

Learning Python

原书第5版(上册)  
更新Python 2.7和3.3

# Python

学习手册



O'REILLY®



机械工业出版社  
China Machine Press



HZ BOOKS  
华章IT

Mark Lutz 著  
秦鹤 林明 译  
林涵菲 审校

原书第5版

# Python学习手册

(上册)

Mark Lutz 著

秦鹤 林明 译

林涵菲 审校



Beijing • Boston • Farnham • Sebastopol • Tokyo **O'REILLY®**

O'Reilly Media, Inc. 授权机械工业出版社出版

机械工业出版社

## 图书在版编目 (CIP) 数据

Python 学习手册：原书第 5 版 / (美) 马克·卢茨 (Mark Lutz) 著；秦鹤，林明译。  
—北京：机械工业出版社，2018.8

(O'Reilly 精品图书系列)

书名原文：Learning Python, Fifth Edition

ISBN 978-7-111-60366-5

I. P… II. ①马… ②秦… ③林… III. 软件工具—程序设计—手册 IV.TP311.561-62

中国版本图书馆 CIP 数据核字 (2018) 第 146242 号

北京市版权局著作权合同登记

图字：01-2013-5994 号

©2013 Mark Lutz. All rights reserved.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and China Machine Press, 2018. Authorized translation of the English edition, 2018 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。

简体中文版由机械工业出版社出版 2018。英文原版的翻译得到 O'Reilly Media, Inc. 的授权。此简体中文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。

封底无防伪标均为盗版

本书法律顾问

北京大成律师事务所 韩光 / 邹晓东

书 名 / Python 学习手册 (原书第 5 版)

书 号 / ISBN 978-7-111-60366-5

责任编辑 / 陈佳媛

封面设计 / Randy Comer, 张健

出版发行 / 机械工业出版社

地 址 / 北京市西城区百万庄大街 22 号 (邮政编码 100037)

印 刷 / 北京市兆成印刷有限责任公司

开 本 / 178 毫米 × 233 毫米 16 开本 49.75 印张

版 次 / 2018 年 11 月第 1 版 2018 年 11 月第 1 次印刷

定 价 / 219.00 元 (共 2 册)

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：(010)88379426；88361066

购书热线：(010)68326294；88379649；68995259

投稿热线：(010)88379604

读者信箱：hzit@hzbook.com

# O'Reilly Media, Inc. 介绍

O'Reilly Media 通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自 1978 年开始，O'Reilly 一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O'Reilly 的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly 为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了《Make》杂志，从而成为 DIY 革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly 的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O'Reilly 现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版，在线服务或者面授课程，每一项 O'Reilly 的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

## 业界评论

“O'Reilly Radar 博客有口皆碑。”

——Wired

“O'Reilly 凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——Business 2.0

“O'Reilly Conference 是聚集关键思想领袖的绝对典范。”

——CRN

“一本 O'Reilly 的书就代表一个有用、有前途、需要学习的主题。”

——Irish Times

“Tim 是位特立独行的商人，他不光放眼于最长远、最广阔的视野并且切实地按照 Yogi Berra 的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去 Tim 似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——Linux Journal

# 译者序

如果你想在2018年制定一个“学习编程”的年度计划，那么我们会首先推荐Python这门编程语言。根据目前全球最大的开源代码仓库网站GitHub提供的数据，在2017年全球各地的程序员向GitHub网站发起了约100万次Python代码的提交。按照这一指标，Python无疑是目前世界范围内排名第二的编程语言。而放眼当今的IT界，Python自己也像无数励志故事的主角一样，不断成长壮大，并从一众编程语言中脱颖而出，在人工智能、大数据分析、科学计算、大型网站服务搭建等领域大放异彩。

而这本《Python学习手册》，正是帮助你入门Python或是进阶学习的优质资源。当前市面上Python学习的书籍资源非常丰富，相比之下本书有两个主要特点：一是，本书以目前更主流的Python 3.X系列为主，同时兼顾到Python 2.X的内容，而不是只停留在2.X的范畴；二是，本书不贪多，并不企图对所有Python的类库都浅尝辄止，而是专注地把Python语言的核心知识讲解透彻。正因如此，本书适合作为学习Python的第一本入门书，同时也可用作强化Python核心编程的进阶读物。读者可以放心地用本书构筑起稳固的Python核心基础，在此之上，再选购其他书籍，深入学习Python在其他特定领域的应用。

