Calls	388
Polymorphism Revisited	389
Segue: Local Variables	389
Chapter Summary	390
Test Your Knowledge: Quiz	390
Test Your Knowledge: Answers	390
17. Scopes.....	393
Python Scopes Basics	393
Scopes Overview	394
Name Resolution: The LEGB Rule	396
Scopes Examples	398
The Built-in Scope	399
The global Statement	401
Program Design: Minimize Global Variables	402
Program Design: Minimize Cross-File Changes	404
Other Ways to Access Globals	405
Nested Functions and Scopes	406
Nested Scopes Overview	406
Nested Scopes Examples	406
Closures and Factory Functions	407
Arbitrary Scope Nesting	409
The nonlocal Statement	409
nonlocal Basics	410
nonlocal in Action	410
nonlocal Boundary Cases	411
State-Retention Options	412
Nonlocals: Changeable, Per-Call, LEGB	413
Globals: Changeable but Shared	413
Function Attributes: Changeable, Per-Call, Explicit	414
Classes: Changeable, Per-Call, OOP	416
And the Winner Is...	416
Scopes and Argument Defaults	416
Loops Require Defaults, Not Scopes	418
Chapter Summary	419
Test Your Knowledge: Quiz	420
Test Your Knowledge: Answers	421
18. Arguments.....	423
Argument-Passing Basics	423
Arguments and Shared References	424
Avoiding Mutable Argument Changes	426
Simulating Output Parameters and Multiple Results	427

Special Argument-Matching Modes	428
Argument Matching Overview	428
Argument Matching Syntax	429
Argument Passing Details	430
Keyword and Default Examples	431
Arbitrary Arguments Examples	433
Keyword-Only Arguments	437
Positional-Only Arguments	439
Argument Ordering: The Gritty Details	440
Definition Ordering	440
Calls Ordering	442
Example: The min Wakeup Call	443
Full Credit	443
Bonus Points	445
The Punch Line	446
Example: Generalized Set Functions	446
Testing the Code	447
Example: Rolling Your Own Print	448
Using Keyword-Only Arguments	449
Chapter Summary	451
Test Your Knowledge: Quiz	451
Test Your Knowledge: Answers	452
19. Function Odds and Ends.	455
Function Design Concepts	455
Recursive Functions	457
Summation with Recursion	457
Coding Alternatives	458
Loop Statements Versus Recursion	459
Handling Arbitrary Structures	460
Function Tools: Attributes, Annotations, Etc.	464
The First-Class Object Model	465
Function Introspection	466
Function Attributes	467
Function Annotations and Decorations	468
Anonymous Functions: lambda	470
lambda Basics	471
Why Use lambda?	472
How (Not) to Obfuscate Your Python Code	473
Scopes: lambdas Can Be Nested Too	475
Functional Programming Tools	475
Mapping Functions over Iterables: map	476
Selecting Items in Iterables: filter	477
Combining Items in Iterables: reduce	478

Chapter Summary	479
Test Your Knowledge: Quiz	479
Test Your Knowledge: Answers	479
20. Comprehensions and Generations.....	483
Comprehensions: The Final Act	483
List Comprehensions Review	484
Formal Comprehension Syntax	485
Example: List Comprehensions and Matrixes	487
Generator Functions and Expressions	490
Generator Functions: yield Versus return	490
Generator Expressions: Iterables Meet Comprehensions	496
Generator Functions Versus Generator Expressions	500
Generator Odds and Ends	501
Example: Shuffling Sequences	506
Scrambling Sequences	506
Permutating Sequences	509
Example: Emulating zip and map	513
Coding Your Own map	514
Coding Your Own zip and 2.X map	515
Asynchronous Functions: The Short Story	517
Async Basics	518
The Async Wrap-Up	524
Chapter Summary	525
Test Your Knowledge: Quiz	525
Test Your Knowledge: Answers	525
21. The Benchmarking Interlude.....	529
Benchmarking with Homegrown Tools	529
Timer Module: Take 1	530
Timer Module: Take 2	531
Timing Runner and Script	533
Iteration Results	535
More Module Mods	538
Benchmarking with Python's timeit	540
Basic timeit Usage	540
Automating timeit Benchmarking	544
Function Gotchas	549
Local Names Are Detected Statically	549
Defaults and Mutable Objects	550
Functions Without returns	552
Miscellaneous Function Gotchas	552
Chapter Summary	553
Test Your Knowledge: Quiz	553

