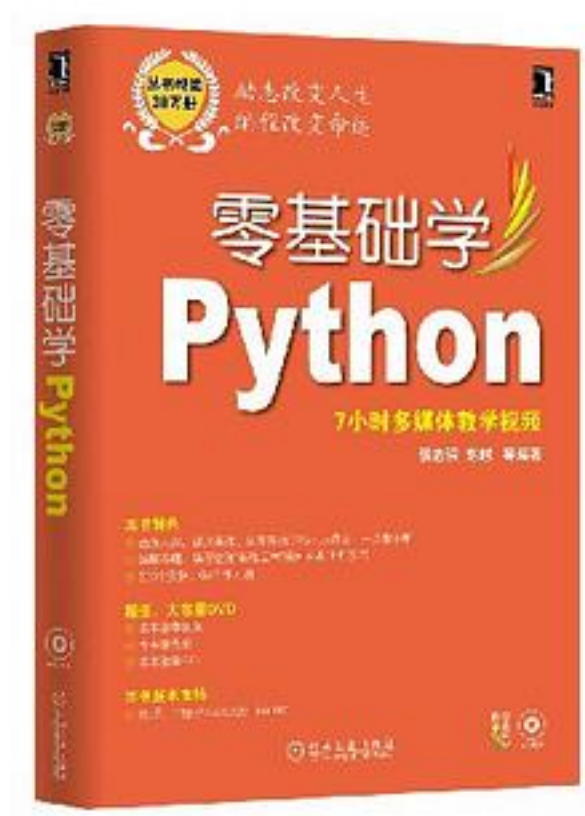


零基础学Python



[零基础学Python 下载链接1](#)

著者:张志强

出版者:机械工业出版社

出版时间:2015-2

装帧:平装

isbn:9787111492115

Python是目前最流行的动态脚本语言之一。本书由浅入深，全面、系统地介绍了使用Python进行开发的各种知识和技巧。

本书内容包括Python环境的安装和配置、Python的基本语法、模块和函数、内置数据结构、字符串和文件的处理、正则表达式的使用、异常的捕获和处理、面向对象的语言特性和设计、Python的数据库编程、Tkinter GUI库的使用、HTML应用、XML应用、Django网页开发框架的使用、测试驱动开发模

式应用、Python中的进程和线程、Python系统管理、网络编程、Python图像处理、Python语言的扩展和嵌入以及Windows下Python开发等。为了便于读者学习，本书每个章节中都提供了详尽的例子，结合实例讲解各个知识点。
本书适合Python爱好者、大中专院校的学生、社会培训班学生，以及系统管理员、界面开发人员、Web开发人员、网络编程人员等有关人员学习、使用。

作者介绍:

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的特色 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 5

