



Python编程基础

Python 是一种跨平台的计算机程序设计语言,也是一种高层次的结合了解释性、编译性、互动性和面向对象的脚本语言。它最初被设计用于编写自动化脚本,随着版本的不断更新和语言新功能的添加,越来越多地用于开发独立的、大型项目。针对 Python 已经开发出广泛的第三程序包和库可供使用,可以让用户用尽可能少的代码实现各种算法。



1.1 Python 编程环境

1.1.1 Python 语言简介

Python 是一种跨平台高级语言,可以用于 Windows、Linux 和 Mac 平台上。Python 语言非常简洁明了,即便是非软件专业的初学者也很容易上手,和其他编程语言相比,实现同一个功能,Python 语言的实现代码往往是最短的。Python 相对于其他编程语言来说,有以下几个优点。

(1) Python 是开源的,也是免费的。开源,也即开放源代码,意思是所有用户都可以看到源代码。Python 的开源体现在程序员使用 Python 编写的代码是开源的,Python 解释器和模块是开源的。开源并不等于免费,开源软件和免费软件是两个概念,只不过大多数的开源软件也是免费软件;Python 就是这样一种语言,它既开源又免费。用户使用 Python 进行开发或者发布自己的程序,不需要支付任何费用,也不用担心版权问题,即使作为商业用途,Python 也是免费的。

(2) 语法简单。和传统的 C/C++、Java、C# 等语言相比,Python 对代码格式的要求没有那么严格,这种宽松使得用户在编写代码时比较轻松,不用在细枝末节上花费太多精力。

(3) Python 是高级语言。这里所说的高级,是指 Python 封装较深,屏蔽了很多底层细

节,比如 Python 会自动管理内存(需要时自动分配,不需要时自动释放)。

(4) Python 是解释型语言,能跨平台。解释型语言一般都是跨平台的(可移植性好),Python 也不例外。

(5) Python 是面向对象的编程语言。面向对象是现代编程语言一般都具备的特性,否则在开发中大型程序时会捉襟见肘。Python 支持面向对象,但它不强制使用面向对象。

(6) 模块众多。Python 的模块众多,基本实现了所有的常见的功能,从简单的字符串处理,到复杂的 3D 图形绘制,借助 Python 模块都可以轻松完成。Python 社区发展良好,除了 Python 官方提供的核心模块外,很多第三方机构也会参与进来开发模块,其中就有 Google、Facebook、Microsoft 等软件巨头。即使是一些小众的功能,Python 往往也有对应的开源模块,甚至有可能不止一个模块。

(7) 可扩展性强。Python 的可扩展性体现在它的模块上,Python 具有脚本语言中最丰富和强大的类库,这些类库覆盖了文件 I/O(输入/输出)、数值计算、GUI、网络编程、数据库访问、文本操作等绝大部分应用场景。这些类库的底层代码不一定是用 Python 编写的,还有很多 C/C++ 语言的身影。当需要一段关键代码运行速度更快时,就可以使用 C/C++ 语言实现,然后在 Python 中调用它们。Python 依靠其良好的扩展性,在一定程度上弥补了运行速度慢的缺点。

1.1.2 Python 编程环境的建立

编写 Python 程序,可以在 Python 自带的交互式界面开发环境中进行,由于自带的开发环境的提示功能和操作功能不强大,Python 程序可以在第三方提供的专业开发环境中编写,例如 PyCharm,然后调用 Python 的解释器运行程序。本书介绍的内容,既可以在 Python 自带的开发环境中进行,也可以在第三方开发环境中进行,由读者根据自己的爱好自行决定。

1. 安装 Python

Python 是开源免费软件,用户可以到 Python 的官网上直接下载 Python 安装程序。登录 Python 的官方网站 <https://www.python.org/downloads/>,其下载页面如图 1-1 所示,可以直接下载不同平台上不同版本的安装程序。最新版本是 3.9.6。Python 的安装程序占用

