

Python 3.5.x、Python 3.6.x

及更新版本的内置对象和标准库对象的高级用法，注重Python内功修炼



# Python

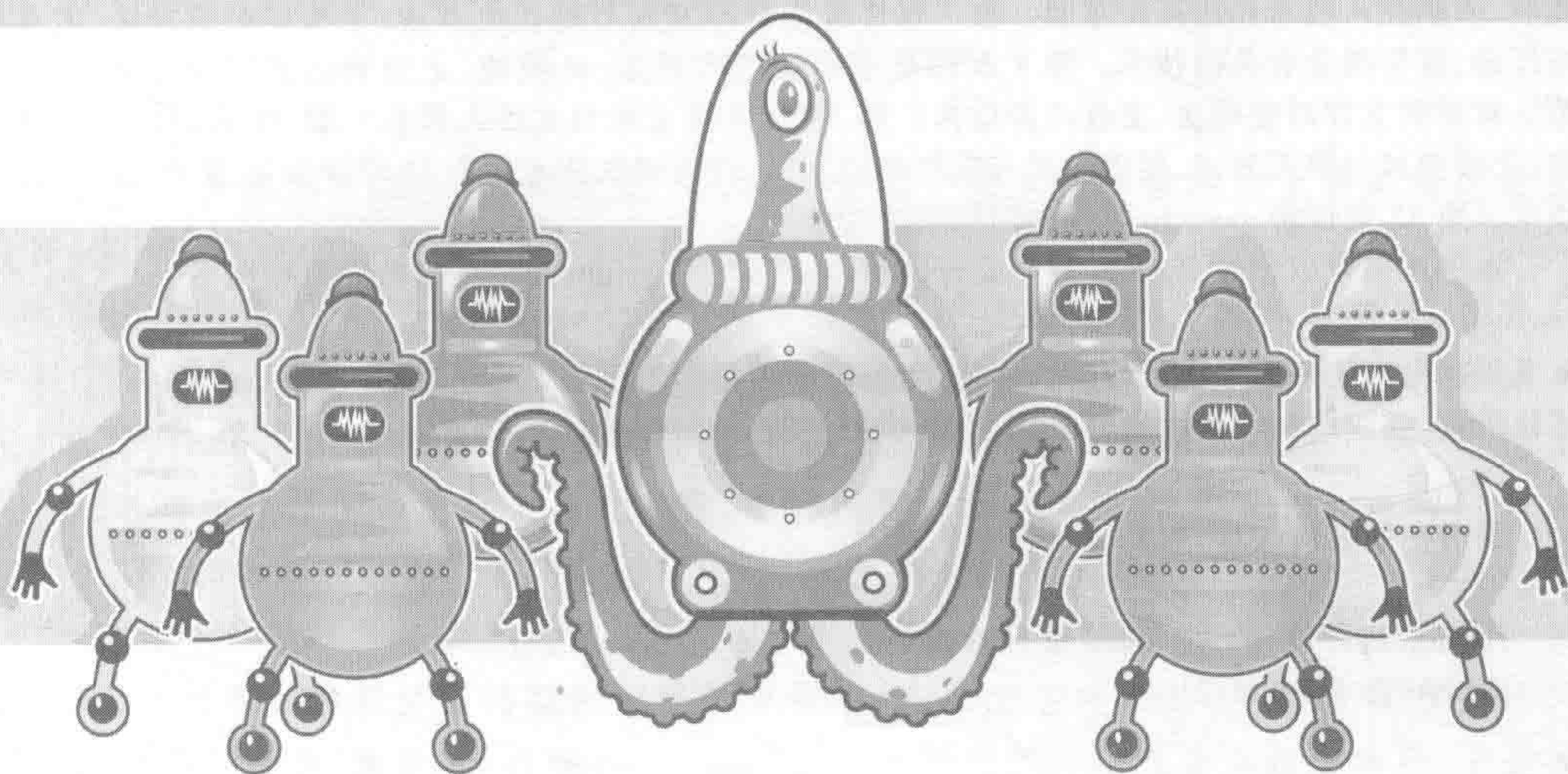
## 程序设计开发宝典

董付国◎著

- ◇ 150个案例源代码和超过1000个演示性代码片段，完美诠释Pythonic真谛。
- ◇ 实时跟踪Python动态，紧跟时代发展潮流，介绍GPU加速、分布式、异步通信等最新技术特性。