1.5 Python的开发工具 5

1.5.1 PyCharm的使用 6

1.5.2 Eclipse IDE的介绍 6

1.5.3 EditPlus编辑器环境的配置 7

1.6 不同平台下的Python 9

1.7 小结 9

1.8 习题 9

第2章 Python必须知道的基础语法 10

2.1 Python的文件类型 10

2.1.1 源代码 10

2.1.2 字节代码 10

2.1.3 优化代码 11

2.2 Python的编码规范 11

2.2.1 命名规则 11

2.2.2 代码缩进与冒号 13

2.2.3 模块导入的规范 15

2.2.4 使用空行分隔代码 16

2.2.5 正确的注释 16

2.2.6 语句的分隔 19

2.3 变量和常量 20

2.3.1 变量的命名 20

2.3.2 变量的赋值 21

2.3.3 局部变量 22

2.3.4 全局变量 23

2.3.5 常量 25

2.4 数据类型 26

2.4.1 数字 26

2.4.2 字符串 27

2.5 运算符与表达式 29

2.5.1 算术运算符和算术表达式 29

2.5.2 关系运算符和关系表达式 30

2.5.3 逻辑运算符和逻辑表达式 31

2.6 小结 32

2.7 习题 32

第3章 Python的控制语句 33

- 3.1 结构化程序设计 33
- 3.2 条件判断语句 34
 - 3.2.1 if条件语句 34
 - 3.2.2 if...elif...else判断语句 35
 - 3.2.3 if语句也可以嵌套 36
 - 3.2.4 switch语句的替代方案 37
- 3.3 循环语句 40
 - 3.3.1 while循环 40
 - 3.3.2 for循环 41
 - 3.3.3 break和continue语句 42
- 3.4 结构化程序示例 44
- 3.5 小结 46
- 3.6 习题 46
- 第4章 Python数据结构 47
 - 4.1 元组结构 47
 - 4.1.1 元组的创建 47
 - 4.1.2 元组的访问 48
 - 4.1.3 元组的遍历 50
 - 4.2 列表结构 51
 - 4.2.1 列表的创建 51
 - 4.2.2 列表的使用 53
 - 4.2.3 列表的查找、排序、反转 54
 - 4.2.4 列表实现堆栈和队列 55
 - 4.3 字典结构 57
 - 4.3.1 字典的创建 57
 - 4.3.2 字典的访问 58
 - 4.3.3 字典的方法 60
 - 4.3.4 字典的排序、复制 63
 - 4.3.5 全局字典——sys.modules模块 64
 - 4.4 序列 65
 - 4.5 小结 67
 - 4.6 习题 67
- 第5章 模块与函数 68
 - 5.1 Python程序的结构 68
 - 5.2 模块 69
 - 5.2.1 模块的创建 69
 - 5.2.2 模块的导入 69
 - 5.2.3 模块的属性 71
 - 5.2.4 模块的内置函数 71
 - 5.2.5 自定义包 74
 - 5.3 函数 76
 - 5.3.1 函数的定义 76
 - 5.3.2 函数的参数 77
 - 5.3.3 函数的返回值 80
 - 5.3.4 函数的嵌套 82
 - 5.3.5 递归函数 84
 - 5.3.6 lambda函数 85
 - 5.3.7 Generator函数 86
 - 5.4 小结 88
 - 5.5 习题 88
- 第6章 字符串与正则表达式 89
 - 6.1 常见的字符串操作 89
 - 6.1.1 字符串的格式化 89
 - 6.1.2 字符串的转义符 91

- 6.1.3 字符串的合并 93
- 6.1.4 字符串的截取 94
- 6.1.5 字符串的比较 95
- 6.1.6 字符串的反转 96
- 6.1.7 字符串的查找和替换 98
- 6.1.8 字符串与日期的转换 99
- 6.2 正则表达式应用 101
 - 6.2.1 正则表达式简介 101
 - 6.2.2 使用re模块处理正则表达式 103
- 6.3 小结 108
- 6.4 习题 108
- 第7章 使用Python处理文件 109
 - 7.1 文件的常见操作 109
 - 7.1.1 文件的创建 109
 - 7.1.2 文件的读取 111
 - 7.1.3 文件的写入 113
 - 7.1.4 文件的删除 114
 - 7.1.5 文件的复制 115
 - 7.1.6 文件的重命名 116
 - 7.1.7 文件内容的搜索和替换 117
 - 7.1.8 文件的比较 118
 - 7.1.9 配置文件的访问 119
 - 7.2 目录的常见操作 121
 - 7.2.1 创建和删除目录 122
 - 7.2.2 目录的遍历 122
 - 7.3 文件和流 124
 - 7.3.1 Python的流对象 124
 - 7.3.2 模拟Java的输入、输出流 126
 - 7.4 文件处理示例——文件属性浏览程序 127
 - 7.5 小结 128
 - 7.6 习题 128
- 第8章 面向对象编程 129
 - 8.1 面向对象的概述 129
 - 8.2 类和对象 130
 - 8.2.1 类和对象的区别 130
 - 8.2.2 类的定义 131
 - 8.2.3 对象的创建 131
 - 8.3 属性和方法 132
 - 8.3.1 类的属性 132
 - 8.3.2 类的方法 134
 - 8.3.3 内部类的使用 136
 - 8.3.4 `__init__`方法 137
 - 8.3.5 `__del__`方法 138
 - 8.3.6 垃圾回收机制 139
 - 8.3.7 类的内置方法 140
 - 8.3.8 方法的动态特性 144
 - 8.4 继承 145
 - 8.4.1 使用继承 145
 - 8.4.2 抽象基类 147
 - 8.4.3 多态性 148
 - 8.4.4 多重继承 149
 - 8.4.5 Mixin机制 150
 - 8.5 运算符的重载 152
 - 8.6 Python与设计模式 154

