

《学习PYTHON 第三版(英文影印版)》

图书基本信息

书名：《学习PYTHON 第三版(英文影印版)》

13位ISBN编号：9787564112400

10位ISBN编号：7564112409

出版时间：2008-8-1

出版社：东南大学出版社

作者：(美国)(Lutz.M.)卢茨

页数：700 页

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu111.com

前言

About This Third Edition . In the four years since the second edition of this book was published in late 2003 , there have been substantial changes in Python itself, and in the topics I present in Python training sessions. Although I have attempted to retain as much of the prior version of this text as possible, this new edition reflects many recent changes in the Python language and in Python training, as well as a handful of structural changes.

This Edition ' s Python Language Changes On the language front, this edition has been thoroughly updated to reflect Python 2.5 and all changes to the language since the publication of the second edition. (The second edition was based largely on Python 2.2, with some 2.3 features grafted on at the end of the project.) In addition, discussions of anticipated changes in the upcoming Python 3.0 release have been incorporated where appropriate. Here are some of the major language topics of which you ' ll find new or expanded coverage in this edition:

The new B if A else C conditional expression (Chapter 12). with/as context managers (Chapter 27). try/except/finally unification (Chapter 27). Relative import syntax (Chapter 21). .. Generator expressions (Chapter 17). New generator function features (Chapter 17). Function decorators (Chapter 26). The set object type (Chapter 5). New built-in functions: sorted, sum, any, all, enumerate (Chapters 4 and 13). The decimal fixed-precision object type (Chapter 5). New and expanded material on files, list comprehensions, iterators, and more (Chapters 13 and 17). New development tools coverage: Eclipse, distutils, unittest and doctest, IDLE enhancements, Shedskin, and so on (Chapters 3 and 29). Smaller language changes (for instance, the widespread use of True and False, the new sys.exc_info for fetching exception details, and the demise of string-based exceptions, string methods, and the apply and reduce built-ins) are discussed throughout the book. In addition, some of the features that were new in the prior edition enjoy substantially expanded coverage here, including three-limit slices, and the arbitrary arguments call syntax that subsumes apply.

《学习PYTHON 第三版(英文影印版)》

内容概要

Python是一种理想的独立程序和脚本程序语言，具有可移植、强大和易用的特点。无论您是初学程序设计还是初学Python，都能通过《学习Python 第三版》快速而高效地掌握Python核心语言的基础。通读《学习Python 第三版》，您可以获得足够多的知识，能够学习将Python用于您所选择的任何应用领域。

《学习Python（第三版）》基于作者Mark Lutz在过去10年间讲授培训课程所使用的资料。除了大量的注解插图和每章总结外，作者还为每章加入了独特的一节，通过实际练习和复习提问的方式，让您在学习过程中练习新的技能，测验对概念的理解程度。

类型和运算——深入Python内置对象类型：数值、列表、字典及其他。

语句和语法——用来创建和处理对象的Python代码以及Python通用语法模型。

函数——Python的基本结构化和代码重用工具。..

模块——将语句、函数和其他工具组织为更大的组件。

类和面向对象编程——可选的Python面向对象程序设计工具，用于定制和重用结构化代码。

异常和工具——处理语句和模块异常，并介绍编写较大型程序所使用的开发工具。

有关即将发布的Python 3.0的一些讨论。

《学习Python 第三版》为您理解Python语言提供了完整而又深入的介绍，帮助您领会在以后的工作中会遇到的应用级Python实例。如果您准备好探索Google和YouTube究竟从Python中发现了什么，那么《学习Python 第三版》就是一个最好的开始。

《学习PYTHON 第三版(英文影印版)》

作者简介

作为Python培训界的领军人物，Mark Lutz是有关Python方面的最早和最畅销书籍的作者，并且从1992年起就是Python社区的先驱之一。同时他还是O'Reilly的《Programming Python》和《Python Pocket Reference》两本书的作者。Mark从1997年开始教授Python课程，截至2007年已经指导了超过200次培训。此外Mark拥有计算机科学学士和硕士学位，以及25年的软件开发经验。

