巍巍交大　百年书香
www.jiaodapress.com.cn
bookinfo@sjtu.edu.cn

策划编辑　高　锐
责任编辑　胡思佳
封面设计　张瑞阳

Python程序设计简明教程

免费提供 精品教学资料包
服务热线：400-615-1233
www.huatengedu.com.cn

扫描二维码
关注上海交通大学出版社
官方微信

ISBN 978-7-313-31514-4
9 787313 315144 >
定价：49.90元

主编　张春飞

上海交通大学出版社

Python程序设计简明教程

高等教育计算机类课程精品教材

Python程序设计
简明教程

主编　张春飞

上海交通大学出版社
SHANGHAI JIAO TONG UNIVERSITY PRESS

高等教育计算机类课程精品教材

Python程序设计

简明教程

主　编　张春飞
副主编　曹婧华　刘　通　段云娜
　　李晓峰　李慧春

上海交通大学出版社
SHANGHAI JIAO TONG UNIVERSITY PRESS

内容提要

本书分为10章，内容包括Python语言概述、初识Python、选择结构与循环结构、Python组合数据类型、函数、面向对象程序设计、模块及第三方库、文件及目录操作、常用数据分析工具、数据可视化分析。本书内容组织合理，涵盖了Python程序设计语言的全部语法点，并力求反映Python语言的重要应用领域。

本书既适合作为高等教育Python程序设计课程的教材，也可供广大Python程序设计爱好者学习使用。

图书在版编目（CIP）数据

Python程序设计简明教程 / 张春飞主编. -- 上海 ：上海交通大学出版社，2024.9. -- ISBN 978-7-313-31514-4

Ⅰ. TP312.8

中国国家版本馆CIP数据核字第2024D3S362号

Python程序设计简明教程

Python CHENGXU SHEJI JIANMING JIAOCHENG

主　　编：张春飞

出版发行：上海交通大学出版社　　地　　址：上海市番禺路951号

邮政编码：200030　　电　　话：021-64071208

印　　制：三河市骏杰印刷有限公司　　经　　销：全国新华书店

开　　本：850 mm×1168 mm　1/16　　印　　张：16

字　　数：430千字

版　　次：2024年9月第1版　　印　　次：2024年9月第1次印刷

书　　号：ISBN 978-7-313-31514-4　　电子书号：ISBN 978-7-89424-887-9

定　　价：49.90元

版权所有　侵权必究

告读者：如您发现本书有印装质量问题请与印刷厂质量科联系

联系电话：0316-3662258

前　言

Preface

Python 是由荷兰人 Guido van Rossum 于 20 世纪 80 年代末至 90 年代初设计出来的，是一种面向对象的解释型高级编程语言，它以“优雅、明确、简单”的设计理念迅速捕获了众多编程爱好者的心。Python 已形成了全球范围最大的单一语言编程社区，超过 9 万个第三方编程库，覆盖了从数据到智能、从文本处理到虚拟现实、从控制逻辑到系统结构等几乎所有的计算领域，因此受到了全球学习者的高度关注，近年来不仅在自然科学领域，而且在人文社科领域，都掀起了学习 Python 的热潮。

本书就是为了适应这种需求而编写的。其宗旨是在全面讲授 Python 程序设计语言基本语法的基础上，引入大量的案例，从不同知识点和不同角度引导学生进行学习，强调理论与实践相结合，注重培养学生的计算思维能力，进而从根本上提升学生的理论水平和运用理论指导实践的能力。

在遵循 Python 程序设计语言结构完整性的基础上，编者结合当代大学生的认知能力特点和社会需求，将全书分为 10 章，内容涵盖了 Python 程序设计语言的全部语法点，并力求反映 Python 语言的重要应用领域。通篇语言简洁明了，结构组织合理，内容贴近实际，易于学生学习和理解。本书具有以下主要特点。

（1）内容全面。几乎涵盖了 Python 语言的全部语法点，因此能够满足不同专业和不同层次学生的学习需求。

（2）组织合理。本书的编者均是长期从事一线教学和科研工作的教师，对 Python 语言的学习规律和学生的认知能力有较为深刻的理解，因而在内容组织上能够做到循序渐进。

（3）注重实践。学习程序设计语言的目的在于利用计算机解决实际问题，因此本书在编写过程中配备了丰富的例题，覆盖面广，针对性强。

（4）体现先进教学理念。本书在内容上按“点—线—面—体”的方式组织，在应用上按“由小到大，由浅入深，由单一到综合”的方式逐步扩展。

由于 Python 语言本身具有灵活性，初学者可不必过多拘泥于细节，应把主要精力放在最基本和最常用的内容上。随着学习的逐渐深入和实践经验的不断积累，很多内容自然就能够理解了，而有些知识点则需要长期的编程实践才能真正掌握。

本书由吉林大学张春飞任主编，吉林大学曹婧华、刘通、段云娜、李晓峰和李慧春任副主编。编写分工如下：第 1 章和第 3 章由张春飞编写，第 2 章和第 4 章由曹婧华编写，第 5 章和第 6 章由刘通编写，第 7 章和第 8 章由段云娜编写，第 9 章由李慧春编写，

第 10 章由李晓峰编写，全书由张春飞统稿。

在编写本书的过程中，编者得到了许多同行的帮助，还参阅了众多的相关资料，在此向各位同行和相关资料的作者表示诚挚的感谢。

尽管本书的编者都是长期从事一线教学及科研工作的教师，但由于 Python 语言本身在不断发展和完善，加之编者水平有限，书中难免存在疏漏之处，恳请广大读者给予批评和指正。

编　者

目　录

Contents

第 1 章　Python 语言概述 …… 1

1.1　程序设计语言 …… 1
1.2　Python 语言简介 …… 3
1.3　Python 语言的开发环境与工具 …… 6
1.4　第一个 Python 程序 …… 11
本章小结 …… 13
习题 1 …… 13

第 2 章　初识 Python …… 15

2.1　Python 的书写特点 …… 16
2.2　Python 的变量 …… 18
2.3　Python 的基本数据类型 …… 22
2.4　Python 的运算符和表达式 …… 27
2.5　Python 的关键字、内置函数及常用标准库函数 …… 36
2.6　顺序结构程序设计实例 …… 43
本章小结 …… 44
习题 2 …… 44

第 3 章　选择结构与循环结构 …… 46

3.1　选择结构 …… 46
3.2　循环结构 …… 51
3.3　跳转语句 …… 56
3.4　异常处理 …… 61
3.5　应用实例 …… 66
本章小结 …… 68
习题 3 …… 68

第 4 章　Python 组合数据类型 …… 71

4.1　字符串 …… 71
4.2　列表 …… 87
4.3　元组 …… 101
4.4　字典 …… 103

4.5 集合对象 …… 110
4.6 应用实例 …… 114
本章小结 …… 115
习题 4 …… 115

第 5 章 函数 …… 118

5.1 函数的定义 …… 118
5.2 名字的作用域 …… 128
5.3 递归 …… 130
5.4 匿名函数与生成器 …… 134
5.5 应用实例 …… 137
本章小结 …… 138
习题 5 …… 138