- 8.6.1 设计模式简介 154
- 8.6.2 设计模式示例——Python实现工厂方法 155
- 8.7 小结 156
- 8.8 习题 156
- 第9章 异常处理与程序调试 157
- 9.1 异常的处理 157
 - 9.1.1 Python中的异常 157
 - 9.1.2 try...except的使用 158
 - 9.1.3 try...finally的使用 160
 - 9.1.4 使用raise抛出异常 161
 - 9.1.5 自定义异常 161
 - 9.1.6 assert语句的使用 162
 - 9.1.7 异常信息 163
- 9.2 使用自带IDLE调试程序 164
- 9.3 使用Easy Eclipse for Python调试程序 165
 - 9.3.1 新建工程 166
 - 9.3.2 配置调试 167
 - 9.3.3 设置断点 168
- 9.4 小结 170
- 9.5 习题 170
- 第10章 Python 数据库编程 171
- 10.1 Python环境下的数据库编程 171
 - 10.1.1 通过ODBC访问数据库 171
 - 10.1.2 使用DAO对象访问数据库 173
 - 10.1.3 使用ActiveX Data Object访问数据库 174
 - 10.1.4 Python连接数据库的专用模块 176
- 10.2 使用Python的持久化模块读写数据 179
- 10.3 嵌入式数据库SQLite 179
 - 10.3.1 SQLite的命令行工具 179
 - 10.3.2 使用sqlite3模块访问SQLite数据库 181
- 10.4 小结 182
- 10.5 习题 182
- 第二篇 Python的GUI程序设计
- 第11章 Python的GUI开发 183
- 11.1 Python的GUI开发选择 183
 - 11.1.1 认识Python内置的GUI库Tkinter 183
 - 11.1.2 使用Tkinter进行开发 184
 - 11.1.3 认识PyQT GUI库 186
 - 11.1.4 使用PyQT GUI库进行开发 187
- 11.2 小结 188
- 11.3 习题 188
- 第12章 GUI编程与Tkinter相关组件介绍 189
- 12.1 GUI程序开发简介 189
- 12.2 Tkinter与主要组件 190
 - 12.2.1 在程序中使用Tkinter 190
 - 12.2.2 顶层窗口 190
 - 12.2.3 标签 191
 - 12.2.4 框架 191
 - 12.2.5 按钮 192
 - 12.2.6 输入框 192
 - 12.2.7 单选按钮 193
 - 12.2.8 复选按钮 193
 - 12.2.9 消息 194
 - 12.2.10 滚动条 194

12.2.11 列表框	195
12.3 Tkinter所有组件简介	195
12.4 小结	196
12.5 习题	196
第三篇 Python的Web开发	
第13章 Python的HTML应用	197
13.1 HTML介绍	197
13.1.1 HTML的历史	197
13.1.2 SGML、HTML、XHTML、HTML5的关系	198
13.1.3 HTML的标签	198
13.1.4 HTML的框架组成	199
13.2 URL的处理	200
13.2.1 统一资源定位符URL	200
13.2.2 URL的解析	201
13.2.3 URL的拼合	203
13.2.4 URL的分解	204
13.2.5 URL的编解码	205
13.2.6 中文的编解码	207
13.2.7 查询参数的编码	208
13.3 CGI的使用	210
13.3.1 CGI介绍	210
13.3.2 获取CGI环境信息	211
13.3.3 解析用户的输入	214
13.4 获取HTML资源	216
13.4.1 使用urlopen和urlretrieve获取HTTP资源	217
13.4.2 分析返回资源的相关信息	221
13.4.3 自定义获取资源方式	223
13.4.4 使用http.client模块获取资源	226
13.5 HTML文档的解析	228
13.6 小结	230
13.7 习题	230
第14章 Python和XML	231
14.1 XML介绍	231
14.1.1 XML的演进历史	231
14.1.2 XML的优点和限制	232
14.1.3 XML技术的Python支持	233
14.2 XML文档概览和验证	234
14.2.1 XML文档的基础概念	234
14.2.2 XML文档的结构良好性验证	234
14.2.3 XML文档的有效性验证	237
14.3 分析XML文档结构	239
14.3.1 XML的元素和标签	239
14.3.2 元素的属性	242
14.3.3 XML的名字	244
14.3.4 字符实体	245
14.3.5 CDATA段	246
14.3.6 注释	248
14.3.7 处理指令	248
14.3.8 XML定义	249
14.4 使用SAX处理XML文档	249
14.4.1 SAX介绍	250
14.4.2 SAX处理的组成部分	250
14.5 使用DOM处理XML文档	255
14.5.1 DOM介绍	255

