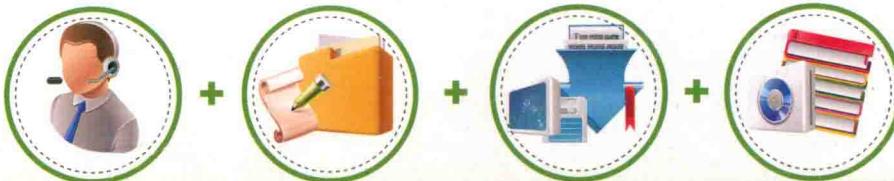


Python

数据分析基础

余本国 编著



- ◆ 由浅入深，配有大量示例代码，有利于尽快进入角色
- ◆ 理论与实际应用紧密结合，通过案例讲解，突出实用性
- ◆ 从实例出发，结合课后练习，让读者少走弯路
- ◆ 配备免费教学资源——电子课件、习题答案



全国高等院校应用型创新规划教材 · 计算机系列

Python 数据分析基础

余本国 编 著

清华大学出版社
北京

内 容 简 介

Python 是由 Guido van Rossum 于 20 世纪 80 年代末和 90 年代初, 在荷兰国家数学和计算机科学研究所设计出来的。它是一种面向对象的、用途非常广泛的编程语言, 具有非常清晰的语法特点, 适用于多种操作系统。目前 Python 在国际上非常流行, 正在得到越来越多的应用。

Python 可以完成许多任务, 功能非常强大, 其利用 Pandas 处理大数据的过程, 由于 Pandas 库的使用能够很好地展现数据结构, 成为近来 Python 项目中经常使用的热门技术, 并且 R 和 Spark 对 Python 都有很好的调用接口, 甚至在内存使用方面都有优化。

本书根据作者多年教学经验编写, 条理清楚, 内容深浅适中, 尽量让读者从实例出发, 结合课后练习, 少走弯路。本书涉及的内容主要包括 Python 数据类型与运算、流程控制及函数与类、Pandas 库的数据处理与分析等。在本书的最后, 还附带了一些文件读写、网络爬虫、矩阵计算等最基本的内容。

本书可以作为本科生、研究生以及科研人员学习 Python 的基础教材。

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 数据分析基础/余本国编著. —北京: 清华大学出版社, 2017
(全国高等院校应用型创新规划教材·计算机系列)

ISBN 978-7-302-47890-4

I. ①P… II. ①余… 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(2017)第 184452 号

责任编辑: 秦 甲

封面设计: 杨玉兰

责任校对: 宋延清

责任印制: 沈 露

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>, 010-62791865

印 装 者: 三河市金元印装有限公司

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 14.5 字 数: 279 千字

版 次: 2017 年 8 月第 1 版 印 次: 2017 年 8 月第 1 次印刷

印 数: 1~2500

定 价: 39.00 元

产品编号: 071752-01

前　　言

在写作本书的时候，国内大多数参考书还是 Python 2.7 版本，为了给在校大学生开设这门 Python 课程，我们选择了 Python 3.x，毕竟 Python 3.x 才是未来。与其让学生们从 Python 2.7 开始学，还不如直接从 Python 3.x 上手，以掌握更加完善的知识。

作者通过近三轮的教学，对 Python 3.x 的基础知识进行了筛选和总结，特编写此书，希望能够给准备使用 Python 的读者提供一些方便。

本书由浅入深，比较适合那些从未接触过计算机语言的读者。每章配有大量的示例代码，希望读者在使用本书的时候，能够尽可能自己敲代码，少用复制粘贴的方法，这样有利于读者尽快进入“角色”，毕竟“拷贝得来终觉浅”。

本书的前 3 章是 Python 的基础知识；第 4 章是利用 Pandas 库对数据进行处理、分析以及实现数据可视化；在第 5 章还列出了 Python 对文件的读取、存储方法，对网络爬虫、矩阵运算也做了简单的介绍。

作者在编写本书的过程中，得到了 Python 工程师齐伟的帮助。在开设这门课的时候，齐伟通过视频的形式与我们一起分享了 Python 开发经验。本书在完稿时，得到了研究生闫青、陈文华、马秀、樊宇凯和卢超在文字校对上的帮助。