第 6 章 面向对象程序设计 …… 141

6.1 基本概念 …… 141
6.2 类的定义与对象的创建 …… 142
6.3 属性和方法 …… 146
6.4 继承、重载与魔术方法 …… 151
6.5 应用实例 …… 156
本章小结 …… 158
习题 6 …… 158

第 7 章 模块及第三方库 …… 162

7.1 自定义模块 …… 162
7.2 常用内置模块 …… 173
7.3 第三方库 …… 178
7.4 应用实例 …… 183
本章小结 …… 185
习题 7 …… 185

第 8 章 文件及目录操作 …… 187

8.1 文件对象 …… 187
8.2 基本文件操作 …… 189
8.3 目录操作 …… 205
本章小结 …… 215
习题 8 …… 215

第 9 章　常用数据分析工具 …… 217

9.1　数据分析简介 …… 217
9.2　NumPy 模块 …… 217
9.3　Pandas 模块 …… 224
9.4　数据库模块 …… 230
9.5　应用实例 …… 234
本章小结 …… 238
习题 9 …… 238

第 10 章　数据可视化分析 …… 240

10.1　数据可视化原理 …… 240
10.2　turtle 库 …… 241
10.3　Matplotlib 库 …… 244
10.4　常见图表的绘制实例 …… 246
本章小结 …… 247
习题 10 …… 247

参考文献 …… 248

第 1 章

Python 语言概述

作为 20 世纪人类最伟大的发明之一，计算机已经或正在影响和改变着人们的生产、生活方式，几乎在每个领域都可以看到它的影子，而与之同行的计算机程序设计语言也如雨后春笋般发展起来，先后出现了机器语言、汇编语言和高级语言。

Python 语言是诸多程序设计语言中一颗耀眼的新星，它以“优雅、明确、简单”的设计理念迅速捕获了众多编程爱好者的心。Python 常常作为“胶水”语言来组合其他语言编写的功能模块，多用于科学计算、大数据、人工智能、金融科技等领域，成为当今程序设计语言的主流。

1.1 程序设计语言

程序是计算机实现自动运算的基础，而程序设计语言则是程序员实现程序的基本工具。程序设计语言是计算机能够理解和识别用户操作意图的一种交互体系，它按照特定规则组织计算机指令，使计算机能够自动进行各种运算处理。

1.1.1 程序设计语言概述

人与人之间交换信息最重要的手段是语言：口头语言、文字语言或图形文字语言。但不同国家的人和不同民族的人有不同的语言，人们将这些语言统称为自然语言。目前，通用的计算机不能识别自然语言，只能识别特定的计算机语言。计算机语言即程序设计语言，是程序设计的工具，按其发展顺序，先后出现了机器语言、汇编语言和高级语言。

1. 机器语言

机器语言是一种二进制语言，它采用二进制形式（0 和 1）表示数据和指令，是计算机硬件可以直接识别和执行的程序设计语言。机器语言是一种面向机器的计算机语言，与计算机型号有关，每种计算机只能识别自己的机器语言。机器语言中最小的执行单位是指令，每条指令完成一个最基本的功能，由两部分组成，即操作码和操作数。操作码指明指令的性质，也就是进行什么样的运算，操作数指明操作的对象，也就是对谁做运算。例如，11010010 0111011 就是一条机器指令，表示的是在 16 位计算机上执行数字 3 和 2 的加法。

用机器语言编写的计算机程序可以被计算机直接识别和处理，运算效率较高，但是缺点也

是显而易见的。第一，这种程序只能在特定的计算机上使用，可移植性较差。第二，机器指令本身很难表达出指令的含义，对于编程者来说记忆困难。第三，每条机器指令的功能过于单一，编程效率低。

2. 汇编语言

机器语言的缺点制约了计算机的使用效率和使用范围，因此出现了汇编语言。汇编语言用具有一定含义的符号即助记符表示机器指令，从而方便编程人员记忆和书写，提高了程序的编写效率。例如，“ADD　AX, 2”是一条汇编指令，通常表示将寄存器 AX 中存储的数据加上 2 再存入寄存器 AX 中。从指令本身可以直接看出这条指令能够完成加法操作。

与机器语言相比，汇编语言容易记忆，含义明显，便于编程人员使用。但是，用汇编语言编写的程序，计算机不能直接识别与处理，必须经过一种处理将其转换成二进制的机器指令执行，这种处理过程称为汇编，完成这种处理功能的程序称为汇编程序。汇编语言中的汇编指令与机器语言中的机器指令是一一对应的，汇编过程主要就是将助记符式的汇编指令转换为二进制的机器指令的过程，因此，汇编语言也是机器相关的计算机语言，与机器语言统称低级语言。

3. 高级语言

高级语言的出现使计算机语言脱离了对计算机硬件的依赖，高级语言成为一种接近于人类自然语言的计算机语言，可以更容易地描述计算问题并利用计算机解决计算问题，从而为高效编程与计算机的广泛应用提供了可能，是计算机语言发展史上的一次飞跃。例如，“y=x+1”就是一条高级语言的语句，表示将变量 x 的值加上 1 后存入变量 y 中，显然，这个代码只与编程语言有关，与计算机结构无关，同一种编程语言在不同计算机上的表达方式是一致的。

与汇编语言相比，高级语言具有以下优点。

（1）高级语言通常有一套特定的语法，这个语法与具体的计算机系统无关，用高级语言书写的程序，可以独立于具体的计算机系统。也就是说，使用高级语言书写的程序，几乎不需做任何修改，就可以运行在支持该语言的计算机系统上，可移植性较好。

（2）高级语言更接近于人类常用的表述方式，语句的含义更加明显，容易学习与记忆。

（3）一条高级语言的语句所完成的功能相当于多条机器指令，编程效率高。

第一个得到广泛应用的高级语言是 1972 年诞生的 C 语言，其他常见的程序设计高级语言有 BASIC 语言、FORTRAN 语言、COBOL 语言、PASCAL 语言、Ada 语言、LISP 语言、Java 语言、C++ 语言和 Python 语言等。这些语言都有各自的特点和适用领域。例如，FORTRAN 适用于数值计算，COBOL 适用于商业管理，C 语言适用于编写系统软件等。

一般来说，通用编程语言比专用于某些领域的编程语言生命力更强，它能够用于编写多种用途的程序，具有更加广阔的生存空间，这也是其能长盛不衰的根本原因。Python 语言就是一种功能强大的通用编程语言，也是近些年来诸多编程爱好者竞相追逐的对象。

视频：高级语言翻译机器语言的方式

1.1.2　编译和解释

高级语言编写的程序计算机是无法直接执行的，必须翻译成机器语言所对

应的指令序列才能被计算机执行。根据翻译的方式不同，可以分为编译和解释两种。编译是将源代码转换成目标代码的过程，通常源代码是高级语言编写的代码，目标代码是机器语言对应的代码，执行编译功能的计算机程序称为编译器。编译方式的程序处理过程如图 1-1 所示。

