

Python基础教程（第3版）

作者：Magnus Lie Hetland

关注微信公众号：[爱书小站](#)

[点击进入爱书小站](#)，你爱的书可能都在这。

历史传记，人文社科，励志成功，小说文学，金融理财各系列应用尽有。

公众号回复书号即可免费获得图书。

本书仅供个人学习之用，请勿用于商业用途。如果阅读后觉得好，请购买正版书籍！

Table of Contents

1. [版权信息](#)
2. [前言](#)
 1. [第2版前言](#)
 2. [第1版前言](#)
3. [引言](#)
4. [第1章 快速上手：基础知识](#)
 1. [1.1 交互式解释器](#)
 2. [1.2 算法是什么](#)
 3. [1.3 数和表达式](#)
 1. [十六进制、八进制和二进制](#)
 4. [1.4 变量](#)
 5. [1.5 语句](#)
 6. [1.6 获取用户输入](#)
 7. [1.7 函数](#)
 8. [1.8 模块](#)
 1. [1.8.1 cmath和复数](#)
 2. [1.8.2 回到未来](#)
 9. [1.9 保存并执行程序](#)
 1. [1.9.1 从命令提示符运行Python脚本](#)
 2. [1.9.2 让脚本像普通程序一样](#)
 3. [1.9.3 注释](#)
 10. [1.10 字符串](#)
 1. [1.10.1 单引号字符串以及对引号转义](#)
 2. [1.10.2 拼接字符串](#)
 3. [1.10.3 字符串表示str和repr](#)
 4. [1.10.4 长字符串、原始字符串和字节](#)
 11. [1.11 小结](#)
 1. [1.11.1 本章介绍的新函数](#)
 2. [1.11.2 预告](#)
5. [第2章 列表和元组](#)
 1. [2.1 序列概述](#)
 2. [2.2 通用的序列操作](#)
 1. [2.2.1 索引](#)
 2. [2.2.2 切片](#)
 3. [2.2.3 序列相加](#)
 4. [2.2.4 乘法](#)
 5. [2.2.5 成员资格](#)
 3. [2.3 列表：Python的主力](#)
 1. [2.3.1 函数list](#)
 2. [2.3.2 基本的列表操作](#)
 3. [2.3.3 列表方法](#)
 4. [2.4 元组：不可修改的序列](#)
 5. [2.5 小结](#)
 1. [2.5.1 本章介绍的新函数](#)
 2. [2.5.2 预告](#)
6. [第3章 使用字符串](#)
 1. [3.1 字符串基本操作](#)
 2. [3.2 设置字符串的格式：精简版](#)
 3. [3.3 设置字符串的格式：完整版](#)
 1. [3.3.1 替换字段名](#)
 2. [3.3.2 基本转换](#)
 3. [3.3.3 宽度、精度和千位分隔符](#)
 4. [3.3.4 符号、对齐和用0填充](#)
 4. [3.4 字符串方法](#)
 1. [3.4.1 center](#)
 2. [3.4.2 find](#)
 3. [3.4.3 join](#)
 4. [3.4.4 lower](#)
 5. [3.4.5 replace](#)
 6. [3.4.6 split](#)
 7. [3.4.7 strip](#)
 8. [3.4.8 translate](#)
 9. [3.4.9 判断字符串是否满足特定的条件](#)
 5. [3.5 小结](#)
 1. [3.5.1 本章介绍的新函数](#)
 2. [3.5.2 预告](#)
7. [第4章 当索引行不通时](#)
 1. [4.1 字典的用途](#)
 2. [4.2 创建和使用字典](#)
 1. [4.2.1 函数dict](#)
 2. [4.2.2 基本的字典操作](#)
 3. [4.2.3 将字符串格式设置功能用于字典](#)
 4. [4.2.4 字典方法](#)
 3. [4.3 小结](#)
 1. [4.3.1 本章介绍的新函数](#)
 2. [4.3.2 预告](#)
8. [第5章 条件、循环及其他语句](#)
 1. [5.1 再谈print和import](#)
 1. [5.1.1 打印多个参数](#)
 2. [5.1.2 导入时重命名](#)
 2. [5.2 赋值魔法](#)
 1. [5.2.1 序列解包](#)
 2. [5.2.2 链式赋值](#)
 3. [5.2.3 增强赋值](#)
 3. [5.3 代码块：缩进的乐趣](#)
 4. [5.4 条件和条件语句](#)
 1. [5.4.1 这正是布尔值的用武之地](#)
 2. [5.4.2 有条件地执行和if语句](#)
 3. [5.4.3 else子句](#)
 4. [5.4.4 elif子句](#)
 5. [5.4.5 代码块嵌套](#)
 6. [5.4.6 更复杂的条件](#)

