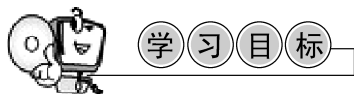


第1章



Python 简介



学习目标

1. 了解编程语言的分类,Python 语言的类属。
2. 了解 Python 语言的特点及优缺点。
3. 掌握 Python 程序的安装方法,了解工作目录的设置方法。
4. 掌握 IDLE 的使用方法。
5. 掌握 Python 语言的基本语法特点。
6. 掌握 Turtle 库的使用方法,能够使用 Turtle 绘制图形。

1.1 编程语言简介

1946 年世界上第一台电子计算机问世后,人类利用电子计算机进行科学计算、数据处理、工业控制、事务处理、娱乐、通信等,计算机把人类从繁重的计算、繁杂的事务、烦琐的管理中解脱出来。计算机的发展也经历了电子管计算机、晶体管计算机、集成电路计算机、大规模集成电路计算机等几个阶段。英特尔公司创始人戈登·摩尔(Gorden E. Moore)曾经提出过一个预言:单位面积集成电路上集成的晶体管的数量每两年翻一倍。这就是摩尔定律,它揭示了计算机技术在飞速地发展。正是集成电路技术、软件技术、网络技术等相关技术的发展,使电子计算机的体积变得越来越小,性能越来越强大,应用范围越来越广泛。在计算机技术飞速发展的过程中,编程语言也在不断地发展,下面将简要介绍计算机与编程语言的发展过程。

1.1.1 低级语言

1. 机器语言

从 1946 年第一台电子计算机诞生,到 1954 年 Fortran 语言发布,这一阶段使用的语言



教学视频

称为低级语言。这一时期计算机的运行速度和存储容量都比较低,操作系统的功能也比较弱,编程语言主要是机器语言和汇编语言。机器语言是计算机最原始的语言,由二进制计算机指令组成,是计算机硬件可以直接识别和执行的程序语言,如求 $2+3$ 的和,16 位计算机上的机器指令为 11010010 00111011,不同的计算机指令还会不同。

2. 汇编语言

由于机器语言不便于编程、阅读和修改,用助记符与机器指令一一对应,可以帮助程序员高效率地编写程序,于是汇编语言诞生了,如求 $2+3$ 的和,汇编语言的语句为 `add 2, 3, result`,也就是 $2+3$ 的结果放入 `result` 中。不同计算机的指令是不同的,因此汇编语句也是不同的。机器语言与汇编语言直接对计算机的硬件进行操作,因此称为低级语言。低级语言最大的优点是执行速度快,但代码编写难度较大,可读性较差。另外,低级语言编写的程序只能在一种计算机上运行,想要运行在不同类型的机器上,必须重写。低级语言是早期的一种计算机编程语言,现在只在很少的特殊场景中使用。

1.1.2 高级语言

从 1954 年第一个高级编程语言 Fortran 诞生开始,不断有新的高级编程语言产生,如 Pascal、COBOL、Basic、C、C++、Java、C# 等,尤其 C 和 C++ 语言已经成为程序员进行系统开发的通用语言。高级语言主要是相对于汇编语言而言的,高级语言把多条汇编指令合成一个表达式,对于高级语言来说需要一个编译器来完成高级语言到汇编语言的转换。所以对比不同的 CPU(中央处理器)结构,只需要有不同的编译器和汇编器,程序就可以在不同的 CPU 运行了。例如在 VS2010 编译器中,可以选择程序编译的目标平台 x86、x64、ARM 等。当然除了这些编译类的语言之外还有解释类型的语言,如 Python。高级语言基本脱离了机器的硬件系统,用人们更易理解的自然语言和数学公式的方式编程,编写的程序称为源程序。高级语言有更强的表达能力,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,而且容易学习掌握。例如求 $2+3$ 的和,高级语言可以表示为 `result=2+3`,代码不再与硬件相关,通用性更强,但高级语言编译生成的程序代码一般比用汇编语言设计的程序代码要长,执行的速度相对较慢。

