

普通高等教育“十三五”规划教材
新工科建设之路·计算机类专业规划教材

Python

程序设计

从基础入门到实战应用

- 王雷春 主编
- 黄红胜 李珊枝 朱晓钢 艾芳菊 副主编

普通高等教育“十二五”规划教材
新工科建设之路·计算机类专业规划教材

Python

程序设计

从基础入门到实战应用

● 王雷春 主编
● 黄红胜 李珊枝 朱晓钢 艾芳菊 副主编



電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

Python 优雅、简洁，有大量的标准库模块，并且支持数量众多的第三方库，可供科学计算、数据分析、人工智能等相关行业的人员学习和研究，对学科交叉应用也很有帮助。本书分为两部分：教学篇和实验篇。教学篇共 15 章，内容包括 Python 入门，Python 基础，程序设计结构，组合数据，函数，面向对象程序设计，模块、包和库，正则表达式，文件访问，异常处理和单元测试，数据库访问，图形用户界面编程，多进程与多线程，网络程序设计，Python 与人工智能；书中内容以程序设计应用为导向，突出使用 Python 解决实际问题的方法和能力训练。实验篇包括与各章知识对应的实验内容，通过实验培养学生使用 Python 解决实际问题的编程能力。

本书完全面向 Python 3.X，全部案例在 Python 3.7.2 和 PyCharm 2018 Professional 3.5 环境中编写、测试。除基本案例外，教学篇在各章（不包括第 1 章）中都精选和安排了与实际结合紧密的典型用例，让读者既可以通过基本案例学到 Python 基础知识和使用方法，又可以通过典型用例对所学知识进行综合练习和应用，进一步提高编程能力。

本书可作为高等学校计算机程序设计课程的教材，也可供 Python 从业者学习和作为工具书使用。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目 (CIP) 数据

Python 程序设计：从基础入门到实战应用 / 王雷春主编. —北京：电子工业出版社，2019.8
ISBN 978-7-121-36496-9

I. ①P… II. ①王… 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 089261 号

责任编辑：戴晨辰 特约编辑：张燕虹

印 刷：三河市华成印务有限公司

装 订：三河市华成印务有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编：100036

开 本：787×1092 1/16 印张：20.5 字数：524 千字

版 次：2019 年 8 月第 1 版

印 次：2019 年 8 月第 1 次印刷

定 价：59.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010)88254888，88258888。

质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：dcc@phei.com.cn。

前言

Preface

计算机程序设计是高校计算机各专业教学的核心课程，它以高级程序设计语言作为工具，介绍程序设计的思想和方法，为后续相关计算机课程学习奠定基础，也是训练学生使用计算机编程解决实际的重要手段。

与其他程序设计语言(如 C/C++、C#、Java 等)相比，Python 是一门免费、开源、跨平台的高级动态编程语言，在 1991 年推出第 1 版后就迅速得到了各行业人士的青睐，连续多年在 TIOBE 推出的语言排行榜上名列前茅，甚至在 2017 年 7 月由 IEEE Spectrum 推出的编程语言排行榜上位居第 1 名。目前，Python 已经在计算机科学、统计分析、图形图像处理、人工智能、网络爬虫、系统运维等不同行业和领域中得到了广泛的应用和研究。

Python 支持命令式编程、函数式编程和面向对象程序设计，其语法简洁清晰，拥有大量功能强大的标准库模块和第三方库，可让学习者从语法细节中摆脱出来，专注于问题本身的分析、设计和解决该问题的逻辑与方法，不用过多考虑语言本身的细节，因而开发效率高，可以帮助不同领域的设计人员、研究人员、项目管理人员快速实现自己的思路 and 创意。

在国外，Python 已经成为很多著名高校，如卡耐基梅隆大学、麻省理工学院、加州大学伯克利分校、哈佛大学等计算机专业、非计算机专业的重要程序语言教程，甚至一些高中也把它作为程序设计的入门语言。在国内，一些大学的不同专业也陆续开设了 Python 程序设计课程。

考虑到 Python 入门容易，程序简洁，跨平台，以及在科学计算、数据分析和人工智能等方面的良好应用，我们编写了本书。本书以 Python 作为实现工具，介绍程序设计的基本思想和方法，培养学生利用 Python 解决实际问题的能力。

本书分为两部分：教学篇和实验篇。教学篇共 15 章：第 1 章介绍 Python 基本知识 with 概念，开发环境，程序开发步骤，程序结构与编码规范，输入、输出函数等；第 2 章介绍数据类型、常量和变量、运算符、常用特殊内置函数、程序调试等；第 3 章介绍程序设计结构，包括顺序结构、选择结构和循环结构；第 4 章介绍组合数据的使用方法，包括列表、元组、字典和集合；第 5 章介绍函数的定义和调用、参数类型、特殊函数、装饰器及变量作用域等；第 6 章介绍类与对象，类的成员和方法，属性，类的继承与多态，抽象类等；第 7 章介绍常用标准库模块，常用第三方库及模块的使用，自定义模块的创建、调用方法；第 8 章介绍正则表达式的语法规则、使用正则表达式模块匹配和过滤字符串的步骤与方法等；第 9 章介绍文本文件和二进制文件的访问方法；第 10 章介绍程序异常处理和单元测试的方法与步骤；第 11 章介绍在 Python 程序中访问不同类型数据库的方法；第 12 章介绍基于第三方库 wxPython 的图形用户界面程序设计；第 13 章介绍多进程与多线程的创建、使用、通信、同步等；第 14 章介绍网络程序设计，包括基于 TCP 和 UDP 的套接字(Socket)编程、Web 编程等；第 15 章介绍人工智能的基本概念、使用 Sklearn 库和 TensorFlow 框架进行人工智能开发的初步知识。实验篇包括精选的、与各章知识对应的实验题目，以方便读者上机练习。