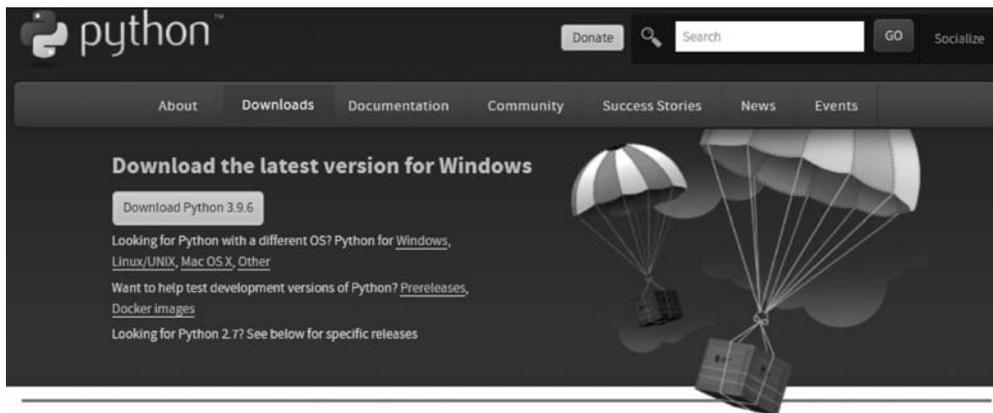


图 1-1 Python 官方下载页面

空间不大,最新 3.9.6 版只有 24.83MB。单击 Downloads,可以找到不同系统下的各个版本的 Python 安装程序。下载 Python 安装程序时,根据自己的计算机是 32 位还是 64 位,选择相应的下载包,例如单击 Windows installer(64-bin)可以下载 64 位的可执行安装程序,一般选择该项即可;单击 Windows embeddable package(64-bit)表示使用 zip 格式的绿色免安装版本,可以直接嵌入(集成)到其他的应用程序中;单击 web-based installer 表示通过网络安装,也就是说下载的是一个空壳,安装过程中还需要联网下载真正的 Python 安装包。Python 安装程序也可以在国内的一些下载网站上找到,例如在搜索引擎中输入“Python 下载”,就可以找到下载链接。

以管理员身份运行 Python 的安装程序 python-3.9.6-amd64.exe,在第 1 步中,如图 1-2 所示,选中 Add Python 3.9 to PATH,单击 Customize installation 项;在第 2 步中,勾选所有项,其中 pip 项专门用于下载第三方 Python 包。单击 Next 按钮进入第 3 步,勾选 Install for all users 项,如图 1-3 所示,并设置安装路径,不建议安装到系统盘中,单击 Install 按钮开始安装。安装路径会自动保存到 Windows 的环境变量 PATH 中,Python 多个版本可以共存在一台机器上。安装完成后,在 Python 的安装目录 Scripts 下出现 pip.exe 和 pip3.exe 文件,用于下载其他安装包。



图 1-2 Python 安装第 1 步

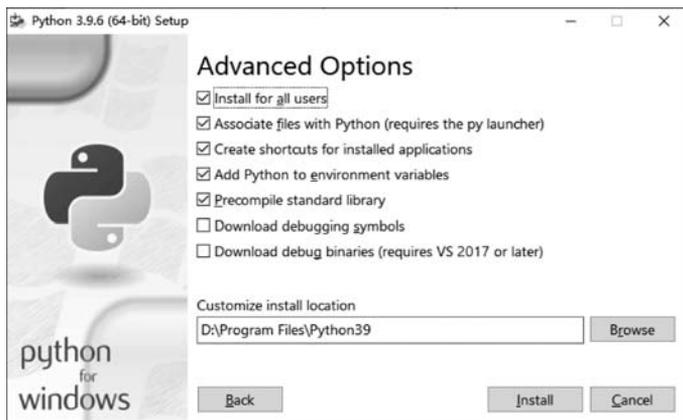


图 1-3 Python 安装第 3 步

安装完成后,需要测试一下 Python 是否能正常运行。从 Windows 的已安装程序中找到 Python 自己的集成开发环境 IDLE,如图 1-4 所示,在“>>>”提示下输入“1+2”或者“print('hello')”并按 Enter 键,如果能返回 3 或者 hello,说明 Python 运行正常。

