

高等院校计算机应用系列教材

# Python 语言程序设计

田晖 应晖 编著

清华大学出版社

北 京

## 内 容 简 介

Python 是目前深受用户欢迎的程序设计语言之一。由于 Python 语言具有简洁性、易读性及可扩展性等优点，国内外采用 Python 进行科学计算的研究机构日益增多，一些知名大学已经采用 Python 来教授程序设计课程。

本书从入门学习者的角度出发，详细介绍了 Python 语言的基础语法。全书可分成三大部分：第 1 部分包括第 1~6 章，主要介绍了 Python 语言程序设计的三种基本结构；第 2 部分包括第 7~10 章，主要介绍了 Python 语言的序列结构、面向对象的思考方式、异常的处理及文件的读写；第 3 部分，即第 11 章，以贪吃蛇游戏为例介绍了基于 Python 的完整项目开发过程。本书语言通俗易懂，提供了大量的编程实例，引导初学者领悟编写程序的方法，并在每一章的最后提供了大量的课后习题，供读者检查自己的学习成果。

本书可作为高等院校程序设计语言通识课的教材，也可作为 Python 语言入门读者的学习参考书。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。举报：010-62782989，beiqinquan@tup.tsinghua.edu.cn

### 图书在版编目(CIP)数据

Python 语言程序设计 / 田晖，应晖编著. —北京：清华大学出版社，2022.1

高等院校计算机应用系列教材

ISBN 978-7-302-59636-3

I. ①P… II. ①田… ②应… III. ①软件工具—程序设计—高等学校—教材 IV. ①TP311.561

中国版本图书馆 CIP 数据核字(2021)第 248802 号

责任编辑：王 定

封面设计：高娟妮

版式设计：孔祥峰

责任校对：成凤进

责任印制：丛怀宇

出版发行：清华大学出版社

网 址：<http://www.tup.com.cn>，<http://www.wqbook.com>

地 址：北京清华大学学研大厦 A 座 邮 编：100084

社总机：010-62770175 邮 购：010-62786544

投稿与读者服务：010-62776969，[c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

质量反馈：010-62772015，[zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印 装 者：北京同文印刷有限责任公司

经 销：全国新华书店

开 本：185mm×260mm 印 张：17 字 数：435 千字

版 次：2022 年 2 月第 1 版 印 次：2022 年 2 月第 1 次印刷

定 价：59.80 元

---

产品编号：091284-01

# 前 言

Python 语言近些年风生水起，程序员们在体会了 Python 第三方库带来的各种好处之后，不断地为 Python 添加好的口碑，并且带动更多的程序员投入 Python 应用的开发之中。业界也公认：Python 语言易学、易懂、易用。

Python 语言实现应用是一件简单的事情。有一种广为流传的说法：使用 C 语言开发人脸识别需要 5000 行代码，使用 Java 语言需要 1000 行代码，使用 Python 只需要 7 行。这种说法可能过于夸张，因为 7 行代码只是调用第三方库。第三方库的开发者的工作量并没有被统计进去。但是，作为 Python 语言的初学者，与学习其他语言的初学者相比，的确是站在了“巨人”的肩膀之上，而且这个“巨人”还在成长，未来的肩膀必然会更高、更宽。近些年，人工智能的火爆成为 Python 口碑爆炸的背后推手，Python 在人工智能领域的应用也是基于各种第三方库被不断推出、推广和完善。

从实际应用出发，很多人学习 Python 的目的就是使用 Python 快速实现某些场景的应用开发。以强大的 Python 算法库为基础，大多数编程爱好者在经过简单的 Python 语言学习之后，即可实现自己的开发目标。

本书作者都是华侨大学的一线教师。华侨大学是中央统战部直属，由中央统战部、教育部、福建省人民政府共同建设的普通高等学校，是 1960 年经周恩来总理批准创办的我国唯一所以“华侨”命名的高等学府。华侨大学每年招收近 7000 名新生，其中包括 1500 多名港澳台侨及留学生。华侨大学的教师们每年都会面对不同基础、不同语言、不同文化背景的学生。科学与技术学院的教师们在几十年的教学过程中积累了丰富的教学经验，结合各类学生学习程序设计语言的特点，以“易教、易学、易懂、易做”为目标，历经数载，最终完成了本书的编写。

本书具有以下特点：

(1) 本书内容浅显易懂，图文并茂，力图通过通俗易懂的例题带领读者快速进入 Python 开发领域。

(2) 本书设计了 100 多道典型例题，在例题讲解时，以行为单位解析代码，不仅帮助读者了解代码的含义而且启发读者掌握程序设计的思维方式。

(3) 本书的例题讲解已被录制成视频，读者研读本书的时候可在视频的帮助下加深理解。

(4) 本书例题的源代码可以扫二维码获取。

(5) 本书提供了大量的精选习题（参考答案可以扫二维码获取），读者可在完成每一章的学习之后，对自己的学习成果进行检测。

(6) 本书免费提供教学课件、电子教案、教学大纲，扫二维码即可下载。

本书是集体智慧的结晶，由田晖、应晖编著，衷心感谢为本书出谋划策、提供帮助的华侨大学计算机公共基础教学部全体教师。同时，感谢李美丽和毕司祈同学参与本书讲解视频的录制。

由于 Python 语言程序设计的内容涉及面广，要将其很好地贯穿起来难度较大，加之作者水平有限，书中不足之处在所难免，恳请专家、学者和广大读者多提宝贵意见。



教学课件



电子教案



教学大纲



例题源代码



习题参考答案

编 者  
2021 年 10 月

# 目 录

## 第 1 部分 基础篇

第 1 章 Python 概述	3
1.1 Python 简介	3
1.2 学习环境的搭建	5
1.2.1 Python 的安装	5
1.2.2 PyCharm 的安装	11
1.3 HelloWorld 程序	14
1.4 程序注释	23
1.4.1 单行注释	23
1.4.2 多行注释	23
1.5 课后习题	24
第 2 章 Python 基本编程	31
2.1 使用变量进行编程	33
2.2 使用键盘输入	35
2.2.1 input() 函数	35
2.2.2 int() 函数	36
2.2.3 eval() 函数	37
2.3 标识符与关键字	38
2.3.1 标识符	38
2.3.2 关键字	38
2.3.3 标识符使用规则	39
2.4 赋值	40
2.4.1 变量与常量	40
2.4.2 单个变量的赋值	40
2.4.3 多个变量的赋值	40
2.5 数学运算符的使用	42
2.5.1 四则运算符及括号的使用	42
2.5.2 特殊运算符的使用	43
2.6 课后习题	46
第 3 章 函数与字符串	55
3.1 内置函数的介绍	55
3.2 导入模块库与调用模块函数	56
3.2.1 math 模块库	56
3.2.2 time 模块库	58
3.2.3 random 模块库	59
3.3 ASCII 与 Unicode 介绍	60
3.3.1 ord() 函数	63
3.3.2 chr() 函数	64
3.4 字符串	65
3.4.1 连接字符串	65
3.4.2 转义字符与特殊字符构成的字符串	66
3.4.3 print() 函数中的 end 参数	68
3.4.4 连接字符串与数字	68
3.5 课后习题	70
第 4 章 选择结构	81
4.1 布尔值与比较运算符	81
4.1.1 布尔值	82
4.1.2 比较运算符	82
4.2 if 选择语句	83
4.3 if··else 分支语句	85
4.4 if··elif··elif··else 多路分支语句	88
4.5 逻辑运算	93
4.6 课后习题	95
第 5 章 循环结构	107
5.1 while 循环的要点	107
5.2 for 循环的要点	112

5.3	嵌套循环	115	7.3.1	元组中元素的访问	177
5.4	break和continue关键字	118	7.3.2	列表、元组之间的转换	178
	5.4.1 break 关键字	118	7.4	字典	179
	5.4.2 continue 关键字	120	7.4.1	创建字典	179
5.5	循环结构例题讲解	121	7.4.2	对字典中的键值对进行操作	179
5.6	课后习题	126	7.5	集合	180
<b>第6章</b>	<b>函数</b>	<b>141</b>	7.6	字符串的序列操作	182
6.1	定义、调用函数的方法	141	7.7	课后习题	184
	6.1.1 定义函数的方法	141	<b>第8章</b>	<b>类与对象</b>	<b>189</b>
	6.1.2 同一源文件中调用函数的方法	142	8.1	类与对象的概念	189
	6.1.3 在不同源文件中调用函数	143	8.2	定义类的方法	190
6.2	return关键字的使用	144	8.3	对象及对象的使用	192
	6.2.1 返回None	144	8.4	类的继承	193
	6.2.2 使用return返回计算结果	145	8.5	面向对象的编程	197
6.3	使用参数向函数传递值	149	8.6	UML图纸(类图模型)	198
	6.3.1 位置参数	150	8.7	课后习题	201
	6.3.2 关键字参数	150	<b>第9章</b>	<b>异常</b>	<b>207</b>
	6.3.3 默认参数	151	9.1	异常的概念	207
	6.3.4 不定长参数	153	9.2	异常的处理	209
6.4	变量的作用域范围	153	9.3	抛出异常	211
6.5	模块化设计程序	155	9.4	使用自定义类来处理异常	213
6.6	模块化实现代码	157	9.5	课后习题	214
6.7	函数的递归调用	158	<b>第10章</b>	<b>文件处理</b>	<b>219</b>
6.8	课后习题	160	10.1	文本的读写操作	219
			10.1.1	文件的目录结构	219
			10.1.2	打开一个文件	220
			10.1.3	将信息写入一个文件	220
			10.1.4	从文件中读取信息	222
			10.2	使用第三方库进行读写	223
			10.2.1	第三方库	223
			10.2.2	导入第三方库的方法	224
			10.2.3	使用第三方库写入Excel文件	226
			10.2.4	使用第三方库读Excel文件 信息	235
			10.3	课后习题	237

## 第2部分 提高篇

<b>第7章</b>	<b>Python的序列结构</b>	<b>171</b>
7.1	序列	171
	7.1.1 序列的索引	171
	7.1.2 序列的通用操作	171
7.2	列表	172
	7.2.1 列表中元素的访问	173
	7.2.2 向列表中添加元素	174
	7.2.3 从列表中删除元素	175
	7.2.4 修改列表元素	176
	7.2.5 序列的遍历	176
7.3	元组	177

## 第 3 部分 实战篇

第 11 章 一个小游戏	243
11.1 pygame 的安装	243
11.2 创建游戏窗体	244
11.3 设置窗体	246
11.4 Snake 类的设计与实现	250
11.4.1 Snake 类的设计	250
11.4.2 Snake 类的结构	250
11.4.3 Snake 类中的 addBlock() 函数	251
11.4.4 Snake 类中的 move() 函数	252
11.4.5 Snake 类中的 isDead() 函数	252
11.4.6 Snake 类中的 turn() 函数	253
11.4.7 Snake 类的构造函数调用 addBlock() 函数	253
11.5 Food 类的设计与实现	253
11.5.1 Food 类的设计	254
11.5.2 Food 类的结构	254
11.5.3 Food 类中的 beEaten() 函数	254
11.6 贪食蛇的实现	255
11.7 游戏代码	257
参考文献	261

# 第 1 部分 基础篇



# 第 1 章

## Python概述

很多同学会问一个问题：“哪一种计算机编程语言的使用范围最广？”PYPL(Popularity of Programming Language, 编程语言流行指数)网站(<https://pypl.github.io/PYPL.html>)于 2021 年 7 月发布了图 1-1 所示的调查结果, 回答了这个问题。

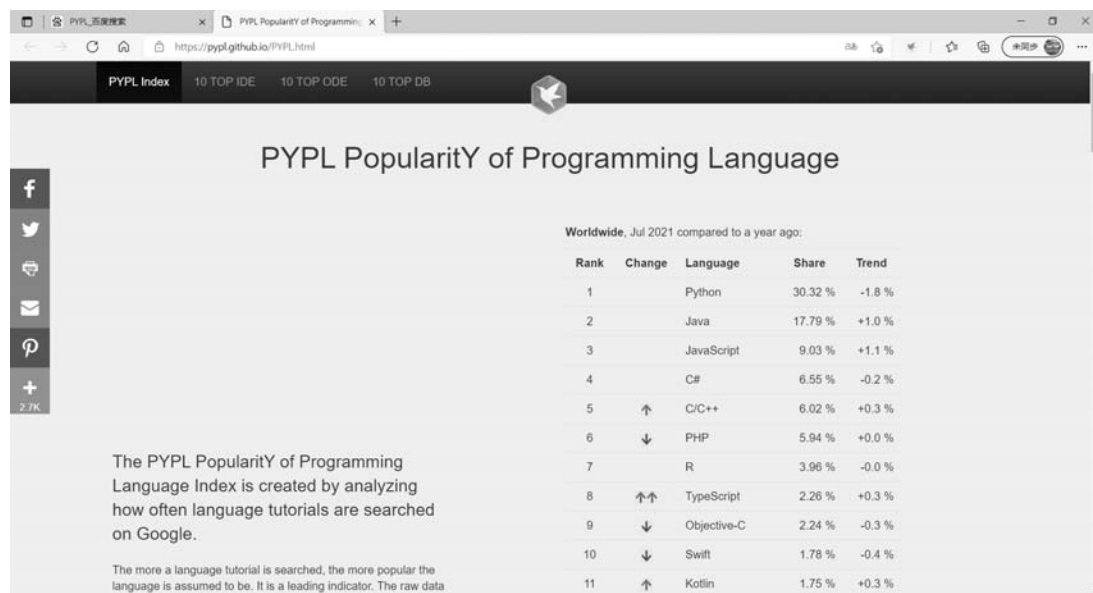


图 1-1 计算机编程语言流行指数 Top10

由图 1-1 能够看出, Python 语言风头正盛。

事实上, 每种语言都有自己擅长的领域。编程语言, 特别是高级语言, 没有绝对的好与不好, 但是对于初学者来说, Python 无疑是一个较好的入门选择。

### 1.1 Python 简介

Python 语言诞生于 1991 年, 由荷兰人吉多·范罗苏姆(Guido Van Rossum, 见图 1-2)创造。英国有一部电视喜剧集, 名为 *Monty Python's Flying Circus*(巨蟒飞行马戏团)在当时颇为流行。

