

高等学校计算机类特色教材

上海市高等学校信息技术水平考试参考教材

Python 程序设计基础

(第3版)

主 编 李东方

文欣秀 张向东
副主编
常 珊 郑 奋

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书是上海市高等学校信息技术水平考试（二三级）Python 程序设计考试科目的参考教材，并在教学内容和要求上兼容全国计算机等级考试二级 Python 语言程序设计考试大纲。本书面向初学者，通过大量实例由浅入深、循序渐进地讲述 Python 程序设计的基本概念和基本方法，内容包括 Python 语言及其编程环境、Python 的基本语法、Python 程序的基本流程控制、Python 的组合数据类型、文件与基于文本文件的数据分析、函数、面向对象的程序设计与 Python 生态、图形化界面设计、图形绘制与数据可视化、正则表达式与简单爬虫、数据库操作和基于第三方库的应用举例。本书的例题源代码、电子课件、习题素材等资源均可扫码下载，也可登录华信教育资源网（www.hxedu.com.cn）注册后免费下载。

本书适合作为文、理、工、农、医、法、商、管等非计算机专业高等学校通识教育阶段计算机程序设计课程教材，也可供 Python 语言爱好者自学以及计算机科学与技术相关专业人员参考。

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 程序设计基础 / 李东方主编. —3 版. —北京：电子工业出版社，2023.6
ISBN 978-7-121-45627-5

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

中国国家版本馆 CIP 数据核字（2023）第 089886 号

责任编辑：冉 哲

印 刷：

装 订：

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：16 字数：409.6 千字

版 次：2017 年 1 月第 1 版

2023 年 6 月第 3 版

印 次：2023 年 6 月第 1 次印刷

定 价：49.00 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：ran@phei.com.cn。

前 言

程序设计基础是高等学校计算机基础教学的核心课程。通常,选择一门高级程序设计语言作为教学语言,并以此贯彻程序设计的基本思想方法,培养学生的计算思维,为他们打下理解需求、求解问题、编程实现的坚实基础。

在上海市教育委员会高教处、上海市教育考试院、上海市高等学校信息技术水平考试委员会和上海市计算机基础教育协会的组织领导下,上海各高等学校计算机基础教育工作者团结协作,经过长期的探索和实践,确立了“夯实基础、面向环境、培养创新”的计算机基础教育培养目标,构建了包括分类分层次的课程体系、自主学习环境、信息技术水平考试、创新能力培养等内容的多平台综合教育环境,取得了丰硕的成果。2016年,原上海市高等学校计算机等级考试率先增考了二级 Python 程序设计科目。本书是上海市高等学校信息技术水平考试(二三级)Python 程序设计考试科目的参考教材,并在教学内容和要求上兼容全国计算机等级考试二级 Python 语言程序设计考试大纲。

Python 语言是一种解释运行、面向对象、扩展性强的程序设计语言,是学生学习编程、理解用计算机解决问题的方法的有效工具。通过对 Python 语言的学习,学生应能掌握其基本语法和基本编程方法,理解程序设计中的计算思维,并能上机调试、运行程序,解决简单的实际问题。

本书面向初学者,由浅入深、循序渐进地讲述 Python 程序设计的基本概念和基本方法。本书由海军军医大学、华东理工大学、复旦大学、东华大学等学校常年工作在计算机基础教学第一线、具有丰富教学经验的教师集体编写,力图简明实用、条理分明。本书通过大量实例进行讲解,不拘泥于语法细节,避免曲折烦琐,同时力图体现 Python 语言追求优雅、明确、简单的风格。

全书共 12 章,内容包括 Python 语言及其编程环境、Python 的基本语法、Python 程序的基本流程控制、Python 的组合数据类型、文件与基于文本文件的数据分析、函数、面向对象的程序设计与 Python 生态、图形化界面设计、图形绘制与数据可视化、正则表达式与简单爬虫、数据库操作和基于第三方库的应用举例。每章均配有教学目标和习题,书后附有上海市高等学校信息技术水平考试(二三级)Python 程序设计考试大纲和全国计算机等级考试二级 Python 语言程序设计考试大纲。全书采用 Python 3.x 版本。本书的例题源代码、电子课件、习题素材等资源均可扫码下载,也可登录华信教育资源网(www.hxedu.com.cn)注册后免费下载。

本书由李东方(第 1、5~10、12 章)、张向东(第 2 章)、文欣秀(第 3 章)、常姗(第 4 章)、郑奋(第 11 章)编写,文欣秀为各章编配了习题,全书由李东方统稿。本书在编写过程中还得到了同济大学、上海大学、华东师范大学、华东政法大学、上海对外经贸大学等学校相关教师的指导与支持。本书部分实例采用了上海市高等学校信息技术水平考试既往试题中的部分素材和网上佚名素材,在此一并表示诚挚的感谢。

