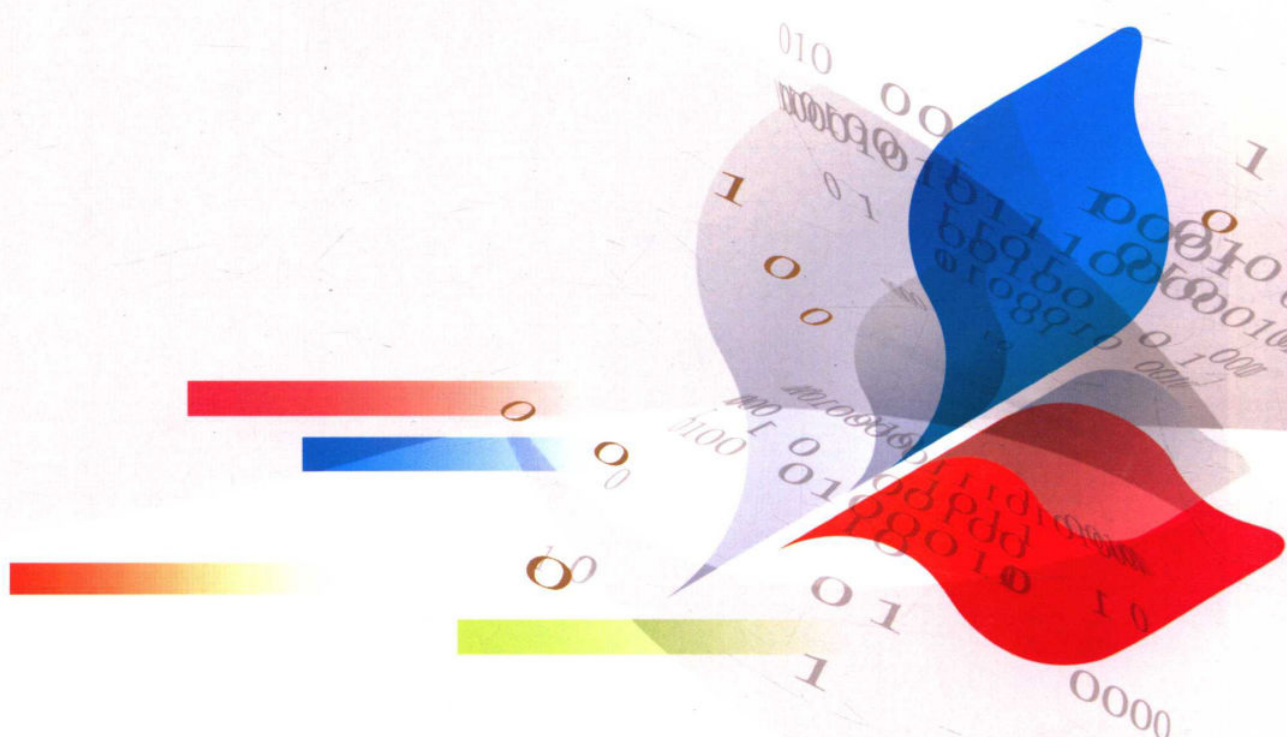




普通高等教育“十三五”规划教材



上海市高等学校计算机等级考试（二级）参考教材

Python程序设计基础

（第2版）

◎ 李东方 文欣秀 张向东 主编



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

普通高等教育“十三五”规划教材
上海市高等学校计算机等级考试（二级）参考教材

Python 程序设计基础

（第2版）

李东方 文欣秀 张向东 主编

常 珊 吴 萍
编著

刘 洋 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计算机等级考试(二级)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。本教材以程序设计初学者为教学对象,从程序设计基本概念出发,通过大量示例由浅入深、循序渐进地讲述 Python 程序设计的基本概念和基本方法。本书内容包括 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。本书的例题源码、电子课件、习题素材等资源提供扫码下载。

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 语言爱好者自学和计算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,文欣秀,张向东主编. —2 版. —北京:电子工业出版社,2020.1
ISBN 978-7-121-37605-4

