



励志改变人生
编程改变命运

Zero
Basis

零基础学

Python (第2版)

黄传祿 张克强 赵越◎编著

本书特色

- ◎ 由浅入深，循序渐进，从零开始学Python语言，一点都不难
- ◎ 编程基础、编程进阶、编程应用、项目实战、上机练习
- ◎ 300个实例、64个练习题

超值赠送

- ◎ 本书教学视频
- ◎ 本书源代码
- ◎ 本书教学PPT

赠送资料获取地址

www.hzbook.com

本书技术支持

论坛：<http://www.rzchina.net>

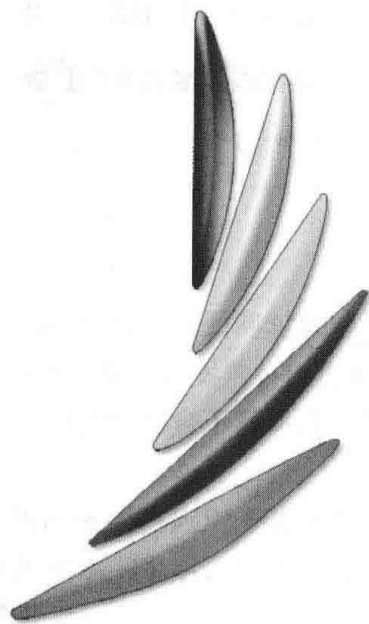


机械工业出版社
China Machine Press

Zero
Basis

零基础学

Python (第2版)



黄传祿 张克强 赵越◎编著



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

零基础学 Python / 黄传禄, 张克强, 赵越编著. —2 版. —北京: 机械工业出版社, 2020.5

ISBN 978-7-111-65535-0

I. 零… II. ①黄… ②张… ③赵… III. 软件工具—程序设计 IV. TP311.561

中国版本图书馆 CIP 数据核字 (2020) 第 081219 号

零基础学 Python (第 2 版)

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 李华君

责任校对: 殷虹

印刷: 北京市荣盛彩色印刷有限公司

版次: 2020 年 6 月第 2 版第 1 次印刷

开本: 185mm × 260mm 1/16

印张: 29.75

书号: ISBN 978-7-111-65535-0

定价: 99.00 元

客服电话: (010) 88361066 88379833 68326294

投稿热线: (010) 88379604

华章网站: www.hzbook.com

读者信箱: hzit@hzbook.com

版权所有·侵权必究

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

前 言

作为最流行的脚本语言之一，Python 具有内置的高级数据结构和简单有效的面向对象编程思想实现。同时，其语法简洁清晰，类库丰富强大，非常适合进行快速原型开发。另外，Python 可以运行在多种系统平台下，从而使得只需要编写一次代码，就可以在多个系统平台下都保持同等的功能。

为了使广大读者既能够掌握 Python 语言的基础知识，又能够将 Python 语言应用于某个特定的领域（如 Web 开发），本书将全面介绍和 Python 相关的内容。在学习完本书之后，读者应该能够很好地掌握 Python 语言，同时可以使用 Python 语言进行实际项目的开发。

本书特点

1. 循序渐进，由浅入深

为了方便读者学习，本书首先让读者了解 Python 的历史和特点，再通过具体的例子逐渐把读者带入 Python 的世界，使他们掌握 Python 语言的基本知识要点以及基础类库、常用库和工具的使用。

2. 技术全面，内容充实

本书在保证内容实用的前提下，详细介绍了 Python 语言的各个知识点。同时，本书所涉及的内容非常全面，无论从事什么行业的读者，都可以从本书找到可应用 Python 于自身所处行业的地方。

3. 对比讲解，理解深刻

有很多读者具备 Java 开发经验，因此本书注意结合 Python 与 Java 语法的异同点进行讲解。同时本书注意对 Python 中相似的函数和方法进行对比。通过对比讲解的方式，帮助读者解决一些疑难问题，加深读者对 Python 语法要点的理解。

4. 分析原理，步骤清晰

每种编程语言都有自己独特的魅力。掌握一门技术首先需要理解原理，本书注意讲解各个知识点的原理，总结对应的实现思路和步骤。读者可以根据具体步骤实现书中的例子，理论结合实践更利于学习。