由于时间仓促和水平有限,书中难免存在不妥之处,竭诚欢迎读者提出宝贵意见。作者联系邮箱:dfli@smmu.edu.cn。

作 者

教学建议

建议 48~64 学时，其中 16~32 学时为实验教学。建议采用机房“现场授课”方式，边讲边练，以提高教学效率。

教 学 内 容	64 学时教学分配			48 学时教学分配		
	课堂教学	实验教学	课外作业	课堂教学	实验教学	课外作业
第 1 章 Python 语言及其编程环境	2		1	2		1
第 2 章 Python 的基本语法	2	2	1	2	2	1
第 3 章 Python 程序的基本流程控制	4	4	2	4	4	2
第 4 章 Python 的组合数据类型	6	6	2	6	6	2
第 5 章 文件与基于文本文件的数据分析	4	4	1	4	4	1
第 6 章 函数	2	2	1	2	2	1
第 7 章 面向对象的程序设计与 Python 生态	2	2	1	2		1
第 8 章 图形化界面设计	4	4	3	2	2	2
第 9 章 图形绘制与数据可视化	2	2	2	2	2	2
第 10 章 正则表达式与简单爬虫	2	2	1			
第 11 章 数据库操作	2	2	1			
第 12 章 基于第三方库的应用举例		2	1			
合 计	64		17	48		13

教学软件环境：Python 3.x，建议 3.7 以上或 Anaconda 相应版本，可选装 PyCharm、VS Code、PyScripter、Wing IDE、Spyder、Thonny 等编程调试环境。

目 录

第 1 章 Python 语言及其编程环境	(1)	3.3.2 if-elif-else 语句	(43)
1.1 Python 语言概述	(1)	3.3.3 分支语句嵌套	(46)
1.2 Python 的安装	(3)	3.4 循环结构	(47)
1.2.1 Python 的官网下载和安装	(3)	3.4.1 while 语句	(47)
1.2.2 Anaconda 集成开发环境	(4)	3.4.2 for 语句和内置函数 range()	(49)
1.3 Python 程序设计步骤	(5)	3.4.3 循环语句嵌套	(51)
1.4 常用的 Python 第三方编辑器	(7)	3.4.4 转移和中断语句	(53)
习题 1	(11)	3.5 程序调试	(55)
第 2 章 Python 的基本语法	(12)	3.5.1 语法错误与逻辑错误	(55)
2.1 Python 程序的书写格式与基本 规则	(12)	3.5.2 常见语法错误	(55)
2.1.1 基本词法单位	(12)	3.5.3 排查程序错误的方法	(59)
2.1.2 程序的书写格式与基本 规则	(14)	3.5.4 程序运行中 try-except 异常 处理	(61)
2.2 Python 的基本数据类型	(18)	习题 3	(62)
2.2.1 数值类型	(18)	第 4 章 Python 的组合数据类型	(64)
2.2.2 字符串类型	(21)	4.1 序列型组合数据	(64)
2.3 Python 的基本运算和表达式	(25)	4.1.1 字符串	(64)
2.3.1 变量的操作	(25)	4.1.2 列表	(70)
2.3.2 运算符	(29)	4.1.3 元组	(78)
2.3.3 算术运算	(30)	4.2 映射型组合数据——字典	(85)
2.3.4 输入与输出语句	(32)	4.2.1 创建字典	(85)
2.3.5 赋值、关系和逻辑运算	(33)	4.2.2 访问字典中的值	(86)
2.3.6 表达式	(36)	4.2.3 字典的更新与合并	(87)
2.3.7 赋值语句	(37)	4.2.4 字典的操作	(87)
习题 2	(39)	4.3 集合型组合数据——集合	(91)
第 3 章 Python 程序的基本流程 控制	(40)	4.3.1 集合的创建	(91)
3.1 计算思维和程序设计基本方法	(40)	4.3.2 集合的修改	(93)
3.1.1 计算思维	(40)	4.3.3 集合的数学运算	(95)
3.1.2 程序设计基本方法	(41)	习题 4	(96)
3.2 顺序结构	(41)	第 5 章 文件与基于文本文件的数据 分析	(99)
3.3 分支结构	(42)	5.1 文件的基本概念	(99)
3.3.1 if 语句	(43)	5.1.1 文件与访问路径	(99)

5.1.2 文件与编码·····	(100)	7.2.3 用函数表示类的行为 特征·····	(135)
5.2 文件的操作·····	(101)	7.3 面向对象的特征·····	(135)
5.2.1 文件的打开和关闭·····	(102)	7.3.1 封装·····	(135)
5.2.2 文本文件的读取、写入和 追加写入·····	(103)	7.3.2 继承·····	(136)
5.3 基于文本文件的数据分析·····	(106)	7.3.3 多态·····	(137)
5.3.1 对文本文件的读取和 遍历·····	(106)	7.4 Python 程序的组织和管理·····	(138)
5.3.2 词频分析·····	(108)	7.4.1 程序和模块结构·····	(138)
5.3.3 第三方库 jieba 和中文词频 分析·····	(110)	7.4.2 包和库·····	(139)
5.3.4 第三方库 wordcloud 和词语 可视化·····	(111)	7.5 Python 的生态·····	(141)
习题 5·····	(112)	7.5.1 第三方库的安装·····	(141)
第 6 章 函数 ·····	(114)	7.5.2 库与模块的导入·····	(144)
6.1 函数的定义与调用·····	(114)	7.6 可执行文件与 pyinstaller 库·····	(146)
6.2 参数的传递·····	(115)	习题 7·····	(147)
6.2.1 关键字参数和默认 参数·····	(115)	第 8 章 图形化界面设计 ·····	(148)
6.2.2 位置参数·····	(116)	8.1 窗体控件布局·····	(148)
6.2.3 可变参数·····	(116)	8.1.1 根窗体·····	(148)
6.2.4 变量的作用域·····	(118)	8.1.2 tkinter 常用控件·····	(149)
6.3 匿名函数·····	(119)	8.1.3 控件布局·····	(153)
6.4 高阶函数·····	(120)	8.2 tkinter 常用控件的特有属性·····	(156)
6.4.1 map()函数·····	(120)	8.2.1 文本输入/输出相关控件·····	(156)
6.4.2 filter()函数·····	(121)	8.2.2 按钮控件·····	(159)
6.4.3 reduce()函数·····	(121)	8.2.3 单选按钮控件·····	(161)
6.5 递归·····	(122)	8.2.4 复选框控件·····	(162)
6.6 函数的高级应用·····	(125)	8.2.5 列表框与组合框控件·····	(166)
6.6.1 生成器·····	(125)	8.2.6 滑块控件·····	(171)
6.6.2 装饰器与闭包·····	(126)	8.2.7 菜单控件·····	(173)
习题 6·····	(127)	8.2.8 窗体控件·····	(175)
第 7 章 面向对象的程序设计与 Python 生态 ·····	(131)	8.2.9 模态对话框·····	(176)
7.1 面向对象的概念·····	(131)	8.3 事件响应·····	(180)
7.2 类与实例·····	(132)	习题 8·····	(182)
7.2.1 创建类和子类·····	(132)	第 9 章 图形绘制与数据可视化 ·····	(185)
7.2.2 增加属于类的实例·····	(133)	9.1 tkinter 库的 Canvas 图形绘制 方法·····	(185)
		9.1.1 Canvas 基本绘图方法·····	(186)
		9.1.2 绘制函数图形·····	(189)
		9.2 turtle 库的图形绘制方法·····	(192)
		9.2.1 turtle 基本绘图方法·····	(193)

