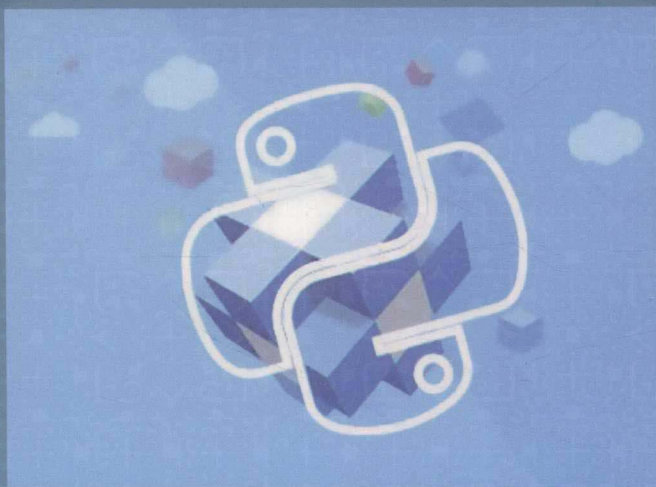


“十三五”普通高等教育规划教材

Python 3

程序设计基础

周元哲 编著



提供电子课件、习题答案、源代码

<http://www.cmpedu.com>



机械工业出版社
CHINA MACHINE PRESS

“十三五”普通高等教育规划教材

Python 3 程序设计基础

周元哲 编著



机械工业出版社

本书对 Python 3 进行了全面的阐述,内容分为基础和应用两部分。其中,基础部分包括 Python 编程概述、基本数据类型、组合数据类型、输入输出文件、顺序与选择结构、循环结构、函数与模块、面向对象程序设计基础。应用部分包括图形用户界面、科学计算、网络爬虫、数据库应用、Web 开发、电子游戏开发和异常处理与测试。附录给出了全国计算机等级考试二级 Python 语言程序设计考试大纲、Python 内置函数和基础部分的题目的参考答案。

本书适合作为高等院校相关专业 Python 程序设计的教材或教学参考书,也可供从事计算机应用开发的各类技术人员参考,亦可作为全国计算机等级考试、软件技术资格与水平考试的培训资料。

本书配套授课电子课件,需要的教师可登录 www.cmpedu.com 免费注册,审核通过后下载,或联系编辑索取(QQ: 2850823885; 电话: 010-88379739)。

图书在版编目(CIP)数据

Python 3 程序设计基础 / 周元哲编著. —北京:机械工业出版社, 2019.6
“十三五”普通高等教育规划教材
ISBN 978-7-111-62778-4

I. ①P… II. ①周… 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 096041 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策划编辑:郝建伟 责任编辑:颀天

责任校对:张艳霞 责任印制:张博

三河市国英印务有限公司印刷

2019 年 6 月第 1 版·第 1 次印刷

184mm×260mm·15.5 印张·384 千字

0001—2500 册

标准书号:ISBN 978-7-111-62778-4

定价:49.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

客服电话:010-88361066

010-88379833

010-88326294

网络服务

机工官网:www.cmpbook.com

机工官博:weibo.com/cmp1952

金书网:www.golden-book.com

封面无防伪标均为盗版 教育服务网:www.cmpedu.com

前 言

Python 已经成为继 Java、C++之后的第三大语言，作为一种面向对象的解释型计算机程序设计语言，具有简单易学、免费开源、丰富和强大的库等特点。

本书内容分为基础部分和应用部分。其中，基础部分包括 Python 编程概述、基本数据类型、组合数据类型、输入输出文件、顺序与选择结构、循环结构、函数与模块、面向对象程序设计基础。应用部分包括图形用户界面、科学计算、网络爬虫、数据库应用、Web 开发、电子游戏开发和异常处理与测试。附录给出了全国计算机等级考试二级 Python 语言程序设计考试大纲、Python 内置函数和基础部分的习题答案。

学习 Python，最重要的是学习编程思想，Python 2 和 Python 3 只是存在少量的语法差异，其编程思想基本共通。本书的作者多年从事计算机编程语言的教学工作，编写过 C 语言、VB 6.0、VB.NET、Python 2、算法等相关教材，一直致力于培养学生掌握编程思想及编程方法，意在提高学生的编程开发能力。学习编程语言的重要方法就是去实践，本书的所有程序都在 Anaconda 进行调试和运行，希望读者注重读代码和写代码的异同点，注重养成编写代码的习惯。

本书内容精练、文字简洁、结构合理，实训题目经典实用、综合性强，明确定位面向初、中级读者，由“入门”起步，侧重“提高”；特别适合作为高等院校相关专业 Python 程序设计的教材或教学参考书，也可供从事计算机应用开发的各类技术人员参考，亦可作为全国计算机等级考试、软件技术资格与水平考试的培训资料。

在本书编写过程中，西安邮电大学惠小强、焦继业、孟伟君、张庆生、李晓戈审阅部分手稿，提出了很多宝贵的意见。本书参阅了大量中英文的专著、教材、论文、报告及网上的资料，由于篇幅所限，未能一一列出。在此，一并表示敬意和衷心的感谢。