- 7. [5.4.7 断言](#)
- 5. [5.5 循环](#)
 - 1. [5.5.1 while循环](#)
 - 2. [5.5.2 for循环](#)
 - 3. [5.5.3 迭代字典](#)
 - 4. [5.5.4 一些迭代工具](#)
 - 5. [5.5.5 跳出循环](#)
 - 6. [5.5.6 循环中的else子句](#)
- 6. [5.6 简单推导](#)
- 7. [5.7 三人行](#)
 - 1. [5.7.1 什么都不做](#)
 - 2. [5.7.2 使用del删除](#)
 - 3. [5.7.3 使用exec和eval执行字符串及计算其结果](#)
- 8. [5.8 小结](#)
 - 1. [5.8.1 本章介绍的新函数](#)
 - 2. [5.8.2 预告](#)
- 9. [第6章 抽象](#)
 - 1. [6.1 懒惰是一种美德](#)
 - 2. [6.2 抽象和结构](#)
 - 3. [6.3 自定义函数](#)
 - 1. [6.3.1 给函数编写文档](#)
 - 2. [6.3.2 其实并不是函数的函数](#)
 - 4. [6.4 参数魔法](#)
 - 1. [6.4.1 值从哪里来](#)
 - 2. [6.4.2 我能修改参数吗](#)
 - 3. [6.4.3 关键字参数和默认值](#)
 - 4. [6.4.4 收集参数](#)
 - 5. [6.4.5 分配参数](#)
 - 6. [6.4.6 练习使用参数](#)
 - 5. [6.5 作用域](#)
 - 6. [6.6 递归](#)
 - 1. [6.6.1 两个经典案例：阶乘和幂](#)
 - 2. [6.6.2 另一个经典案例：二分查找](#)
 - 7. [6.7 小结](#)
 - 1. [6.7.1 本章介绍的新函数](#)
 - 2. [6.7.2 预告](#)
- 10. [第7章 再谈抽象](#)
 - 1. [7.1 对象魔法](#)
 - 1. [7.1.1 多态](#)
 - 2. [7.1.2 多态和方法](#)
 - 3. [7.1.3 封装](#)
 - 4. [7.1.4 继承](#)
 - 2. [7.2 类](#)
 - 1. [7.2.1 类到底是什么](#)
 - 2. [7.2.2 创建自定义类](#)
 - 3. [7.2.3 属性、函数和方法](#)
 - 4. [7.2.4 再谈隐藏](#)
 - 5. [7.2.5 类的命名空间](#)
 - 6. [7.2.6 指定超类](#)
 - 7. [7.2.7 深入探讨继承](#)
 - 8. [7.2.8 多个超类](#)
 - 9. [7.2.9 接口和内省](#)
 - 10. [7.2.10 抽象基类](#)
 - 3. [7.3 关于面向对象设计的一些思考](#)
 - 4. [7.4 小结](#)
 - 1. [7.4.1 本章介绍的新函数](#)
 - 2. [7.4.2 预告](#)
- 11. [第8章 异常](#)
 - 1. [8.1 异常是什么](#)
 - 2. [8.2 让事情沿你指定的轨道出错](#)
 - 1. [8.2.1 raise语句](#)
 - 2. [8.2.2 自定义的异常类](#)
 - 3. [8.3 捕获异常](#)
 - 1. [8.3.1 不用提供参数](#)
 - 2. [8.3.2 多个except子句](#)
 - 3. [8.3.3 一箭双雕](#)
 - 4. [8.3.4 捕获对象](#)
 - 5. [8.3.5 一网打尽](#)
 - 6. [8.3.6 万事大吉时](#)
 - 7. [8.3.7 最后](#)
 - 4. [8.4 异常和函数](#)
 - 5. [8.5 异常之禅](#)
 - 6. [8.6 不那么异常的情况](#)
 - 7. [8.7 小结](#)
 - 1. [8.7.1 本章介绍的新函数](#)
 - 2. [8.7.2 预告](#)
- 12. [第9章 魔法方法、特性和迭代器](#)
 - 1. [9.1 如果你使用的不是Python 3](#)
 - 2. [9.2 构造函数](#)
 - 1. [9.2.1 重写普通方法和特殊的构造函数](#)
 - 2. [9.2.2 调用未关联的超类构造函数](#)
 - 3. [9.2.3 使用函数super](#)
 - 3. [9.3 元素访问](#)
 - 1. [9.3.1 基本的序列和映射协议](#)
 - 2. [9.3.2 从list、dict和str派生](#)
 - 4. [9.4 其他魔法方法](#)
 - 5. [9.5 特性](#)
 - 1. [9.5.1 函数property](#)
 - 2. [9.5.2 静态方法和类方法](#)
 - 3. [9.5.3 __getattr__、__setattr__等方法](#)
 - 6. [9.6 迭代器](#)
 - 1. [9.6.1 迭代器协议](#)
 - 2. [9.6.2 从迭代器创建序列](#)
 - 7. [9.7 生成器](#)
 - 1. [9.7.1 创建生成器](#)
 - 2. [9.7.2 递归式生成器](#)
 - 3. [9.7.3 通用生成器](#)