最后感谢广大读者选择了本书。作者 E-mail: yubg@nuc.edu.cn, QQ: 120487362。欢迎各位读者批评指正。

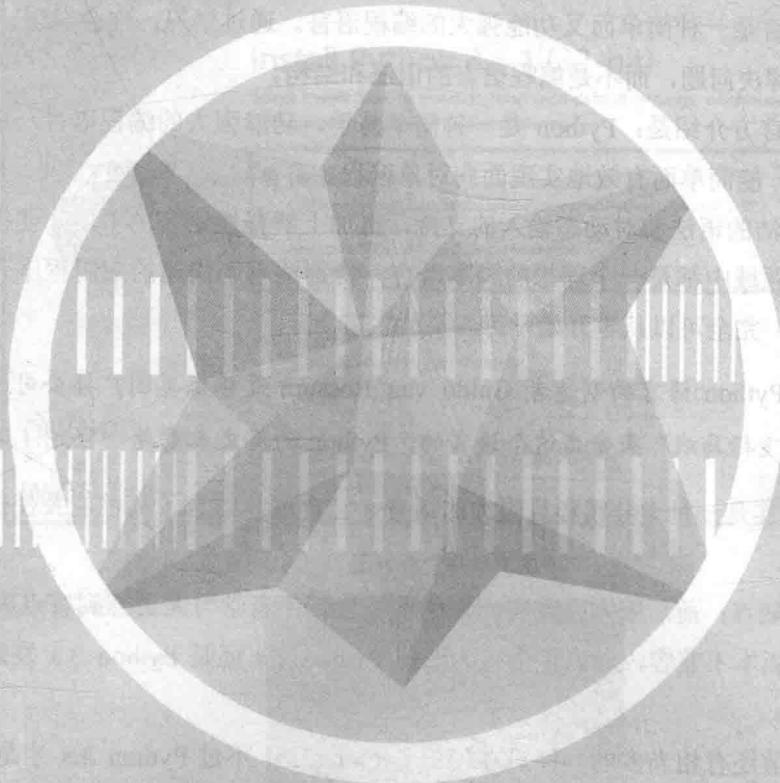
编　者

目录

第1章 Python简介	1	3.2.2 列表与元组的遍历	59
1.1 安装 Python	2	3.3 函数	61
1.2 Python 2 和 Python 3 的区别	5	3.3.1 函数的定义	61
本章小结	8	3.3.2 函数的使用	62
练习	8	3.3.3 形参和实参	63
第2章 Python数据类型与运算	9	3.3.4 参数的传递和改变	63
2.1 数据类型	11	3.3.5 变量的作用域	66
2.2 运算符与功能命令	12	3.3.6 函数参数的类型	68
2.2.1 算数运算符	12	3.3.7 任意个数的参数	70
2.2.2 比较运算符	12	3.3.8 函数调用	71
2.2.3 赋值运算符	13	3.4 函数式编程	74
2.2.4 常量与变量	15	3.4.1 lambda	74
2.2.5 字符串	16	3.4.2 reduce()	75
2.2.6 字符串索引与切片	18	3.4.3 filter()	76
2.2.7 输入和输出	20	3.4.4 map()	77
2.2.8 原始字符串	21	3.4.5 行函数	77
2.2.9 range	22	3.5 常用的内置函数	78
2.2.10 元组、列表、字典、集合	22	3.5.1 sum	78
2.2.11 格式化输出	37	3.5.2 zip	79
2.2.12 strip()、split()	40	3.5.3 enumerate	80
2.2.13 divmod()	42	3.5.4 max 和 min	81
2.2.14 join()	42	3.5.5 eval	81
本章小结	43	3.5.6 判断函数	83
练习	47	3.6 常见的错误显示	86
第3章 流程控制及函数与类	49	3.6.1 常见的错误类型	87
3.1 流程控制	52	3.6.2 初学者常犯的错误	89
3.1.1 if-else	52	3.6.3 try	93
3.1.2 for 循环	53	3.6.4 assert	95
3.1.3 while 循环	54	3.6.5 raise	95
3.1.4 continue 和 break	54	3.7 模块和包	96
3.2 遍历	56	3.7.1 模块(module)	96
3.2.1 range()函数	56	3.7.2 包(package)	100
		3.7.3 datetime 和 calendar 模块	101
		3.7.4 urllib 模块	105