14.5.2 xml.dom模块中的接口操作	256
14.5.3 对XML文档的操作	264
14.6 小结	269
14.7 习题	270
第15章 Python的Web开发——Django框架的应用	271
15.1 常见的Web开发框架	271
15.1.1 Zope	271
15.1.2 TurboGears	273
15.1.3 Django	273
15.1.4 其他Web开发框架	274
15.1.5 根据自身所需选择合适的开发框架	275
15.2 MVC模式	275
15.2.1 MVC模式介绍	275
15.2.2 MVC模式的优缺点	276
15.2.3 Django框架中的MVC	277
15.3 Django开发环境的搭建	277
15.3.1 Django框架的安装	277
15.3.2 数据库的配置	278
15.4 Django框架的应用	279
15.4.1 Web应用的创建	279
15.4.2 Django中的开发服务器	280
15.4.3 创建数据库	282
15.4.4 生成Django应用	283
15.4.5 创建数据模型	284
15.4.6 URL设计	285
15.4.7 创建视图	286
15.4.8 模板系统	287
15.4.9 发布Django项目	289
15.5 Django框架的高级应用	289
15.5.1 管理界面	290
15.5.2 生成数据库数据	291
15.5.3 Session功能	292
15.5.4 国际化	295
15.6 小结	296
15.7 习题	296
第四篇 Python其他应用	
第16章 敏捷方法学在Python中的应用——测试驱动开发	297
16.1 测试驱动开发	297
16.1.1 测试驱动开发模式	297
16.1.2 TDD的优势	298
16.1.3 TDD的使用步骤	299
16.2 unittest测试框架	300
16.2.1 unittest模块介绍	300
16.2.2 构建测试用例	301
16.2.3 构建测试固件	302
16.2.4 组织多个测试用例	304
16.2.5 构建测试套件	305
16.2.6 重构代码	307
16.2.7 执行测试	308
16.3 使用doctest进行测试	311
16.3.1 doctest模块介绍	311
16.3.2 构建可执行文档	312
16.3.3 执行doctest测试	313
16.4 小结	315

- 16.5 习题 315
- 第17章 Python中的进程和线程 316
 - 17.1 进程和线程 316
 - 17.1.1 进程和线程的概念 316
 - 17.1.2 Python中对于进程和线程处理的支持 317
 - 17.2 Python下的进程编程 318
 - 17.2.1 进程运行环境 318
 - 17.2.2 创建进程 319
 - 17.2.3 终止进程 320
 - 17.3 使用subprocess模块管理进程 320
 - 17.3.1 使用Popen类管理进程 321
 - 17.3.2 调用外部系统命令 324
 - 17.3.3 替代其他进程创建函数 324
 - 17.4 进程间的信号机制 325
 - 17.4.1 信号的处理 325
 - 17.4.2 信号使用的规则 327
 - 17.5 多线程概述 328
 - 17.5.1 什么是多线程 328
 - 17.5.2 线程的状态 328
 - 17.5.3 Python中的线程支持 329
 - 17.6 生成和终止线程 329
 - 17.6.1 使用_thread模块 329
 - 17.6.2 使用threading.Thread类 332
 - 17.7 管理线程 334
 - 17.7.1 线程状态转移 334
 - 17.7.2 主线程对子线程的控制 334
 - 17.7.3 线程中的局部变量 335
 - 17.8 线程之间的同步 336
 - 17.8.1 临界资源和临界区 336
 - 17.8.2 锁机制 338
 - 17.8.3 条件变量 339
 - 17.8.4 信号量 342
 - 17.8.5 同步队列 342
 - 17.8.6 线程同步小结 344
 - 17.9 小结 344
 - 17.10 习题 344
- 第18章 基于Python的系统管理 345
 - 18.1 增强的交互式环境IPython 345
 - 18.1.1 IPython介绍 345
 - 18.1.2 IPython的安装 346
 - 18.1.3 IPython的启动 347
 - 18.1.4 IPython的环境配置 348
 - 18.2 和IPython的简单交互 349
 - 18.2.1 IPython中的输入和输出 349
 - 18.2.2 输出提示符的区别 349
 - 18.2.3 输出提示符区别的原因 350
 - 18.3 IPython中的magic函数 352
 - 18.3.1 magic函数的使用和构造 352
 - 18.3.2 目录管理 352
 - 18.3.3 对象信息的收集 356
 - 18.3.4 magic函数小结 359
 - 18.4 IPython适合于系统管理的特点 359
 - 18.4.1 Tab补全 359
 - 18.4.2 历史记录功能 361