4. [9.7.4 生成器的方法](#)
5. [9.7.5 模拟生成器](#)
8. [9.8 八皇后问题](#)
 1. [9.8.1 生成器的回溯](#)
 2. [9.8.2 问题](#)
 3. [9.8.3 状态表示](#)
 4. [9.8.4 检测冲突](#)
 5. [9.8.5 基线条件](#)
 6. [9.8.6 递归条件](#)
 7. [9.8.7 归尾工作](#)
9. [9.9 小结](#)
 1. [9.9.1 本章介绍的新函数](#)
 2. [9.9.2 预告](#)
13. [第10章 开箱即用](#)
 1. [10.1 模块](#)
 1. [10.1.1 模块就是程序](#)
 2. [10.1.2 模块是用来定义下的](#)
 3. [10.1.3 让模块可用](#)
 4. [10.1.4 包](#)
 2. [10.2 探索模块](#)
 1. [10.2.1 模块包含什么](#)
 2. [10.2.2 使用help获取帮助](#)
 3. [10.2.3 文档](#)
 4. [10.2.4 使用源代码](#)
 3. [10.3 标准库：一些受欢迎的模块](#)
 1. [10.3.1 sys](#)
 2. [10.3.2 os](#)
 3. [10.3.3 fileinput](#)
 4. [10.3.4 集合、堆和两端队列](#)
 5. [10.3.5 time](#)
 6. [10.3.6 random](#)
 7. [10.3.7 shelve和json](#)
 8. [10.3.8 re](#)
 9. [10.3.9 其他有趣的标准模块](#)
 4. [10.4 小结](#)
 1. [10.4.1 本章介绍的新函数](#)
 2. [10.4.2 预告](#)
14. [第11章 文件](#)
 1. [11.1 打开文件](#)
 1. [文件模式](#)
 2. [11.2 文件的基本方法](#)
 1. [11.2.1 读取和写入](#)
 2. [11.2.2 使用管道重定向输出](#)
 3. [11.2.3 读取和写入门](#)
 4. [11.2.4 关闭文件](#)
 5. [11.2.5 使用文件的基本方法](#)
 3. [11.3 迭代文件内容](#)
 1. [11.3.1 每次一个字符（或字节）](#)
 2. [11.3.2 每次一行](#)
 3. [11.3.3 读取所有内容](#)
 4. [11.3.4 使用fileinput实现延迟迭代](#)
 5. [11.3.5 文件迭代器](#)
 4. [11.4 小结](#)
 1. [11.4.1 本章介绍的新函数](#)
 2. [11.4.2 预告](#)
15. [第12章 图形用户界面](#)
 1. [12.1 创建GUI示例应用程序](#)
 1. [12.1.1 初探](#)
 2. [12.1.2 布局](#)
 3. [12.1.3 事件处理](#)
 4. [12.1.4 最终的程序](#)
 2. [12.2 使用其他GUI工具包](#)
 3. [12.3 小结](#)
 1. [预告](#)
16. [第13章 数据库支持](#)
 1. [13.1 Python数据库API](#)
 1. [13.1.1 全局变量](#)
 2. [13.1.2 异常](#)
 3. [13.1.3 连接和游标](#)
 4. [13.1.4 类型](#)
 2. [13.2 SQLite和PySQLite](#)
 1. [13.2.1 起步](#)
 2. [13.2.2 数据库应用程序示例](#)
 3. [13.3 小结](#)
 1. [13.3.1 本章介绍的新函数](#)
 2. [13.3.2 预告](#)
17. [第14章 网络编程](#)
 1. [14.1 几个网络模块](#)
 1. [14.1.1 模块socket](#)
 2. [14.1.2 模块urllib和urllib2](#)
 3. [14.1.3 其他模块](#)
 2. [14.2 SocketServer及相关的类](#)
 3. [14.3 多个连接](#)
 1. [14.3.1 使用SocketServer实现分叉和线程化](#)
 2. [14.3.2 使用select和poll实现异步I/O](#)
 4. [14.4 Twisted](#)
 1. [14.4.1 下载并安装Twisted](#)
 2. [14.4.2 编写Twisted服务器](#)
 5. [14.5 小结](#)
 1. [14.5.1 本章介绍的新函数](#)
 2. [14.5.2 预告](#)
18. [第15章 Python和Web](#)
 1. [15.1 屏幕抓取](#)
 1. [15.1.1 Tkidy和XHTML解析](#)
 2. [15.1.2 BeautifulSoup](#)
 2. [15.2 使用CGI创建动态网页](#)
 1. [15.2.1 第一步：准备Web服务器](#)