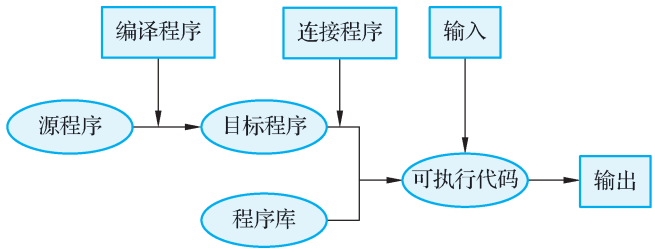

图 1-1　编译方式的程序处理过程

编译方式用相应语言的编译器对高级语言描述的源代码进行若干次扫描（如语法分析、语义分析、生成各种对照表、生成中间代码、优化等）后生成目标代码。但目标代码并不能直接执行，还需要经过连接、装配成可执行指令序列后才能运行，进而得到运算结果。

解释方式是用解释器对源代码逐条扫描、翻译并转换成目标代码，同时逐条执行目标代码的过程。解释方式的程序处理过程如图 1-2 所示。

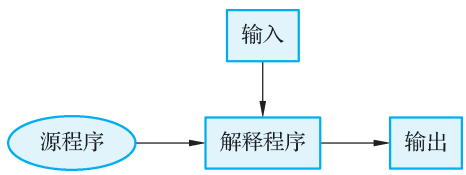

图 1-2　解释方式的程序处理过程

编译方式和解释方式的根本区别在于，编译是一次性地翻译，一旦程序被编译，只要源代码不被修改，则不再需要编译程序或源代码。而解释方式则在每次程序运行时都需要解释器和源代码。两者的区别如果用文学翻译类比，编译方式相当于笔译，译成一篇完整的可反复阅读的译文，解释方式相当于同声传译，边听边译。一般而言，编译方式适合于结构复杂、要求运行效率较高的程序设计语言，解释方式则更适合于交互式程序设计语言的翻译。

采用编译方式执行的编程语言是静态语言，如 C 语言、Pascal 语言、Java 语言等；采用解释方式执行的编程语言是脚本语言，如 PHP 语言、JavaScript 语言等。Python 语言是一种被广泛使用的高级通用脚本编程语言，虽然采用解释执行方式，但它的解释器也具有编译器的部分功能，随着程序的运行，解释器也会生成一个完整的目标代码。

1.2　Python 语言简介

Python 是近些年非常流行的程序设计语言之一，它是最接近人类自然语言的计算机语言，也是全球排名前三的编程语言。Python 因其语言的简洁性、易读性及可扩展性，得到了越来越多的科学家、工程技术人员及编程爱好者的青睐，在众多的领域都得到了极其广泛的应用。

视频：了解 Python 语言

1.2.1 Python 语言的起源

Python 是由 Guido van Rossum（图 1–3）在 20 世纪 80 年代末至 90 年代初在荷兰国家数学和计算机科学研究所设计出来的。“Python”的英文直译是“大蟒蛇”，这个名字来自 Guido 非常喜欢的由 BBC 播放的室内情景幽默剧 *Montry Python's Flying Circus*。

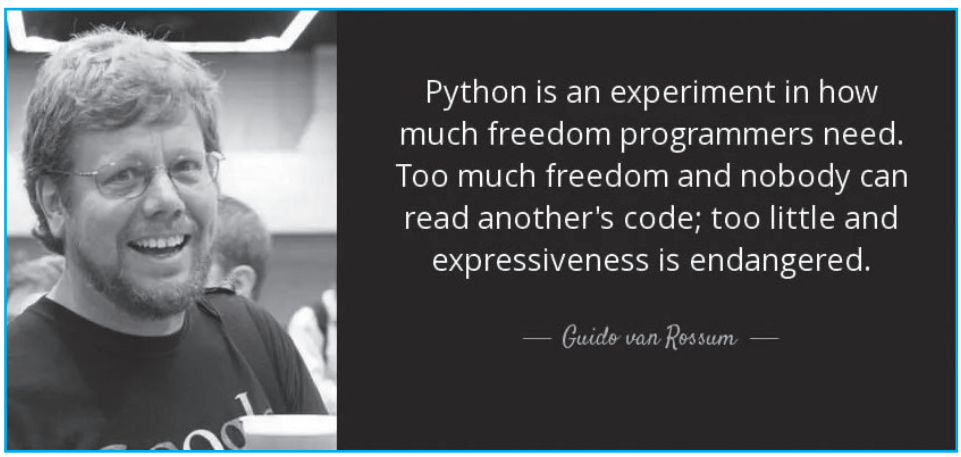

图 1–3 Guido van Rossum

Python 的设计哲学是简单、明确和优雅，实际上，Python 也始终贯彻着这一理念。它既拥有简单脚本语言和解释型程序设计语言的易用性，也拥有传统编译型程序设计语言强大、通用的功能，被当成模块导入的 Python 文件在经过解释器编译后生成的字节码对象 pyc 文件可以实现 Python 源代码不修改，无须编译即可由 Python 解释器逐条解释执行。它介于 C 和 Shell 之间，非常好学好用，有十分丰富的数据类型和全面的功能。Python 的优雅则在其表达方式和语法形式等多个方面均有体现。

Python 本身也是由诸多其他语言发展而来的，包括 ABC、Modula–3、C、C++、ALGOL 68、SmallTalk、UNIX Shell 和其他脚本语言等。现在 Python 由一个核心开发团队在维护，Guido van Rossum 仍然占据着至关重要的作用，并指导其发展。Python 自发布以来，主要有三个版本：1994 年发布的 Python 1.0 版本（已过时）、2000 年发布的 Python 2.0 版本（到 2020 年 4 月更新到 2.7.18，现在已经停止更新）和 2008 年发布的 3.0 版本（2023 年 12 月已经更新到 3.12.1）。

1.2.2 Python 语言的特点

“人生苦短，我用 Python”作为网络上广泛流传的格言，揭示了 Python 语言无可比拟的开发效率。在众多的高级程序设计语言中，Python 语言之所以能够存在和发展，并具有旺盛的生命力，主要源于其自身的诸多优良特性。具体来说，Python 语言具有以下特点。

（1）语法简洁。Python 语言中的关键字很少，并且不需要使用分号、大括号等标记符号，使用循环和条件语句进行数据结构初始化。因此实现相同的功能，Python 语言的代码行数仅相当于其他语言的 1/5~1/3。

（2）与平台无关。Python 语言作为一种脚本语言，具有跨平台的特性。Python 程序可以在任何安装解释器的计算机环境中运行，因此，用该语言编写的程序可以不经任何修改地实现跨