本书既介绍了 Python 基础知识,如 Python 编程基础、程序设计结构、函数、面向对象程序设计、常用标准库模块和常用第三方库等;又介绍了 Python 中较为专业的内容,如数据库访问、图形用户界面编程、多进程与多线程、网络程序设计等;还对当前的研究热点——人工智能进行了探索,介绍了基于 Python 的机器学习库 Sklearn 和深度学习框架 TensorFlow 的初步使用方法。本书有 300 余个案例,分为两类:一类是与书中每个知识点对应的基本案例;另一类是面向应用的典型案例。前者让读者通过基本案例学习并掌握 Python 基本理论和实践知识,后者通过典型案例提升读者解决较为复杂实际问题的能力。

在 Python 发展过程中,形成了 Python 2.X 和 Python 3.X 两个版本。考虑到 Python 3.X 是 Python 的主流,本书选择 Windows 64 位操作系统下的 Python 3.7.2 作为实现环境。同时,也介绍了开发效率较高的集成开发软件 PyCharm 的使用方法。

本书提供配套的教学资源,包括教学大纲、电子课件、案例源代码、课后习题答案、实验参考答案等。读者可以登录华信教育资源网(www.hxedu.com.cn)下载,也可与作者联系索取。作者邮箱:2430179820@qq.com。

本书可作为高等学校计算机程序设计课程的教材,也可供 Python 从业者学习和作为工具书使用。

本书第 11~15 章由王雷春(湖北大学)编写,第 1~3 章由黄红胜(陆军勤务学院)和李珊枝(武汉晴川学院)编写,第 4~9 章由王雷春和李珊枝编写,第 10 章由朱晓钢(湖北大学)编写,习题和实验部分由艾芳菊(湖北大学)和肖蓉(湖北大学)编写,全书由王雷春统稿。此外,参与本书部分章节编写和案例实现的还有周国玉、吴珊、周鹏等。

由于作者学识水平有限,书中难免存在疏漏和不妥之处,恳请广大读者批评指正。

王雷春
于湖北武汉



Contents

第1部分 教学篇

第1章 Python入门	1	2.2.1 整型	23
1.1 概述	1	2.2.2 浮点类型	23
1.1.1 Python简介	1	2.2.3 复数类型	24
1.1.2 Python的特点	2	2.2.4 布尔类型	25
1.1.3 Python的应用	2	2.2.5 数字类型转换	25
1.2 开发环境	3	2.2.6 分数和高精度实数	25
1.2.1 Python版本	3	2.2.7 常用数学函数	26
1.2.2 常用开发环境	3	2.3 字符串类型	27
1.2.3 Python下载和安装	4	2.3.1 字符串及创建	28
1.2.4 PyCharm下载和安装	5	2.3.2 字符串访问	29
1.2.5 第三方库安装	6	2.3.3 字符串运算	29
1.3 程序开发	8	2.3.4 字符串函数	30
1.3.1 程序运行方式	8	2.3.5 转义字符	34
1.3.2 使用Python自带工具开发 Python程序	8	2.4 常量和变量	34
1.3.3 使用PyCharm开发Python 程序	11	2.4.1 常量	34
1.4 程序结构和编码规范	12	2.4.2 变量	35
1.4.1 文件类型	12	2.5 运算符和表达式	38
1.4.2 代码结构	13	2.5.1 常用运算符和表达式	38
1.4.3 程序编码规范	14	2.5.2 运算符优先级	42
1.5 输入、输出函数	19	2.5.3 补充说明	43
1.5.1 input()函数	19	2.6 特殊内置函数	43
1.5.2 print()函数	19	2.6.1 内置函数简介	43
练习题1	21	2.6.2 特殊内置函数	43
第2章 Python基础	22	2.7 程序调试	46
2.1 数据类型	22	2.7.1 错误类型	46
2.2 数字类型	23	2.7.2 使用Python自带的Shell工具 进行程序调试	46
		2.7.3 在PyCharm中调试程序	48
		2.8 典型案例	48