I. ①P… II. ①李…②文…③张… III. ①软件工具—程序设计—高等学校—水平考试—教材
IV. ①TP311.561

中国版本图书馆 CIP 数据核字(2019)第 219357 号

责任编辑:冉 哲

印 刷:三河市鑫金马印装有限公司

装 订:三河市鑫金马印装有限公司

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:16.25 字数:414.4 千字

版 次:2017 年 1 月第 1 版

2020 年 1 月第 2 版

印 次:2020 年 1 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 zlts@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，为在更加信息化的将来所从事的专业活动，打下理解需求、求解问题、程序实现的扎实基础。

在上海市教育委员会高教处、上海市教育考试院、上海市高等学校计算机等级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、计算机等级考试、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年上海市高等学校率先增考了二级 Python 程序设计科目。本书是上海市高等学校计算机等级考试（二级）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。

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，是学生学习计算机编程、理解计算机解决问题的方法的有效工具。学生通过对 Python 语言的学习，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。

本书以初学者为起点，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、华东师范大学、东华大学、华东政法大学等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。通过实例进行讲解，不拘泥于语法细节，避免曲折烦琐，也力图体现 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计算机等级考试（二级）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。本书的例题源码、电子课件、习题素材等资源提供扫码下载。

本书由李东方（第 1、6、8、9、10、12 章）、张向东（第 2 章）、文欣秀（第 3 章）、常姗（第 4 章）、吴萍（第 5 章）、刘洋（第 7 章）、郑奋（第 11 章）编写，文欣秀为各章编配了习题，最后由李东方统稿。在编写过程中还得到了同济大学、上海大学和上海对外经贸大学等校教师的指导与支持。本书部分实例应用了上海市高等学校计算机等级考试既往试题素材和部分网上佚名素材，在此一并表示诚挚感谢。

由于时间仓促和水平有限，书中难免有不妥之处，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

建议学时数为 48~64 学时，其中 16~32 学时为实验课。建议以机房“现场授课”方式教学，边讲边练，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 3.5.2 以上或 Anaconda 相应版本，可选装 PyCharm、VS Code、PyScripter、Wing IDE、Spyder 或 Notepad++ 等编程调试环境。

目 录

第 1 章 Python 语言及其编程环境	(1)
1.1 Python 语言概述	(1)
1.2 Python 的安装	(3)
1.2.1 Python 的官网下载安装	(3)
1.2.2 Anaconda 集成开发环境	(4)
1.3 Python 程序设计步骤	(5)
1.4 常用的 Python 第三方编辑器	(7)
习题 1	(11)
第 2 章 Python 的基本语法	(12)
2.1 Python 源程序的书写格式与基本规则	(12)
2.1.1 基本词法单位、标识符/常量/运算符等的构成规则与关键字	(12)
2.1.2 源程序的书写格式与基本规则	(14)
2.2 Python 的基本数据类型	(18)
2.2.1 数值类型	(18)
2.2.2 字符串类型	(21)
2.3 Python 的基本运算和表达式	(24)
2.3.1 变量的操作	(24)
2.3.2 运算符	(29)
2.3.3 算术运算	(29)
2.3.4 输入与输出	(31)
2.3.5 赋值、关系和逻辑运算	(33)
2.3.6 表达式	(35)
2.3.7 赋值语句	(36)
习题 2	(38)
第 3 章 Python 程序的基本流程控制	(40)
3.1 计算思维和程序设计基本方法	(40)
3.1.1 计算思维	(40)
3.1.2 程序设计基本方法	(41)
3.2 顺序结构	(41)
3.3 分支结构	(43)
3.3.1 if 语句	(43)
3.3.2 if-elif-else 语句	(43)
3.3.3 分支语句嵌套	(46)
3.4 循环结构	(47)