清华大学出版社



# Python

## 程序设计开发宝典

董付国◎著

清华大学出版社  
北京

## 内 容 简 介

全书共 13 章,面向 Python 3.5.x、Python 3.6.x 和 Python 3.7.x,重点关注 Python 内置对象和标准库对象的高级应用以及比较前沿的一些新技术,偶尔涉及一点扩展库用法。第 1 章介绍 Python 语言编程规范与代码优化建议、开发环境配置、程序伪编译与打包。第 2 章详解运算符与内置函数的用法。第 3 章详解各种序列对象、推导式、切片和序列解包。第 4 章详解选择结构和循环结构,关键字 else、break 与 continue。第 5 章详解函数的基本用法,可调用对象与修饰器、函数参数、变量的作用域、生成器函数、lambda 表达式、函数柯里化、泛型函数、协程函数和回调函数。第 6 章详解类的定义、不同类型的成员、依赖注入技术和运算符重载。第 7 章详解字符串编码与格式化方法、字符串对象方法、文本排版与压缩、汉字拼音有关的技术。第 8 章详解正则表达式语法、re 模块、正则表达式对象与 match 对象。第 9 章详解文件对象用法、文件内容操作。第 10 章详解文件与文件夹操作。第 11 章详解异常处理结构、文档测试与单元测试、覆盖测试与软件性能测试、代码调试技术。第 12 章详解不同类型的并行处理技术。第 13 章详解 asyncio 提供的网络通信功能。

本书不但可以作为 Python 程序设计教材,还可作为 Python 开发工程师的指导用书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

### 图书在版编目(CIP)数据

Python 程序设计开发宝典/董付国著. —北京:清华大学出版社,2017  
ISBN 978-7-302-47210-0

I. ①P… II. ①董… III. ①软件工具—程序设计 IV. ①TP311.561

中国版本图书馆 CIP 数据核字(2017)第 125790 号

责任编辑:白立军

封面设计:杨玉兰

责任校对:焦丽丽

责任印制:王静怡

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, [c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

质量反馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:三河市金元印装有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:25.25

字 数:584 千字

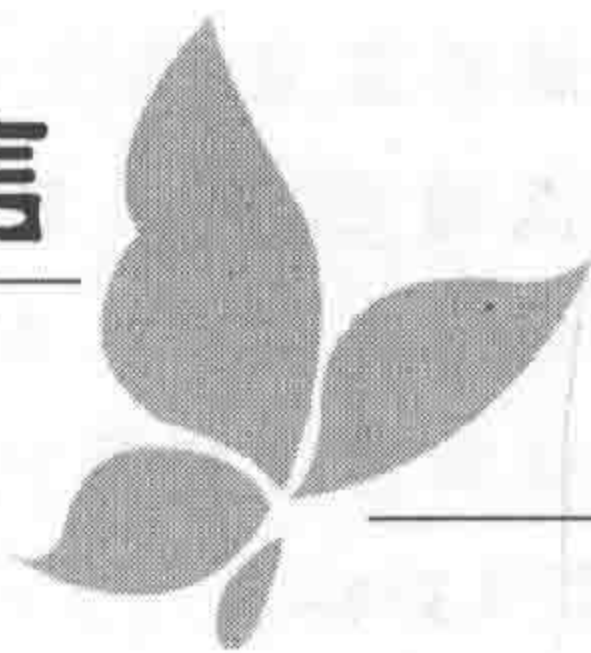
版 次:2017 年 8 月第 1 版

印 次:2017 年 8 月第 1 次印刷

印 数:1~2000

定 价:69.00 元

产品编号:072406-01



作者第一次接触 Python 大概是在 2002 年,在几个著名的安全网站上看到有人用这个语言,当时的作者正痴迷于 C 语言和汇编语言,内心不屑于学习这种脚本语言。2010 年的时候因为要用 IDA Pro 分析一个 PE 文件而不得不借助于 IDAPython 插件,才真正感觉到了 Python 的方便与强大,于是购买了很多书籍并且阅读了大量在线文档开始系统地学习 Python。2013 年学校组织各专业教研室修订新版人才培养方案时,经过慎重考虑并且与其他几位老师进行了多次沟通,最终确定为数字媒体技术专业和服务外包专业增加了“Python 程序设计”这门课程。然而,虽然当时市面上已经有了一些 Python 书籍,但是适合作为教材的却寥寥无几,有的过于专注某个专业领域,有的则是泛泛地介绍一点皮毛,并且很多书里放置的插图占用了大量篇幅,知识密度很小,不是作者喜欢的风格。在翻看了超过 20 本 Python 图书以后,作者决定动手写一本适合作教材的书,于是就有了面向计算机及相关专业的《Python 程序设计》(书号为 9787302407232,2015 年 8 月出版)和面向非计算机专业的《Python 程序设计基础》(书号为 9787302410584,2015 年 8 月出版,2016 年 3 月第 2 次印刷,2017 年 1 月第 3 次印刷),前者主要介绍 Python 2.7.x 的基本语法以及在各领域的应用,后者主要介绍 Python 3.4.x 的基本语法而没有涉及太多的应用。本来当时写这两本书的目的仅仅是为了自己上课用起来方便,然而出版不到一年就被近 30 所院校选作教材,还有几十所院校的图书馆也采购了这两本书供学生借阅,反响非常好。应广大读者和用书老师的要求,也为了紧跟 Python 飞速发展的步伐,作者于 2016 年 6 月又出版了《Python 程序设计(第 2 版)》(书号为 9787302436515),这本书使用 Python 3.5.x 重写了第 1 版中的所有案例,并且新增案例近百个,出版后迅速被多所院校选作教材,各大网上书店也频频缺货,2016 年 11 月初进行了第 2 次印刷,2017 年 3 月进行了第 3 次印刷。考虑到更多的 Python 爱好者并没有听老师讲课的机会,看教材自学可能比较吃力,作者于 2017 年 1 月份出版了《Python 可以这样学》(书号为 9787302456469),全书 500 多页,使用生动活泼的语言讲解 Python 3.5.x 的知识和应用(绝大部分内容也适用于 Python 3.6.x 和 Python 3.7.x),在《Python 程序设计(第 2 版)》的基础上删掉了“软件分析与逆向工程”和“安卓平台的 Python 编程”内容,新增了大量案例,并且融入了道德经、周易、太极拳理论中的核心思想以及大量中外名人名言,通过小提示、小技巧、注意、拓展知识等多种形式扩充了大量知识,尤其适合 Python 爱好者自学,也可作为进阶工具书进行查阅。该书出版之后迅速得到社会各界人士的一致认可,第一批印刷的书很快被抢购一空,不到两个月就进行了第二次印刷。回头想想,自己二十年如一日地每天熬夜看书学习写代码还是值得的。