书籍目录

Preface Part 1. Getting Started 1. A Python Q&A Session Why Do People Use Python? Software Quality Developer Productivity Is Python a “ Scripting Language ” ? OK , but What ’ S the Downside? Who Uses Python Today? What Can I Do with Python? Systems Programming GUIs Internet Scripting Component Integration Database Programming Rapid Prototyping Numeric and Scientific Programming Gaming , AI , XML , Robots , and More What Are Python ’ S Technical Strengths? It ’ S Object Oriented It ’ S Free It ’ S Portable It ’ S Powerful It ’ S Mixable It ’ S Easy to Use It ’ S Easy to Learn It ’ s Named After Monty Python How Does Python Stack Up to Language X? Chapter Summary Brain Builder Chapter Quiz Quiz Answers 2. How Python Runs Programs Introducing the Python Interpreter Program Execution The Programmer ’ S View Python ’ S View Byte code compilation The Python Virtual Machine (PVM) Performance implications Development implications Execution Model Variations Python Implementation Alternatives CPython Jython IronPython Execution Optimization Tools The Psyco just-in-time compiler The Shedskin C++ translator Frozen Binaries Future Possibilities? Chapter Summary Brain Builder Chapter Quiz Quiz Answers 3. How You Run Programs Interactive Coding Using the Interactive Prompt System Command Lines and Files Using Command Lines and Files Unix Executable Scripts (!) Clicking File Icons Clicking Icons on Windows The raw_input Trick Other Icon-Click Limitations Module Imports and Reloads The Grander Module Story : Attributes Modules and namespaces import and reload Usage Notes The IDLE User Interface IDLE Basics Using IDLE Advanced IDLE Tools Other IDEs Embedding Calls Frozen Binary Executables Text Editor Launch Options Other Launch Options Future Possibilities? Which Option Should I Use? Chapter Summary Brain Builder Chapter Quiz Quiz Answers Brain Builder : Part I Exercises Part 4 . Types and Operations 4 . Introducing Python Object Types Why Use Built—in Types? Python ’ S Core Data Types Numbers Strings Sequence Operations Immutability Type—Specific Methods Getting Help Other Ways to Code Strings Pattern Matching Lists Sequence Operations Type—Specific Operations Bounds Checking Nesting List Comprehensions Dictionaries Mapping Operations Nesting Revisited Sorting Keys : for Loops Iteration and Optimization Missing Keys : if Tests Tuples Why Tuples? Files Other File-Like Tools Other Core Types How to Break Your Code ’ S Flexibility User—Defined Classes And Everything Else Chapter Summary Brain Builder Chapter Quiz Quiz Answers 5 . Numbers Python Numeric Types Numeric Literals Built—in Numeric Tools and Extensions Python Expression Operators Mixed Operators Follow Operator Precedence Parentheses Group Subexpressions Mixed Types Are Converted Up Preview : Operator Overloading Numbers in Action Variables and Basic Expression S Numeric Display Formats Division : Classic , Floor , and True Bitwise Operations Long Integers Complex Numbers Hexadecimal and Octal Notation Other Built . in Numeric Tools Other Numeric Types Decimal Numbers Sets Booleans Third . Party Extensions Chapter Summary Brain Builder Chapter Quiz Quiz Answers 6 . The Dynamic Typing Interlude The Case of the Missing Declaration Statements Variables , Objects , and References Types Live with Objects , Not Variables Objects Are Garbage —Collected Shared Refe : rences Shared References and In—Place Changes Shared References and Equality Dynamic Typing Is Everywhere Chapter Summary Brain Builder Chapter Quiz Quiz Answers 7. Strings String Literals Single , and Double . Quoted Strings Are the Same Escape Sequences Represent Special Bytes Raw Strings Suppress Escapes Triple Quotes Code Multiline Block Strings Unicode Strings Encode Larger Character Sets Strings in Action Basic Operations Indexing and Slicing Extended slicing : the third limit String Conversion Tools Character code conversions Changing Strings String Formatting Advanced String Formatting Dictionary —Based String Formatting String Methods String Method Examples : Changing Strings String Method Examples : Parsing Text Other Common String Methods in Action The Original string Module General Type Categories Types Share Operation Sets by Categories Mutable Types Can Be Changed In—Place Chapter Summary Brain Builder Chapter Quiz Quiz Answers 8. Lists and Dictionaries Lists Lists in Action Basic List Operations Indexing , Slicing , and Matrixes Changing Lists In—Place Index and slice assignments List method calls Other common list operations Dictionaries Dictionaries in Action Basic Dictionary Operations Changing Dictionaries In—Place More Dictionary Methods A Languages Table Dictionary Usage Notes Using dictionaries to simulate flexible lists Using dictionaries for sparse data structures Avoiding missing-key errors Using dictionaries as