3.4.1	while 语句	(47)
3.4.2	for 语句和 range() 内置函数	(49)
3.4.3	循环语句嵌套	(51)
3.4.4	转移和中断语句	(53)
3.5	程序调试	(55)
3.5.1	语法错误与逻辑错误	(55)
3.5.2	常见语法错误	(55)
3.5.3	排查程序错误的方法	(59)
3.5.4	程序运行中 try-except 异常处理	(61)
习题 3		(63)
第 4 章	Python 的组合数据类型	(65)
4.1	序列型组合数据类型	(65)
4.1.1	字符串	(66)
4.1.2	列表	(71)
4.1.3	元组	(77)
4.2	映射型组合数据类型——字典	(83)
4.2.1	创建字典	(83)
4.2.2	访问字典中的值	(84)
4.2.3	更新字典	(84)
4.2.4	字典的操作	(84)
4.3	集合型组合数据——集合	(88)
4.3.1	集合的创建	(88)
4.3.2	集合的修改	(90)
4.3.3	集合的数学运算	(92)
习题 4		(93)
第 5 章	文件与基于文件的数据分析	(96)
5.1	文件的基本概念	(96)
5.1.1	文件和目录	(96)
5.1.2	文件的编码	(97)
5.1.3	文本文件和二进制文件的区别	(99)
5.2	文件的操作	(99)
5.2.1	文件的打开和关闭	(99)
5.2.2	定位	(100)
5.2.3	文件的读取、写入、追加	(102)
5.3	基于文件的数据分析	(105)
5.3.1	对数据文件的读取和遍历	(105)
5.3.2	词频分析	(108)
5.3.3	第三方库 jieba 和中文词频分析	(110)

5.3.4	第三方库 wordcloud 和词语可视化	(111)
习题 5	(112)
第 6 章	函数	(114)
6.1	函数的定义与调用	(114)
6.1.1	函数的声明	(114)
6.1.2	函数的调用	(115)
6.2	参数的传递	(115)
6.2.1	参数按位置依次传递	(115)
6.2.2	参数赋值和参数默认值传递	(115)
6.2.3	元组类型可变长参数传递	(116)
6.2.4	字典类型可变长参数传递	(116)
6.2.5	高阶函数	(117)
6.2.6	函数中变量的作用域	(118)
6.3	匿名函数	(119)
6.4	函数的递归	(120)
6.5	函数的高级应用	(122)
6.5.1	生成器	(122)
6.5.2	装饰器与闭包	(122)
习题 6	(124)
第 7 章	面向对象的程序设计与 Python 生态	(128)
7.1	面向对象的概念	(128)
7.2	类与实例	(129)
7.2.1	创建类和子类	(129)
7.2.2	增加属于类的对象实例	(130)
7.2.3	用函数表示类的行为特征	(132)
7.3	面向对象的特征	(132)
7.3.1	封装	(132)
7.3.2	继承	(133)
7.3.3	多态	(134)
7.4	Python 程序的组织和管理	(135)
7.4.1	源程序和模块结构	(135)
7.4.2	包和库	(136)
7.4.3	库的模块引用	(137)
7.5	Python 的生态	(139)
7.5.1	内置库与第三方库	(139)
7.5.2	第三方库的安装	(141)
7.6	可执行文件与 pyinstaller 库	(144)
习题 7	(145)

第 8 章 图形化界面设计	(146)
8.1 窗体控件布局	(146)
8.1.1 根窗体	(146)
8.1.2 tkinter 常用控件	(147)
8.1.3 控件布局	(151)
8.2 tkinter 常见控件的特有属性	(154)
8.2.1 文本输入/输出相关控件	(154)
8.2.2 按钮	(157)
8.2.3 单选按钮	(159)
8.2.4 复选框	(160)
8.2.5 列表框与组合框	(164)
8.2.6 滑块	(170)
8.2.7 菜单	(171)
8.2.8 子窗体	(173)
8.2.9 模式对话框	(174)
8.3 事件响应	(178)
习题 8	(181)
第 9 章 图形绘制与数据可视化	(184)
9.1 tkinter 库的 Canvas 图形绘制方法	(184)
9.1.1 Canvas 绘图的基本方法	(185)
9.1.2 Canvas 的函数图形绘制	(189)
9.2 turtle 库的图形绘制方法	(192)
9.2.1 turtle 的基本方法	(193)
9.2.2 turtle 图形绘制	(194)
9.3 matplotlib 库的图形绘制方法	(202)
9.3.1 环境安装和基本方法	(202)
9.3.2 绘制二维函数图形	(205)
习题 9	(208)
第 10 章 正则表达式与简单爬虫	(210)
10.1 正则表达式	(210)
10.1.1 非贪婪匹配模式	(211)
10.1.2 贪婪匹配模式	(212)
10.2 re 库的内置函数	(212)
10.2.1 匹配与搜索	(213)
10.2.2 切分与分组	(215)
10.2.3 替换	(215)
10.3 正则表达式的应用: 简单爬虫	(216)
10.3.1 爬取新闻链接和标题	(216)