1. 可视化编程语言

1984 年苹果公司发布了运行于 Macintosh 系列计算机上的操作系统 Mac OS,它是首个商用图形用户界面的操作系统,1985 年微软公司发行了运行于 IBM 兼容机上的 Microsoft Windows 1.0。以后两家公司不断地对操作系统进行升级换代,图形化用户界面的操作系统成为主流。为满足人们图形化程序开发的需求,出现了可视化编程语言,如 Delphi、Visual Basic、Visual C++、Visual Foxpro 等。可视化编程首先设计窗口,然后在窗口上放置文本框、单选按钮、复选框、下拉列表、菜单、按钮等控件,通过事件和方法来完成事务的处理。可视化编程增强了程序的可操作性和易用性。

2. 跨平台编程语言

高级语言要解决的最大问题是程序的通用性,也就是编完代码后,不依赖计算机硬件,可以在不同的计算机上运行。不同的计算机上运行着不同的操作系统,如 Unix、Windows、Mac OS、Linux 等,为了实现一次编码到处运行的目标,Java 语言应运而生,Java 语言首先

把源代码编译成为字节码,在不同的操作系统上运行的 Java 虚拟机解释执行字节码,达到了 Java 语言跨平台的目的。Python 作为高级解释型语言,它也具有跨平台性,在 Mac OS 和 Linux 操作系统上已经集成了 Python 解释器,Windows 操作系统也可以安装 Python 解释器运行 Python 程序。

3. 网络编程语言

20 世纪 90 年代中期,互联网开始流行起来,与互联网相关的编程语言开始出现。例如,描述网页的 HTML(Hyper Text Markup Language,超文本标记语言),设计动态网页的 ASP、PHP、JavaScript 等,网络编程语言使网页更生动、更绚烂多彩。

4. 新生代语言

进入 21 世纪,新的技术不断涌现,在计算机领域,尤其云计算、大数据和人工智能技术极大地促进了人类社会的发展。云计算将计算能力、存储能力、软件应用等视作自来水一样可以按需使用、按量付费的资源。大数据是指需要处理的数据量超过传统处理方式所能处理的量级,并且数据类型也是多样的(二维表式的结构化数据、文本的无结构数据及介于两者之间的半结构化数据)。大数据技术将海量数据划分为 N 个数据块,由集群中的计算机分别完成处理任务,最后进行综合。适合于大数据处理的编程语言有 R 语言、Scala、Python。R 语言适用于统计分析、图形的绘制。Scala 是一门多范式的编程语言——可以进行面向对象编程和函数式编程,能够实现可伸缩编程以适应不同规模的计算机集群,Scala 与大数据分析平台 Spark 是一对绝佳搭档。Spark 平台除了支持 Scala 以外,也支持 Python 语言,再加上大量第三方库的支持,Python 在大数据处理方面也是一门非常流行的编程语言。

随着 AlphaGo(围棋机器人)打败围棋世界冠军、职业九段棋手李世石以及世界排名第一的柯洁,人工智能技术迎来了又一次繁荣。人工智能(Artificial Intelligence,缩写为 AI)是研究、开发用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的一门新的技术科学。人工智能典型的实际应用有机器视觉、指纹/掌纹识别、人脸识别、视网膜识别、虹膜识别、专家系统、自动规划、智能搜索、定理证明、博弈、自动程序设计、智能控制、机器人学、语言和图像理解、遗传编程等。人工智能的首选语言是 Python,Python 拥有强大的 AI 技术库 Numpy 和 Scikit-Learn,深度学习框架 TensorFlow,Caffe 的主体就是用 Python 实现的,并提供 Python 原生接口。