平台运行。

（3）开源理念。对于高级程序员，Python 语言开源的解释器和函数库具有强大的吸引力，更重要的是，Python 语言倡导的开源软件理念为该语言的快速发展奠定了坚实的群众基础，使其具有良好的编程生态。

（4）黏性扩展。Python 语言在设计之初，就将编程语言的可扩展性考虑其中，具有优异的扩展性。它可以集成 C、C++、Java 等语言编写的程序代码，通过接口和函数库等方式将它们“粘起来”。此外，Python 语言本身提供了良好的语法和执行扩展接口，能够整合各类程序代码。

（5）模式多样。尽管 Python 3.x 解释器内部采用面向对象方式实现，但 Python 语法层面却能够同时支持面向过程和面向对象两种编程方式，这也就为编程人员提供了灵活的编程模式。

（6）支持中文。Python 3.x 解释器采用 UTF-8 编码表示所有字符信息。UTF-8 编码可以表达英文、中文、韩文、法文等各类语言，因此，Python 程序在处理中文时更加灵活且高效。

（7）通用灵活。Python 语言是一个通用编程语言，可以用于编写各领域的应用程序，这为该语言提供了广阔的应用空间。从科学计算、数据处理到人工智能、机器人几乎各类应用，Python 语言都能够发挥重要作用。

（8）类库丰富。Python 解释器不仅提供了丰富的内置类和库函数，还拥有大量的第三方类库可以使用。这使得 Python 语言不管是在科学计算和游戏开发，还是机器学习与图像处理等领域，都具有无可比拟的优势。另外，其由于易用性和健壮性，受到越来越多编程人士的喜爱，不仅有数量庞大的商业机构在为 Python 构建功能强大的类库，而且有数不清的 Python 爱好者通过开源平台为 Python 社区提供力量，具有良好的编程生态。

正是基于以上优良的特性，Python 语言短时间内在竞争激烈的编程市场中异军突起，迅速占领各个相关领域。2010 年，Python 荣膺 TIOBE 2010 年度语言桂冠；2020 年，在 IEEE Spectrum 发布的 2020 年度编程语言排行榜中，Python 位居第一名。

1.2.3 Python 语言的应用领域

Python 作为一种功能强大的编程语言因其优良的特性在众多领域站稳脚跟，并得到了迅速的发展，概括起来主要有以下几个应用领域。

（1）Web 开发。

（2）大数据分析。

（3）人工智能。

（4）自动化运维。

（5）云计算。

（6）爬虫。

（7）游戏开发。

从目前关注度最高的人工智能和大数据分析领域来看，Python 语言无论在机器学习、数据

分析、深度学习还是人工神经网络方面都是主流的编程语言。而且基于大数据分析和深度学习发展而来的人工智能本质上也已经无法离开 Python 语言的支持。目前世界上优秀的人工智能学习框架，如 Google 的 TensorFlow、Facebook 的 PyTorch 和开源社区的神经网络库 Karas 等都是用 Python 语言实现的。因此，从某种程度上来说，掌握了 Python 语言就等于拥有了在时代潮流中弄潮的双桨。

1.3 Python 语言的开发环境与工具

Python 不仅安装和配置开发环境非常简单，而且可以进行跨平台安装和开发。它可以在多个操作系统上进行编程，编写好的程序也可以在不同系统上运行。常用的操作系统及其说明如表 1-1 所示。

表 1-1 进行 Python 开发常用的操作系统及其说明

操 作 系 统	说　明
Windows	推荐使用 Windows 10 或以上版本
macOS	从 macOS 10.3(Panther) 开始已经包含 Python
Linux	推荐 Ubuntu 版本

1.3.1 Python 开发环境的安装

Python 语言是开源项目的优秀代表，其解释器的全部代码都是开源的，可以在 Python 语言的官方网站（https://www.python.org/）免费下载，如图 1-4 所示。在下载界面可以根据自己的操作系统版本选择相应的 Python 3.x 系列进行安装。

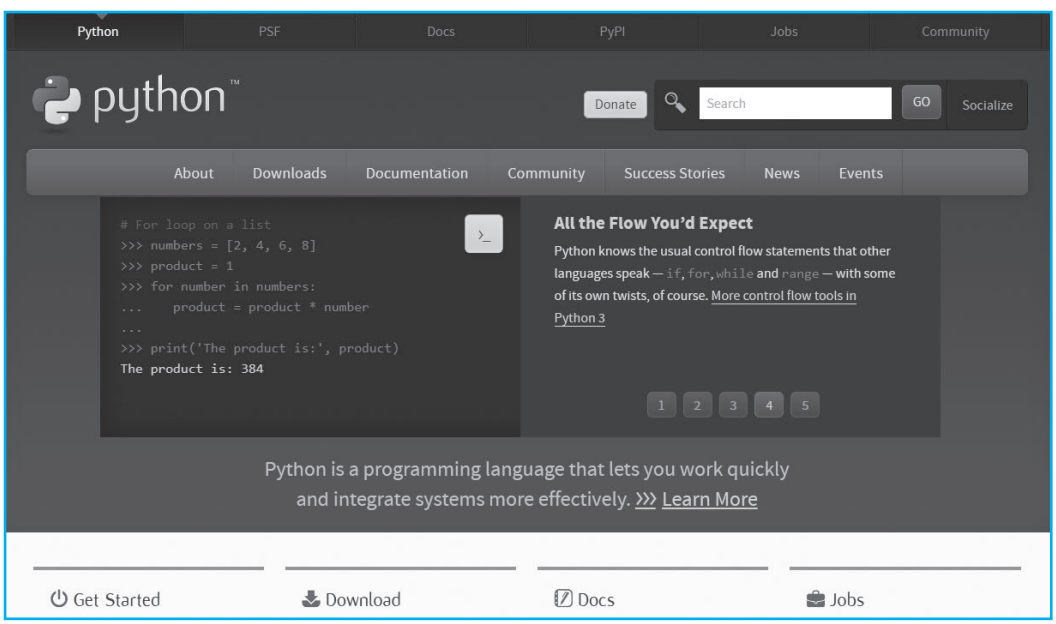

图 1-4 Python 官方网站首页

在 Windows 平台安装 Python 的步骤如下。

（1）在 Python 官方网站选择 Downloads 页面的 Windows 平台，如图 1–5 所示。此处以 Python 3.11.4 为例。

（2）双击下载后得到的安装文件 Python–3.11.4–amd64.exe，将显示安装向导对话框，选择“Add python.exe to PATH”复选框，表示将自动配置环境变量，如果不选，则需要手动设置环境变量，如图 1–6 所示。