10.3.2	爬取新闻图片素材	(218)
习题 10		(219)
第 11 章	数据库操作	(220)
11.1	访问 SQLite 数据库	(220)
11.1.1	SQLite 数据库连接对象及表的 SQL 操作	(221)
11.1.2	游标对象和 SQL 查询	(224)
11.2	访问 Access、MySQL 和 SQL Server 数据库	(225)
习题 11		(227)
第 12 章	基于第三方库的应用举例	(231)
12.1	对 Excel 文件的操作	(231)
12.1.1	用 openpyxl 在内存中创建工作表	(231)
12.1.2	数据的读取	(232)
12.1.3	编辑操作和写入	(233)
12.2	图片操作	(236)
12.2.1	Image 模块及应用	(236)
12.2.2	ImageChops 特效与合成	(238)
习题 12		(241)
附录 A	上海市高等学校计算机等级考试（二级）Python 程序设计考试大纲	(242)
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	(247)
参考文献		(250)

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

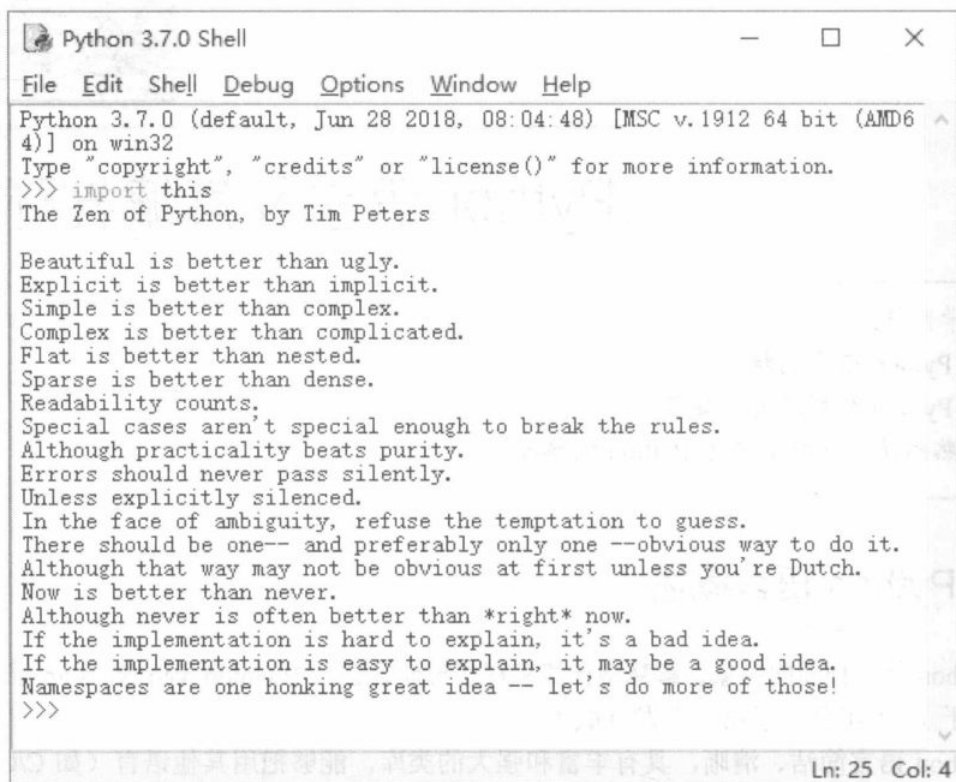
Python 是一种面向对象、解释型计算机程序设计语言，由 Guido van Rossum 于 1989 年发明，于 1991 年公开了第一个发行版本。