当然，这本书的篇幅也十分浩大，难免令人心生畏惧。你也许会有这两个疑惑：Python核心编程知识真的需要这么多篇幅来讲述吗？作者除了讲解Python，是不是也聊了些其他内容？

确实，如果只是列举Python语法和库函数，那么估计几十页的一本小册子就能胜任，但是如果帮助读者真正理解和掌握，就不能跳过知识点背后的原理、思想和例子。在翻译本书的过程中，译者能够真切地体会到作者Mark Lutz先生的耐心与细致。作者孜孜不倦地将许多Python中看似高深的主题掰开揉碎，娓娓道来。对于几乎每一个知识，都会以知识点、思想、示例代码的方式详细展开。一方面，读者可以自行选择略读一些段落；另一方面，这样的讲解对于理清复杂的知识非常有益。

诚然，本书确实聊了一些其他的内容，例如，在第二部分和第五部分，深入到 Python 解释器的层面来讲程序运行和包导入；在第四部分，从软件工程中程序设计的角度来讲函数的编写；在第六部分，结合设计模式的思想来讲 Python 中类的编写等。可以说，作者不满足于让读者会写 Python，而是希望读者能写得“优雅”，做到熟练应用 Python 中的最佳实践。

再聊聊如何阅读本书。面对这样一本“大部头”，译者自己的第一感觉是要将它作为工具书以供查阅。但诚如作者在前言中所说，本书更适合作为教材，更值得从头到尾按章节顺序翻阅学习一遍。不过，如果读者确实已经熟悉 Python，则可以通过目录来判断某一章节是否需要跳过，并直接尝试去做跳过章节的“本章习题”。本书附录 D 给出了各章及各部分的习题答案。同时，每一部分的最后一章都用一节来专门总结该章“陷阱”，直接阅读“陷阱”节能够帮助读者快速地补齐很多“知识漏洞”。不过，对于不熟悉的章节，还是建议读者完整阅读。值得一提的是，书中代码的注释部分本身就带有解释性。因此，我们希望能有能力的读者仔细阅读代码的英文注释，从而加深对代码和概念的理解。

从内容上来说，本书前五部分（上半册）主要介绍了：语言概览（第一部分）、核心类型及操作（第二部分）、语句和语法（第三部分）、函数和生成器（第四部分）、模块和包（第五部分）。这五部分交代了 Python 语言的基础，其中的翔实讲解非常适合初学者系统学习，以及进阶者查缺补漏。本书后四部分（下半册）主要介绍了：类和面向对象（第六部分）、异常（第七部分）、高级主题（第八部分）。对于通常的 Python 编程，阅读到第七部分基本能够胜任。第八部分可以作为选读部分，在读者遇到相应问题的时候，再做针对性的补充学习。

本书注释部分包含原书注和译注两部分。其中译注部分，旨在帮助读者更好地理解原书的意图。第 5 版的整体翻译风格轻松活泼，保持了英文原书原汁原味的风趣幽默。同时，相较本书之前的中文版，我们也做了大量翻译上的调整，从而方便读者阅读。

在所有翻译决定的背后，是对基本概念讲述的仔细推敲。翻译团队内部频繁而密切的交流，加之与原书作者持续的沟通，我们力求保证翻译的质量。当然，最终成书中也不免会有纰漏，希望读者能够包容，并不吝斧正。如果有任何勘误信息或意见建议，或者希望与译者交流，请联系邮箱 [LearningPython5EChineseBook@outlook.com](mailto:LearningPython5EChineseBook@outlook.com) 或者 GitHub 仓库 [qinacme/Learning-Python-5E-Chinese-Book](https://github.com/qinacme/Learning-Python-5E-Chinese-Book)。衷心希望各位读者能够尽情享受阅读和学习 Python 的快乐，并把这份热爱带往今后的编程和生活中。