9.2.2 turtle 绘图举例·····	(194)	11.1.2 游标对象和 SQL 查询···	(222)
9.3 matplotlib 库的图形绘制方法···	(198)	11.2 访问 Access、MySQL 和 SQL Server 数据库·····	(223)
9.3.1 环境安装和基本绘图 方法·····	(199)	习题 11·····	(225)
9.3.2 绘制二维函数图形·····	(201)	第 12 章 基于第三方库的应用	
习题 9·····	(205)	举例 ·····	(228)
第 10 章 正则表达式与简单爬虫 ···	(208)	12.1 对 Excel 文件的操作·····	(228)
10.1 正则表达式·····	(208)	12.1.1 用 openpyxl 在内存中创建 工作簿·····	(228)
10.1.1 正则表达式的构成·····	(209)	12.1.2 数据的读取·····	(229)
10.1.2 贪婪匹配模式与非贪婪 匹配模式·····	(210)	12.1.3 编辑和保存·····	(231)
10.2 re 库的内置函数·····	(211)	12.2 图像操作·····	(233)
10.2.1 匹配与搜索·····	(211)	12.2.1 Image 模块及其应用···	(233)
10.2.2 切分与分组·····	(213)	12.2.2 ImageChops 特效与 合成·····	(236)
10.2.3 替换·····	(214)	习题 12·····	(238)
10.3 正则表达式的应用：简单爬虫···	(214)	附录 A 上海市高等学校信息技术水平 考试（二三级）Python 程序 设计考试大纲 ·····	(239)
10.3.1 抓取新闻链接和标题···	(214)	附录 B 全国计算机等级考试二级 Python 语言程序设计考试大纲 ·····	(245)
10.3.2 抓取新闻图片素材·····	(216)	参考文献 ·····	(248)
习题 10·····	(217)		
第 11 章 数据库操作 ·····	(218)		
11.1 访问 SQLite 数据库·····	(218)		
11.1.1 SQLite 数据库连接对象及表 的 SQL 操作·····	(219)		

Python语言及其编程环境

本章教学目标:

- 了解 Python 语言的特点。
- 学会 Python 编程环境的安装。
- 逐步熟悉使用一种第三方 Python 编辑器。

1.1 Python 语言概述

Python 是一种解释运行、面向对象的程序设计语言，由 Guido van Rossum 于 1989 年发明，并于 1991 年公开了第一个发行版本。

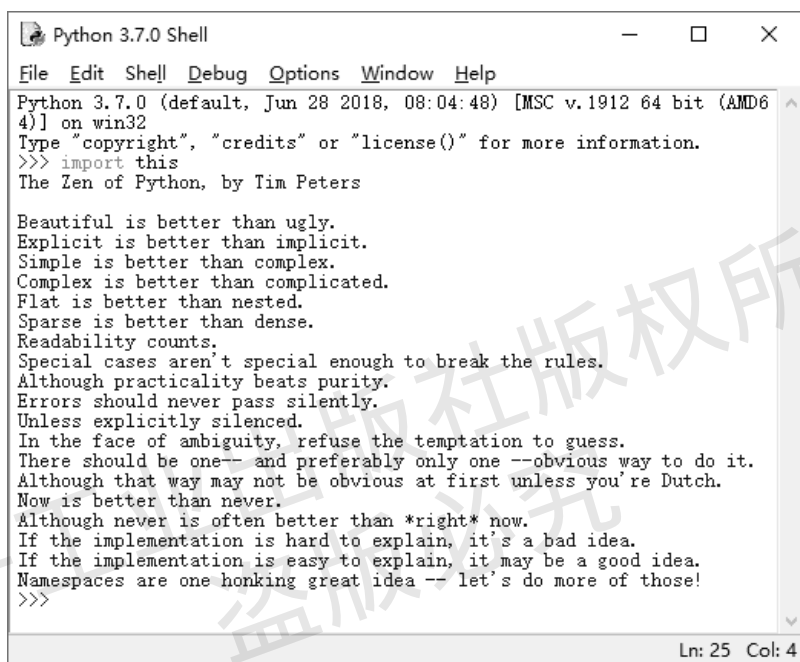
由于 Python 语言简洁、优雅，开发效率高，使用 Python 语言既能快速地生成程序的原型，又能方便地封装成可调用的扩展类库，程序无须修改就能在 Windows、Linux、UNIX 和 macOS 等操作系统上跨平台使用，因此 Python 语言常被用于网站开发、网络编程、图形处理、黑客攻防等，已迅速上升为第一大广泛应用的编程语言。图 1-1 为 2023 年 4 月 TIOBE 编程语言排行榜部分语言排名。