当 2016 年 6 月份作者前几本 Python 系列图书的责任编辑白立军老师约作者再编写一本面向高级程序员的 Python 图书时,说实话,内心是喜悦的,很高兴自己的努力得到广大 Python 爱好者的认可。但同时作者内心也有些担心,感觉在写前 4 本书时已经用完了自己的洪荒之力。在这本新书里再写点什么好呢?内容该如何组织呢?如何避免过多地重复利用前几本书里的案例呢?反复思考了近 2 个月,考虑目前很多 Python 程序员的现状:喜欢直接使用各种扩展库来解决问题,不重视对 Python 语言本身的理解,内功不够深厚,导致很多代码粗制滥造。最终作者做出决定,在这本新书里,不再介绍太多扩展库的应用,而是把重点放在 Python 语法和内置对象、标准库对象的高级应用上,注重 Python 的内涵,注重内功的修炼,尽量往纵深发展,争取用最简练的语言介绍那些作者认为比较高级的用法。在编写过程中,尽量减少与前几本书中内容的重复,补充大量新案例和高级用法。当然,前面几本书里的有些案例在这本书里又出现了,但是仔细的读者应该能发现,很多案例代码都进行了必要的改写和优化,更加 Pythonic,更加优雅和高效。自从答应了写这本书之后,作者在不影响正常教学和科研工作的情况下每天拿出至少 10 个小时查阅资料、编写代码或者整理书稿,前后用了一年左右的时间,现在回头想想也挺值得,在整理资料和编写案例代码的同时作者自己也进步了很多,对 Python 有了更加深入的认识和理解。

## 内容组织与阅读建议

全书共 13 章,面向 Python 3.5.x、Python 3.6.x 和 Python 3.7.x,重点关注 Python 内置对象和标准库对象的高级应用,以及比较前沿的、刚刚引入的一些新技术和新特性的用法,偶尔涉及一些扩展库用法。几乎每个知识点都配有大量的案例,把这些案例简单拼凑和集成就可以实现很多功能,实用性非常强。建议读者按章节顺序阅读,并且前后结合地反复阅读,不要随意跳过任何内容,或许不经意间会发现自己正需要的知识或者得到某种灵感。另外,虽然本书的定位是 Python 高级编程,但也同样适用于初学者,请初学者不要觉得有压力,如果有些地方暂时看不懂,可以先跳过去,或许过几天再看就明白了。当然,如果实在看不懂的话,及时和作者沟通应该会得到帮助。

**第 1 章 管中窥豹:Python 概述。**介绍 Python 语言的特点与主流版本,编程规范与代码优化建议,虚拟开发环境的创建与配置,扩展库的安装方法,开发与发布自己的包,Python 程序伪编译与打包,从命令行和外部文件获取配置信息。

**第 2 章 万丈高楼平地起:运算符、表达式与内置对象。**详细介绍 Python 的运算符与内置函数的用法,以及变量与常量的概念。

**第 3 章 玄之又玄,众妙之门:详解 Python 序列结构。**详解列表、元组、字典、集合等对象的特点与用法,列表推导式、生成器推导式、字典推导式与集合推导式,切片操作,序列解包,枚举、数组、队列、堆等常用结构用法。

**第 4 章 反者,道之动:程序控制结构。**详解 Python 中的选择结构和循环结构,else 的几种用法,选择结构的多种实现方式,break 与 continue 语句,循环结构的代码优化技巧。

**第 5 章 代码复用技术(一):函数。**详解函数的定义与嵌套定义语法,可调用对象与



修饰器原理,位置参数、默认值参数、关键参数以及参数的序列解包,局部变量、全局变量与 nonlocal 变量,生成器函数,lambda 表达式,函数柯里化,泛型函数,协程函数,回调函数。

