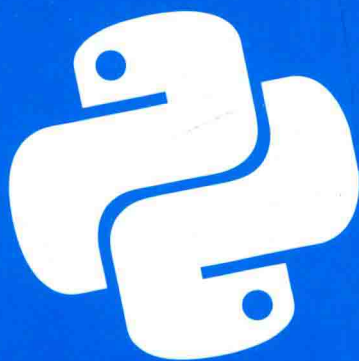


一本书包含 **3** 大板块的内容：**基础、范例、综合实战**

320 多个实例，更多的实践演练机会 / 630 多个拓展范例，真正地举一反三
1920 分钟视频讲解，降低学习难度 / 63 个技术解惑，破解学习难点
77 个课后练习，巩固学习效果 / 综合实例开发，达到学以致用

扫描书中二维码，看视频讲解

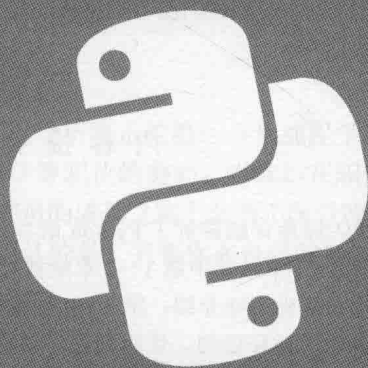


叶维忠◎编著

Python

编程 | 从入门到精通





叶维忠◎编著

Python



编程 | 从入门到精通



人民邮电出版社

北京

图书在版编目 (CIP) 数据

Python编程从入门到精通 / 叶维忠编著. — 北京 :
人民邮电出版社, 2018. 11
ISBN 978-7-115-47880-1

I. ①P… II. ①叶… 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(2018)第024768号

内 容 提 要

本书循序渐进、由浅入深地详细讲解了 Python 语言开发技术,并通过具体实例演练了各个知识点的具体使用流程。全书共 23 章,其中第 1~2 章是基础知识部分,讲解了 Python 语言开发的基础知识,包括搭建开发环境和基础语法介绍;第 3~9 章是核心技术部分,分别讲解了简单数据类型,运算符和表达式,条件语句,循环语句,使用列表,使用元组、字典和集合,使用函数等知识,这些内容都是 Python 语言中最重要的语法知识;第 10~15 章是知识进阶部分,分别讲解了面向对象(上)、面向对象(下)、文件操作处理、异常处理、正则表达式、多线程开发知识,这部分内容是 Python 语言开发技术的重点和核心;第 16~22 章是典型应用部分,分别讲解了 Tkinter 图形化界面开发、网络编程、数据库开发、Python 动态 Web 开发基础、使用 Pygame 开发游戏、使用 Pillow 库处理图形、使用 Matplotlib 实现数据挖掘等知识,这部分内容是读者学习并实践 Python 开发技术的核心;第 23 章是综合实战部分,通过综合实例的实现过程,介绍了 Python 语言在综合项目中的使用流程。全书内容循序渐进,以“技术解惑”和“范例演练”贯穿全书,引领读者全面掌握 Python 语言。

本书不仅适用 Python 语言的初学者,也适合有一定 Python 语言基础的读者学习,还可以作为高等院校相关专业的教学用书和培训学校的教材。

◆ 编 著 叶维忠

责任编辑 张 涛

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>

三河市祥达印刷包装有限公司印刷

◆ 开本: 787×1092 1/16

印张: 28

字数: 745 千字

2018 年 11 月第 1 版

印数: 1-5 000 册

2018 年 11 月河北第 1 次印刷

定价: 79.00 元(附小册子)

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

反盗版热线: (010)81055315

广告经营许可证: 京东工商广登字 20170147 号

前 言

从你开始学习编程的那一刻起，就注定了以后所要走的路——从编程学习者开始，依次经历实习生、程序员、软件工程师、架构师、CTO 等职位的磨砺；当你站在职位顶峰的位置蓦然回首时，会发现自己的成功并不是偶然，在程序员的成长之路上会有不断修改代码、寻找并解决 Bug、不停测试程序和修改项目的经历。不可否认的是，只要你在自己的开发生涯中稳扎稳打，并且善于总结和学习，最终将会得到可喜的收获。

选择一本合适的书

对于一名程序开发初学者来说，究竟如何学习才能提高自己的开发技术呢？答案之一就是买合适的书籍进行学习。但是，市面上许多面向初学者的编程书籍中的大多数篇幅都是基础知识讲解，多偏向于理论，读者读了以后面对实战项目时还是无从下手。如何实现从理论平滑过渡到项目实战，是初学者迫切需要解决的难题，为此，特意编写了本书。

本书用一本书的容量讲解了入门类、范例类和项目实战类 3 类图书的知识，并且对实战知识不是点到为止地讲解，而是深入地探讨。用“纸质书+视频和源程序+网络答疑”的方式，实现了“入门+范例演练+项目实战”的完美呈现，帮助读者从入门顺利过渡到适应项目实战的角色。

本书的特色

1. 以“从入门到精通”的写作方法构建内容，让读者入门容易

为了使读者能够完全看懂本书的内容，本书遵循“从入门到精通”基础类图书的写法，循序渐进地讲解这门开发语言的基本知识。

2. 破解语言难点，“技术解惑”贯穿全书，绕过学习中的陷阱

本书不采用编程语言知识点的罗列式讲解，为了帮助读者学懂基本知识点，书中有“技术解惑”板块，让读者知其然又知其所以然，也就是看得明白，学得通。

3. 全书有 950 多个示例，几乎和“实例大全”类图书同数量级

书中一共有 950 多个示例，其中 320 多个正文实例，一个综合实例。每一个正文实例基本上都穿插加入了与知识点相关的范例，全书额外包含了 630 多个范例。通过对这些实例及范例的练习，实现了对知识点的横向切入和纵向比较，让读者有更多的实践演练机会，并且可以从不同的角度展现一个知识点的用法，真正达到举一反三的效果。

