

第 5 章 Python 语言入门

问题导入

Python 语言的诞生

Python 语言是当今最火的语言,那么你了解它台前幕后的小故事吗?

说到 Python 语言,它的诞生是极具戏剧性的,重度肥皂剧爱好者吉多(Guido van Rossum)为了打发圣诞节的无聊,开发了一种新的脚本解释程序,这就是传说中的 Python 语言。之所以会选择 Python(蟒蛇)作为该编程语言的名字,是因为吉多是一个叫 *Python's Flying Circus* 戏剧团体的忠实粉丝。

5.1 Python 语言概述

每台计算机都有自己的指令集合,每条指令可以让计算机完成一个最基本的操作。程序(program)则是由一系列指令根据特定规则组合而成,在计算机上执行程序的过程实质上就是组成程序的各项指令按顺序依次执行的过程。

对于程序来说,其功能通常可以抽象为如图 5-1 所示的形式,包括输入、输出和数据处理。

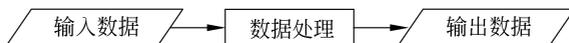


图 5-1 程序功能

输入:从键盘、文件或者其他设备获取待处理数据。

输出:把处理后的结果数据输出到屏幕、文件或者其他设备。

数据处理:对输入数据进行各种运算,得到输出结果。

5.1.1 Python 语言发展史

Python 语言于 20 世纪 90 年代初由荷兰数学和计算机研究所(Centrum Wiskunde & Informatica,CWI)的吉多基于 C 语言开发。在 Python 语言的开发过程中,虽然也有其他



课程思政:
持之以恒
的力量

开发者做了许多贡献,但吉多被认为是 Python 语言的主要作者。之所以选择 Python 作为该编程语言的名字,是因为吉多是室内情景幽默剧 *Monty Python's Flying Circus* 的忠实观众。

1995 年,吉多在弗吉尼亚州雷斯顿的国家研究计划公司(Corporation for National Research Initiatives,CNRI)继续他的 Python 语言开发工作,并发布了 Python 语言的多个版本。

2000 年 5 月,吉多和 Python 语言核心开发团队转移到 BeOpen.com,组建了 BeOpen PythonLabs 团队。同年 10 月,PythonLabs 团队转移到 Digital Creations 公司(现为 Zope 公司)。2001 年,Python 软件基金会(PSF,请参阅 <https://www.python.org/psf/>)成立,这是一个专门为拥有与 Python 语言相关的知识产权而创建的非营利组织。

2020 年 11 月,吉多加入微软开发者部落,如今的微软全面拥抱开源,吉多表示加入微软后将会继续开发优化 Python 语言,让它变得更加好用。

所有 Python 语言版本都是开源的,目前使用的 Python 语言版本主要有 Python 2.x 和 Python 3.x 两种。Python 3.x 并不完全兼容 Python 2.x 的语法,因此,在 Python 2.x 环境中编写的程序不一定能在 Python 3.x 环境中正常运行。

5.1.2 特点

(1) 简单易学: Python 是一种代表简单主义思想的语言,可以使用尽量少的代码完成更多工作。Python 语言使开发者能够专注于解决问题而不是去弄明白语言本身。另外,Python 语言有极其简单的说明文档,使得初学者很容易上手。

(2) 免费开源: FLOSS(free/libre and open source software)的中文含义是自由、开源软件,其已被证实为当今最好的开放、合作、国际化产品和开发样例之一,已经为全世界各大机构,包括政府、政策、商业、学术研究和开源领域带来巨大的利益。

(3) 跨平台性: 由于 Python 语言的开源本质,它已经被移植到许多平台上,在 Linux、Windows、Macintosh、Android 等操作系统平台上都可以运行 Python 语言编写的程序。

(4) 高层语言: 与 C/C++ 语言不同,使用 Python 语言编写程序时无须考虑诸如“管理程序使用的内存”一类的底层细节,从而使得开发者可以在忽略底层细节的情况下、专注于如何使用 Python 语言解决问题。