吉多很喜欢追这个剧，因此吉多将他创造的语言命名为 Python，实际上是希望 Python 语言能够像这部喜剧集一样在西方世界流行起来。在 30 年后的今天，Python 成为目前世界上最流行的高级语言之一。

Python 是一种通用语言，它的目标是服务于任何编程任务。以下是几个使用 Python 的知名机构与场景：

- (1) NASA 使用 Python 进行数据分析与运算；
- (2) 谷歌使用 Python 作为网络应用的后端；
- (3) 雅虎使用 Python 完成了群组交流平台；
- (4) 豆瓣所有的业务都是 Python 开发的；
- (5) 大家常用的网络下载工具 BitTorrent 是使用 Python

完成的：

- (6) 一些游戏中使用了 Python，如《战地 2》《文明 4》。

计算机程序设计语言的发展经历了机器语言、汇编语言和高级语言三个阶段。

(1) 机器语言是由 0、1 组成的，不同的 CPU 有不同的指令集，难学、难记、难用，使用机器语言的编程效率很低，不利于计算机语言的普及。

(2) 汇编语言是机器语言的一种进化，是人类为了提高计算机编程效率而做出的第一次翻译进化尝试。汇编语言与机器语言存在一一对应的关系。机器语言存在的问题，汇编语言依然存在，虽然人类能够读懂汇编语言，能够编写汇编程序，但是依然难以大量创建计算机应用程序。

(3) 高级语言可以独立于计算机本身，在形式上更接近英文(因此学计算机编程必须有英文基础)，使用通俗易懂的单词作为关键字；一条高级语言程序能取代若干条汇编语句、机器语句。Python 的某些命令能代替几百条，甚至几千条机器语言语句(如 turtle 模块中的绘图命令)。

Python 和 C、Java 一样，都是高级语言，不同的地方在于，C 与 Java 使用编译器把源程序编译为计算机目标文件，然后让计算机执行这个目标文件，而 Python 使用的则是翻译器。Python 源代码完成之后，翻译器逐条将源代码翻译为计算机可执行的机器码，并让计算机逐条执行。因此，Python 的运行效率要低于 C 和 Java。

Python 是一种面向对象的计算机语言。面向对象是一种编程方法。C 是面向过程的，Java 是面向对象的，Python 也是面向对象的。Java 更倾向于团队协作完成大型开发项目，而 Python 则被更多的小型开发者所使用。Python 中的数据类型是基于对象创建，更接近人类思维模式。在对象类中，定义有对象属性与对象方法，因此 Python 的可重用性得到了保障。同一类型的多个对象能使用同样的对象方法进行操作。

最初的 Python 到目前的 Python 3.9 版本，Python 在不断地进化。目前使用较多的是 Python 3 以上版本。需要指出的是，Python 3 版本并不向下兼容 Python 2 版本。全球存在许多活跃的 Python 网上社区，不断地推出 Python 的各种第三方模块，数据分析领域、数据可视化编程、游戏开发、文件读写，甚至神经网络模型搭建，都有成熟的第三方模块可以调用。感兴趣的同学可以从网上获取各种资源进行研究。



图 1-2 吉多·范罗苏姆

## 1.2 学习环境的搭建

进行程序设计的时候，需要安装编程语言的编程环境。条件允许的话，可以安装该语言的集成开发环境。本书主要讲解 Python 的编程方法，尽管 Python 有自带的内置集成开发工具 IDLE，但是在本书中，我们会以功能更加强大的集成开发环境 PyCharm 作为编程工具。本节将介绍 Python 和 PyCharm(Community)的安装。

### 1.2.1 Python 的安装

在 Microsoft Edge 浏览器的地址栏输入 Python 的官网地址 <https://www.python.org/>，进入 Python 官网。本书中使用的 Python 环境为 Python 3.9.1，在 Python 官网的 Downloads 页面即可下载该编程环境的安装包。本书以 Windows 10 操作系统为例，介绍 Python 的安装方法。

Python 的安装步骤如下。

(1) 打开 Python 官网(<https://www.python.org/>)的欢迎页面，如图 1-3 所示。

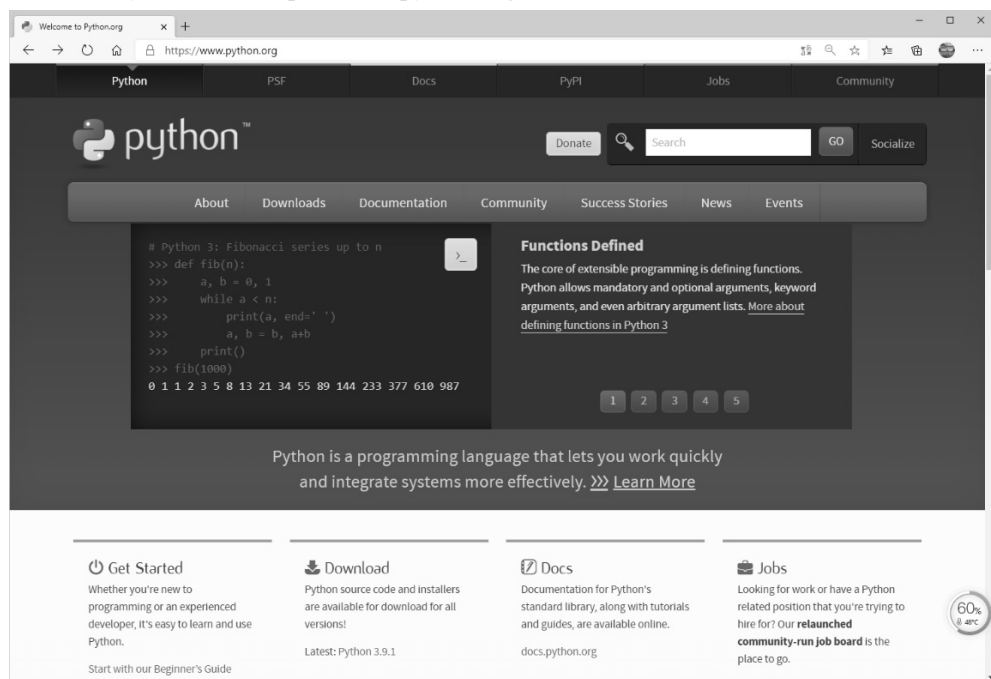


图 1-3 Python 官网的欢迎页面

(2) 根据自己的计算机选择对应的操作系统。例如使用 Windows 10 操作系统，可以在 Downloads 页面选择 Windows(如果使用 Mac 系统，可以选择 Mac OS X)，如图 1-4 所示。

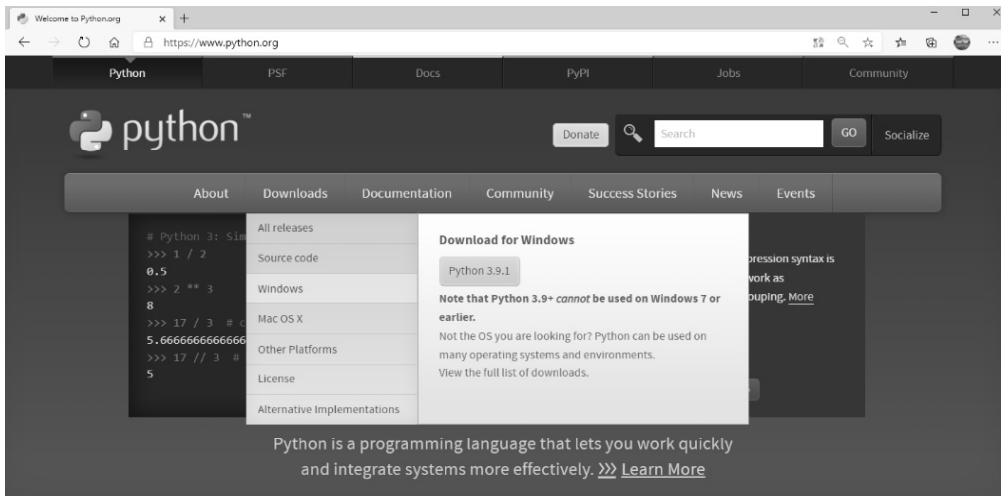


图 1-4 选择对应的操作系统

编写本书的时候，Python 的最新版本为 Python 3.9.1，这个版本的 Python 已经无法在 Windows 7 上使用了。同学们可以选择下载最新版的 Python 安装包，或者根据自己的计算机操作系统的情况，选择合适的 Python 安装包。