5. 代码完整，讲解详尽

书中的每个知识点都配有一段示例代码，代码的关键点也有注释说明。每段代码的后面都有详细的分析，同时给出了代码运行结果。读者可以参考运行结果阅读源程序（可从华章官网

hzbook.com 下载获得，为了和本书引用的路径相符，读者可将其下载至 D 盘），加深对程序的理解。

本书内容

第 1 章：如果读者还是一个新手，可通过这一章了解 Python 的由来，Python 的特性和优势，逐渐步入 Python 的世界。

第 2 章：详细介绍 Python 的语法知识，深入讲解 Python 的编码规范、变量和常量的声明及使用、数据类型、运算符与表达式。通过该章的学习，读者能够掌握一些 Python 编码规范以及基本概念。

第 3 章：详细介绍 Python 中的控制语句以及一些习惯用法，结合示例讲解 Python 结构化程序的要点。

第 4 章：介绍 Python 的内置数据结构——元组、列表、字典和序列。该章根据使用习惯分别介绍这些内置数据结构的特点以及区别。

第 5 章：讲解 Python 中模块和函数的概念，重点介绍 Python 的常用内置模块、函数的参数、递归函数、lambda 函数、Generator 函数等内容。

第 6 章：重点介绍 Python 中字符串的处理（包括字符串的格式化、合并、截取、比较、查找、替换等），讲解正则表达式的概念以及使用 re 模块处理正则表达式。

第 7 章：介绍 Python 对文件的基本操作（包括文件的创建、读写、删除、复制、重命名、搜索、替换和比较），重点介绍 Python 对目录的遍历。

第 8 章：介绍面向对象编程的要点，重点讲解 Python 如何实现面向对象的特性，以及 Python 中的设计模式。

第 9 章：详细介绍 Python 对异常的处理、对异常的捕获和抛出、自定义异常等内容，讲解如何使用 IDLE 和 Eclipse IDE 调试 Python 程序。

第 10 章：介绍 Python 数据库编程，通过实例重点讲解了 Python 使用专用模块连接并操作 Oracle、MySQL、SQL Server、SQLite 数据库的方法。

第 11 章：介绍 Python 的 GUI 开发，主要讲解当前 Python 中使用比较多的 Tkinter 和 PyQt。

第 12 章：介绍 Python 自带的 GUI 开发库 Tkinter 的基本组件及其使用方法，并给出每种组件的详细示例代码与图示。

第 13 章：介绍 Python 的 HTML 应用，详细介绍 URL 的解析以及 HTML 资源的获取，同时对 CGI 的使用和 HTML 文档的解析进行介绍。

第 14 章：讲解 Python 的 XML 操作，包括 XML 的各种内容实体，还对 XML 文档的两种处理方式——SAX 和 DOM 进行详细的讲解。

第 15 章：对 Python 中常用的 Web 开发框架进行介绍。在介绍 MVC 和 MVT 模式的基础上，对 Django 框架进行详细的讲解，包括它的基本使用方法和高级使用方法。

第 16 章：介绍 Python 中的测试框架，主要讲解 Python 中的两种测试框架——unittest 和 doctest。讲解的时候，都配以示例，以便于读者掌握。