(5) 面向对象: Python 语言既支持面向过程的编程,也支持面向对象的编程。

(6) 丰富的库: Python 语言官方本身提供了非常完善的标准代码库,它可以帮助处理各种工作,包括网络编程、输入输出、文件系统、图形处理、数据库、文本处理等。除了内置库,开源社区和独立开发者长期为 Python 语言贡献了丰富大量的第三方库,如用于科学计算的 NumPy、用于 Web 开发的 Django、用于网页爬虫的 Scrapy 和用于图像处理的 OpenCV 等,其数量远超其他主流编程语言。

(7) 胶水语言: Python 语言本身被设计成具有可扩展性,它提供了丰富的 API 和工具,以便开发者能够轻松使用包括 C、C++ 等主流编程语言编写的模块来扩充程序。例如,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,可以部分程序用 C 或 C++ 语

言编写,然后在 Python 程序中使用它们。Python 语言就像使用胶水一样把用其他编程语言编写的模块黏合过来,让整个程序同时兼备其他语言的优点,起到了黏合剂的作用。正是这种“胶水”的角色让 Python 语言近几年在开发者团体中声名鹊起,因为互联网与移动互联网时代的需求量急速倍增,大量开发者亟须一种极速、敏捷的工具来助其处理与日俱增的工作,Python 语言发展至今的形态正好满足了这种需求。

5.1.3 环境安装

在 Linux、Windows、Macintosh、Android 等操作系统平台上,都可以安装 Python 语言环境以支持 Python 语言程序的运行,这里仅介绍 Windows 操作系统平台上的 Python 语言环境安装方法。本书所使用的 Python 语言版本为 2023 年 8 月的 3.11.5,读者可从 Python 语言官网(<https://www.python.org>)的 Downloads 页面下载各平台的安装包,如图 5-2 所示。

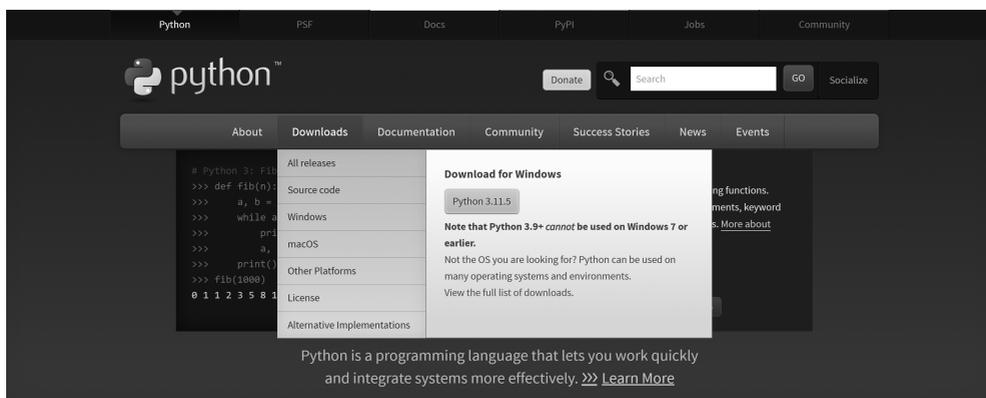


图 5-2 Python 语言官网

在 Python 语言官网中,选择 Downloads→All releases 选项,可以看到所有已发布的版本,如图 5-3 所示。

Looking for a specific release?
Python releases by version number:

Release version	Release date		Click for more
Python 3.11.5	Aug. 24, 2023	 Download	Release Notes
Python 3.10.13	Aug. 24, 2023	 Download	Release Notes
Python 3.9.18	Aug. 24, 2023	 Download	Release Notes
Python 3.8.18	Aug. 24, 2023	 Download	Release Notes
Python 3.10.12	June 6, 2023	 Download	Release Notes
Python 3.11.4	June 6, 2023	 Download	Release Notes
Python 3.7.17	June 6, 2023	 Download	Release Notes

[View older releases](#)