感谢林涵菲和陈英教授对本书细致的审校。感谢龚靖渝、陈灵健在本书翻译过程中提供的帮助。

译者

2018 年 9 月

# 目录

## 上册

|                                  |           |
|----------------------------------|-----------|
| 前言 .....                         | 1         |
| <b>第一部分 使用入门</b>                 |           |
| <b>第1章 问答环节 .....</b>            | <b>21</b> |
| 人们为何使用Python .....               | 21        |
| 软件质量 .....                       | 22        |
| 开发者效率.....                       | 23        |
| Python是一门“脚本语言”吗 .....           | 23        |
| 好吧，Python的缺点是什么.....             | 25        |
| 如今谁在使用Python .....               | 25        |
| 其他的Python设计权衡：一些难以把握的方面 .....    | 26        |
| 使用Python可以做些什么.....              | 28        |
| 系统编程 .....                       | 28        |
| 图形用户界面（GUI） .....                | 29        |
| Internet脚本.....                  | 29        |
| 组件集成 .....                       | 30        |
| 数据库编程.....                       | 30        |
| 快速原型 .....                       | 31        |
| 数值计算和科学计算编程 .....                | 31        |
| 更多内容：游戏、图像、数据挖掘、机器人、Excel等 ..... | 31        |
| Python如何开发并获得支持.....             | 32        |
| 开源的权衡.....                       | 33        |

|                               |           |
|-------------------------------|-----------|
| Python有哪些技术上的优点 .....         | 33        |
| 面向对象和函数式 .....                | 33        |
| 免费 .....                      | 34        |
| 可移植 .....                     | 34        |
| 功能强大 .....                    | 35        |
| 可混合 .....                     | 36        |
| 相对简单易用 .....                  | 36        |
| 相对简单易学 .....                  | 37        |
| 以Monty Python命名 .....         | 37        |
| Python和其他语言比较起来怎么样 .....      | 38        |
| 本章小结 .....                    | 39        |
| 本章习题 .....                    | 39        |
| 习题解答 .....                    | 40        |
| Python是工程，不是艺术 .....          | 41        |
| <b>第2章 Python如何运行程序 .....</b> | <b>43</b> |
| Python解释器简介 .....             | 43        |
| 程序执行 .....                    | 45        |
| 程序员的视角 .....                  | 45        |
| Python的视角 .....               | 46        |
| 执行模型的变体 .....                 | 48        |
| Python的各种实现 .....             | 49        |
| 执行优化工具 .....                  | 52        |
| 冻结二进制文件 .....                 | 53        |
| 未来的可能性 .....                  | 54        |
| 本章小结 .....                    | 55        |
| 本章习题 .....                    | 55        |
| 习题解答 .....                    | 56        |
| <b>第3章 你应如何运行程序 .....</b>     | <b>57</b> |
| 交互式命令行模式 .....                | 57        |
| 开始一个交互式会话 .....               | 57        |
| Windows平台上的交互式命令行在哪里 .....    | 59        |



|  |    |
|--|----|
| 系统路径 .....                               | 59 |
| Python 3.3中的新Windows选项：PATH和启动器 .....    | 60 |
| 运行的位置：代码目录 .....                         | 61 |
| 不需要输入的内容：提示符和注释 .....                    | 62 |
| 交互式地运行代码 .....                           | 63 |
| 为什么要使用交互式命令行模式 .....                     | 64 |
| 使用注意：交互命令行模式 .....                       | 65 |
| 系统命令行和文件 .....                           | 67 |
| 第一段脚本 .....                              | 68 |
| 使用命令行运行文件 .....                          | 69 |
| 不同的命令行使用方式 .....                         | 70 |
| 使用注意：命令行和文件 .....                        | 71 |
| UNIX风格可执行脚本：#! .....                     | 72 |
| UNIX脚本基础 .....                           | 72 |
| UNIX env查找技巧 .....                       | 73 |
| Python 3.3 Windows启动器：Windows也有#!了 ..... | 73 |
| 点击文件图标 .....                             | 75 |
| 图标点击基础知识 .....                           | 75 |
| 在Windows上点击图标 .....                      | 76 |
| Windows上输入的技巧 .....                      | 77 |
| 其他图标点击的限制 .....                          | 79 |
| 模块导入和重载 .....                            | 79 |
| 导入和重载基础知识 .....                          | 80 |
| 模块的宏观视角：属性 .....                         | 82 |
| 使用注意：import和reload .....                 | 85 |
| 使用exec运行模块文件 .....                       | 85 |
| IDLE用户界面 .....                           | 86 |
| IDLE启动细节 .....                           | 87 |
| IDLE基础用法 .....                           | 88 |
| IDLE功能特性 .....                           | 89 |
| 高级IDLE工具 .....                           | 90 |
| 使用注意：IDLE .....                          | 90 |