1.1.3 编译型语言与解释型语言

高级语言的源代码是用字母、数字、符号组成的,源代码便于人类编写、修改,但不能直接在计算机上运行,需要用编译器(Compiler)将源代码翻译成机器能够识别、运行的机器代码,这个翻译过程称为编译,如图 1-1 所示。需要用编译器将源代码编译为机器代码的语言为编译型语言。典型的编译型语言有 C/C++、Pascal/Object Pascal(Delphi)等。编译型语言需要一个专门的编译过程,把源程序编译成为机器语言的文件,如 EXE 文件,以后若要运行就不必重新翻译,直接使用编译的结果就可以了(EXE 文件),因为翻译只做了一次,运行时不需要翻译,所以编译型语言编写的程序执行效率较高。

解释性语言在运行程序的时候才将源代码翻译成机器代码,例如,解释型 Basic 语言,专门有一个解释器能够直接执行 Basic 程序,每个语句都是执行的时候才翻译,如图 1-2 所

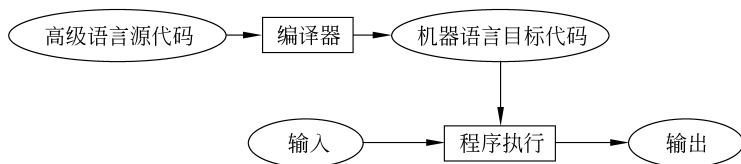


图 1-1 程序的编译与执行

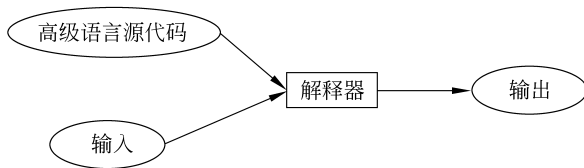


图 1-2 程序的解释与执行

示。解释型语言每执行一次就需要翻译一次,效率比较低,但程序纠错和维护十分方便,在不同的操作系统上都能运行,因此可移植性较好。典型的解释型语言有 JavaScript、VBScript、Perl、Python、Ruby、MATLAB 等。

1.1.4 专用编程语言与通用编程语言

专门用于某一特定领域的编程语言称为专用编程语言。例如,专用于动态网页制作的语言 PHP、ASP、JavaScript 等;用于控制机床的数控语言(NC Language);数据库结构化查询语言(SQL)。专用编程语言通常只能使用于特定领域,在该领域外,此语言功能则较弱,应用领域受到限制。

通用编程语言是指能够用于多个领域、具有多种用途的编程语言。例如,C 语言既可以用于编写操作系统,也可以用于编写硬件的驱动程序,还可以用于嵌入式开发和其他程序的开发;Python 语言安装 Pygame 库后可以用于游戏的开发,安装 PIL 库后可以用于图形图像处理,安装 Flask/Django 等框架后可以用于 Web 程序开发,安装 PyQt 库后可以用于图形化程序开发,因应用领域较广,因此,Python 也是通用编程语言。



教学视频

1.2 Python 语言简介

Python 是一门优雅的、面向对象的、解释型的计算机高级程序设计语言,它由荷兰的吉多·范罗萨姆(Guido van Rossum)在 1989 年年底设计开发。Python 是一种体现简单主义思想的语言,阅读一段良好的语言就像在阅读英语一样,不用专注于语言的学习与功能的实现,而是专注于解决问题,因此 Python 非常容易上手,通过其简单的文档,用户可以非常快速地掌握这门语言。另外,Python 拥有丰富而强大的第三方库,可以把别人发布的模块贴到程序中,为自己所用,因此,Python 也被称为“胶水语言”,这是许多人喜爱它的原因。

1989 年,吉多·范罗萨姆还是荷兰的 CWI(Centrum voor Wiskunde en Informatica, 国家数学和计算机科学研究院)的研究人员,正在进行一个研究项目,他们用现有工具努力工作,他们想开发出一种新的工具,以便能够简单有效地进行研究工作。吉多·范罗萨姆拥

