新工科人才培养计算机类“十三五”规划教材

Python程序设计与案例教程

主　编　王小银
副主编　王曙燕

西安电子科技大学出版社
http://www.xduph.com

新工科人才培养计算机类“十三五”规划教材

Python 程序设计与案例教程

主　编　王小银

副主编　王曙燕

西安电子科技大学出版社

内 容 简 介

本书以程序设计为主线，以程序设计初学者作为教学对象，由浅入深、循序渐进地讲述了Python 语言的基本概念、基本语法和数据结构等基础知识。全书共分 14 章，内容包括程序设计基础与 Python 概述，数据类型，Python 程序设计基础，基本程序设计结构(顺序、选择和循环三种)，组合数据类型，函数与模块，文件，异常处理，面向对象程序设计，图形用户界面设计，Python 的标准库和第三方库，基于 Pygame 的游戏开发。

本书实例丰富，可作为高等院校相关专业 Python 程序设计课程的教材或教学参考书，也可作为大学各专业程序设计公共教材和全国计算机等级考试参考用书，还可供计算机应用开发技术人员和计算机爱好者自学使用。

图书在版编目(CIP)数据

Python 程序设计与案例教程/王小银主编. —西安：西安电子科技大学出版社，2019.1
ISBN 978-7-5606-5172-9

Ⅰ. ① P… Ⅱ. ① 王… Ⅲ. ① 软件工具—程序设计—教材 Ⅳ. ① TP311.561

中国版本图书馆 CIP 数据核字(2018)第 286842 号

策划编辑 李惠萍
责任编辑 唐小玉 雷鸿俊
出版发行 西安电子科技大学出版社(西安市太白南路 2 号)
电 话 (029)88242885 88201467 邮 编 710071
网 址 www.xduph.com 电子邮箱 xdupfxb001@163.com
经 销 新华书店
印刷单位 陕西天意印务有限责任公司
版 次 2019 年 1 月第 1 版 2019 年 1 月第 1 次印刷
开 本 787 毫米×1092 毫米 1/16 印 张 17.5
字 数 414 千字
印 数 1～3000 册
定 价 39.00 元
ISBN 978-7-5606-5172-9 / TP

XDUP 5474001-1

前　言

Python 语言由荷兰国家数字与计算机科学研究院研究员吉多·范罗苏姆(Guido van Rossum)于 1989 年发明，第一个公开发行的版本发行于 1991 年。Python 语言的设计哲学是“优雅”、“明确”和“简单”。吉多·范罗苏姆在设计 Python 时，目的是想设计出一种优美而功能强大，可提供给非专业程序设计师使用的语言，同时采取开放策略，使 Python 能够完美结合如 C、C++和 Java 等其他语言。

经过二十多年的发展，Python 已经广泛应用于计算机科学与技术、科学计算、数据统计分析、移动终端开发、图形图像处理、人工智能、游戏设计、网站开发等领域。Python 是一种面向对象、解释运行、扩展性很强的程序设计语言，语法简洁清晰，同时拥有功能丰富的标准库和扩展库。标准库提供了系统管理、网络通信、文本处理、数据库接口、图形系统、XML 处理等功能；扩展库覆盖科学计算、Web 开发、数据库接口、图形系统等多个领域，并且大多功能成熟而稳定。

通过 Python 语言程序设计课程的学习，读者可以掌握 Python 语言的程序结构、语法规则和编程方法，达到独立编写常规 Python 语言应用程序的能力，同时为设计大型应用程序和系统程序打下坚实的基础。本课程是数据结构、操作系统和软件工程等课程的基础，并可为这些课程提供实践工具。

本书以程序设计为主线，以初学者为起点，由浅入深、循序渐进地讲述了 Python 语言的基本概念、基本语法和数据结构等基础知识，同时对 Python 语言的标准库和第三方库及其应用进行了较全面的讲述。全书共 14 章，第 1 章介绍了程序设计基础与 Python 的基本概念；第 2 章介绍了 Python 语言的基本数据类型、运算符和表达式；第 3～5 章介绍了 Python 程序设计基础与三种基本程序设计结构(顺序结构、选择结构和循环结构)；第 6 章介绍了 Python 语言中的组合数据类型；第 7 章介绍了函数、模块的定义和使用；第 8、9 章介绍了文件、异常处理的基本知识；第 10 章介绍了面向对象程序设计的相关知识及应用；第 11 章介绍了使用 Python 进行图形用户界面设计的方法；第 12、13 章分别对 Python 中常用的标准库和第三方库进行了解析，并给出了应用实例；第 14 章介绍了基于 Pygame 模块进行游戏开发的基本方法及其实例辨析。本书中丰富的例题均在 Python 3.7 运行环境中调试通过。