2.8.1	计算复杂算术运算表达式的值	48	4.3.5	元组统计	75
2.8.2	求几何面、几何体的(表)面积 或体积	49	4.4	字典	76
2.8.3	解一元二次方程	49	4.4.1	字典创建	76
2.8.4	验证码验证	50	4.4.2	字典访问	77
练习题 2		50	4.4.3	字典复制和删除	79
第 3 章	程序设计结构	52	4.5	集合	79
3.1	概述	52	4.5.1	集合创建	79
3.2	顺序结构	52	4.5.2	集合访问	80
3.3	选择结构	52	4.5.3	集合复制和删除	81
3.3.1	单分支结构	52	4.5.4	集合运算	81
3.3.2	二分支结构	53	4.5.5	集合统计	82
3.3.3	多分支结构	54	4.6	嵌套组合数据	82
3.3.4	条件运算	55	4.7	典型案例	83
3.3.5	选择结构嵌套	56	4.7.1	查找	83
3.4	循环结构	56	4.7.2	排序	85
3.4.1	while 语句	56	4.7.3	推导式	87
3.4.2	for 语句	57	4.7.4	列表作为堆栈和队列使用	88
3.4.3	break、continue 和 pass 语句	59	4.7.5	基于组合数据的游戏角色 管理	89
3.4.4	循环结构的嵌套	60	练习题 4		91
3.5	典型案例	61	第 5 章	函数	93
3.5.1	计算部分级数和	61	5.1	函数定义和调用	93
3.5.2	使用选择结构计算员工工资	61	5.1.1	函数定义	93
3.5.3	用递推法求解实际问题	62	5.1.2	函数调用	94
3.5.4	“试凑法”解方程	63	5.2	函数参数	94
3.5.5	计算机猜数	63	5.2.1	参数传递	94
3.5.6	模拟自动饮料机	64	5.2.2	参数类型	95
练习题 3		65	5.2.3	参数传递的序列解包	97
第 4 章	组合数据	67	5.3	特殊函数	97
4.1	概述	67	5.3.1	匿名函数	97
4.2	列表	67	5.3.2	递归函数	98
4.2.1	列表创建	68	5.3.3	嵌套函数	99
4.2.2	列表访问	68	5.4	装饰器	100
4.2.3	列表复制和删除	70	5.4.1	装饰器的定义和调用	100
4.2.4	列表运算	71	5.4.2	带参数的装饰器	101
4.2.5	列表统计	72	5.5	变量作用域	102
4.2.6	列表查找与排序	72	5.6	典型案例	103
4.3	元组	73	5.6.1	加密和解密	103
4.3.1	元组创建	73	5.6.2	求最大公约数	104
4.3.2	元组访问	74	5.6.3	使用装饰器检查函数参数 合法性	106
4.3.3	元组复制和删除	75	5.6.4	模拟轮盘抽奖游戏	106
4.3.4	元组运算	75			

练习题 5	107	7.5.3 使用 Pandas 分析股票交易数据	154
第 6 章 面向对象程序设计	109	7.5.4 使用图像处理库处理和显示图像	155
6.1 概述	109	练习题 7	156
6.2 类与对象	110	第 8 章 正则表达式	158
6.2.1 类的定义	110	8.1 概述	158
6.2.2 对象创建和使用	111	8.2 正则表达式语法	159
6.3 类的成员	111	8.3 使用正则表达式模块处理字符串	161
6.3.1 成员类型	111	8.3.1 Re 模块中的常用函数	161
6.3.2 内置成员	113	8.3.2 常用正则表达式	164
6.4 类的方法	114	8.4 典型案例	167
6.4.1 类的方法类型	114	8.4.1 提取并汇总字符串中的费用	167
6.4.2 属性	115	8.4.2 校验字符串合法性	167
6.4.3 特殊方法	116	8.4.3 解析网页内容	169
6.5 类的继承与多态	117	练习题 8	170
6.5.1 类的继承	117	第 9 章 文件访问	172
6.5.2 类的多态	118	9.1 概述	172
6.6 抽象类和抽象方法	119	9.2 文本文件访问	172
6.7 典型案例——书籍出租管理系统	120	9.2.1 打开文件	173
练习题 6	123	9.2.2 文件操作	174
第 7 章 模块、包和库	124	9.3 二进制文件访问	178
7.1 概述	124	9.3.1 使用 Pickle 模块读/写二进制文件	178
7.2 常用标准库模块	124	9.3.2 使用 Struct 模块读/写二进制文件	179
7.2.1 Turtle 模块	125	9.3.3 使用 Marshal 模块读/写二进制文件	180
7.2.2 Random 模块	127	9.3.4 使用 Shelve 模块读/写二进制文件	181
7.2.3 Time 模块和 Datetime 模块	130	9.4 典型案例	182
7.2.4 Os 模块	132	9.4.1 合并文件	182
7.2.5 Sys 模块	134	9.4.2 CSV 文件操作	183
7.2.6 Timeit 模块	134	练习题 9	185
7.2.7 Zlib 模块	134	第 10 章 异常处理和单元测试	186
7.3 常用第三方库	135	10.1 异常类和异常处理	186
7.3.1 NumPy 库	135	10.1.1 异常和异常类	186
7.3.2 Pandas 库	140	10.1.2 异常处理	187
7.3.3 SciPy 库	142	10.1.3 断言语句	189
7.3.4 Matplotlib 库	143	10.2 单元测试	190
7.3.5 Jieba 库	146		
7.3.6 Pyinstaller 库	148		
7.4 自定义模块	149		
7.5 典型案例	151		
7.5.1 使用 Turtle 绘制表面填充正方体	151		
7.5.2 使用 NumPy 和 Matplotlib 分析股票	153		