18.4.3 执行外部系统命令和运行文件	363
18.4.4 对象查看和自省	367
18.4.5 直接编辑代码	370
18.4.6 设置别名和宏	371
18.5 使用Python进行文件管理	373
18.5.1 文件的比较	373
18.5.2 文件的归档	375
18.5.3 文件的压缩	377
18.6 使用Python定时执行任务	379
18.6.1 使用休眠功能	379
18.6.2 使用sched模块来定时执行任务	380
18.7 小结	380
18.8 习题	380
第19章 Python和网络编程	381
19.1 网络模型介绍	381
19.1.1 OSI简介	381
19.1.2 TCP/IP简介	382
19.2 Socket应用	383
19.2.1 Socket基础	383
19.2.2 Socket的工作方式	383
19.3 服务器端和客户端通信	384
19.3.1 服务器端的构建	384
19.3.2 客户端的构建	389
19.4 异步通信方式	391
19.4.1 使用Fork方式	391
19.4.2 使用线程方式	392
19.4.3 使用异步IO方式	393
19.4.4 使用asyncore模块	397
19.5 Twisted网络框架	399
19.5.1 Twisted框架介绍	400
19.5.2 Twisted框架下服务器端的实现	400
19.5.3 Twisted框架下服务器端的其他处理	401
19.6 小结	404
19.7 习题	404
第20章 常见的Python网络应用	405
20.1 使用FTP传输文件	405
20.1.1 FTP的工作原理和Python库支持	405
20.1.2 FTP的登录和退出	406
20.1.3 FTP的数据传输	407
20.2 使用POP3获取邮件	409
20.2.1 POP3协议介绍	409
20.2.2 poplib模块的使用	410
20.3 使用SMTP发送邮件	411
20.3.1 SMTP协议介绍	411
20.3.2 smtpplib模块的使用	411
20.4 使用Telnet远程登录	413
20.4.1 Telnet协议介绍和Python库支持	413
20.4.2 telnetlib模块的使用	413
20.5 使用SNMP管理网络	414
20.5.1 SNMP协议组成	415
20.5.2 PySNMP框架介绍及使用	415
20.6 网络分析	417
20.6.1 网络分析概述	417
20.6.2 使用Scapy在网络中抓包分析	418