“ records ” Other ways to make dictionariesChapter SummaryBrain BuilderChapter QuizQuiz Answers9.Tuples,Files,and Everthing ElseTuplesTuples in ActionTuple syntax peculiarities : commas and parenthesesConversions and immutabilityWhy Lists and Tuples?FilesOpening FilesUsing FilesFiles in ActionStoring and parsing Python objects in filesStoring native Python objects with pickleStoring and parsing packed binary data in filesOther File ToolsType Categories RevisitedObject FlexibilityReferences Versus CopiesComparisons , Equality , and TruthThe Meaning of True and False in PythonPython ' S Type HierarchiesOther Types in PythonBuilt . in Type GotchasAssignment Creates References , Not CopiesRepetition Adds One Level Deep Beware of Cyclic Data StructuresImmutable Types Can ' t Be Changed In-PlaceChapter SummaryBrain BuilderChapter QuizQuiz AnswersBrain Builder : Part II ExercisesPart Statements and SyntaxIntroducing Python StatementsPython Program Structure RevisitedPython ' S StatementsATale ofTwo ifsWhat Python AddsWhat Python RemovesParentheses are optionalEnd of line iS end of statementEnd of indentation iS end of blockWhy Indentation Syntax?A Few Special CasesStatement rule special casesBlock rule special caseA Quick Example : Interactive LoopsA Simple Interactive LoopDoing Math on User InputsHandling Errors by Testing InputsHandling Errors with try StatementsNesting Code Three Levels DeepChapter SummaryBrain BuilderChapter QuizQuiz Answers11 . Assignment , Expressions , and printAssignment StatementsAssignment Statement FormsSequence AssignmentsAdvanced sequence assignment patternsMultiple—Target AssignmentsMultiple—target assignment and shared referencesAugmented AssignmentsAugmented assignment and shared referencesVariable Name RulesNaming conventionsNames have no type , but objects doExpression StatementsExpression Statements and In-Place Changesprint StatementsThe Python “ Hello World ” ProgramRedirecting the Output StreamThe print file ExtensionChapter SummaryBrain BuilderChapter QuizQuiz Answers12 . if Tesisif Statements General FormatBasic ExamplesMultiway BranchingPython Syntax RulesBlock DelimitersStatement DelimitersA Few Special CasesTruth TestsThe if/else Ternary ExpressionChapter SummaryBrain BuilderChapter QuizQuiz Answers13 . while and for Loopswhile LoopsGeneral FormatExamplesbreak , continue , pass , and the Loop elseGeneral Loop FormatExamplespasscontinuebreakelseMore on the loop else clauseforLoopsGeneral FormatExamplesBasic usageOther data typesTuple assignment in forNested for loopsIterators : A First LookFile IteratorsOther Built—in Type IteratorsOther Iteration ContextsUser—Defined IteratorsLoop Coding TechniquesCounter Loops : while and rangeNonexhaustive Traversals : rangeChanging Lists : rangeParallel Traversals : zip and mapDictionary construction with zipGenerating Both Offsets and Items : enumerateList Comprehensions : A First LookList Comprehension BasicsUsing List Comprehensions on FilesExtended List Comprehension SyntaxChapter SummaryBrain BuilderChapter QuizQuiz Answers14 . The Documentation InterludePython Documentation Sources#CommentsThe dir FunctionDocstrings : --doc--User-defined docstringsDocstring standardsBuilt-in docstringsPyDoc : The help FunctionPyDoc : HTML ReportsStandard Manual SetWeb ResourcesPublished BooksCommon Coding Gotchas.....Part Function15.Function Basics16.Scopes and Arguments17.Advanced Function TopicsPart .Modules18.Modules:The Big Picture19.Module Coding Basics20.Module Packages21.Advanced Module TopicsPart Classes and OOP22.OOP:The Big Picture23.Class Coding Basics24.Class Coding Details25.Designing with Classes26.Advanced Class TopicsPart Exceptions and Tools27.Exception Basics28.Exception Objects29.Designing with ExceptionsPart AppendixesA.Installation and ConfigurationB.Solutions to End-of-Part ExercisesIndex

《学习PYTHON 第三版(英文影印版)》

编辑推荐

卢茨（Lutz，M），作为Python培训界的领军人物，Mark Lutz是有关Python方面的最早和最畅销书籍的作者，并且从1992年起就是Python社区的先驱之一。同时他还是OReilly的《Programming Python》和《Python Pocket Reference》两本书的作者。Mark从1997年开始教授Python课程，截至2007年已经指导了超过200次培训。此外Mark拥有计算机科学学士和硕士学位，以及25年的软件开发经验。

精彩短评

1、 <https://book.douban.com/review/3196548/>

严重同意这个书评，读了简直让你想哭。看完了一两百页后居然都不会完整编一个程序，改看a byte of python了

2、 finally....

