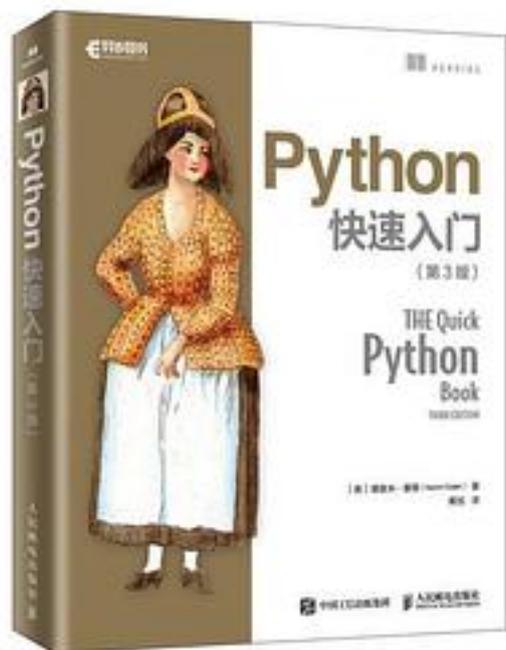


Python快速入门



[Python快速入门 下载链接1](#)

著者:[美] 娜奥米·塞德 (Naomi Ceder)

出版者:人民邮电出版社

出版时间:2019-8

装帧:平装

isbn:9787115509352

这是一本Python快速入门书，基于Python 3.6编写。本书分为4部分，第一部分讲解Python的基础知识，对Python进行概要的介绍；第二部分介绍Python编程的重点，涉及列表、元组、集合、字符串、字典、流程控制、函数、模块和作用域、文件系统、异常等内容；第三部分阐释Python的高级特性，涉及类和面向对象、正则表达式、数据类型即对象、包、Python库等内容；第四部分关注数据处理，涉及数据文件的处理、网络数据、数据的保存和数据探索，最后给出了相关的案例。

本书框架结构清晰，内容编排合理，讲解循序渐进，并结合大量示例和习题，让读者可

以快速学习和掌握Python，既适合Python初学者学习，也适合作为专业程序员的简明Python参考书。

作者介绍:

娜奥米·塞德 (Naomi Ceder) 是Python软件基金会主席。自2001年以来，她一直在学习、使用和教授Python。她的编程生涯已经持续了近30年，使用过多种编程语言，做过Linux系统管理员、编程教师、开发人员和系统架构师。她从2001年开始使用Python，从此向各个层次的用户讲授Python，从12岁的孩子到专业编程人员都有。她向所有人宣传Python，宣讲加入内容丰富的社区的好处。她现在领导着Dick Blick Art Materials的一支开发团队。