目录

3.8 类.....	106
本章小结	109
练习	109
第 4 章 Python 数据分析实战	113
4.1 关于 Pandas	114
4.1.1 什么是 Pandas	114
4.1.2 Pandas 中的数据结构	114
4.1.3 Pandas 的安装方法	114
4.1.4 在 Anaconda 中安装 第三方库	118
4.2 数据准备.....	119
4.2.1 数据类型.....	119
4.2.2 数据结构.....	120
4.2.3 数据导入.....	128
4.2.4 数据导出.....	131
4.3 数据处理.....	133
4.3.1 数据清洗.....	133
4.3.2 数据抽取.....	138
4.3.3 排名索引.....	147
4.3.4 数据合并.....	151
4.3.5 数据计算.....	154
4.3.6 数据分组.....	156
4.3.7 日期处理.....	157
4.4 数据分析.....	162
4.4.1 基本统计.....	162
4.4.2 分组分析.....	163
4.4.3 分布分析.....	165
4.4.4 交叉分析.....	167
4.4.5 结构分析.....	169
4.4.6 相关分析.....	170
4.5 数据可视化.....	172
4.5.1 饼图	172
4.5.2 散点图	174
4.5.3 折线图	176
4.5.4 柱形图	180
4.5.5 直方图	183
本章小结	184
练习	184
第 5 章 其他	187
5.1 文件读写操作	188
5.1.1 文件的读写方法	189
5.1.2 文件的其他方法	190
5.1.3 文件的存储和读取	190
5.2 with 语句	192
5.3 Anaconda 下安装 statsmodels 包	193
5.4 关于 Spyder 界面恢复默认状态的 处理	195
5.5 关于 Python 计算精度的问题	197
5.6 矩阵运算	200
5.6.1 创建矩阵	200
5.6.2 矩阵属性	200
5.6.3 解线性方程组	201
5.6.4 线性规划最优解	202
5.7 正则表达式	203
5.8 使用 urllib 打开网页	209
5.9 网页数据抓取	212
5.10 读取文档	217
本章小结	222
练习	222
参考文献	224



第 1 章

Python 简介



Python 语言是一种简单而又功能强大的编程语言。通过学习，你会发现，Python 语言注重的是如何解决问题，而不是编程语言的语法和结构。

Python 的官方介绍是：Python 是一种简单易学、功能强大的编程语言，它有高效率的高层数据结构，能简单而有效地实现面向对象编程。

Python 简洁的语法和对动态输入的支持，再加上解释性语言的本质，使得它在大多数平台上的许多领域中都是一个理想的脚本语言，特别适用于快速的应用程序开发。不需要任何编程基础，完全可以从零开始学习。

注意： Python 语言的创造者 Guido van Rossum 是根据英国广播公司的节目“蟒蛇飞行马戏”来命名这个语言的，Python 的英文本意是“巨蛇，大蟒”。

Python 确实是一种十分精彩且强大的语言。它合理地结合了高性能及使得编写程序简单有趣的特色。

Python 的缺点：前后版本不兼容。这确实是让新、老学习人员感到有点头痛的事情。

因为前后版本不兼容，导致许多人为选择 Python 2.x 还是 Python 3.x 发愁。本书推荐使用 Python 3.x。

的确，当前还有相当多的程序员在使用 Python 2.7，不过 Python 3.x 才是 Python 发展的未来，这就像 Windows 7 和 Windows 10 谁是未来一样。

我们发现，Python 3.x 中的新特性确实很妙，很值得进行深入学习。

其实，我们也不用担心，如果了解了 Python 3.x，则 Python 2.7 的代码阅读起来是根本不成问题的。

1.1 安装 Python

Windows 用户可以访问 <https://python.org/download>，从网站中下载最新的版本，大小约为 27.4MB。与其他大多数语言相比，Python 的安装包算是十分紧凑的，其安装过程与其他 Windows 软件类似。