Python 语言简洁、清晰，具有丰富和强大的类库，能够把用其他语言（如 C/C++）制作的各种模块很轻松地联结在一起，又被昵称为胶水语言。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。由于 Python 语言的简洁、优雅、开发效率高，既能快速生成程序的原型，又能方便地将 Python 程序封装成可调用的扩展类库，程序无须修改就能在 Windows，Linux，UNIX，Mac OS 等操作系统上跨平台使用，因此，它常被用于网站开发、网络编程、图形处理、黑客攻防等。根据 2019 年 8 月 TIOBE 编程语言排行榜（见图 1-1），Python 已迅速上升为第三大广泛应用的编程语言。

Aug 2019	Aug 2018	Change	Programming Language	Ratings	Change
1	1		Java	16.028%	-0.85%
2	2		C	15.154%	+0.19%
3	4	^	Python	10.020%	+3.03%
4	3	v	C++	6.057%	-1.41%
5	6	^	C#	3.842%	+0.30%
6	5	v	Visual Basic .NET	3.695%	-1.07%
7	8	^	JavaScript	2.258%	-0.15%
8	7	v	PHP	2.075%	-0.85%
9	14	^	Objective-C	1.690%	+0.33%
10	9	v	SQL	1.625%	-0.69%

图 1-1 2019 年 8 月 TIOBE 编程语言排行榜

Python 语言崇尚优雅、明确、简单。在其行命令编程环境中输入“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!
>>>
```

Ln: 25 Col: 4

图 1-2 Python 之禅

Python 之禅的中文译意是：

- 优美胜于丑陋，显式胜于隐式。
- 简洁胜于复合，复合胜于复杂。
- 扁平胜于嵌套，稀疏胜于密集。
- 可读性很重要。
- 虽然语言的实用性胜于其纯粹性，但也不可因此而打破规则。