第 6 章 代码复用技术(二):面向对象程序设计。详解类的定义语法与实例化方法,数据成员、成员方法、属性以及静态方法与类方法,继承与多态,依赖注入技术的 Python 实现,特殊成员重写与运算符重载。

第 7 章 文本处理(一):字符串。详解字符串编码与字符串格式化方法、字符串对象方法、文本排版与压缩、分词、汉字拼音有关的技术。

第 8 章 文本处理(二):正则表达式。详解正则表达式基本语法与扩展语法,正则表达式模块 re 的用法,正则表达式对象与 match 对象的用法。

第 9 章 数据永久化:文件内容操作。详解内置函数 open()与上下文管理语句 with 的用法,文本文件与二进制文件的操作,Excel、Word、zip、rar 等常见二进制文件操作技术。

第 10 章 文件与文件夹操作。详解 os、os.path、shutil、glob、fnmatch 等模块在文件与文件夹操作方面的用法。

第 11 章 代码质量保障:异常处理结构、程序调试与测试。详解异常处理结构,文档测试与单元测试技术,覆盖测试与软件性能测试技术,IDLE、pdb、Eclipse+Pydev 等不同的代码调试技术。

第 12 章 多任务与并行处理:线程、进程、协程、分布式、GPU 加速。详解多线程与多进程编程技术,线程与进程的同步技术,不同进程间数据交换与共享技术,协程,spark 并行计算与 GPU 编程。

第 13 章 互通互联:asyncio 提供的网络通信功能。详解 asyncio 提供的网络通信功能,重点介绍 Transport、Protocol、StreamReader 以及 StreamWriter 等类的用法。

## 配套资源

本书提供所有案例源代码,可以登录清华大学出版社网站(www.tup.com.cn)下载,或加入本书读者群(QQ 群号 456324891,加入时请注明是读者,如果这个群满了的话会在群简介中给出下一个群号)下载最新配套资源并与作者交流,当然也欢迎关注微信公众号“Python 小屋”及时阅读作者写的最新案例代码,有些代码是在本书完稿之后新写的,是书上没有的,算作是一个很好的补充。

## 适用读者

- 已经具有一定 Python 水平的软件开发工程师。
- 打算深入探究 Python 高级编程的狂热爱好者。
- 各专业研究生、本科生、专科生的程序设计教材。
- 可能有些内容看起来会稍微有些吃力的其他 Python 初学者。



## 致谢

首先感谢父母的养育之恩,在当年那么艰苦的条件下还坚决支持我读书,没有让我像其他有的同龄的孩子一样辍学。感谢姐姐、姐夫多年来对我的爱护以及在老家对父母的照顾,感谢善良的弟弟、弟媳在老家对父母的照顾,正是有了你们,远离家乡的我才能安心工作。当然,最应该感谢的是妻子和孩子对我这个技术狂人的理解和宽容,这些年来她们已经习惯了正在吃饭的我突然萌发一个思路就放下饭碗去计算机前写代码的情景,习惯了我每个周末和假期都在教研室看书或写代码而不能陪她们玩,也习惯了周末的中午和晚上她们做好饭后再打电话催我回家。为了支持我的工作,她们还在我的几本书正式出版之前阅读了我的书稿。

感谢我的领导冯烟利教授提供了良好的教学科研环境,这样的工作环境让人觉得非常舒适,每个人都可以安心做好自己的事,发挥出最大潜力。

感谢选用 Python 程序设计系列教材的老师和长期关注微信公众号“Python 小屋”的朋友以及系列图书读者 QQ 群里的朋友,感谢你们指出前几本书里存在的几个小错误,和你们的交流也给了我很多启发!

感谢每一位读者,感谢您在茫茫书海中选择了本书,衷心希望您能够从本书中受益,学到真正需要的知识!同时也期待每一位读者的热心反馈,随时欢迎您指出书中的不足!

本书在编写出版过程中也得到清华大学出版社的大力支持和帮助,尤其是非常有远见的责任编辑白立军老师对这套 Python 系列图书的策划,一并表示衷心的感谢。

董付国 于山东烟台

2017 年 3 月



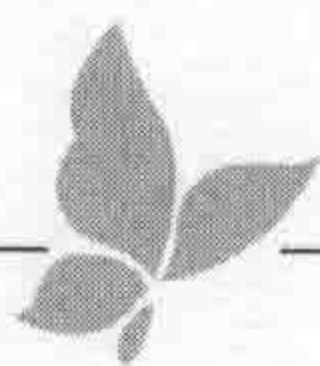
<b>第 1 章</b>	<b>管中窥豹：Python 概述</b>	<b>1</b>
1.1	Python 是这样一种语言	1
1.2	Python 版本之争	1
1.3	Python 编程规范与代码优化建议	2
1.4	Python 虚拟开发环境的搭建	4
1.5	Eclipse+PyDev 环境搭建和使用	4
1.6	安装扩展库的几种方法	6
1.7	标准库与扩展库中对象的导入与使用	7
1.7.1	import 模块名 [as 别名]	8
1.7.2	from 模块名 import 对象名 [as 别名]	8
1.7.3	from 模块名 import *	9
1.7.4	模块导入时的搜索路径	10
1.8	编写与发布自己的包	10
1.9	Python 程序伪编译与打包	12
1.10	从命令行参数和配置文件获取信息	14
<b>第 2 章</b>	<b>万丈高楼平地起：运算符、表达式与内置对象</b>	<b>16</b>
2.1	Python 常用内置对象	16
2.1.1	常量与变量	17
2.1.2	数字	18
2.1.3	字符串	21
2.1.4	列表、元组、字典、集合	22
2.2	Python 运算符与表达式	23
2.2.1	算术运算符	24
2.2.2	关系运算符	25
2.2.3	成员测试运算符 in 与同一性测试运算符 is	26
2.2.4	位运算符与集合运算符	27
2.2.5	逻辑运算符	28
2.2.6	矩阵乘法运算符@	29