Part V. Modules and Packages

22. Modules: The Big Picture.....	559
Module Essentials	559
Why Use Modules?	560
Python Program Architecture	560
How to Structure a Program	561
Imports and Attributes	561
Standard-Library Modules	563
How Imports Work	563
Step 1: Find It	564
Step 2: Compile It (Maybe)	564
Step 3: Run It	567
The Module Search Path	567
Search-Path Components	567
Configuring the Search Path	570
The sys.path List	570
Module File Selection	571
Path Outliers: Standalones and Packages	572
Chapter Summary	573
Test Your Knowledge: Quiz	573
Test Your Knowledge: Answers	573
23. Module Coding Basics.....	575
Creating Modules	575
Module Filenames	575
Other Kinds of Modules	576
Using Modules	576
The import Statement	576
The from Statement	577
The from * Statement	577
Imports Happen Only Once	578
Imports Are Runtime Assignments	579
import and from Equivalence	580
Potential Pitfalls of the from Statement	580
Module Namespaces	582
How Files Generate Namespaces	582
Namespace Dictionaries: __dict__	583
Attribute Name Qualification	584
Imports Versus Scopes	585

Namespace Nesting	586
Reloading Modules	587
reload Basics	588
reload Example	588
reload Odds and Ends	590
Chapter Summary	590
Test Your Knowledge: Quiz	590
Test Your Knowledge: Answers	591
24. Module Packages.....	593
Using Packages	593
Package Imports	594
Packages and the Module Search Path	594
Creating Packages	595
Basic Package Structure	595
Package <code>__init__.py</code> Files	597
Package <code>__main__.py</code> Files	599
Why Packages?	600
A Tale of Two Systems	600
The Roles of <code>__init__.py</code> Files	603
Package-Relative Imports	604
Relative and Absolute Imports	604
Relative-Import Rationales and Trade-Offs	605
Package-Relative Imports in Action	605
Namespace Packages	609
Python Import Models	609
Namespace-Package Rationales	610
The Module Search Algorithm	610
Namespace Packages in Action	611
Chapter Summary	613
Test Your Knowledge: Quiz	614
Test Your Knowledge: Answers	614
25. Module Odds and Ends.....	617
Module Design Concepts	617
Data Hiding in Modules	618
Minimizing from * Damage: <code>__X__</code> and <code>__all__</code>	619
Managing Attribute Access: <code>__getattr__</code> and <code>__dir__</code>	620
Enabling Language Changes: <code>__future__</code>	622
Dual-Usage Modes: <code>__name__</code> and <code>__main__</code>	622
Example: Unit Tests with <code>__name__</code>	623
The <code>as</code> Extension for <code>import</code> and <code>from</code>	625
Module Introspection	626
Example: Listing Modules with <code>__dict__</code>	627

Importing Modules by Name String	629
Running Code Strings	629
Direct Calls: Two Options	630
Example: Transitive Module Reloads	630
Module Gotchas	638
Module Name Clashes: Package and Package-Relative Imports	638
Statement Order Matters in Top-Level Code	638
from Copies Names but Doesn't Link	639
from * Can Obscure the Meaning of Variables	640
reload May Not Impact from Imports	640
reload, from, and Interactive Testing	641
Recursive from Imports May Not Work	642
Chapter Summary	643
Test Your Knowledge: Quiz	643
Test Your Knowledge: Answers	643
Test Your Knowledge: Part V Exercises	644