4. 售后 QQ 群提供答疑服务，帮助读者快速解决学习问题

无论书中的疑惑，还是在学习中的问题，作者都将在第一时间为读者解答问题。

5. 视频讲解，降低学习难度

书中每一章均提供声、图并茂的教学视频，这些视频能够引导初学者快速入门，增强学习的信心，从而快速理解所学知识。

6. 贴心提示和注意事项提醒

本书根据需要在文中安排了很多“注意”小板块，让读者可以在学习过程中更轻松地理解相关知识点及概念，更快地掌握个别技术的应用技巧。

7. 源程序+视频+PPT 丰富的学习资料，让学习更轻松

因为本书的内容非常多，不可能用一本书的篇幅囊括“基础+范例+项目案例”的内容，所以需要配备学习资源来辅助实现。在本书的学习资源中不但有全书的源代码，而且还精心制作了实例讲解视频、知识点讲解视频等。本书配套的 PPT 资料可以在网站下载 (www.toppr.net)。读者可以扫描书中提供的二维码观看视频。

8. QQ 群+网站论坛实现教学互动，形成互帮互学的朋友圈

本书作者为了方便给读者答疑，特提供了网站论坛、QQ 群等技术支持，并且随时在线与读者互动。让大家在互学互帮中形成一个良好的学习编程的氛围。

本书的学习论坛网址是：www.toppr.net。

本书的 QQ 群是：292693408。

内容版式

本书的最大特色是实现了入门知识、实例演示、范例演练、技术解惑、综合实战 5 大部分内容的融合。内容由以下模块构成。

① 入门知识：循序渐进地讲解 Python 语言开发的基本知识点。

② 实例演示：遵循理论加实践的教学模式，用 320 多个实例演示了各个入门知识点的用法。

③ 范例演练：为了加深对知识点的融会贯通，每个实例基本上配备了拓展范例，全书共计 630 多个拓展范例，多角度演示了各个入门知识点的用法和技巧。

④ 技术解惑：把读者容易混淆的部分单独用一个板块进行讲解和剖析，对读者所学的知识实现了“拔高”处理。

下面以本书第 4 章为例，展示内容版式的具体结构。

4.4.1 基本赋值运算符和表达式

基本赋值运算符记为“=”，由“=”连接的式子称为赋值表达式。在 Python 语言程序中使用基本赋值运算符的基本格式如下所示。

① 入门知识

变量=表达式

例如，下面代码列出的都是基本的赋值处理：

```
x=a+b           #将x的值赋值为a和b的和
w=sin(a)+sin(b) #将w的值赋值为：sin(a)+sin(b)
y=i+++--j       #将y的值赋值为：i+++--j
```

Python 程序中的变量不需要声明，变量的赋值操作即是变量声明和定义的过程。每个变量在内存中创建，都包括变量的标识、名称和数据这些信息。每个变量在使用前都必须赋值，变量赋值以后该变量才会创建。等号 (=) 用来给变量赋值。等号 (=) 运算符左边是一个变量名，等号 (=) 运算符右边是存储在变量中的值。例如下面的实例代码演示了基本赋值运算符的用法。

② 实例演示
③ 范例演练

实例 4-4

使用赋值运算符

源码路径 daima\4\4-4

实例文件 fuzhi.py 的具体实现代码如下所示。

```
a = 21           #设置a的值是21
b = 10           #设置b的值是10
c = 0            #设置c的值是0
c = a + b        #重新赋值c的值是
                 #a+b，也就是31

print ("1 - c 的值为：", c) #输出c的值
c += a           #设置c=c+a，也就是31+21
print ("2 - c 的值为：", c) #输出c的值
c *= a           #设置c = c * a
print ("3 - c 的值为：", c) #输出c的值
c /= a           #设置c = c / a
print ("4 - c 的值为：", c) #输出c的值
```

拓展范例及视频二维码

范例 031：计算面积和
周长

源码路径：范例\031\



范例 032：多变量赋值并
交换

源码路径：范例\032\



| | |
|-----------------------|--|
| ② 实例 演 示 | <pre> c = 2 c %= a print ("5 - c 的值为: ", c) c **= a print ("6 - c 的值为: ", c) c //= a print ("7 - c 的值为: ", c) </pre> <p>执行后的效果如图 4-5 所示。</p> |
| ③ 范 例 演 练 | <pre> ===== 1 - c 的值为: 31 2 - c 的值为: 52 3 - c 的值为: 1092 4 - c 的值为: 52.0 5 - c 的值为: 2 6 - c 的值为: 2097152 7 - c 的值为: 99864 >>> </pre> <p>图 4-5 执行效果</p> |
| ④ 技 术 解 惑 | <p>4.10 技术解惑</p> <p>4.10.1 “=” 运算符的秘密</p> <p>4.10.2 身份运算符的特质</p> <p>4.10.3 总结 and 和 or 的用法</p> <p>4.10.4 is 运算符和 “=” 运算符的区别</p> |

本书的读者对象

初学编程的自学者

大中专院校的教师和学生

做毕业设计的学生

软件测试人员

在职程序员

编程爱好者

相关培训机构的教师和学员

初、中级程序开发人员

参加实习的初级程序员

资源下载

本书全部源程序请在人民邮电出版社网站 (www.ptpress.com.cn) 下载, 在网站中搜索本书名, 在弹出的页面中单击“资源下载”链接即可下载。

致谢

本书在编写过程中, 得到了人民邮电出版社编辑的大力支持, 正是各位编辑的求实、耐心和效率, 才使得本书能够在这么短的时间内出版。另外, 也十分感谢我的家人给予的巨大支持。本人水平毕竟有限, 书中纰漏之处在所难免, 诚请读者提出意见或建议, 以便修订并使之更臻完善。编辑和投稿联系邮箱: zhangtao@ptpress.com.cn。

最后感谢您购买本书, 希望本书能成为您编程路上的领航者, 祝您阅读快乐!