由于编著者水平有限，时间紧迫，本书难免有疏漏之处，恳请广大读者批评指正。本书编著者的电子信箱是 zhouyuanzhe@163.com。

编著者

目 录

前言

第1章 Python 编程概述	1	2.5 习题	26
1.1 Python 简介	1	第3章 组合数据类型	28
1.1.1 Python 发展历程	1	3.1 序列概述	28
1.1.2 Python 特点	1	3.2 序列分类	28
1.1.3 Python 应用场合	2	3.2.1 列表	28
1.2 Python 解释器安装	2	3.2.2 元组	31
1.2.1 Linux 下安装 Python	2	3.2.3 字符串	32
1.2.2 Windows 下安装 Python	3	3.3 序列通用	34
1.3 Python 开发环境	5	3.3.1 通用操作	34
1.3.1 IDLE	5	3.3.2 通用方法	35
1.3.2 PyCharm	5	3.3.3 通用函数	36
1.3.3 Anaconda	7	3.4 字典	37
1.4 Python 2.x 和 Python 3.x 的区别	13	3.4.1 字典概念	37
1.5 学习建议	13	3.4.2 字典操作	38
1.6 习题	14	3.5 集合	40
第2章 基本数据类型	15	3.5.1 集合概念	40
2.1 数据类型	15	3.5.2 集合操作	41
2.1.1 数值	15	3.5.3 集合运算	42
2.1.2 字符串	16	3.6 数据类型转换	43
2.2 变量	18	3.7 习题	43
2.2.1 标识符	18	第4章 输入输出文件	44
2.2.2 变量引用	19	4.1 输入输出	44
2.3 运算符	19	4.1.1 输入	44
2.3.1 算术运算符	19	4.1.2 输出	44
2.3.2 关系运算符	20	4.2 文件	46
2.3.3 赋值运算符	21	4.2.1 ASCII 文件	46
2.3.4 逻辑运算符	22	4.2.2 二进制文件	47
2.3.5 位运算符	23	4.3 文件操作	47
2.3.6 成员运算符	23	4.3.1 打开关闭文件	47
2.3.7 身份运算符	24	4.3.2 读写文件	48
2.4 表达式	24	4.3.3 文件相关方法	50
2.4.1 表达式计算	24	4.4 习题	53
2.4.2 数据类型转换	25	第5章 顺序与选择结构	54

5.1 程序设计流程	54	7.2 函数调用	81
5.1.1 算法	54	7.3 参数传递	82
5.1.2 程序流程图	55	7.3.1 实参与形参	82
5.1.3 三种基本结构	56	7.3.2 传对象引用	83
5.2 顺序结构	57	7.4 参数分类	84
5.3 选择结构	57	7.4.1 必备参数	84
5.3.1 单分支	58	7.4.2 默认参数	84
5.3.2 双分支	59	7.4.3 关键参数	84
5.3.3 多分支	59	7.4.4 不定长参数	85
5.3.4 分支嵌套	61	7.5 两类特殊函数	85
5.4 代码书写规则	63	7.5.1 匿名函数	85
5.4.1 缩进	63	7.5.2 递归函数	86
5.4.2 多行语句	63	7.6 变量作用域	90
5.4.3 空行	64	7.6.1 局部变量	90
5.4.4 注释	64	7.6.2 全局变量	90
5.5 习题	64	7.7 模块	91
第 6 章 循环结构	66	7.7.1 命名空间	91
6.1 循环概述	66	7.7.2 模块定义与导入	91
6.1.1 三个表达式	66	7.8 习题	92
6.1.2 循环分类	67	第 8 章 面向对象程序设计基础	93
6.2 while 语句	67	8.1 面向对象概述	93
6.2.1 基本形式	67	8.1.1 基本概念	93
6.2.2 else 语句	68	8.1.2 与面向过程对比	94
6.2.3 无限循环	69	8.1.3 三大特性	94
6.3 for 语句	69	8.2 类和对象	95
6.3.1 应用序列类型	69	8.3 类属性与实例属性	96
6.3.2 内置函数 range()	70	8.3.1 实例属性	96
6.4 循环嵌套	72	8.3.2 类属性	97
6.4.1 基本形式	72	8.4 方法	98
6.4.2 循环嵌套实现	73	8.4.1 对象方法	98
6.5 辅助语句	76	8.4.2 类方法	99
6.5.1 break 语句	76	8.4.3 静态方法	99
6.5.2 continue 语句	77	8.5 构造函数与析构函数	100
6.5.3 pass 语句	78	8.5.1 构造函数	100
6.6 编程规范	78	8.5.2 析构函数	101
6.7 习题	79	8.6 继承性	101
第 7 章 函数与模块	80	8.6.1 单继承	101
7.1 函数概述	80	8.6.2 多继承	103
7.2 函数定义与调用	80	8.7 多态性	103
7.2.1 函数定义	80	8.8 习题	104