2.2.7	补充说明 .....	29
2.3	Python 关键字简要说明 .....	30
2.4	Python 常用内置函数用法精要 .....	31
2.4.1	类型转换与类型判断 .....	34
2.4.2	最值与求和 .....	38
2.4.3	基本输入输出 .....	40
2.4.4	排序与逆序 .....	42
2.4.5	枚举与迭代 .....	43
2.4.6	map()、reduce()、filter() .....	45
2.4.7	range() .....	48
2.4.8	zip() .....	48
2.4.9	eval()、exec() .....	49
<b>第 3 章</b>	<b>玄之又玄,众妙之门: 详解 Python 序列结构 .....</b>	<b>51</b>
3.1	列表: 打了激素的数组 .....	51
3.1.1	列表创建与删除 .....	52
3.1.2	列表元素访问 .....	53
3.1.3	列表常用方法 .....	54
3.1.4	列表对象支持的运算符 .....	59
3.1.5	内置函数对列表的操作 .....	61
3.1.6	使用列表模拟向量运算 .....	62
3.1.7	列表推导式语法与应用案例 .....	63
3.1.8	切片操作的强大功能 .....	70
3.2	元组: 轻量级列表 .....	73
3.2.1	元组创建与元素访问 .....	73
3.2.2	元组与列表的异同点 .....	74
3.2.3	生成器推导式 .....	75
3.3	字典: 反映对应关系的映射类型 .....	76
3.3.1	字典创建与删除 .....	77
3.3.2	字典元素的访问 .....	77
3.3.3	元素的添加、修改与删除 .....	79
3.3.4	标准库 collections 中与字典有关的类 .....	80
3.4	集合: 元素之间不允许重复 .....	81
3.4.1	集合对象的创建与删除 .....	81
3.4.2	集合操作与运算 .....	82
3.4.3	不可变集合 frozenset .....	84
3.4.4	集合应用案例 .....	84
3.5	序列解包的多种形式和用法 .....	86



3.6	标准库中的其他常用数据类型	88
3.6.1	枚举类型	88
3.6.2	数组类型	89
3.6.3	队列	90
3.6.4	具名元组	92
3.6.5	堆	92
<b>第4章</b>	<b>反者,道之动: 程序控制结构</b>	<b>94</b>
4.1	条件表达式	94
4.2	选择结构	96
4.2.1	单分支选择结构	96
4.2.2	双分支选择结构	97
4.2.3	多分支选择结构	98
4.2.4	选择结构的嵌套	99
4.2.5	构建跳转表实现多分支选择结构	100
4.3	循环结构	100
4.3.1	for 循环与 while 循环	100
4.3.2	break 与 continue 语句	101
4.3.3	循环代码优化技巧	102
4.4	精彩案例赏析	103
<b>第5章</b>	<b>代码复用技术(一): 函数</b>	<b>111</b>
5.1	函数定义与使用	111
5.1.1	基本语法	111
5.1.2	函数嵌套定义、可调用对象与修饰器	113
5.1.3	函数对象成员的动态性	117
5.1.4	函数递归调用	117
5.2	函数参数	118
5.2.1	位置参数	120
5.2.2	默认值参数	120
5.2.3	关键参数	122
5.2.4	可变长度参数	122
5.2.5	强制函数的某些参数必须以关键参数形式进行传值	123
5.2.6	强制函数的所有参数必须以位置参数形式进行传值	124
5.2.7	传递参数时的序列解包	125
5.2.8	标注函数参数与返回值类型	126
5.3	变量作用域	127
5.3.1	全局变量与局部变量	127



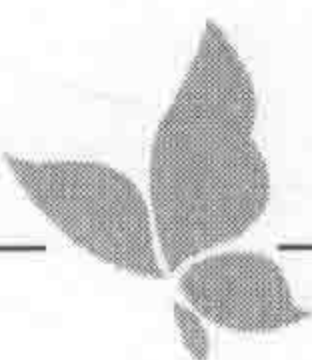
5.3.2	nonlocal 变量 .....	130
5.4	lambda 表达式 .....	130
5.5	生成器函数设计要点 .....	133
5.6	偏函数与函数柯里化 .....	135
5.7	单分发器与泛型函数 .....	136
5.8	协程函数 .....	138
5.9	注册程序退出时必须执行的函数 .....	140
5.10	回调函数原理与实现 .....	140
5.11	精彩案例赏析 .....	141