10.3 典型案例	191	13.2.2 创建线程	247
10.3.1 自定义异常类	191	13.2.3 线程通信	250
10.3.2 自定义测试类	192	13.2.4 线程同步	251
练习题 10	194	13.3 典型案例	254
第 11 章 数据库访问	195	13.3.1 使用多进程导入/导出数据	254
11.1 概述	195	13.3.2 使用多线程模拟彩票发行	255
11.2 常用关系数据库访问	196	练习题 13	257
11.2.1 常用关系数据库简介	196	第 14 章 网络程序设计	258
11.2.2 Python DB-API 规范	198	14.1 网络协议	258
11.2.3 典型案例——访问关系数据库	199	14.1.1 互联网协议族	258
11.3 非关系数据库访问	207	14.1.2 TCP/IP	258
11.3.1 MongoDB	207	14.2 套接字(Socket)编程	259
11.3.2 典型案例——访问 MongoDB	209	14.2.1 套接字简介	259
练习题 11	212	14.2.2 基于 TCP 的套接字编程	260
第 12 章 图形用户界面编程	213	14.2.3 基于 UDP 的套接字编程	261
12.1 wxPython 库简介	213	14.3 Web 编程	263
12.2 事件处理	214	14.3.1 Web 编程概述	263
12.3 常用控件	215	14.3.2 Django	264
12.3.1 窗体	215	14.4 典型案例	271
12.3.2 按钮、标签和文本框	216	14.4.1 简单嗅探器	271
12.3.3 单选按钮、复选框	218	14.4.2 多线程端口扫描	272
12.3.4 列表框、组合框	219	14.4.3 用网络爬虫爬取全国城市天气信息	273
12.3.5 菜单	220	14.4.4 基于 Django 的个人博客	280
12.3.6 工具栏、状态栏	224	练习题 14	288
12.3.7 对话框	226	第 15 章 Python 与人工智能	290
12.4 布局	230	15.1 概述	290
12.4.1 布局及其类型	230	15.1.1 人工智能简介	290
12.4.2 wx.BoxSizer	231	15.1.2 Python 与人工智能	291
12.4.3 wx.GridSizer	232	15.2 机器学习	292
12.5 典型案例——专利管理系统	233	15.2.1 Sklearn	292
练习题 12	240	15.2.2 典型案例——使用线性回归模型训练、测试和预测数据	294
第 13 章 多进程与多线程	241	15.3 深度学习	296
13.1 Python 中的多进程	241	15.3.1 TensorFlow	297
13.1.1 进程的含义	241	15.3.2 典型案例——识别模糊的手写数字图片	299
13.1.2 创建进程	241	练习题 15	303
13.1.3 进程通信	243		
13.1.4 进程同步	245		
13.2 Python 中的多线程	247		
13.2.1 线程含义	247		

第2部分 实验篇

实验1 Python入门	305	二、实验内容	310
一、实验目的	305	实验9 文件访问	311
二、实验内容	305	一、实验目的	311
实验2 Python基础	305	二、实验内容	311
一、实验目的	305	实验10 异常处理和单元测试	311
二、实验内容	306	一、实验目的	311
实验3 程序设计结构	306	二、实验内容	312
一、实验目的	306	实验11 数据库访问	312
二、实验内容	306	一、实验目的	312
实验4 组合数据	307	二、实验内容	312
一、实验目的	307	实验12 图形用户界面编程	312
二、实验内容	307	一、实验目的	312
实验5 函数	308	二、实验内容	313
一、实验目的	308	实验13 多进程与多线程	313
二、实验内容	308	一、实验目的	313
实验6 面向对象程序设计	309	二、实验内容	313
一、实验目的	309	实验14 网络程序设计	313
二、实验内容	309	一、实验目的	313
实验7 模块、包和库	310	二、实验内容	314
一、实验目的	310	实验15 Python与人工智能	314
二、实验内容	310	一、实验目的	314
实验8 正则表达式	310	二、实验内容	314
一、实验目的	310		
参考文献	315		

第1部分 教学篇

第1章 Python入门

本章内容:

- 概述
- 开发环境
- 程序开发
- 程序结构和编码规范
- 输入、输出函数

1.1 概述

1.1.1 Python简介

Python 是一个高层次的结合了解释性、编译性、互动性和面向对象的脚本语言，支持命令式编程、函数式编程和面向对象程序设计，具有广泛的应用领域。Python 由荷兰人 Guido van Rossum 于 1989 年发明，第一个公开版本发行于 1991 年。

Python 已经成为最受欢迎的程序设计语言之一。自 2004 年以后，Python 的使用率呈线性增长。2011 年 1 月，Python 被 TIOBE 编程语言排行榜评为 2010 年度编程语言。2017 年 7 月 20 日，IEEE Spectrum 发布了第四届顶级编程语言交互排行榜，Python 高居首位(见图 1.1)。

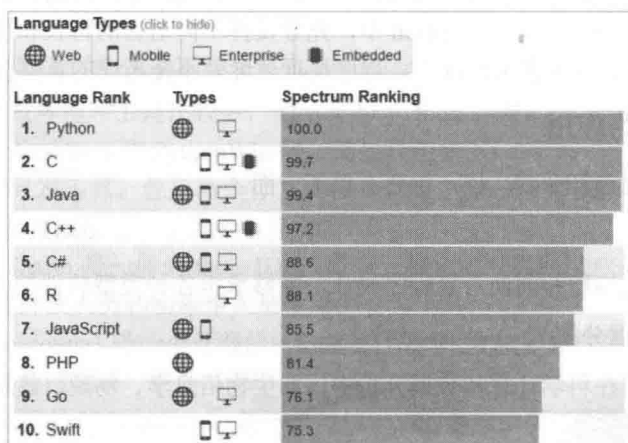


图 1.1 2017 年 IEEE Spectrum 第四届顶级编程语言交互排行榜

由于 Python 具有简洁性、易读性和可扩展性，在国外用 Python 做科学计算的研究机构日益增

多,一些世界著名大学已经采用 Python 来教授程序设计课程,如卡耐基梅隆大学的编程基础、麻省理工学院的计算机科学及编程导论就使用 Python 讲授。

众多开源的科学计算库都提供了 Python 的调用接口,如著名的计算机视觉库 OpenCV、三维可视化库 VTK、医学图像处理库 ITK 等。Python 专用的第三方库也很多,如科学计算库 NumPy、SciPy、Pandas 分别提供了快速数组处理、科学计算和数据分析功能。