第 17 章：介绍 Python 中进程和线程的概念，主要内容包括进程和线程的创建及管理。同时，

重点讲解多线程环境下的数据同步机制。

第 18 章：介绍 Python 的系统管理，其中主要讲解 IPython 的应用，不仅包括 IPython 的介绍，还包括其常见应用。

第 19 章：讲解 Python 中和网络编程相关的内容，包括服务器端和客户端的通信，随后着重介绍网络中的异步通信方式，最后对 Twisted 框架进行介绍。

第 20 章：讲解 Python 开发中常见的网络应用，主要包括文件传输、邮件的获取和发送、远程登录。

第 21 章：讲解 Python 图像处理，通过各种示例，读者能够迅速掌握相关知识点并用于实践。

第 22 章：介绍 Windows 下的 Python 开发，主要介绍组件对象模型。同时，使用示例来讲解 Windows 下的常见 Python 应用。

受众

- Python 爱好者
- 高等教育学校的学生
- 社会培训班的学生
- 系统管理员
- 界面开发人员
- Web 开发人员
- 网络编程人员

本书作者

本书由黄传禄、张克强、赵越编著，黄传禄负责设计全书的总体结构和撰写。本书在编写的过程中得到了南昌翰诚信息咨询有限公司相关人员的支持和帮助，并参考了作者所在软件培训公司的一些工作成果和相关文献。在此对曾为本书做出贡献的人员表示诚挚的谢意。

由于作者水平有限，书中疏漏、错误之处在所难免，敬请同行及广大读者批评指正。

黄传禄

2020 年 1 月

目 录

前言

第一篇 Python语言基础

第 1 章 进入 Python 的世界	1
1.1 Python 的由来	1
1.2 Python 的特性	2
1.3 第一个 Python 程序	3
1.4 搭建开发环境	4
1.4.1 Python 的下载和安装	4
1.4.2 交互式命令行的使用	10
1.5 Python 的开发工具	11
1.5.1 PyCharm 的使用	11
1.5.2 Eclipse IDE 的介绍	13
1.5.3 EditPlus 编辑器环境的配置	15
1.6 不同平台下的 Python	17
1.7 小结	17
1.8 习题	18
第 2 章 必须知道的 Python 基础语法	19
2.1 Python 的文件类型	19
2.1.1 源代码	19
2.1.2 字节代码	19
2.1.3 优化代码	20
2.2 Python 的编码规范	20
2.2.1 命名规则	20
2.2.2 代码缩进与冒号	23
2.2.3 模块导入的规范	24
2.2.4 使用空行分隔代码	25
2.2.5 正确的注释	26

2.2.6 语句的分隔	28
2.3 变量和常量	30
2.3.1 变量的命名	30
2.3.2 变量的赋值	31
2.3.3 局部变量	32
2.3.4 全局变量	32
2.3.5 常量	34
2.4 数据类型	35
2.4.1 数字	35
2.4.2 字符串	37
2.5 运算符与表达式	38
2.5.1 算术运算符和算术表达式	38
2.5.2 关系运算符和关系表达式	40
2.5.3 逻辑运算符和逻辑表达式	40
2.6 小结	41
2.7 习题	42
第 3 章 Python 的控制语句	43
3.1 结构化程序设计	43
3.2 条件判断语句	44
3.2.1 if 条件语句	44
3.2.2 if...elif...else 判断语句	46
3.2.3 if 语句也可以嵌套	46
3.2.4 switch 语句的替代方案	48
3.3 循环语句	50
3.3.1 while 循环	50
3.3.2 for 循环	51
3.3.3 break 和 continue 语句	53
3.4 结构化程序示例	54
3.5 小结	56
3.6 习题	57
第 4 章 Python 数据结构	58
4.1 元组结构	58
4.1.1 元组的创建	58
4.1.2 元组的访问	59
4.1.3 元组的遍历	61
4.2 列表结构	62
4.2.1 列表的创建	62

4.2.2	列表的使用	64
4.2.3	列表的查找、排序、反转	65
4.2.4	列表实现堆栈和队列	66
4.3	字典结构	68
4.3.1	字典的创建	68
4.3.2	字典的访问	69
4.3.3	字典的方法	71
4.3.4	字典的排序、复制	74
4.3.5	全局字典——sys.modules 模块	76
4.4	序列	77
4.5	小结	79
4.6	习题	79
第 5 章	模块与函数	80
5.1	Python 程序的结构	80
5.2	模块	81
5.2.1	模块的创建	81
5.2.2	模块的导入	81
5.2.3	模块的属性	83
5.2.4	模块的内置函数	83
5.2.5	自定义包	86
5.3	函数	88
5.3.1	函数的定义	88
5.3.2	函数的参数	89
5.3.3	函数的返回值	92
5.3.4	函数的嵌套	95
5.3.5	递归函数	97
5.3.6	lambda 函数	98
5.3.7	Generator 函数	99
5.4	小结	101
5.5	习题	101
第 6 章	字符串与正则表达式	102
6.1	常见的字符串操作	102
6.1.1	字符串的格式化	102
6.1.2	字符串的转义符	104
6.1.3	字符串的合并	106
6.1.4	字符串的截取	107
6.1.5	字符串的比较	108