图 5-3 Python 语言发布版本列表

单击 Python 3.11.5 链接, 可以看到该版本下的可下载文件列表, 如图 5-4 所示。对于 Windows 操作系统用户, 可以下载 Windows installer(64-bit) 文件或 Windows installer(32-bit) 文件。

Version	Operating System	Description	MD5 Sum	File Size	GPG	Sigstore
Gzipped source tarball	Source release		b628f21aae5e2c3006a12380905bb640	26571003	SIG	.sigstore
XZ compressed source tarball	Source release		393856f1b7713aa8bba4b642ab9985d3	20053580	SIG	.sigstore
macOS 64-bit universal2 installer	macOS	for macOS 10.9 and later	7a24f8b4eeca34899b7d75caec3bc73	44239554	SIG	.sigstore
Windows embeddable package (32-bit)	Windows		add17856887d34c04a9cfd6c051c4bea	10053367	SIG	.sigstore
Windows embeddable package (64-bit)	Windows		c5e83dc45630df2236720a18170bf941	11170359	SIG	.sigstore
Windows embeddable package (ARM64)	Windows		8fc7d74daf27882f2a32a1b10c3a3a2c	10428395	SIG	.sigstore
Windows installer (32-bit)	Windows		ac8e48a759a6222ce9332691568fe67a	24662424	SIG	.sigstore
Windows installer (64-bit)	Windows	Recommended	3afd5b0ba1549f5b9a90c1e3aa8f041e	25932664	SIG	.sigstore
Windows installer (ARM64)	Windows	Experimental	cd2bfd6bb39a6c84dbf9d1615b9f5b5	25197192	SIG	.sigstore

图 5-4 可下载文件列表

5.1.4 Windows 操作系统平台上安装 Python 语言环境

下载 Windows 操作系统版本的 Python 3.11.5 语言环境安装包后, 即可开始安装, 安装步骤如下。

步骤 1: 双击安装包, 即可出现如图 5-5(a) 所示的安装向导界面, 勾选 Add Python.exe to PATH 的复选框。

步骤 2: 选择如图 5-5(a) 中的 Install Now 选项, 出现如图 5-5 (b) 所示界面。

步骤 3: 在如图 5-5(b) 所示界面中不需做任何操作, 等待系统安装, 出现如图 5-5(c) 所示界面, 表示安装成功, 单击 Close 按钮结束安装。

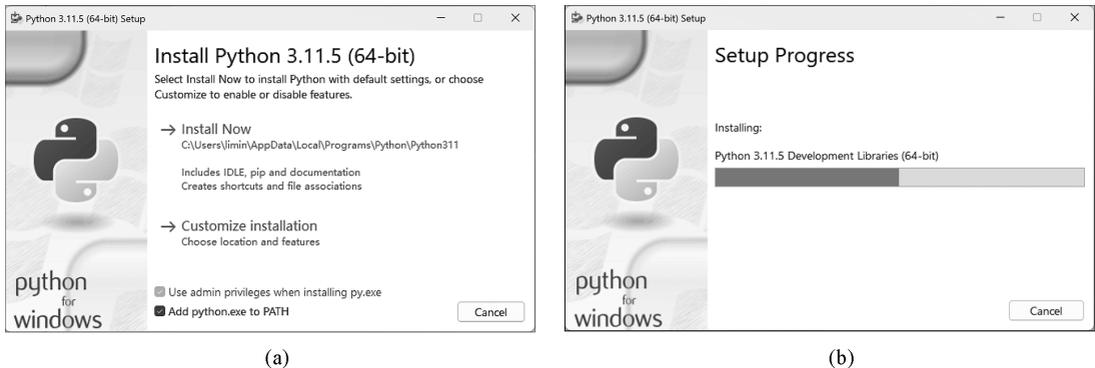
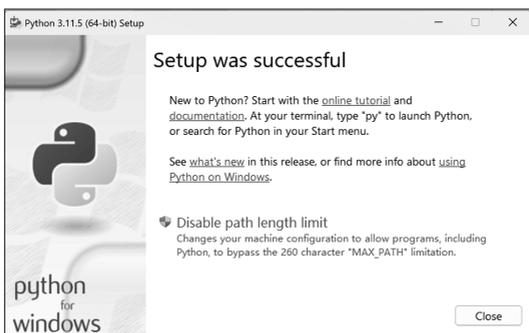


