

高等院校信息技术规划教材

Python语言 程序设计（微课版）

曹洁 张志锋 孙玉胜 崔霄 等编著

清华大学出版社

高等院校信息技术规划教材

Python语言程序设计

(微课版)

曹洁 张志锋 孙玉胜 崔霄 等编著

清华大学出版社
北京

内 容 简 介

Python 是一门简单易学、功能强大的优雅编程语言。它内建了高效的数据结构,丰富的第三方开发库,能够用简单而高效的方式编程。本书由浅入深、步步引导、循序渐进地讲述 Python 语言的基础知识、基本语法。本书用 12 章的篇幅来介绍 Python 语言,包括 Python 语言基础、选择结构程序设计、循环结构程序设计、函数、正则表达式、文件与文件夹操作、面向对象程序设计、模块和包、算法与数据结构基础、错误和异常处理、图形用户界面、用 matplotlib 实现数据可视化。

本书可作为高等院校各专业的 Python 语言教材,也可作为软件开发人员的参考资料,还可作为 Python 语言自学者的参考书。

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 语言程序设计:微课版/曹洁等编著. —北京:清华大学出版社,2019
(高等院校信息技术规划教材)
ISBN 978-7-302-53086-2

I. ①P… II. ①曹… III. ①软件工具—程序设计—高等学校—教材 IV. ①TP311.561

中国版本图书馆 CIP 数据核字(2019)第 114772 号

责任编辑:白立军

封面设计:常雪影

责任校对:焦丽丽

责任印制:丛怀宇

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

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

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

印 装 者:清华大学印刷厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:22.5

字 数:514 千字

版 次:2019 年 8 月第 1 版

印 次:2019 年 8 月第 1 次印刷

定 价:49.80 元

产品编号:081915-01

在 IEEE 发布的 2018 年编程语言排行榜中,Python 排名第一。Python 的语法非常接近英语,被称为最优雅的编程语言之一。阅读 Python 代码就像阅读一篇优美的文章。Python 语法简洁清晰,代码可读性强,编程模式非常符合人的思维方式,易学易用。对于同样的功能,用 Python 写的代码更短、更简洁。Python 拥有很多面向不同应用的开源扩展库,你能想到的功能基本上都已经有人替你开发了,你只需把想要的程序代码拿来组装便可构建个性化的应用。Python 支持命令式编程、函数式编程,支持面向对象程序设计。Python 是一门很受青睐的编程语言,被广泛用于数据分析、Web 开发、科学计算、人工智能、云计算、系统运维、数据可视化和图形开发等领域。

1. 本书编写特色

- (1) 本书全面涵盖 Python3 基础编程知识,基于 Python 3.6.2 构建 Python 开发平台。
- (2) 针对零基础读者,快速掌握 Python 语言开发。
- (3) 通过大量的实例,由浅入深、步步引导、循序渐进地讲述 Python 语言的基础知识和基本语法。
- (4) 注释详尽的代码示例。
- (5) 详尽的归纳与总结,帮读者集中深入掌握知识要点。
- (6) 丰富的数据可视化案例,助读者迅速掌握数据可视化技术。

2. 本书内容组织

第 1 章 Python 语言基础。介绍 Python 语言的特点,Python 开发环境的安装方法,编写 Python 代码的方式;重点介绍 Python 的基本数据类型:数字、字符串、列表、元组、字典、集合,针对每种类型详细介绍其操作命令,并给出相应的实例;之后,介绍人机交互的输入和输出,给出 Python 的多样化格式输出;然后,简单介绍 Python 如何读写文件;最后,介绍 Python 库的导入以及 Python 扩展库的安装。

第2章 选择结构程序设计。讲解布尔表达式、关系运算符和逻辑运算符,选择结构中的单向 if 语句、双向 if-else 语句、嵌套 if 和多向 if-elif-else 语句及条件表达式。

第3章 循环结构程序设计。讲解 while 循环及循环控制策略,for 循环、for 循环与 range() 函数的结合使用,break、continue 和 else 控制循环的方式。

第4章 函数。讲解怎样定义函数、函数的调用方式、参数传递、函数参数的类型、函数模块化、lambda 表达式、变量的作用域、函数的递归调用和常用内置函数。

第5章 正则表达式。讲解正则表达式的构成,正则表达式的边界匹配,正则表达式的分组、选择和引用匹配,正则表达式的贪婪匹配与懒惰匹配,正则表达式模块 re,正则表达式对象以及 Match 对象。