本书第 1～10 章和第 12 章及附录部分由王小银编写，第 11、13 和 14 章由王曙燕编写，全书由王小银统稿。研究生张一权和张海清参与了部分校对工作，作者在此一并向他们表示衷心的感谢。

本书可作为高等学校 Python 语言程序设计课程的教材，也可作为工程技术人员和计算机爱好者的参考用书。

由于编者水平有限，书中难免存在不足之处，恳请广大读者多提宝贵意见。

作者联系方式：wangxiaoyinxy@126.com。

作　者

2018 年 10 月

目　录

第 1 章　程序设计基础与 Python 概述

计算机是一种能按照事先存储的程序，自动、高速进行大量数值计算和各种信息处理的现代化智能电子装置。计算机不仅能自动、高速、精确地进行信息处理，还具有存储信息、记忆信息、判断推理等功能，在很大程度上能够代替人的部分脑力劳动，具备类似人脑的“思维能力”，因此计算机也简称为“电脑”。计算机的广泛应用标志着信息化时代的到来，它对社会、商业、政治以及人际交往的方式和人的生活产生了深远的影响。计算机的迅速发展推动了人类的智力解放，使科技、生产、社会和人类活动发生了重大变化，人们利用计算机能高效解决科学计算、工程设计、经营管理、过程控制、人工智能等各种问题。现在计算机已成为最基本的信息处理工具。

人们使用计算机解决问题时，必须用某种“语言”来和计算机进行交流。计算机语言是人与计算机之间交流信息的工具，计算机语言由计算机能够识别的语句组成，它使用一整套带有严格规定的符号体系来描述计算机语言的词法、语法、语义和语用。

1.1　程序设计与程序设计语言

计算机系统由硬件系统和软件系统两大部分组成，硬件系统是系统运行的物理设备总称，软件是管理、维护计算机系统和完成各项应用任务的程序。软件的主体是程序，程序是由计算机语言编写而成的。

1.1.1　程序设计与计算思维

1. 程序设计

程序设计是给出解决特定问题程序的过程，是软件构造活动中的重要组成部分。程序设计往往以某种程序设计语言为工具，并在这种语言下编写程序。程序设计过程包括分析、设计、编码、测试、排错等不同阶段。专业的程序设计人员常被称为程序员。一个正确的程序通常包括两层含义：一是书写正确，二是结果正确。书写正确是指源程序在语法上正确，符合程序设计语言的规则；而结果正确通常是指对应于正确的输入，程序能产生所期望的输出，符合使用者对程序功能的要求。

2. 计算思维

计算思维是当前国际计算机界广为关注的一个重要概念。2006 年 3 月，美国卡内基·梅隆大学计算机科学系主任周以真(Jeannette M. Wing)教授在美国计算机权威期刊

《Communications of the ACM》杂志上发表了题为“Computational Thinking”的论文。他在文中将计算思维定义为：计算思维是运用计算机科学的基础概念进行问题求解、系统设计以及人类行为理解等涵盖计算机科学之广度的一系列思维活动。

计算思维吸取了问题解决所采用的一般数学思维方法，包括现实世界中巨大复杂系统的设计与评估的一般工程思维方法，以及复杂性、智能、心理、人类行为的理解等一般科学思维方法。计算思维的本质是抽象与自动化。与数学和物理科学相比，计算思维中的抽象显得更为丰富，也更为复杂。

1.1.2 程序设计语言

程序设计语言通常称为编程语言，是能够被计算机和编程人员双方所理解和认可的交流工具。当软件开发人员希望计算机完成一件工作或解决一个问题时，首先需要确定解决问题的方法和步骤；然后再把这个方法和步骤用计算机能够理解和执行的程序设计语言表述出来，形成一组语句的集合，即程序。