作者

资源与支持

本书由异步社区出品，社区（<https://www.epubit.com/>）为您提供相关资源和后续服务。

配套资源

本书提供如下资源：

- 本书源代码；
- 本书的视频文件。

要获得以上配套资源，请在异步社区本书页面中点击 **配套资源**，跳转到下载界面，按提示进行操作即可。注意：为保证购书读者的权益，该操作会给出相关提示，要求输入提取码进行验证。

如果您是教师，希望获得教学配套资源，请在社区本书页面中直接联系本书的责任编辑。

提交勘误

作者和编辑尽最大努力来确保书中内容的准确性，但难免会存在疏漏。欢迎您将发现的问题反馈给我们，帮助我们提升图书的质量。

当您发现错误时，请登录异步社区，按书名搜索，进入本书页面，点击“提交勘误”，输入勘误信息，点击“提交”按钮即可。本书的作者和编辑会对您提交的勘误进行审核，确认并接受后，您将获赠异步社区的 100 积分。积分可用于在异步社区兑换优惠券、样书或奖品。

The screenshot shows a web form for reporting errors. At the top, there are three tabs: '详细信息' (Detailed Information), '写书评' (Write a Review), and '提交勘误' (Submit勘误), with the latter being the active tab. Below the tabs, there are three input fields: '页码:' (Page Number), '页内位置 (行数):' (Page Position (Line Number)), and '勘误次数:' (勘误次数). Below these fields is a rich text editor with a toolbar containing icons for bold (B), italic (I), underline (U), strikethrough (ABC), bulleted list (•), numbered list (1-2-3), link (🔗), and unlink (🔗). At the bottom right of the form, there is a '提交' (Submit) button.

扫码关注本书

扫描下方二维码，您将会在异步社区微信服务号中看到本书信息及相关的服务提示。



与我们联系

我们的联系邮箱是 contact@epubit.com.cn。

如果您对本书有任何疑问或建议，请您发邮件给我们，并请在邮件标题中注明本书书名，以便我们更高效地做出反馈。

如果您有兴趣出版图书、录制教学视频，或者参与图书翻译、技术审校等工作，可以发邮件给我们；有意出版图书的作者也可以到异步社区在线提交投稿（直接访问 www.epubit.com/selfpublish/submission 即可）。

如果您是学校、培训机构或企业，想批量购买本书或异步社区出版的其他图书，也可以发邮件给我们。

如果您在网上发现有针对异步社区出品图书的各种形式的盗版行为，包括对图书全部或部分内容的非授权传播，请您将怀疑有侵权行为的链接发邮件给我们。您的这一举动是对作者权益的保护，也是我们持续为您提供有价值的内容的动力之源。

关于异步社区和异步图书

“异步社区”是人民邮电出版社旗下 IT 专业图书社区，致力于出版精品 IT 技术图书和相关学习产品，为译者提供优质出版服务。异步社区创办于 2015 年 8 月，提供大量精品 IT 技术图书和电子书，以及高品质技术文章和视频课程。更多详情请访问异步社区官网 <https://www.epubit.com>。

“异步图书”是由异步社区编辑团队策划出版的精品 IT 专业图书的品牌，依托于人民邮电出版社近 30 年的计算机图书出版积累和专业编辑团队，相关图书在封面上印有异步图书的 LOGO。异步图书的出版领域包括软件开发、大数据、AI、测试、前端、网络技术 etc。



异步社区



微信服务号

目 录

前言 I

第 1 章 Python 如日中天 1
(视频总计 47min, 实例 1 个, 范例两个)

- 1.1 Python 语言基础 2
 - 1.1.1 编程世界的“琅琊榜” 2
 - 1.1.2 Python 为什么这么火 2
 - 1.1.3 Python 语言的特点 3
- 1.2 安装 Python 3
 - 1.2.1 选择版本 4
 - 1.2.2 在 Windows 系统中下载并安装 Python 4
 - 1.2.3 在 Mac 系统中下载并安装 Python 5
 - 1.2.4 在 Linux 系统中下载并安装 Python 6
- 1.3 Python 开发工具介绍 6
 - 1.3.1 使用 IDLE 6
 - 1.3.2 使用 Emacs 7
- 1.4 认识第一段 Python 程序 10
 - 1.4.1 编码并运行 10
 - 1.4.2 其他运行方式 11
- 1.5 技术解惑 12
 - 1.5.1 提高开发效率——安装文本编辑器 12
 - 1.5.2 快速运行 Hello World 程序 12
 - 1.5.3 在终端会话中运行 Python 代码 13
- 1.6 课后练习 13

第 2 章 Python 基础语法 14
(视频总计 85min, 实例 6 个, 范例 12 个)

- 2.1 缩进规则 15
- 2.2 注释 16
- 2.3 编码 17
 - 2.3.1 字符编码 17
 - 2.3.2 Unicode 编码和 UTF-8 编码 17
 - 2.3.3 Python 中的编码 18
- 2.4 标识符和关键字 19
- 2.5 变量 20
- 2.6 输入和输出 21
 - 2.6.1 实现输入功能 22
 - 2.6.2 实现输出功能 22
- 2.7 技术解惑 23
 - 2.7.1 使用注释时的注意事项 23
 - 2.7.2 注意变量的命名规则和建议 23
 - 2.7.3 注意 Python 语言的缩进规则 24
 - 2.7.4 变量赋值的真正意义 24
 - 2.7.5 解码字节流 24
- 2.8 课后练习 24

第 3 章 简单数据类型 25
(视频总计 56min, 实例 6 个, 范例 12 个)