(3) 如果希望下载最新版本的安装包，单击图 1-4 中的 Python3.9.1 按钮开始下载，Edge 中将出现图 1-5 所示的下载提示。

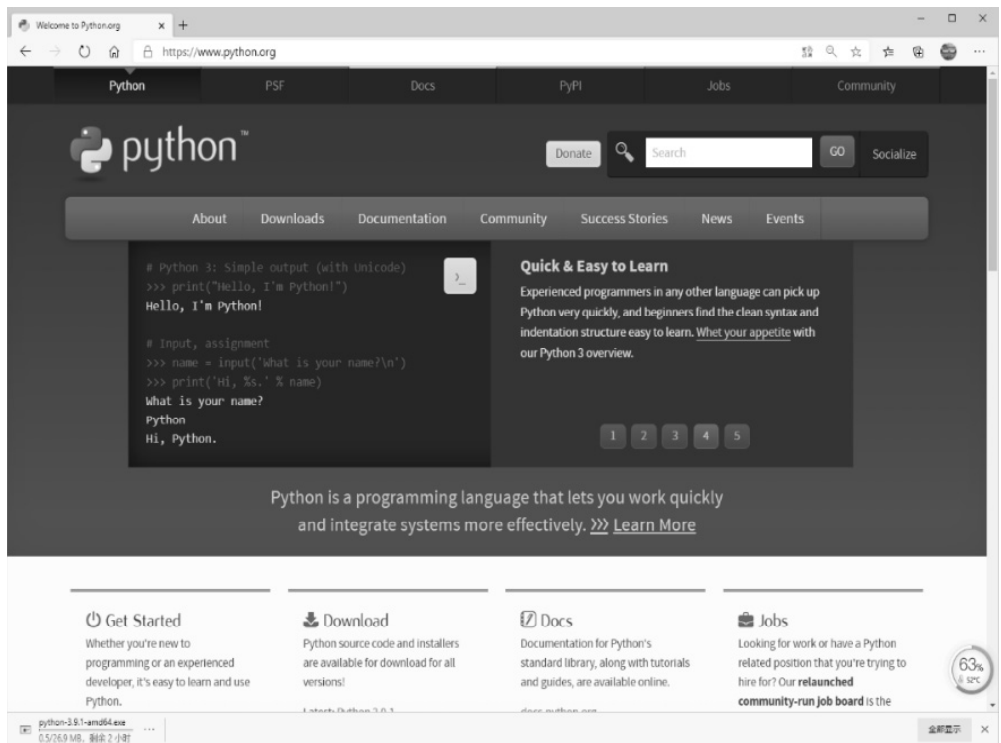


图 1-5 下载提示

(4) 下载完成，如图 1-6 所示，单击“打开文件”，即可开始安装过程。

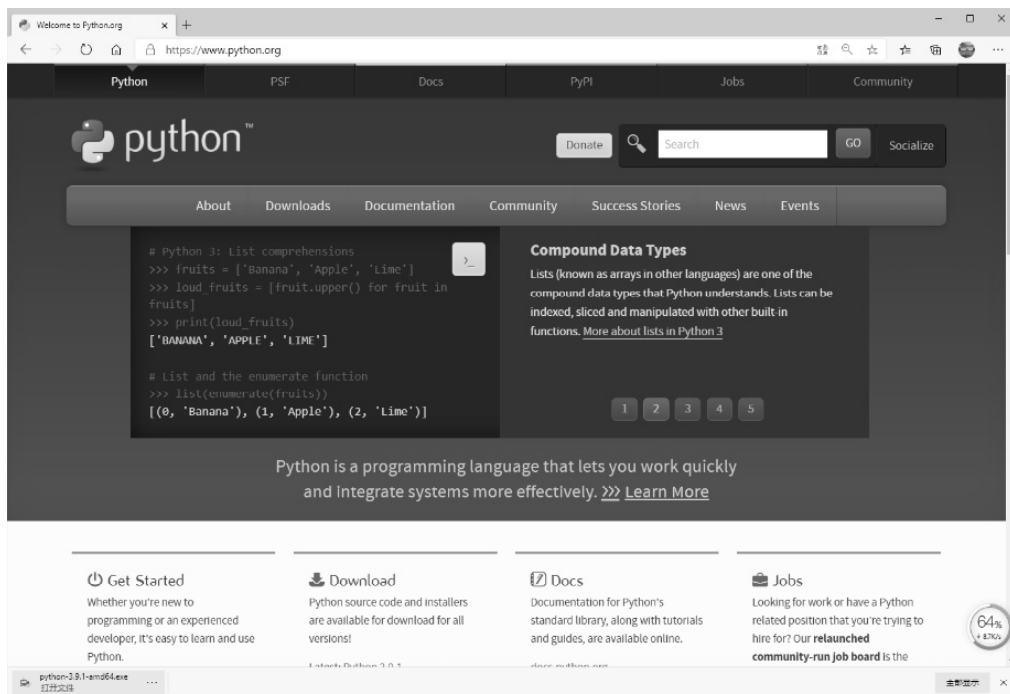


图 1-6 下载完成

也可以双击已经下载好的 Python 安装文件进行安装，如图 1-7 所示。

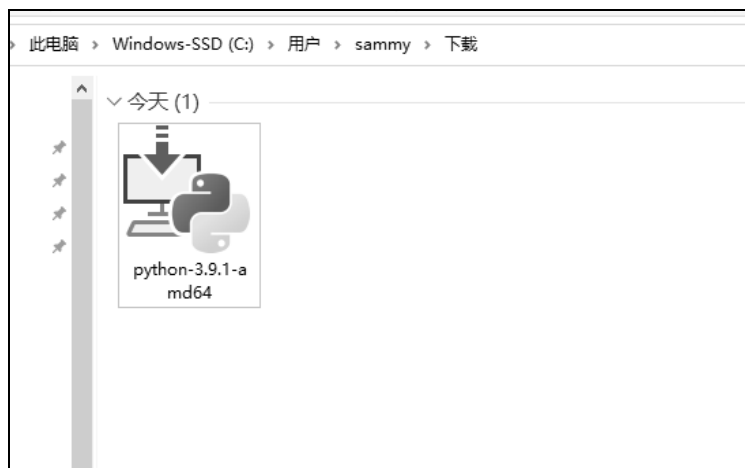


图 1-7 双击 Python 安装文件进行安装

如果想下载其他版本的安装文件，可以访问 <https://www.python.org/downloads/windows/>，在网页上选择下载希望使用的 Python 安装文件进行安装，如图 1-8 所示。

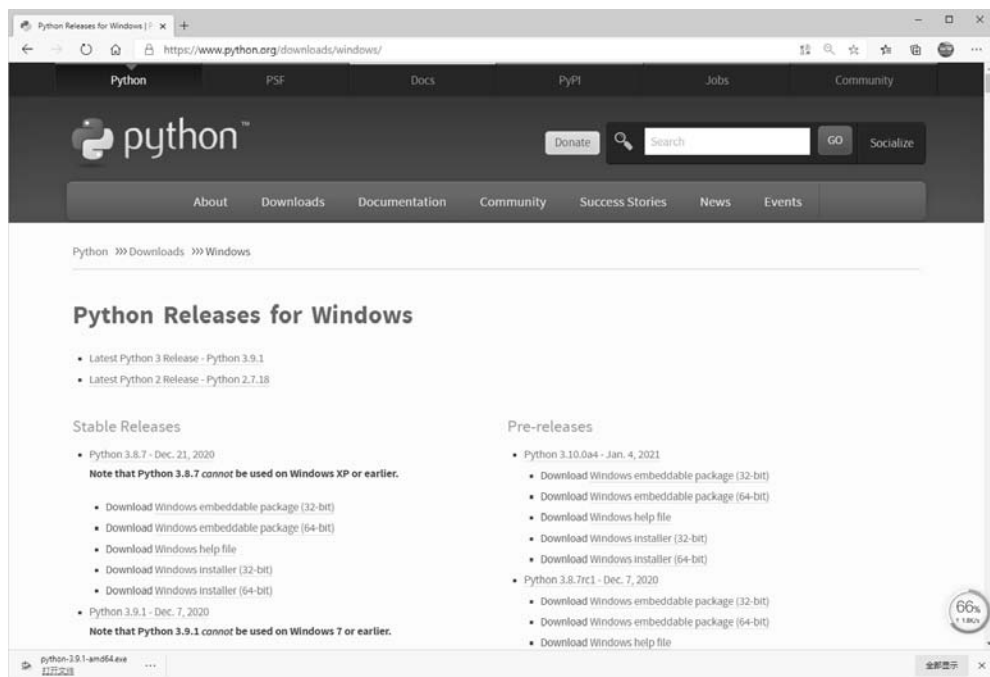


图 1-8 其他版本的 Python 安装文件

如果想下载适合在 Mac OS 系统上使用的 Python，可以访问 <https://www.python.org/downloads/mac-osx/>，在网页上选择并下载希望使用的 Python 版本安装文件进行安装，如图 1-9 所示。

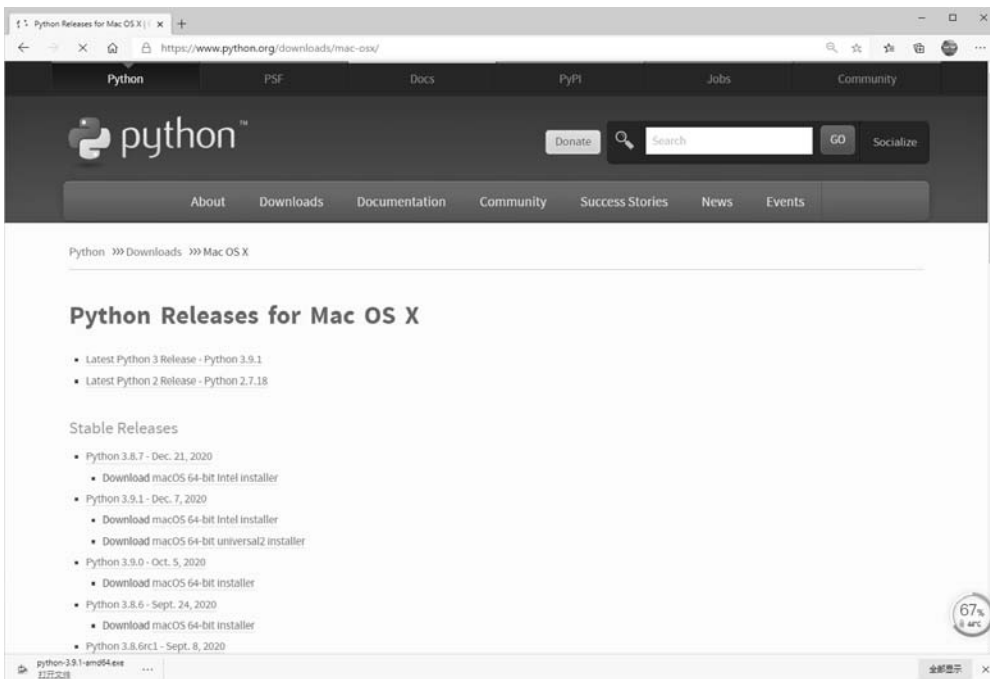


图 1-9 可以在 Mac OS 系统上安装的 Python 版本

(5) 下载完成后，在本地计算机上安装 Python 编程环境，单击“运行”按钮开始安装，如图 1-10 所示。



图 1-10 运行 Python 安装文件

(6) 首先勾选 Add Python 3.9 to PATH 选项，如图 1-11 所示，此步骤非常关键。勾选了这个选项之后，Windows 系统会自动配置环境变量。设置好之后，使用命令行窗口命令的时候，会自动查看 Python 安装路径文件夹中的文件。

单击 Install Now 就可以将 Python 安装至默认目录。如果需要将 Python 安装到自己指定的文件夹中，则需要单击 Customize installation。此处建议选择默认目录安装，如图 1-11 所示。

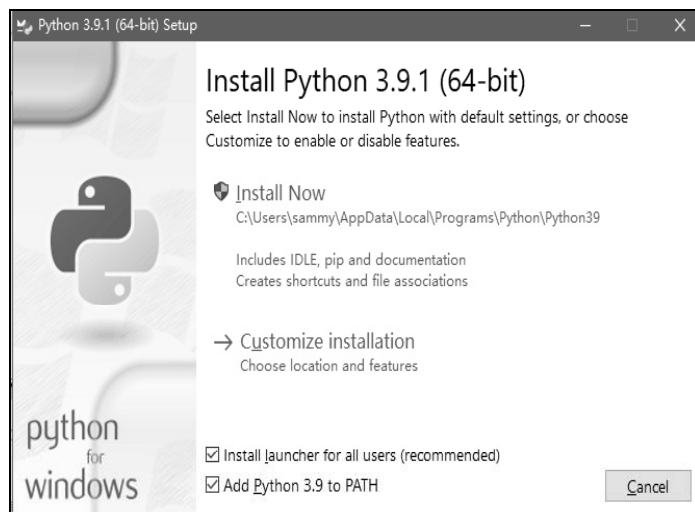


图 1-11 将 Python 安装至默认目录

在 Windows 系统中安装 Python 的过程如图 1-12 所示。

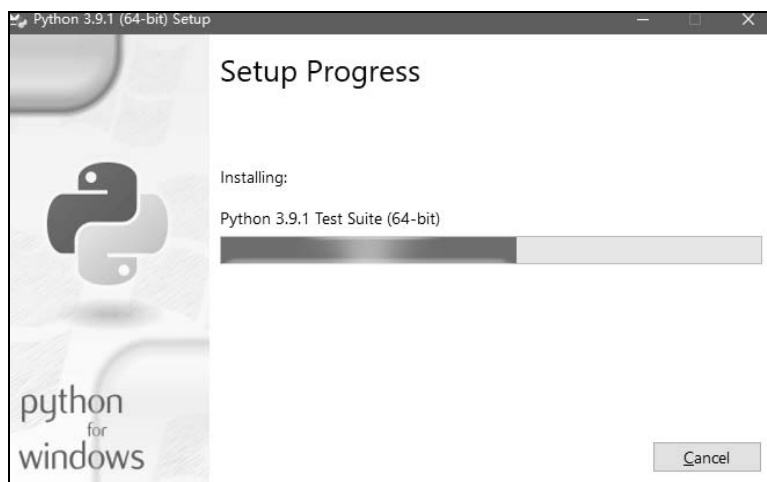


图 1-12 安装 Python 的过程

(7) 完成安装后，会出现安装成功的提示对话框，如图 1-13 所示。

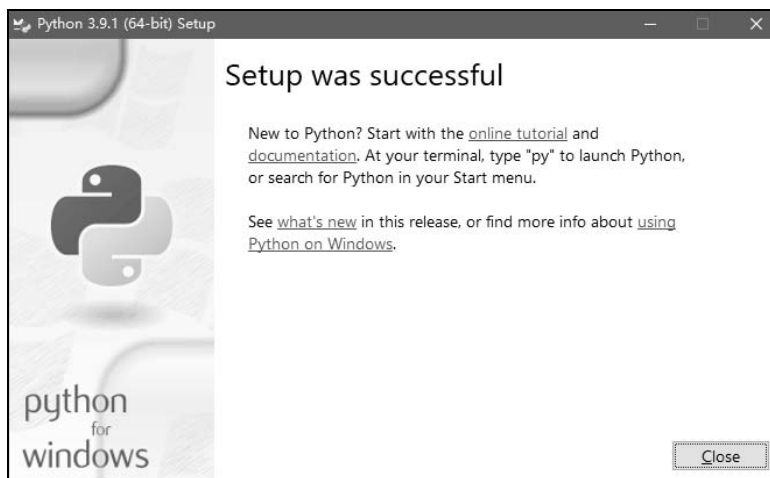


图 1-13 安装成功

这时，在 Windows 的开始菜单中，最近添加的程序中会出现图 1-14 所示的 4 个快捷方式。

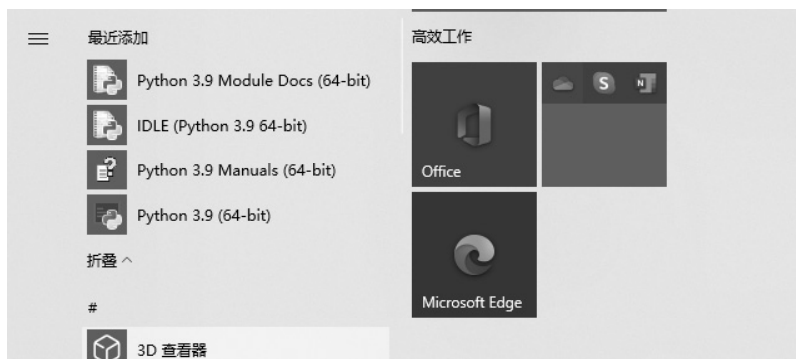


图 1-14 4 个快捷方式



4 个快捷方式的说明如下：

- Python 3.9 Module Docs：模块说明文档。
- IDLE：Python 自带的简易开发环境，可供安装 Python 的计算机使用。
- Python 3.9 Manuals：Python 手册。
- Python 3.9：运行 Python。单击 Python 3.9 就可以打开一个运行 Python 的命令行窗口，如图 1-15 所示。