程序设计语言并不唯一。在计算机技术发展的过程中，先后形成了数百种不同的程序设计语言。程序设计语言按照使用方式和功能可分为机器语言、汇编语言、高级语言。

1. 机器语言

机器语言(Machine Language)是用二进制代码表示的计算机能直接识别和执行的一种机器指令的集合，与计算机同时诞生，属于第一代计算机语言。机器语言在形式上是由“0”和“1”构成的一串二进制代码，有一定的位数，并被分成若干段，各段的编码表示不同的含义。用机器语言编写的程序不必通过任何翻译处理，计算机就能够直接识别和执行。机器语言具有灵活、直接执行和速度快等特点。不同型号的计算机其机器语言是不相通的，按照一种计算机的机器指令编制的程序，不能在另一种计算机上执行。例如运行在 IBM PC 机上的机器语言程序不能在 51 单片机上运行。

机器指令由操作码和操作数组成，操作码指出要计算机进行什么样的操作，操作数指出完成该操作的数或该数在内存中的地址。

例如，计算 1+2 的机器语言程序如下：

```
10110000 00000001      ; 将 1 存入寄存器 AL 中
00000100 00000010      ; 将 2 与寄存器 AL 中的值相加，结果放在寄存器 AL 中
11110100               ; 停机
```

由此可见，用机器语言编写程序时，编程人员首先要熟记所用计算机的全部指令代码和代码的含义，必须自己处理每条指令和每一数据的存储分配与输入/输出，还需要记住编程过程中每步所使用的工作单元处在何种状态。这是一件十分烦琐的工作，编写程序花费的时间很长。而且，用机器语言编写的程序全是些 0 和 1 的指令代码，直观性差，难学，难记，易出错，无特征，难检查，难修改，属于低级语言。

2. 汇编语言

汇编语言(Assembly Language)也是面向机器的程序设计语言。为了克服机器语言的缺点，人们采用了有助于记忆的符号(称为指令助记符)与符号地址来代替机器指令中的操作码和操作数。指令助记符是一些有意义的英文单词的缩写和符号，如用 ADD(Addition)表示

加法，用 SUB(Subtract)表示减法，用 MOV(Move)表示数据的传送，等等。而操作数可以直接用十进制数书写，地址码可以用寄存器名、存储单元的符号地址等表示。这种表示计算机指令的语言称为汇编语言。

例如，上述计算 1+2 的汇编语言程序如下：

```
MOV   AL,1     ; 将 1 存入寄存器 AL 中
ADD   AL,2     ; 将 2 与寄存器 AL 中的值相加，结果放在寄存器 AL 中
HLT            ; 停机
```

由此可见，汇编语言克服了机器语言难读难改的缺点，同时保持了占存储空间小、执行速度快的优点，因此许多系统软件的核心部分仍采用汇编语言编制。但是，汇编语言仍是一种面向机器的语言，每条汇编命令都一一对应于机器指令，而不同的计算机在指令长度、寻址方式、寄存器数目等方面都不一样，因此汇编语言通用性差，可读性也较差。

3. 高级语言

高级语言是更接近自然语言、数学语言的程序设计语言，是面向应用的计算机语言，与具体的机器无关。面向过程的高级语言的优点是符合人类叙述问题的习惯，而且简单易学。高级语言与计算机的硬件结构及指令系统无关，它有更强的表达能力，可方便地表示数据的运算和程序的控制结构，能更好地描述各种算法，而且容易学习和掌握，但高级语言编译生成的程序代码一般比用汇编程序语言设计的程序代码要长，执行的速度也慢。

例如，上述计算 1+2 的 BASIC 语言程序如下：

```
A=1+2     ; 将 1 加 2 的结果存入变量 A 中
PRINT A   ; 输出 A 的值
END       ; 程序结束
```

这个程序和我们平时的数学思维是相似的，非常直观易懂，容易记忆。

相对面向机器的低级语言，高级语言具有以下优点：

(1) 高级语言更接近人类自然语言，易学、易掌握。

(2) 高级语言为程序员提供了结构化程序设计的环境和工具，使设计出来的程序可读性好，可维护性强，可靠性强。

(3) 高级语言程序设计与具体的计算机硬件关系不大，因而所写的程序可移植性好，重用率高。

(4) 高级语言将繁琐的事物交给编译程序去做，自动化程度高，开发周期短。

高级语言分为面向过程和面向对象的语言两种。

1) 面向过程的高级语言

面向过程(Procedure Oriented)的程序设计是以要解决的问题为核心，分析问题中所涉及的数据及数据之间的逻辑关系(数据结构)，进而确定解决问题的方法(算法)。因此，使用面向过程的语言编程时，编程者的主要精力放在算法过程的设计和描述上。由于面向过程的程序设计语言是以要解决的问题为核心编程，因此如果问题稍微发生改变，就需要重新编写程序。在 20 世纪 70 年代和 80 年代，大多数流行的高级语言都是面向过程的程序设计语言，如 Basic、Fortran、Pascal 和 C 等。