目录: 第一部分 开始篇

第1章 关于Python 3

1.1 用Python的理由 3

1.2 Python的长处 3

1.2.1 Python易于使用 4

1.2.2 Python富有表现力 4

1.2.3 Python可读性好 5

1.2.4 Python功能齐备 5

1.2.5 Python跨平台 6

1.2.6 Python免费 6

1.3 Python的短板 6

1.3.1 Python不是速度最快的语言 7

1.3.2 Python的库不算最多 7

1.3.3 Python在编译时不检查变量类型 7

1.3.4 Python对移动应用的支持不足 8

1.3.5 Python对多处理器的利用不充分 8

1.4 学Python 3的理由 8

1.5 小结 9

第2章 入门 10

2.1 Python的安装 10

2.2 基础交互模式和IDLE 12

2.2.1 基础交互模式 12

2.2.2 IDLE集成开发环境 13

2.2.3 基础交互模式和IDLE的适用场景 13

2.3 使用IDLE的Python shell窗口 13

2.4 第一个程序 “Hello, world” 14

2.5 利用交互式提示符探索Python 15

2.6 小结 16

第3章 Python概述 17

3.1 Python简介 17

3.2 内置数据类型 18

3.2.1 数值 18

3.2.2 列表 20

3.2.3 元组 21

3.2.4 字符串 22

3.2.5 字典 23

3.2.6 集合 23

3.2.7 文件对象 23

3.3 流程控制语句结构 24

3.3.1 布尔值和表达式 24

- 3.3.2 if-elif-else语句 25
- 3.3.3 while循环 25
- 3.3.4 for循环 25
- 3.3.5 函数定义 26
- 3.3.6 异常 27
- 3.3.7 用关键字with控制上下文 27
- 3.4 创建模块 28
- 3.5 面向对象编程 29
- 3.6 小结 31
- 第二部分 重点内容
- 第4章 基础知识 35
 - 4.1 缩进和代码块构建 35
 - 4.2 识别注释 36
 - 4.3 变量和赋值 37
 - 4.4 表达式 38
 - 4.5 字符串 39
 - 4.6 数值 40
 - 4.6.1 内置数值处理函数 41
 - 4.6.2 高级数值处理函数 41
 - 4.6.3 数值计算 41
 - 4.6.4 复数 41
 - 4.6.5 高级复数函数 42
 - 4.7 None值 43
 - 4.8 获取用户输入 43
 - 4.9 内置操作符 44
 - 4.10 基本的Python编码风格 44
 - 4.11 小结 45
- 第5章 列表、元组和集合 46
 - 5.1 列表类似于数组 46
 - 5.2 列表的索引机制 47
 - 5.3 修改列表 49
 - 5.4 对列表排序 51
 - 5.4.1 自定义排序 52
 - 5.4.2 sorted()函数 53
 - 5.5 其他常用的列表操作 53
 - 5.5.1 用in操作符判断列表成员 53
 - 5.5.2 用+操作符拼接列表 54
 - 5.5.3 用*操作符初始化列表 54
 - 5.5.4 用min和max方法求列表的最小值和最大值 54
 - 5.5.5 用index方法搜索列表 54
 - 5.5.6 用count方法对匹配项计数 55
 - 5.5.7 列表操作小结 55
 - 5.6 嵌套列表和深复制 56
 - 5.7 元组 58
 - 5.7.1 元组的基础知识 59
 - 5.7.2 单个元素的元组应加上逗号 60
 - 5.7.3 元组的打包和拆包 60
 - 5.7.4 列表和元组的相互转换 61
 - 5.8 集合 62
 - 5.8.1 集合的操作 62
 - 5.8.2 不可变集合 63
 - 5.9 小结 64
- 第6章 字符串 65
 - 6.1 将字符串理解为字符序列 65

- 6.2 基本的字符串操作 66
- 6.3 特殊字符和转义序列 66
 - 6.3.1 基本的转义序列 66
 - 6.3.2 数字格式（八进制、十六进制）和Unicode编码的转义序列 67
 - 6.3.3 对带特殊字符的字符串打印和求值的差异 68
- 6.4 字符串方法 68
 - 6.4.1 字符串的split和join方法 69
 - 6.4.2 将字符串转换为数值 70
 - 6.4.3 去除多余的空白符 71
 - 6.4.4 字符串搜索 72
 - 6.4.5 字符串修改 74
 - 6.4.6 利用列表修改字符串 75
 - 6.4.7 其他有用的字符串方法和常量 75
- 6.5 将对象转换为字符串 77
- 6.6 使用format方法 78
 - 6.6.1 format方法和位置参数 78
 - 6.6.2 format方法和命名参数 78
 - 6.6.3 格式描述符 79
- 6.7 用%格式化字符串 79
 - 6.7.1 使用格式化序列 80
 - 6.7.2 命名参数和格式化序列 81
- 6.8 字符串内插 82
- 6.9 bytes对象 82
- 6.10 小结 84
- 第7章 字典 85
 - 7.1 何为字典 85
 - 7.2 字典的其他操作 87
 - 7.3 单词计数 90
 - 7.4 可用作字典键的对象 90
 - 7.5 稀疏矩阵 91
 - 7.6 将字典用作缓存 92
 - 7.7 字典的效率 93
 - 7.8 小结 93
- 第8章 流程控制 94
 - 8.1 while循环 94
 - 8.2 if-elif-else语句 95
 - 8.3 for循环 96
 - 8.3.1 range函数 97
 - 8.3.2 用初值和步进值控制range函数 97
 - 8.3.3 在for循环中使用break和continue语句 98
 - 8.3.4 for循环和元组拆包 98
 - 8.3.5 enumerate函数 98
 - 8.3.6 zip函数 99
 - 8.4 列表和字典推导式 99
 - 8.5 语句、代码块和缩进 101
 - 8.6 布尔值和布尔表达式 103
 - 8.6.1 大多数Python对象都能用作布尔类型 104
 - 8.6.2 比较操作符和布尔操作符 104
 - 8.7 编写简单的文本文件分析程序 105
 - 8.8 小结 106
- 第9章 函数 107
 - 9.1 基本的函数定义 107
 - 9.2 多种函数参数 108