第6章 文件与文件夹操作。讲解文本文件的打开、读写以及文件指针的定位,二进制文件的打开与读写,os、os.path、shutil 对文件与文件夹的操作, csv 文件的读取和写入。

第7章 面向对象程序设计。讲解类的定义与使用,类的对象属性、类属性、私有属性、公有属性以及@property 装饰器,类的对象方法、类方法以及类的静态方法,类的单继承、多重继承、类成员的继承和重写,查看继承的层次关系,所有类的基类 object,对象的引用、对象的浅复制和对象的深复制。

第8章 模块和包。讲解模块的创建、模块的导入和使用、模块的主要属性,导入模块时搜索目录的顺序,使用 sys.path.append() 临时增添系统目录,使用.pth 文件永久添加系统目录,使用 PYTHONPATH 环境变量永久添加系统目录,包的创建、包的导入与使用。

第9章 算法与数据结构基础。讲解顺序查找、二分查找、插值查找算法,冒泡排序、选择排序、插入排序、归并排序、快速排序算法,自定义矩阵、栈、队列和二叉树类型。

第10章 错误和异常处理。讲解编写 Python 程序常犯的错误,异常类型、异常处理、主动抛出异常以及自定义异常类,断言定义及使用方法,启用/禁用断言,断言使用场景,使用 print 调试程序、IDLE 调试程序以及使用 pdb 调试程序。

第11章 图形用户界面。讲解使用 tkinter 制作图形用户界面,tkinter 主要的构件类,pack 布局管理器、grid 布局管理器、place 布局管理器。

第12章 用 matplotlib 实现数据可视化。讲解 matplotlib 架构的后端层、表现层和脚本层,使用 matplotlib 的 pyplot 子库绘制线形图、直方图、条形图、饼图和散点图。

3. 本书适用读者

- (1) 学习 Python 语言程序设计课程的本科、专科或研究生。
- (2) 编程爱好者。
- (3) 其他对 Python 感兴趣的人员。

本书由曹洁、张志锋、孙玉胜、崔霄、王博、范乃梅和周开来编写。

在本书的编写和出版过程中得到了郑州轻工业大学、清华大学出版社的大力支持和帮助,在此表示感谢。

在本书的撰写过程中,参考了大量专业书籍和网络资料,在此向相关作者表示感谢。

由于编写时间仓促,编者水平有限,书中难免会有缺点和不足,热切期望得到专家和读者的批评指正。

除了配套制作的教学课件、教学日历、教学大纲外,本书还提供书中示例的源代码和各章部分内容的视频讲解(可从清华大学出版社网站 www.tup.com.cn 下载),以提供更多更便捷的教学资源服务。

编 者

于郑州轻工业大学数据融合与知识工程实验室

2019年3月

第 1 章 Python 语言基础	1
1.1 Python 语言的特点	1
1.2 Python 应用领域	2
1.3 Python 解释器	3
1.4 Python 开发环境的安装	3
1.5 编写 Python 代码	6
1.5.1 用文本编辑器编写代码	6
1.5.2 用命令行格式的 Python Shell 编写代码	9
1.5.3 用带图形界面的 Python Shell 编写交互式 代码	10
1.5.4 用带图形界面的 Python Shell 编写程序代码	11
1.6 Python 中的注释	12
1.6.1 Python 中的单行注释	12
1.6.2 Python 中的多行注释	12
1.7 Python 中的对象	12
1.7.1 对象的身份	13
1.7.2 对象的类型	13
1.7.3 对象的值	13
1.7.4 对象的引用	13
1.7.5 对象的共享引用	14
1.7.6 对象是否相等的判断	14
1.8 Python 中的变量	15
1.9 Python 中的基本数据类型	15
1.9.1 number	16
1.9.2 string	17
1.9.3 list	31
1.9.4 tuple	39