2) 面向对象的高级语言

面向对象(Object-Oriented)的程序设计出发点是为了能更直接地描述问题域中客观存在的事物(即对象)以及它们之间的关系，是以对象作为程序基本结构单位的程序设计语言。面向对象程序设计的核心是对象，把世界上的任何一个个体都看成是一个对象，每个对象都有自己的特点。面向对象技术追求的是软件系统对现实世界的直接模拟，是将现实世界中的事物直接映射到软件系统的解空间。常见的面向对象的程序设计语言包括 C++、Python、Java 和 Smalltalk 等。

在面向对象的程序设计语言中，可以把程序描述为如下的公式：

程序 = 对象+ 消息

面向对象的编程语言使程序能够比较直接地反映客观世界的本来面目，并且使软件开发人员能够运用人类认识事物所采用的一般思维方法来进行软件开发。面向对象语言刻画客观世界较为自然，便于软件扩充与复用。

以上各种语言各有优缺点，程序员在使用时，需根据应用场合选用。一般在实时控制中，特别是在对程序的空间和时间要求很高，需要直接控制设备的场合，通常采用汇编语言；在系统程序设计、多媒体应用、数据库等诸多领域采用面向对象语言比较合适。本书采用 Python 程序设计语言为背景，介绍程序设计的基本概念和方法。

1.2 Python 语言概述

Python 是一门面向对象的解释型高级程序设计语言，是一种脚本解释语言。Python 语言诞生至今已发展为一门功能强大的通用型语言，它不仅开源，而且支持命令式编程、面向对象程序设计和函数式编程，包含丰富且易理解的标准库和扩展库，可以快速生成程序的原型，帮助开发者高效地完成任务。Python 语言在设计上坚持清晰划一的风格，借鉴了简单脚本和解释语言的易用性，它的简洁、易读、易维护以及可扩展性，使其在科学计算领域受到国内外的广泛关注。Python 语言能够与多种程序设计语言完美融合，被称为胶水语言。它能够实现与多种编程语言的无缝拼接，充分发挥各种语言的编程优势。

1.2.1 Python 语言的发展

Python 自 1989 年推出至今已有二十多年的历史，其发展成熟且稳定。第一个 Python 编译器于 1991 年诞生，它既继承了传统语言的强大性和通用性，也具有脚本解释程序的易用性。

Python 是荷兰国家数字与计算机科学研究院的研究员吉多·范罗苏姆(Guido van Rossum)创建的。1989 年圣诞节期间，在阿姆斯特丹，吉多为了打发圣诞节的无趣，决心开发一个新的脚本解释程序，做为 ABC 语言的一种继承。之所以选择 Python(大蟒蛇)作为该编程语言的名字，是因为吉多是一个叫 Monty Python 大蟒蛇飞行马戏团的爱好者。

ABC 是由吉多参加设计的一种教学语言。吉多认为 ABC 语言非常优美和强大，是专门为非专业程序员设计的。但是 ABC 语言并没有成功，究其原因，吉多认为是其非开放性造成的。吉多决心在 Python 中避免这一错误，同时实现在 ABC 中闪现过但未曾实现的东

西。可以说，Python 是从 ABC 发展起来的，主要受到了 Modula-3 的影响，并且结合了 Unix shell 和 C 的习惯。

1.2.2　Python 语言的特点

1. 面向对象

Python 是完全面向对象的语言。面向对象编程支持将特定的功能与所要处理的数据相结合，即程序围绕着对象构建。如函数、模块、数字、字符串都是对象，并且完全支持继承、重载、派生、多继承，有益于增强代码的复用性。Python 借鉴了多种语言的特性，支持重载运算符和动态类型。

2. 丰富的数据类型

除了提供整型、浮点型、布尔类型等基本数据类型之外，Python 语言还提供了列表、元组、集合、字典、字符串等复合数据类型。利用这些数据类型，可以更方便地解决实际问题，简化程序设计，缩短代码长度，并且简明易懂，方便维护。