有 ABC 编程语言丰富的经验,但 ABC 语言开发能力有限,于是他就产生了开发一种通用的功能强大的解释型语言的想法。

1989 年圣诞节期间,吉多·范罗萨姆为了打发圣诞节的无趣,决心开发一个新的脚本解释程序,作为 ABC 语言的一种继承。ABC 语言是吉多·范罗萨姆参与设计的一种教学语言,在他看来 ABC 语言虽然优美而强大,但它并没有广泛流行的原因在于它不具有开放性,他要开发一种优雅而强大的解释型语言,并且借鉴其他语言的优点。于是 Python 语言就诞生了,1991 年发布了第一个公开发行版。之所以选中 Python(大蟒蛇的意思)作为编程语言的名字,是因为他是一个叫 Monty Python 的喜剧团体的爱好者。

1. Python 的优点

(1) 简单易学。这是 Python 受欢迎的重要原因。在设计之初,吉多·范罗萨姆就想要把它设计成为非专业人员使用的一种极易上手的解释型语言。Python 语言中没有其他语言中常见的美元符号(\$)、分号(;)、波浪号(~)等,这些符号使语言晦涩难懂。并且 Python 有极其简单的说明文档,这也是学习和使用 Python 语言的基础。

(2) 速度快。Python 的底层是用 C 语言写的,很多标准库和第三方库也都是用 C 语言写的,运行速度非常快。

(3) 开源、免费。Python 由一个非营利组织——Python 软件基金会负责运维,实行开源、免费、共享。使用者可以自由地发布这个软件的拷贝、阅读它的源代码,对它做改动,把它的一部分用于新的自由软件中。Python 是 FLOSS(自由/开放源码软件)之一,FLOSS 是基于一个团体分享知识的概念。

(4) 高级语言。Python 是一门高级语言,程序员编写程序时无须考虑内存回收等底层细节,同时它拥有其他语言没有的一些数据结构,如 Python 内建了列表(可变的数组)和字典(哈希表),这是 C、C++ 和 Java 等语言不可比的。

(5) 跨平台性。由于 Python 是开源的,它已经被移植到许多平台上,包括 Windows、Linux、Macintosh、FreeBSD、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、Pocket PC、Symbian 以及 Google 基于 Linux 开发的 Android 平台。

(6) 解释性。一个用编译语言比如 C 或 C++ 写的程序可以从源文件(即 C 或 C++ 语言)转换到计算机使用的语言(二进制代码,即 0 和 1)。这个过程通过编译器和不同的标记、选项完成。

运行程序的时候,连接器/转载器软件把程序从硬盘复制到内存中并且运行。而 Python 语言写的程序不需要编译成二进制代码,可以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为中间形式——字节码,然后把它翻译成计算机使用的机器语言并运行。这使 Python 的使用更加简单,也使 Python 程序更加易于移植。

(7) 面向对象。Python 既支持面向过程的编程,也支持面向对象的编程。在“面向过程”的语言中,程序是由过程或仅仅是可重用代码的函数构建起来的。在“面向对象”的语言中,程序是由数据和功能组合而成的对象构建起来的。Python 不仅是一门面向对象的语言,它还融合了多种编程风格,如借鉴了 Lisp 等函数编程的特点。

(8) 可扩展性。如果需要一段关键代码运行得更快或者希望某些算法不公开,这部分

程序可以用 C 语言或 C++ 语言编写,然后在 Python 程序中使用它们。Python 语言具有丰富和强大的第三方库,能够把用其他语言制作的各种模块(尤其是 C/C++)很轻松地联结在一起,扩展了 Python 的功能。

(9) 可嵌入性。Python 可以嵌入 C 和 C++ 的项目中,使程序具有脚本语言的特点,向程序用户提供脚本功能。

(10) 丰富的库。Python 标准库非常庞大。它可以帮助处理各种工作,包括正则表达式、文档生成、单元测试、线程、数据库、网页浏览器、CGI、FTP、电子邮件、XML、XML-RPC、HTML、WAV 文件、密码系统、GUI(图形用户界面)、Tk 和其他与系统有关的操作。这被称为 Python 的“功能齐全”理念。除了标准库以外,还有许多其他高质量的库,如 wxPython、Twisted 和 Python 图像库等。