Apr 2023	Apr 2022	Change	Programming Language	Ratings	Change
1	1		 Python	14.51%	+0.59%
2	2		 C	14.41%	+1.71%
3	3		 Java	13.23%	+2.41%
4	4		 C++	12.96%	+4.68%
5	5		 C#	8.21%	+1.39%
6	6		 Visual Basic	4.40%	-1.00%
7	7		 JavaScript	2.10%	-0.31%
8	9	^	 SQL	1.68%	-0.61%
9	10	^	 PHP	1.36%	-0.28%
10	13	^	 Go	1.28%	+0.20%

图 1-1 2023 年 4 月 TIOBE 编程语言排行榜部分语言排名

Python 语言具有丰富和强大的类库，能够把用其他语言（如 C/C++）编写的各种模块很轻松地联结在一起，因此被昵称为胶水语言。Python 开发环境是纯粹的自由软件，源代码和解释器 CPython 均遵循 GPL（General Public License）协议。

Python 语言崇尚优雅、明确、简单。在其命令行运行环境中输入“import this”，就会呈现出 Tim Peters 编写的、被业界称为“Python 之禅”的编程格言，如图 1-2 所示。这些格言逐渐成为 Python 程序开发者追求“More Pythonic（更具有 Python 风格）”的指导思想。



```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (default, Jun 28 2018, 08:04:48) [MSC v.1912 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
>>>
```

图 1-2 Python 之禅

Python 之禅的中文意译如下：

- 优美胜于丑陋，显式胜于隐式。
- 简洁胜于复合，复合胜于晦涩。
- 扁平胜于嵌套，稀疏胜于密集。
- 可读性更重要。
- 虽然实用性胜于纯粹性，但特殊情况并不足以打破规则。
- 错误永远不应被悄悄避开，除非被明确表明要忽略。
- 面对歧义时，拒绝猜测。
- 应该有一种——最好只有一种——明确的解决方案。
- 解决方案最初可能并不是明确的，除非你就是那个荷兰人（指 Guido）。
- 马上做胜于永远不做，尽管有时立即盲目动手做还不如不做。
- 难以释义的实现通常是糟糕的方案。
- 容易释义的实现才是好方案。
- 利用命名空间是一个很棒的主意——让我们做更多这样的事情吧！

1.2 Python 的安装

Python 开发环境是完全免费的自由软件，下载前应考虑如下问题：

- **支持的操作系统。**Python 支持 Windows、Linux、UNIX 和 macOS 等不同的操作系统，应选择对应的安装程序。
- **操作系统的字长。**应根据操作系统的字长（32 位或 64 位）选择对应的安装程序，以获得最佳的运行环境。
- **Python 的版本。**Python 2.x 至 2.7 版后不再升级，3.x 版与 2.x 版不完全兼容，大批用 2.x 版编写的库函数无法在 3.x 版下直接使用。2015 年以后，绝大多数用 Python 编写的库函数都可以稳定、高效地在 3.x 版下运行。因此，除需要继承 2.x 版才能稳定运行的特殊应用外，均应使用 3.x 版。