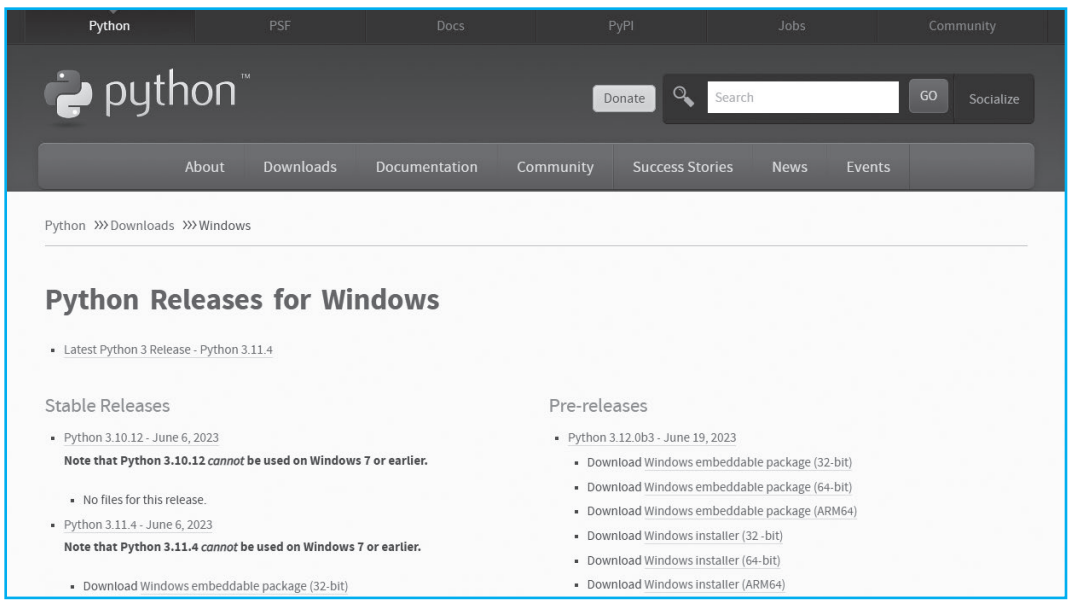

图 1–5　适合 Windows 系统的 Python 下载版本

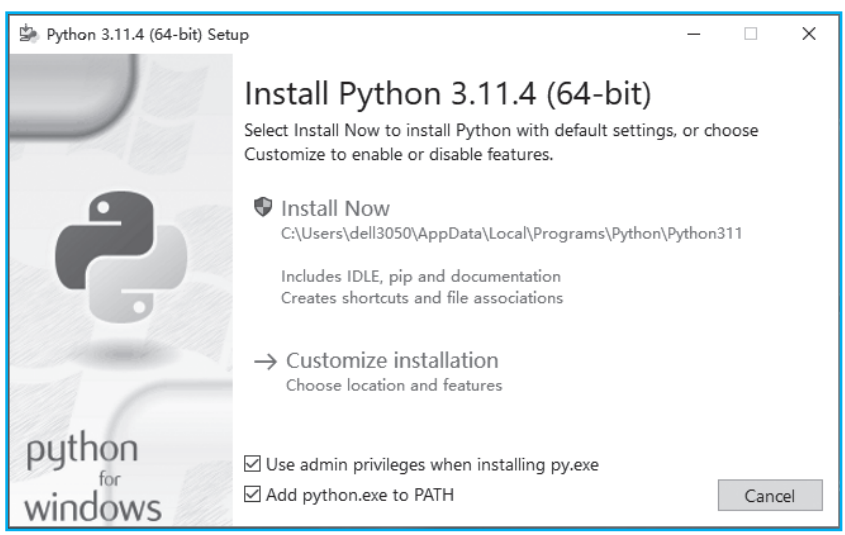

图 1–6　Python 安装向导

（3）单击“Customize installation”链接，进行自定义安装，可以修改安装路径，在弹出的“Optional Features”界面中可以直接选用默认设置，如图 1–7 所示。

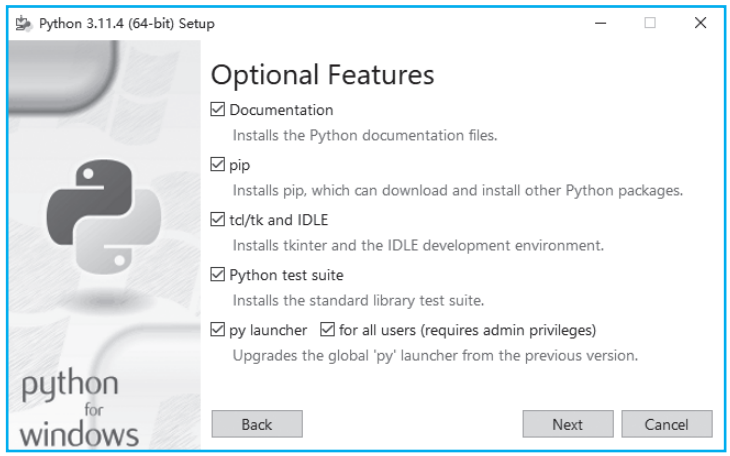

图 1–7　设置安装选项

（4）单击“Next”按钮，将打开“Advanced Options”界面，在该界面中选择“Install Python 3.11 for all users”复选框，同时可以设置安装路径为“C:\Program Files\Python311”，如图 1–8 所示。

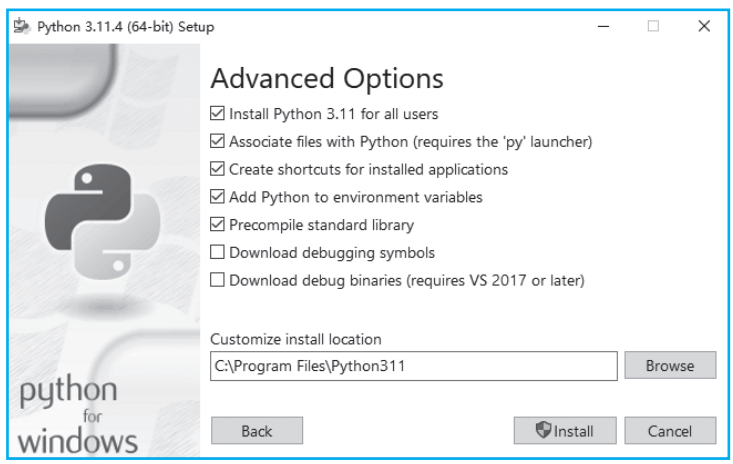

图 1–8　“Advanced Options”界面

（5）单击“Install”按钮，开始安装 Python，安装完成后将显示图 1–9 所示的界面。

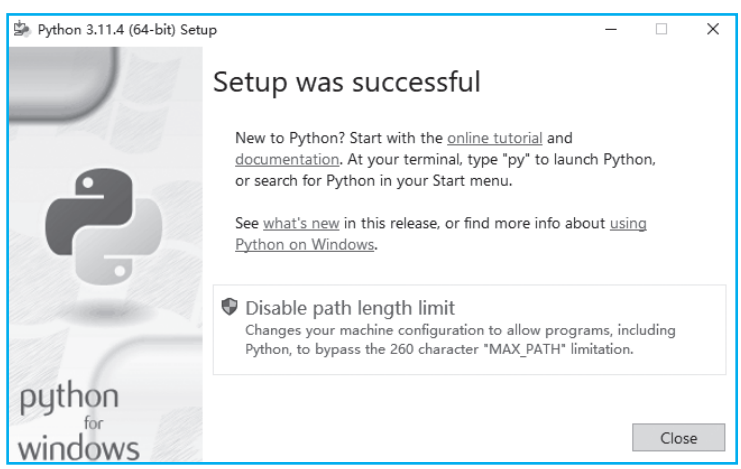

图 1–9　安装完成界面

（6）Python 安装完成后，还需要检测 Python 是否成功安装。在 Windows“开始”搜索框中输入“cmd”并按“Enter”键，在弹出的命令提示符窗口中执行“python”命令，若出现图 1-10 所示的信息，则说明 Python 安装成功。至此，就可以开始 Python 程序设计之旅了。