- 9.2.1 按位置给出形参 108
- 9.2.2 按形参名称传递实参 109
- 9.2.3 变长实参 110
- 9.2.4 多种参数传递方式的混用 111
- 9.3 将可变对象用作函数实参 111
- 9.4 局部变量、非局部变量和全局变量 113
- 9.5 将函数赋给变量 115
- 9.6 lambda表达式 115
- 9.7 生成器函数 116
- 9.8 装饰器 117
- 9.9 小结 118
- 第10章 模块和作用域规则 120
- 10.1 何为模块 120
- 10.2 编写第一个模块 121
- 10.3 import语句 123
- 10.4 模块搜索路径 124
- 10.5 模块内部私有名称 126
- 10.6 库和第三方模块 127
- 10.7 Python作用域规则和命名空间 127
- 10.8 小结 133
- 第11章 Python程序 134
- 11.1 创建一个很简单的程序 134
- 11.1.1 从命令行启动脚本 135
- 11.1.2 命令行参数 135
- 11.1.3 脚本输入/输出的重定向 136
- 11.1.4 argparse模块 137
- 11.1.5 fileinput模块的使用 138
- 11.2 让脚本在UNIX下直接运行 140
- 11.3 macOS系统中的脚本 140
- 11.4 Windows中多种脚本执行方式 140
- 11.4.1 从命令窗口或PowerShell中启动脚本 141
- 11.4.2 Windows中的其他运行方式 141
- 11.5 程序和模块 141
- 11.6 发布Python应用程序 146
- 11.6.1 wheel包 146
- 11.6.2 zipapp和pex 146
- 11.6.3 py2exe和py2app 147
- 11.6.4 用freeze创建可执行程序 147
- 11.7 小结 147
- 第12章 文件系统的使用 149
- 12.1 os、os.path和pathlib的对比 149
- 12.2 路径和路径名 150
- 12.2.1 绝对路径和相对路径 150
- 12.2.2 当前工作目录 151
- 12.2.3 用pathlib模块访问目录 152
- 12.2.4 路径名的处理 152
- 12.2.5 用pathlib处理路径名 154
- 12.2.6 常用变量和函数 155
- 12.3 获取文件信息 156
- 12.4 文件系统的其他操作 158
- 12.5 处理目录树下的所有文件 161
- 12.6 小结 162
- 第13章 文件的读写 165

- 13.1 打开文件及file对象 165
- 13.2 关闭文件 166
- 13.3 以写入等模式打开文件 166
- 13.4 读写文本及二进制数据的函数 167
- 13.5 用pathlib读写文件 169
- 13.6 屏幕输入/输出及重定向 169
- 13.7 用struct模块读取结构化的二进制数据 172
- 13.8 用pickle将对象存入文件 173
- 13.9 用shelve保存对象 176
- 13.10 小结 178
- 第14章 异常 179
 - 14.1 异常简介 179
 - 14.1.1 错误和异常处理的一般原则 179
 - 14.1.2 异常较为正式的定义 182
 - 14.1.3 多种异常的处理 182
 - 14.2 Python中的异常 182
 - 14.2.1 Python异常的类型 183
 - 14.2.2 引发异常 185
 - 14.2.3 捕获并处理异常 185
 - 14.2.4 自定义新的异常 187
 - 14.2.5 用assert语句调试程序 188
 - 14.2.6 异常的继承架构 188
 - 14.2.7 示例：用Python编写的磁盘写入程序 189
 - 14.2.8 示例：正常计算过程中的异常 189
 - 14.2.9 异常的适用场合 190
 - 14.3 用到with关键字的上下文管理器 191
 - 14.4 小结 192
- 第三部分 高级特性
- 第15章 类和面向对象编程 195
 - 15.1 定义类 195
 - 15.2 实例变量 196
 - 15.3 方法 197
 - 15.4 类变量 199
 - 15.5 静态方法和类方法 201
 - 15.5.1 静态方法 201
 - 15.5.2 类方法 202
 - 15.6 继承 203
 - 15.7 类及实例变量的继承 205
 - 15.8 概括：Python类的基础知识 206
 - 15.9 私有变量和私有方法 208
 - 15.10 用@property获得更为灵活的实例变量 209
 - 15.11 类实例的作用域规则和命名空间 210
 - 15.12 析构函数和内存管理 214
 - 15.13 多重继承 214
 - 15.14 小结 216
- 第16章 正则表达式 218
 - 16.1 何为正则表达式 218
 - 16.2 带特殊字符的正则表达式 219
 - 16.3 正则表达式和原始字符串 220
 - 16.4 从字符串提取出匹配文本 221
 - 16.5 用正则表达式替换文本 224
 - 16.6 小结 226
- 第17章 数据类型即对象 227
 - 17.1 类型即对象 227