目前，可通过其官网下载并安装 Python，也可以直接安装 Anaconda 集成环境。

1.2.1 Python 的官网下载和安装

最新版本的安装程序可从 Python 官网免费下载，如图 1-3 所示。

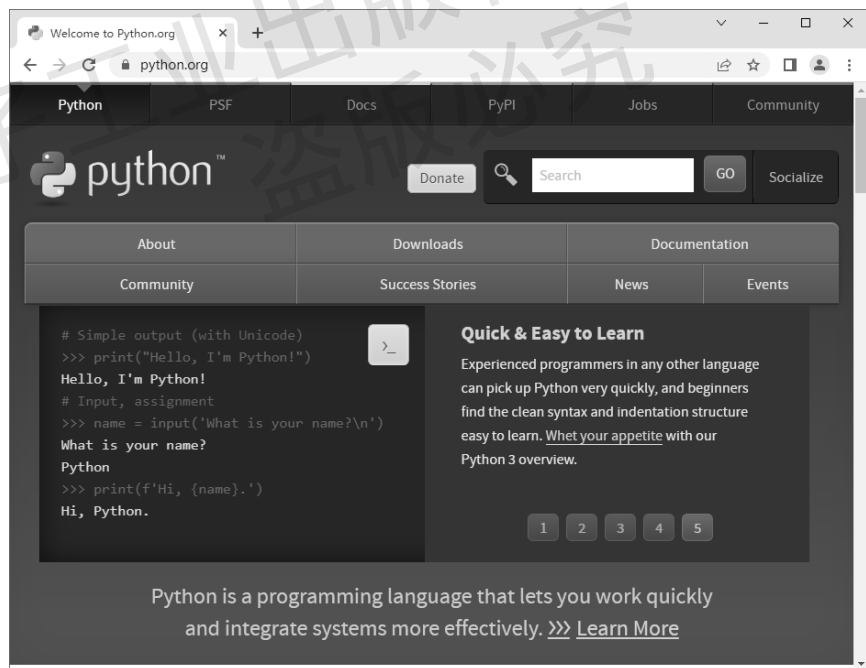


图 1-3 官网下载页面

在官网下载页面中可选择操作系统类型和字长，以及安装包的形式。这里以 Windows 操作系统可执行程序安装为例：双击下载的 python-3.x.x.exe 文件（版本不同，文件名会有所不同），即可按向导提示进行安装。例如，Python 3.7.3（64 位）安装向导如图 1-4 所示。



图 1-4 Python 3.7.3（64 位）安装向导

为使以后操作系统中任意路径上的 Python 程序都能正确找到安装路径，可在安装时勾选“Add Python 3.7 to PATH”前的复选框（见图 1-4 的下部）。为方便今后对安装路径的操作，建议选择“Customize installation”，将安装路径（Customize install location）设置为“C:\Python37”。

添加 Python 安装路径也可通过设置操作系统环境变量实现，步骤如下：打开控制面板主页，单击“高级系统设置”项，在“系统属性”对话框中，单击“高级”选项卡中的“环境变量”按钮，在“环境变量”对话框的“系统变量”列表框中，选择“Path”项，单击“编辑”按钮，在打开的对话框中添加安装路径（如“C:\Python37”）和脚本工具安装路径（如“C:\Python37\Scripts”），如图 1-5 所示。

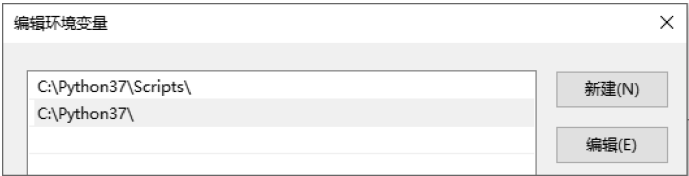


图 1-5 向操作系统环境变量中添加 Python 安装路径

1.2.2 Anaconda 集成开发环境

Anaconda 是一个 Python 科学计算集成开发环境的开源发行版本，可从其官网（见图 1-6）或其国内镜像网站免费下载安装。

Anaconda 在 Windows 操作系统中安装成功后，可见图 1-7 所示的程序组。其中包括常用的第三方包管理工具 Anaconda Navigator、命令行窗口 Anaconda Prompt、交互笔记 Jupyter Notebook、编程环境 Spyder 等。

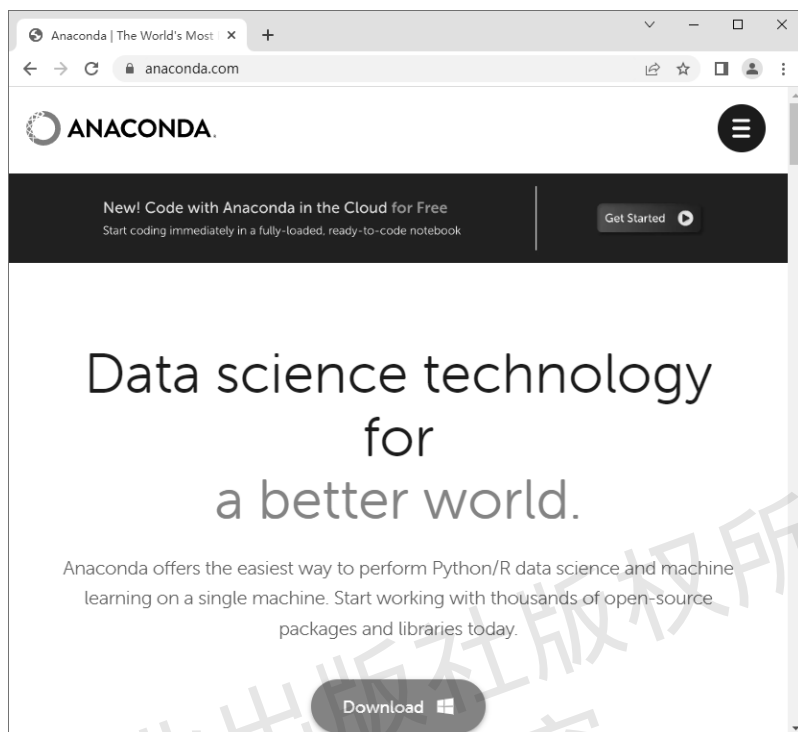


图 1-6 Anaconda 官网

Anaconda 不仅集成了 Python 开发环境，还包含 numpy、pandas、scipy、matplotlib、PIL、NLTK 等 200 余个科学计算常用第三方包。