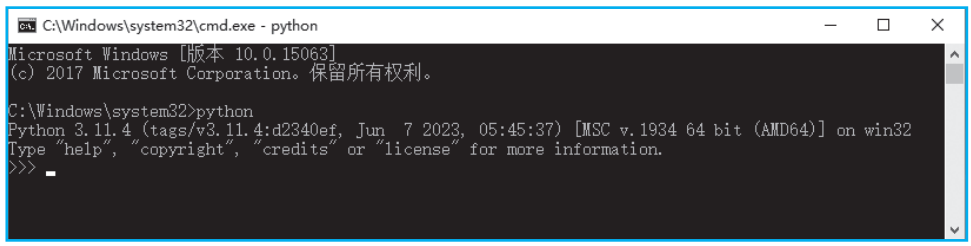

图 1-10 安装成功的控制台输出信息

1.3.2 Python 的集成开发环境

Python 的开发工具有很多，其中一部分是文本编辑器，还有一部分是集成开发环境（integrated development environment，IDE）。IDE 是用于程序开发的软件，一般包括代码编辑器、解释器、调试器和图形用户界面工具。IDE 为用户使用编程语言开发项目提供了很大的便利。Python 常用的集成开发环境包括 IDLE、Anaconda、PyCharm 和 Eclipse 等。

IDLE 是 Python 官方内置的一个简单、小巧的 IDE，它包含了编辑器、交互式命令行和调试器等基本组件，可以应付大多数 Python 的简单应用，但不适合开发大型项目。本节主要介绍几种功能更加丰富的 IDE。

1. Anaconda

Anaconda 不仅包含 Spyder 和 Jupyter Notebook 这样强大的编辑器，支持编辑、运行和调试功能，而且最重要的是集成了大量流行的科学计算和数据分析第三方库，因此 Anaconda 的官方定位是一个 Python 数据科学平台。

Anaconda 中的 Spyder 编辑器的操作界面如图 1-11 所示，Spyder 的默认主窗口分为左、右上和右下 3 部分，分别是代码编辑窗口，变量、图、文件等的观察窗口和控制台窗口。这种将几个功能窗口组合在同一主窗口的方式比单一窗口更便于程序的编辑、运行和调试。

2. PyCharm

PyCharm 是一款非常好用的 Python IDE，使用 Java 语言开发，有收费版本和社区免费版本，可以在网站 https://www.jetbrains.com/pycharm/download/ 上下载免费版本。

PyCharm 具有一般 IDE 具备的功能，如调试、语法加亮、Project 管理、代码跳转、智能提示、自动完成、单元测试、版本控制等。另外，PyCharm 还提供了一些很好的用于 Django 的开发功能，同时支持 Google APP Engine 和 IronPython。

下载 PyCharm 的安装包并安装，选择设置主题等操作后程序会自动启动并打开 PyCharm 程序。PyCharm 主界面如图 1-12 所示。选择“File”→“new”→“Python file”选项即可新建文件，并在其中编写代码。编写完成后选择“run”选项或按“Alt+Shift+F10”快捷键即可运行代码。