- 错误永远不会悄悄过去，除非明确地忽略错误。
- 面对歧义，拒绝猜测的诱惑。
- 应该有一种（最好只有一种）显而易见的解决方案。
- 解决方案最初可能并不是显而易见的，除非你就是那个荷兰人（指 Guido）。
- 马上做胜于永远不做，但盲目动手做有时还不如不做。
- 如果某个方案难以阐释，那么这通常是一个糟糕的方案。
- 如果某个方案容易解释，那么它可能是一个好主意。
- 命名空间是一个很棒的主意——我们可以多做些尝试。

1.2 Python 的安装

Python 开发环境是完全免费的自由软件，但下载安装前应考虑如下问题。

- 支持的操作系统。Python 支持 Windows, Linux, UNIX, Mac OS 等不同操作系统，应选择对应的安装程序。
- 操作系统字长。应根据操作系统的 32 位或 64 位字长选择对应的安装程序，以获得最佳运行环境。
- Python 的版本。选择 3.x 版还是 2.7 版？3.x 版与 2.x 版并不完全兼容，大批用 Python 2.x 版编写的库函数无法在 3.x 版下使用。虽然 2.x 版已经被广泛应用多年，且较为成熟，网上有大量类库资源，但 Python 2.x 版到 2.7 版后不再升级，3.x 版以后的版本升级将是 Python 语言的未来。至 2015 年年初，绝大多数 Python 语言编写的库函数都可以稳定高效地在 Python 3.x 版下运行。基于以上原因，本教材使用 3.x 版。目前，Python 安装主要通过官网下载安装和 Anaconda 集成环境下载安装两种途径。

1.2.1 Python 的官网下载安装

最新版本的安装程序可从 Python 官网 www.python.org 免费下载，如图 1-3 所示。

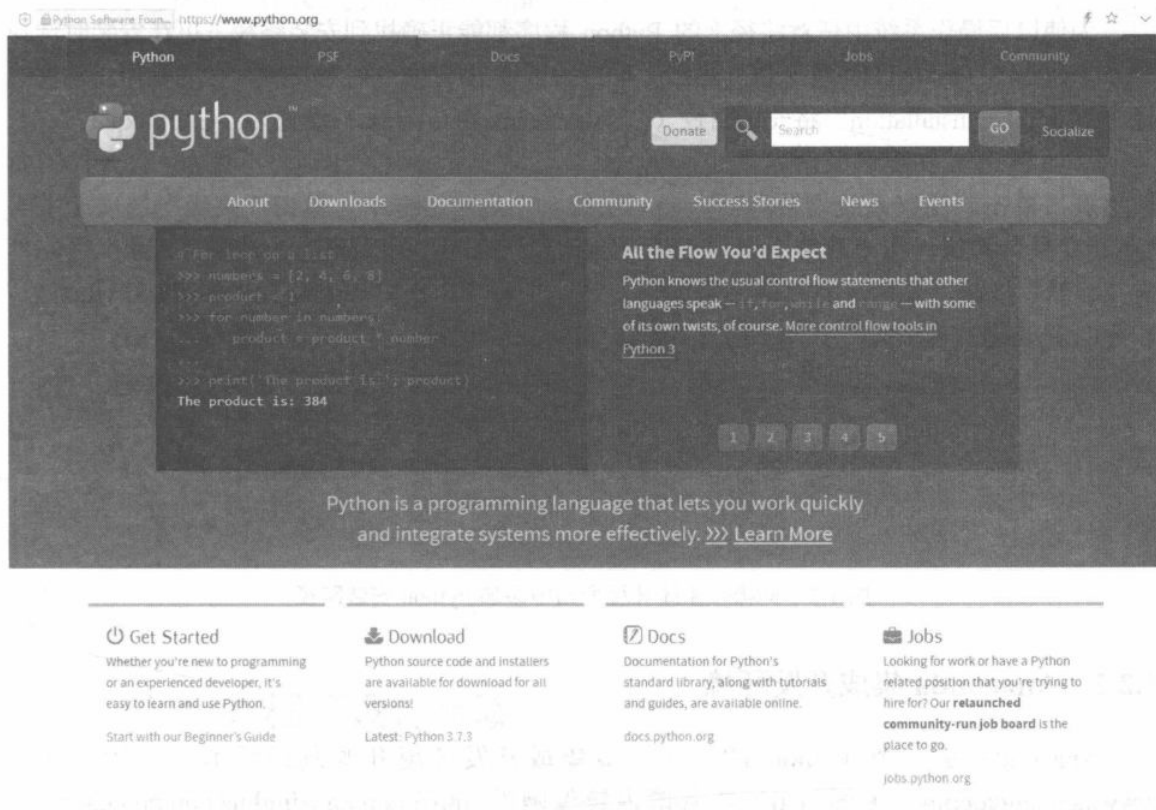


图 1-3 官网下载页面

在官网下载页面中可选择安装的操作系统和系统字长,以及安装包的形式。以 Windows 操作系统可执行程序安装为例,双击 python-3.x.x.exe 文件,即可按向导提示进行安装,如图 1-4 所示。

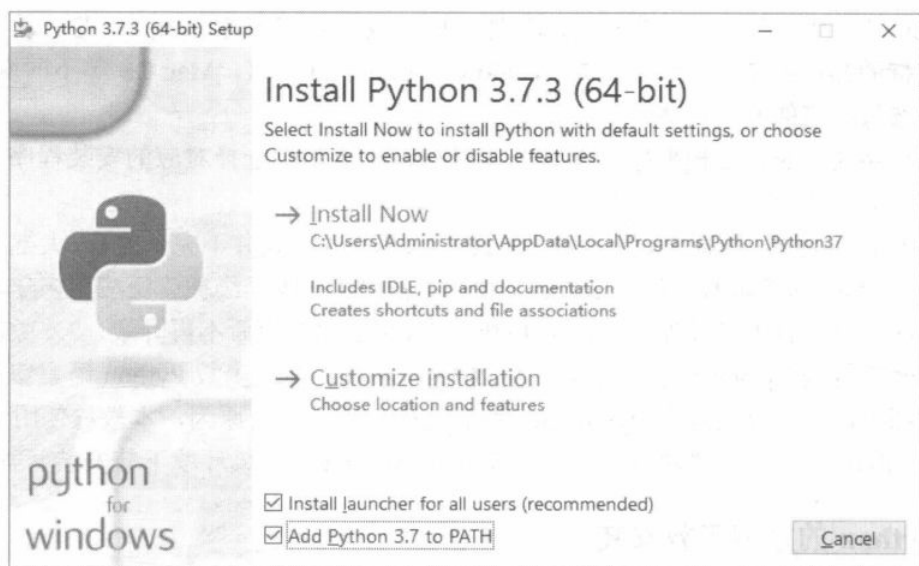