在本书即将完成的时候，我们使用的是最新版 Python 3.5.1，所使用的计算机系统为 Windows 10。

安装 Python 很简单，双击 python-3.5.1.exe，勾选 Add Python 3.5 to PATH，再单击 Install Now 即可，如图 1-1 所示，其下方已经显示了安装路径。安装完毕后，会显示安装成功界面，最后单击 Close 按钮就可以使用了。

安装完成后，在“开始”菜单中会显示安装目录，如图 1-2 所示。当我们要编写代码时，直接选择 IDLE 命令即可。

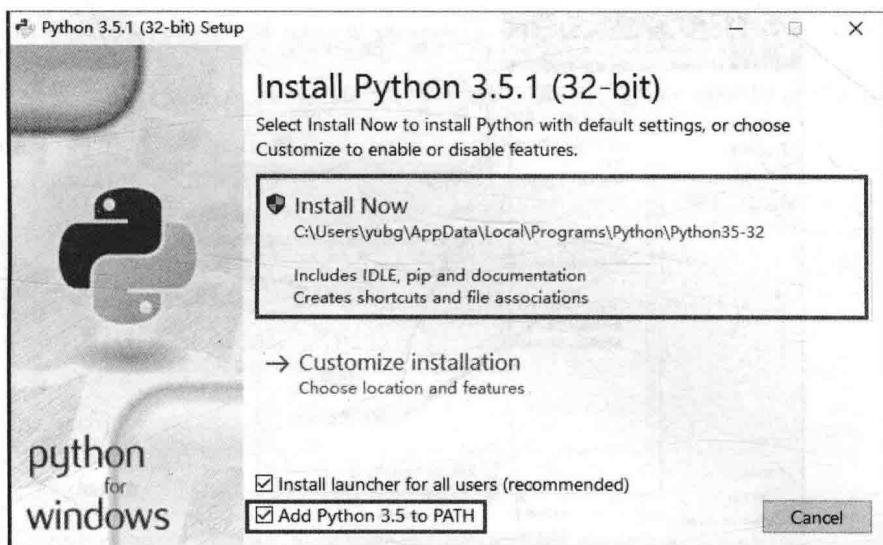


图 1-1 安装界面



图 1-2 “开始”菜单中的安装目录

如果是在 Windows 7 系统中，安装完毕后，还要进行环境的配置(以下是在 Windows 7 系统上安装的 Python 3.3 版本，3.5 版本的安装方法大致相同)，具体方法如下。

打开“控制面板”窗口，单击“系统”图标，打开“系统”窗口，在左侧单击“高级系统设置”图标，将弹出“系统属性”对话框。在该对话框中单击“环境变量”按钮，将弹出“环境变量”对话框，在“系统变量”列表框中选择 Path 选项，然后单击“编辑”按钮，在弹出的“编辑系统变量”对话框中编辑 Path 变量。把“;C:\Python33”添加到变量值的末尾，如图 1-3 所示。当然，前提是 Python 已经正确地安装在 C 盘的根目录下，即 C 盘中已经存在 Python33 文件夹。