|                 |    |
|-----------------|----|
| 其他IDE.....      | 92 |
| 其他启动选项.....     | 93 |
| 嵌入式调用.....      | 93 |
| 冻结二进制可执行文件..... | 94 |
| 文本编辑器启动方式.....  | 94 |
| 其他的启动方式.....    | 95 |
| 未来的可能.....      | 95 |
| 应该选用哪种方式.....   | 95 |
| 调试Python代码..... | 96 |
| 本章小结.....       | 97 |
| 本章习题.....       | 97 |
| 习题解答.....       | 98 |
| 第一部分练习题.....    | 99 |

## 第二部分 类型和运算

|                              |            |
|------------------------------|------------|
| <b>第4章 介绍Python对象类型.....</b> | <b>105</b> |
| Python知识结构.....              | 105        |
| 为什么要使用内置类型.....              | 106        |
| Python核心数据类型.....            | 107        |
| 数字.....                      | 108        |
| 字符串.....                     | 110        |
| 序列操作.....                    | 110        |
| 不可变性.....                    | 112        |
| 特定类型的方法.....                 | 113        |
| 寻求帮助.....                    | 115        |
| 字符串编程的其他方式.....              | 116        |
| Unicode字符串.....              | 117        |
| 模式匹配.....                    | 119        |
| 列表.....                      | 120        |
| 序列操作.....                    | 120        |
| 特定的操作.....                   | 121        |

|                    |     |
|--------------------|-----|
| 边界检查 .....         | 121 |
| 嵌套 .....           | 122 |
| 推导 .....           | 122 |
| 字典 .....           | 125 |
| 映射操作 .....         | 125 |
| 重访嵌套 .....         | 126 |
| 不存在的键: if 测试 ..... | 128 |
| 键的排序: for 循环 ..... | 129 |
| 迭代和优化 .....        | 131 |
| 元组 .....           | 132 |
| 为什么要使用元组 .....     | 133 |
| 文件 .....           | 133 |
| 二进制字节文件 .....      | 134 |
| Unicode文本文件 .....  | 135 |
| 其他类文件工具 .....      | 137 |
| 其他核心类型 .....       | 137 |
| 如何破坏代码的灵活性 .....   | 139 |
| 用户定义的类 .....       | 140 |
| 剩余的内容 .....        | 141 |
| 本章小结 .....         | 141 |
| 本章习题 .....         | 141 |
| 习题解答 .....         | 142 |

## 第5章 数值类型 ..... 143

|                    |     |
|--------------------|-----|
| 数值类型基础知识 .....     | 143 |
| 数值字面量 .....        | 144 |
| 内置数值工具 .....       | 146 |
| Python表达式运算符 ..... | 146 |
| 数字的实际应用 .....      | 151 |
| 变量与基础表达式 .....     | 151 |
| 数值的显示格式 .....      | 153 |
| str和repr显示格式 ..... | 154 |

|                          |     |
|--------------------------|-----|
| 普通比较与链式比较.....           | 154 |
| 除法：经典除法、向下取整除法和真除法.....  | 156 |
| 整数精度.....                | 160 |
| 复数.....                  | 160 |
| 十六进制、八进制和二进制：字面量与转换..... | 161 |
| 按位操作.....                | 163 |
| 其他内置数值工具.....            | 164 |
| 其他数值类型.....              | 167 |
| 小数类型.....                | 167 |
| 分数类型.....                | 169 |
| 集合.....                  | 173 |
| 布尔型.....                 | 180 |
| 数值扩展.....                | 181 |
| 本章小结.....                | 182 |
| 本章习题.....                | 182 |
| 习题解答.....                | 182 |