图 5-5 Windows 操作系统和平台 Python 语言环境安装步骤



(c)

图 5-5 (续)

 提示

如果在如图 5-5(a)所示界面中未勾选 Add Python.exe to PATH 的复选框,则在命令提示符中执行 Python、pip 等程序时需要指定程序所在路径,否则系统会找不到可运行的程序。另外一种方法是通过编辑环境变量将 Python 相关程序所在路径添加到 Path 中,如图 5-6 所示,读者应根据自己实际安装路径作相应修改。

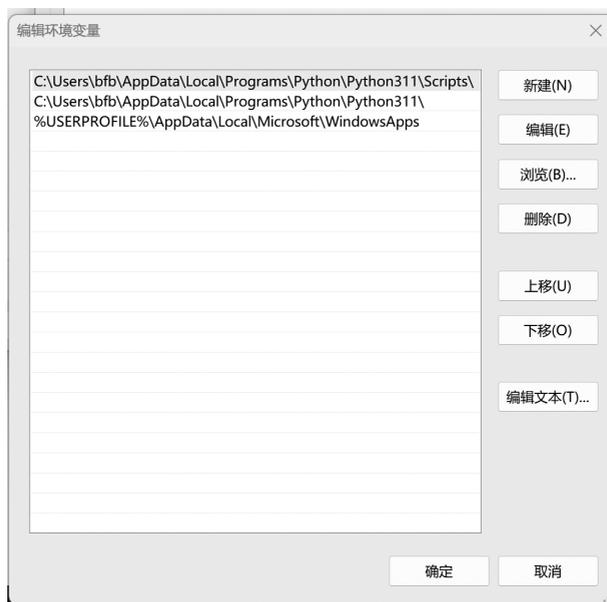


图 5-6 Python 环境路径设置

安装 Python 3.11.5 语言环境后,在 Windows 操作系统命令提示符中输入 python,即可进入 Python 解释器控制台,Python 提示符为“>>>”,如图 5-7 所示。

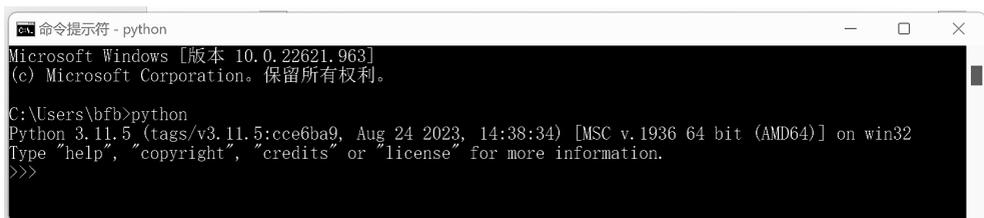


图 5-7 Python 语言环境控制台

5.2 HelloWorld 程序

Python 程序支持两种运行方式：交互式和脚本式。下面以代码清单 5-1 中所示的 HelloWorld 程序为例介绍这两种运行方式。

代码清单 5-1 HelloWorld 程序。

```
1 print("Hello World!")           #在屏幕上输出 Hello World!
```

对于交互式运行方式，可以在操作系统的命令提示符中输入 `python` 启动 Python 解释器，然后在 Python 提示符“`>>>`”后面依次输入每行代码后按回车键，即可看到如图 5-8 所示的结果。