第 9 章 图形用户界面	105	10.5.2 Series 和 DataFrame	135
9.1 概述	105	10.5.3 pandas 与 Excel	137
9.1.1 界面设计准则	105	第 11 章 网络爬虫	141
9.1.2 GUI 库简介	105	11.1 网络爬虫概述	141
9.2 tkinter	106	11.1.1 概念	141
9.2.1 简介	106	11.1.2 爬取流程	141
9.2.2 基本控件	106	11.2 爬取网页	141
9.2.3 布局	110	11.2.1 urllib 库	141
9.2.4 绘图	113	11.2.2 requests 库	142
9.3 PyQt	117	11.3 解析信息	143
9.3.1 简介	117	11.3.1 正则表达式	143
9.3.2 PyQt5 举例	118	11.3.2 BeautifulSoup 库	147
9.4 wxPython	119	11.3.3 selenium_phantomjs	151
9.4.1 简介	119	11.4 爬虫框架 Scrapy	153
9.4.2 wxPython 举例	119	11.4.1 Scrapy 简介	153
第 10 章 科学计算	121	11.4.2 爬取流程	155
10.1 科学计算概述	121	11.5 综合比较	157
10.1.1 Python 与科学计算	121	第 12 章 数据库应用	158
10.1.2 与 MATLAB 相比	121	12.1 数据库概念	158
10.2 NumPy	123	12.1.1 关系型数据库	158
10.2.1 NumPy 简介	123	12.1.2 结构化查询语言	159
10.2.2 创建数组	123	12.2 SQLite	160
10.2.3 查看数组	125	12.2.1 连接对象和游标	160
10.2.4 索引和切片	125	12.2.2 操作 SQLite	161
10.2.5 矩阵运算	126	12.3 MySQL	164
10.3 Matplotlib	127	12.3.1 安装 MySQL	164
10.3.1 Matplotlib 简介	127	12.3.2 PyMySQL 操作数据库	166
10.3.2 线性图	127	12.4 MongoDB	169
10.3.3 散点图	128	12.4.1 安装 MongoDB	169
10.3.4 饼状图	129	12.4.2 Pymongo 操作数据库	170
10.3.5 条形图	129	12.5 SQLAlchemy	171
10.3.6 直方图	130	12.5.1 SQLAlchemy 引入	171
10.4 SciPy	130	12.5.2 SQLAlchemy 操作流程	172
10.4.1 SciPy 简介	131	第 13 章 Web 开发	174
10.4.2 二项分布	131	13.1 Web 开发简介	174
10.4.3 正态分布	132	13.1.1 Web 开发历程	174
10.4.4 beta 分布	133	13.1.2 MVC 设计模式	174
10.4.5 指数分布	133	13.2 web2py 框架	175
10.5 pandas	134	13.2.1 安装 web2py	175
10.5.1 pandas 简介	134	13.2.2 web2py 应用	176

13.3 Django 框架·····	181	15.3.1 print 函数·····	205
13.3.1 安装 Django·····	181	15.3.2 pdb·····	205
13.3.2 Django 应用·····	182	15.3.3 ipdb·····	206
13.4 Flask 框架·····	184	15.3.4 Spyder 调试·····	207
13.4.1 安装 Flask·····	184	15.4 测试·····	209
13.4.2 Flask 应用·····	184	15.4.1 测试驱动·····	209
第 14 章 电子游戏开发 ·····	187	15.4.2 单元测试·····	209
14.1 电子游戏与 Pygame·····	187	15.4.3 Web 测试·····	211
14.2 Pygame·····	187	附录 ·····	215
14.2.1 Pygame 开发流程·····	187	附录 A 全国计算机等级考试二级	
14.2.2 Pygame 模块·····	189	Python 语言程序设计考试	
14.3 贪吃蛇·····	193	大纲 (2018 年版)·····	215
14.3.1 需求·····	193	附录 B 内置函数·····	216
14.3.2 具体实现·····	194	B.1 数学函数·····	216
第 15 章 异常处理与测试 ·····	201	B.2 转换函数·····	217
15.1 错误类型·····	201	B.3 随机数函数·····	218
15.1.1 语法错误·····	201	B.4 时间函数·····	218
15.1.2 运行时错误·····	201	B.5 列表函数·····	220
15.1.3 逻辑错误·····	201	B.6 元组函数·····	220
15.2 捕获和处理异常·····	202	B.7 字符串函数·····	220
15.2.1 try/except/else 语句·····	202	B.8 字典函数·····	221
15.2.2 try/finally 语句·····	203	B.9 集合函数·····	222
15.2.3 raise 语句·····	203	附录 C 参考答案·····	222
15.2.4 自定义异常·····	204	参考文献 ·····	240
15.3 调试方法·····	205		

第1章 Python 编程概述

Python 是继 Java、C++ 之后的第三大程序设计语言。本章讲述了 Python 的发展历程、特点以及应用场合，介绍了 Python 在 Linux 和 Windows 下的安装步骤，IDLE、PyCharm 和 Anaconda 三个编辑器的安装和配置，最后给出了 Python 2.x 和 Python 3.x 的区别，以及学习 Python 的一些建议。