(11) 规范的代码。Python 采用强制缩进的方式使得代码具有较好的可读性。而 Python 语言写的程序不需要编译成二进制代码。

2. Python 的缺点

(1) 单行语句和命令行输出问题。很多时候不能将程序连写成一,如 `import sys;for i in sys.path:print i`。而 Perl 和 AWK 就无此限制,可以较为方便地在 Shell 下完成简单程序,不需要如 Python 一样,必须将程序写入一个 .py 文件。

(2) 独特的语法。这也许不应该被称为缺点,但是它用缩进来区分语句关系的方式还是给很多初学者带来了困惑。即便是很有经验的 Python 程序员,也可能陷入陷阱中。最常见的情况是 Tab 和空格的混用会导致错误,而这是用肉眼无法区分的。

(3) 运行速度慢。因为 Python 是解释型脚本语言,相比较而言,它显得较慢,但随着硬件性能的提升,这个问题将不再是问题。

Python 已经成为最受欢迎的程序设计语言之一。截至 2022 年 1 月,Python 被 TIOBE 编程语言排行榜(<https://www.tiobe.com/tiobe-index/>)5 次评为年度最佳编程语言,也是获奖次数最多的编程语言;2021 年 10 月,Python 终于在 TIOBE 编程语言排行榜上超过 C 和 Java 语言,荣升第一。截至 2022 年 6 月,在根据 Google 搜索做出的 PYPL 排名中(<http://pypl.github.io/PYPL.html>)Python 连续 5 年保持在第一的位置,排名如图 1-3 所示。

Worldwide, Sept 2022 compared to a year ago:				
Rank	Change	Language	Share	Trend
1		Python	28.11 %	-2.6 %
2		Java	17.35 %	-0.9 %
3		JavaScript	9.48 %	+0.2 %
4		C#	7.08 %	+0.1 %
5		C/C++	6.19 %	-0.3 %
6		PHP	5.47 %	-0.8 %
7		R	4.35 %	+0.6 %
8	↑↑	TypeScript	2.79 %	+1.1 %
9	↑↑	Swift	2.09 %	+0.5 %
10	↓↓	Objective-C	2.03 %	+0.2 %

图 1-3 2022 年 9 月的 TIOBE 编程语言排行榜

自从 20 世纪 90 年代初 Python 语言诞生至今,它逐渐被广泛应用于多个领域。由于它简洁、易读以及可扩展性,一些知名的大学已经采用 Python 作为程序设计课程的入门课程。如卡耐基·梅隆大学和麻省理工学院等。众多开源的软件包都提供了 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。Python 专用的科学计算扩展库也越来越多,如 NumPy、SciPy 和 Matplotlib,它们分别为 Python 提供了快速数组处理、数值计算以及绘图功能。因此 Python 非常适合于工程技术人员、科研人员等进行数据处理、图表制作、科学计算等。

1.3 Python 的安装



教学视频

因为 Python 具有跨平台性,所以在不同的平台上需要安装不同的版本。Python 安装程序的下载地址为 <https://www.python.org/>,可依据自己所使用的平台,选择合适的版本。Python 目前最新发行版本有 2.7.x 和 3.x.x 两个版本,Python 3.x.x 不向下兼容 Python 2.x.x,而绝大多数原有的组件和扩展都是兼容 Python 2.x.x 的,因此,如果需要与原有第三方模块兼容,则选择 Python 2.x.x 比较合适。但 Python 3.x.x 是未来的发展趋势,将来大部分程序将运行在 Python 3.x.x 上,因此这里选 Python 3.x.x 作为学习研究的平台。

在 Windows 平台下安装时,最好将 Python.exe 添加到 Path 环境变量中,也就是在安装时选中 Add Python.exe to Path 选项。