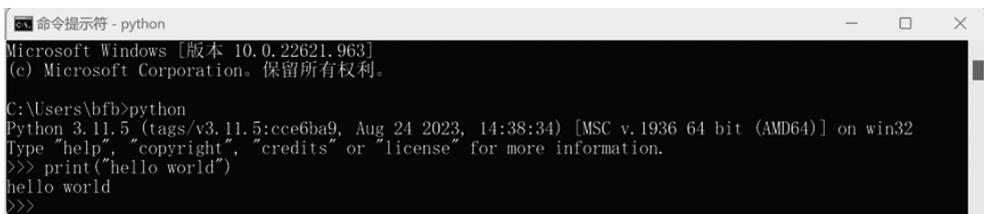


图 5-8 交互式运行结果

对于脚本式运行方式，请参看 5.3 节 IDLE 环境介绍。

5.2.1 中文编码

在 Python 3.x 的语言环境中，默认使用 UTF-8 编码，因此，可以直接支持中文。比如，将代码清单 5-1 中的代码修改如下：

代码清单 5-2 带中文的 HelloWorld 程序。

```
1 print("你好,世界!")           #在屏幕上输出“你好,世界!”
```

代码清单 5-2 在 Python 3.x 环境中可以正常运行并在屏幕上输出“你好,世界!”。

 注意

使用 Python 3.x 环境创建 Python 脚本文件时,需要将文件编码格式设置为 UTF-8,否则运行脚本时可能会报错。例如,如果在使用 ANSI 编码的 Python 脚本文件中输入代码清单 5-2 并运行,则会出现如下错误信息提示。

```
SyntaxError: Non-UTF-8 code starting with '\xcd' in file d:/pythonsamplecode/01/helloworld.py on line 4, but no encoding declared; see http://python.org/dev/peps/pep-0263/ for detailsScriptObject
```

5.2.2 单行注释

注释是为了增强代码可读性而添加的描述文字。在代码被编译或解释时,编译器或解释器会自动过滤掉注释文字。也就是说,注释的主要作用是供开发者查看、使得开发者能够更容易理解代码的作用和含义,在代码运行时注释文字并不会被执行。

Python 语言提供了单行注释和多行注释两种方式。单行注释以“#”作为开始符,“#”后面的文字都是注释。例如,在代码清单 5-1 中,代码中即包含单行注释:在屏幕上输出 Hello World!。因此,第 1 行代码实际只会执行 `print("Hello World!")`。

```
1 print("Hello World!")           #在屏幕上输出 Hello World!
```

 注意

虽然在编写程序时是否对代码添加注释不会影响程序的实际运行结果,但良好的注释将有助于增强程序的可读性、从而提高程序的可维护性。建议读者在进行软件开发时,无论多么简单的功能,也一定要加上一些注释来说明实现的思路以及变量、函数和关键语句的作用,这样不仅可以帮助其他开发者快速理解这些代码,也能够帮助开发者本人在一段时间之后仍然能够回忆起当时的实现方法。

5.2.3 多行注释

Python 语言的多行注释以连续的三个单引号('')或三个双引号("")作为开始符和结束符。例如,在代码清单 5-3 中,第 1~第 4 行代码即为用三个连续单引号括起来的多行注释。

代码清单 5-3 多行注释程序。

```
1 '''
2 This is my first Python program
3 Create Date: 07/29/2023
4 '''
```

将其中第 1 行和第 4 行的三个连续单引号改为三个连续双引号,也可以实现同样的多行注释作用。

5.2.4 输入和输出

如图 5-1 所示,任何一个程序都包括输入、输出和数据处理。数据输入/输出形式多样,这里仅介绍键盘输入和屏幕输出。

1. input() 函数

input() 函数的功能是接收标准输入数据(即从键盘输入),返回为 string(字符串)类型,其语法格式如下:

```
input([prompt])
```

其中,“[]”表示其内部的内容 prompt 是一个可选参数,给用户的提示信息。使用 input() 函数时,不给出该参数,则函数运行时没有提示信息,用户直接从键盘输入数据,数据在光标后显示。