1.9.5	dictionary	41
1.9.6	set	46
1.9.7	Python 数据类型之间的转换	49
1.10	Python 中的运算符	51
1.11	Python 中的数据输入	56
1.12	Python 中的数据输出	57
1.12.1	表达式语句输出	57
1.12.2	print() 函数输出	58
1.12.3	字符串对象的 format 方法的格式化输出	60
1.13	Python 中的文件的基本操作	63
1.14	Python 库的导入与扩展库的安装	64
1.14.1	库的导入	64
1.14.2	扩展库的安装	65
	习题	66
第 2 章	选择结构程序设计	68
2.1	布尔表达式	68
2.2	选择结构	69
2.2.1	单向 if 语句	69
2.2.2	双向 if-else 语句	70
2.2.3	嵌套 if 和多向 if-elif-else 语句	71
2.3	条件表达式	73
2.4	选择结构程序举例	74
	习题	77
第 3 章	循环结构程序设计	78
3.1	while 循环	78
3.2	循环控制策略	83
3.2.1	交互式循环	83
3.2.2	哨兵式循环	84
3.2.3	文件式循环	85
3.3	for 循环	86
3.3.1	for 循环的基本用法	86
3.3.2	for 循环与 range() 函数的结合使用	89
3.4	循环中的 break、continue 和 else	92
3.4.1	用 break 语句提前终止循环	92
3.4.2	用 continue 语句提前结束本次循环	93

3.4.3 循环语句的 else 子句	93
3.5 循环结构程序举例	96
习题	100
第 4 章 函数	101
4.1 为什么要用函数	101
4.2 怎样定义函数	103
4.3 函数调用	104
4.3.1 带有返回值的函数调用	104
4.3.2 不带返回值的函数调用	107
4.4 函数参数传递	107
4.5 函数参数的类型	108
4.5.1 位置参数	108
4.5.2 关键字参数	108
4.5.3 默认值参数	109
4.5.4 可变长度参数	109
4.5.5 序列解包参数	110
4.6 函数模块化	112
4.7 lambda 表达式	113
4.7.1 lambda 和 def 的区别	114
4.7.2 自由变量对 lambda 表达式的影响	116
4.8 变量的作用域	117
4.8.1 变量的局部作用域	117
4.8.2 变量的全局作用域	118
4.8.3 变量的嵌套作用域	119
4.9 函数的递归调用	120
4.10 常用内置函数	124
4.10.1 map() 函数	124
4.10.2 reduce() 函数	125
4.10.3 filter() 函数	127
4.11 函数举例	127
习题	133
第 5 章 正则表达式	134
5.1 什么是正则表达式	134
5.2 正则表达式的构成	134
5.3 正则表达式的模式匹配	137

5.3.1	正则表达式的边界匹配	137
5.3.2	正则表达式的分组、选择和引用匹配	138
5.3.3	正则表达式的贪婪匹配与懒惰匹配	141
5.4	正则表达式模块 re	142
5.5	正则表达式对象	145
5.6	Match 对象	148
5.7	正则表达式举例	150
	习题	152
第 6 章	文件与文件夹操作	154
6.1	文本文件	154
6.1.1	文本文件的字符编码	154
6.1.2	文本文件的打开	156
6.1.3	文本文件的写入	159
6.1.4	文本文件的读取	160
6.1.5	文本文件指针的定位	162
6.2	二进制文件	163
6.2.1	二进制文件的写入	163
6.2.2	二进制文件的读取	164
6.2.3	字节数据类型的转换	164
6.3	文件与文件夹操作	166
6.3.1	使用 os 操作文件与文件夹	166
6.3.2	使用 os.path 操作文件与文件夹	168
6.3.3	使用 shutil 操作文件与文件夹	170
6.4	csv 文件的读取和写入	172
6.4.1	使用 csv.reader() 读取 csv 文件	172
6.4.2	使用 csv.writer() 写入 csv 文件	173
6.4.3	使用 csv.DictReader() 读取 csv 文件	175
6.4.4	使用 csv.DictWriter() 写入 csv 文件	176
6.4.5	csv 文件的格式化参数	177
6.4.6	自定义 dialect	179
6.5	文件与文件操作举例	180
	习题	181
第 7 章	面向对象程序设计	182
7.1	定义类	183
7.2	创建类的对象	184

7.3 类中的属性	185
7.3.1 类的对象属性	185
7.3.2 类属性	186
7.3.3 私有属性和公有属性	188
7.3.4 @property 装饰器	189
7.4 类中的方法	192
7.4.1 类的对象方法	192
7.4.2 类方法	194
7.4.3 类的静态方法	195
7.5 类的继承	196
7.5.1 单继承	196
7.5.2 类的多重继承	200
7.5.3 类成员的继承和重写	203
7.5.4 查看继承的层次关系	203
7.6 object 类	204
7.7 对象的引用、浅复制和深复制	205
7.7.1 对象的引用	205
7.7.2 对象的浅复制	208
7.7.3 对象的深复制	209
7.8 面向对象程序举例	210
习题	213
第 8 章 模块和包	214
8.1 模块	214
8.1.1 模块的创建	214
8.1.2 模块的导入和使用	215
8.1.3 模块的主要属性	216
8.2 导入模块时搜索目录的顺序与系统目录的添加	219
8.2.1 导入模块时搜索目录的顺序	219
8.2.2 使用 sys.path.append() 临时增添系统目录	220
8.2.3 使用.pth 文件永久添加系统目录	220
8.2.4 使用 PYTHONPATH 环境变量永久添加系统目录	221
8.3 包	221
8.3.1 包的创建	221
8.3.2 包的导入与使用	222
习题	223