然后在 DOS Shell 命令提示符下输入“python”，如果看到如图 1-4 所示的信息，就说明 Python 已经安装成功了。

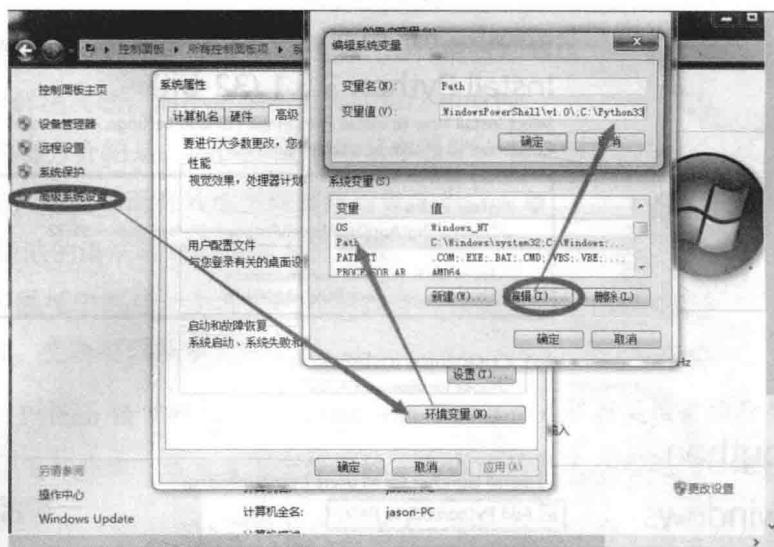


图 1-3 系统变量的设置

```
命令提示符
Microsoft Windows [版本 6.3.9600]
(c) 2013 Microsoft Corporation。保留所有权利。
C:\Users\jubg>cd ..
C:\Users>cd ..
C:\>cd python33
C:\Python33>python
Python 3.3.5 (v3.3.5:62cf4e77f785, Mar  9 2014, 10:35:05) [MSC v.1600 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> ^Z

C:\Python33>
```

图 1-4 在命令提示符下测试 Python 安装

图 1-4 中显示的是在 C 盘下已安装 Python，目录为 C:\Python33。如果与此不一致，需要先下载并安装 Python。

后面还会介绍 Python 的其他安装方法，目的是为了避免安装复杂的 Python 数据包，如 pandas、numpy 等。

关于 Python 下载和学习的网站很多，例如：

- <http://freelycode.com/fcode/downloadinstall?listall=True>
- www.freelycode.com
- <http://pythontutor.com/>

综上所述，对于 Windows 系统，要安装 Python，只须下载安装程序，然后双击它就可以了，是非常简单的。从现在起，我们假设已经在计算机系统里安装了 Python 3.5。

打开 Python 的 IDLE，启动 Python 解释器。

我们在>>>提示符后面输入 print('Hello World')，然后按 Enter 键，应该可以看到输出了 Hello World，如图 1-5 所示。

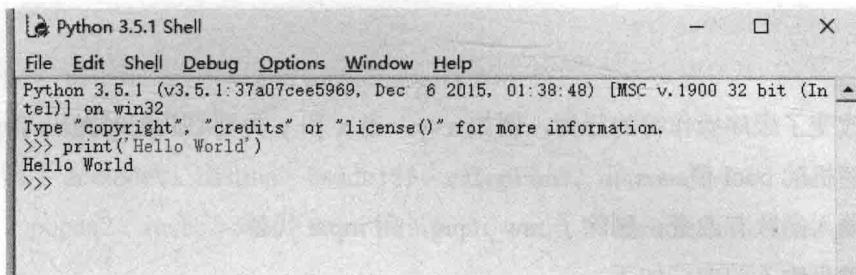


图 1-5 IDLE 界面

注意： 此处的>>>为系统自动显示的提示符，不需要人为地输入，而程序中所涉及的括号()、引号“”等，都需要在英文半角状态下输入。

1.2 Python 2 和 Python 3 的区别

本节我们讲解 Python 2 和 Python 3 的主要区别。

1. 性能

Python 3.0 运行 PyStone Benchmark 的速度比 Python 2.5 慢 30%。Guido 认为 Python 3.0 有极大的优化空间，在字符串和整型操作上可以取得很好的优化结果。目前 Python 3.5 版本的性能已经优于版本 2.7。

2. 编码

Python 3.x 源码文件默认使用 utf-8 编码，这就使得以下代码是合法的，但在 Python 2.x 中是不可思议的事情，对于中国人来说是个“福音”：

```
>>> 中国 = 'china'
>>> print(中国)
china
>>>
```

3. 语法

- (1) 去除了不等号<>，全部改用!=。
- (2) 关键词加入 as 和 with，还有 True、False、None。
- (3) 整型除法返回浮点数，要得到整型结果，须使用//。



(4) 去除 print 语句，加入 print()函数实现相同的功能。同样地，还有 exec 语句，已经改为 exec()函数。

例如：

```
print "The answer is", 2*2          # 2.x
print("The answer is", 2*2)         # 3.x
```

(5) 改变了顺序操作符的行为，例如 $x < y$ ，当 x 和 y 类型不匹配时抛出 TypeError，而不是返回随机的 bool 值。

(6) 输入函数有改变，删除了 raw_input，用 input 替代。

读取键盘输入的方法如下：

```
guess = int(raw_input('Enter an integer : '))      # 2.x
guess = int(input('Enter an integer : '))           # 3.x
```