2. [15.2.2 第二步：添加#行](#)
 3. [15.2.3 第三步：设置文件权限](#)
 4. [15.2.4 CGI安全风险](#)
 5. [15.2.5 简单的CGI脚本](#)
 6. [15.2.6 使用cgitb进行调试](#)
 7. [15.2.7 使用模块cgi](#)
 8. [15.2.8 简单的表单](#)
 3. [15.3 使用Web框架](#)
 1. [其他Web应用框架](#)
 4. [15.4 Web服务：更高级的抓取](#)
 1. [15.4.1 RSS和相关内容](#)
 2. [15.4.2 使用XML-RPC进行远程过程调用](#)
 3. [15.4.3 SOAP](#)
 5. [15.5 小结](#)
 1. [15.5.1 本章介绍的新函数](#)
 2. [15.5.2 预告](#)
 19. [第16章 测试基础](#)
 1. [16.1 先测试再编码](#)
 1. [16.1.1 准确的需求说明](#)
 2. [16.1.2 做好应对变化的准备](#)
 3. [16.1.3 测试四步曲](#)
 2. [16.2 测试工具](#)
 1. [16.2.1 doctest](#)
 2. [16.2.2 unittest](#)
 3. [16.3 超越单元测试](#)
 1. [16.3.1 使用PyChecker和PyLint检查源代码](#)
 2. [16.3.2 性能分析](#)
 4. [16.4 小结](#)
 1. [16.4.1 本章介绍的新函数](#)
 2. [16.4.2 预告](#)
 20. [第17章 扩展Python](#)
 1. [17.1 鱼和熊掌兼得](#)
 2. [17.2 简单易行的方式：Jython和IronPython](#)
 3. [17.3 编写C语言扩展](#)
 1. [17.3.1 SWIG](#)
 2. [17.3.2 手工编写扩展](#)
 4. [17.4 小结](#)
 1. [17.4.1 本章介绍的新函数](#)
 2. [17.4.2 预告](#)
 21. [第18章 程序打包](#)
 1. [18.1 Setuptools基础](#)
 2. [18.2 打包](#)
 3. [18.3 编译扩展](#)
 4. [18.4 使用py2exe创建可执行程序](#)
 5. [18.5 小结](#)
 1. [18.5.1 本章介绍的新函数](#)
 2. [18.5.2 预告](#)
 22. [第19章 趣味编程](#)
 1. [19.1 为何要有趣](#)
 2. [19.2 编程艺术](#)
 3. [19.3 原型设计](#)
 4. [19.4 配置](#)
 1. [19.4.1 提取常量](#)
 2. [19.4.2 配置文件](#)
 5. [19.5 日志](#)
 6. [19.6 如果你已不胜其烦](#)
 7. [19.7 如果你想深入学习](#)
 8. [19.8 小结](#)
 1. [预告](#)
 23. [第20章 项目1：自动添加标签](#)
 1. [20.1 问题描述](#)
 2. [20.2 有用的工具](#)
 3. [20.3 准备工作](#)
 4. [20.4 初次实现](#)
 1. [20.4.1 找出文本块](#)
 2. [20.4.2 添加一些标记](#)
 5. [20.5 再次实现](#)
 1. [20.5.1 处理程序](#)
 2. [20.5.2 处理程序的超类](#)
 3. [20.5.3 规则](#)
 4. [20.5.4 规则的超类](#)
 5. [20.5.5 过滤器](#)
 6. [20.5.6 解析器](#)
 7. [20.5.7 创建规则和过滤器](#)
 8. [20.5.8 整合起来](#)
 6. [20.6 进一步探索](#)
 1. [预告](#)
 24. [第21章 项目2：绘制图表](#)
 1. [21.1 问题描述](#)
 2. [21.2 有用的工具](#)
 3. [21.3 准备工作](#)
 4. [21.4 初次实现](#)
 1. [21.4.1 使用ReportLab绘图](#)
 2. [21.4.2 绘制折线](#)
 3. [21.4.3 编写原型](#)
 5. [21.5 再次实现](#)
 1. [21.5.1 获取数据](#)
 2. [21.5.2 使用LinePlot类](#)
 6. [21.6 进一步探索](#)
 1. [预告](#)
 25. [第22章 项目3：万能的XML](#)
 1. [22.1 问题描述](#)
 2. [22.2 有用的工具](#)
 3. [22.3 准备工作](#)
 4. [22.4 初次实现](#)
 1. [22.4.1 创建简单的内容处理程序](#)
 2. [22.4.2 创建HTML页面](#)