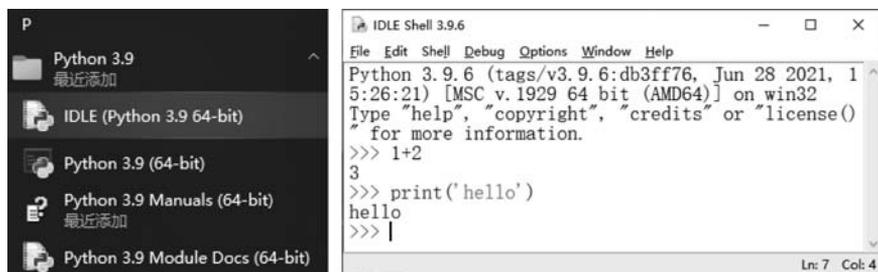


图 1-4 测试 Python

2. 安装科学计算包

安装完 Python 后,接下来需要安装与科学计算有关的包。本书中用到的包有 NumPy、matplotlib、SciPy、SymPy、openpyxl、PyQt5 和 pyinstaller,每个包可以单独安装,也可以一次安装多个。下面介绍 Windows 系统中安装 NumPy 的步骤。以管理员身份运行 Windows 的 cmd 命令窗口,输入“pip install numpy”后按 Enter 键就可以安装 NumPy 包,如图 1-5 所示。也可以用“pip install numpy matplotlib scipy sympy openpyxl pyqt5 pyinstaller”命令一次安装多个包。如果要卸载包,可以使用“pip uninstall numpy”命令。



图 1-5 安装科学计算包

有些安装包比较大,例如 PyQt5 有 53MB,如果直接从国外的网站上下载 PyQt5 可能比较慢,可以使用镜像网站下载,例如清华大学的镜像网站,格式如下:

```
pip install pyqt5 -i https://pypi.tuna.tsinghua.edu.cn/simple
```

3. 安装 PyCharm

如果只是编写简单的程序,在 Python 自带的开发环境中写代码是可以的。但对于专业的程序员来说,其编写的程序比较复杂,在 Python 自带的开发环境中编写代码就有些捉襟见肘了,尤其是编写面向对象的程序,无论是代码提示功能还是出错信息的提示功能远没有专业开发环境的功能强大。PyCharm 是一个专门为 Python 打造的集成开发环境(IDE),带有一整套可以帮助用户在使用 Python 语言开发时提高其效率的工具,比如调试、语法高

亮、项目管理、代码跳转、智能提示、自动完成、单元测试、版本控制等。PyCharm 可以直接调用 Python 的解释器,运行 Python 程序,极大提高 Python 的开发效率。

PyCharm 由 JetBrains 公司开发,可以在 <https://www.jetbrains.com/pycharm/download> 上下载 PyCharm,如图 1-6 所示,PyCharm 有两个版本,分别是 Professional(专业版)和 Community(社区版)。专业版是收费的;社区版是完全免费的,单击 Community 下的 Download 按钮可以下载社区版 PyCharm。在搜索引擎中输入“PyCharm 下载”,也可以在其他下载平台找到 PyCharm 下载链接。