以下语句调用 input() 函数让用户输入姓名,并将输入的姓名保存在变量 name 中。

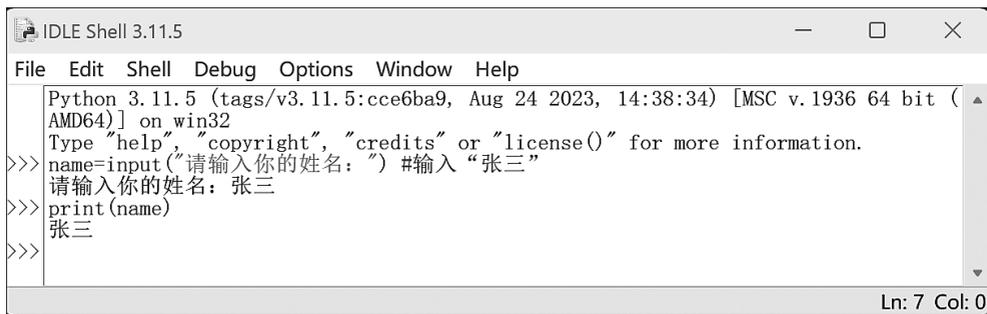
```
name=input("请输入你的姓名: ") #输入“张三”
```

执行上面语句后,屏幕上会显示提示信息“请输入你的姓名:”,此时从键盘上输入“张三”并按回车键,则会将键盘上输入的“张三”保存在变量 name 中。

然后,执行以下语句:

```
print(name)
```

则会在屏幕上显示变量 name 中保存的数据“张三”,如图 5-9 所示。



```
IDLE Shell 3.11.5
File Edit Shell Debug Options Window Help
Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023, 14:38:34) [MSC v.1936 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> name=input("请输入你的姓名: ") #输入“张三”
请输入你的姓名: 张三
>>> print(name)
张三
>>>
```

图 5-9 input() 函数运行结果

2. print() 函数

print() 函数的功能是将各种类型的数据(字符串、整数、浮点数、列表、字典等)输出到屏幕上,其语法格式如下:

```
print(object)
```

其中,对象 object 是要输出的数据。下面的代码展示了 print() 函数的使用方法。

```
1 print("Hello World!")           #输出 Hello World!
2 print(10)                       #输出 10
3 print(3.5)                      #输出 3.5
4 print([1,3,5,'list'])          #输出[1, 3, 5, 'list']
5 print({'1':'A', 2:'B', 3:'C', 4:'D'}) #输出{1: 'A', 2: 'B', 3: 'C', 4: 'D'}
```

提示

上面代码的第 1 行至第 5 行分别输出了字符串、整数、浮点数、列表和字典类型的数据,关于 Python 语言中的数据类型会在第 6 章中介绍。

3. eval() 函数

eval() 函数的功能是计算字符串所对应的表达式的值,返回表达式的计算结果,其语法格式如下:

```
eval(expression)
```

其中,expression 是字符串类型的参数,对应一个有效的 Python 表达式。

提示

eval() 函数的完整语法格式为: eval(expression, globals=None, locals=None)。在实际使用 eval() 函数时,参数 globals 和 locals 通常使用默认值 None。本书在介绍各函数的语法格式时,仅给出其常用的使用方法。关于函数的完整语法格式及各参数说明,请读者参考 Python 语言官方帮助文档。