图 1-4 Python 安装向导

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,可在安装时选择“Add Python 3.7 to PATH”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 安装路径也可通过对操作系统环境变量的设置实现,步骤为: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和脚本工具安装路径(如“C:\Python37”和“C:\Python37\Scripts”),如图 1-5 所示。

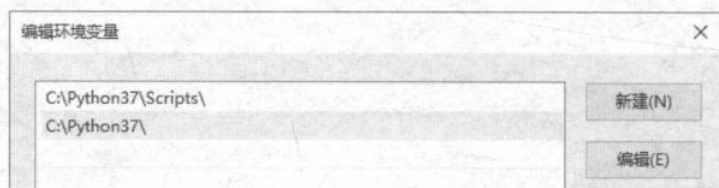


图 1-5 向操作系统环境变量中添加 Python 安装路径

1.2.2 Anaconda 集成开发环境

Anaconda 是一个 Python 的科学计算集成开发环境开源发行版本,可从其官网 www.anaconda.com (见图 1-6) 或其国内镜像网站 mirrors.tuna.tsinghua.edu.cn/anaconda/archive 免费下载安装。



图 1-6 Anaconda 官网

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

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

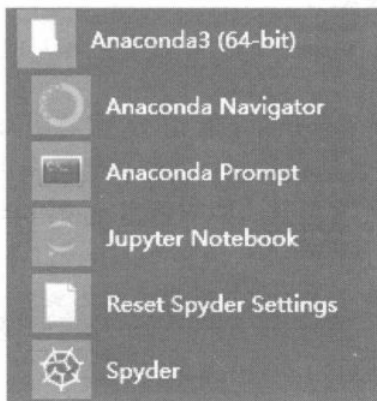
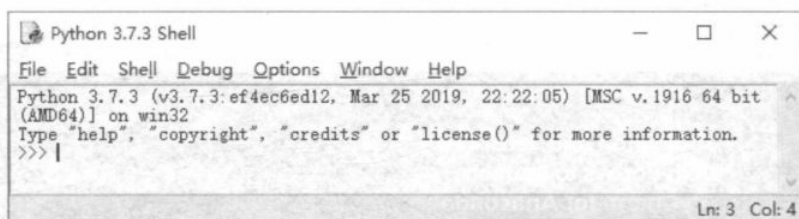


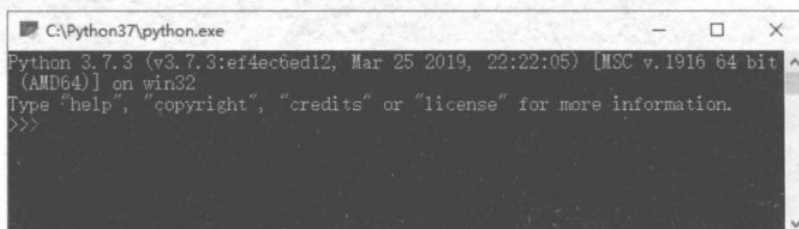
图 1-7 Anaconda 程序组

1.3 Python 程序设计步骤

Python 自带的 IDLE 运行环境 (GUI Shell), 如图 1-8 (a) 所示, 其行命令运行环境如图 1-8 (b) 所示。



(a) Python 的 IDLE 运行环境



