

Python核心编程（第3版）

作者：[美] Wesley Chun

目录

封面

扉页

版权

内容提要

本书赞誉

关于作者

前言

第1部分 通用应用主题

第1章 正则表达式

1.1 简介/动机

1.2 特殊符号和字符

1.2.1 使用择一匹配符号匹配多个正则表达式模式

1.2.2 匹配任意单个字符

1.2.3 从字符串起始或者结尾或者单词边界匹配

1.2.4 创建字符集

1.2.5 限定范围和否定

1.2.6 使用闭包操作符实现存在性和频数匹配

1.2.7 表示字符集的特殊字符

1.2.8 使用圆括号指定分组

1.2.9 扩展表示法

1.3 正则表达式和Python语言

1.3.1 re模块：核心函数和方法

1.3.2 使用compile()函数编译正则表达式

1.3.3 匹配对象以及group()和groups()方法

1.3.4 使用match()方法匹配字符串

1.3.5 使用search()在一个字符串中查找模式（搜索与匹配的对比）

1.3.6 匹配多个字符串

1.3.7 匹配任何单个字符

1.3.8 创建字符集（[]）

1.3.9 重复、特殊字符以及分组

1.3.10 匹配字符串的起始和结尾以及单词边界

1.3.11 使用findall()和finditer()查找每一次出现的位置

1.3.12 使用sub()和subn()搜索与替换

1.3.13 在限定模式上使用split()分隔字符串

1.3.14 扩展符号

1.3.15 杂项

1.4 一些正则表达式示例

1.5 更长的正则表达式示例

1.5.1 匹配字符串

1.5.2 搜索与匹配……还有贪婪

1.6 练习

第2章 网络编程

2.1 简介

2.2 客户端/服务器架构

2.2.1 硬件客户端/服务器架构

2.2.2 软件客户端/服务器架构

2.2.3 银行出纳员作为服务器吗

2.2.4 客户端/服务器网络编程

2.3 套接字：通信端点

2.3.1 套接字

2.3.2 套接字地址：主机-端口对

2.3.3 面向连接的套接字与无连接的套接字

2.4 Python中的网络编程

2.4.1 socket()模块函数

2.4.2 套接字对象（内置）方法

2.4.3 创建TCP服务器

2.4.4 创建TCP客户端

2.4.5 执行TCP服务器和客户端

2.4.6 创建UDP服务器

2.4.7 创建UDP客户端

2.4.8 执行UDP服务器和客户端

2.4.9 socket模块属性

2.5 *SocketServer模块

2.5.1 创建SocketServer TCP服务器

2.5.2 创建SocketServer TCP客户端

2.5.3 执行TCP服务器和客户端

2.6 *Twisted框架介绍

2.6.1 创建Twisted Reactor TCP服务器

2.6.2 创建Twisted Reactor TCP客户端

2.6.3 执行TCP服务器和客户端

2.7 相关模块

2.8 练习

第3章 因特网客户端编程

3.1 因特网客户端简介

- 3.2 文件传输
 - 3.2.1 文件传输因特网协议
 - 3.2.2 文件传输协议
 - 3.2.3 Python和FTP
 - 3.2.4 ftplib.FTP类的方法
 - 3.2.5 交互式FTP示例
 - 3.2.6 客户端FTP程序示例
 - 3.2.7 FTP的其他内容
- 3.3 网络新闻
 - 3.3.1 Usenet与新闻组
 - 3.3.2 网络新闻传输协议
 - 3.3.3 Python和NNTP
 - 3.3.4 nntplib.NNTP类方法
 - 3.3.5 交互式NNTP示例
 - 3.3.6 客户端程序NNTP示例
 - 3.3.7 NNTP的其他内容
- 3.4 电子邮件
 - 3.4.1 电子邮件系统组件和协议
 - 3.4.2 发送电子邮件
 - 3.4.3 Python和SMTP
 - 3.4.4 smtplib.SMTP类方法
 - 3.4.5 交互式SMTP示例
 - 3.4.6 SMTP的其他内容
 - 3.4.7 接收电子邮件
 - 3.4.8 POP和IMAP
 - 3.4.9 Python和POP3
 - 3.4.10 交互式POP3示例
 - 3.4.11 poplib.POP3类方法
 - 3.4.12 客户端程序SMTP和POP3示例
 - 3.4.13 Python和IMAP4
 - 3.4.14 交互式IMAP4示例
 - 3.4.15 imaplib.IMAP4类中的常用方法
- 3.5 实战
 - 3.5.1 生成电子邮件
 - 3.5.2 解析电子邮件
 - 3.5.3 基于Web的云电子邮件服务
 - 3.5.4 最佳实践：安全、重构
 - 3.5.5 Yahoo! Mail
 - 3.5.6 Gmail
- 3.6 相关模块
 - 3.6.1 电子邮件
 - 3.6.2 其他因特网客户端协议
- 3.7 练习
- 第4章 多线程编程
 - 4.1 简介/动机
 - 4.2 线程和进程
 - 4.2.1 进程
 - 4.2.2 线程
 - 4.3 线程和Python
 - 4.3.1 全局解释器锁
 - 4.3.2 退出线程
 - 4.3.3 在Python中使用线程