6.1.6	字符串的反转	109
6.1.7	字符串的查找和替换	110
6.1.8	字符串与日期的转换	112
6.2	正则表达式应用	113
6.2.1	正则表达式简介	113
6.2.2	使用 re 模块处理正则表达式	115
6.3	小结	121
6.4	习题	121
第 7 章	使用 Python 处理文件	122
7.1	文件的常见操作	122
7.1.1	文件的创建	122
7.1.2	文件的读取	124
7.1.3	文件的写入	126
7.1.4	文件的删除	127
7.1.5	文件的复制	128
7.1.6	文件的重命名	129
7.1.7	文件内容的搜索和替换	130
7.1.8	文件的比较	131
7.1.9	配置文件的访问	133
7.2	目录的常见操作	135
7.2.1	创建和删除目录	135
7.2.2	目录的遍历	136
7.3	文件处理示例——文件属性浏览程序	137
7.4	小结	139
7.5	习题	140
第 8 章	面向对象编程	141
8.1	面向对象的概述	141
8.2	类和对象	142
8.2.1	类和对象的区别	142
8.2.2	类的定义	143
8.2.3	对象的创建	143
8.3	属性和方法	144
8.3.1	类的属性	144
8.3.2	类的方法	146
8.3.3	内部类的使用	148
8.3.4	__init__ 方法	149
8.3.5	__del__ 方法	150

8.3.6	垃圾回收机制	151
8.3.7	类的内置方法	152
8.3.8	方法的动态特性	156
8.4	继承	157
8.4.1	使用继承	157
8.4.2	抽象基类	159
8.4.3	多态性	160
8.4.4	多重继承	161
8.4.5	Mixin 机制	163
8.5	运算符的重载	164
8.6	Python 与设计模式	167
8.6.1	设计模式简介	167
8.6.2	设计模式示例——Python 实现工厂方法	167
8.7	小结	168
8.8	习题	169
第 9 章	异常处理与程序调试	170
9.1	异常的处理	170
9.1.1	Python 中的异常	170
9.1.2	try...except 的使用	171
9.1.3	try...finally 的使用	173
9.1.4	使用 raise 抛出异常	174
9.1.5	自定义异常	174
9.1.6	assert 语句的使用	175
9.1.7	异常信息	176
9.2	使用自带 IDLE 调试程序	177
9.3	使用 Eclipse IDE 调试程序	179
9.3.1	新建工程	179
9.3.2	配置调试	179
9.3.3	设置断点	182
9.4	小结	183
9.5	习题	184
第 10 章	Python 数据库编程	185
10.1	数据库简介	185
10.2	Python DATABASE API 简介	186
10.3	Python 连接数据库的专用模块	187
10.3.1	数据库专用模块的下载与安装	187
10.3.2	Python 连接 Oracle 数据库	190

10.3.3 Python 连接 MySQL 数据库	191
10.3.4 Python 连接 SQL Server 数据库	194
10.3.5 Python 连接 SQLite 数据库	196
10.4 使用 Python 的持久化模块读写数据	197
10.5 小结	198
10.6 习题	198

第二篇 Python的GUI程序设计

第 11 章 Python 的 GUI 开发	199
11.1 Python 的 GUI 开发选择	199
11.1.1 认识 Python 内置的 GUI 库 Tkinter	199
11.1.2 使用 Tkinter 进行开发	200
11.1.3 认识 PyQt GUI 库	203
11.1.4 使用 PyQt GUI 库进行开发	203
11.2 小结	204
11.3 习题	204
第 12 章 GUI 编程与 Tkinter 相关组件介绍	205
12.1 GUI 程序开发简介	205
12.2 Tkinter 及其主要组件	206
12.2.1 在程序中使用 Tkinter	206
12.2.2 顶层窗口	206
12.2.3 标签	207
12.2.4 框架	207
12.2.5 按钮	208
12.2.6 输入框	208
12.2.7 单选按钮	209
12.2.8 复选按钮	209
12.2.9 消息	210
12.2.10 滚动条	211
12.2.11 列表框	211
12.3 Tkinter 所有组件简介	212
12.4 小结	212
12.5 习题	213