## **第 6 章** 代码复用技术(二): 面向对象程序设计 ..... 167

6.1	类的定义与使用 .....	167
6.1.1	基本语法 .....	167
6.1.2	type 类 .....	168
6.1.3	定义带修饰器的类 .....	169
6.2	数据成员与成员方法 .....	171
6.2.1	私有成员与公有成员 .....	171
6.2.2	数据成员 .....	172
6.2.3	成员方法、类方法、静态方法、抽象方法 .....	173
6.2.4	属性 .....	175
6.2.5	类与对象的动态性、混入机制 .....	177
6.3	继承、多态、依赖注入 .....	179
6.3.1	继承 .....	179
6.3.2	多态 .....	181
6.3.3	依赖注入技术的不同实现方法 .....	182
6.4	特殊方法与运算符重载 .....	185
6.5	精彩案例赏析 .....	187
6.5.1	自定义队列 .....	187
6.5.2	自定义栈 .....	191
6.5.3	自定义集合 .....	193
6.5.4	自定义数组 .....	199
6.5.5	自定义双链表 .....	204
6.5.6	自定义常量类 .....	206
6.5.7	自定义不允许修改值的字典 .....	207
6.5.8	自定义支持 with 关键字的类 .....	208

## **第 7 章** 文本处理(一): 字符串 ..... 209

7.1	字符串编码格式简介 .....	210
-----	-----------------	-----



7.2	转义字符与原始字符串 .....	211
7.3	字符串格式化 .....	212
7.3.1	使用%符号进行格式化 .....	212
7.3.2	使用 format() 方法进行字符串格式化 .....	213
7.3.3	格式化的字符串常量 .....	215
7.3.4	使用 Template 模板进行格式化 .....	215
7.4	字符串常用操作 .....	216
7.4.1	find()、rfind()、index()、rindex()、count() .....	216
7.4.2	split()、rsplit()、partition()、rpartition() .....	217
7.4.3	join() .....	219
7.4.4	lower()、upper()、capitalize()、title()、swapcase() .....	220
7.4.5	replace()、maketrans()、translate() .....	220
7.4.6	strip()、rstrip()、lstrip() .....	221
7.4.7	startswith()、endswith() .....	222
7.4.8	isalnum()、isalpha()、isdigit()、isdecimal()、isnumeric()、 isspace()、isupper()、islower() .....	222
7.4.9	center()、ljust()、rjust()、zfill() .....	223
7.4.10	字符串对象支持的运算符 .....	224
7.4.11	适用于字符串对象的内置函数 .....	226
7.4.12	字符串对象的切片操作 .....	228
7.5	其他有关模块 .....	228
7.5.1	textwrap 模块 .....	228
7.5.2	zlib 模块提供的压缩功能 .....	231
7.6	字符串常量 .....	232
7.7	可变字符串 .....	232
7.8	中英文分词 .....	234
7.9	汉字到拼音的转换 .....	235
7.10	精彩案例赏析 .....	236

## **第 8 章** 文本处理(二): 正则表达式 ..... 239

8.1	正则表达式语法 .....	239
8.1.1	正则表达式基本语法 .....	239
8.1.2	正则表达式扩展语法 .....	240
8.1.3	正则表达式锦集 .....	241
8.2	直接使用正则表达式模块 re 处理字符串 .....	242
8.3	使用正则表达式对象处理字符串 .....	246
8.4	match 对象 .....	248
8.5	精彩案例赏析 .....	250