 - 4.3.4 不使用线程的情况
 - 4.3.5 Python的threading模块
 - 4.4 thread模块
 - 4.5 threading模块
 - 4.5.1 Thread类
 - 4.5.2 threading模块的其他函数
 - 4.6 单线程和多线程执行对比
 - 4.7 多线程实践
 - 4.7.1 图书排名示例
 - 4.7.2 同步原语
 - 4.7.3 锁示例
 - 4.7.4 信号量示例
 - 4.8 生产者-消费者问题和Queue/queue模块
 - 4.9 线程的替代方案
 - 4.9.1 subprocess模块
 - 4.9.2 multiprocessing模块
 - 4.9.3 concurrent.futures模块
 - 4.10 相关模块
 - 4.11 练习
- 第5章 GUI编程
 - 5.1 简介
 - 5.1.1 Tcl、Tk和Tkinter
 - 5.1.2 安装和使用Tkinter
 - 5.1.3 客户端/服务端架构
 - 5.2 Tkinter和Python编程
 - 5.2.1 Tkinter模块：添加Tk到应用中
 - 5.2.2 GUI编程介绍
 - 5.2.3 顶层窗口：Tkinter.Tk()

- [5.2.4 Tk控件](#)
- [5.3 Tkinter示例](#)
- [5.3.1 Label控件](#)
- [5.3.2 Button控件](#)
- [5.3.3 Label和Button控件](#)
- [5.3.4 Label、Button和Scale控件](#)
- [5.3.5 偏函数应用示例](#)
- [5.3.6 中级Tkinter示例](#)
- [5.4 其他GUI简介](#)
- [5.4.1 Tk接口扩展 \(Tk\)](#)
- [5.4.2 Python MegaWidgets \(PMW\)](#)
- [5.4.3 wxWidgets和wxPython](#)
- [5.4.4 GTK+和PyGTK](#)
- [5.4.5 Tile/Tk](#)
- [5.5 相关模块和其他GUI](#)
- [5.6 练习](#)
- [第6章 数据库编程](#)
- [6.1 简介](#)
- [6.1.1 持久化存储](#)
- [6.1.2 数据库基本操作和SQL](#)
- [6.1.3 数据库和Python](#)
- [6.2 Python的DB-API](#)
- [6.2.1 模块属性](#)
- [6.2.2 Connection对象](#)
- [6.2.3 Cursor对象](#)
- [6.2.4 类型对象和构造函数](#)
- [6.2.5 关系数据库](#)
- [6.2.6 数据库和Python: 适配器](#)
- [6.2.7 使用数据库适配器的示例](#)
- [6.2.8 数据库适配器示例应用](#)
- [6.3 ORM](#)
- [6.3.1 考虑对象, 而不是SQL](#)
- [6.3.2 Python和ORM](#)
- [6.3.3 员工角色数据库示例](#)
- [6.3.4 SQLAlchemy](#)
- [6.3.5 SQLAlchemy](#)
- [6.4 非关系数据库](#)
- [6.4.1 NoSQL介绍](#)
- [6.4.2 MongoDB](#)
- [6.4.3 PyMongo: MongoDB和Python](#)
- [6.4.4 总结](#)
- [6.5 相关文献](#)
- [6.6 练习](#)
- [第7章 *Microsoft Office编程](#)
- [7.1 简介](#)
- [7.2 使用Python进行COM客户端编程](#)
- [7.2.1 客户端COM编程](#)
- [7.2.2 入门](#)
- [7.3 入门示例](#)
- [7.3.1 Excel](#)
- [7.3.2 Word](#)
- [7.3.3 PowerPoint](#)
- [7.3.4 Outlook](#)
- [7.4 中级示例](#)
- [7.4.1 Excel](#)
- [7.4.2 Outlook](#)
- [7.4.3 PowerPoint](#)
- [7.4.4 总结](#)
- [7.5 相关模块/包](#)
- [7.6 练习](#)
- [第8章 扩展Python](#)
- [8.1 简介和动机](#)
- [8.1.1 Python扩展简介](#)
- [8.1.2 什么情况下需要扩展Python](#)
- [8.1.3 什么情况下不应该扩展Python](#)
- [8.2 编写Python扩展](#)
- [8.2.1 创建应用代码](#)
- [8.2.2 根据样板编写封装代码](#)
- [8.2.3 编译](#)
- [8.2.4 导入并测试](#)
- [8.2.5 引用计数](#)
- [8.2.6 线程和全局解释器锁](#)
- [8.3 相关主题](#)
- [8.3.1 SWIG](#)
- [8.3.2 Pyrex](#)
- [8.3.3 Cython](#)
- [8.3.4 Psyco](#)
- [8.3.5 PyPy](#)
- [8.3.6 嵌入Python](#)
- [8.4 练习](#)