(7) 删除了 cmp() 比较函数。

4. 数据类型

(1) Python 3.x 去除了 long 类型，现在只有一种整型——int，但它的行为就像 2.x 版本的 long。

(2) dict 的.keys()、.items 和.values()方法返回迭代器，而先前的 iterkeys()等函数都被废弃。同时去掉的还有 dict.has_key()，用 in 替代。

5. 异常

(1) 所有异常都从 BaseException 继承，并删除了 StandardError。

(2) 去除了异常类的序列行为和.message 属性。

(3) 用 raise Exception(args) 替代 raise Exception, args 语法。

(4) 捕获异常的语法改变，引入了 as 关键字来标识异常实例。在 Python 2.5 中：

```
>>> try:
    raise NotImplementedError('Error')
except NotImplementedError, error:
    print error.message

Error
>>>
```

在 Python 3.0 中：

```
>>> try:
    raise NotImplementedError('Error')
except NotImplementedError as error:      # 注意这里的 as
```

```
print(str(error))
Error
>>>
```

6. 模块变动

- (1) 移除了 cPickle 模块，使用 pickle 模块代替。
- (2) 移除了 imageop 模块。
- (3) 移除了 audiodev、bastion、bsddb185、exceptions、linuxaudiodev、md5、mimify、MimeWriter、popen2、rexec、sets、sha、stringold、strop、sunaudiodev、timing 和 xmllib 模块。
- (4) 移除了 bsddb 模块(单独发布，可以从 <http://www.jcea.es/programacion/pybsddb.htm> 获取)。
- (5) 移除了 new 模块。
- (6) os.tmpnam()和 os.tmpfile()函数被移动到 tmpfile 模块下。

7. 其他

- (1) xrange()改名为 range()，要想使用 range()获得一个 list，必须显式调用：

```
>>> list(range(10))
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>>
```

- (2) bytes 对象不能 hash，也不支持 b.lower()、b.strip()和 b.split()方法，但对于后两者，可以使用 b.strip(b'\n\t\f')和 b.split(b' ')来达到相同的目的。
- (3) zip()、map()和 filter()都返回迭代器。
- (4) file 类被废弃，在 Python 2.5 中：

```
>>> file
<type 'file'>
>>>
```

在 Python 3.x 中：

```
>>> file
Traceback (most recent call last):
File "<pyshell120>", line 1, in <module>
file
NameError: name 'file' is not defined
>>>
```

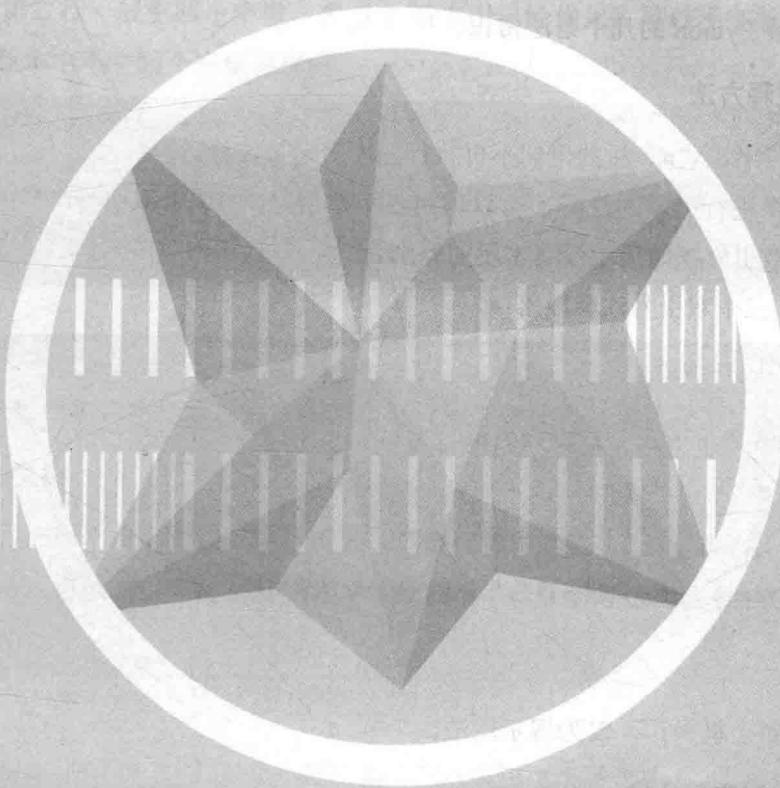


本 章 小 结

本章主要学习了 Python 的安装和 IDLE 的启用，以及了解了 Python 2.7 和 Python 3.x 之间的差异。

练 习

- (1) 请将 IDLE 的 Shell 界面字体调试成 18 号等线 light 字体。
- (2) 在 Shell 编辑器的>>>后输入 help(), 查看本机安装的 Python 版本信息。