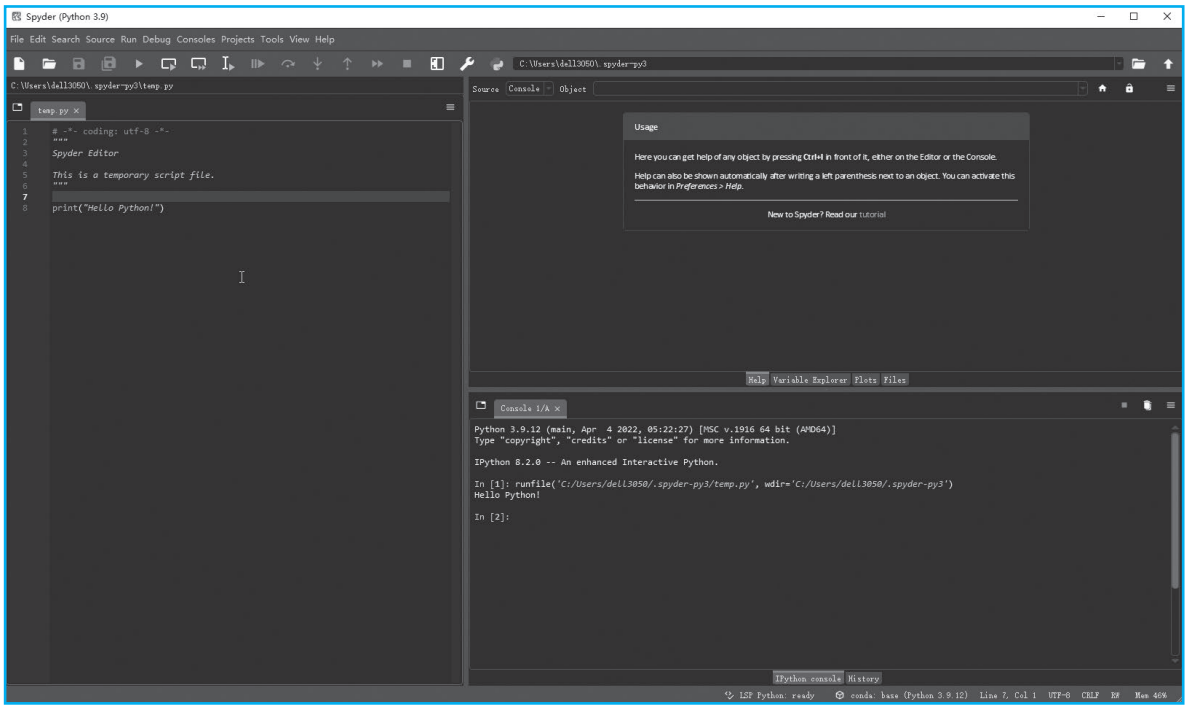

图 1–11　Anaconda 中的 Spyder 编辑器的操作界面

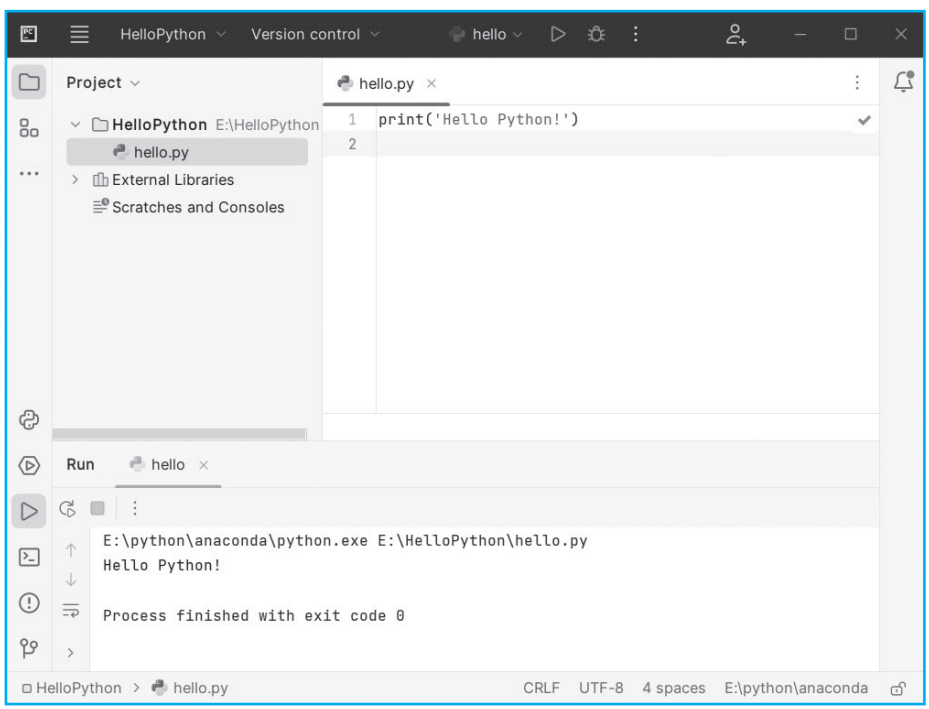

图 1–12　PyCharm 主界面

3. Eclipse

Eclipse 是用 Java 语言开发的一个集成开发环境，而且是一个开源项目。Eclipse 具有很好的扩展性，不但其原生程序可以作为 Java 的 IDE，而且有大量的插件来支持其他语言的开发。

在 Eclipse 平台上安装 pydev 插件就可以进行 Python 的开发工作了。

Eclipse 也非常强大，实现了 Python 代码的语法加亮、代码提示和代码补全等智能化功能。使用 Eclipse 加 pydev 插件搭建的 Python 开发环境如图 1-13 所示。

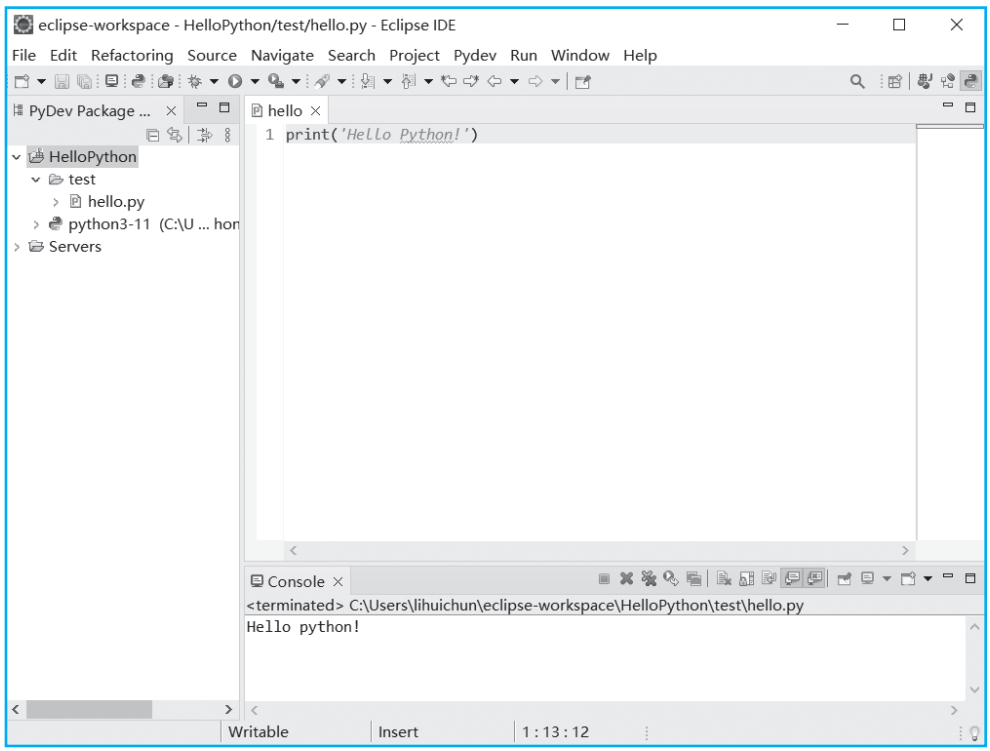

图 1-13 使用 Eclipse 加 pydev 插件搭建的 Python 开发环境

1.4 第一个 Python 程序

运行 Python 程序有两种方式：交互式和文件式。其中，交互式指 Python 解释器即时响应用户输入的每条代码，并给出输出结果。文件式也称批量式，指用户将 Python 程序写在一个或多个文件中，然后启动 Python 解释器批量执行文件中的代码。交互式一般用于调试少量代码的情况，文件式则是最常用的编程方式。

接下来的第一个 Python 程序分别用上述两种方式输出“我爱你中国 !”。

1. 交互式启动和运行方法

交互式有两种启动和运行方法。

1）在命令提示符窗口中启动 Python 解释器实现

在 Windows“开始”搜索框中输入“cmd”并按“Enter”键，在弹出的命令提示符窗口中执行“python”命令，进入 Python 解释器中。在命令提示符 >>> 后输入如下代码，并按“Enter”键：

```
print(" 我爱你中国 !")
```

运行结果如图 1-14 所示。

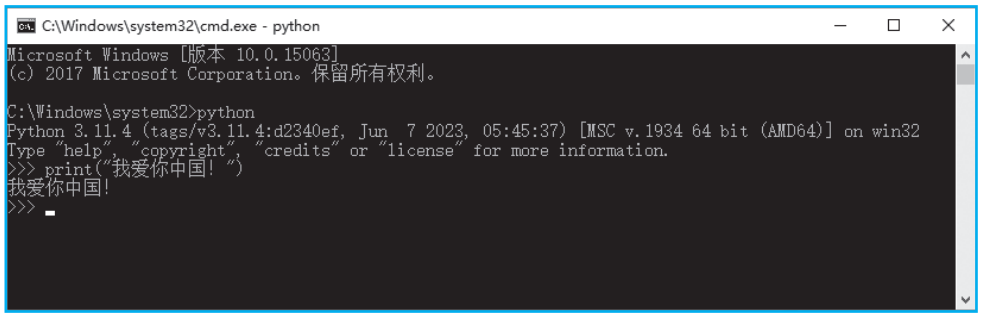

图 1-14　通过命令行启动交互式 Python 运行环境

2）在 Python 自带的 IDLE 中实现

单击 Windows 系统的“开始”按钮，然后选择“所有程序”→“Python 3.11”→“IDLE”选项，即可打开 IDLE 窗口。在命令提示符 >>> 后输入如下代码，并按“Enter”键：

```
print(" 我爱你中国 !")
```