1.1 Python 简介

本节介绍 Python 的发展历程、特点，以及 Python 的应用场合。

1.1.1 Python 发展历程

Python 由 Guido van Rossum 于 1989 年底在荷兰国家数学和计算机科学研究所以发明，第一个公开发行人版发行于 1991 年。Python 2.0 于 2000 年 10 月 16 日发布，实现垃圾回收，并支持 Unicode。Python 3.0 被称为 Python 3000，或简称 Py3k，发布于 2008 年 12 月 3 日，相对于 Python 的早期版本做了较大的升级，但未考虑向下相容，导致早期 Python 版本设计的程序无法在 Python 3.0 上正常执行。其中，Python 2.6 和 2.7 作为一个过渡版本，基本使用了 Python 2.x 的语法和库，同时考虑了向 Python 3.0 的迁移，允许使用部分 Python 3.0 的语法与函数。

2018 年 3 月，Python 作者宣布 Python 2.7 将于 2020 年 1 月 1 日终止支持，只支持 Python 3。

1.1.2 Python 特点

Python 具有简单、现代、类型安全、性能优良、效率高等特点，如下所示。

1) 简单易学：Python 有相对较少的关键字，结构简单，语法简单，使得初学者快速养成良好的编码习惯，在学习过程中便于专注程序本身的逻辑和算法，易于快速上手学习。

2) 易于阅读：Python 代码定义得更清晰。

3) 丰富的数据类型：Python 具有序列、列表、元组和字典等数据结构，便于实现各种算法。

4) 丰富的标准库：Python 具有众多的库，支持图形处理、科学计算、Web 开发、爬虫、人工智能等。

5) 可移植：基于其开放源代码的特性，Python 被移植到许多平台，与 UNIX、Windows 和 MacOS 兼容。

6) 可嵌入: Python 可以嵌入到 C/C++ 程序中, 具有“脚本化”的能力。

7) 面向对象: Python 是强面向对象的语言, 程序中任何内容统称为对象, 包括数字、字符串、函数等。

8) 解释型语言: 计算机高级编程语言必须将源程序通过“翻译程序”翻译成目标程序, 计算机才能识别和执行。这种“翻译”通常有两种方式: 一种是编译执行; 另一种是解释执行。C、C++ 等采用编译执行方式, 编译执行是指源程序代码先由编译器编译成二进制的可执行指令, 通常执行效率高。Python 和 Java 属于解释执行, 解释执行是指源代码程序被解释器转换成称为字节码的中间形式, 由虚拟机负责在不同的计算机运行, 便于移植。

1.1.3 Python 应用场合

Python 功能强大, 在图形处理、科学计算、数据库编程、Web 开发、爬虫、机器学习、人工智能等方面都有应用, 具体如下。

(1) GUI 软件开发

Python 具有 wxPython、PyQT 等工具, 使得 Python 可以快速开发出图形用户界面。

(2) 网络应用开发

Python 提供了标准 Internet 模块, 可以广泛应用到各种网络任务中。Web.py、Django、Flask 等网络框架能够快速构建功能完善和高质量的网站。

(3) 游戏开发

Pygame 是建立在 SDL (Simple DirectMedia Layer) 基础上的软件包, 提供了简单的方式控制媒体信息 (如图像、声音等), 专为电子游戏设计使用。

(4) 科学计算

随着 NumPy、SciPy、Matplotlib 等众多程序库的开发, Python 越来越适合于做科学计算、绘制高质量的 2D 和 3D 图像。相对于科学计算领域最流行的商业软件 Matlab 而言, Python 是一门通用的程序设计语言, 其应用范围更广泛, 有更多的第三方库的支持。

(5) 数据库开发

Python 提供所有主要商业数据库的接口, 如 Oracle、Sybase、MySQL、MongoDB 等。

1.2 Python 解释器安装

本节介绍在 Linux 和 Windows 下, Python 解释器安装的具体步骤。

1.2.1 Linux 下安装 Python

Ubuntu (乌班图) 是基于 Linux 的免费开源 PC 操作系统, 目标在于为一般用户提供一个最新的、同时又相当稳定的主要由自由软件构建而成的操作系统。Ubuntu 具有庞大的社区力量, 用户可以方便地从社区获得帮助。