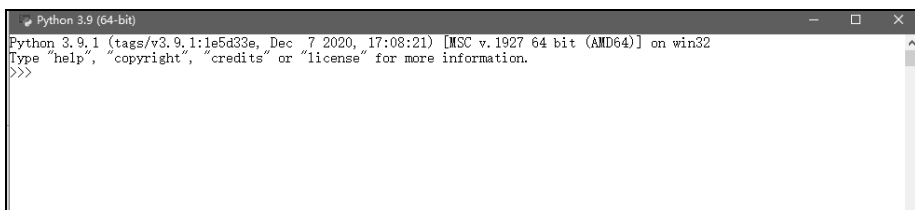


图 1-15 运行 Python 的命令行窗口

图 1-15 中显示了安装的 Python 版本号及符号“>>>”，说明执行 Python 3.9 命令成功，这时 Python 编程环境已经可以使用了。

例如，输入 1+2 后，按 Enter 键，运行结果如图 1-16 所示，说明 Python 已经处于可用状态。

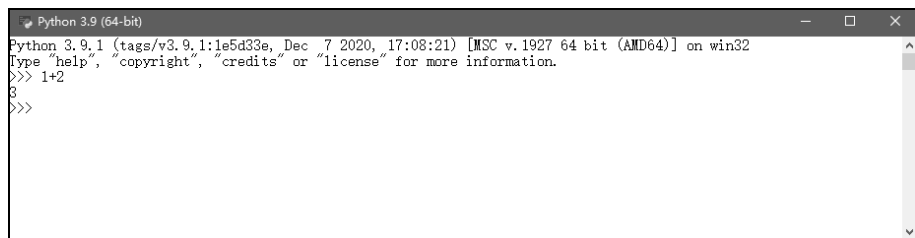


图 1-16 运行 Python 命令

如果命令行窗口中的命令运行失败，极大可能是在安装的时候未选择 PATH 环境变量，需要重新安装一次，或者在网上搜索失败原因并修改 PATH 环境变量。对于初学者来说，卸载后重新安装，并在重新安装的过程中特别注意勾选图 1-11 中的 Add Python 3.9 to PATH 选项，解决问题的速度更快。

## 1.2.2 PyCharm 的安装

PyCharm 是一个比较好用的 Python 语言编程集成开发环境，包括 Professional(收费版)和 Community(免费版)两个版本。对于非计算机专业的学生，Community 版本足够使用。如果希望学习 Python 的进阶功能，最好使用 Professional 版本。本书以 Community 版本为例，介绍 PyCharm 集成开发环境，全书均使用 Community 版本的 PyCharm 集成开发环境。

PyCharm 集成开发环境的官网地址是 <https://www.jetbrains.com/pycharm/>，可以按照以下步骤进行下载和安装操作。

- (1) 进入 PyCharm 的官方网页 <https://www.jetbrains.com/pycharm/>，如图 1-17 所示。

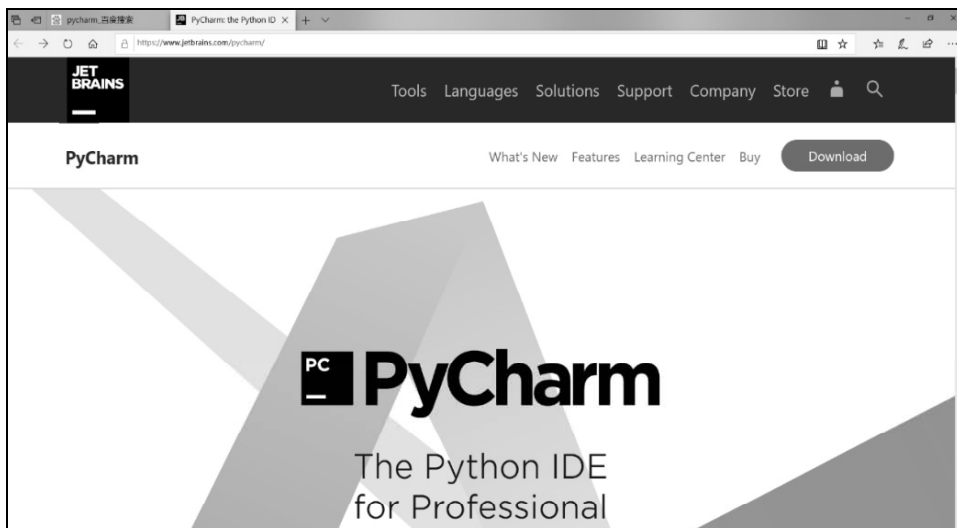


图 1-17 PyCharm 的官方网站

(2) 单击 Download 按钮进入下载页面，如图 1-18 所示。此处选择 Community 版本，单击 Download 按钮下载。

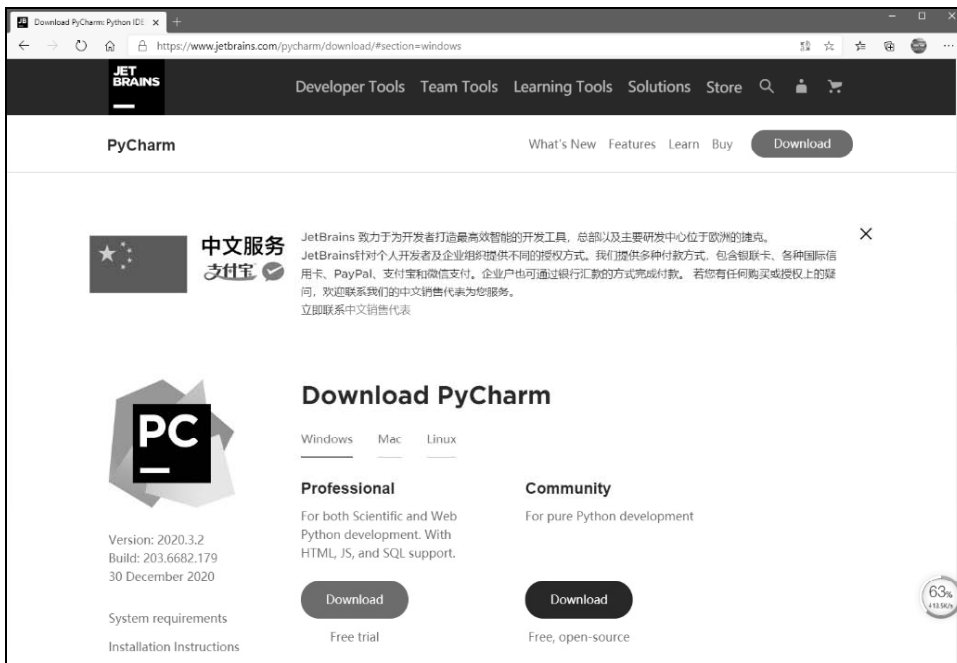


图 1-18 单击 Download 按钮下载

这时页面会跳转，不用做什么操作，等待下载即可，如图 1-19 所示。

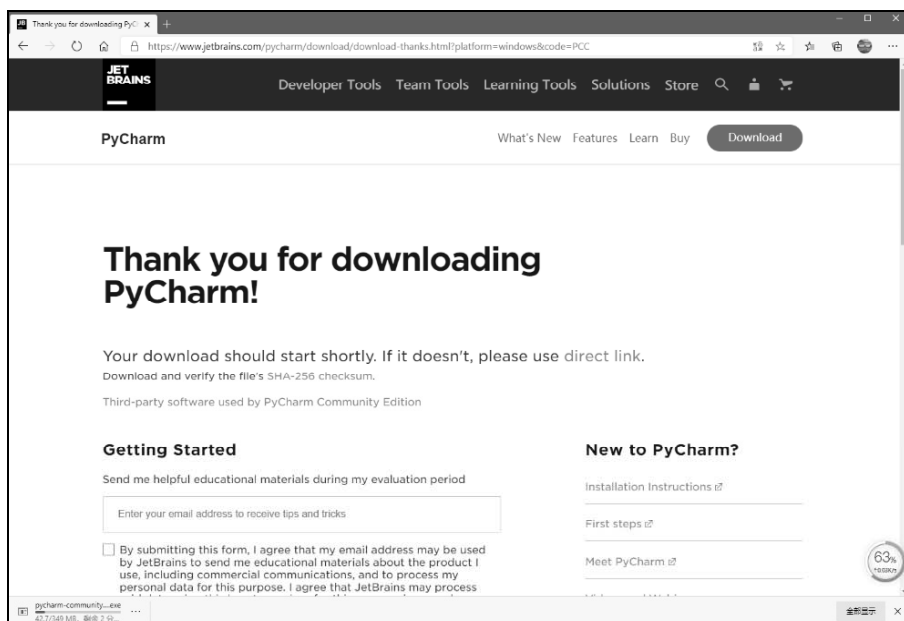


图 1-19 等待下载

(3) 下载完成后, 就可以开始安装工作了。双击已经下载好的 PyCharm 安装文件进行安装。需要指出的是, Professional 和 Community 两个版本均不支持 32 位系统的安装, 如果读者的计算机是 Windows 7 的 32 位系统, 而不是 Windows 10 的 64 位系统, 则需要寻找其他版本的安装文件。

开始安装的时候, 直接单击“运行”按钮, 就可以开始安装 PyCharm 了, 如图 1-20 所示。

(4) 单击 Next 按钮, 在安装过程中可以自定义安装路径, 也可以使用默认安装路径, 如图 1-21 所示。

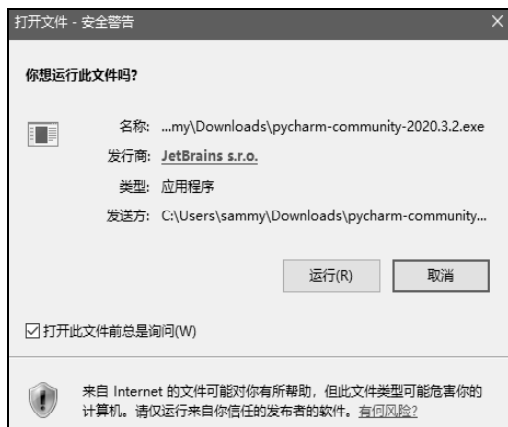


图 1-20 运行 PyCharm 安装文件

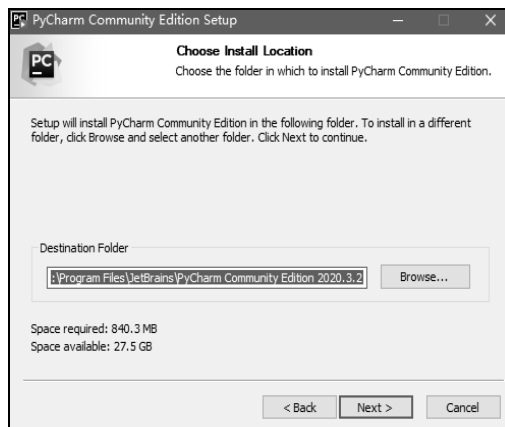


图 1-21 PyCharm 的安装路径

在对话框中单击 Next 按钮, 选择安装选项时, 建议选中所有的可选项, 如图 1-22 所示。单击 Next 按钮, 然后单击 Install 按钮, 开始安装操作, 如图 1-23 所示。