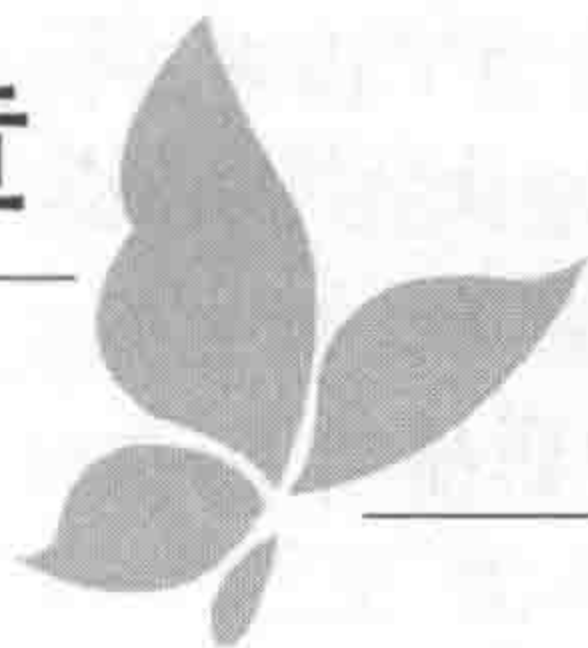
<b>第 9 章</b>	<b>数据永久化：文件内容操作</b>	<b>257</b>
9.1	文件操作基本知识	258
9.1.1	内置函数 open()	258
9.1.2	文件对象属性与常用方法	260
9.1.3	上下文管理语句 with	261
9.2	文本文件内容操作案例精选	261
9.3	二进制文件操作案例精选	266
9.3.1	使用 pickle 模块读写二进制文件	267
9.3.2	使用 struct 模块读写二进制文件	268
9.3.3	使用 shelve 模块操作二进制文件	269
9.3.4	使用 marshal 模块操作二进制文件	270
9.3.5	其他常见类型二进制文件操作案例	271
<b>第 10 章</b>	<b>文件与文件夹操作</b>	<b>281</b>
10.1	os 模块	281
10.2	os.path 模块	284
10.3	shutil 模块	286
10.4	其他常用模块	287
10.4.1	glob 模块	287
10.4.2	fnmatch 模块	288
10.5	精彩案例赏析	289
<b>第 11 章</b>	<b>代码质量保障：异常处理结构、程序调试与测试</b>	<b>293</b>
11.1	异常处理结构	293
11.1.1	异常的概念与表现形式	293
11.1.2	Python 内置异常类层次结构	294
11.1.3	异常处理结构	296
11.1.4	断言与上下文管理语句	301
11.2	文档测试 doctest	301
11.3	单元测试 unittest	304
11.4	覆盖测试	307
11.5	软件性能测试	308
11.6	代码调试	310
11.6.1	使用 IDLE 调试	310
11.6.2	使用 Eclipse+PyDev 进行代码调试	311
11.6.3	使用 pdb 调试	313



<b>第 12 章</b>	<b>多任务与并行处理：线程、进程、协程、分布式、GPU 加速</b>	<b>318</b>
12.1	多线程编程	318
12.1.1	线程概念与标准库 threading	318
12.1.2	线程对象	320
12.1.3	线程调度	323
12.1.4	Lock/RLock 对象	324
12.1.5	Condition 对象	326
12.1.6	Queue 对象	328
12.1.7	Event 对象	332
12.1.8	Semaphore 与 BoundedSemaphore	333
12.1.9	Barrier 对象	334
12.2	多进程编程	335
12.2.1	进程创建与管理	335
12.2.2	进程同步技术	336
12.2.3	Pool 对象	337
12.2.4	Manager 对象	341
12.2.5	Listener 与 Client 对象	345
12.2.6	进程间数据交换与共享	345
12.2.7	标准库 subprocess	348
12.3	协程	349
12.4	concurrent.futures 模块提供的并发执行功能	351
12.5	pySpark 并行计算与分布式计算框架	353
12.6	GPU 编程	359
12.6.1	使用 pycuda 实现 GPU 加速	359
12.6.2	使用 pyopencl 实现 GPU 加速	360
12.6.3	使用 tensorflow 实现 GPU 加速	362
<b>第 13 章</b>	<b>互通互联：asyncio 提供的网络通信功能</b>	<b>364</b>
13.1	Transport 类与 Protocol 类	364
13.2	StreamReader 与 StreamWriter	369
<b>附录</b>	<b>精彩在继续</b>	<b>373</b>
附录 A	GUI 开发	373
附录 B	SQLite 数据库操作	376
附录 C	计算机图形学编程	377
附录 D	图像编程	380
附录 E	数据分析、机器学习、科学计算可视化	383



附录 F 密码学编程 .....	383
附录 G 系统运维 .....	384
附录 H Windows 系统编程 .....	384
附录 I 软件分析与逆向工程 .....	386
<b>参考文献 .....</b>	<b>388</b>



### 1.1 Python 是这样一种语言

有不少人说 Python 是一种“大蟒蛇语言”。虽然在英语中 Python 确实有大蟒蛇的意思,但 Python 语言和大蟒蛇却没有任何关系。Python 语言的名字来自于一个著名的电视剧 *Monty Python's Flying Circus*, Python 之父 Guido van Rossum 是这部电视剧的狂热爱好者,所以把他设计的语言命名为 Python。

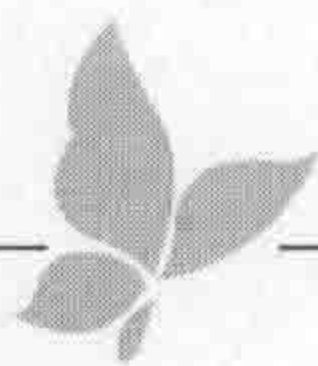
也有人说 Python 是一门脚本语言,这也是不准确的,远远不足以反映 Python 的强大。Python 并不仅仅是一门脚本语言,更是一门跨平台、开源、免费的解释型高级动态编程语言,是一种通用编程语言。除了可以解释执行之外,Python 还支持将源代码伪编译为字节码来优化程序提高运行速度并对源代码进行保密,也支持使用 py2exe、pyinstaller、cx\_Freeze 或其他类似工具将 Python 程序及其所有依赖库打包成为各种平台上的可执行文件,当然也包括扩展名为 exe 的 Windows 可执行程序,从而可以脱离 Python 解释器环境和相关依赖库,能够在 Windows 平台上独立运行,并且还支持制作成 .msi 安装包;Python 支持命令式编程(How to do)和函数式编程(What to do)两种方式,完全支持面向对象程序设计(虽然并不强制要求处处体现面向对象编程的思想和有关特征,但实际上人们无时无刻不在使用),语法简洁清晰,功能强大且易学易用,更重要的是拥有大量的几乎支持所有领域应用开发的成熟扩展库和狂热支持者。