Ubuntu 下自带含有 Python 2, 在终端下输入 Python, 默认为 Python 2, 如图 1-1 所示。

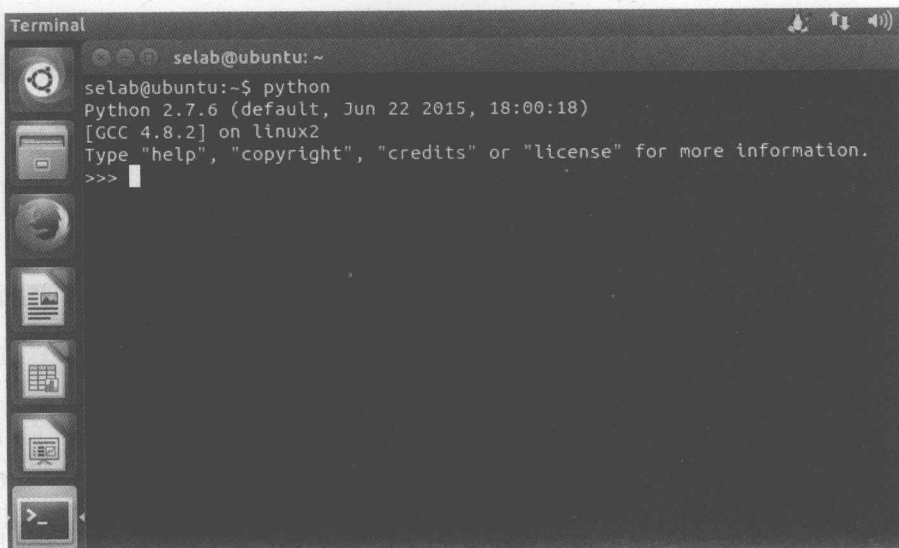


图 1-1 Ubuntu 下内置 Python 2

在 Ubuntu 下安装 Python 3 版本，步骤如下。

步骤 1：下载。

```
wget https://www.python.org/ftp/python/3.6.0/Python-3.6.0a1.tar.xz
```

步骤 2：解压。

```
tar xvf Python-3.6.0a1.tar.xz
```

步骤 3：创建安装文件的路径。

```
mkdir /usr/local/python3
```

步骤 4：编译安装。

```
./configure --prefix=/usr/local/python3  
make  
make install
```

步骤 5：测试。

输入 Python 3 进行测试，输入〈Ctrl + D〉退出。

1.2.2 Windows 下安装 Python

Windows 下安装 Python，一般有如下步骤。

步骤 1：下载 Python 3.6.0 安装包进行安装。在浏览器中输入 <http://www.Python.org>，如图 1-2 所示。

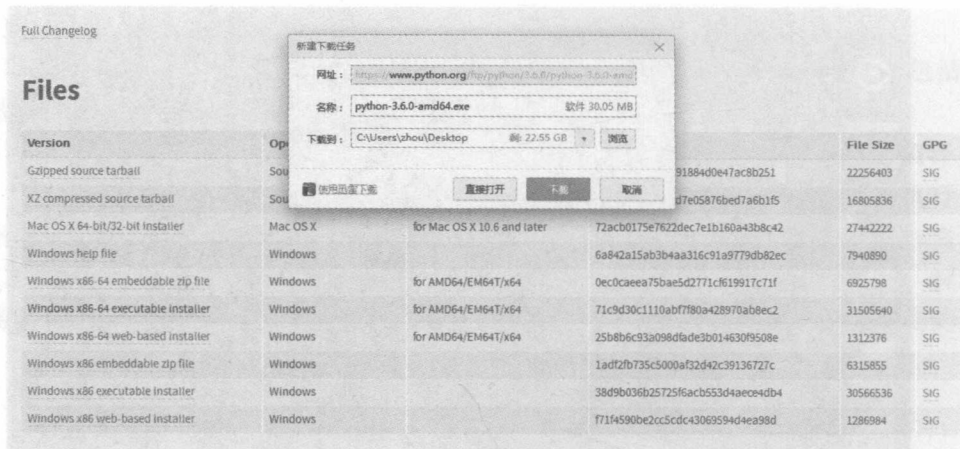


图 1-2 下载 Python 3.6.0

步骤 2: 将 Python 的安装目录添加到 Windows 下的 PATH 变量中, 如图 1-3 所示。

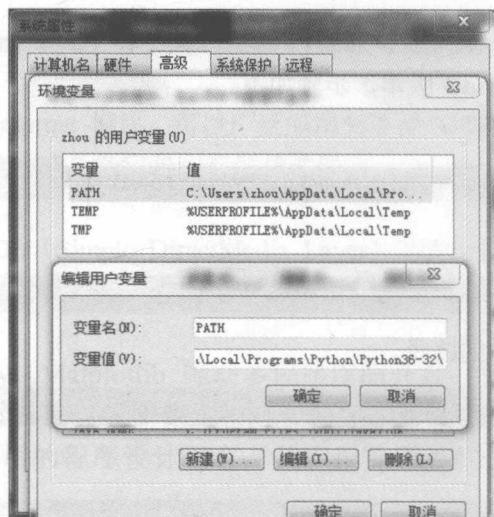


图 1-3 设置环境变量

步骤 3: 测试 Python 安装是否成功。在 Windows 下使用 cmd 打开命令行, 输入 Python 命令, 图 1-4 表示安装成功。

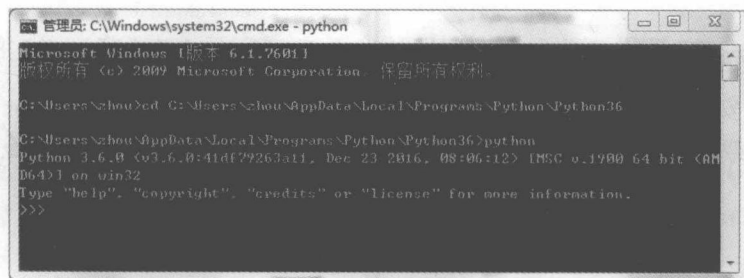


图 1-4 测试 Python 安装是否成功

1.3 Python 开发环境

Python 编辑器众多，除了 Python 自带的 IDLE 之外，还有 notepad++、Eclipse+pydev、Ulipad、Vim 和 emacs 等。其中，Linux 下的 Eclipse with PyDev 和 Windows 下的 PyCharm 功能较为强大，以及 Anaconda 应用较为广泛。