- 3.1 Python 中的数据类型 26
- 3.2 字符串 26
 - 3.2.1 访问字符串中的值 26
 - 3.2.2 更新字符串 27
 - 3.2.3 转义字符 27

| | | | | | |
|-------------------------------|------------------|-----------|--------------------------------|-------------------------|-----------|
| 3.2.4 | 格式化字符串 | 28 | 5.3 | 使用 if...else 语句 | 50 |
| 3.2.5 | 字符串处理函数 | 29 | 5.4 | 使用 if...elif...else 语句 | 51 |
| 3.3 | 数字类型 | 30 | 5.5 | if 语句的嵌套 | 52 |
| 3.3.1 | 整型 | 30 | 5.6 | 实现 switch 语句的功能 | 53 |
| 3.3.2 | 浮点型 | 31 | 5.6.1 | 使用 elif 实现 | 54 |
| 3.3.3 | 布尔型 | 31 | 5.6.2 | 使用字典实现 | 54 |
| 3.3.4 | 复数型 | 32 | 5.6.3 | 自定义编写一个类实现 | 55 |
| 3.4 | 技术解惑 | 33 | 5.7 | 技术解惑 | 56 |
| 3.4.1 | 总结整数支持的运算符 | 33 | 5.7.1 | 剖析 True 和 False 条件判断的用法 | 56 |
| 3.4.2 | 总结 Python 中的数学函数 | 33 | 5.7.2 | 再次提醒不支持 switch 语句的问题 | 56 |
| 3.4.3 | 字符串的格式化技巧 | 34 | 5.7.3 | 最简洁的条件判断语句写法 | 56 |
| 3.5 | 课后练习 | 34 | 5.8 | 课后练习 | 57 |
| 第 4 章 | 运算符和表达式 | 35 | 第 6 章 | 循环语句 | 58 |
| (视频总计 69min, 实例 9 个, 范例 16 个) | | | (视频总计 67min, 实例 11 个, 范例 22 个) | | |
| 4.1 | 什么是运算符和表达式 | 36 | 6.1 | 使用 for 循环语句 | 59 |
| 4.2 | 算术运算符和算术表达式 | 36 | 6.1.1 | 基本的 for 循环语句 | 59 |
| 4.3 | 比较运算符和比较表达式 | 37 | 6.1.2 | 通过序列索引迭代 | 60 |
| 4.4 | 赋值运算符和赋值表达式 | 38 | 6.1.3 | 使用 for... else 循环语句 | 60 |
| 4.4.1 | 基本赋值运算符和表达式 | 39 | 6.1.4 | 嵌套 for 循环语句 | 61 |
| 4.4.2 | 复合赋值运算符和表达式 | 39 | 6.2 | 使用 while 循环语句 | 62 |
| 4.5 | 位运算符和位表达式 | 40 | 6.2.1 | 基本的 while 循环语句 | 62 |
| 4.6 | 逻辑运算符和逻辑表达式 | 41 | 6.2.2 | 使用 while...else 循环语句 | 63 |
| 4.7 | 成员运算符和成员表达式 | 42 | 6.2.3 | 死循环问题 | 63 |
| 4.8 | 身份运算符和身份表达式 | 43 | 6.2.4 | 使用 while 循环嵌套语句 | 64 |
| 4.9 | 运算符的优先级 | 44 | 6.3 | 使用循环控制语句 | 65 |
| 4.10 | 技术解惑 | 45 | 6.3.1 | 使用 break 语句 | 65 |
| 4.10.1 | “=” 运算符的秘密 | 45 | 6.3.2 | 使用 continue 语句 | 66 |
| 4.10.2 | 身份运算符的特质 | 46 | 6.3.3 | 使用 pass 语句 | 67 |
| 4.10.3 | 总结 and 和 or 的用法 | 46 | 6.4 | 技术解惑 | 67 |
| 4.10.4 | is 运算符和“=”运算符的区别 | 47 | 6.4.1 | 总结 for 循环语句 | 67 |
| 4.11 | 课后练习 | 47 | 6.4.2 | 总结 break 和 continue 语句 | 68 |
| 第 5 章 | 条件语句 | 48 | 6.4.3 | 使用 while 循环的注意事项 | 68 |
| (视频总计 68min, 实例 7 个, 范例 14 个) | | | | | |
| 5.1 | 什么是条件语句 | 49 | | | |
| 5.2 | 最简单的 if 语句 | 49 | | | |

| | | | |
|--|-----------|--|-----|
| 6.5 课后练习 | 68 | 8.1.2 修改元组 | 87 |
| 第7章 使用列表 | 69 | 8.1.3 删除元组 | 88 |
| (视频总计 105min, 实例 22 个, 范例 42 个) | | 8.1.4 元组索引和截取 | 88 |
| 7.1 列表类型基础 | 70 | 8.1.5 使用内置方法操作元组 | 89 |
| 7.1.1 创建数字列表 | 70 | 8.2 使用字典 | 89 |
| 7.1.2 访问列表中的值 | 71 | 8.2.1 创建并访问字典 | 89 |
| 7.1.3 使用列表中的值 | 72 | 8.2.2 向字典中添加数据 | 90 |
| 7.2 列表的基本操作 | 73 | 8.2.3 修改字典 | 91 |
| 7.2.1 更新列表元素 | 73 | 8.2.4 删除字典中的元素 | 91 |
| 7.2.2 插入新的元素 | 73 | 8.2.5 创建空字典 | 92 |
| 7.2.3 在列表中删除元素 | 75 | 8.2.6 和字典有关的内置函数 | 92 |
| 7.3 列表排列处理 | 77 | 8.3 遍历字典 | 92 |
| 7.3.1 使用方法 sort()对列表进行 永久性排序 | 77 | 8.3.1 一次性遍历所有的 “键值”对 | 93 |
| 7.3.2 使用方法 sorted()对列表 进行临时排序 | 78 | 8.3.2 遍历字典中的所有键 | 93 |
| 7.3.3 倒序输出列表中的信息 | 78 | 8.3.3 按序遍历字典中的 所有键 | 94 |
| 7.3.4 获取列表的长度 | 79 | 8.3.4 遍历字典中的所有值 | 94 |
| 7.4 列表的高级操作 | 79 | 8.4 字典嵌套 | 95 |
| 7.4.1 列表中的运算符 | 79 | 8.4.1 字典列表 | 95 |
| 7.4.2 列表截取与拼接 | 80 | 8.4.2 在字典中存储字典 | 96 |
| 7.4.3 列表嵌套 | 80 | 8.4.3 在字典中存储列表 | 97 |
| 7.4.4 获取列表元素中的最大值 和最小值 | 81 | 8.5 使用其他内置方法 | 97 |
| 7.4.5 追加其他列表中的值 | 81 | 8.5.1 使用方法 clear()清空 字典 | 97 |
| 7.4.6 在列表中统计某元素出现 的次数 | 82 | 8.5.2 使用方法 copy()复制 字典 | 98 |
| 7.4.7 清空列表中的元素 | 82 | 8.5.3 使用方法 fromkeys()创建 新字典 | 98 |
| 7.4.8 复制列表中的元素 | 83 | 8.5.4 使用方法 get()获取指定的 键值 | 99 |
| 7.4.9 获取列表中某个元素的 索引 | 83 | 8.5.5 使用方法 setdefault()获取 指定的键值 | 99 |
| 7.5 技术解惑 | 84 | 8.5.6 使用方法 update()修改 字典 | 100 |
| 7.5.1 注意排列顺序的多样性 | 84 | 8.6 使用集合 | 100 |
| 7.5.2 尝试有意引发错误 | 84 | 8.7 类型转换 | 101 |
| 7.6 课后练习 | 84 | 8.7.1 内置类型转换函数 | 101 |
| 第8章 使用元组、字典和集合 | 85 | 8.7.2 类型转换综合演练 | 102 |
| (视频总计 137min, 实例 30 个, 范例 56 个) | | | |
| 8.1 使用元组类型 | 86 | | |
| 8.1.1 创建并访问元组 | 86 | | |