3. 可移植性

由于 Python 的开源特性，它已经被移植在许多平台上。如果在编程时多加留意系统特性，小心地避免使用依赖于系统的特性，那么所有 Python 程序无需修改就可以在各种平台上面运行。这些平台包括 Linux、Windows、FreeBSD、Macintosh、Solaris、OS/2、Amiga、AROS、AS/400、BeOS、OS/390、z/OS、Palm OS、QNX、VMS、Psion、Acom RISC OS、VxWorks、PlayStation、Sharp Zaurus、Windows CE、PocketPC、Symbian 等。

4. 解释性

Python 是一种解释性语言，在开发过程中没有编译环节。使用 Python 语言编写的程序不需要编译成二进制代码，可直接从源代码运行程序。在计算机内部，Python 解释器把源代码转换成近似机器语言的中间形式字节码，然后再把它翻译成计算机使用的机器语言并运行，使 Python 程序更简单、更加易于移植，从而改善了 Python 的性能。

5. 可扩展性

如果需要一段关键代码运行更快或者希望某些算法不公开，可以部分程序用 C 或 C++ 编写，然后在 Python 程序中使用它们，从而实现对 Python 程序的扩展。Python 本身被设计为可扩充的，提供了丰富的 API 和工具，其标准实现是使用 C 完成的(CPython)，程序员能够轻松地使用 C 语言、C++来编写 Python 扩充模块，缩短开发周期。Python 编译器本身也可以被集成到其他需要脚本语言的程序内，因此很多人还把 Python 作为一种“胶水语言”(Glue Language)使用，可以用 Python 将其他语言编写的程序进行集成和封装。

6. 丰富的库

Python 有数百个标准库模块，包括 sys 模块、os 系统模块、re 模式匹配模块、字符串模块等。标准库可以帮助用户快速实现一些功能，不必重复开发已有的代码，可以提高效率和代码质量。除了标准库，Python 还提供了大量高质量第三方库，可以在 Python 包索引找到它们。Python 的第三方库使用方式与标准库类似，功能强大，提供了数据挖掘、大数据分析、图像处理等功能。

7. 健壮性

Python 提供了安全合理的异常退出机制，能捕获程序的异常情况，允许程序员在错误发生的时候根据出错条件提供处理机制。一旦异常发生，Python 解释器会转出一个包含使程序发生异常的全部可用信息到堆栈并进行跟踪，此时程序员可以通过 Python 监控这些异常并采取相应措施。

1.3 Python 语言开发环境

Python 是跨平台编程语言，可以兼容很多平台。这里以 Windows 平台为例，介绍下载和安装 Python 开发环境的方法。

1.3.1 Windows 环境下安装 Python 开发环境

(1) 在 Python 官网 https://www.python.org/下载安装包，选择 Windows 平台下的安装包，如图 1.1 所示。

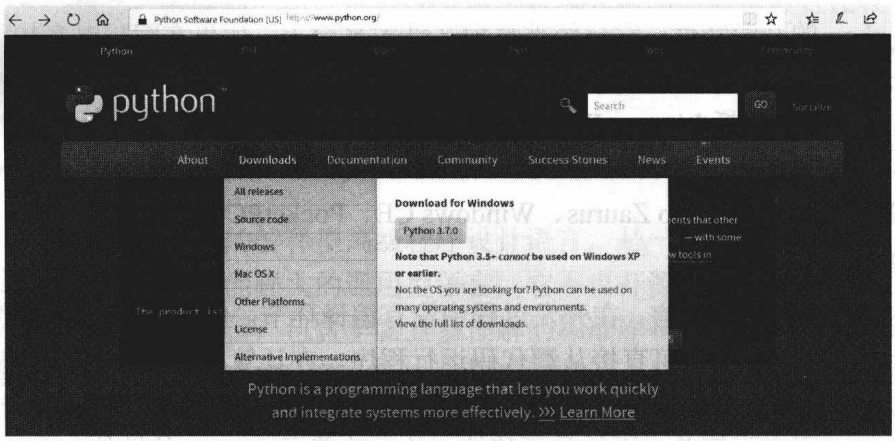

图 1.1 Python 安装包下载

(2) 单击图 1.1 中的 Python 3.7.0 下载，下载后的文件名为 Python-3.7.0.exe。双击该文件，进入 Python 安装界面，如图 1.2 所示。