3、 讲得太细太繁琐了感觉

4、 python入门经典，讲解很细致到位。

5、 看的很爽，尤其是讲class是如何动态的时候。第三本Python书，算是比较全面的学过了。

6、 最后两张还是有可读性的，就是啰嗦

7、 似乎中文版没办法翻译出原版的神韵，每每看到中文版中的“元组”之类，都会觉得别扭。这本书虽然前面有些啰嗦，比如Python中的数据存储方式，但读到后来会发现啰嗦一些是对的，不然后面会犯一些需要更多时间来纠正的错误，比如为什么我swap了这两个数它们却没被swap。另外结合《Invent Your Own Computer Games with Python 2nd》能比较好，不放到实际项目里体会一下一整个程序的结构可不行。不适合从来没接触过编程的初学者，适合接触过编程如C++却没有多少项目开发经验的。开发效率一流，写些小脚本用来完成些小任务再合适不过。

8、 不错的入门书籍，不过老外写书的特点之一就是罗嗦，很多小东西都要扯半天，这点究竟是好还是不好实在没法评断。

9、 好基础好基础的，看得有点乏味。现在可以把它当工具书使了。

10、 适合初学者入门，讲解得非常详细以至于有些啰嗦。过了一遍，弥补了些以前不知道的知识点，也算是系统地补了一下Python的基础吧。

11、 Python

12、 感觉讲得太啰嗦了，适合初学者

13、 这本书讲的很清楚明白 浅显易懂 是初学者的好素材。

14、 影印很好, 原味又便宜.

15、 1、 比较详细

2、 比较啰嗦

精彩书评

- 1、这是一本相当不符合python哲学的书——它太啰嗦了：一个内容，先在第一章告诉你个大概（概览），然后在下个章节多告诉你一点（更多细节），过了几十页之后，终于在最后一个章节把最后一部分知识补完（全部细节）。各个部分中的内容有些不断重复，有些却是新的，你不能完全跳过，但又害怕与新内容失之交臂，于是不得不翻遍几十页，只为了看其中真正有用的五六页，这种阅读体验真是让人不胜其烦；这本书读起来的感觉像上吊，每翻一页你都感到自己多窒息了一点。对读者有用的书不是堆给他们一堆细节和重复又重复的内容，让他们看上百页才把什么是模块、怎么import说明白，而是应该赶紧把有用的内容告诉他们，让他们尽快动手尝试，开始写真正的程序，才是对读者有用的学习方式（为什么你就不能一次把整个故事说完呢？）。这本书作为语言的细节参考，有可取之处，但作为python新手的第一本书，绝对是不适合的。Python社区崇尚大道至简的Pythonic哲学，而一本用七八百页纸写python语法的书，本身就很pythonic——这种事情还是留给c++和java那群人做吧，python程序员，不需要变成死相。人生苦短，远离Learning Python.
- 2、现在想学习Python了，在网上查了查，大家都说这本书比较好，如何好呢？大家往下看：一.作者是Python培训界的领军人物，Python社区的先驱之一，25年的软件开发经验；（大牛啊）二.这书讲的是Python的核心功能，是一本教程指南，不是以说明怎么进行应用级编程（Application Programming）为目的的，也不是作为手册使用的。如果了解如何进行应用级编程看这本：Programming Python；如果要看手册有其它的书：Python in the nushell、Python Cookbook、Python Pocket Reference、Python Essential Referenct等等；（看来我要慢慢看了）三.本书的架构是由粗到细，有全貌到细化的知识点，在内容组织上有板有眼，几句话把怎么学，如何学说的清清楚楚，唯恐读者浪费时间和精力。总之，一句话如果您刚开始接触Python，那这本书将为你带来一个良好的开端。（买来寒假看，不过价格太贵了88元呢）
- 3、似乎中文版没办法翻译出原版的神韵，每每看到中文版中的“元组”之类，都会觉得别扭。这本书虽然前面有些啰嗦，比如Python中的数据存储方式，但读到后来会发现啰嗦一些是对的，不然后面会犯一些需要更多时间来纠正的错误，比如为什么我swap了这两个数它们却没被swap。另外结合《Invent Your Own Computer Games with Python 2nd》能比较好，不放到实际项目里体会一下一整个程序的结构可不行。不适合从来没接触过编程的初学者，适合接触过编程如C++却没有多少项目开发经验的。开发效率一流，写些小脚本用来完成些小任务再合适不过。

章节试读

1、《学习PYTHON 第三版(英文影印版)》的笔记-第82页

Every object in Python is classified as either immutable (unchangeable) or not. In terms of the core types, numbers, strings, and tuples are immutable; lists and dictionaries are not (they can be changed in-place freely). Among other things, immutability can be used to guarantee that an object remains constant throughout your program.

《学习PYTHON 第三版(英文影印版)》

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:www.tushu111.com