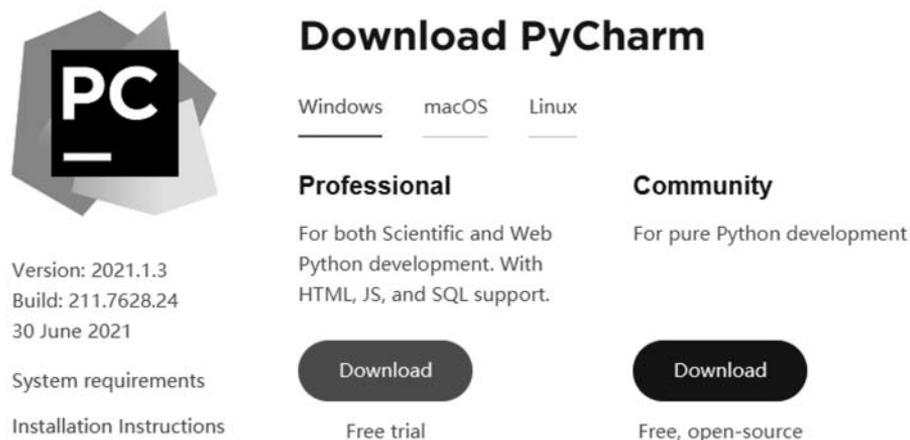


图 1-6 下载 PyCharm 页面

以管理员身份运行下载的安装程序 `pycharm-community-2021.1.3.exe`(读者下载的版本可能与此不同),在第 1 个安装对话框中单击 Next 按钮,在第 2 个安装对话框中设置安装路径,如图 1-7 所示。单击 Next 按钮,在第 3 个安装对话框中勾选“.py”项,如图 1-8 所示,将 py 文件与 PyCharm 关联。如果读者的计算机是 64 位系统,则勾选 64-bit launcher。单击 Next 按钮,在第 4 个安装对话框中,单击 Install 按钮开始安装。最后单击 Finish 按钮完成安装。

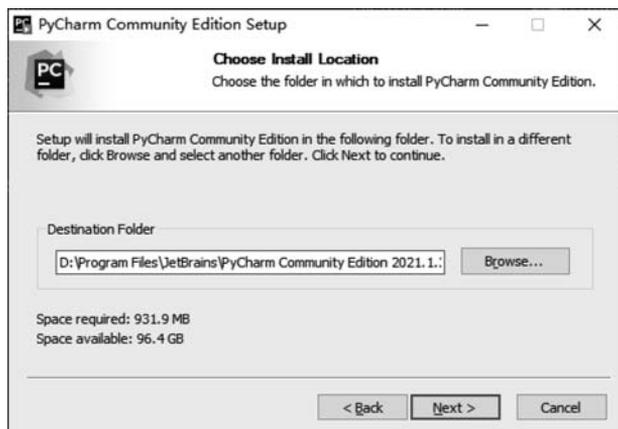


图 1-7 PyCharm 的第 2 个安装对话框

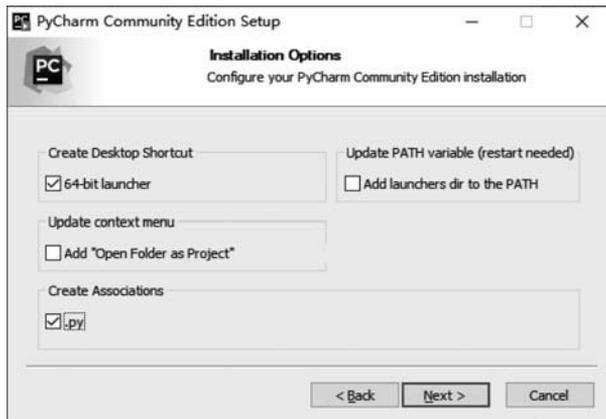


图 1-8 PyCharm 的第 3 个安装对话框



1.2 Python 开发环境使用基础

1.2.1 Python 自带集成开发环境

在安装 Python 时,同时也会安装一个集成开发环境 IDLE,它是一个 Python Shell (可以在打开的 IDLE 窗口的标题栏上看到),在“>>>”提示下逐行输入 Python 程序,每输入一行后按 Enter 键,Python 就执行这一行的内容。前面我们已经应用 IDLE 输出了简单的语句,但在实际开发中,需要编写多行代码时,应在写完代码后一起执行所有的代码,以提高编程效率,为此可以单独创建一个文件保存这些代码,待全部编写完成后一起执行。

在 IDLE 主窗口的菜单栏上选择 File→New File 命令,将打开 Python 的文件窗口,如图 1-9 所示,在该窗口中直接编写 Python 代码。输入一行代码后按 Enter 键,将自动换到下一行,等待继续输入。单击菜单 File→Save 后,再单击菜单 Run→Run Module 或按 F5 键就可以执行,结果将在 Shell 中显示。文件窗口的 Edit 和 Format 菜单是常用的菜单,Edit 菜单用于编辑查找,Format 菜单用于格式程序,例如使用 Format→Indent Region 可以使选中的代码右缩进。单击菜单 Options→Configure IDLE 可以对 Python 进行设置,例如更改编程代码的字体样式、字体大小、字体颜色、标准缩进长度、快捷键等。