5. [22.5 再次实现](#)
 1. [22.5.1 分派器混合类](#)
 2. [22.5.2 将头部和尾部写入文件的方法以及默认处理程序](#)
 3. [22.5.3 支持目录](#)
 4. [22.5.4 事件处理程序](#)
6. [22.6 进一步探索](#)
 1. [预告](#)
26. [第 23 章 项目4：新闻汇总](#)
 1. [23.1 问题描述](#)
 2. [23.2 有用的工具](#)
 3. [23.3 准备工作](#)
 4. [23.4 初次实现](#)
 5. [23.5 再次实现](#)
 6. [23.6 进一步探索](#)
 1. [预告](#)
27. [第 24 章 项目5：虚拟茶话会](#)
 1. [24.1 问题描述](#)
 2. [24.2 有用的工具](#)
 3. [24.3 准备工作](#)
 4. [24.4 初次实现](#)
 1. [24.4.1 ChatServer类](#)
 2. [24.4.2 ChatSession类](#)
 3. [24.4.3 整合起来](#)
 5. [24.5 再次实现](#)
 1. [24.5.1 基本的命令解释功能](#)
 2. [24.5.2 聊天室](#)
 3. [24.5.3 登录和退出聊天室](#)
 4. [24.5.4 主聊天室](#)
 5. [24.5.5 新的服务器](#)
 6. [24.6 进一步探索](#)
 1. [预告](#)
28. [第 25 章 项目6：使用CGI进行远程编辑](#)
 1. [25.1 问题描述](#)
 2. [25.2 有用的工具](#)
 3. [25.3 准备工作](#)
 4. [25.4 初次实现](#)
 5. [25.5 再次实现](#)
 1. [25.5.1 创建文件名表单](#)
 2. [25.5.2 编写编辑器脚本](#)
 3. [25.5.3 编写保存脚本](#)
 4. [25.5.4 运行编辑器](#)
 6. [25.6 进一步探索](#)
 1. [预告](#)
29. [第 26 章 项目7：自建公告板](#)
 1. [26.1 问题描述](#)
 2. [26.2 有用的工具](#)
 3. [26.3 准备工作](#)
 4. [26.4 初次实现](#)
 5. [26.5 再次实现](#)
 1. [26.5.1 编写脚本main.cgi](#)
 2. [26.5.2 编写脚本view.cgi](#)
 3. [26.5.3 编写脚本edit.cgi](#)
 4. [26.5.4 编写脚本save.cgi](#)
 5. [26.5.5 尝试使用](#)
 6. [26.6 进一步探索](#)
 1. [预告](#)
30. [第 27 章 项目8：使用XML-RPC共享文件](#)
 1. [27.1 问题描述](#)
 2. [27.2 有用的工具](#)
 3. [27.3 准备工作](#)
 4. [27.4 初次实现](#)
 1. [27.4.1 实现简单的节点](#)
 2. [27.4.2 尝试使用](#)
 5. [27.5 再次实现](#)
 1. [27.5.1 创建客户端界面](#)
 2. [27.5.2 引发异常](#)
 3. [27.5.3 验证文件名](#)
 4. [27.5.4 尝试使用](#)
 6. [27.6 进一步探索](#)
 1. [预告](#)
31. [第 28 章 项目9：使用GUI共享文件](#)
 1. [28.1 问题描述](#)
 2. [28.2 有用的工具](#)
 3. [28.3 准备工作](#)
 4. [28.4 初次实现](#)
 5. [28.5 再次实现](#)
 6. [28.6 进一步探索](#)
 1. [预告](#)
32. [第 29 章 项目10：自制街机游戏](#)
 1. [29.1 问题描述](#)
 2. [29.2 有用的工具](#)
 1. [29.2.1 pygame](#)
 2. [29.2.2 pygame.locals](#)
 3. [29.2.3 pygame.display](#)
 4. [29.2.4 pygame.font](#)
 5. [29.2.5 pygame.sprite](#)
 6. [29.2.6 pygame.mouse](#)
 7. [29.2.7 pygame.event](#)
 8. [29.2.8 pygame.image](#)
 3. [29.3 准备工作](#)
 4. [29.4 初次实现](#)
 5. [29.5 再次实现](#)
 6. [29.6 进一步探索](#)
 1. [预告](#)
33. [附录 A 简明教程](#)
 1. [A.1 基础知识](#)
 2. [A.2 函数](#)