eval() 函数可以与 input() 函数结合使用,将 input() 函数输入的字符串转换为对应的表达式并计算结果,具体使用方法如下面代码所示。

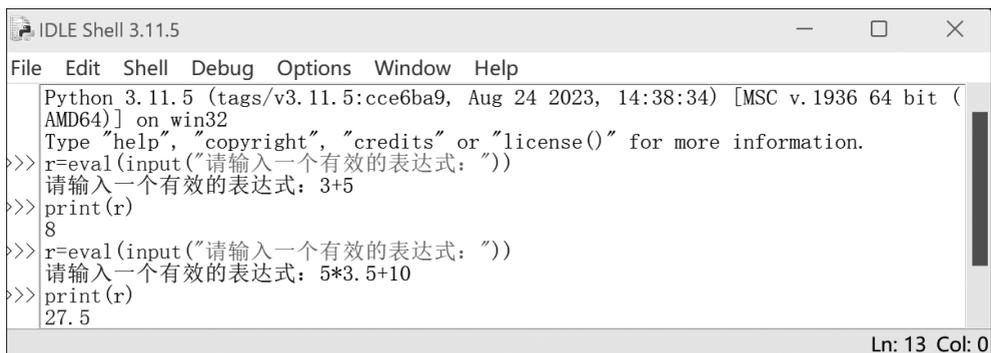
```
r=eval(input("请输入一个有效的表达式: "))
```

运行上面代码后,如果输入 3+5,通过 print(r) 可得到结果 8;如果输入 5 * 3.5+10,通过 print(r) 可得到结果 27.5,如图 5-10 所示;如果输入 5 * /3,则会因其不是一个有效的表达式而报 SyntaxError 错误。

如果不使用 eval() 函数,则输入的是一个字符串,不会进行表达式的计算,如图 5-11 所示。

5.2.5 书写规范

Python 语言通过缩进方式体现各条语句之间的逻辑关系。如代码清单 5-4 所示,与第 2 行相比,第 3 行和第 4 行的行首有缩进(此处是输入了 4 个空格)。因此,从逻辑关系上来说,第 3 行和第 4 行是第 2 行的下一层代码,当第 2 行的 $N\%2==0$ 为 True 时第 3 行和第

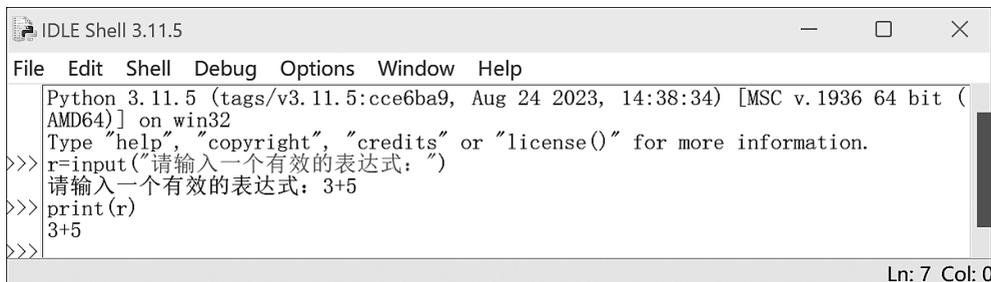


```

Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023, 14:38:34) [MSC v.1936 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> r=eval(input("请输入一个有效的表达式: "))
请输入一个有效的表达式: 3+5
>>> print(r)
8
>>> r=eval(input("请输入一个有效的表达式: "))
请输入一个有效的表达式: 5*3.5+10
>>> print(r)
27.5
Ln: 13 Col: 0

```

图 5-10 eval()函数运行结果 1



```

Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023, 14:38:34) [MSC v.1936 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> r=input("请输入一个有效的表达式: ")
请输入一个有效的表达式: 3+5
>>> print(r)
3+5
>>>
Ln: 7 Col: 0

```

图 5-11 input()函数运行结果 2

4 行代码才会被执行。第 5 行与第 2 行代码的行首都没有缩进,所以二者是同一层上的代码,当 $N\%2==0$ 的值为 False 时,第 5 行代码将会执行。

代码清单 5-4 Python 语言中的强制缩进。

```

1 N=eval(input("请输入一个整数: "))
2     if N%2==0:
3         print("偶数")
4         print("even number")
5     else:
6         print("奇数")
7         print("odd number")

```

运行代码清单 5-4,输入 31 后,将得到如下结果:“奇数”、odd number。

注意

Python 语言对于行首缩进的方式没有严格限制,既可以使用空格也可以使用制表符 (Tab 键),常用对代码进行一个层次缩进的方式有: 1 个制表符,2 个空格,或者 4 个空格。对于同一层次的代码,必须使用相同的缩进方式,否则会报错。例如,如果将代码清单 5-4 中第 3 行行首缩进 2 个空格,而第 4 行行首缩进 4 个空格,则会报如下错误: IndentationError: unexpected indent。