当然,也有人喜欢把 Python 称为“胶水语言”,这确实是 Python 的重要特点之一,它可以把多种不同语言编写的程序融合到一起实现无缝拼接,更好地发挥不同语言和工具的优势,满足不同应用领域的需求。

### 1.2 Python 版本之争

众所周知,Python 官方网站同时发行和维护着 Python 2. x 和 Python 3. x 两个不同系列的版本,并且版本更新速度非常快(6 个月左右更新一次小版本号)。目前最新版本分别是 Python 2. 7. 13、Python 3. 4. 6、Python 3. 5. 3 和 Python 3. 6. 1,Python 3. 7 已在研发中,估计很快就会推出。Python 2. x 和 Python 3. x 这两个系列的版本之间很多用法是不兼容的(让人欣慰的是,除了一些新特性、运算符和标准库对象之外,同一个系列的不同版本之间绝大多数用法是完全一致的),除了基本输入输出方式有所不同,很多内置函





数和标准库对象的用法也有非常大的区别,Python 3. x 在增加了很多新标准库的同时也删除了一些 Python 2. x 的标准库,还有些 Python 2. x 的标准库在 Python 3. x 中进行了合并和拆分。当然,适用于 Python 2. x 和 Python 3. x 的扩展库之间更是差别巨大,这应该是现有系统进行版本迁移时最大的障碍。因此,在正式开始使用 Python 之前,必须要选择合适的版本,以免浪费时间。

在选择 Python 版本的时候,一定要先考虑清楚自己学习 Python 的目的是什么,打算做哪方面的开发,该领域或方向有哪些扩展库可用,这些扩展库最高支持哪个版本的 Python,是否还在维护和更新。这些问题全部明确以后,再最终确定选择哪个版本,这样才能事半功倍,而不至于把太多时间浪费在 Python 以及各种扩展库的反复安装和卸载上,虽然这并不是非常麻烦。另外,当较新的 Python 版本推出之后,不要急于安装,而是应该在确定自己所必须使用的扩展库也推出了与之匹配的稳定版本之后再一起进行更新。

总体来看,Python 3. x 的设计理念更加合理、高效和人性化,全面普及和应用是必然的,越来越多的扩展库也以非常快的速度推出了与最新 Python 版本相适应的版本。如果暂时还没想到要做什么行业领域的应用开发,或者仅仅是为了尝试一种新的、好玩的语言,那么请毫不犹豫地选择 Python 3. x 系列的最高版本。

### 1.3 Python 编程规范与代码优化建议

没有规矩,不成方圆。任何一种语言都有一些约定俗成的编码规范,Python 也不例外。Python 非常重视代码的可读性,对代码布局和排版有更加严格的要求。虽然一些大型软件公司对自己公司程序员编写的代码在布局、结构、标识符命名等方面有一些特殊的要求,但其中很多思想是相同的,目的也是一致的。这里重点介绍 Python 社区对代码编写的一些共同的要求、规范和一些常用的代码优化建议,最好在开始编写第一段代码的时候就要遵循这些规范和建议,养成一个好的习惯。

(1) 严格使用缩进来体现代码的逻辑从属关系。Python 对代码缩进是硬性要求,这一点必须时刻注意。如果某个代码段的缩进不对,那么整个程序就是错的,要么是语法错误无法执行,要么是逻辑错误导致错误结果,而检查这样的错误会花费很多时间。

(2) 每个 import 语句只导入一个模块,最好按标准库、扩展库、自定义库的顺序依次导入。尽量避免导入整个库,最好只导入确实需要使用的对象,这会让程序运行更快。

(3) 最好在每个类、函数定义和一段完整的功能代码之后增加一个空行,在运算符两侧各增加一个空格,逗号后面增加一个空格。按照这样的规范写出来的代码布局和排版比较松散,阅读起来更加轻松。不论是前面第一条讲的缩进,还是这里谈的空行与空格,主要是提高代码可读性,正如 The Zen of Python 所说: Sparse is better than dense、Readability counts。稍微有点例外的是,在正常的赋值表达式中等号两侧都是各增加一个空格,但在定义函数的默认值参数和使用关键参数调用函数时一般并不在参数赋值的等号两侧增加空格。这样松中有紧也是为了提高代码的可读性,正所谓“张而不弛,文武弗能也;弛而不张,文武弗为也;一张一弛,文武之道也。”