[第2部分 Web开发](#)

[第9章 Web客户端和服务端](#)

[9.1 简介](#)

[9.1.1 Web应用：客户端/服务器计算](#)

[9.1.2 因特网](#)

[9.2 Python Web客户端工具](#)

[9.2.1 统一资源定位符](#)

[9.2.2 urlparse模块](#)

[9.2.3 urllib模块/包](#)

[9.2.4 使用urllib2 HTTP验证的示例](#)

[9.2.5 将HTTP验证示例移植到Python 3中](#)

[9.3 Web客户端](#)

[9.3.1 一个简单的Web爬虫/蜘蛛/机器人](#)

[9.3.2 解析Web页面](#)

[9.3.3 可编程的Web浏览](#)

[9.4 Web \(HTTP\) 服务器](#)

[9.5 相关模块](#)

[9.6 练习](#)

[第10章 Web编程：CGI和WSGI](#)

[10.1 简介](#)

[10.2 帮助Web服务器处理客户端数据](#)

[10.2.1 CGI简介](#)

[10.2.2 CGI应用程序](#)

[10.2.3 cgi模块](#)

[10.2.4 cgiib模块](#)

[10.3 构建CGI应用程序](#)

[10.3.1 构建Web服务器](#)

[10.3.2 建立表单页](#)

[10.3.3 生成结果页面](#)

[10.3.4 生成表单和结果页面](#)

[10.3.5 全面交互的Web站点](#)

[10.4 在CGI中使用Unicode](#)

[10.5 高级CGI](#)

[10.5.1 multipart表单提交和文件上传](#)

[10.5.2 多值字段](#)

[10.5.3 cookie](#)

[10.5.4 cookie和文件上传](#)

[10.6 WSGI简介](#)

[10.6.1 动机（替代CGI）](#)

[10.6.2 服务器集成](#)

[10.6.3 外部进程](#)

[10.6.4 WSGI简介](#)

[10.6.5 WSGI服务器](#)

[10.6.6 参考服务器](#)

[10.6.7 WSGI应用示例](#)

[10.6.8 中间件及封装WSGI应用](#)

[10.6.9 在Python 3中使用WSGI](#)

[10.7 现实世界中的Web开发](#)

[10.8 相关模块](#)

[10.9 练习](#)

[第11章 Web框架：Django](#)

[11.1 简介](#)

[11.2 Web框架](#)

[11.3 Django简介](#)

[11.4 项目和应用](#)

[11.4.1 在Django中创建项目](#)

[11.4.2 运行开发服务器](#)

[11.5“Hello World”应用（一个博客）](#)

[11.6 创建模型来添加数据库服务](#)

[11.6.1 设置数据库](#)

[11.6.2 创建表](#)

[11.7 Python 应用shell](#)

[11.7.1 在Django中使用Python shell](#)

[11.7.2 测试数据模型](#)

[11.8 Django管理应用](#)

[11.8.1 设置admin](#)

[11.8.2 使用admin](#)

[11.9 创建博客的用户界面](#)

[11.9.1 创建模板](#)

[11.9.2 创建URL模式](#)

[11.9.3 创建视图函数](#)

[11.10 改进输出](#)

[11.11 处理用户输入](#)

[11.11.1 模板：添加HTML表单](#)

[11.11.2 添加URLcon项](#)

[11.11.3 视图：处理用户输入](#)

[11.11.4 跨站点请求伪造](#)

[11.12 表单和模型表单](#)