1.3.1 IDLE

IDLE 作为 Python 安装后内置的集成开发工具，包括能够利用颜色突出显示语法的编辑器、调试工具、Python Shell，以及完整的 Python 3 在线文档集。

Python 的 IDLE 具有命令行和图形用户界面两种方式，选择“开始”菜单→“所有程序”→“Python 3.6”→“IDLE (PythonGUI)”来启动 IDLE，直接进入命令行交互式执行 Python 语句，如图 1-5 所示。命令行交互式执行方便快捷，但必须逐条输入语句，不能重复执行，适合测试少量的 Python 代码，不适合复杂的程序设计。

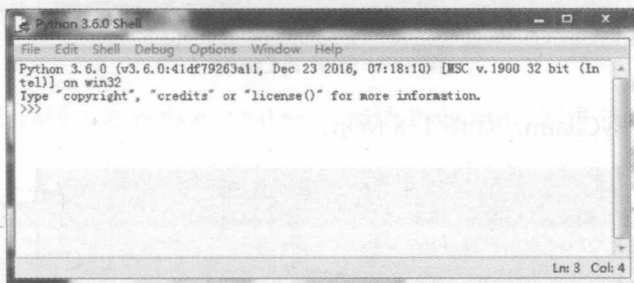


图 1-5 IDLE 的命令行交互模式

Python 的 IDLE 的图形用户界面模式，如图 1-6 所示。

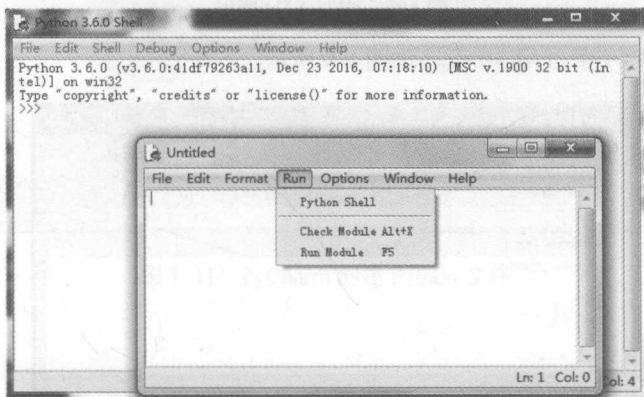


图 1-6 IDLE 的图形用户界面模式

1.3.2 PyCharm

PyCharm 具有一整套可以帮助用户在使用 Python 语言开发时提高效率的工具，比如调

试、语法高亮、Project 管理、代码跳转、智能提示、自动完成、单元测试、版本控制。此外，PyCharm 提供了一些高级功能，以用于支持 Django 框架下的专业 Web 开发。下载 PyCharm 双击安装，如图 1-7 所示。

