

学习Python (影印版)

第五版
下册

Learning

Python



O'REILLY®

東南大學出版社

Mark Lutz 著

(第五版/下册)

学习Python (影印版)

Learning Python

Mark Lutz 著

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

O'Reilly Media, Inc. 授权东南大学出版社出版

南京 东南大学出版社

图书在版编目 (CIP) 数据

学习 Python: 第 5 版: 英文 / (美) 鲁特兹 (Lutz, M.)
著. —影印本. —南京: 东南大学出版社, 2014.1
书名原文: Learning Python, 5E
ISBN 978-7-5641-4597-2

I. ① 学… II. ① 鲁… III. ① 软件工具—程序设计—
英文 IV. ① TP311.56

中国版本图书馆 CIP 数据核字 (2013) 第 246109 号

江苏省版权局著作权合同登记

图字: 10-2013-369 号

©2013 by O'Reilly Media, Inc.

Reprint of the English Edition, jointly published by O'Reilly Media, Inc. and Southeast University Press, 2014. Authorized reprint of the original English edition, 2013 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2013。

英文影印版由东南大学出版社出版 2014。此影印版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式重制。

学习 Python 第五版 (影印版)

出版发行: 东南大学出版社
地 址: 南京四牌楼 2 号 邮编: 210096
出 版 人: 江建中
网 址: <http://www.seupress.com>
电子邮件: press@seupress.com
印 刷: 扬中市印刷有限公司
开 本: 787 毫米 × 980 毫米 16 开本
印 张: 101.75
字 数: 1993 千字
版 次: 2014 年 1 月第 1 版
印 次: 2014 年 1 月第 1 次印刷
书 号: ISBN 978-7-5641-4597-2
定 价: 148.00 元 (上下册)

本社图书若有印装质量问题, 请直接与营销部联系。电话 (传真): 025-83791830

Table of Contents

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: <code>__pycache__</code> 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