[11.12.1 Django表单简介](#)

[11.12.2 模型表单示例](#)

- [11.12.3 使用ModelForm来生成HTML表单](#)
- [11.12.4 处理ModelForm数据](#)
- [11.13 视图进阶](#)
- [11.14 *改善外观](#)
- [11.15 *单元测试](#)
- [11.15.1 blog应用的代码审查](#)
- [11.15.2 blog应用总结](#)
- [11.16 *中级Django应用: TweetApprover](#)
- [11.16.1 创建项目文件结构](#)
- [11.16.2 安装Twython库](#)
- [11.16.3 URL结构](#)
- [11.16.4 数据模型](#)
- [11.16.5 提交新推文以便审核](#)
- [11.16.6 审核推文](#)
- [11.17 资源](#)
- [11.18 总结](#)
- [11.19 练习](#)
- [第12章 云计算: Google App Engine](#)
- [12.1 简介](#)
- [12.2 云计算](#)
- [12.2.1 云计算服务的层次](#)
- [12.2.2 App Engine](#)
- [12.3 沙盒和App Engine SDK](#)
- [12.4 选择一个App Engine框架](#)
- [12.5 Python 2.7支持](#)
- [12.5.1 一般差异](#)
- [12.5.2 代码中的差异](#)
- [12.6 与Django比较](#)
- [12.6.1 开始“Hello World”](#)
- [12.6.2 手动创建“Hello World”\(Zip文件用户\)](#)
- [12.7 将“Hello World”改成一个简单的博客](#)
- [12.7.1 快速发现改动: 30秒内将纯文本转成HTML](#)
- [12.7.2 添加表单](#)
- [12.7.3 添加Datastore服务](#)
- [12.7.4 迭代改进](#)
- [12.7.5 开发/SDK控制台](#)
- [12.8 添加Memcache服务](#)
- [12.9 静态文件](#)
- [12.10 添加用户服务](#)
- [12.10.1 Google账号验证](#)
- [12.10.2 联合验证](#)
- [12.11 远程API shell](#)
- [12.12 问与答 \(Python实现\)](#)
- [12.12.1 发送电子邮件](#)
- [12.12.2 接收电子邮件](#)
- [12.13 使用XMPP发送即时消息](#)
- [12.14 处理图片](#)
- [12.15 任务队列 \(非定期任务\)](#)
- [12.15.1 创建任务](#)
- [12.15.2 配置app.yaml](#)
- [12.15.3 其他任务创建和配置选项](#)
- [12.15.4 将发送电子邮件作为任务](#)
- [12.15.5 deferred包](#)
- [12.16 使用Appstats进行分析](#)
- [12.16.1 在app.yaml中添加标准处理程序](#)
- [12.16.2 添加自定义Admin Console页面](#)
- [12.16.3 作为内置界面启用界面](#)
- [12.17 URL fetch服务](#)
- [12.18 问与答 \(无Python实现\)](#)
- [12.18.1 Cron服务 \(计划任务作业\)](#)
- [12.18.2 预热请求](#)
- [12.18.3 DoS保护](#)
- [12.19 厂商锁定](#)
- [12.20 资源](#)
- [12.21 总结](#)
- [12.22 练习](#)
- [第13章 Web服务](#)
- [13.1 简介](#)
- [13.2 Yahoo!金融股票报价服务器](#)
- [13.3 Twitter微博](#)
- [13.3.1 社交网络](#)
- [13.3.2 Twitter和Python](#)
- [13.3.3 稍微长一点的API组合应用示例](#)
- [13.3.4 总结](#)
- [13.3.5 额外在线资源](#)
- [13.4 练习](#)
- [第3部分 补充/实验章节](#)
- [第14章 文本处理](#)
- [14.1 逗号分隔值 \(CSV\)](#)
- [14.1.1 CSV简介](#)