在文件窗口中输入下面一段代码,按 F5 键运行程序,在 Shell 窗口中可以输出一首诗,如图 1-9 所示。

```
# Demo 1_1.py
print(' ' * 20)
print(' ' * 10 + '春晓')
print(' ' * 15 + '---- 孟浩然')

print('春眠不觉晓,处处闻啼鸟.')
print('夜来风雨声,花落知多少.')
```



图 1-9 Python 文件窗口和 Shell 窗口

在文件窗口中打开本书实例 Demo1_2.py, 见下面的代码, 按 F5 键后运行程序, 得到一个窗口。对该程序更多的解释, 请参考本书编者所著《Python 基础与 PyQt 可视化编程详解》。

```
import sys # Demo1_2.py
from PyQt5 import QtCore, QtGui, QtWidgets

app = QtWidgets.QApplication(sys.argv)
myWindow = QtWidgets.QWidget()
myWindow.setWindowTitle('Demo1_2')
myWindow.resize(500,400)

myButton = QtWidgets.QPushButton(myWindow)
myButton.setGeometry(150,300,150,50)
myButton.setText('关闭')
str1_1 = ' ' * 10 + '程序员之歌\n'
str1_2 = ' ' * 15 + '---《江城子》改编\n'
str1_3 = ''
十年生死两茫茫,写程序,到天亮.\n\
千行代码,Bug 何处藏.\n\
纵使上线又怎样,朝令改,夕断肠.\n\
领导每天新想法,天天改,日日忙.\n\
相顾无言,惟有泪千行.\n\
每晚灯火阑珊处,程序员,正加班.
'''

peo = str1_1 + str1_2 + str1_3
myLabel = QtWidgets.QLabel(myWindow)
myLabel.setText(peo)
myLabel.setGeometry(50,10,400,300)
font = QtGui.QFont()
font.setPointSize(15)
myLabel.setFont(font)
myButton.setFont(font)
myButton.clicked.connect(myWindow.close)
myWindow.show()
sys.exit(app.exec())
```

1.2.2 PyCharm 集成开发环境

要使 PyCharm 成为 Python 的集成开发环境,需要将 Python 设置成 PyCharm 的解释器。启动 PyCharm,如图 1-10 所示,在欢迎对话框中选择 New Project 项,弹出 New Project 设置对话框,在 Location 中输入项目文件的保存路径,该路径需要是空路径,选中 New environment using,并选择 Virtualenv,从 Base interpreter 中选择 Python 的解释器 python.exe,勾选 Inherit global site-packages 和 Make available to all projects,将已经安装的包集成到当前项目中,并将该配置应用于所有的项目。最后单击 Create 按钮,进入 PyCharm 开发环境。