## **第6章 动态类型 ..... 184**

|                        |     |
|------------------------|-----|
| 缺少声明语句的情况.....         | 184 |
| 变量、对象和引用.....          | 184 |
| 类型属于对象，而不是变量.....      | 186 |
| 对象的垃圾收集.....           | 187 |
| 关于Python垃圾回收的更多讨论..... | 188 |
| 共享引用.....              | 189 |
| 共享引用和在原位置修改.....       | 190 |
| 共享引用和相等.....           | 192 |
| 动态类型随处可见.....          | 193 |
| “弱”引用.....             | 194 |
| 本章小结.....              | 194 |
| 本章习题.....              | 194 |
| 习题解答.....              | 195 |

|                                     |            |
|-------------------------------------|------------|
| <b>第7章 字符串基础</b> .....              | <b>196</b> |
| 本章范围 .....                          | 196        |
| Unicode简介 .....                     | 197        |
| 字符串基础 .....                         | 197        |
| 字符串字面量 .....                        | 199        |
| 单引号和双引号字符串是一样的 .....                | 200        |
| 转义序列代表特殊字符 .....                    | 200        |
| 原始字符串阻止转义 .....                     | 203        |
| 三引号编写多行块字符串 .....                   | 205        |
| 实际应用中的字符串 .....                     | 206        |
| 基本操作 .....                          | 206        |
| 索引和分片 .....                         | 207        |
| 请注意：分片 .....                        | 211        |
| 字符串转换工具 .....                       | 212        |
| 修改字符串I .....                        | 214        |
| 字符串方法 .....                         | 216        |
| 方法调用语法 .....                        | 216        |
| 字符串的方法 .....                        | 217        |
| 字符串方法示例：修改字符串II .....               | 218        |
| 字符串方法示例：解析文本 .....                  | 220        |
| 实际应用中的其他常见字符串方法 .....               | 220        |
| 原始string模块的函数（在Python 3.X中删除） ..... | 221        |
| 字符串格式化表达式 .....                     | 223        |
| 格式化表达式基础 .....                      | 224        |
| 高级格式化表达式语法 .....                    | 225        |
| 高级格式化表达式举例 .....                    | 226        |
| 基于字典的格式化表达式 .....                   | 227        |
| 字符串格式化方法调用 .....                    | 228        |
| 字符串格式化方法基础 .....                    | 228        |
| 添加键、属性和偏移量 .....                    | 229        |
| 高级格式化方法语法 .....                     | 230        |
| 高级格式化方法举例 .....                     | 231        |

|                              |            |
|------------------------------|------------|
| 与%格式化表达式比较.....              | 233        |
| 为什么使用格式化方法.....              | 236        |
| 通用类型分类.....                  | 241        |
| 同一分类中的类型共享同一个操作集.....        | 241        |
| 可变类型能够在原位置修改.....            | 242        |
| 本章小结.....                    | 242        |
| 本章习题.....                    | 243        |
| 习题解答.....                    | 243        |
| <b>第8章 列表与字典.....</b>        | <b>245</b> |
| 列表.....                      | 245        |
| 列表的实际应用.....                 | 248        |
| 基本列表操作.....                  | 248        |
| 列表迭代和推导.....                 | 248        |
| 索引、分片和矩阵.....                | 249        |
| 原位置修改列表.....                 | 250        |
| 字典.....                      | 256        |
| 字典的实际应用.....                 | 258        |
| 字典的基本操作.....                 | 258        |
| 原位置修改字典.....                 | 259        |
| 其他字典方法.....                  | 260        |
| 示例：电影数据库.....                | 262        |
| 字典用法注意事项.....                | 263        |
| 创建字典的其他方式.....               | 267        |
| 请留意：字典vs列表.....              | 268        |
| Python 3.X和2.7中的字典变化.....    | 269        |
| 请留意：字典接口.....                | 276        |
| 本章小结.....                    | 277        |
| 本章习题.....                    | 277        |
| 习题解答.....                    | 277        |
| <b>第9章 元组、文件与其他核心类型.....</b> | <b>279</b> |
| 元组.....                      | 280        |