| | | | |
|---------------------------------|------------|---------------------------------|-----|
| 8.8 技术解惑 | 103 | 9.9 课后练习 | 121 |
| 8.8.1 for 遍历方式并不是万能的 | 103 | 第 10 章 面向对象 (上) | 122 |
| 8.8.2 两种字典遍历方式的性能对比 | 104 | (视频总计 138min, 实例 22 个, 范例 44 个) | |
| 8.9 课后练习 | 104 | 10.1 定义并使用类 | 123 |
| 第 9 章 使用函数 | 105 | 10.1.1 定义类 | 123 |
| (视频总计 118min, 实例 21 个, 范例 42 个) | | 10.1.2 类的基本用法 | 123 |
| 9.1 函数基础 | 106 | 10.2 类对象 | 124 |
| 9.1.1 定义函数 | 106 | 10.3 类方法 | 124 |
| 9.1.2 调用函数 | 107 | 10.3.1 定义并使用类方法 | 124 |
| 9.2 函数的参数 | 107 | 10.3.2 构造方法 | 125 |
| 9.2.1 形参和实参 | 108 | 10.3.3 方法调用 | 126 |
| 9.2.2 必需参数 | 108 | 10.3.4 创建多个实例 | 127 |
| 9.2.3 关键字参数 | 108 | 10.3.5 使用私有方法 | 128 |
| 9.2.4 默认参数 | 109 | 10.3.6 析构方法 | 129 |
| 9.2.5 不定长参数 | 109 | 10.3.7 静态方法和类方法 | 129 |
| 9.2.6 按值传递参数和按引用传递参数 | 110 | 10.3.8 类的专有方法 | 130 |
| 9.3 函数的返回值 | 111 | 10.4 类属性 | 131 |
| 9.3.1 返回一个简单值 | 111 | 10.4.1 认识属性 | 131 |
| 9.3.2 可选实参 | 111 | 10.4.2 类属性和实例属性 | 131 |
| 9.3.3 返回一个字典 | 112 | 10.4.3 设置属性的默认值 | 132 |
| 9.4 变量的作用域 | 113 | 10.4.4 修改属性的值 | 133 |
| 9.5 使用函数传递列表 | 114 | 10.4.5 使用私有属性 | 135 |
| 9.5.1 访问列表中的元素 | 114 | 10.5 继承 | 135 |
| 9.5.2 在函数中修改列表 | 115 | 10.5.1 定义子类 | 135 |
| 9.6 使用匿名函数 | 115 | 10.5.2 在子类中定义方法和属性 | 137 |
| 9.7 函数和模块开发 | 116 | 10.5.3 子类可以继续派生新类 | 137 |
| 9.7.1 导入整个模块文件 | 116 | 10.5.4 私有属性和私有方法 | 138 |
| 9.7.2 只导入指定的函数 | 117 | 10.5.5 多重继承 | 139 |
| 9.7.3 使用 as 指定函数别名 | 118 | 10.6 方法重写 | 140 |
| 9.7.4 使用 as 指定模块别名 | 118 | 10.7 技术解惑 | 141 |
| 9.7.5 导入所有函数 | 119 | 10.7.1 究竟什么是面向对象 | 141 |
| 9.8 技术解惑 | 119 | 10.7.2 Python 语言的面向对象编程 | 142 |
| 9.8.1 Python 内置函数大全 | 119 | 10.7.3 必须掌握的统一建模语言 | 143 |
| 9.8.2 一个项目引发的问题 | 120 | 10.7.4 构造函数和析构函数的特殊说明 | 143 |
| 9.8.3 使用递归方法展开多层列表 | 120 | | |