因此,Python 及其众多的第三方库构成的开发环境十分适合工程技术人员、科研人员进行数据处理、分析等,或者开发科学计算应用程序。

1.1.2 Python 的特点

1. Python 的优势

与其他编程语言相比较,Python 具有如下优势:

(1)简单、易学。Python 有相对较少的关键字,其结构简单,语法定义明确,学习起来非常快捷,从而能让学习者更多地关注解决问题的方法和程序本身的算法、逻辑。

(2)可移植性。Python 源代码开放,在一个平台(如 Windows)开发的 Python 程序几乎可以不加修改地运行在其他平台上,如 Linux、FreeBSD、Macintosh、Solaris 等。

(3)解释性。Python 程序不需要编译成二进制代码,可以直接从源文件运行。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执行。

(4)面向对象。Python 既支持面向过程程序设计也支持面向对象程序设计,完全支持类的继承、重载和派生等。

(5)可扩展性和可嵌入性。Python 支持 C/C++接口,可以方便地嵌入由 C/C++编写的程序。同时,也可以把 Python 代码嵌入 C/C++程序中,从而提供脚本功能。

(6)丰富的库和模块。Python 拥有能够处理各种任务的标准库模块,如数学函数模块 Math、正则表达式模块 Re、随机数生成模块 Random、图形界面编程模块 Tkinter 等。而且,Python 还支持很多功能强大的第三方库,如 NumPy(科学计算)、Pandas(数据分析)和 Matplotlib(数据可视化)等。

2. Python 的局限性

Python 虽然是一个非常成功的语言,其开发效率很高,但也有其不足和局限性。和 C、C++等程序设计语言相比,Python 程序运行速度较慢。因此对于这类程序,可以将运行速度要求较高的部分使用 C、C++等编写,再将其嵌入 Python 中,充分发挥不同语言的特长和优势。不过,现在计算机硬件的配置不断提高,在大多数情况下,程序运行速度并不是考虑的首要问题。

1.1.3 Python 的应用

Python 的应用领域已经非常广泛,主要有以下方面。

1. 常规软件开发

Python 支持函数式编程和面向对象程序设计,能够承担各种类型软件的开发工作。

2. 科学计算与数据分析

Python 被广泛运用在科学计算和数据分析中,如生物信息学、物理、建筑、地理信息系统、图像可视化分析、生命科学等。

3. 网络爬虫

网络爬虫也称为网络蜘蛛,是大数据行业获取数据的核心工具。能够编写网络爬虫的编程语言不少,但 Python 绝对是其中的主流语言,基于 Python 的 Scrapy 是使用最多的爬虫框架之一。

4. Web 应用开发

Python 具有一些优秀的 Web 框架, 如 Django、Flask 等。很多大型网站使用基于 Python 的 Web 框架开发, 如 Youtube、Dropbox、豆瓣等。

5. 系统网络运维

在运维的工作中, 有大量重复性工作, 并需要采用管理系统、监控系统、发布系统等实现工作自动化, 提高工作效率。在这样的场景, Python 是一门非常合适的语言。

6. 人工智能与机器学习

由于 Python 具有动态和良好的性能; 现在几个非常有影响力的人工智能框架(如 TensorFlow、PyTorch 等)都提供了 Python 支持, 一些机器学习方向、深度学习方向和自然语言处理方向的网站基本都是通过 Python 实现的。

1.2 开发环境

1.2.1 Python 版本

Python 目前有两个不同序列的版本: Python 2.X 和 Python 3.X。Python 2.X 最早的版本发布于 2000 年年底, 最新版本为 Python 2.7。Python 3.X 的最新版本为 Python 3.7.3。

Python 2.X 和 Python 3.X 两个序列的版本之间有很多用法是不兼容的, 除基本的输入、输出方式有所不同外, 很多内置函数和标准库模块用法也有较大的差别, 适用于这两个版本的第三方库的差别更大。

总体来说, Python 3.X 的设计理念更加合理、高效和人性化, 一些第三方库也不断推出与 Python 3.X 相适应的新版本。而且, Python 开发团队已经重申了终止对 Python 2.X 的支持。

考虑到 Windows 系统的使用者众多和 Python 开发团队对 Python 3.X 的支持, 本书所有程序均基于 Windows 平台下的 Python 3.X 版本。

在本书后续内容中, 若无特别声明, 则 Python 均指 Python 3.X。

1.2.2 常用开发环境

学习 Python 程序设计, 首先要搭建开发环境, 然后才能编辑、测试和运行 Python 程序, 检验所学的 Python 知识。

Python 是一种脚本语言, 它并没有提供一种官方的开发环境, 需要用户自主选择开发工具。目前, 支持 Python 的开发环境和工具比较多, 如常见的 Python 自带的开发环境和工具、PyCharm、Anaconda3、Vim、Sublime Text 和 PythonWin 等, 简要介绍如下:

(1) Python 自带的开发工具。包括命令行方式和一个纯 Python 的集成开发环境 (IDE)。当安装好 Python 以后, 就可以直接使用, 不需要另外安装。

(2) PyCharm。PyCharm 是一种 Python IDE, 带有一整套可以帮助用户在使用 Python 开发时提高效率的工具, 专业开发人员和刚起步人员使用的有力工具。

(3) Anaconda3。Anaconda3 是一个开源的 Python 发行版本, 专注于数据分析, 包含 Conda、Python 等 180 多个科学包及其依赖项。

(4) Vim。Vim 的全称是 Vi Improved, 是一个类似于 Vi 的著名的功能强大、高度可定制的文本编辑器, 在 Vi 的基础上改进和增加了很多特性。Vim 支持包括 Python 的多种版本。

(5) Sublime Text。Sublime Text 支持多种编程语言的语法高亮、拥有优秀的代码自动完成功能, 还拥有代码片段 (Snippet) 的功能, 可以将常用的代码片段保存起来, 在需要时随时调用。

(6) PythonWin。PythonWin 是一个 Python 集成开发环境，在许多方面都比 IDLE（纯 Python 下自带的简洁的集成开发环境）优秀，这个工具是针对 Win32 用户的。

本书仅介绍 Windows 环境下的 Python 自带开发工具和 PyCharm 的下载、安装、使用。

1.2.3 Python 下载和安装

1. Python 下载

Python 可从网址为 <https://www.python.org/downloads/windows/> 的站点选择相应的版本下载。本书案例对应下载的是支持 64 位 Windows 的 Python 3.7.2 安装文件 python-3.7.2-amd64.exe。

2. Python 安装

Python 安装过程如下：

(1) 双击下载的 Python 3.7.2 安装文件 python-3.7.2-amd64.exe，进入“打开文件”对话框。

(2) 在该对话框中单击“运行”按钮(见图 1.2)，进入 Python 安装位置和 Path 设置对话框。

(3) 在该对话框中选择勾选“Add Python 3.7 to PATH”复选框，然后单击“Install Now”（立即安装）或“Customize installation”（自定义安装），开始安装(见图 1.3)。

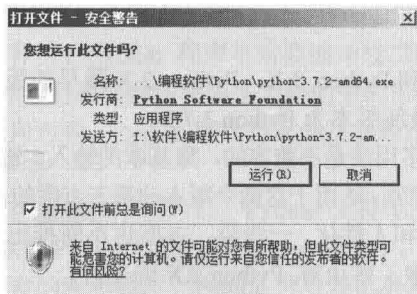


图 1.2 “打开文件”对话框



图 1.3 设置安装选项

(4) 单击“Close”按钮，完成安装(见图 1.4)。

3. Python 运行环境测试

完成 Python 安装后，打开 Windows 命令行，在命令行界面中输入“Python”后回车，测试结果见图 1.5，则说明安装成功。如果需要退出 Python 运行环境，则使用 quit() 函数或 exit() 函数。

在 Python 的 IDLE 中使用 exit() 函数或 quit() 函数，相当于关闭 IDLE 程序(见图 1.6)。

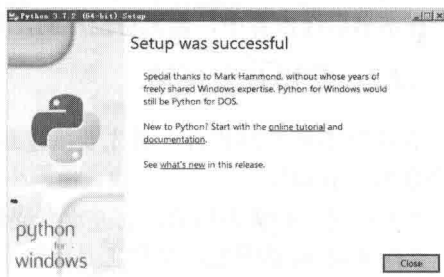


图 1.4 完成安装界面

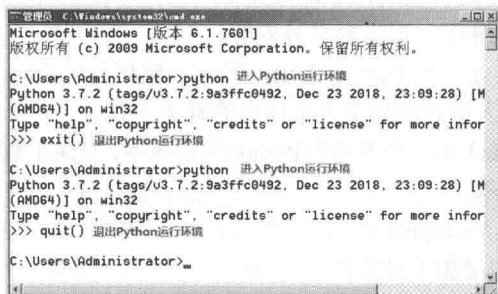


图 1.5 Python 测试结果

4. Web 应用开发

Python 具有一些优秀的 Web 框架, 如 Django、Flask 等。很多大型网站使用基于 Python 的 Web 框架开发, 如 Youtube、Dropbox、豆瓣等。

5. 系统网络运维

在运维的工作中, 有大量重复性工作, 并需要采用管理系统、监控系统、发布系统等实现工作自动化, 提高工作效率。在这样的场景, Python 是一门非常合适的语言。

6. 人工智能与机器学习

由于 Python 具有动态和良好的性能; 现在几个非常有影响力的人工智能框架(如 TensorFlow、PyTorch 等)都提供了 Python 支持, 一些机器学习方向、深度学习方向和自然语言处理方向的网站基本都是通过 Python 实现的。

1.2 开发环境

1.2.1 Python 版本

Python 目前有两个不同序列的版本: Python 2.X 和 Python 3.X。Python 2.X 最早的版本发布于 2000 年年底, 最新版本为 Python 2.7。Python 3.X 的最新版本为 Python 3.7.3。