Part VI. Classes and OOP

26. OOP: The Big Picture	649
Why Use Classes?	649
OOP from 30,000 Feet	651
Attribute Inheritance Search	651
Classes and Instances	653
Method Calls	653
Coding Class Trees	654
Operator Overloading	656
OOP Is About Code Reuse	656
Chapter Summary	659
Test Your Knowledge: Quiz	659
Test Your Knowledge: Answers	659
27. Class Coding Basics	661
Classes Generate Multiple Instance Objects	661
Class Objects Provide Default Behavior	662
Instance Objects Are Concrete Items	662
A First Example	662
Classes Are Customized by Inheritance	664
A Second Example	665
Classes Are Attributes in Modules	666
Classes Can Intercept Python Operators	667
A Third Example	669
The World's Simplest Python Class	671

Classes: Under the Hood	671
Records Revisited: Classes Versus Dictionaries	673
Chapter Summary	675
Test Your Knowledge: Quiz	675
Test Your Knowledge: Answers	676
28. A More Realistic Example.....	679
Step 1: Making Instances	679
Coding Constructors	680
Testing as You Go	681
Using Code Two Ways	682
Step 2: Adding Behavior Methods	683
Coding Methods	684
Step 3: Operator Overloading	686
Providing Print Displays	686
Step 4: Customizing Behavior by Subclassing	688
Coding Subclasses	688
Augmenting Methods: The Bad Way	689
Augmenting Methods: The Good Way	689
Polymorphism in Action	691
Inherit, Customize, and Extend	692
OOP: The Big Idea	693
Step 5: Customizing Constructors, Too	693
OOP Is Simpler Than You May Think	695
Other Ways to Combine Classes: Composites	695
Step 6: Using Introspection Tools	698
Special Class Attributes	699
A Generic Display Tool	700
Instance Versus Class Attributes	701
Name Considerations in Tool Classes	702
Our Classes' Final Form	702
Step 7 (Final): Storing Objects in a Database	704
Pickles and Shelves	704
Storing Objects on a shelf Database	705
Exploring Shelves Interactively	706
Updating Objects on a Shelf	708
Future Directions	709
Chapter Summary	709
Test Your Knowledge: Quiz	710
Test Your Knowledge: Answers	710
29. Class Coding Details.....	713
The class Statement	713
General Syntax and Usage	713

Example: Class Attributes	714
Methods	716
Method Example	716
Other Method-Call Possibilities	717
Inheritance	718
Attribute Tree Construction	718
Inheritance Fine Print	719
Specializing Inherited Methods	719
Class Interface Techniques	721
Abstract Superclasses	722
Namespaces: The Conclusion	724
Simple Names: Global Unless Assigned	725
Attribute Names: Object Namespaces	725
The “Zen” of Namespaces: Assignments Classify Names	726
Nested Classes: The LEGB Scopes Rule Revisited	728
Namespace Dictionaries: Review	730
Namespace Links: A Tree Climber	732
Documentation Strings Revisited	734
Classes Versus Modules	735
Chapter Summary	736
Test Your Knowledge: Quiz	736
Test Your Knowledge: Answers	736
30. Operator Overloading.....	739
The Basics	739
Constructors and Expressions: <code>__init__</code> and <code>__sub__</code>	740
Common Operator-Overloading Methods	740
Indexing and Slicing: <code>__getitem__</code> and <code>__setitem__</code>	742
Intercepting Slices	743
Intercepting Item Assignments	744
But <code>__index__</code> Means As-Integer	745
Index Iteration: <code>__getitem__</code>	745
Iterable Objects: <code>__iter__</code> and <code>__next__</code>	746
User-Defined Iterables	747
Multiple Iterators on One Object	749
Coding Alternative: <code>__iter__</code> Plus <code>yield</code>	752
Membership: <code>__contains__</code> , <code>__iter__</code> , and <code>__getitem__</code>	756
Attribute Access: <code>__getattr__</code> and <code>__setattr__</code>	759
Attribute Reference	759
Attribute Assignment and Deletion	760
Other Attribute-Management Tools	761
Emulating Privacy for Instance Attributes: Part 1	761
String Representation: <code>__repr__</code> and <code>__str__</code>	762
Why Two Display Methods?	763