| | | | |
|---|------------|---|------------|
| 10.8 课后练习 | 143 | 11.7.3 命名空间的生命周期 | 165 |
| 第 11 章 面向对象 (下) | 144 | 11.7.4 命名空间访问函数 locals() 与 globals() | 165 |
| (视频总计 129min, 实例 32 个, 范例 60 个) | | 11.8 闭包 | 167 |
| 11.1 模块架构 | 145 | 11.8.1 什么是闭包 | 167 |
| 11.1.1 最基本的模块调用 | 145 | 11.8.2 闭包和嵌套函数 | 168 |
| 11.1.2 目录 “ <code>__pycache__</code> ” | 146 | 11.8.3 使用闭包记录函数的 调用次数 | 168 |
| 11.1.3 使用 “ <code>__name__</code> ” 属性 | 147 | 11.8.4 使用闭包实现延迟 请求 | 169 |
| 11.2 使用包 | 148 | 11.8.5 闭包和装饰器 | 169 |
| 11.2.1 表示包 | 148 | 11.8.6 使用闭包定义泛型 函数 | 170 |
| 11.2.2 创建并使用包 | 149 | 11.9 技术解惑 | 171 |
| 11.2.3 实战演练 | 150 | 11.9.1 导入包的秘诀 | 171 |
| 11.3 导入类 | 151 | 11.9.2 无限迭代器的秘密 | 172 |
| 11.3.1 只导入一个类 | 151 | 11.10 课后练习 | 172 |
| 11.3.2 导入指定的类 | 152 | 第 12 章 文件操作处理 | 173 |
| 11.3.3 从一个模块中导入 多个类 | 152 | (视频总计 125min, 实例 29 个, 范例 58 个) | |
| 11.3.4 导入整个模块 | 153 | 12.1 使用 <code>open()</code> 函数打开文件 | 174 |
| 11.3.5 在一个模块中导入另 一个模块 | 153 | 12.2 使用 <code>File</code> 操作文件 | 175 |
| 11.4 迭代器 | 154 | 12.2.1 <code>File</code> 对象介绍 | 175 |
| 11.4.1 什么是迭代器 | 154 | 12.2.2 使用 <code>close()</code> 方法关闭 操作 | 176 |
| 11.4.2 创建并使用迭代器 | 155 | 12.2.3 使用方法 <code>flush()</code> | 176 |
| 11.4.3 使用内置迭代器方法 <code>iter()</code> | 156 | 12.2.4 使用方法 <code>fileno()</code> | 177 |
| 11.5 生成器 | 157 | 12.2.5 使用方法 <code>isatty()</code> | 177 |
| 11.5.1 生成器的运行机制 | 157 | 12.2.6 使用方法 <code>next()</code> | 178 |
| 11.5.2 创建生成器 | 158 | 12.2.7 使用方法 <code>read()</code> | 178 |
| 11.5.3 注意生成器的第一次 调用 | 159 | 12.2.8 使用方法 <code>readline()</code> | 179 |
| 11.5.4 使用协程重置生成器 序列 | 160 | 12.2.9 使用方法 <code>readlines()</code> | 180 |
| 11.6 装饰器 | 160 | 12.2.10 使用方法 <code>seek()</code> | 180 |
| 11.6.1 创建装饰器 | 160 | 12.2.11 使用方法 <code>tell()</code> | 181 |
| 11.6.2 使用装饰器装饰函数 | 161 | 12.2.12 使用方法 <code>truncate()</code> | 182 |
| 11.6.3 使用装饰器装饰类 | 162 | 12.2.13 使用方法 <code>writelines()</code> | 182 |
| 11.7 命名空间 | 163 | 12.3 使用 <code>OS</code> 对象 | 183 |
| 11.7.1 命名空间的本质 | 163 | 12.3.1 <code>OS</code> 对象介绍 | 183 |
| 11.7.2 查找命名空间 | 164 | 12.3.2 使用方法 <code>access()</code> | 185 |
| | | 12.3.3 使用方法 <code>chdir()</code> | 186 |

| | | | | | |
|--------------------------------|---|-----|--------------------------------|---|-----|
| 12.3.4 | 使用方法 chmod() | 186 | 13.5 | 技术解惑 | 209 |
| 12.3.5 | 打开、写入和关闭 | 187 | 13.5.1 | 注意 assert 语句的 妙用 | 209 |
| 12.3.6 | 打开、读取和关闭 | 189 | 13.5.2 | 定义清理行为 | 209 |
| 12.3.7 | 创建目录 | 189 | 13.5.3 | 妙用预定义的清理 行为 | 210 |
| 12.3.8 | 获取目录下的信息 | 190 | 13.6 | 课后练习 | 210 |
| 12.3.9 | 修改目录 | 192 | 第 14 章 正则表达式 | 211 | |
| 12.3.10 | 删除目录 | 193 | (视频总计 53min, 实例 6 个, 范例 12 个) | | |
| 12.4 | 其他常见的文件操作 | 194 | 14.1 | 基本语法 | 212 |
| 12.4.1 | 使用 fileinput 模块 | 194 | 14.1.1 | 普通字符 | 212 |
| 12.4.2 | 批量获取文件名 | 195 | 14.1.2 | 非打印字符 | 212 |
| 12.5 | 技术解惑 | 196 | 14.1.3 | 特殊字符 | 213 |
| 12.5.1 | 注意包含文件的具体 范围 | 196 | 14.1.4 | 限定符 | 216 |
| 12.5.2 | 4 点注意事项 | 196 | 14.1.5 | 定位符 | 217 |
| 12.6 | 课后练习 | 196 | 14.1.6 | 限定范围和否定 | 218 |
| 第 13 章 异常处理 | 197 | | 14.1.7 | 运算符优先级 | 218 |
| (视频总计 53min, 实例 13 个, 范例 26 个) | | | 14.2 | 使用 re 模块 | 218 |
| 13.1 | 语法错误 | 198 | 14.2.1 | re 模块库函数介绍 | 219 |
| 13.2 | 异常处理 | 199 | 14.2.2 | 使用函数 compile() | 219 |
| 13.2.1 | 异常的特殊之处 | 199 | 14.2.3 | 使用函数 match() | 220 |
| 13.2.2 | 使用 “try...except” 处理 异常 | 199 | 14.2.4 | 使用函数 search() | 221 |
| 13.2.3 | 使用 “try...except...else” 处理异常 | 201 | 14.2.5 | 使用函数 findall() | 222 |
| 13.2.4 | 使用 “try...except...finally” 语句 | 202 | 14.2.6 | sub()和 subn()函数 | 223 |
| 13.3 | 抛出异常 | 202 | 14.3 | 使用 Pattern 对象 | 224 |
| 13.3.1 | 使用 raise 抛出异常 | 203 | 14.4 | 正则表达式模式 | 224 |
| 13.3.2 | 使用 assert 语句 | 203 | 14.5 | 技术解惑 | 228 |
| 13.3.3 | 自定义异常 | 204 | 14.5.1 | 生活中的正则表达式 | 228 |
| 13.4 | 内置异常类 | 205 | 14.5.2 | 为什么使用正则表达 | 228 |
| 13.4.1 | 处理 ZeroDivisionError 异常 | 205 | 14.5.3 | 分析函数 re.match()和函 数 re.search()的区别 | 228 |
| 13.4.2 | FileNotFoundError 异常 | 206 | 14.5.4 | 不能将限定符与定位点 一起使用 | 228 |
| 13.4.3 | except 捕获方式 | 207 | 14.6 | 课后练习 | 229 |
| 13.4.4 | 使用函数 testmod() | 207 | 第 15 章 多线程开发 | 230 | |
| 13.4.5 | 使用单元测试函数 testfile() | 208 | (视频总计 68min, 实例 10 个, 范例 20 个) | | |
| | | | 15.1 | 线程和进程基础 | 231 |
| | | | 15.2 | Python 线程处理 | 231 |
| | | | 15.2.1 | 使用 _thread 模块 | 231 |