第三篇 Python的Web开发

第 13 章 Python 的 HTML 应用	215
13.1 HTML 介绍	215

13.1.1	HTML 的历史	215
13.1.2	SGML、HTML、XHTML、HTML5 的关系	216
13.1.3	HTML 的标签	216
13.1.4	HTML 的框架组成	217
13.2	URL 的处理	218
13.2.1	统一资源定位符	218
13.2.2	URL 的解析	219
13.2.3	URL 的拼合	221
13.2.4	URL 的分解	222
13.2.5	URL 的编解码	223
13.2.6	中文的编解码	225
13.2.7	查询参数的编码	226
13.3	CGI 的使用	228
13.3.1	CGI 介绍	228
13.3.2	获取 CGI 环境信息	230
13.3.3	解析用户的输入	232
13.4	获取 HTML 资源	235
13.4.1	使用 <code>urlopen</code> 和 <code>urlretrieve</code> 获取 HTTP 资源	235
13.4.2	分析返回资源的相关信息	240
13.4.3	使用 <code>http.client</code> 模块获取资源	242
13.5	HTML 文档的解析	244
13.6	小结	246
13.7	习题	246
第 14 章	Python 和 XML	247
14.1	XML 介绍	247
14.1.1	XML 的演进历史	247
14.1.2	XML 的优点和限制	248
14.1.3	XML 技术的 Python 支持	249
14.2	XML 文档概览和验证	250
14.2.1	XML 文档的基础概念	250
14.2.2	XML 文档的结构良好性验证	251
14.2.3	XML 文档的有效性验证	253
14.3	分析 XML 文档结构	255
14.3.1	XML 的元素和标签	256
14.3.2	元素的属性	258
14.3.3	XML 的名字	259
14.3.4	字符实体	261

14.3.5	CDATA 段	262
14.3.6	注释	264
14.3.7	处理指令	265
14.3.8	XML 定义	266
14.4	使用 SAX 处理 XML 文档	266
14.4.1	SAX 介绍	266
14.4.2	SAX 处理的组成部分	267
14.5	使用 DOM 处理 XML 文档	271
14.5.1	DOM 介绍	271
14.5.2	xml.dom 模块中的接口操作	272
14.5.3	对 XML 文档的操作	280
14.6	小结	286
14.7	习题	286
第 15 章	Python 的 Web 开发——Django 框架的应用	287
15.1	常见的 Web 开发框架	287
15.2	MVC 模式	290
15.2.1	MVC 模式介绍	290
15.2.2	MVC 模式的优缺点	290
15.2.3	Django 框架中的 MVT	291
15.3	Django 开发环境的搭建	292
15.3.1	Django 框架的安装	292
15.3.2	数据库的配置	293
15.4	Django 框架的基本应用	294
15.4.1	Web 应用的创建	294
15.4.2	Django 中的开发服务器	296
15.4.3	创建数据库	297
15.4.4	生成 Django 应用	298
15.4.5	创建数据模型	299
15.4.6	URL 设计	300
15.4.7	创建视图	302
15.4.8	模板系统	303
15.5	Django 框架的高级应用	305
15.5.1	管理界面	305
15.5.2	生成数据库数据	307
15.5.3	Session 功能	308
15.6	小结	310
15.7	习题	310