Display Usage Notes	764
Right-Side and In-Place Ops: <code>__radd__</code> and <code>__iadd__</code>	765
Right-Side Addition	766
In-Place Addition	770
Call Expressions: <code>__call__</code>	770
Function Interfaces and Callback-Based Code	772
Comparisons: <code>__lt__</code> , <code>__gt__</code> , and Others	773
Boolean Tests: <code>__bool__</code> and <code>__len__</code>	775
Object Destruction: <code>__del__</code>	776
Destructor Usage Notes	776
Chapter Summary	777
Test Your Knowledge: Quiz	777
Test Your Knowledge: Answers	778
31. Designing with Classes.....	779
Python and OOP	779
Polymorphism Means Interfaces, Not Call Signatures	780
OOP and Inheritance: “Is-a” Relationships	780
OOP and Composition: “Has-a” Relationships	782
Stream Processors Revisited	784
OOP and Delegation: “Like-a” Relationships	786
Pseudoprivate Class Attributes	788
Name Mangling Overview	789
Why Use Pseudoprivate Attributes?	789
Method Objects: Bound or Not	791
Bound Methods in Action	792
Classes Are Objects: Generic Object Factories	795
Why Factories?	796
Multiple Inheritance and the MRO	796
How Multiple Inheritance Works	797
How the MRO Works	799
Attribute Conflict Resolution	801
Example: “Mix-in” Attribute Listers	802
Example: Mapping Attributes to Inheritance Sources	811
Other Design-Related Topics	815
Chapter Summary	815
Test Your Knowledge: Quiz	815
Test Your Knowledge: Answers	816
32. Class Odds and Ends.....	817
Extending Built-in Object Types	817
Extending Types by Embedding	818
Extending Types by Subclassing	819
The Python Object Model	821

Classes Are Types Are Classes	821
Some Instances Are More Equal Than Others	822
The Inheritance Bifurcation	823
The Metaclass/Class Dichotomy	824
And One “object” to Rule Them All	825
Advanced Attribute Tools	826
Slots: Attribute Declarations	826
Properties: Attribute Accessors	834
__getattr__ and Descriptors: Attribute Implementations	836
Static and Class Methods	837
Why the Special Methods?	837
Plain-Function Methods	838
Static Method Alternatives	839
Using Static and Class Methods	840
Counting Instances with Static Methods	841
Counting Instances with Class Methods	842
Decorators and Metaclasses	845
Function Decorator Basics	845
A First Look at User-Defined Function Decorators	847
A First Look at Class Decorators and Metaclasses	848
For More Details	850
The super Function	851
The super Basics	851
The super Details	852
The super Wrap-Up	859
Class Gotchas	860
Changing Class Attributes Can Have Side Effects	860
Changing Mutable Class Attributes Can Have Side Effects, Too	861
Multiple Inheritance: Order Matters	862
Scopes in Methods and Classes	863
Miscellaneous Class Gotchas	864
“Overwrapping-itis”	865
Chapter Summary	865
Test Your Knowledge: Quiz	865
Test Your Knowledge: Answers	866
Test Your Knowledge: Part VI Exercises	866

Part VII. Exceptions

33. Exception Basics.....	875
Why Use Exceptions?	875
Exception Roles	876
Exceptions: The Short Story	877

Default Exception Handler	877
Catching Exceptions	878
Raising Exceptions	879
User-Defined Exceptions	880
Termination Actions	881
Chapter Summary	882
Test Your Knowledge: Quiz	883
Test Your Knowledge: Answers	883
34. Exception Coding Details.....	885
The try Statement	885
try Statement Clauses	885
The except and else Clauses	886
The finally Clause	892
Combined try Clauses	894
The raise Statement	898
Raising Exceptions	898
The except as hook	898
Scopes and except as	899
Propagating Exceptions with raise	900
Exception Chaining: raise from	900
The assert Statement	902
Example: Trapping Constraints (but Not Errors!)	903
The with Statement and Context Managers	904
Basic with Usage	904
The Context-Management Protocol	905
Multiple Context Managers	907
The Termination-Handlers Shoot-Out	908
Chapter Summary	909
Test Your Knowledge: Quiz	910
Test Your Knowledge: Answers	910
35. Exception Objects.....	911
Exception Classes	912
Coding Exceptions Classes	912
Why Exception Hierarchies?	914
Built-in Exception Classes	916
Built-in Exception Categories	917
Default Printing and State	918
Custom Print Displays	919
Custom State and Behavior	920
Providing Exception Details	921
Providing Exception Methods	921
Exception Groups: Yet Another Star!	923