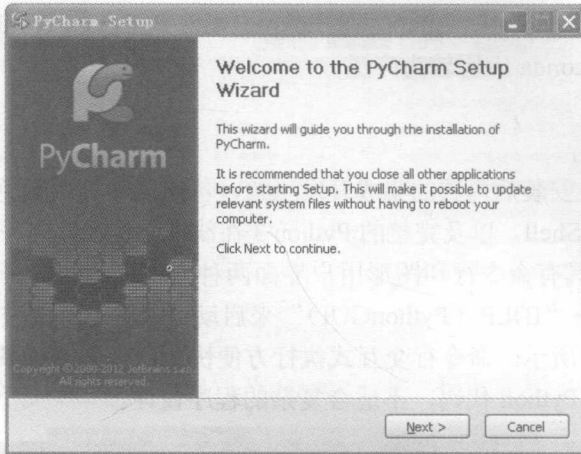


图 1-7 安装 PyCharm

安装结束，运行 PyCharm，如图 1-8 所示。

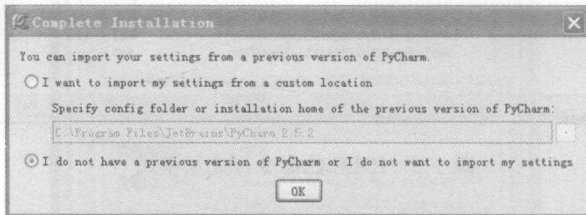


图 1-8 运行 PyCharm

下一步，可以选择免费试用 30 天，如图 1-9 所示。

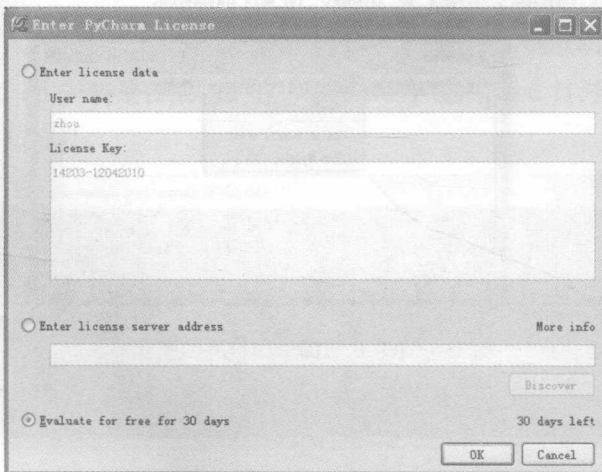


图 1-9 激活 PyCharm

单击 Create New Project，输入项目名、路径、选择 Python 解释器。如果没有出现 Python 解释器，如图 1-10 所示。

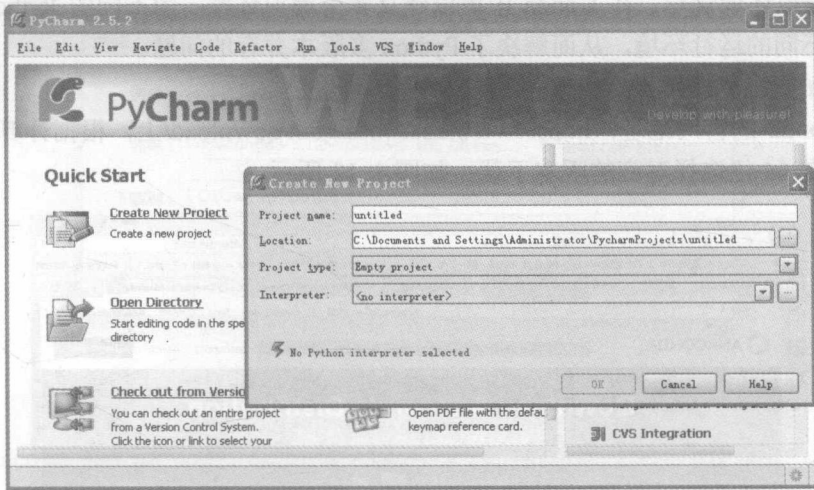


图 1-10 未出现 Python 解释器

选择 Python 解释器版本为 python-3.6.0.msi。启动 PyCharm，创建 Python 文件，如图 1-11 所示。

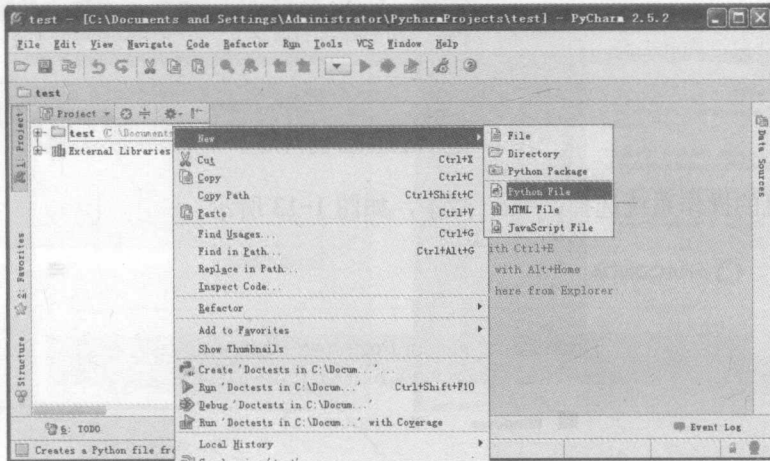


图 1-11 PyCharm 创建 Python 文件

1.3.3 Anaconda

Anaconda 是一个开源的 Python 发行版本，其包含了 Conda、Python 等 180 多个科学包及其依赖项，在数据可视化、机器学习、深度学习等多方面都有涉及，本书重点介绍 Anaconda，所有程序均在 Anaconda 下调试与运行。Anaconda 具有如下特点。

- 提供包管理。使用 Conda 和 pip 安装、更新、卸载第三方工具包简单方便，不需要考虑版本等问题。

- 关注于数据科学相关的工具包。Anaconda 集成了如 NumPy、SciPy、pandas 等数据分析的各类第三方包。
- 提供虚拟环境管理。在 Conda 中可以建立多个虚拟环境，为不同的 Python 版本项目建立不同的运行环境，从而解决了 Python 多版本并存的问题。

Anaconda 安装步骤如下。

在 Anaconda 的官网地址：<https://www.anaconda.com/download/>上，根据计算机的操作系统是 32 位还是 64 位选择对应的版本下载，如图 1-12 所示。