(b) Python 的行命令运行环境

图 1-8 Python 的运行环境

在 IDLE 运行环境中, 使用“File”→“New File”菜单命令, 可呼出程序编辑器, 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)。

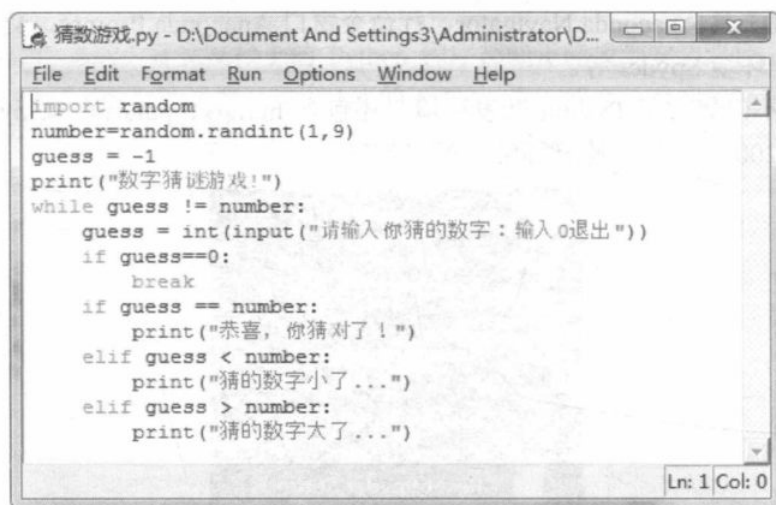


图 1-9 IDLE 的程序编辑器

Python 源程序以.py 为扩展名。当运行.py 源程序时, 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 Python 编译优化后的字节编译文件。

Python 使用缩进来表示代码块, 习惯上缩进 4 个半角空格, 同一个代码块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制表位键。

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。在程序调试时，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 的源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）。

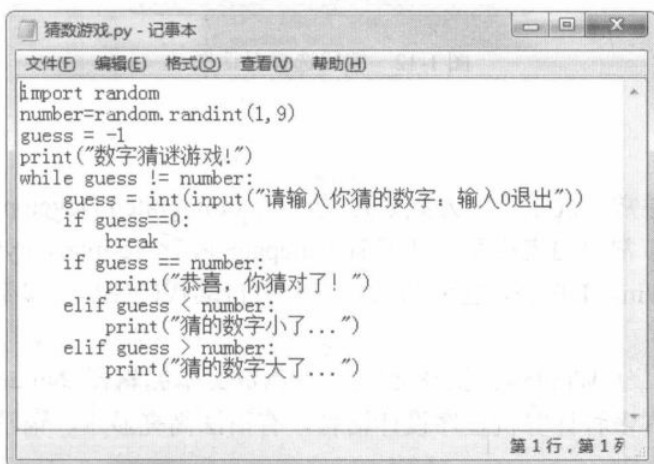


图 1-10 用记事本编写 Python 源程序

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，在弹出的保存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 3.x 版的 Python 源程序中，若包含中文等非英文字符，也可直接选择 UTF-8 编码方式保存（见图 1-11）。

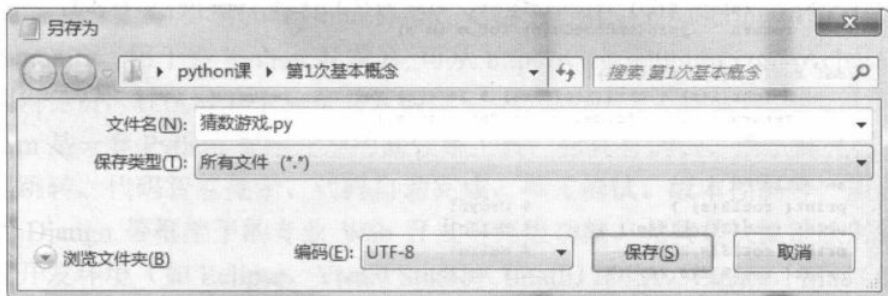


图 1-11 Python 源程序的保存类型和编码

如果以 ANSI 编码方式保存的 Python 源程序中含有中文等非英文字符，在打开时可能