第 9 章 算法与数据结构基础	224
9.1 算法概述	224
9.2 查找算法	225
9.2.1 顺序查找	225
9.2.2 二分查找	226
9.2.3 插值查找	228
9.3 排序算法	229
9.3.1 冒泡排序	229
9.3.2 选择排序	230
9.3.3 插入排序	231
9.3.4 归并排序	232
9.3.5 快速排序	233
9.4 常用数据结构	235
9.4.1 自定义矩阵	235
9.4.2 自定义栈	240
9.4.3 自定义队列	242
9.4.4 自定义二叉树	245
习题	252
第 10 章 错误和异常处理	253
10.1 程序的错误	253
10.1.1 常犯的 9 个错误	253
10.1.2 常见的错误类型	256
10.2 异常处理概述	259
10.2.1 异常概述	259
10.2.2 异常类型	259
10.2.3 异常处理	260
10.2.4 主动抛出异常	265
10.2.5 自定义异常类	267
10.3 断言处理	269
10.3.1 断言处理概述	269
10.3.2 启用/禁用断言	270
10.3.3 断言使用场景	270
10.4 程序的调试方法	272
10.4.1 使用 print 调试	272
10.4.2 使用 IDLE 调试	272

10.4.3 使用 pdb 调试	276
习题	283
第 11 章 图形用户界面	284
11.1 图形用户界面库	284
11.2 tkinter 模块	285
11.2.1 tkinter 概述	285
11.2.2 tkinter 图形用户界面的构成	286
11.3 tkinter 主要的构件类	287
11.3.1 Label 类	287
11.3.2 Button 类	291
11.3.3 Radiobutton 类	295
11.3.4 Checkbutton 类	296
11.3.5 Listbox 类	298
11.3.6 Menu 类	301
11.3.7 Message 类	305
11.3.8 Entry 类	307
11.3.9 Text 类	308
11.3.10 Frame 类	312
11.4 几何布局管理器	312
11.4.1 pack 布局管理器	313
11.4.2 grid 布局管理器	314
11.4.3 place 布局管理器	316
习题	317
第 12 章 用 matplotlib 实现数据可视化	318
12.1 matplotlib 架构	318
12.1.1 后端层	319
12.1.2 表现层	319
12.1.3 脚本层	321
12.2 matplotlib 的 pyplot 子库	322
12.2.1 绘制线形图	324
12.2.2 绘制直方图	331
12.2.3 绘制条形图	333
12.2.4 绘制饼图	336
12.2.5 绘制散点图	339
习题	341
参考文献	342

Python 语言基础

本章主要介绍 Python 的基础知识,为后续章节学习相关内容做铺垫。

1.1 Python 语言的特点

Python 是从 ABC 语言发展而来的,是一种解释型、面向对象、动态数据类型的高级程序设计语言,具有丰富而强大的库。Python 常被称为胶水语言,能够把用其他语言制作的各种模块(尤其是 C/C++)很轻松地连接在一起。Python 的语法简洁清晰,强制用空白符(white space)作为语句缩进。Python 目前存在两个版本:Python2 和 Python3。Python3 是比较新的版本,但是它不向后兼容 Python2。本书讲述如何使用 Python3 来进行程序设计。

Python 语言的特点如下。

(1) 简单。阅读一个良好的 Python 程序,感觉就像是在读英语一样,Python 的这种伪代码本质上能够使人们专注于解决问题而不是去搞明白语言本身。

(2) 开源。Python 是 FLOSS(Free/Libre and Open Source Software,自由/开放源代码软件)之一。每一个模块和库都是开源的,它们的代码可以从网上找到。每个月,庞大的开发者社区都会为 Python 带来很多改进。