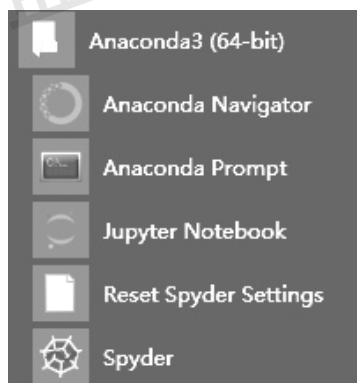
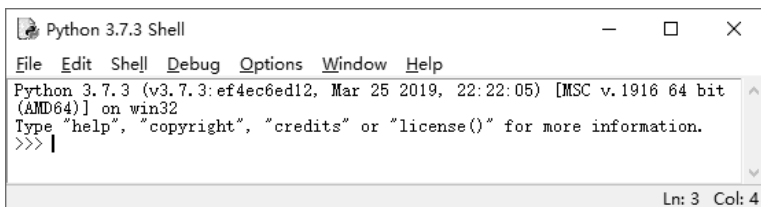


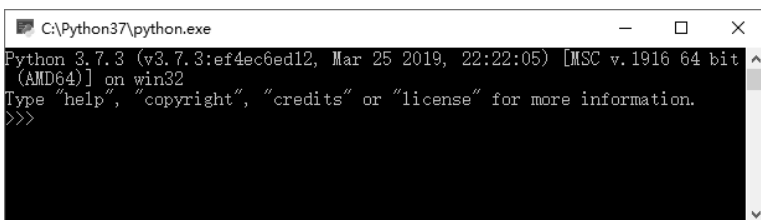
图 1-7 Anaconda 程序组

1.3 Python 程序设计步骤

Python 自带的集成开发环境 IDLE (Integrated Development and Learning Environment) 如图 1-8 (a) 所示，命令行运行环境如图 1-8 (b) 所示。



(a) Python 的 IDLE



(b) Python 的命令行运行环境

图 1-8 Python 的运行环境

在 IDLE 中, 使用 “File” → “New File” 菜单命令, 可打开程序编辑器, 该编辑器除文本编辑功能外, 还包含关键字颜色区分、简单的智能提示、自动缩进等辅助编辑功能 (见图 1-9)。

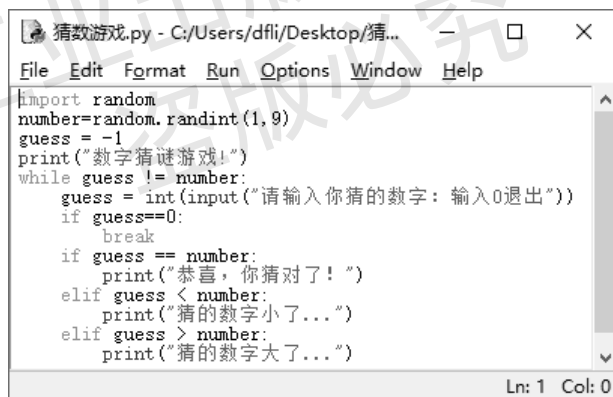


图 1-9 IDLE 的程序编辑器

Python (源) 程序以 .py 为扩展名。当运行 .py 程序时, 系统会自动生成一个对应的 .pyc 字节编译文件, 用于跨平台运行和提高运行速度。另外, 还有一种扩展名为 .pyo 的文件, 是编译优化后的字节编译文件。

Python 使用缩进来表示代码块层次, 习惯上一层缩进 4 个半角空格, 同一个代码块中的语句必须包含相同的缩进空格数, 不建议随意变化缩进空格数或使用 Tab 键。

Python 通常是一行写完一条语句, 但如果语句很长, 可以使用反斜杠 “\” 来实现语句转行。

Python 可以在同一行中放置多条语句, 语句之间使用分号 “;” 分隔, 但为易读起见, 不建议在同一行中放置多条语句。

Python 中单行注释以“#”开头。在调试程序时，如果临时需要不执行某些行，建议在不执行的行前加“#”，可避免大量删改操作。

1.4 常用的 Python 第三方编辑器

1. 记事本

Python 程序与其他高级语言一样，是纯文本文件，可以用操作系统自带的记事本打开和编辑（见图 1-10）。

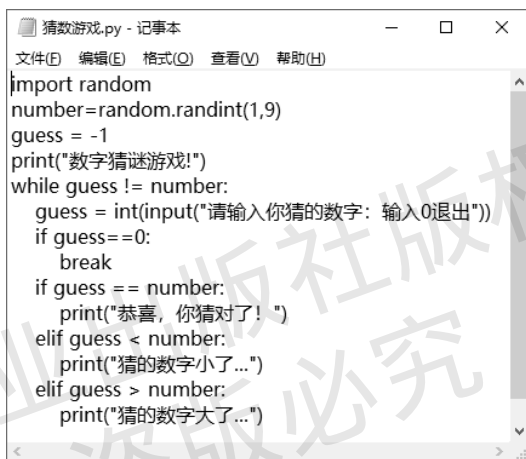


图 1-10 用记事本编写 Python 程序

值得注意的是，记事本默认保存为 ANSI 编码的.txt 文件（关于编码，详见第 5 章），可使用“另存为”命令，在弹出的“另存为”对话框中选择保存类型为“所有文件(*.*)”，并手工添加文件扩展名.py。在 Python 程序中，若包含中文等非英文字符，可选择 UTF-8 编码保存（见图 1-11）。



图 1-11 Python 程序的保存类型和编码

如果以 ANSI 编码保存的 Python 程序中含有中文等非英文字符，在用 IDLE 打开时可能出现如图 1-12 所示的编码选择对话框，让用户确认以何种编码读取。可使用与 ISO-8859-1