- 17.2 类型的使用 228
- 17.3 类型和用户自定义类 228
- 17.4 鸭子类型 230
- 17.5 何为特殊方法属性 231
- 17.6 让对象像列表一样工作 232
- 17.7 特殊方法属性__getitem__ 232
 - 17.7.1 工作原理 233
 - 17.7.2 实现完整的列表功能 234
- 17.8 完整实现列表功能的对象 235
- 17.9 由内置类型派生子类 237
 - 17.9.1 由列表类型派生子类 237
 - 17.9.2 由UserList派生子类 238
- 17.10 特殊方法属性的适用场景 239
- 17.11 小结 239
- 第18章 包 240
 - 18.1 何为包 240
 - 18.2 包的第一个示例 240
 - 18.3 包的实际例子 242
 - 18.3.1 包内的__init__.py文件 243
 - 18.3.2 mathproj包的基本用法 243
 - 18.3.3 子包和子模块的加载 244
 - 18.3.4 包内的import语句 245
 - 18.4 __all__属性 246
 - 18.5 包的合理使用 246
 - 18.6 小结 247
- 第19章 Python库的使用 248
 - 19.1 "功能齐备"的标准库 248
 - 19.1.1 各种数据类型的管理 248
 - 19.1.2 文件和存储操作 250
 - 19.1.3 操作系统服务的访问 251
 - 19.1.4 互联网协议及其数据格式的使用 251
 - 19.1.5 开发调试工具及运行时服务 252
 - 19.2 标准库之外的库 253
 - 19.3 添加其他Python库 253
 - 19.4 通过pip和venv安装Python库 253
 - 19.4.1 带--user标志的安装 254
 - 19.4.2 虚拟环境 254
 - 19.5 PyPI (即“奶酪商店”) 255
 - 19.6 小结 255
- 第四部分 数据处理
- 第20章 简单的文件问题 259
 - 20.1 问题：没完没了的数据文件流 259
 - 20.2 场景：无穷无尽的产品源数据 259
 - 20.3 引入更多目录结构 261
 - 20.4 节省存储空间：压缩和整理 263
 - 20.4.1 文件压缩 263
 - 20.4.2 文件清理 264
 - 20.5 小结 265
- 第21章 数据文件的处理 266
 - 21.1 ETL简介 266
 - 21.2 文本文件的读取 267
 - 21.2.1 文本编码：ASCII、

- Unicode等 267
- 21.2.2 非结构化文本 268
- 21.2.3 带分隔符的普通文本文件 270
- 21.2.4 csv模块 271
- 21.2.5 读取csv文件并存储为字典的列表 274
- 21.3 Excel文件 275
- 21.4 数据清洗 276
 - 21.4.1 清洗 276
 - 21.4.2 排序 277
 - 21.4.3 数据清洗时的问题和陷阱 278
- 21.5 数据文件的写入 279
 - 21.5.1 CSV和其他带分隔符的文件 279
 - 21.5.2 Excel文件的写入 280
 - 21.5.3 数据文件打包 280
- 21.6 小结 281
- 第22章 网络数据 282
 - 22.1 获取文件 282
 - 22.1.1 用Python从FTP服务器获取文件 283
 - 22.1.2 通过SFTP协议获取文件 284
 - 22.1.3 通过HTTP/HTTPS协议获取文件 284
 - 22.2 通过API获取数据 285
 - 22.3 结构化数据格式 287
 - 22.3.1 JSON数据 287
 - 22.3.2 XML数据 291
 - 22.4 抓取Web数据 295
 - 22.5 小结 299
- 第23章 数据的保存 300
 - 23.1 关系数据库 300
 - 23.2 SQLite: sqlite3数据库的用法 301
 - 23.3 MySQL、PostgreSQL和其他关系数据库的使用 303
 - 23.4 利用ORM简化数据库操作 303
 - 23.4.1 SQLAlchemy 304
 - 23.4.2 用Alembic修改数据库结构 307
 - 23.5 NoSQL数据库 310
 - 23.6 用Redis实现键/值存储 310
 - 23.7 MongoDB中的文档 313
 - 23.8 小结 315
- 第24章 数据探索 316
 - 24.1 Python的数据探索工具 316
 - 24.1.1 Python用于数据探索的优势 316
 - 24.1.2 Python能比电子表格做得更好 317
 - 24.2 Jupyter记事本 317
 - 24.2.1 启动内核 318
 - 24.2.2 执行单元格中的代码 318
 - 24.3 Python和pandas 319
 - 24.3.1 为什么要选用pandas 319
 - 24.3.2 pandas的安装 320
 - 24.3.3 Data Frame 320
 - 24.4 数据清洗 322
 - 24.4.1 用pandas加载并保存数据 322
 - 24.4.2 用Data Frame进行数据清洗 324
 - 24.5 数据聚合和处理 326
 - 24.5.1 Data Frame的合并 326