3. [A.3 对象及相关内容](#)
4. [A.4 知识点补充](#)
34. [附录 B Python参考手册](#)
 1. [B.1 表达式](#)
 2. [B.2 语句](#)
 1. [B.2.1 简单语句](#)
 2. [B.2.2 复合语句](#)
35. [看完了](#)

版权信息

书名: Python基础教程 (第3版)

作者: [挪] Magnus Lie Hetland 译者: 袁国忠

ISBN: 978-7-115-47488-9

本书由北京图灵文化发展有限公司发行数字版。版权所有，侵权必究。

您购买的图灵电子书仅供您个人使用，未经授权，不得以任何方式复制和传播本书内容。

我们愿意相信读者具有这样的良知和觉悟，与我们共同保护知识产权。

如果购买者有侵权行为，我们可能对该用户实施包括但不限于关闭该帐号等维权措施，并可能追究法律责任。

版权声明

Original English language edition, entitled *Beginning Python: From Novice to Professional, Third Edition* by Magnus Lie Hetland, published by Apress, 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705 USA.

Copyright © 2017 by Magnus Lie Hetland. Simplified Chinese-language edition copyright © 2018 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由Apress L. P.授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

前言

巨蟒剧团1 有首老歌唱道：“又来了一个，它走了又来。又来了一个，什么时候是个头？”自本书第2版面世以来，Python 3 的普及程度得到了极大提高，因此这一版完全转向了Python 3。在此期间，还有其他的变化：在Python生态系统中，各种包轮番登场，各种编码实践大行其道后又日渐式微。在必要之处或对读者有帮助的前提下，本书的有些部分完全重写，但最初的痕迹还依稀可见。例如，本书前身*Practical Python* 于21世纪初出版时，Usenet依然应用广泛，可现在大多数互联网用户可能从未听说过它。因此，对于第23章创建连接到NNTP服务器的项目，与其说是为了介绍主流编程生涯中将用到的编程技能，不如说是让读者了解一下历史。一些比较古怪的内容也依旧保留，因为它们是很不错的编程示例，也是本书悠长历史的见证。

1 Monty Python，英国的一个超现实幽默表演团体，喜剧界的披头士，在20世纪70年代风靡全球。Python语言的命名来源于此。——编者注