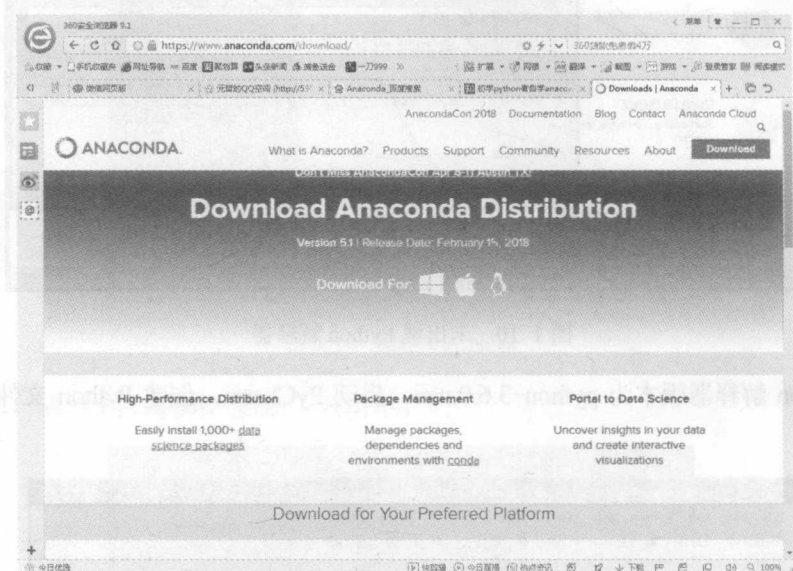


图 1-12 Anaconda 的网站

根据计算机的操作系统选择 Python 版本，如图 1-13 所示。

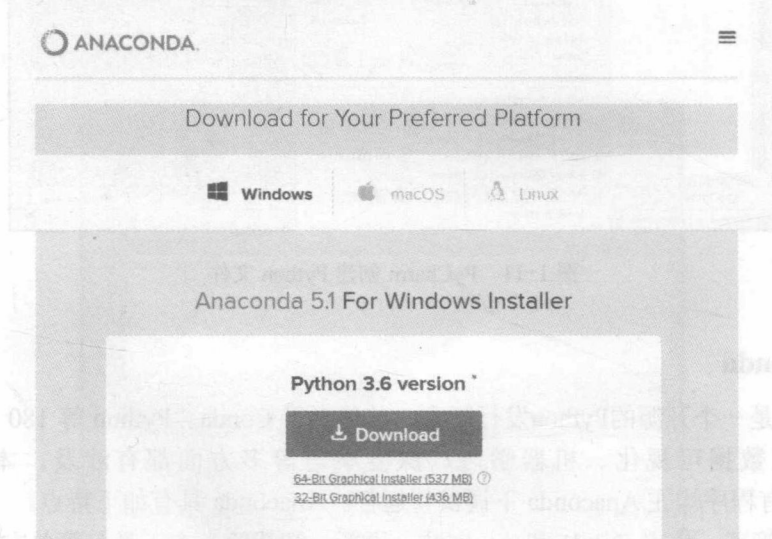


图 1-13 选择 Python 3.6

单击下载 Python 3.6 version, 如图 1-14 所示。



图 1-14 下载 Anaconda 文件

下载 Anaconda3-5.1.0-Windows-x86_64.exe, 大约 500 MB。

注意: 如果是 Windows 10 系统, 注意在安装 Anaconda 软件的时候, 右击“安装软件”→选择“以管理员的身份运行”。

选择安装路径, 例如 C:\Anaconda3, 如图 1-15 所示, 程序运行结果如图 1-16 所示。

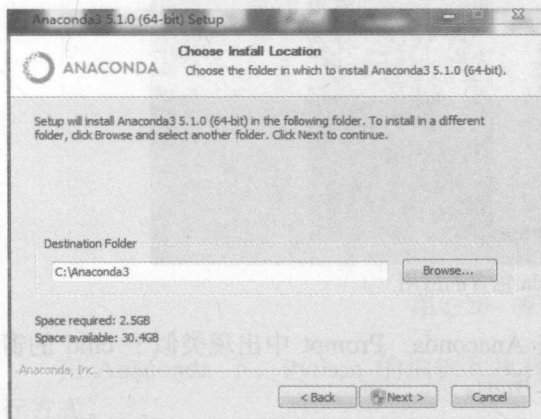


图 1-15 选择安装路径

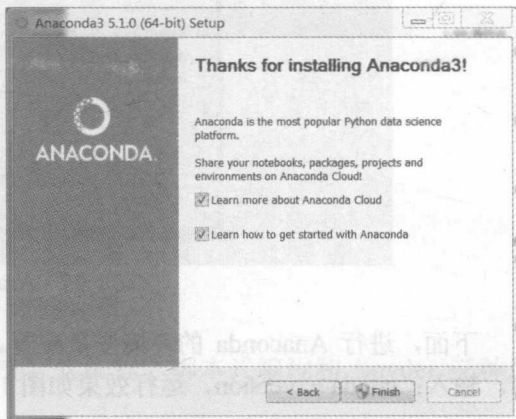


图 1-16 程序运行结果

Anaconda 包含如下应用, 如图 1-17 所示。

- Anaconda Navigator: 用于管理工具包和环境的图形用户界面, 后续涉及的众多管理命令也可以在 Navigator 中手工实现。
- Anaconda Prompt: Python 的交互式运行环境。
- Jupyter Notebook: 基于 Web 的交互式计算环境, 可以编辑易于人们阅读的文档, 用于展示数据分析的过程。
- Spyder: 一个使用 Python 语言、跨平台的、科学运算集成开发环境。相对于 PyDev、PyCharm、PTVS 等 Python 编辑器, Spyder 对内存的需求小很多。