Chapter Summary	925
Test Your Knowledge: Quiz	926
Test Your Knowledge: Answers	926
36. Exception Odds and Ends	927
Nesting Exception Handlers	927
Example: Control-Flow Nesting	928
Example: Syntactic Nesting	930
Exception Idioms	931
Breaking Out of Multiple Nested Loops: “go to”	931
Exceptions Aren’t Always Errors	932
Functions Can Signal Conditions with raise	933
Closing Files and Server Connections	934
Debugging with Outer try Statements	934
Running In-Process Tests	935
More on sys.exc_info	935
Displaying Errors and Tracebacks	937
Exception Design Tips and Gotchas	938
What Should Be Wrapped	938
Catching Too Much: Avoid Empty except and Exception	938
Catching Too Little: Use Class-Based Categories	940
Core Language Wrap-Up	941
The Python Toolset	941
Development Tools for Larger Projects	941
Chapter Summary	944
Test Your Knowledge: Quiz	945
Test Your Knowledge: Answers	945
Test Your Knowledge: Part VII Exercises	945

Part VIII. Advanced Topics

37. Unicode and Byte Strings	949
Unicode Foundations	950
Character Representations	950
Character Encodings	951
Introducing Python String Tools	954
The str Object	954
The bytes Object	954
The bytearray Object	955
Text and Binary Files	955
Using Text Strings	956
Literals and Basic Properties	956
String Type Conversions	957

Coding Unicode Strings in Python	959
Source-File Encoding Declarations	964
Using Byte Strings	966
Methods	966
Sequence Operations	967
Formatting	968
Other Ways to Make Bytes	968
Mixing String Types	969
The bytearray Object	970
Using Text and Binary Files	972
Text-File Basics	973
Text and Binary Modes	973
Unicode-Text Files	975
Unicode, Bytes, and Other String Tools	977
The re Pattern-Matching Module	977
The struct Binary-Data Module	978
The pickle and json Serialization Modules	978
Filenames in open and Other Filename Tools	980
The Unicode Twilight Zone	983
Dropping the BOM in Python	983
Unicode Normalization: Whither Standard?	987
Chapter Summary	989
Test Your Knowledge: Quiz	989
Test Your Knowledge: Answers	989
38. Managed Attributes.....	991
39. Decorators.....	993
40. Metaclasses and Inheritance.....	995
To Metaclass or Not to Metaclass	996
The Downside of “Helper” Functions	996
Metaclasses Versus Class Decorators: Round 1	998
The Metaclass Model	999
Classes Are Instances of type	999
Metaclasses Are Subclasses of type	1000
Class Statements Call a type	1001
Class Statements Can Choose a type	1002
Metaclass Method Protocol	1003
Coding Metaclasses	1003
A Basic Metaclass	1003
Customizing Construction and Initialization	1005
Other Metaclass Coding Techniques	1006
Managing Classes with Metaclasses and Decorators	1009

Inheritance: The Finale	1014
Metaclass Versus Superclass	1016
Metaclass Inheritance	1017
Python Inheritance Algorithm: The Simple Version	1018
Python Inheritance Algorithm: The Less Simple Version	1021
The Inheritance Wrap-Up	1023
Metaclass Methods	1024
Metaclass Methods Versus Class Methods	1025
Operator Overloading in Metaclass Methods	1026
Metaclass Methods Versus Instance Methods	1026
Chapter Summary	1028
Test Your Knowledge: Quiz	1028
Test Your Knowledge: Answers	1029
41. All Good Things.....	1031
The Python Tsunami	1031
The Python Sandbox	1033
The Python Upside	1033
Closing Thoughts	1034
Where to Go from Here	1034
Encore: Print Your Own Completion Certificate!	1034

Part IX. Appendixes

A. Platform Usage Tips.....	1041
B. Solutions to End-of-Part Exercises.....	1063
Index.....	1103