- [14.1.2 再论股票投资组合示例](#)
- [14.2 JSON](#)
- [14.3 可扩展标记语言](#)
- [14.3.1 XML简介](#)
- [14.3.2 Python和XML](#)
- [14.3.3 XML实战](#)
- [14.3.4 *使用XML-RPC的客户端-服务器服务](#)
- [14.4 参考文献](#)
- [14.5 相关模块](#)
- [14.6 练习](#)
- [第15章 其他内容](#)
- [15.1 Jython](#)
- [15.1.1 Jython简介](#)
- [15.1.2 Swing GUI开发示例](#)
- [15.2 Google+](#)
- [15.2.1 Google+平台简介](#)
- [15.2.2 Python和Google+API](#)
- [15.2.3 一个简单的社交媒体分析工具](#)
- [15.3 练习](#)
- [附录A 部分练习参考答案](#)
- [附录B 参考表](#)
- [B.1 Python关键字](#)
- [B.2 Python标准操作符和函数](#)
- [B.3 数值类型操作符和函数](#)
- [B.4 序列类型操作符和函数](#)
- [B.5 字符串格式化操作符转换符号](#)
- [B.6 字符串格式化操作符指令](#)
- [B.7 字符串类型内置方法](#)
- [B.8 列表类型内置方法](#)
- [B.9 字典类型内置方法](#)
- [B.10 集合类型操作符和内置函数](#)
- [B.11 文件对象方法和数据属性](#)
- [B.12 Python异常](#)
- [B.13 类的特殊方法](#)
- [B.14 Python操作符汇总](#)
- [附录C Python 3：一种编程语言进化的产物](#)
- [C.1 为何Python在变化](#)
- [C.2 都发生了哪些变化](#)
- [C.2.1 print变成了print\(\)](#)
- [C.2.2 字符串：默认为Unicode编码](#)
- [C.2.3 单类类型](#)
- [C.2.4 更新异常的语法](#)
- [C.2.5 整数的更新](#)
- [C.2.6 迭代器无处不在](#)
- [C.3 迁移工具](#)
- [C.3.1 2to3工具](#)
- [C.3.2 Python 2.6+](#)
- [C.4 结论](#)
- [C.5 参考资料](#)
- [附录D 利用Python 2.6+向Python 3迁移](#)
- [D.1 Python 3：Python的下一代](#)
- [D.2 整数](#)
- [D.2.1 单整数类型](#)
- [D.2.2 新型二进制和改进的八进制字符](#)
- [D.2.3 经典除法或真除法](#)
- [D.3 内置函数](#)
- [D.3.1 print语句或print\(\)函数](#)
- [D.3.2 reduce\(\)转移到了functools模块中](#)
- [D.3.3 其他更新](#)
- [D.4 面向对象编程：两种不同的类对象](#)
- [D.5 字符串](#)
- [D.6 异常](#)
- [D.6.1 处理异常（使用as）](#)
- [D.6.2 抛出异常](#)
- [D.7 其他转换工具和技巧](#)
- [D.8 编写兼容版本2.x和3.x的代码](#)
- [D.8.1 对比print和print\(\)](#)
- [D.8.2 将你的方法导入解决方案中](#)
- [D.8.3 整合在一起](#)
- [D.9 结论](#)

Pearson

core PYTHON APPLICATIONS programming THIRD EDITION

Python核心编程（第3版）

[美]Wesley Chun 著

孙波翔 李斌 李晗 译

人民邮电出版社

北京

图书在版编目 (CIP) 数据

Python核心编程: 第3版/(美)春(Chun,W.)著;孙波翔,李斌,李晗译.--北京:人民邮电出版社,2016.6

ISBN 978-7-115-41477-9

I.①P... II.①春...②孙...③李...④李... III.①软件工具—程序设计 IV.①TP311.56

中国版本图书馆CIP数据核字(2016)第084805号

版权声明

Authorized translation from the English language edition,entitled Core Python Applications Programming(3rd edition),9780132678209 by Wesley J.Chun published by Pearson Education,Inc.,publishing as Prentice Hall,Copyright © 2012 Pearson Education,Inc.

All rights reserved.No part of this book may be reproduced or transmitted in any form or by any means,electronic or mechanical,including photocopying,recording or by any information storage retrieval system,without permission from Pearson Education Inc.CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD.,and POSTS & TELECOMMUNICATIONS PRESS Copyright © 2016.

本书封面贴有Pearson Education(培生教育出版集团)激光防伪标签。无标签者不得销售。

◆著 [美]Wesley Chun

译 孙波翔 李斌 李晗

责任编辑 傅道坤

责任印制 焦志炜

◆人民邮电出版社出版发行 北京市丰台区成寿寺路11号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

三河市海波印务有限公司印刷

◆开本: 800×1000 1/16

印张: 41.25

字数: 903千字 2016年6月第1版

印数: 1-8000册 2016年6月河北第1次印刷

著作权合同登记号 图字: 01-2012-5027号

定价: 99.00元

读者服务热线: (010)81055410 印装质量热线: (010)81055316

反盗版热线: (010)81055315

内容提要

本书是经典畅销图书《Python核心编程（第二版）》的全新升级版，总共分为3部分。第1部分讲解了Python的一些通用应用，包括正则表达式、网络编程、Internet客户端编程、多线程编程、GUI编程、数据库编程、Microsoft Office编程、扩展Python等内容。第2部分讲解了与Web开发相关的主题，包括Web客户端和服务端、CGI和WSGI相关的Web编程、Django Web框架、云计算、高级Web服务。第3部分则为一个补充/实验章节，包括文本处理以及一些其他内容。