完全兼容、几乎可以表示世界上所有字符的字符编码 UTF-8，也可使用操作系统默认的中文简体扩展字符集编码 cp936 (GBK)。

预先在 Python 程序最前面添加编码注释“`#coding:GBK`”、“`#coding:UTF-8`”或“`_*_coding=utf-8*_`” (表示编码的字符串用大小写字母均可) 等，可以避免在运行程序前弹出编码选择对话框。

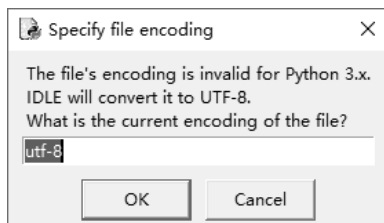


图 1-12 编码选择对话框

Python 程序员通常会选用第三方集成开发环境 (Integrated Development Environment, IDE) 进行程序设计。常用的集成开发环境有 PyCharm、VS Code、Thonny、PyScripter、Eclipse with PyDev、Komodo、Wing IDE 等，它们通常具有一些自动代码完成、参数提示、代码错误检查等功能。

2. PyCharm

如图 1-13 所示的 PyCharm 是 JetBrains 公司出品的集成开发环境，分为专业版 (Professional) 和社区版 (Community)，可从其官网下载。专业版试用期内免费，社区版完全免费并开源。

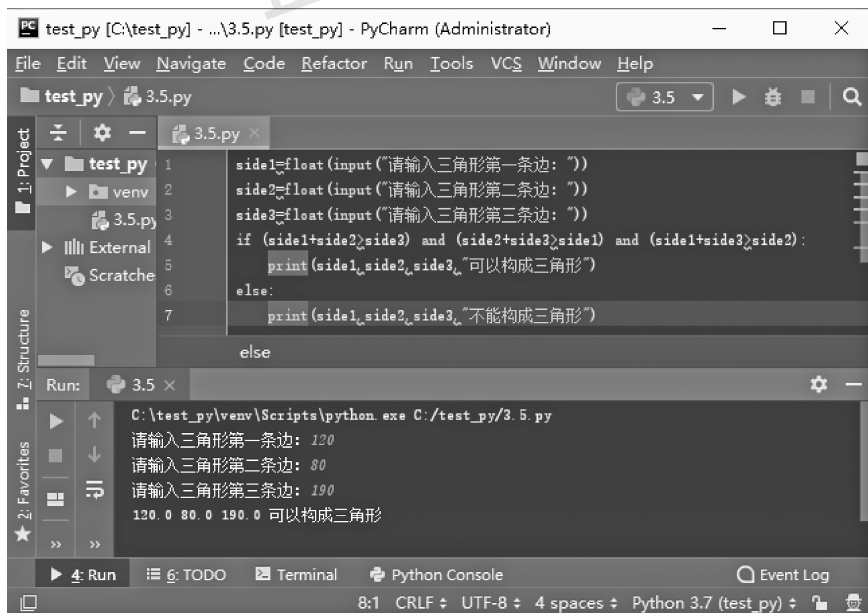


图 1-13 PyCharm

PyCharm 是一套 Python 程序开发的高效率工具，除具有调试、语法高亮显示、Project（项目）管理、代码跳转、代码智能提示、代码自动完成、单元测试、版本控制等一般功能外，还提供了支持 Django 等框架的专业 Web 开发等高级功能。尤其是快捷键映射设置，可兼容常见集成开发环境（如 Eclipse、Visual Studio、IntelliJ IDEA 等）的使用习惯（见图 1-14），让使用其他语言的程序员尽快适应其编程环境。

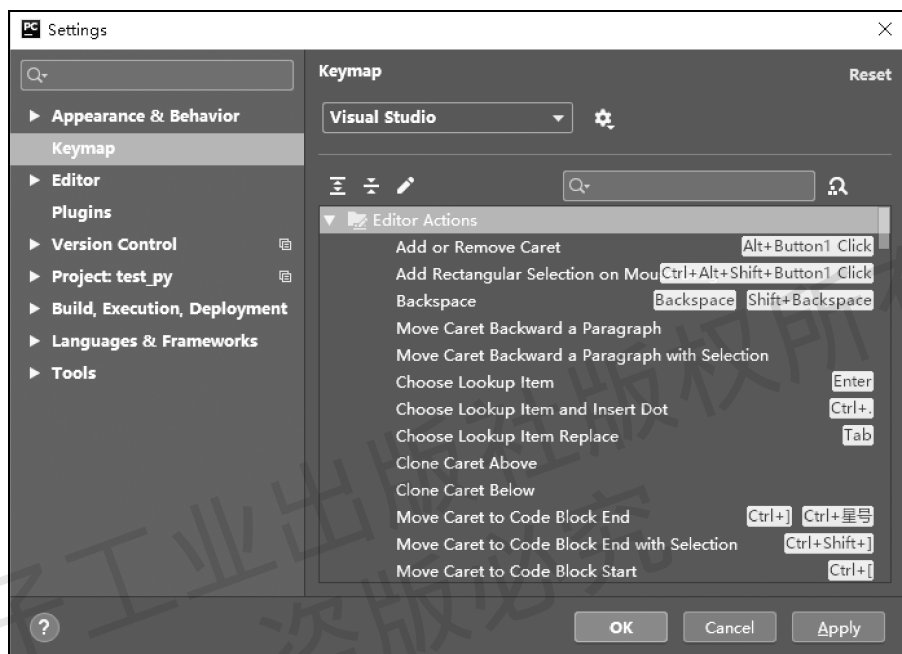


图 1-14 PyCharm 的快捷键映射设置

3. VS Code

VS Code（Visual Studio Code）是微软公司出品的轻量级代码编辑器（见图 1-15），支持 Windows、macOS 和 Linux 操作系统。它有丰富的插件生态系统，支持 C++、C#、Java、PHP、Python、Visual Basic、XML、R、Objective-C、JavaScript、JSON、HTML、CSS 等语言和语法结构。编辑器界面沿用了微软经典的 VS 风格，支持语法高亮显示、代码格式化、代码智能提示、括号匹配等编辑功能，并可直接访问 Git 托管平台。通常建议在 VS Code 中用以文件夹方式打开和编辑项目，也可以编辑单个文件。