单击 Install Now 按钮就可以开始安装了,依次单击 Next 按钮就可以完成安装,但这种安装方式有点问题,就是安装路径较深,以后保存文件时较难查找。建议还是选择自定义安装,如图 1-4 所示,选中 Add Python 3.7 to PATH 后,单击 Customize installation 按钮开始自定义安装。

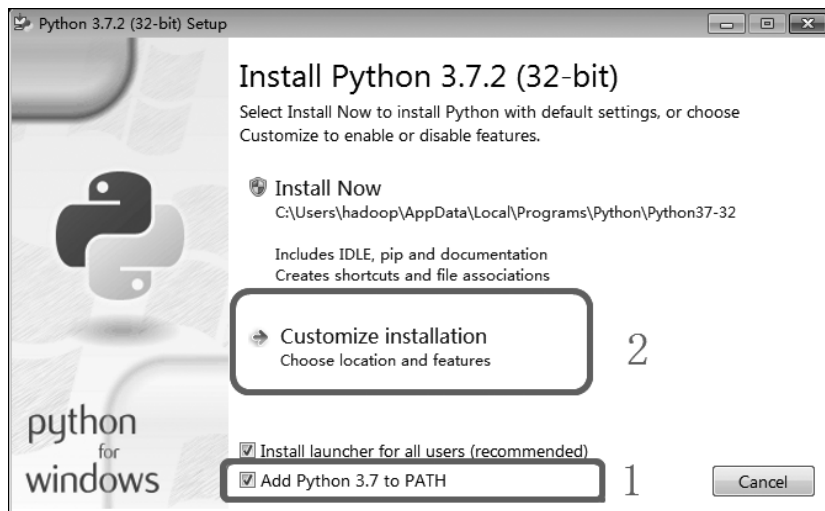


图 1-4 将 Python 添加到 Path 环境变量中

然后单击 Next 按钮进入如图 1-5 所示界面后,单击 Install for all users,使本机的所有用户都能使用 Python。然后在下面的文本框中输入安装路径,如输入的路径为 C:\

Python37, 这样就把 Python 安装到了 C:\Python37 路径下, 也可以单击 Browse 选择一个已经存在的文件夹安装。

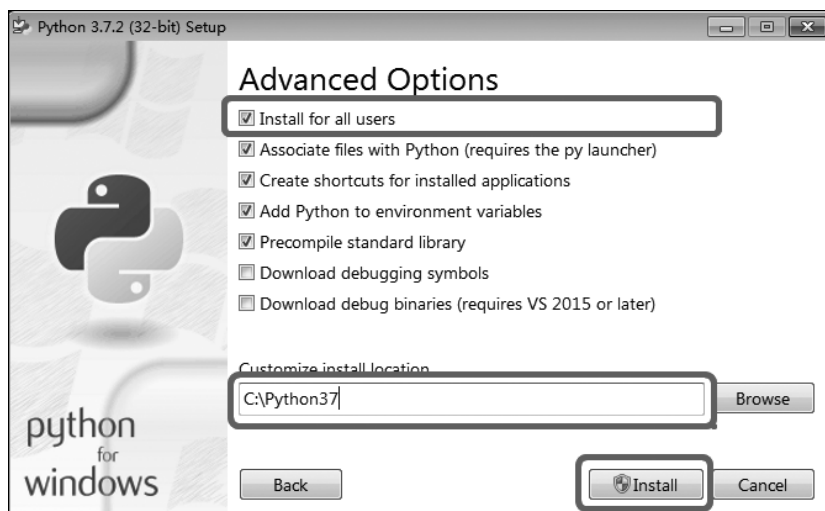
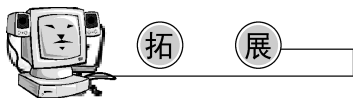


图 1-5 设置安装路径

最后单击 Close 按钮, 完成安装。以后编辑的程序文件.py 默认就保存在该安装路径下。



IDLE 下自定义程序保存位置。自定义安装后, 程序默认保存在自定义安装文件夹中。也可以创建 IDLE 桌面快捷方式(选择“开始”→Python3.x→右击 IDLE→“发送到”→“桌面快捷方式”命令), 右击快捷方式, 选择“属性”命令, 在弹出窗口的“起始位置”框中输入准备保存程序文件的文件夹路径。那么 IDLE 中编辑的程序就会保存在“起始位置”指定的文件夹中。