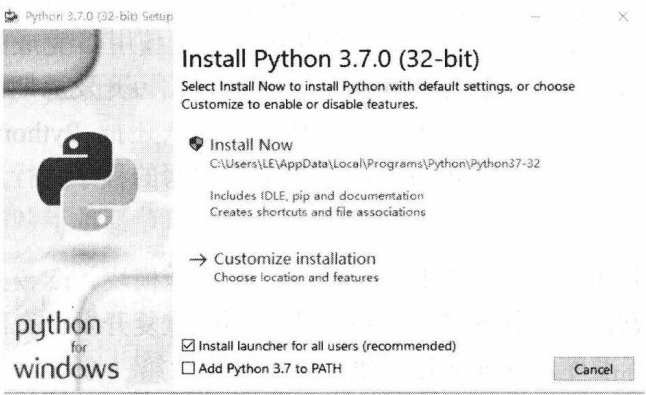

图 1.2 选择安装方式

在图 1.2 中，提示有两种安装方式。第一种是采用默认的安装方式；第二种是自定义方式，可以选择软件的安装路径及安装包。这两种安装方式可任选其一。

(3) 这里选择第一种安装方式，安装过程如图 1.3 所示。

(4) 安装成功后，提示信息如图 1.4 所示。

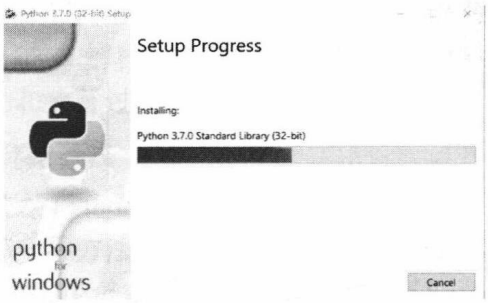

图 1.3　Python 安装过程

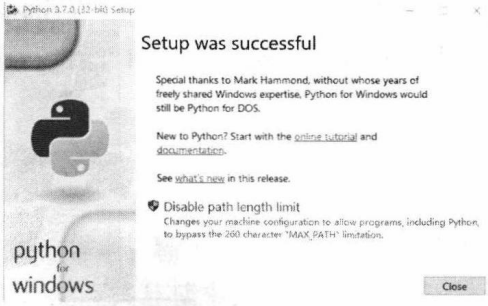

图 1.4　安装成功的信息提示

注意：在图 1.2 中选择安装方式时，最下面有个选项【Add Python 3.7 to PATH】，如果勾选了该选项，那么后续配置环境的步骤可以省略。如果没有勾选，安装完 Python 之后就需要手动配置环境变量。

(5) 手动添加环境变量。鼠标右击【计算机】→【属性】→【高级】，弹出如图 1.5 所示【系统属性】对话框。

(6) 单击图 1.5 中的【环境变量】，在弹出的【环境变量】对话框中，选择环境变量中的【Path】，如图 1.6 所示。

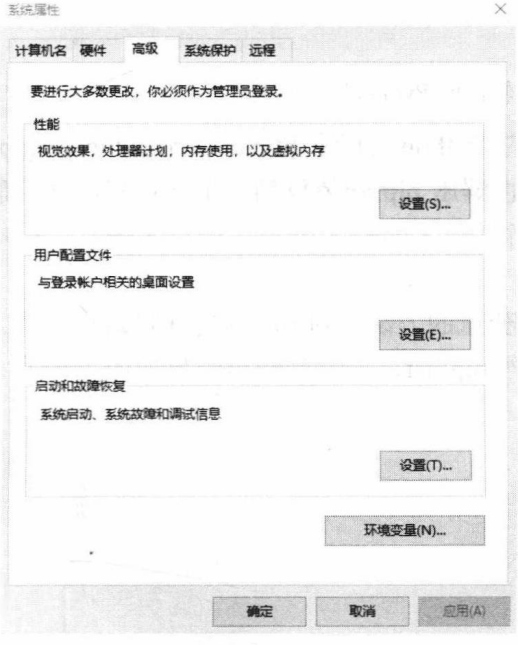

图 1.5　系统属性设置图

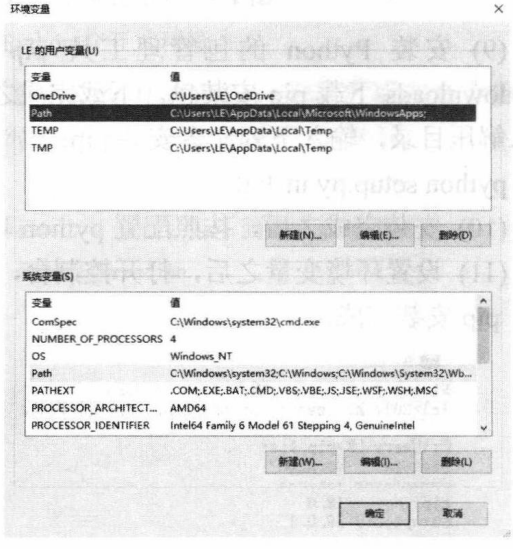

图 1.6　设置环境变量

(7) 单击图 1.6 中的【编辑】按钮，弹出【编辑环境变量】对话框，如图 1.7 所示。单击图 1.7 中的【新建】按钮，在增加的一行编辑框中输入 Python 的安装路径，如图 1.8 所示。最后单击【确定】按钮，完成环境变量的配置。