4. Thonny

Thonny 为面向初学者的免费 Python 集成开发环境，如图 1-16 所示。它支持多种语言并适应多种编码，提供语法高亮显示、代码提示等功能。由于软件容量很小，因此适用于树莓派等物联网软件开发。



图 1-15 VS Code

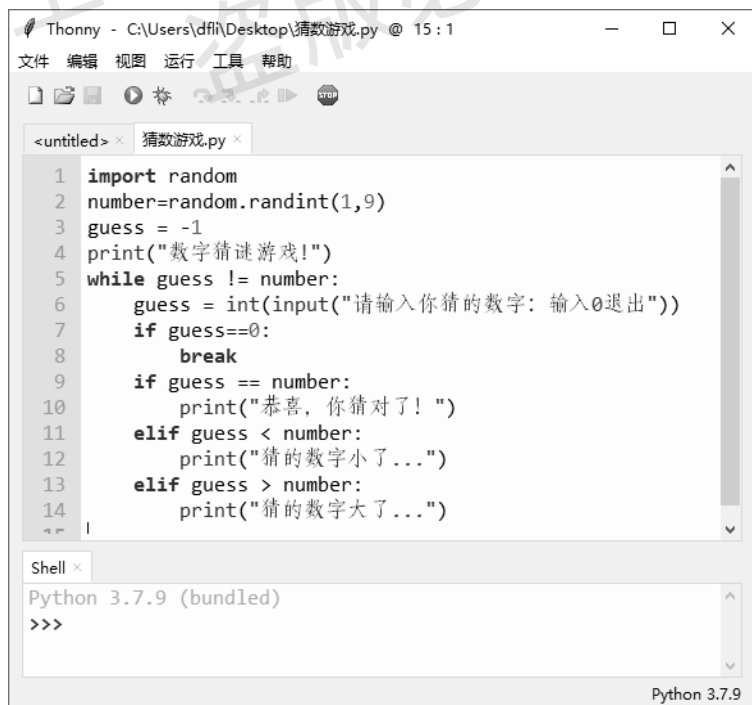


图 1-16 Thonny

5. PyScripter

如图 1-17 所示为开源的 Python 集成开发环境 PyScripter，可从 GitHub 网站免费下载。其具有语法高亮显示、语法自动补全、语法检查、断点调试等功能，还可以编辑 JavaScript、PHP、HTML、XML 等类型的文件。

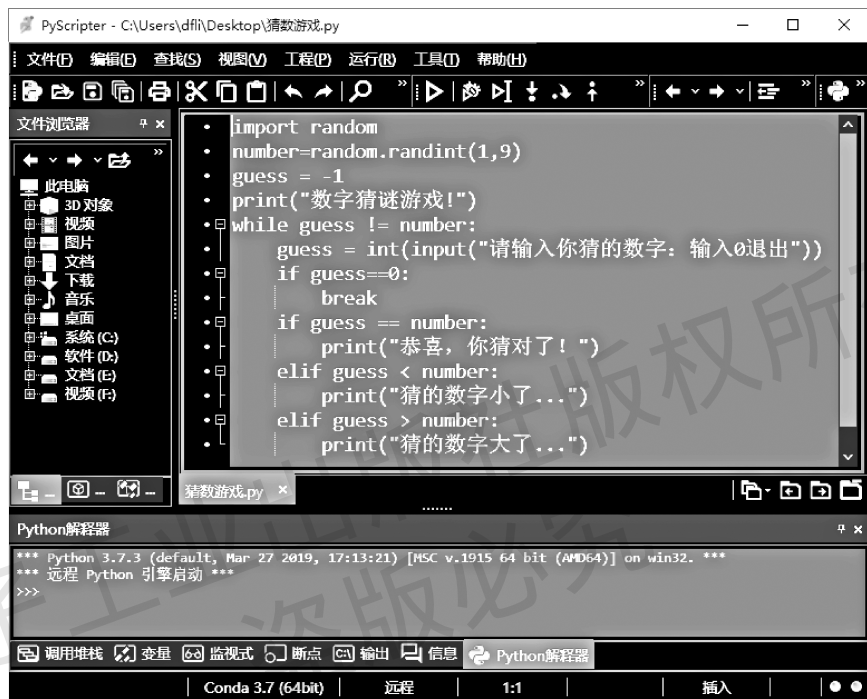


图 1-17 PyScripter

习题 1

1. 下载并安装 Python 3.x 版，检查系统变量 Path 中的安装路径，体验并编写一个简单的 Python 程序。
2. 下载并安装一种第三方 IDE，并逐渐熟悉使用它。



获取本章资源