运行结果如图 1-15 所示。

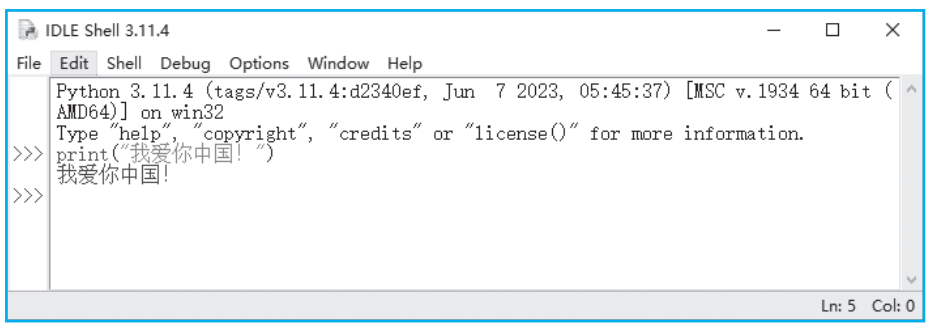

图 1-15　通过 IDLE 启动交互式 Python 运行环境

2. 文件式启动和运行方法

文件式启动和运行方法是更为实用的 Python 编程方式。

打开 Python 自带的 IDLE，选择“File”→“New File”选项，或按“Ctrl+N”快捷键打开一个新窗口。这个窗口不是交互模式，它是一个具备 Python 语法高亮辅助的编辑器，可以在其中进行代码编辑。输入代码“print(" 我爱你中国 !")”，并将程序保存为 first.py 文件，如图 1-16 所示。

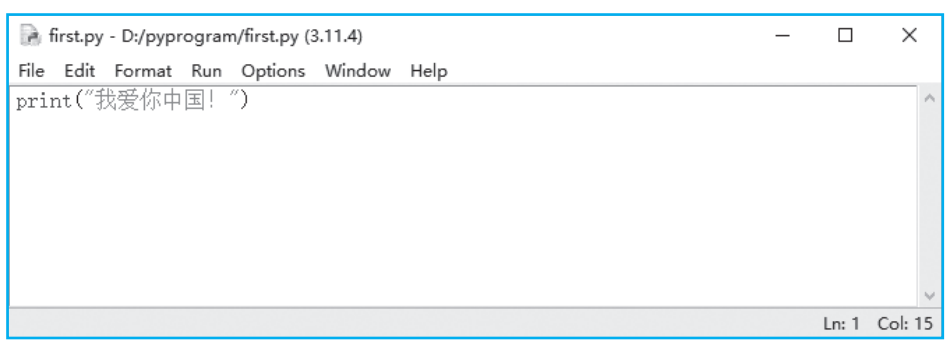

图 1-16　通过 IDLE 编写代码

在 IDLE 中按“F5”键，或选择“Run”→“Run Module”选项，执行后的结果显示在 IDLE Shell 中，如图 1-17 所示。

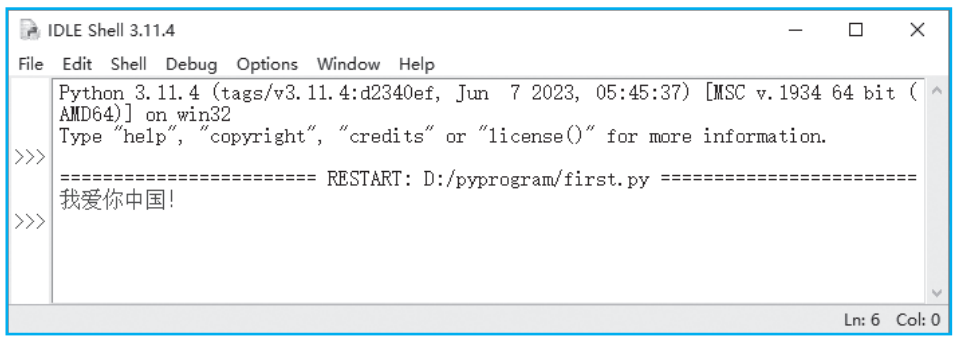

图 1-17 通过 IDLE 运行代码

Python 自带的 IDLE 是一个使用简单且十分有效的集成开发环境，交互式或文件式都有助于快速编写和调试代码，因此成为小规模 Python 软件项目的主要编写工具。本书所有程序均可通过 IDLE 进行编写和调试，在使用方面，对于单行的代码往往使用简单的交互式进行描述，而对于多行成段的代码，则采用 IDLE 的文件式。

本章小结

本章介绍了程序设计语言的基础知识，包括程序设计语言的发展、编译和解释、Python 语言的起源和特点、Python 解释器的安装、常用的 Python 集成开发环境、Python 程序的编辑及其调试运行。其中，重点是 Python 解释器的安装及其 IDLE 的使用，要求学习者能够在 IDLE 开发环境中熟练地编写及运行简单的 Python 程序，为今后的学习打下良好的基础。

习题 1

1. 选择题

（1）在计算机上可以直接运行的程序是（　　）。

A. 高级语言程序　　B. 汇编语言程序

C. 机器语言程序　　D. Python 语言程序

（2）采用解释翻译方式的高级程序设计语言有（　　）。

A. C 语言　　B. Python 语言

C. Java 语言　　D. Pascal 语言

（3）Python 语言不具有（　　）的特点。

A. 语法简单　　B. 跨平台

C. 不支持中文　　D. 开源性

（4）以下（　　）不属于 Python 语言的设计理念。

A. 优雅　　B. 结构复杂　　C. 简单　　D. 明确

（5）以下关于 Python 语言的叙述中正确的是（　　）。

A．Python 语言比其他语言高级

B．Python 语言编写的程序可以不用翻译就被计算机直接识别执行

C．运行 Python 程序有两种方式，即交互式和文件式

D．Python 语言出现得晚，所以具有其他语言的一切优点

2. 判断题

（1）高级程序设计语言的执行效率比汇编语言高。（　　）

（2）用高级语言编写的源程序，用解释或编译方式进行翻译，结果是相同的，都可以生成可执行文件。（　　）

（3）Python 既可以在 Shell 中运行，也可以存储成以 py 为扩展名的文本文件，然后用 Python 解释器执行。（　　）

3. 编程题

采用交互式和文件式两种方法，输出“绿水青山就是金山银山”。