Python 2.X 和 Python 3.X 两个序列的版本之间有很多用法是不兼容的, 除基本的输入、输出方式有所不同外, 很多内置函数和标准库模块用法也有较大的差别, 适用于这两个版本的第三方库的差别更大。

总体来说, Python 3.X 的设计理念更加合理、高效和人性化, 一些第三方库也不断推出与 Python 3.X 相适应的新版本。而且, Python 开发团队已经重申了终止对 Python 2.X 的支持。

考虑到 Windows 系统的使用者众多和 Python 开发团队对 Python 3.X 的支持, 本书所有程序均基于 Windows 平台下的 Python 3.X 版本。

在本书后续内容中, 若无特别声明, 则 Python 均指 Python 3.X。

1.2.2 常用开发环境

学习 Python 程序设计, 首先要搭建开发环境, 然后才能编辑、测试和运行 Python 程序, 检验所学的 Python 知识。

Python 是一种脚本语言, 它并没有提供一种官方的开发环境, 需要用户自主选择开发工具。目前, 支持 Python 的开发环境和工具比较多, 如常见的 Python 自带的开发环境和工具、PyCharm、Anaconda3、Vim、Sublime Text 和 PythonWin 等, 简要介绍如下:

(1) Python 自带的开发工具。包括命令行方式和一个纯 Python 的集成开发环境 (IDE)。当安装好 Python 以后, 就可以直接使用, 不需要另外安装。

(2) PyCharm。PyCharm 是一种 Python IDE, 带有一整套可以帮助用户在使用 Python 开发时提高效率的工具, 专业开发人员和刚起步人员使用的有力工具。

(3) Anaconda3。Anaconda3 是一个开源的 Python 发行版本, 专注于数据分析, 包含 Conda、Python 等 180 多个科学包及其依赖项。

(4) Vim。Vim 的全称是 Vi Improved, 是一个类似于 Vi 的著名的功能强大、高度可定制的文本编辑器, 在 Vi 的基础上改进和增加了很多特性。Vim 支持包括 Python 的多种版本。

(5) Sublime Text。Sublime Text 支持多种编程语言的语法高亮、拥有优秀的代码自动完成功能, 还拥有代码片段 (Snippet) 的功能, 可以将常用的代码片段保存起来, 在需要时随时调用。

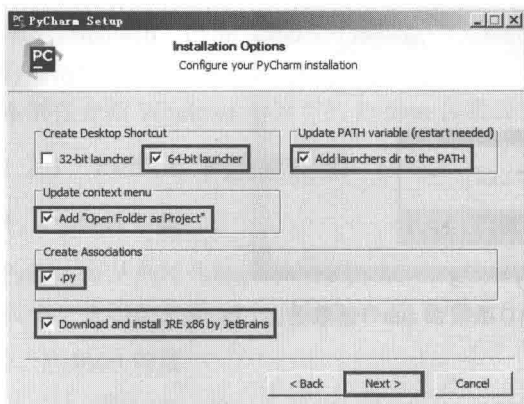


图 1.9 安装选项设置对话框

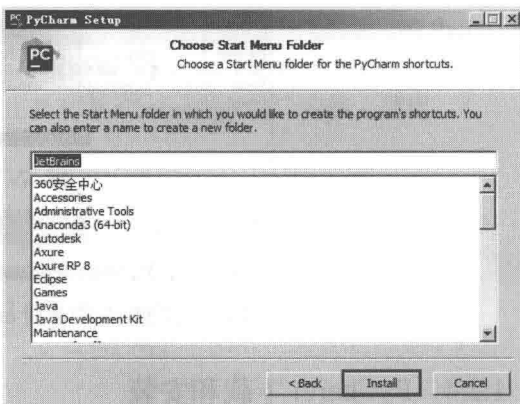


图 1.10 开始安装软件

(6) 单击“Finish”按钮，完成 PyCharm 安装(见图 1.11)。

PyCharm 安装完成后，一般只有 1 个月的试用期。如果需要长期使用，则需要使用激活码激活。

4. PyCharm 开发界面

PyCharm 开发界面(见图 1.12)主要包括菜单栏、工具栏、项目管理窗口、代码编辑窗口和结果输出窗口(显示运行结果)等。

6

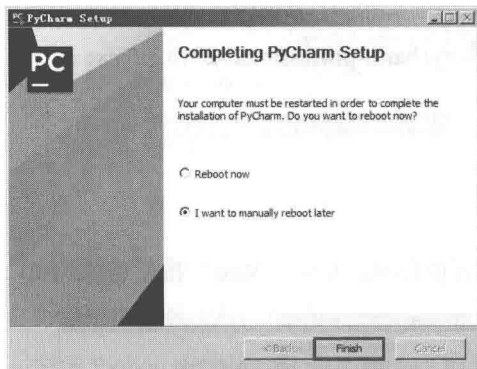


图 1.11 PyCharm 安装完成界面

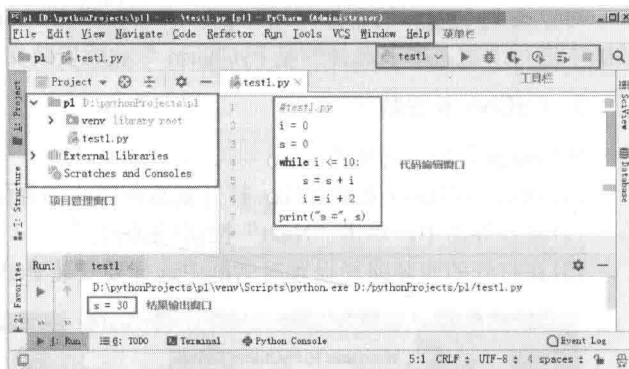


图 1.12 PyCharm 开发界面

1.2.5 第三方库安装

Python 能够在很短的时间内快速发展和广泛使用的原因之一是，Python 支持和拥有类型众多、功能强大的第三方库，如 NumPy、SciPy、Pandas、Matplotlib 等。下面以 NumPy 为例介绍第三方库的下载和安装方法。