与以往一样，我要深深地感谢让本书以前各版得以付梓的人。这里我要特别感谢Mark Powers，感谢他在我进度缓慢时极具耐心。还要感谢Michael Thomas，感谢他出色的技术审阅工作（并指出了原稿中所有的Python 2式`print` 语句）。但愿你喜欢全新升级后的这一版。正如Terry Jones在提到那首老歌时所说：“显然，有一个完整的管弦乐队会更好。”

第2版前言

新版的《Python基础教程》终于和大家见面了。如果算上本书的前身*Practical Python*，实际上这已经是第3版了。这是我将近10年心血的结晶。在此期间，Python发生了很多有趣的变化，我也尽力调整了对这门语言的介绍。当前，Python处于长期以来最为翻天覆地的转变期：Python 3推出。编写本书期间，最终版本还未发布，但其包含的功能已确定，还推出了多个可用的版本。这次修订不向后兼容，这是我编写这一版时面临的最大挑战。换言之，我不能仅仅挑出一些新增的功能进行介绍。另外，它还修改了一些既有的功能，因此有些在Python 2.5中理所当然的事情在新版本中已不再正确。

倘若整个Python社区都立即转向Python 3并修改所有的遗留代码，那根本不成问题。我只需介绍Python 3就行！然而，目前依然存在大量用Python 2编写的代码，而且大家可能还会继续编写这样的代码，直到有一天所有人都认为应使用Python 3。

那么我是如何应对这种变化的呢？首先，虽然有些向后兼容的变化，但Python语言本身总体变化不大。因此，如果完全针对Python 2.5，编写的内容对Python 3来说也是大体正确的（对Python 2.6来说更是如此）。至于那些不再正确的部分，我采取了比较保守的态度，因为大家完全接受Python 3还需要一段时间。因此，这一版主要是基于Python 2.5编写的，同时指出将会改变的情形。另外，我还提供了附录D，让你对重大变化有大致地了解。对大多数读者来说，这样处理是完全可行的。

编写这一版期间，有几个人帮了我很大的忙。与前两版（本书第1版以及之前的*Practical Python*）一样，Jason Gilmore扶我上马，并在项目启动期间扮演了重要角色。项目启动后，Richard Dal Porto、Frank Pohlmann和Dominic Shakeshaft不断伸出援手，保证项目得以顺利进行。在确保代码正确方面，Richard Taylor居功至伟（倘若代码依然存在错误，责任在我），而Marilyn Smith出色地完成了手稿润色工作。这里还要感谢Apress出版社的其他工作人员，他们是Liz Berry、Beth Christmas、Steve Anglin和Tina Nielsen。另外，感谢包括Bob Heinbold和Waclaw Kusnierczyk在内的读者提出勘误和宝贵的建议。最后，在本书前两版的出版过程中，很多人都提供了帮助，这里衷心地感谢他们。

欢迎访问：电子书学习和下载网站 (<https://www.shgis.cn>)

文档名称：《Python基础教程》MagnusLieHetland.epub

请登录 <https://shgis.cn/post/1844.html> 下载完整文档。

手机端请扫码查看：