教学视频

1.4 快速入门

Windows 操作系统下安装标准版的 Python 后, 集成有 Python 解释器, 选择“开始”→“所有程序”→Python 3.x→IDLE(Python 3.x GUI)命令打开 Python 自带的 IDLE 解释器, 如图 1-6 所示。

Python 解释器的提示符为 `>>>`, 在 `>>>` 后输入语句即可解释执行。

比如输入 `3+5`, 按 Enter 键, 即可得到计算结果为 8。

1. 简单的计算器

可以把解释器当作计算器使用, 常用的运算符有 `+`、`-`、`*`、`/`、`//`、`%`、`**` 等。下面来计算几个表达式。# 后为注释, 输入表达式时可以不输入 # 及其后面部分。

```
>>> 50-5*6
```

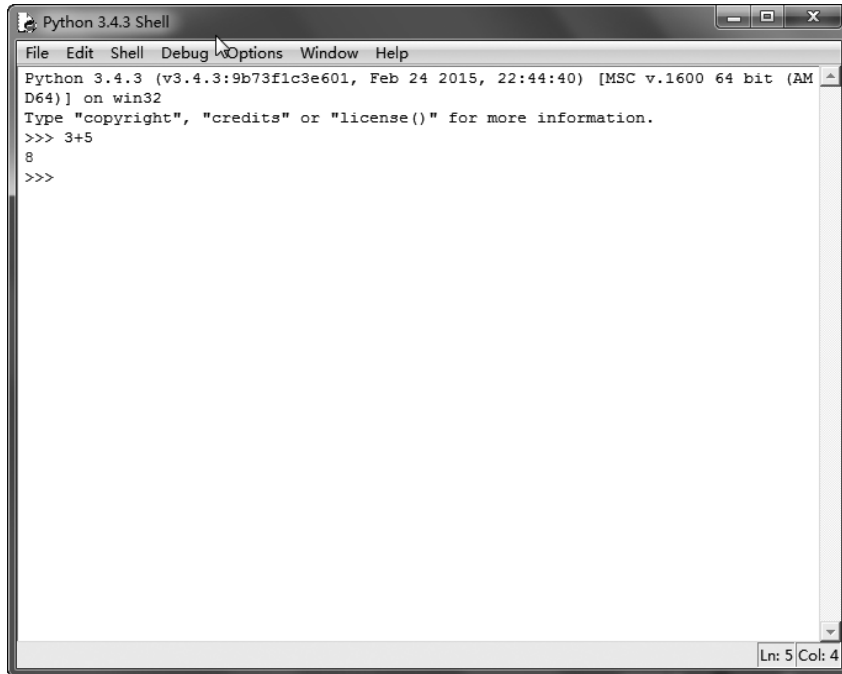


图 1-6 IDLE 解释器

```
20
>>> (50-5*6)/4
5.0
>>> 8/5      # "/"操作符得到的结果将是一个浮点数
1.6
>>> 17//3    # "//"操作符得到除法的整数部分,即整除
5
>>> 17%3     # "%"操作符得到除法的剩余部分,即模运算/取余运算
2
>>> 2**3     # 幂乘方运算
8
```

Python 中的操作符当然有优先级,详情请参见第 2 章,如果不想引起操作符优先级的错误,可以使用()把优先计算的表达式包围起来。

2. 编辑程序

编辑并运行程序,可以选择自己心仪的 IDE 环境,附录中已介绍过几个较常用的 IDE 环境,这里介绍 Python 自带的 IDLE 的简单使用方法。

在 IDLE 中选择菜单 File→New File 命令,打开程序编辑窗口,在这里就可以编辑程序。通常程序的第一句是:

```
# - * - coding:utf-8 - * -
```