第 2 章

Python 数据类型与运算



我们先了解 Python 的几个语法常识。

1. 代码注释方法

(1) 在一行中，“#”后的语句不再执行，而表示被注释。

(2) 如果要进行大段的注释，可以使用三个单引号('')或者双引号("")将注释内容包围。单引号和双引号在使用上没有本质的差别。

【例 2-1】三个双引号注释段落：

```
# -*- coding: utf-8 -*-
"""
Created on Sun Mar 13 21:20:06 2016
@author: yubg
"""

lis=[1,2,3]
for i in lis:    #半角状态冒号不能少，下一行注意缩进
    i+=1
print(i)
```

本例不需要上机操作，仅为展示用法。

2. 用缩进来表示分层

Python 不像 C 语言那样用{}来表示语句块，而是通过让代码缩进 4 个空格来表示分层，当然也可以使用 Tab 键，但不要混合使用 Tab 键和空格来进行缩进，否则会使程序在跨平台时不能正常工作，官方推荐的做法是使用四个空格。

一般来说，行尾遇到“:”就表示下一行缩进的开始，如例 2-1 中的“for i in lis”行尾有冒号，下一行的“i+=1”就需要缩进四个空格。

3. 语句断行

一般来说，Python 中的一条语句占一行，在每条语句的结尾处不需要使用分号();。但在 Python 中也可以使用分号，表示将两条简单语句写在一行。但如果一条语句较长，要分几行来写，可以使用“\”来进行换行。分号还有个作用，使用在一行语句的末尾，表示对本行语句的结果不打印输出。一般地，系统能够自动识别换行，如在一对括号中间或三引号之间均可换行。例如下面代码中的第三行较长，若要对其分行，则必须在括号内进行(包括圆括号、方括号和花括号)：

```
from pandas import DataFrame  #导入模块中的函数，后面再讲
from pandas import Series
df = DataFrame({'age':Series([26,85,64]),'name':Series(['Ben','Joh','Jef'])})
print(df)
```

分行后的第二行一般空四个空格，在3.5版本中已经优化，可以不空四个空格，但是在较低的3.x版本中不空四个空格会报错。

```
from pandas import DataFrame
from pandas import Series
df = DataFrame({'age':Series([26,85,64]),      #此语句分成了两行
                'name':Series(['Ben','Joh','Jef'])})
print(df)
```

4. print()的作用

print()会在输出窗口中显示一些文本或结果，便于验证和显示数据。

5. 使用转义符

如果需要在一个字符串中嵌入一个引号，该如何操作？

有两种方法：可以在引号前加反斜杠(\)，或者用不同的引号包围这个引号。

例如：

```
>>>s1='I\'am a boy.' #可以使用转义符\
>>>print(s1)
I'am a boy.

>>>s2="I'am a boy." #也可以用不同的引号包围起来，此处用双引号是为了区分单引号
>>>print(s2)
I'am a boy.
>>>
```

转义符详见本章2.2.5小节的内容。

2.1 数 据 类 型

Python总共有6种数据类型，分别是数字型(Numbers)、字符串型(String)、列表型(List)、元组型(Tuple)、集合型(Sets)和字典型(Dictionaries)。

数字型又可划分为整数型(int)、浮点型(float)、布尔型(bool)和复数型(complex)。

在Python中有4种类型的数——整数、长整数、浮点数和复数。

例如，2是一个整数的例子。

长整数不过是大一些的整数。

3.23和52.3E-4是浮点数的例子，E标记表示10的幂。52.3E-4表示 52.3×10^{-4} 。

(-5+4j)和(2.3-4.6j)表示的是复数。