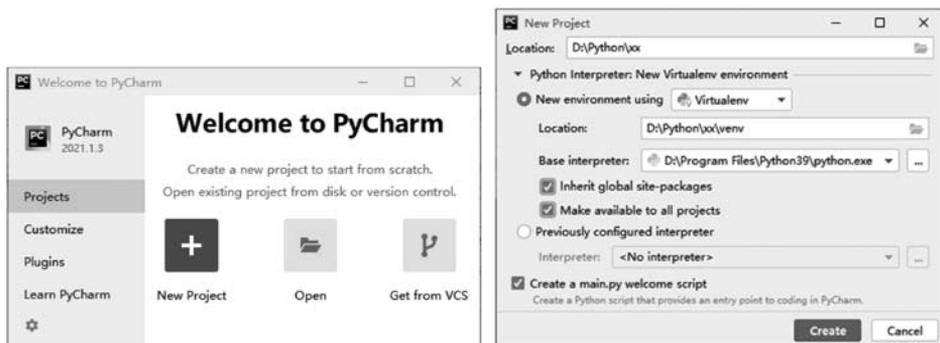


图 1-10 配置 Python 解释器

PyCharm 正常启动后,读者也可以按照下面的步骤添加新的 Python 解释器。单击菜单 File→Settings 打开设置对话框,单击左侧项目下的解释器 Python Interpreter,然后单击右边 Python Interpreter 后面的  按钮,选择“Add...”,弹出添加 Python 解释器的对话框,如图 1-11 所示。在左侧选择 System Interpreter,单击右侧 Interpreter 后的  按钮,弹出选择 Python 解释器的对话框,找到 Python 安装目录下的 python.exe 文件,单击 OK 按钮,回到设置对话框,右边将显示已经安装的第三方程序包。最后单击 OK 按钮关闭所有对话框。

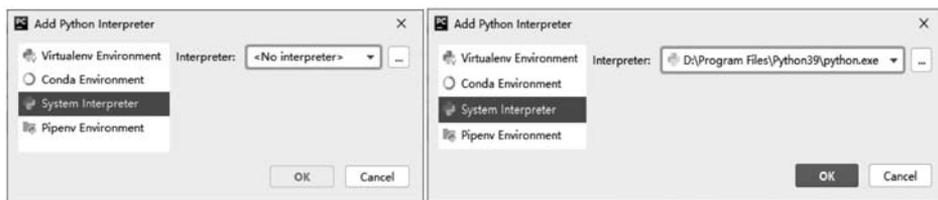


图 1-11 选择 Python 解释器对话框

进入 PyCharm 后,单击 File→New 菜单,然后选择 Python File,输入文件名并按 Enter 键后建立 Python 新文件,输入代码后,要运行程序,需要单击菜单 Run→Run...命令后选择对应的文件,即可调用 Python 解释器运行程序。

在复杂程序中,很难记住每个变量、函数和类的定义位置,因此快速导航和搜索是非常重要的,PyCharm 提供了这些功能。在当前文件中搜索代码段,在 Mac 系统中使用 Cmd+

F 键,在 Windows 和 Linux 系统中使用 Ctrl+F 键;在整个项目中搜索代码段,在 Mac 系统中使用 Cmd+Shift+F 键,在 Windows 和 Linux 系统中使用 Ctrl+Shift+F 键;要搜索类,在 Mac 系统中使用 Cmd+O 键,在 Windows 和 Linux 系统中使用 Ctrl+N 键;要搜索文件,在 Mac 系统中使用 Cmd+Shift+O 键,在 Windows 和 Linux 系统中使用 Ctrl+Shift+N 键;如果不知道要搜索的是文件、类还是代码段,则搜索全部,需要按两次 Shift 键。可使用以下快捷键进行快速导航:前往变量的声明位置,在 Mac 系统中使用 Cmd 键,在 Windows 和 Linux 系统中按住 Ctrl 键不放,然后单击变量;寻找类、方法或文件使用 Alt+F7 键;查看近期更改使用 Shift+Alt+C 键,或者在主菜单中单击 View→RecentChanges;查看近期文件,在 Mac 系统中使用 Cmd+E 键,在 Windows 和 Linux 系统中使用 Ctrl+E 键,或者在主菜单中单击 View→RecentFiles;多次跳转后在导航历史中前进和后退,在 Mac 系统中使用 Cmd+“[”键或 Cmd+“]”键,在 Windows 和 Linux 系统中使用 Ctrl+Alt+“←”键或 Ctrl+Alt+“→”键。



1.3 变量与赋值语句

相对于其他高级语言来讲,Python 的编程语法简单一些,编程也更灵活一些。要学好 Python 编程,必须打好坚实的基础。本节及后续小节讲解 Python 编程的一些基础知识,主要介绍变量、赋值、数据类型、数据类型的转换、转义符、数值表达式、逻辑表达式、if 分支语句、循环语句以及一些常用函数。

1.3.1 变量和赋值的意义

编程软件中的变量(variable)和赋值操作是代码用得最多的符号和操作,变量可以理解成存储数据的一个符号。Python 中可以让同一个变量在不同时刻代表不同类型的数据,但不能同时代表多个数据,只能同时代表一个数据。