|                            |     |
|----------------------------|-----|
| 元组的实际应用 .....              | 281 |
| 为什么有了列表还要元组 .....          | 283 |
| 重访记录：有名元组 .....            | 284 |
| 文件 .....                   | 286 |
| 打开文件 .....                 | 287 |
| 使用文件 .....                 | 288 |
| 文件的实际应用 .....              | 289 |
| 文本和二进制文件：一个简要的故事 .....     | 290 |
| 在文件中存储Python对象：转换 .....    | 291 |
| 存储Python原生对象：pickle .....  | 293 |
| 用JSON格式存储Python对象 .....    | 294 |
| 存储打包二进制数据：struct .....     | 296 |
| 文件上下文管理器 .....             | 297 |
| 其他文件工具 .....               | 297 |
| 核心类型复习与总结 .....            | 299 |
| 请留意：运算符重载 .....            | 300 |
| 对象灵活性 .....                | 300 |
| 引用vs复制 .....               | 301 |
| 比较、等价性和真值 .....            | 303 |
| Python中True和False的含义 ..... | 307 |
| Python的类型层次 .....          | 309 |
| 类型的对象 .....                | 309 |
| Python中的其他类型 .....         | 311 |
| 内置类型陷阱 .....               | 311 |
| 赋值创建引用，而不是复制 .....         | 311 |
| 重复会增加层次深度 .....            | 312 |
| 注意循环数据结构 .....             | 313 |
| 不可变类型不可在原位置改变 .....        | 313 |
| 本章小结 .....                 | 314 |
| 本章习题 .....                 | 314 |
| 习题解答 .....                 | 315 |
| 第二部分练习题 .....              | 315 |

## 第三部分 语句和语法

|                              |            |
|------------------------------|------------|
| <b>第10章 Python语句简介</b> ..... | <b>321</b> |
| 重温Python的知识结构 .....          | 321        |
| Python的语句 .....              | 322        |
| 两种不同的if .....                | 324        |
| Python增加的元素 .....            | 325        |
| Python删除的元素 .....            | 325        |
| 为什么采用缩进语法 .....              | 327        |
| 几种特殊情况 .....                 | 329        |
| 简短示例：交互式循环 .....             | 331        |
| 一个简单的交互式循环 .....             | 331        |
| 对用户输入做数学运算 .....             | 333        |
| 通过测试输入数据来处理错误 .....          | 334        |
| 用try语句处理错误 .....             | 335        |
| 嵌套三层深的代码 .....               | 337        |
| 本章小结 .....                   | 338        |
| 本章习题 .....                   | 338        |
| 习题解答 .....                   | 338        |
| <b>第11章 赋值、表达式和打印</b> .....  | <b>340</b> |
| 赋值语句 .....                   | 340        |
| 赋值语句形式 .....                 | 341        |
| 序列赋值 .....                   | 342        |
| Python 3.X中的扩展序列解包 .....     | 345        |
| 多目标赋值 .....                  | 349        |
| 增量赋值 .....                   | 350        |
| 变量命名规则 .....                 | 353        |
| Python中的废弃协议 .....           | 355        |
| 表达式语句 .....                  | 356        |
| 表达式语句和原位置修改 .....            | 358        |
| 打印操作 .....                   | 358        |



|                                   |            |
|-----------------------------------|------------|
| Python 3.X的print函数 .....          | 359        |
| Python 2.X的print语句 .....          | 362        |
| 打印流重定向 .....                      | 363        |
| 版本中立的打印 .....                     | 367        |
| 为什么你要注意：print和stdout .....        | 369        |
| 本章小结 .....                        | 370        |
| 本章习题 .....                        | 370        |
| 习题解答 .....                        | 370        |
| <b>第12章 if测试和语法规则 .....</b>       | <b>372</b> |
| if语句 .....                        | 372        |
| 一般形式 .....                        | 372        |
| 基础示例 .....                        | 373        |
| 多路分支 .....                        | 373        |
| 复习Python语法规则 .....                | 375        |
| 代码块分隔符：缩进规则 .....                 | 377        |
| 语句分隔符：行与行间连接符 .....               | 378        |
| 一些特殊情况 .....                      | 379        |
| 真值和布尔测试 .....                     | 381        |
| if/else三元表达式 .....                | 383        |
| 请留意：布尔值 .....                     | 384        |
| 本章小结 .....                        | 386        |
| 本章习题 .....                        | 386        |
| 习题解答 .....                        | 386        |
| <b>第13章 while循环和for循环 .....</b>   | <b>387</b> |
| while循环 .....                     | 387        |
| 一般形式 .....                        | 387        |
| 示例 .....                          | 388        |
| break、continue、pass和循环的else ..... | 389        |
| 一般循环形式 .....                      | 389        |
| pass .....                        | 389        |
| continue .....                    | 391        |