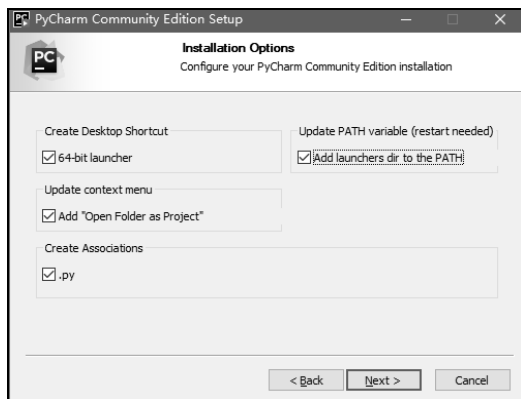


图 1-22 PyCharm 的安装选项

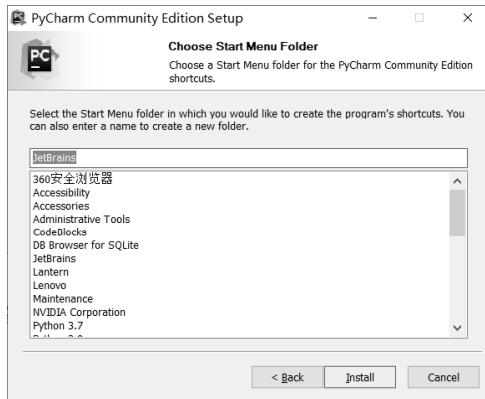


图 1-23 开始安装 PyCharm

(5) 安装完成后，显示如图 1-24 所示界面，单击 Finish 按钮后重启计算机，桌面上会出现 PyCharm Community 快捷方式。

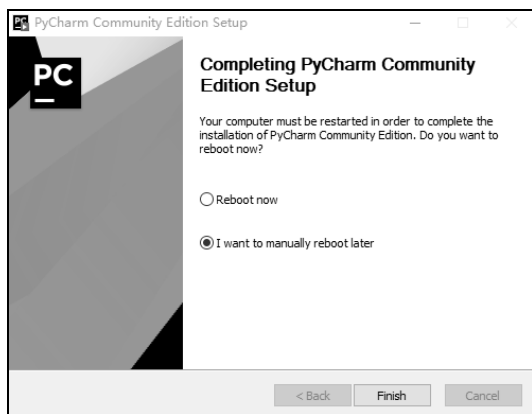


图 1-24 完成 PyCharm 的安装

## 1.3 HelloWorld 程序

双击桌面上的 PyCharm 图标，在 PyCharm 中进行 Python 编程。本节将介绍如何生成简单的 Python 程序。

(1) 第一次运行 PyCharm 会出现图 1-25 所示对话框，可以根据个人喜好进行设置。

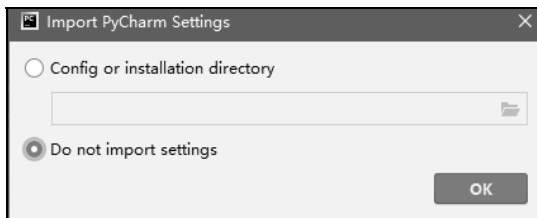


图 1-25 PyCharm 的个人喜好设置

(2) 在 PyCharm 窗口中，选择 Customize 命令，调整主题色调，本书以 IntelliJ Light 主题色调为例进行讲解，如图 1-26 所示。

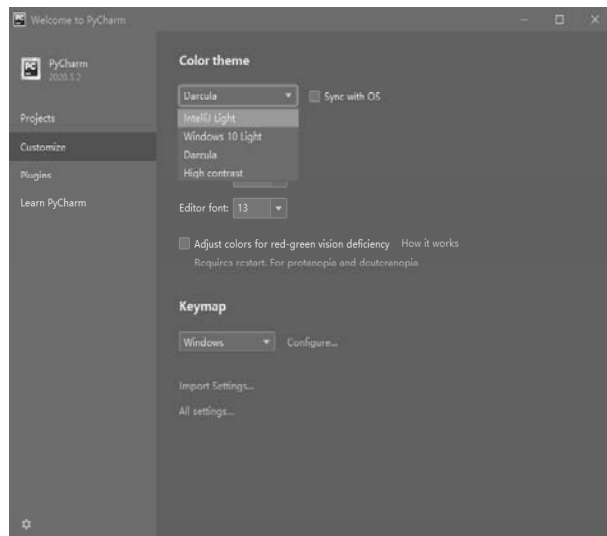


图 1-26 设置主题色调

(3) 选择 Project 选项，在欢迎窗口中单击 New Project 选项按钮，创建新项目，如图 1-27 所示。

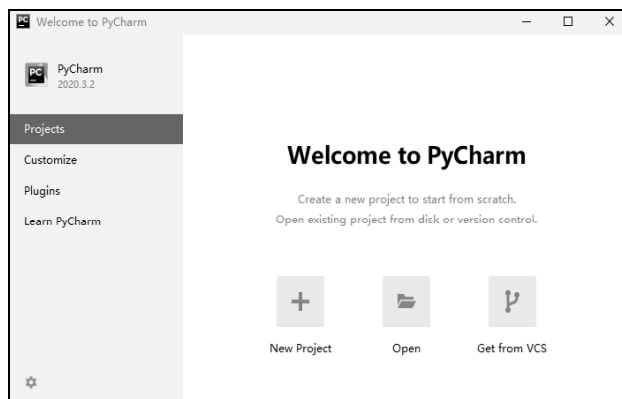


图 1-27 PyCharm 的欢迎窗口

(4) 在 New Projects 对话框的 Location 文本框中设定项目所在文件夹及项目名称，如图 1-28 所示。

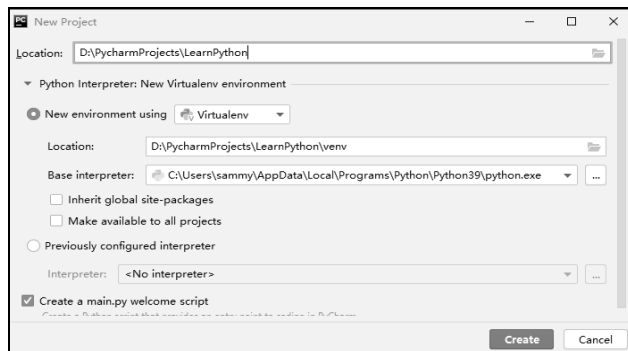


图 1-28 设定项目所在文件夹及项目名称

(5) 单击 **Create** 按钮，这时会跳出每日提示，可以直接单击 **Close** 按钮关闭每日提示，如图 1-29 所示。

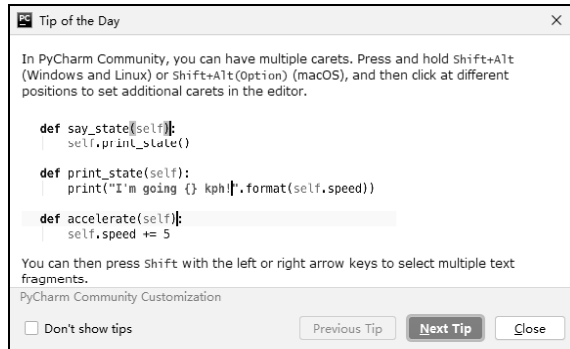


图 1-29 关闭每日提示

(6) 这样就创建了第一个 Python 项目，新建项目所在的位置如图 1-30 所示。



图 1-30 新建项目所在的位置

(7) 第一个项目建好之后，在项目名称上右击，在弹出菜单中选择 **New** → **Python File** 命令创建第一个 Python 文件，如图 1-31 所示。

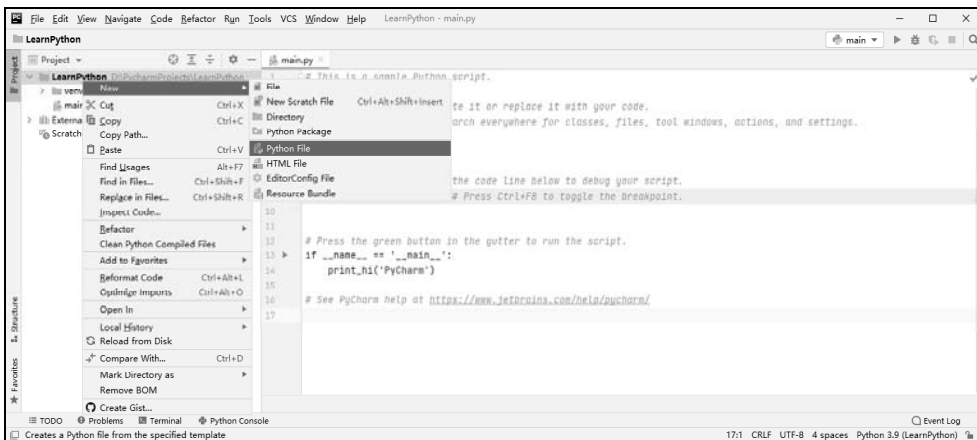


图 1-31 创建第一个 Python 文件

(8) 将这个 Python 文件命名为 HelloWorld，双击 Python file，即可创建 Python 文件，如图 1-32 所示。

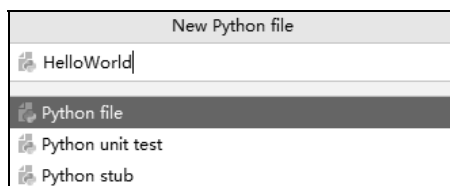


图 1-32 创建名为 HelloWorld 的 Python 文件

(9) 打开刚才创建的名为 HelloWorld.py 的 Python 源文件，右侧空白区域为文件编辑区，可以在该区域输入 Python 命令，如图 1-33 所示。