在 Python 的 Shell 中输入如下代码。第 1 行和第 2 行中,用变量 a 和变量 b 分别存储数字 2 和 5,c 存储 1+3 计算后的值,即 4,通过第 4 行的 print()函数输出 a、b 和 c 的值,从第 5 行中 print()函数的返回值可以看出 a、b 和 c 的值分别是 2、5 和 4。代码中的“=”表示赋值,将“=”右边的值或者表达式的值赋给“=”左边的变量。初学编程的人员可以简单地把“=”理解成“=”左边的变量等于“=”右边的值或表达式的值。

```
1 >>> a = 2
2 >>> b = 5
3 >>> c = 1 + 3
4 >>> print(a, b, c)
5 2 5 4
```

其实对于赋值运算“=”,计算机内部有更深入的操作,对于第 1 句 a=2 和第 2 句 b=5,计算机处理这两句赋值运算是先在内存中开辟两个空间,分别记录数值 2 和 5,然后把记录数值 2 和记录数值 5 的内存空间的起始地址分别赋予变量 a 和 b;对于第 3 句 c=1+3,

是先在内存中开辟两个空间,分别记录数值 1 和 3,然后把 1 和 3 读入 CPU 中,进行 $1+3$ 运算,得到结果 4,然后再在内存中开辟一个新的空间,把 4 输出到这个新空间中,最后把 4 所在的内存空间的起始地址赋值给变量 c,第 4 行的 `print(a,b,c)` 函数是通过 a、b、c 所记录的内存地址,从内存中读取对应地址的值并输出。

对于内存中存储的数据的起始地址可以通过 `id()` 函数获取。在 Shell 中继续输入如下代码,可以看出数值 2 和变量 a 的内存地址是相同的,数值 5 和变量 b 的内存地址是相同的,数值 4 和变量 c 的内存地址也是相同的。

```
6 >>> id(2),id(a)
7 (140715084994240, 140715084994240)
8 >>> id(5),id(b)
9 (140715084994336, 140715084994336)
10 >>> id(4),id(c)
11 (140715084994304, 140715084994304)
```

在 Shell 中继续输入如下代码。其中第 12 行代码为 `a = a+6`,在前面的第 1 行中已经将 2 赋予了变量 a,计算 `a=a+6` 是先计算“=”右边的 `a+6`,得到 8,然后将 8 重新赋值给 a, a 的值变成了 8,由第 13 行和第 14 行可以看出 a 的值是 8,已经不是 2 了。第 12 行 `a=a+6` 的计算过程是,先在内存中开辟一个空间存储 6,然后把变量 a 指向的内存空间的值(这个值是 2)和内存空间中的 6 读入 CPU 中,进行 $2+6$ 的计算,得到结果 8,然后再在内存空间中开辟一个空间,把结果 8 存储到这个新空间中,通过赋值操作“=”,将结果 8 的存储地址赋给变量 a,变量 a 已经不指向 2 的存储地址,所以 a 的值最后是 8,通过第 15 行和第 16 行的代码可以看出,8 和 a 的地址是相同的。第 17 行代码 `a=a+b+c`,先计算赋值操作“=”右边的表达式 `a+b+c` 的值,由于 a 指向 8,b 指向 5,c 指向 4,CPU 从内存中读取 8、5 和 4,完成 $8+5+4$ 的计算,将结果 17 保存在内存中。通过赋值操作,将 a 重新指向结果 17 的内存地址,通过第 20 行代码和第 21 行的返回值可以看出,变量 a 和数值 17 的地址是相同的。

```
12 >>> a = a+6
13 >>> print(a)
14 8
15 >>> id(8),id(a)
16 (140715084994432, 140715084994432)
17 >>> a = a+b+c
18 >>> print(a)
19 17
20 >>> id(17),id(a)
21 (140715084994720, 140715084994720)
```

1.3.2 变量的定义

Python 中的变量在使用时,不需要提前声明,定义变量名称时需要注意以下几方面的事项。

- Python 区分变量名称的大小写,例如 A 和 a 是两个不同的变量。