本书适合具有一定经验的Python开发人员阅读。

本书赞誉

“本书简洁而不失其技术深度，内容丰富全面，历史资料翔实齐全，这让本书成为学习Python的完美教程。本书易于阅读，以极简的文字介绍了复杂的案例，同时涵盖了其他同类图书中很少涉及的历史参考资料。简而言之，本书棒极了！”

——Gloria.W

本书之前版本的赞誉

“期待已久的Core Python Programming（第2版）已经证明了本书确实值得期待——它深度与广度兼备，其中囊括的有用练习可以帮助读者掌握Python并付诸于实践。”

——Alex Martelli, Python in a Nutshell作者兼Python Cookbook编辑

“Wesley Chun的Core Python Programming一书好评如潮，而且它也证明它配得上所有的好评。我想该书是当前学习Python的最佳图书。在市面上众多的Python图书中，我觉得Chun的这本书是最好的，因此强烈推荐本书。”

——David Mertz博士，IBM DeveloperWorks

“在过去多年，我一直在从事Python的研究，发现本书获得了大量的正面评价。这些评价证实了这样一个观点，即Core Python Programming被认为是标准的Python入门读物。”

——Richard Ozaki, Lockheed Martin公司

“终于，一本既可以作为Python教程又可以作为Python编程语言参考的图书问世了！”

——Michael Baxter, Linux Journal

“本书写作相当精良。这是我遇到的最清晰、最友好的Python图书，它在一个广阔背景下介绍了Python。它仔细、深入地剖析了一些重要的Python主题，而且读者无需大量的相关经验也能看懂。与其他所有Python入门类图书不同的是，它不会用隐晦、难以理解的文字来折磨读者，而是始终立足于帮助读者牢固掌握Python的语法和结构。”

——[http://python.org bookstore](http://python.org/bookstore) Web site

“如果我只能有一本Python图书，那它肯定是Wesley Chun著作的Core Python Programming。本书成功地涵盖了Python的多个主题，其详细程度远甚于Learning Python一书，而且涵盖的主题也远非Python核心语言这么简单。如果你只打算购买一本Python图书，我强烈推荐本书。你不仅会爱上本书，而且会爱上本书中包含的真知灼见。重要的是，你将学会Python。更重要的是，你会发现本书会在你每日的Python编程生活中提供各种帮助。写得不错，Chun先生！”

——Ron Stephens, Python学习基金会

“我认为编程初学者的最佳语言是Python，这毋庸置疑！我最喜欢的图书是Core Python Programming。”

——s003apr, MP3Car.com论坛

“就我个人而言，我相当喜欢Python。它易于学习、非常直观、相当灵活，而且执行速度也相当快。在Windows领域中，Python虽然只是刚崭露头角，但是由于越来越多的人发现了它，因此选择从Python起步可以获得大量的支持。要学习Python，我选择从Wesley Chun的这本Core Python Programming起步。”

——Bill Boswell, MCSE, 微软认证专家在线杂志

“如果你通过图书来学习编程，我推荐Core Python Programming，它是目前为止我发现的最佳Python图书。我也是一个Python新手，但是在3个月之后，我就可以在项目中实现Python了（自动处理MSOffice、SQL DB等）。”

——ptonman, Dev Shed论坛

“Python是一种美丽的语言。它易于学习、跨平台，而且能够良好运行。它已经实现了Java一直想要实现的很多技术目标。对Python的一句话描述是‘所有其他语言随着时间发生演进，但Python是设计出来的。’而且Python设计得相当不错。虽然现在市面上有大量的Python图书，但是目前为止我遇到的最好的一本是Core Python Programming。”

——Chris Timmons, C.R.Timmons Consulting公司

“如果你喜欢Prentice Hall出版社的Core系列图书，你需要考虑的另一本写作精良的图书是Core Python Programming。它将其他Python图书中很少涵盖的许多实用主题进行了事无巨细的剖析。”

——Mitchell L.Model, MLM Consulting公司