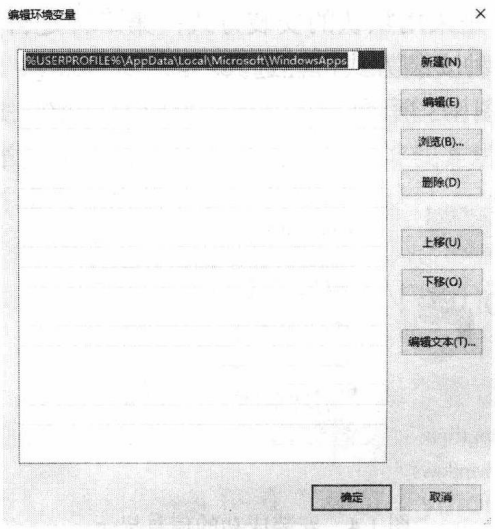

图 1.7　编辑环境变量

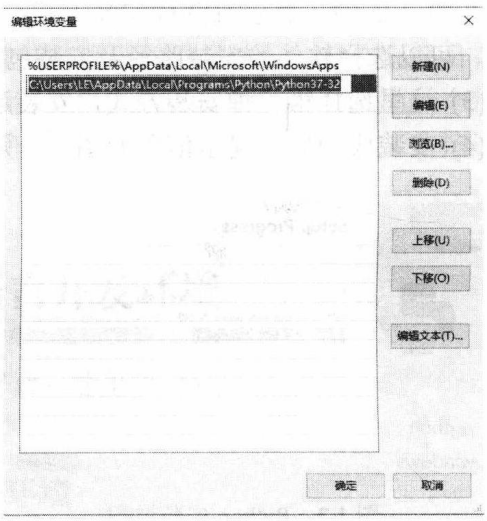

图 1.8　新建环境变量

(8) 此时，在控制台输入 Python 命令，系统会输出 Python 的版本信息，如图 1.9 所示。

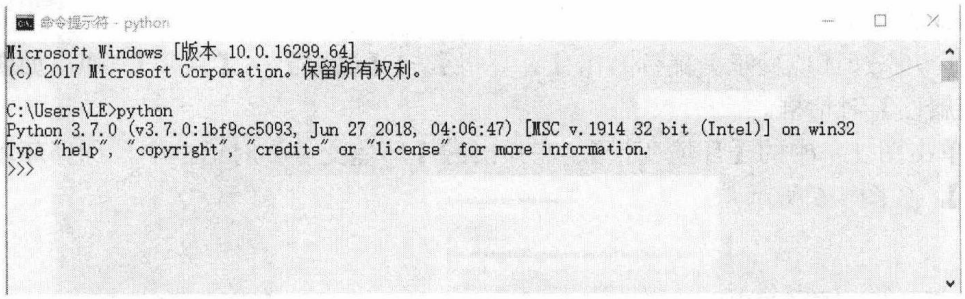

图 1.9　环境变量配置成功后输出的 Python 版本信息

(9) 安装 Python 的包管理工具 pip。在 Python 官网 https://pypi.python.org/pypi/pip#downloads 下载 pip 安装包。下载完成之后，解压 pip 安装包到一个文件夹，从控制台进入解压目录，输入下列命令安装 pip：

```
python setup.py install
```