1. 使用 pip 命令安装 NumPy

(1) 使用 pip 命令直接安装 NumPy

进入命令行界面，输入命令 `pip install numpy` 安装 NumPy。这种方式简单，但安装时间较长。

(2) 使用 pip 命令运行 NumPy 安装文件

进入 NumPy 官方网站 <https://pypi.org/project/numpy/#files>，选择与操作系统和 Python 版本匹配的 NumPy 库进行下载。本书下载的是与 64 位 Windows 7、Python 3.7 匹配的安装文件 `numpy-1.16.2-cp37-cp37m-win_amd64.whl`(见图 1.13)。

numpy-1.16.2-cp37-cp37m-win32.whl (10.0 MB)	Wheel	cp37	Feb 27, 2019
numpy-1.16.2-cp37-cp37m-win_amd64.whl (11.9 MB) 	Wheel	cp37	Feb 27, 2019
numpy-1.16.2.zip (5.1 MB)	Source	None	Feb 27, 2019

图 1.13 选择和下载 NumPy 安装文件

打开命令行界面，进入 NumPy 安装文件所在路径，运行命令 `pip install numpy-1.16.2-cp37-cp37m-win_amd64.whl` 安装 NumPy (见图 1.14)。

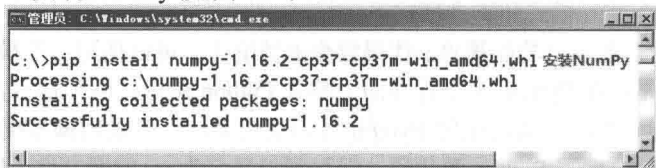


图 1.14 安装 NumPy

提示安装成功后即可在 Python 程序中导入后使用。

2. 在 PyCharm 中下载和安装 NumPy

在 PyCharm 中下载和安装 NumPy 相对简单，具体步骤如下：

(1) 在 PyCharm 界面中选择“File”→“Settings”菜单项 (见图 1.15)。

(2) 选择“Project”→“Project Interpreter”，打开项目解释器窗口，选择“+”，打开安装包对话框 (见图 1.16)。

(3) 在左上角的搜索栏中输入“numpy”后搜索，在下方选中“numpy”，单击底部的“Install Package”按钮安装 NumPy (见图 1.17)。

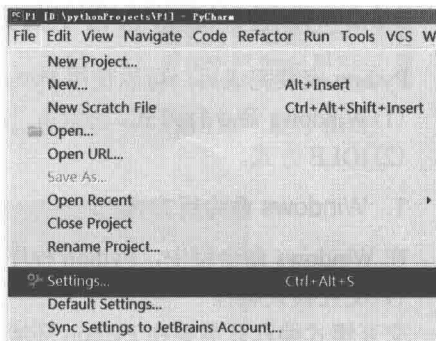


图 1.15 选择菜单项

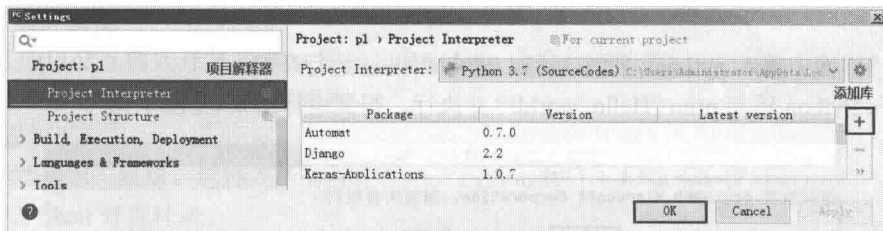


图 1.16 安装包对话框

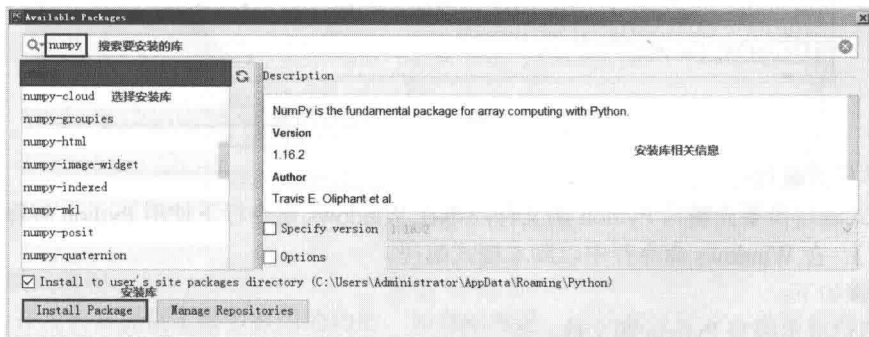


图 1.17 安装 NumPy