24.5.2 数据选取	327
24.5.3 分组与聚合	328
24.6 数据绘图	329
24.7 不用pandas的理由	330
24.8 小结	331
第25章 案例研究	332
25.1 数据的下载	332
25.2 解析inventory数据	335
25.3 根据经纬度选择一个观测站	336
25.4 选择观测站并获取其元数据	338
25.5 获取并解析真实的天气数据	341
25.5.1 获取数据	341
25.5.2 解析天气数据	341
25.6 将天气数据存入数据库（可选）	344
25.7 选取数据并作图	345
25.8 用pandas对数据绘图	346
附录A Python文档使用指南	347
习题答案（网上下载）	
• • • • •	(收起)

[Python快速入门_下载链接1](#)

标签

Python

计算机技术

程序设计

软件开发

Programming

评论

Python软件基金会主席潜心力作，历经三版锤炼，Python入门、进阶全面覆盖，大量Python 3.6源代码示例，边做边学，完整项目案例，指导开发实践，配备精深习题及答案，方

便自学。

作者写得比较简洁。本书适合已经有编程经验的读者。后半部分中高级的内容虽然可以看懂，但是没有经过操作不容易记住。特别是面向对象的编程到处都是x.y.z，还会把一些复杂的对象命名为一个简单的变量名，这个变量名下又有很多方法或类调用其他的类，不熟悉的话不能很快理解。

非常不错，虽然不是项目式的教程，但是解答了我很多疑惑，这些疑惑有些关于计算机层面的，比如路径名那一章节；有些是语言本身，比如编写一个模块。建议挑一本项目式python书籍一起看。

这是一本2019年8月份新出版的Python学习入门书籍，内容比较全面，从数组、字典等基本数据类型，到类/面向对象编程、常见标准库的使用，到数据库、数据分析等应用实例，都覆盖到了。每个知识点基本都是先概要介绍，然后讲解一两个简短的例子来加深理解，行文比较清晰简洁。如果你觉得一开始没有信心直接尝试厚达1000页的《Python学习手册》，可以把这本书作为替代。我是花了几个月时间断断续续先读完了《Python学习手册》，然后读这本书几天就快速翻完了，作为Python知识点手册来复习回顾，感觉也挺不错。

这是一本将理论与实践结合得非常到位的图书，覆盖了从入门到进阶的读者群体。书中划分了四大板块，知识讲解循序渐进，不同层次的读者可以选择阅读。这本书提供的实战项目也是一大亮点，可以带读者体验现实的项目开发过程，快速上手实战。

[Python快速入门_下载链接1](#)

书评

因为大部分内容已经很熟悉了，我只是快速翻阅了一下。首先，作者语言功底不错，虽然是英文但读起来很爽，行文流畅，句子很容易读懂。同样是讲Python的，同样是英文书籍，Pro Python就读的让我吐血，不知道是不是两本书的定位不同，Pro Python的内容更加高阶一些。第二，内容编...

作者在对Python语言做介绍的同时，与大家分享了他自己的许多心得体会，使得读起来犹如在欣赏一篇篇有关Python方方面面的博文。
强烈推荐具有基本英语基础的程序员读读本书，里面很少有比较晦涩的单词和句子，作者用一些简单清晰的单词为我们讲解了Python3。能够坚持看完一本英...

作者文笔很好，由浅入中，避开复杂的又不常用的东西，页数也不多，让我饶有兴趣的迈入了Python大门~

[Python快速入门_下载链接1](#)