(10) 安装完成之后，按照配置 python 环境变量的方法，对 pip 环境变量进行设置。

(11) 设置环境变量之后，打开控制台，输入 pip list，控制台输出结果如图 1.10 所示，此时 pip 安装完成。

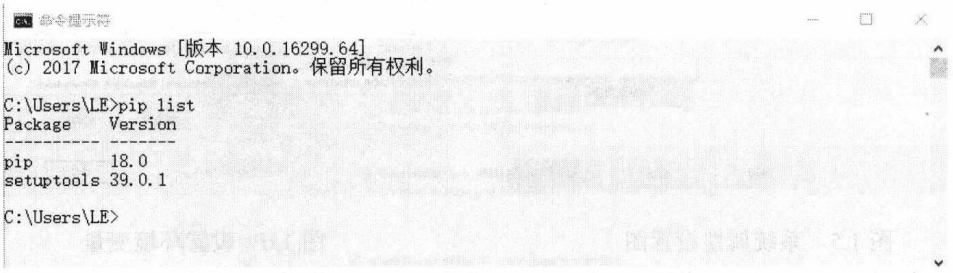

图 1.10　pip 安装及配置成功的输出结果

1.3.2　运行第一个 Python 程序

完成 Python 的安装之后，就可以开始编写 Python 代码并运行 Python 程序了。Python 程序主要的运行方式有交互式和文件式两种。交互式是指 Python 解释器及时响应用户输入的每条代码，给出运行结果。文件式是指用户将 Python 程序写入一个或多个文件中，然后启动 Python 解释器批量执行文件中的代码。下面以输出“Hello World”为例说明两种方法的启动和执行过程。

1. 交互式

进入 Python 交互性环境有两种方法。

第一种方法是启动 Windows 操作系统，打开开始菜单，输入 cmd 之后，进入命令行窗口，在控制台中输入“Python”，键入【Enter】键进入交互式环境中，在命令提示符“>>>”后输入如下程序代码：

```
print("Hello World!")
```

按【Enter】键执行，得到运行结果，如图 1.11 所示。

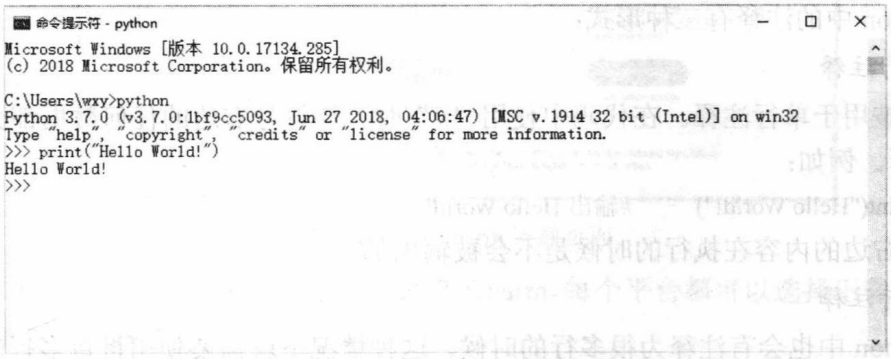

图 1.11　通过命令行启动 Python 交互式环境

第二种方法是调用安装的 Python 自带的 IDLE 启动交互式窗口。启动之后在命令提示符“>>>”后输入代码，再按【Enter】键执行，得到的运行结果如图 1.12 所示。

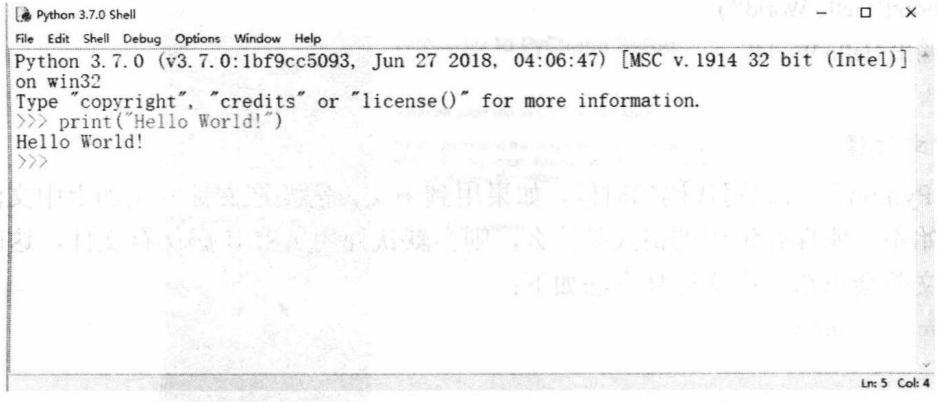

图 1.12　通过 IDLE 启动 Python 交互式环境

在交互式环境中，输入的代码不会被保存下来。关闭 Python 得到运行窗口之后，之前输入的代码不会被保存。在交互式环境中按下键盘中的【↑】【↓】键，可以寻找历史命令，