或

```
# coding:utf-8
```

这一语句,表示程序保存时将使用 utf-8 编码保存,因为 Python 是能跨平台处理世界范围内字符的语言,所以最好指定程序保存所用的编码方式。

像许多教程一样,先做一个简单示例程序,如图 1-7 所示。

保存程序:选择菜单 File→Save 命令即可保存程序文件。

3. 程序的运行

运行程序有以下两种方式。

(1) 在 IDE 环境中运行程序。在 VS Code 和 PyCharm 中运行程序的方法参见附录 B。在 IDLE 中,选择菜单 Run→Run Module F5 命令即可运行,或者按快捷键 F5 也可以运行程序,如图 1-8 所示。

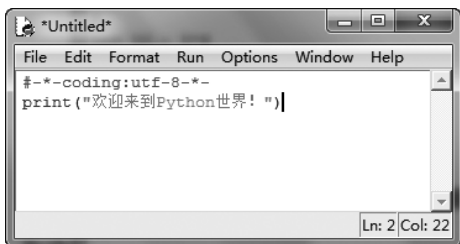


图 1-7 简单示例

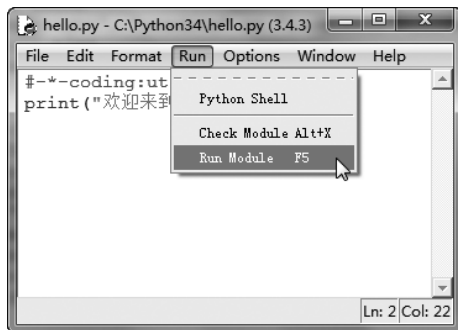


图 1-8 IDLE 中运行程序

(2) 命令行下运行程序。Windows 7 操作系统下,打开保存有准备运行的程序的文件夹,按住 Shift 键,在窗口空白处右击,在弹出的快捷菜单中选择“在此处打开命令窗口”命令,即可打开命令行窗口并且工作目录就是程序所在的目录。输入“python 程序名.py”即可执行程序,如图 1-9 所示。

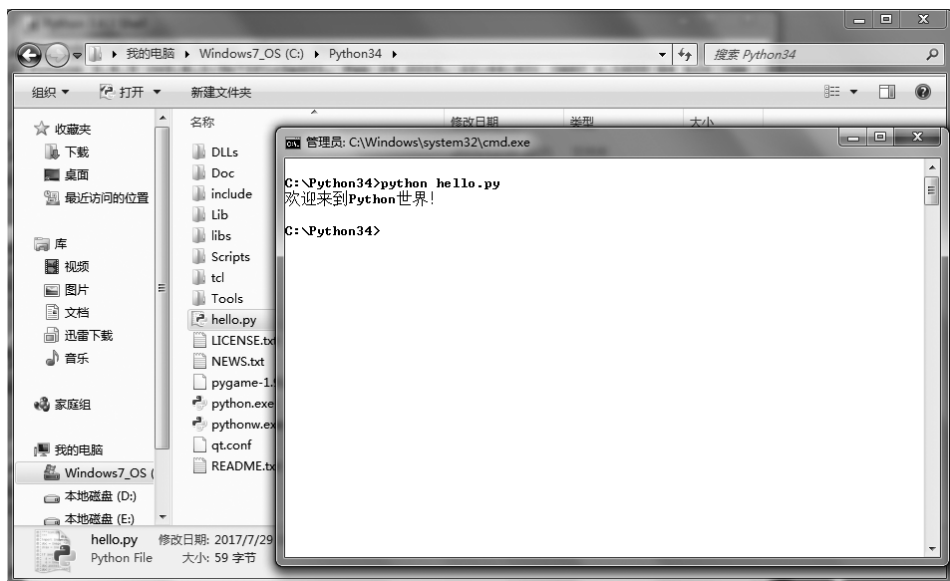


图 1-9 命令行下运行程序