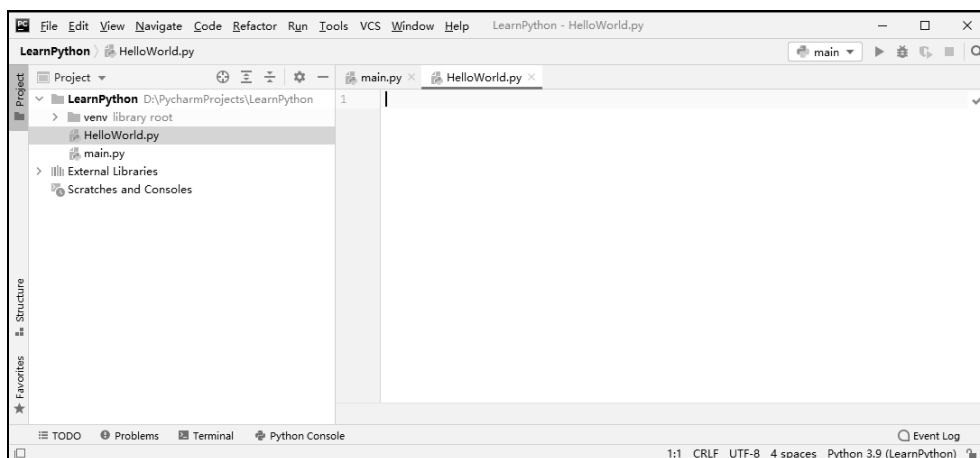


图 1-33 打开 HelloWorld.py 文件

**【例题 1.1】** 编辑 HelloWorld.py 源文件，输出 Hello World。

例题 1.1 源代码如下：

```
print('HelloWorld')
```

在文件编辑区输入源程序代码 `print('HelloWorld')`，如图 1-34 所示。

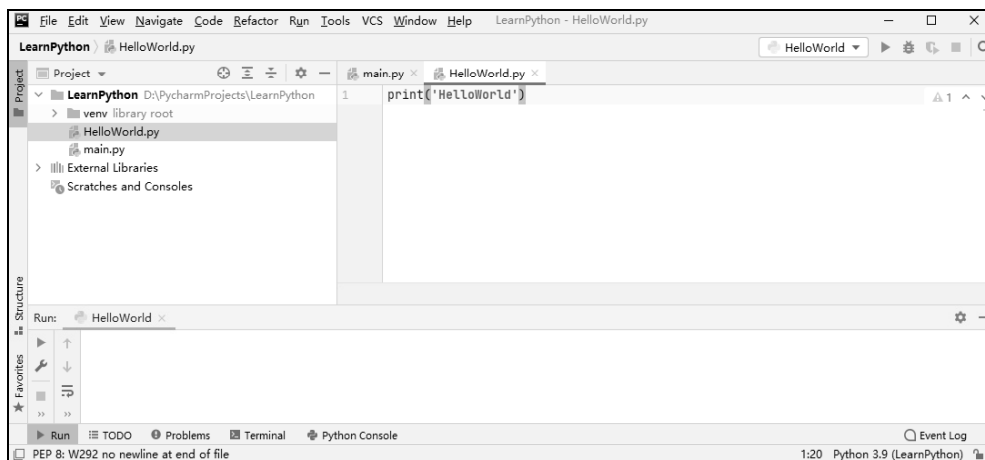


图 1-34 将源代码输入文件编辑区



在菜单栏中选择 Run → Run 命令，运行程序，如图 1-35 所示。

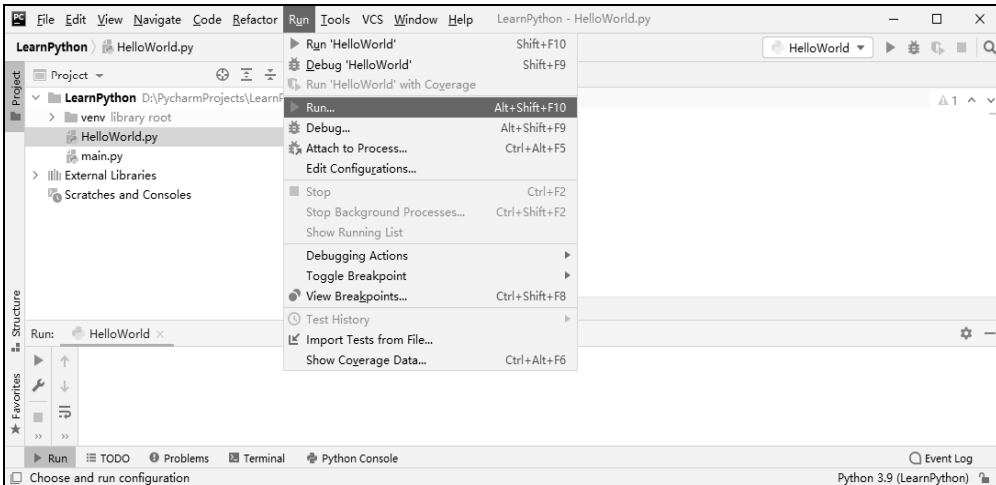


图 1-35 选择 Run→Run 命令

成功运行之后，就会在控制台显示 HelloWorld 的运行结果，如图 1-36 所示。

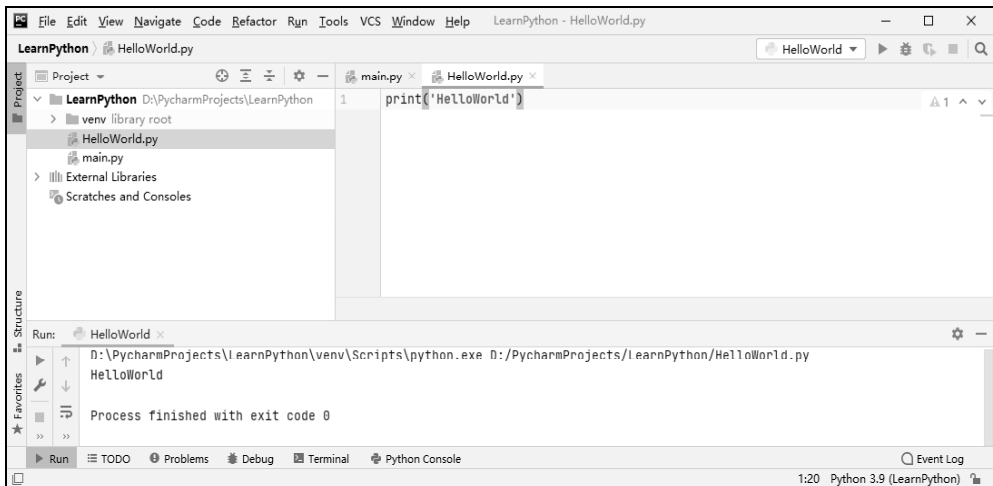


图 1-36 HelloWorld 运行结果

**【代码分析】**这是一个很简单的语句。`print()`是一个函数命令，命令中单引号('...')之间的内容：HelloWorld，会被直接输出显示在控制台上。下面，我们稍微扩展一下这个程序。

**【例题 1.2】**生成一个 SayHi.py 文件，将这个文件放到 Chapter01 文件夹中，编辑源文件，使得程序在控制台的第一行显示 Hi，第二行显示 Good Morning，第三行显示 Nice to meet you。

实现方法如下。

(1) 如图 1-37 所示，在项目名上右击，弹出菜单后，选择 New → Directory 命令，从此处以后，将使用文件夹 Directory 对整个项目的源文件进行管理。





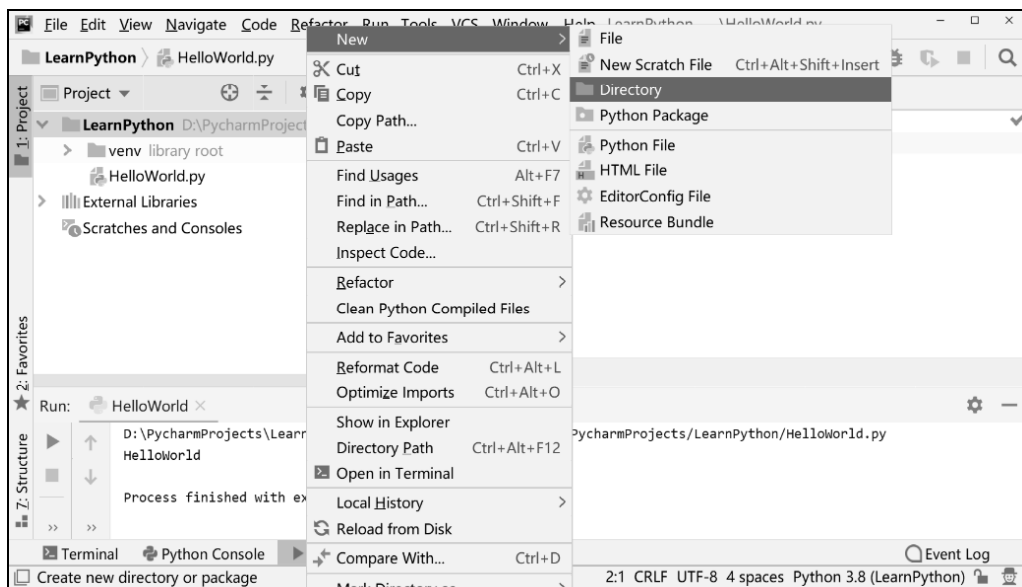


图 1-37 选择 New → Directory 命令

(2) 在图 1-38 所示对话框中输入文件夹名: Chapter01, 然后按 Enter 键, PyCharm 界面上会出现 Chapter01 文件夹, 如图 1-39 所示。可能大家会注意到背景, PyCharm 的工作区域显示的是上一次打开 PyCharm 后最后一次编译过的文件。



图 1-38 命名目录

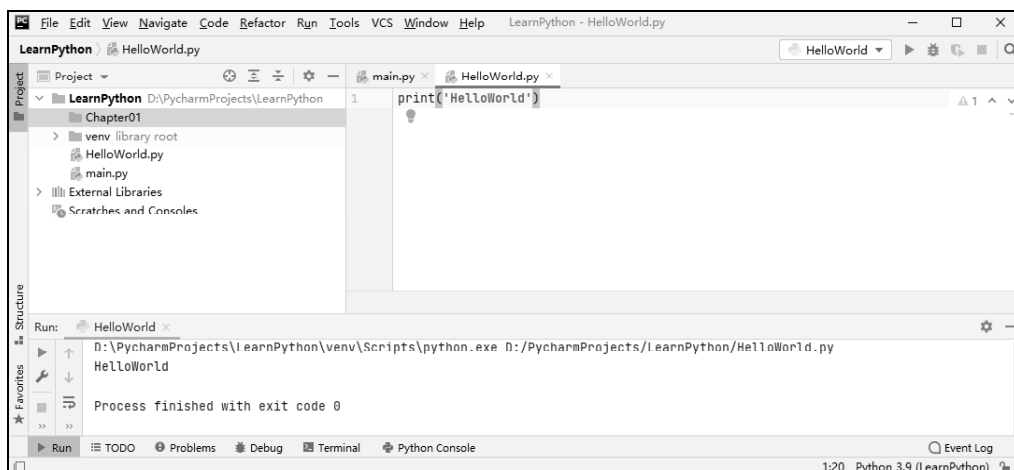


图 1-39 创建 Chapter01 文件夹

(3) 在 Chapter01 文件夹上右击, 弹出菜单后, 选择 New → Python File 命令, 创建 Python 源文件。在对话框中输入文件名: SayHi, 双击 Python file, 即可创立 Python 文件 SayHi, 如图 1-40 所示。

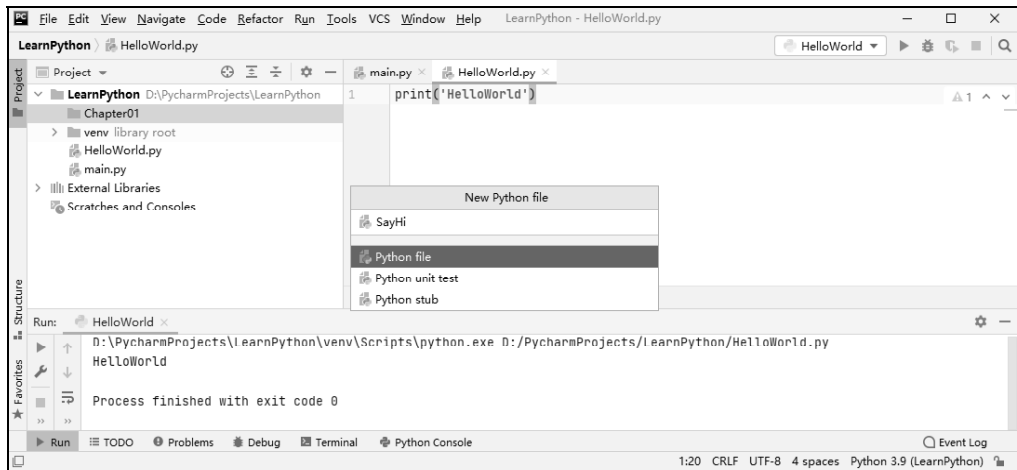


图 1-40 在 Chapter01 文件夹中创建名为 SayHi 的 Python 文件

如图 1-41 所示，Chapter01 目录中出现了一个名为 SayHi.py 的 Python 源文件，可以在文件编辑区对此文件进行编辑。

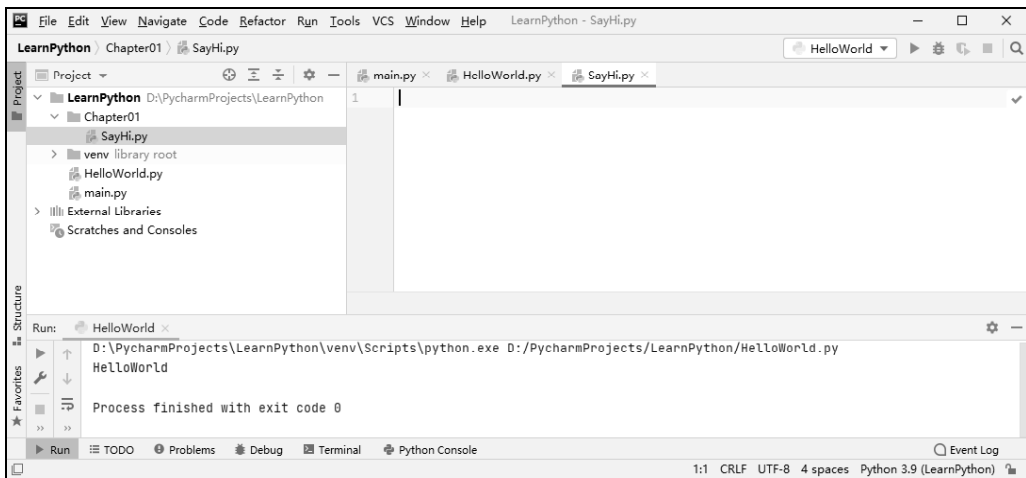


图 1-41 SayHi.py 被创建出来

例题 1.2 要求输出三句话，因此需要使用三次 `print()` 命令。

例题 1.2 源代码如下：

```

1. #【例题 1.2】源代码
2. print('Hi')
3. print('Good morning')
4. print('Nice to meet you')
5. ""
6. 以上为 1.3 中的【例题 1.2】的解答
7. ""