| | | | | | |
|-----------------------------------|---|-----|--------------------------------|------------------------------|-----|
| 15.2.2 | 使用 threading 模块: threading 模块介绍 | 232 | 16.3.3 | 使用文本框控件 | 251 |
| 15.2.3 | 使用 threading 模块: 直接 在线程中运行函数 | 233 | 16.3.4 | 使用菜单控件 | 252 |
| 15.2.4 | 使用 threading 模块: 通过 继承类 threading.Thread 创建 | 233 | 16.3.5 | 使用标签控件 | 253 |
| 15.2.5 | 使用 threading 模块: 线程 等待 | 234 | 16.3.6 | 使用单选按钮和复选 按钮控件 | 254 |
| 15.2.6 | 使用 threading 模块: 线程 同步 | 234 | 16.3.7 | 使用绘图控件 | 256 |
| 15.3 | 线程优先级队列模块 queue | 236 | 16.4 | Tkinter 库的事件 | 257 |
| 15.3.1 | 模块 queue 中的常用 方法 | 236 | 16.4.1 | Tkinter 事件基础 | 258 |
| 15.3.2 | 基本 FIFO 队列 | 236 | 16.4.2 | 动态绘图程序 | 259 |
| 15.3.3 | LIFO 队列 | 237 | 16.5 | 实现对话框效果 | 261 |
| 15.3.4 | 优先级队列 | 237 | 16.5.1 | 创建消息框 | 261 |
| 15.4 | 使用模块 subprocess 创建 进程 | 238 | 16.5.2 | 创建输入对话框 | 263 |
| 15.4.1 | 模块 subprocess 介绍 | 238 | 16.5.3 | 创建打开/保存文件 对话框 | 264 |
| 15.4.2 | 使用类 Popen 创建 进程 | 240 | 16.5.4 | 创建颜色选择对话框 | 265 |
| 15.5 | 技术解惑 | 242 | 16.5.5 | 创建自定义对话框 | 265 |
| 15.5.1 | 线程带来的意义 你知道吗 | 242 | 16.6 | 技术解惑 | 267 |
| 15.5.2 | 线程和进程的区别 | 244 | 16.6.1 | 格外注意方法 pack() 的 参数 | 267 |
| 15.6 | 课后练习 | 244 | 16.6.2 | 请务必注意方法 grid() 的 参数 | 267 |
| 第 16 章 Tkinter 图形化界面开发 245 | | | 16.6.3 | 请务必注意方法 place() 的属性 | 267 |
| (视频总计 88min, 实例 14 个, 范例 28 个) | | | 16.7 | 课后练习 | 267 |
| 16.1 | Python 图形化界面开发基础 | 246 | 第 17 章 网络编程 268 | | |
| 16.1.1 | GUI 介绍 | 246 | (视频总计 96min, 实例 12 个, 范例 24 个) | | |
| 16.1.2 | 使用 Python 语言编写 GUI 程序 | 246 | 17.1 | 网络开发基础 | 269 |
| 16.2 | Tkinter 开发基础 | 247 | 17.1.1 | OSI 七层网络模型 | 269 |
| 16.2.1 | 第一个 Tkinter 程序 | 247 | 17.1.2 | TCP/IP 协议 | 270 |
| 16.2.2 | 向窗体中添加组件 | 248 | 17.2 | 套接字编程 | 270 |
| 16.3 | Tkinter 组件开发详解 | 248 | 17.2.1 | socket() 函数介绍 | 270 |
| 16.3.1 | Tkinter 组件概览 | 249 | 17.2.2 | socket 对象的内置函数和 属性 | 271 |
| 16.3.2 | 使用按钮控件 | 250 | 17.2.3 | 使用套接字建立 TCP “客 户端/服务器” 连接 | 272 |
| | | | 17.2.4 | 使用套接字建立 UDP “客 户端/服务器” 连接 | 274 |
| | | | 17.3 | socketserver 编程 | 275 |
| | | | 17.3.1 | socketserver 模块 | |