第四篇 Python的其他应用

第 16 章 敏捷方法学在 Python 中的应用——测试驱动开发	311
16.1 测试驱动开发	311
16.1.1 测试驱动开发模式	311
16.1.2 TDD 的优势	312
16.1.3 TDD 的使用步骤	313
16.2 unittest 框架	314
16.2.1 unittest 模块介绍	314
16.2.2 构建测试用例	315
16.2.3 构建测试固件	316
16.2.4 组织多个测试用例	318
16.2.5 构建测试套件	319
16.2.6 重构代码	321
16.2.7 执行测试	322
16.3 使用 doctest 进行测试	324
16.3.1 doctest 模块介绍	324
16.3.2 构建可执行文档	325
16.3.3 执行 doctest 测试	326
16.4 小结	328
16.5 习题	328
第 17 章 Python 中的进程和线程	329
17.1 进程和线程	329
17.1.1 进程和线程的概念	329
17.1.2 Python 中对于进程和线程处理的支持	330
17.2 Python 下的进程编程	331
17.2.1 进程运行环境	331
17.2.2 创建进程	332
17.2.3 终止进程	333
17.3 使用 subprocess 模块管理进程	334
17.3.1 使用 Popen 类管理进程	334
17.3.2 调用外部系统命令	337
17.3.3 替代其他进程创建函数	337
17.4 进程间的信号机制	338
17.4.1 信号的处理	338
17.4.2 信号使用的规则	340
17.5 多线程概述	340
17.5.1 什么是多线程	341

17.5.2	线程的状态	341
17.5.3	Python 中的线程支持	341
17.6	生成和终止线程	342
17.6.1	使用 <code>_thread</code> 模块	342
17.6.2	使用 <code>threading.Thread</code> 类	344
17.7	管理线程	346
17.7.1	线程状态转移	346
17.7.2	主线程对子线程的控制	347
17.7.3	线程中的局部变量	348
17.8	线程之间的同步	349
17.8.1	临界资源和临界区	349
17.8.2	锁机制	351
17.8.3	条件变量	352
17.8.4	信号量	355
17.8.5	同步队列	355
17.8.6	线程同步小结	356
17.9	小结	357
17.10	习题	357
第 18 章	基于 Python 的系统管理	358
18.1	增强的交互式环境 IPython	358
18.1.1	IPython 介绍	358
18.1.2	IPython 的安装	359
18.1.3	IPython 的启动	360
18.1.4	IPython 的环境配置	360
18.2	和 IPython 的简单交互	360
18.2.1	IPython 中的输入和输出	360
18.2.2	输出提示符的区别	361
18.2.3	输出提示符有区别的原因	362
18.3	IPython 中的 magic 函数	363
18.3.1	magic 函数的使用和构造	364
18.3.2	目录管理	364
18.3.3	对象信息的收集	368
18.3.4	magic 函数小结	370
18.4	IPython 适合于系统管理的特点	371
18.4.1	Tab 补全	371
18.4.2	历史记录功能	372
18.4.3	执行外部系统命令和运行文件	374

18.4.4	对象查看和自省	377
18.4.5	直接编辑代码	379
18.4.6	设置别名和宏	381
18.5	使用 Python 进行文件管理	383
18.5.1	文件的比较	383
18.5.2	文件的归档	384
18.5.3	文件的压缩	387
18.6	使用 Python 定时执行任务	389
18.6.1	使用休眠功能	389
18.6.2	使用 sched 模块来定时执行任务	389
18.7	小结	390
18.8	习题	390
第 19 章	Python 和网络编程	391
19.1	网络模型介绍	391
19.1.1	OSI 简介	391
19.1.2	TCP/IP 简介	392
19.2	Socket 应用	393
19.2.1	Socket 基础	393
19.2.2	Socket 的工作方式	393
19.3	服务器端和客户端通信	394
19.3.1	服务器端的构建	394
19.3.2	客户端的构建	399
19.4	异步通信方式	400
19.4.1	使用 Fork 方式	400
19.4.2	使用线程方式	401
19.4.3	使用异步 IO 方式	402
19.4.4	使用 asyncore 模块	405
19.5	Twisted 网络框架	408
19.5.1	Twisted 框架介绍	408
19.5.2	Twisted 框架下服务器端的实现	409
19.5.3	Twisted 框架下服务器端的其他处理	410
19.6	小结	413
19.7	习题	413
第 20 章	常见的 Python 网络应用	414
20.1	使用 FTP 传输文件	414
20.1.1	FTP 的工作原理和 Python 库支持	414
20.1.2	FTP 的登录和退出	415