```

(4) 如图 1-42 所示，在 PyCharm 的文件编辑区输入 Python 源代码。

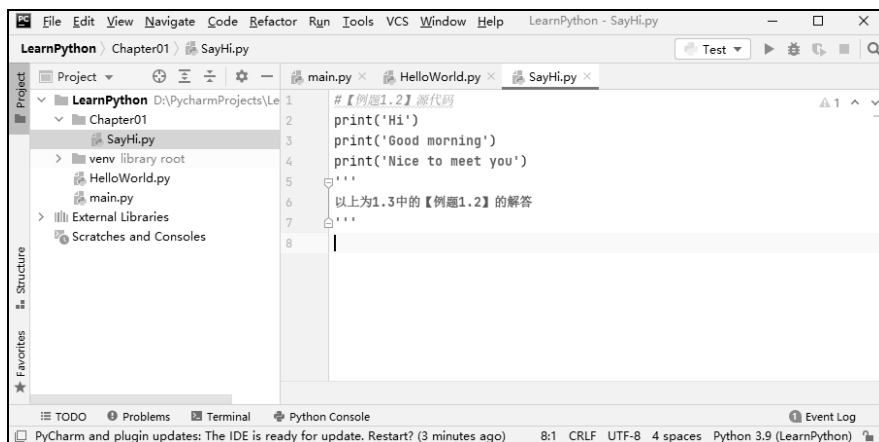


图 1-42 输入例题 1.2 源代码

(5) 选择 Run → Run 命令运行 SayHi.py，如图 1-43 所示。

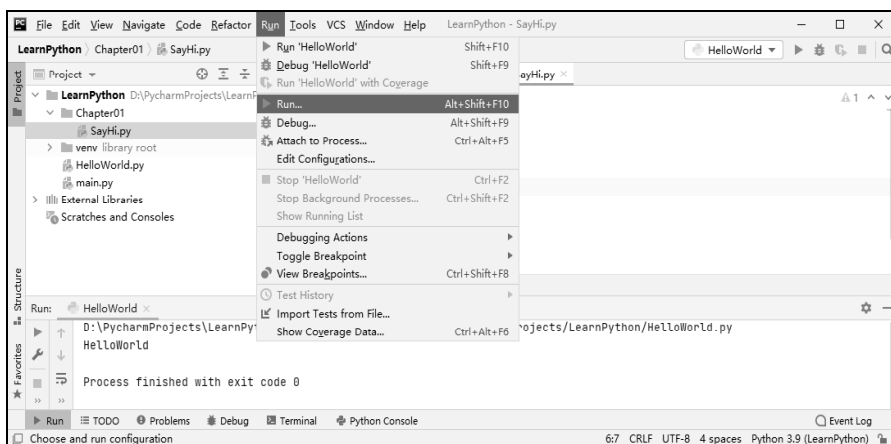


图 1-43 选择 Run → Run 命令

(6) 如图 1-44 所示，在 Run 对话框中选择 SayHi 文件。

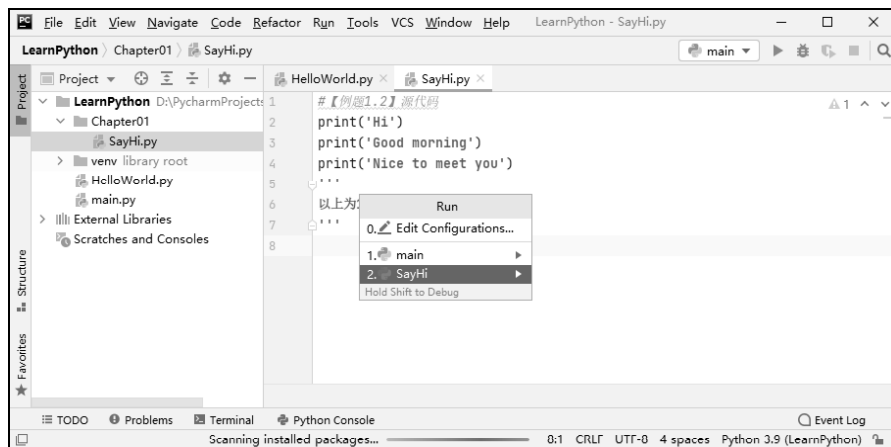


图 1-44 运行 SayHi 文件

控制台上显示例题 1.2 中要求依次输出的三句话，如图 1-45 所示。

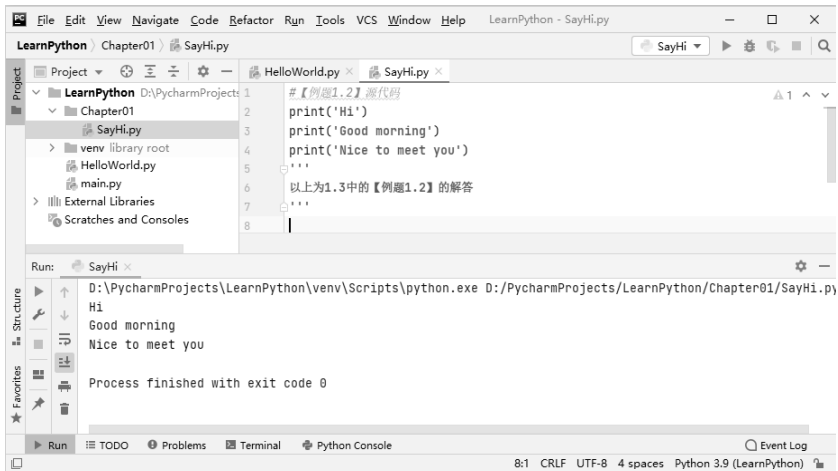


图 1-45 SayHi 运行成功

**【代码分析】**大家可能注意到了，单引号('...')之间的内容被显示在控制台上。在 Python 中，使用一对单引号('...')或者一对双引号("...")都可以标识出一个字符串。字符串在后续章节中会有介绍。

**【例题 1.3】**生成一个名为 TestArea.py 的文件，计算半径为 5 的圆的面积是多少？

例题 1.3 源代码如下：

1. """
2. **【例题 1.3】**
3. 计算半径为 5 的圆的面积
4. """
5. print('圆的面积是', 5 \* 5 \* 3.14 =, 5 \* 5 \* 3.14)



在 Chapter01 目录中新建一个名为 TestArea 的源文件，在文件编辑区输入源代码，运行后的结果如图 1-46 所示。

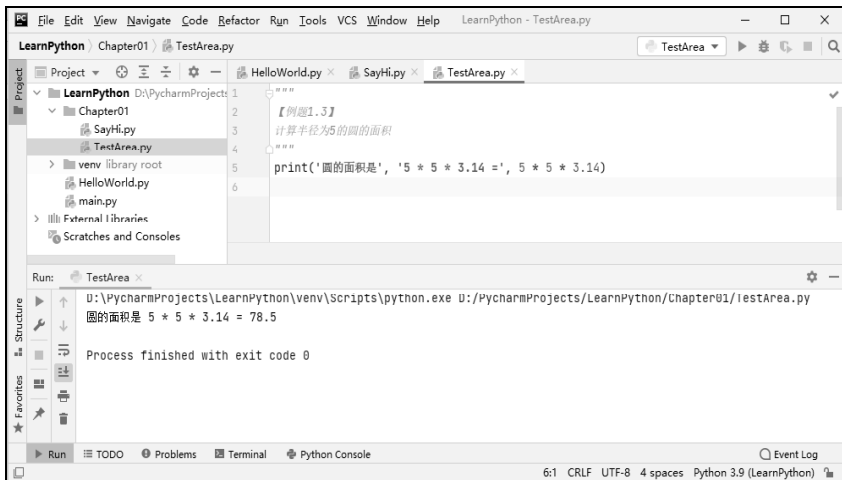


图 1-46 TestArea.py 源文件运行成功

**【代码分析】**由例题 1.1~1.3 可以得知，`print()`方法可以用来显示字符串或者算式的结果。既需要显示字符串，又需要显示算式结果，即需要将多个内容一起显示的时候，可以用英文逗号(,)间隔字符串与算式。一对单引号或者一对双引号之间的内容都可以被称为字符串。

截至目前，书中已经讲解了 3 个例题，如果同学们未运行成功，请观察字母的大小写是否一致，以及所使用的符号是英文符号还是中文符号。Python 编程语言中所使用的符号应为英文符号。无法运行程序的原因很可能是输入的命令中字母的大小写不一致，或括号、引号、逗号等符号为中文符号。

## 1.4 程序注释

程序的注释非常重要。如果大家编写程序的时候遵守统一的注释原则，编写的程序就比较容易被别人理解。在编写程序之前应写上若干行注释，以帮助读者理解代码。

注释是为帮助写代码或者读代码的人理解代码，而由程序员人为添加的，不会被程序编译器翻译或编译成机器语言去执行的字符串。只要是注释，就不会被程序执行。

本书中用到的 Python 注释有两种。

### 1.4.1 单行注释

例题 1.2 源代码中的第 1 行，如图 1-47 所示，被称为单行注释。单行注释的标志符号是#。#之后的同一行中的任意内容的字符串会被翻译器忽略，成为单行的注释内容。



图 1-47 单行注释

### 1.4.2 多行注释

例题 1.2 源代码中的第 5~7 行，如图 1-48 所示，例题 1.3 源代码中的第 1~4 行，如图 1-49 所示，均被称为多行注释。

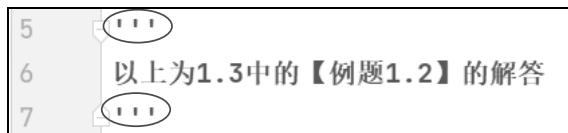


图 1-48 例题 1.2 源代码中的多行注释

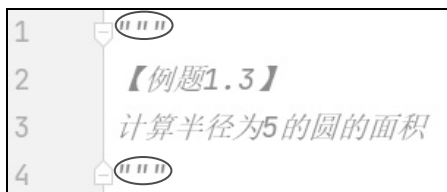


图 1-49 例题 1.3 源代码中的多行注释

多行注释的标志符号为三个单引号对或者三个双引号对。三个单引号对或三个双引号对之间的任意多行内容都会被 Python 的翻译器认为是多行注释，而不去翻译。

在后续章节的学习中，会在源代码中添加一些注释帮助大家理解代码。

## 1.5 课后习题

### 一、选择题

- (1) 下列选项中, 不属于 Python 特点的是( )。
- A. 面向对象                      B. 丰富的库                      C. 运行效率高                      D. 免费和开源
- (2) Python 内置集成开发工具是( )。
- A. PythonWin                      B. Pydev                      C. IDE                      D. IDLE
- (3) 用户程序编写的 Python 程序(避免使用依赖于系统的特性), 无须修改就可以在任何支持 Python 的平台上运行, 这体现了 Python 语言的( )。
- A. 可嵌入性                      B. 可移植性                      C. 可扩展性                      D. 简单
- (4) Python 程序文件的扩展名为( )。
- A. .py                      B. .pt                      C. .python                      D. .cpp
- (5) 下列关于 Python 语言的说法中, 错误的是( )。
- A. Python 是跨平台的  
B. Python 是解释型语言  
C. Python 语言出现较晚, 因而其具有其他高级语言的一切优点  
D. Python 可用于数据抓取(爬虫)
- (6) Python 语言属于( )。
- A. 面向对象的高级语言                      B. 面向过程的高级语言  
C. 机器语言                      D. 汇编语言
- (7) 使用 print 函数将字符串 Hello World 输出到终端, 并要求将这两个单词分行输出, 以下语句正确的是( )。
- A. `print("Hello/nWorld")`                      B. `print("Hello World")`  
C. `print("Hello\nWorld")`                      D. `print("Hello")\nprint ("World")`
- (8) 以下语句中, ( )不是 Python 的注释代码。
- A. `# Python 语言注释`                      B. `// Python 语言注释`  
C. `""" Python 文档"""`                      D. `''' Python 文档'''`
- (9) 以下符号中, ( )不是 Python 的注释符号。
- A. `#`                      B. `"""          """`  
C. `/**          **/`                      D. `'''          '''`
- (10) Python 是一种通用语言, 以下不是使用 Python 的知名机构与场景的是( )。
- A. NASA 使用 Python 进行数据分析和运算  
B. 豆瓣网所有的业务都是 Python 开发的  
C. 百度贴吧使用 Python 作为数据库  
D. 雅虎使用 Python 完成了群组交流平台

- (11) 以下关于 Python 中 `print()`函数的说法中, 错误的是( )。
- A. `print()`函数用于将输出内容输出到终端
  - B. `print()`函数可以用来打印字符串或数值
  - C. `print()`函数可以用于直接打印算式结果
  - D. `print()`函数默认不换行
- (12) 以下有关 Python 注释语句的叙述中, 不正确的是( )。
- A. Python 的注释语句可以出现在任何位置, 以`#`开始, 到行末结束
  - B. Python 的解释器会忽略所有注释语句, 所以注释语句是不会执行的
  - C. 符号`//`可用于 Python 的相关语句注释
  - D. Python 注释语句中可使用中文注释
- (13) 语句 `print("""hello, world! """)`的输出结果为( )。
- A. "hello, world! "
  - B. hello, world!
  - C. hello, world
  - D. hello,world! """"
- (14) 以下关于 Python 语句书写规则的叙述中, 错误的是( )。
- A. 一般情况下, 一行一条语句
  - B. 斜杠(`\`)可用于一个代码跨越多行的情况, 即如果一行语句过长, 想要分为多行, 可使用`\`来作为续行标志
  - C. Python 语句前面可以有任意空格, 不会引起误会
  - D. 英文分号(`;`)可以用于分隔在同一行上书写的多条语句
- (15) 以下有关程序语言的描述中, 不正确的是( )。
- A. Python 是一种脚本语言
  - B. 机器语言是直接操作计算机硬件的语言
  - C. 计算机语言经历了机器语言、汇编语言、脚本语言
  - D. Python 是翻译性语言, 需要逐行对程序进行翻译
- (16) 以下对 Python 程序设计风格的描述中, 错误的选项是( )。
- A. Python 中不允许把多条语句写在同一行
  - B. Python 语句中, 使用缩进表示语句块的所属作用域范围
  - C. Python 可以将一条长语句分成多行显示, 使用斜杠(`\`)分行
  - D. Python 中允许使用英文分号(`;`)分隔不同语句
- (17) Python 的函数不包括( )。
- A. 标准函数
  - B. 第三方函数库
  - C. 内建函数
  - D. 参数函数
- (18) 以下对 Python 程序的缩进格式描述错误的是( )。
- A. 不需要缩进的代码顶格写, 前面不能留空白
  - B. 缩进可以用 Tab 键实现, 也可以用多个空格实现
  - C. 严格的缩进可以约束程序结构, 可以多层缩进
  - D. 缩进是用来美化 Python 程序的格式, 没有什么特殊含义
- (19) 用于将 Python 程序打包成可执行文件的工具是( )。
- A. Panda3D
  - B. pygame
  - C. pip
  - D. PyInstaller
- (20) 关于 Python 语言的特点, 以下选项描述正确的是( )。
- A. Python 语言不支持面向对象, 它是面向过程的编程语言