| | | | |
|--|------------|--------------------------------------|------------|
| 基础 | 275 | 18.2.7 数据库删除操作 | 304 |
| 17.3.2 使用 socketserver 创建 TCP “客户端/服务器” 连接 | 276 | 18.2.8 执行事务 | 304 |
| 17.4 HTTP 协议开发 | 277 | 18.3 使用 MariaDB 数据库 | 305 |
| 17.4.1 使用 urllib 包 | 277 | 18.3.1 搭建 MariaDB 数据库 环境 | 305 |
| 17.4.2 使用 HTTP 包 | 280 | 18.3.2 在 Python 程序中使用 MariaDB 数据库 | 308 |
| 17.5 收发电子邮件 | 281 | 18.4 使用 MongoDB 数据库 | 309 |
| 17.5.1 开发 POP3 邮件协议 程序 | 281 | 18.4.1 搭建 MongoDB 环境 | 309 |
| 17.5.2 开发 SMTP 邮件协议 程序 | 283 | 18.4.2 在 Python 程序中使用 MongoDB 数据库 | 310 |
| 17.6 开发 FTP 文件传输程序 | 285 | 18.5 使用适配器 | 312 |
| 17.6.1 Python 和 FTP | 285 | 18.6 使用 ORM 操作数据库 | 316 |
| 17.6.2 创建一个 FTP 文件传输 客户端 | 287 | 18.6.1 Python 和 ORM | 316 |
| 17.7 解析 XML | 289 | 18.6.2 使用 SQLAlchemy | 317 |
| 17.7.1 SAX 解析方法 | 289 | 18.6.3 使用 mongoengine | 319 |
| 17.7.2 DOM 解析方法 | 291 | 18.7 技术解惑 | 321 |
| 17.8 解析 JSON 数据 | 292 | 18.7.1 灵活使用查询运算符 | 321 |
| 17.8.1 类型转换 | 292 | 18.7.2 掌握 between 关键字的 用法 | 322 |
| 17.8.2 编码和解码 | 293 | 18.7.3 了解关联表操作的 秘密 | 322 |
| 17.9 技术解惑 | 294 | 18.7.4 请课外学习并掌握 SQL 语言的知识 | 322 |
| 17.9.1 详细剖析客户端/服务器 编程模型 | 294 | 18.8 课后练习 | 322 |
| 17.9.2 详细剖析类 HTTPConnection 中的 方法 | 295 | 第 19 章 Python 动态 Web 开发基础 | 323 |
| 17.10 课后练习 | 296 | (视频总计 91min, 实例 18 个, 范例 36 个) | |
| 第 18 章 数据库开发 | 297 | 19.1 Python CGI 编程 | 324 |
| (视频总计 88min, 实例 13 个, 范例 26 个) | | 19.1.1 CGI 介绍 | 324 |
| 18.1 操作 SQLite3 数据库 | 298 | 19.1.2 搭建 CGI 服务器 | 324 |
| 18.2 操作 MySQL 数据库 | 299 | 19.1.3 第一个 CGI 程序 | 325 |
| 18.2.1 搭建 PyMySQL 环境 | 300 | 19.2 使用 Tornado 框架 | 325 |
| 18.2.2 实现数据库连接 | 300 | 19.2.1 Tornado 框架介绍 | 326 |
| 18.2.3 创建数据库表 | 301 | 19.2.2 Python 和 Tornado 框架 | 326 |
| 18.2.4 数据库插入操作 | 302 | 19.2.3 获取请求参数 | 327 |
| 18.2.5 数据库查询操作 | 302 | 19.2.4 使用 cookie | 329 |
| 18.2.6 数据库更新操作 | 303 | 19.2.5 URL 转向 | 330 |
| | | 19.2.6 使用静态资源文件 | 331 |

| | | | |
|---------------------------------------|------------|---------------------------------------|------------|
| 19.3 使用 Django 框架..... | 332 | 20.3.1 规划图形..... | 364 |
| 19.3.1 搭建 Django 环境..... | 332 | 20.3.2 具体实现..... | 365 |
| 19.3.2 常用的 Django 命令..... | 333 | 20.4 技术解惑..... | 371 |
| 19.3.3 第一个 Django 工程..... | 334 | 20.4.1 电脑游戏开发的必备 知识..... | 371 |
| 19.3.4 在 URL 中传递参数..... | 335 | 20.4.2 如何创建 Surface 对象..... | 372 |
| 19.3.5 使用模板..... | 337 | 20.5 课后练习..... | 372 |
| 19.3.6 使用表单..... | 341 | 第 21 章 使用 Pillow 库处理图形..... | 373 |
| 19.3.7 实现基本的数据库 操作..... | 342 | (视频总计 74min, 实例 16 个, 范例 32 个) | |
| 19.3.8 使用 Django 后台系统 开发博客系统..... | 343 | 21.1 安装 Pillow 库..... | 374 |
| 19.4 使用 Flask 框架..... | 346 | 21.2 使用 Image 模块..... | 374 |
| 19.4.1 开始使用 Flask 框架..... | 346 | 21.2.1 打开和新建..... | 374 |
| 19.4.2 传递 URL 参数..... | 347 | 21.2.2 混合..... | 376 |
| 19.4.3 使用 session 和 cookie..... | 349 | 21.2.3 复制和缩放..... | 377 |
| 19.4.4 文件上传..... | 350 | 21.2.4 粘贴和裁剪..... | 378 |
| 19.5 技术解惑..... | 351 | 21.2.5 格式转换..... | 379 |
| 19.5.1 “客户端/服务器”开发 模式..... | 351 | 21.2.6 重设和旋转..... | 380 |
| 19.5.2 Python Web 客户端开发 是大势所趋..... | 351 | 21.2.7 分离和合并..... | 381 |
| 19.5.3 注意 Python 3 的 变化..... | 352 | 21.2.8 滤镜..... | 382 |
| 19.6 课后练习..... | 352 | 21.2.9 其他内置函数..... | 382 |
| 第 20 章 使用 Pygame 开发游戏..... | 353 | 21.3 使用 ImageChops 模块..... | 384 |
| (视频总计 59min, 实例 8 个, 范例 16 个) | | 21.3.1 常用的内置函数..... | 384 |
| 20.1 安装 Pygame..... | 354 | 21.3.2 实现图片合成..... | 385 |
| 20.2 Pygame 开发基础..... | 355 | 21.4 使用 ImageEnhance 模块..... | 386 |
| 20.2.1 Pygame 框架中的 模块..... | 355 | 21.4.1 常用的内置函数..... | 386 |
| 20.2.2 事件操作..... | 356 | 21.4.2 实现图像增强处理..... | 386 |
| 20.2.3 显示模式设置..... | 359 | 21.5 使用 ImageFilter 模块..... | 387 |
| 20.2.4 字体处理..... | 359 | 21.5.1 常用的内置函数..... | 387 |
| 20.2.5 像素和颜色处理..... | 360 | 21.5.2 实现滤镜处理..... | 388 |
| 20.2.6 使用 Surface 绘制 图像..... | 362 | 21.6 使用 ImageDraw 模块..... | 388 |
| 20.2.7 使用 pygame.draw 绘图 函数..... | 362 | 21.6.1 常用的内置函数..... | 388 |
| 20.3 开发一个俄罗斯方块游戏..... | 364 | 21.6.2 绘制二维图像..... | 390 |
| | | 21.7 使用 ImageFont 模块..... | 390 |
| | | 21.8 技术解惑..... | 391 |
| | | 21.8.1 详细剖析 ImageFont 模块的内置函数..... | 391 |
| | | 21.8.2 必须掌握并深入理解的 几个概念..... | 392 |