- 20.7 小结 420
- 20.8 习题 420
- 第21章 图像处理 421
 - 21.1 图像处理相关概念 421
 - 21.1.1 Python下的图像处理包 421
 - 21.1.2 Pillow支持的图像文件格式 422
 - 21.1.3 图像处理中的其他概念 423
 - 21.2 基本的图像处理 424
 - 21.2.1 图像的读写操作 424
 - 21.2.2 获取图像信息 425
 - 21.2.3 图像文件格式的转换 427
 - 21.2.4 图像的裁剪和合成 428
 - 21.2.5 图像的变换 430
 - 21.3 图像处理的高级应用 432
 - 21.3.1 图像的通道操作 432
 - 21.3.2 对图像的增强 435
 - 21.3.3 Pillow中的内置滤镜 437
 - 21.4 小结 438
 - 21.5 习题 438
- 第22章 Python语言的扩展与嵌入 439
 - 22.1 Python语言的扩展 439
 - 22.1.1 Python扩展简介 439
 - 22.1.2 一个C扩展的例子 440
 - 22.1.3 模块方法表和初始化函数 441
 - 22.1.4 编译和测试 442
 - 22.2 Python语言的嵌入 445
 - 22.2.1 Python嵌入简介 445
 - 22.2.2 一个Python嵌入的例子 445
 - 22.2.3 更好的嵌入 446
 - 22.3 小结 449
 - 22.4 习题 449
- 第23章 Windows下的Python开发 450
 - 23.1 组件对象模型 450
 - 23.1.1 组件对象模型介绍 450
 - 23.1.2 COM结构 451
 - 23.1.3 COM对象的交互 451
 - 23.2 Python对COM技术的支持 452
 - 23.2.1 Python中的Windows扩展：PyWin32 452
 - 23.2.2 客户端COM组件 453
 - 23.2.3 实现COM组件 454
 - 23.3 Windows下的常见Python应用 457
 - 23.3.1 对Word的自动访问 457
 - 23.3.2 对Excel的自动访问 458
 - 23.3.3 对PowerPoint的自动访问 460
 - 23.3.4 对Outlook的自动访问 461
 - 23.4 小结 462
 - 23.5 习题 462
- • • • • (收起)

[零基础学Python_下载链接1](#)

标签

编程

python

Python

计算机

python书籍

程序设计

计算机科学

软件开发

评论

挺好一本书，看完learn python the hard way之后再把这本过一本基本就可以自己学着写项目了，就是封皮做成廉价励志书的款
。。。

根本不是零基础才能看懂的嘛。。

入门级阅读不错，开始入门Python

代码，代码说明和注意这三环节比较像正式的讲解教材…
类还是看得我一愣一愣的…后大半本都是蟒蛇应用。

偏网络应用方向，也不适合零基础

没读完，大致翻了下就转别人了，还不错，侧重应用。

碰到故人了。。但是这本书。有很大缺点。既然零基础，那么没讲过的知识点，就不该用来写代码，这也是很多书的通病，你这写过了，我不懂咋办。。。

适合入门

看完一套教学视频之后在多看上发现了这本书，出乎意料的好用呢。再看看说明书什么的吧，用了python总感觉自己已经可以写代码了呢！哈哈哈哈哈

全书没有一个整体架构讲解，零散的知识点罗列，类似指导手册，查询还可以。作为第一本书只会看得云里雾里。不适合零基础。或许看过其他书再看还有用，但这样就不用再看这本了。

大概是从入门到放弃吧。

基础部分讲得还不错！

历时半年多吧，用终于把它看完了！大部分内容还是讲的非常细致，一点基础的人也能看懂的程度，不过后边网络变成那里感觉有点不友好，基本看不太懂…

零基础的

作为初学者，才看了第一章就感觉知识点很杂乱，直接让人放弃，打算重新找一本入门。
。

看完一套教学视频之后在多看上发现了这本书，出乎意料的好用呢。再看看说明书什么的吧，用了python总感觉自己已经可以写代码了呢！哈哈哈哈哈 @2016-08-17 18:11:39

这本书并不是一门适合入门的图书，而且内容也有部分错误，不是很多，不过对于初学者来说已经是大问题了。还好这不是我的第一本入门书，不推荐购买。

目录乍一看感觉又全又有条理，但是知识点没有被归纳，甚至可以说是天花乱坠。东一枪西一枪的感觉。而每一个单个的知识点也缺乏轻重。
怎么说呢，就感觉像是一个包工头转了程序员，现在在带你入门一样。。。建议想学python的人，老老实实下O'Reilly系列吧

内容不全到和课后习题不匹配，不过确实写的挺简单的，是零基础

这本书好垃圾，大家别买！！买后悔了

[零基础学Python 下载链接1](#)

书评

[零基础学Python 下载链接1](#)