- B. Python 语言是解释型语言  
 C. Python 语言是编译型语言  
 D. Python 语言是非跨平台语言
- (21) Python 源程序执行的方式为( )。  
 A. 编译执行                      B. 解释执行                      C. 直接执行                      D. 边翻译边执行
- (22) 在 Python 中, 由于字符串是一串连续的字符序列, 若想用 print()函数方法换行输出一串较长的字符串, 以下使用的方法不能达到要求的是( )。  
 A. 使用换行转义字符  
 B. 使用多个 print()函数分别输出  
 C. 使用空格来填充剩余长度, 使其分行  
 D. 在 print()函数中将其分行书写并用符号"" 引用
- (23) 以下数据类型中, 不被 Python 支持的是( )。  
 A. 字符串                      B. 浮点型数值                      C. 整型数值                      D. char 字符数值
- (24) 以下有关 Python 的叙述正确的是( )。  
 A. Python 是解释语言                      B. Python 是汇编语言  
 C. Python 是非开源的                      D. Python 是面向过程的编程语言
- (25) 在 Python 3 版本中, 语句 print "hello+world!"的输出结果为( )。  
 A. 语法错误                      B. hello+world!                      C. hello world!                      D. helloworld!
- (26) 下列代码正确的运行结果为( )。

```
a = "hello"
# a = "hi"
print(a, ", world")
```

- A. hi, world                      B. hello , world                      C. hello                      D. 语法出错
- (27) 以下程序的输出结果为( )。

```
print("这个圆的面积为: ", 3.14 * 2 * 2)
```

- A. 12.56                      B. 这个圆的面积为: , 12.56  
 C. 这个圆的面积为: 12.56                      D. 语法出错, 无法计算
- (28) Python 的设计具有很强的可读性, 以下关于 Python 具有的特色语法的描述中, 不正确的是( )。  
 A. 交互式                      B. 解释型                      C. 面向对象                      D. 面向服务
- (29) 安装好 Python 之后, 有 3 种方式可以对 Python 源程序进行编辑、运行, 下列描述中, 无法运行 Python 源文件的方式是( )。  
 A. 浏览器中运行                      B. 交互式解释器                      C. 命令行                      D. PyCharm
- (30) 以下关于 Python 的叙述中, 正确的是( )。  
 A. Python 程序的每一行都只能编写一条语句  
 B. 同一层次的 Python 语句必须对齐  
 C. 在编写 Python 程序的过程中, 可发现注释语句中的拼写错误  
 D. Python 程序可以从任意位置开始编写



- (31) 以下解释器中, 不属于 Python 解释器的是( )。
- A. CPython                      B. PyPy                      C. JPython                      D. PyCharm
- (32) Python 解释器的作用为( )。
- A. 运行文件                      B. 编写程序                      C. 编译文件                      D. A、B、C
- (33) 以下不属于 PyCharm 的内部集成功能的是( )。
- A. Project 管理                      B. 调试代码                      C. 解释代码                      D. 管理文件
- (34) Python 语言中用来标识语句块的标记是( )。
- A. 分号                      B. 逗号                      C. 缩进                      D. /
- (35) `print(-5/4)`的结果为( )。
- A. -1.25                      B. -1                      C. -2                      D. 0
- (36) 以下关于 Python 的描述中, 正确的为( )。
- A. Python 是面向对象的编程语言, 其不支持面向过程的编程
- B. Python 语言被广泛用于各行各业, 是当今最流行的语言之一
- C. Python 是一种非开源的脚本语言
- D. 我国学习、使用 Python 语言的人群较多, 同时也拥有众多优秀的原创学习资料及书籍供学习者查阅

## 二、判断题

- (1) Python 中, 一行可以书写多条语句, 每条语句以英文分号(;)分隔即可。 ( )
- (2) Python 程序编写方式灵活, 如果语句过长, 可以直接分成多行书写。 ( )
- (3) Python 中, 多行注释的符号为英文的三个单引号或双引号。 ( )
- (4) 在 Windows 平台上编写的 Python 程序无法在 Unix 平台运行。 ( )
- (5) 不可以同一台计算机上安装多个 Python 版本。 ( )
- (6) Python 变量使用前必须先声明, 一旦声明就不能再改变其类型了。 ( )
- (7) Python 采用的是基于值的自动内存管理方式。 ( )
- (8) 在 Python 3 中, 不可以使用中文作为变量名。 ( )
- (9) Python 使用缩进来体现代码之间的层次逻辑关系。 ( )
- (10) Python 代码的注释只有一种方式, 即使用符号#。 ( )
- (11) 在 Python 3 中, 语句 `print('abcde')` 不能正确执行。 ( )
- (12) 使用 `print()`函数无法将信息写入文件。 ( )
- (13) Python 中, 一切变量都可以称为对象。 ( )
- (14) 在 IDLE 交互模式下, 下画线(\_)表示解释器中最后一次显示的内容或最后一条语句正确执行的输出结果。 ( )
- (15) Python 程序只能在安装了 Python 环境的计算机上以源代码形式运行。 ( )
- (16) Python 使用符号 // 作为注释符号。 ( )
- (17) 在 Python 3 中, 对多个变量进行赋值时, 变量之间需要使用逗号隔开。 ( )
- (18) Python 是一种面向过程的高级程序语言。 ( )
- (19) Python 是一种面向对象的解释型语言。 ( )
- (20) Python 由荷兰人 Guido Van Rossum 于 1991 年正式发布可用版本。 ( )

- (21) Python 有着丰富的扩展库，可以轻易完成各种高级任务，开发者可以用 Python 实现一个完整应用程序所需的大多数功能。 ( )
- (22) 在 Python 中使用变量时也需要像其他语言一样先定义变量。 ( )
- (23) Python 既可以支持多进程，又可以支持多线程。 ( )
- (24) Python 程序文件的扩展名主要有 .py 和 .pyw 两种，而 Python 源代码程序在翻译之后，文件扩展名为 .pyc。 ( )
- (25) Python 是一门交互式语言，即其可在终端进行命令行交互编程。 ( )
- (26) Python 具有易于维护和学习、广泛的标准库(提供了大量的工具)、可扩展性、语法简单等特点，因而其没有什么缺点。 ( )
- (27) Python 程序运行时，先翻译字节码并保存到内存中，当程序运行结束后，Python 解释器将内存中的字节码对象写到 .pyc 文件中。 ( )
- (28) 安装完 Python 运行环境后，不需要再安装集成开发环境，就可以编写复杂的 Python 程序。 ( )
- (29) Python 是一种脚本语言，工作效率优于 C 或 C++ 语言。 ( )
- (30) Python 中没有编译的过程，它是通过解释器直接执行程序的。 ( )
- (31) 由于注释语句用来解释所编写语句的功能、作用、结果、注意点之类的内容，所以注释语句越多越好。 ( )
- (32) Python 支持命令式编程、函数式编程，完全支持面向对象程序设计，拥有大量的第三方扩展库。 ( )
- (33) Python 有多种运行方式，运行代码少的程序时多使用交互式命令符，而随着代码的增多会采用脚本的方式。 ( )
- (34) Python 被广泛应用于各个领域，其中最主要的应用领域有人工智能、云计算、爬虫、数据库管理等。 ( )
- (35) Python 语言和 C、C++、Java 都是静态类型语言。 ( )
- (36) Python 2 和 Python 3 版本中的 print() 函数使用方法一样，没有任何区别。 ( )
- (37) Python 3 中，print() 函数可用 “%” 对输出的内容进行格式化。 ( )
- (38) Python 有四大主要应用：一是网络爬虫，二是 Web 开发，三是人工智能设计，四是自动化运维。 ( )
- (39) Python 应用范围十分广阔，所以其相关方面都有着大量较实用的原创中文资料以供想学习 Python 的人查阅。 ( )
- (40) Python 自发布以来，主要有三个版本：1990 年发布的 Python 版本、2000 年发布的 Python 2.0 版本和 2008 年发布的 Python 3.0 版本。 ( )
- (41) Python 是一门独立的编程语言，并不是由其他语言发展而来。 ( )
- (42) 在 Python 3 中，用 # 表示注释开始，多行注释在第一行开始处使用 # 即可。 ( )
- (43) PyCharm 是一种 Python IDE(集成开发环境)。 ( )
- (44) Python 注释语句的书写方式只有两类，分别是单行注释和多行注释。 ( )
- (45) 在 Python 中，print() 函数不仅可以输出字符串，还可以输出表达式的值。 ( )
- (46) 在 Python 中，print() 函数默认是换行的，要想不换行应该在参数列表中加入 end 参数，如 print(i, end = "")。 ( )

- (47) `print()`函数不能将数据格式化输出。 ( )
- (48) `print()`函数可以输出任何一种类型的数据。 ( )
- (49) Python 3 和 Python 2 中 `print` 函数的用法有很多不同, Python 3 中需要使用括号。 ( )
- (50) Python 中存在丰富的函数库, 即具有其他语言中所有的函数库。 ( )
- (51) Python 语句中一般以换行作为一条语句的结束标志, 但是可以使用斜杠(`\`)将一条语句分为多行显示。 ( )
- (52) Python 采用交互式编程方式编程, 因而不需要创建脚本文件, 可以通过 Python 解释器的交互模式进行代码编写。 ( )
- (53) 空行与代码缩进相同, 空行也是 Python 语法的一部分。书写时不插入空行, Python 解释器运行也会出现相应的错误。 ( )
- (54) Python 可以在同一行中使用多条语句, 但是语句之间要使用分号(`;`)分割。 ( )
- (55) 面向对象的语言具有封装性、继承的、多态的特点, 而 Python 也是面向对象的语言, 故而其也具有这三个特点。 ( )
- (56) 缩进相同的一组语句构成一个代码块, 语句结束必须加分号。 ( )
- (57) Python 语言中使用的运算符可以是中文字符, 也可以是英文字符。 ( )
- (58) 在 Python 语句中, 只要是注释语句, 就一定不会被程序所执行。 ( )
- (59) 可以在同一台计算机上安装多个 Python 版本。 ( )
- (60) Python 语言虽然应用广泛, 但其还未用于开发游戏。 ( )
- (61) 由于 Python 语言的源代码是可以获取的, 所以其是一种开源语言, 但这就造成了其保密性较差的缺点。 ( )

### 三、编程题

(1) 使用 Python 中的 `print()`函数输出两行中文文字, 并注意切换中英文输入法。

第一行: 你好! 中国!

第二行: 我爱我的祖国!

程序运行后期望结果如下所示:

```
你好! 中国!
我爱我的祖国!
```

(2) 已知圆周率为 3.14, 用 `print()`函数输出一个半径为 4 的圆的面积以及周长, 并要求对输出数据有相应的文字介绍。

程序运行后期望结果如下所示:

```
圆的周长是: 25.12
圆的面积是: 50.24
```

(3) 用 `print()`函数输出  $5!$ 的运行结果。(提示:  $5! = 5 * 4 * 3 * 2 * 1$ 。)

程序运行后期望结果如下所示:

```
5! = 120
```

(4) 在控制台输出 3 行 “nice to meet you!”。

程序运行后期望结果如下所示：

```
nice to meet you!  
nice to meet you!  
nice to meet you!
```

(5) 计算一个长为 15.21 米，宽为 11.79 米的矩形房间的面积。

程序运行后期望结果如下所示：

```
房间面积是： 179.3259
```

(6) 将以下代码输入 PyCharm 集成开发环境，创建名为 Ch01Exe06.py 的 Python 程序，运行后观察结果。

```
1. # 在《西游记》中，乱石山碧波潭的两只鱼精。  
2. print("奔波儿灞与灞波儿奔")  
3. # 在《水浒传》中，东京欺行霸市的街头泼皮。  
4. print("牛二")  
5. # 在《三国演义》中，太守韩馥的无双上将。  
6. print("潘凤")  
7. # 《红楼梦》中，只知道姓，不知道名的主要人物。  
8. print("贾母")
```