(3) 解释性。Python 可以直接从源代码运行。在计算机内部,Python 解释器把源代码转换为字节码的中间形式,然后再把它翻译成计算机使用的机器语言并运行。

(4) 面向对象。Python 既支持面向过程的编程也支持面向对象的编程,Python 中的数据都是由类所创建的对象。在面向过程的语言中,程序是由过程或仅仅是可重用代码的函数构建起来的。在面向对象的语言中,程序是由数据和功能组合而成的对象构建起来的。

(5) 可移植性。Python 具有很高的可移植性。用解释器作为接口读取和运行代码的最大优势就是可移植性。事实上,任何现有操作系统(Linux、Windows 和 Mac OS)安装相应版本的解释器后,Python 代码无须修改就能在其上运行。

(6) 可扩展性。部分程序可以使用其他语言编写,如 C/C++,然后在 Python 程序中使用它们。

(7) 可嵌入性。可以把 Python 程序嵌入到 C/C++ 程序中,从而提供脚本功能。

(8) 丰富的库。Python 标准库很庞大,可用来处理正则表达式、文档生成、单元测试、线程、数据库、网页浏览器、CGI、FTP、电子邮件、XML、XML-RPC、HTML、WAV 文件、密码系统、GUI(图形用户界面)、Tk 和其他与系统有关的操作。

1.2 Python 应用领域

Python 被广泛应用于众多领域。

1. Web 开发

Python 拥有很多免费数据函数库、免费 Web 网页模板系统以及与 Web 服务器进行交互的库,可以实现 Web 开发,搭建 Web 框架,目前比较有名的 Python Web 框架为 Django。

2. 爬虫开发

在爬虫领域,Python 几乎处于霸主地位,将网络一切数据作为资源,通过自动化程序进行有针对性的数据采集以及处理。

3. 云计算开发

Python 是从事云计算工作需要掌握的一门编程语言,目前很火的云计算框架 OpenStack 就是用 Python 开发的。

4. 人工智能

Google 公司早期大量使用 Python,为 Python 积累了丰富的科学运算库,当 AI 时代来临后,目前市面上大部分的人工智能的代码都是使用 Python 来编写,尤其是 PyTorch 之后,基本确定 Python 作为 AI 时代首选语言。

5. 自动化运维

Python 是一门综合性的语言,能满足绝大部分自动化运维需求。

6. 数据分析

Python 已成为数据分析和数据科学事实上的标准语言和标准平台之一,NumPy、Pandas、SciPy 和 matplotlib 程序库共同构成了 Python 数据分析的基础。

7. 科学计算

随着 NumPy、SciPy、Matplotlib、Enthought Librarys 等众多程序库的开发,使得 Python 越来越适合做科学计算、绘制高质量的 2D 和 3D 图像。

1.3 Python 解释器

每次运行 Python 命令,Python 解释器都会启动,读取和解释输入到提示符>>>后面的代码。解释器既可以处理单条指令,也可以处理整个 Python 代码文件,两种情况下解释器的处理机制都是相同的。

每次按下 Enter 键后,解释器就开始以单词为单位逐一扫码,将这些单词看作一个个文本片段,把它们组织成为表示程序逻辑结构的树状结构,随后这些代码片段将会被转化为字节码(.pyc 或 .pyo)。生成的字节码随后将交由 Python 虚拟机执行。

Python 的标准解释器称作 CPython,其完全是用 C 语言编写的。此外,还有一些用其他语言编写的解释器,例如用 Java 开发的 Jython、用 C# 开发的 IronPython 以及全部用 Python 开发的 PyPy。

1.4 Python 开发环境的安装

打开 Python 官网,选中 Downloads 下拉菜单中的 Windows,如图 1-1 所示,单击 Windows 打开 Python 软件下载页面如图 1-2 所示,根据自己的操作系统选择 32 位还是 64 位以及相应的版本号,下载 exe 后缀的可执行文件。



图 1-1 Windows 版本的 Python 下载

32 位和 64 位的版本安装起来没有区别,双击打开后,第一步要记得勾选 Add Python 3.6 to PATH 选项,意思是把 Python 的安装路径添加到系统环境变量的 Path 变量中。安装时不要选择默认,Python 安装界面如图 1-3 所示。

单击 Customize installation(自定义安装),进入下一个安装界面,在该界面所有选项全选,所有选项全选界面如图 1-4 所示。



图 1-2 Python 软件下载页面



图 1-3 Python 安装界面



图 1-4 所有选项全选界面