

# 项目 1 项目开发环境搭建

## 学习目标

- 了解 Python 的诞生与发展、特点和应用领域。
- 理解并掌握在 Windows 操作系统中 Python 开发环境的搭建。
- 理解 Python 程序简单示例。

## 育人目标

- 培养学生努力发扬精益求精的工匠精神。
- 培养学生遵循语法规则的习惯，树立牢固的规则意识。
- 培养学生具有正确认识自己，发挥特长，补齐短板。

## 1.1 项目引导

随着人工智能、大数据时代的到来，Python 已经成为当今最受欢迎的通用编程语言之一。Python 在云计算、Web 开发、科学计算、人工智能、系统操作和维护、金融工程等领域都发挥着重要作用。Python 具有简洁、易于学习、易于阅读、可移植和资源丰富等优点，非常适合作为编程初学者的入门语言。本项目要求熟练掌握 Python 开发环境的搭建。项目首先介绍 Python 的诞生与发展、特点、应用领域，接着介绍如何搭建 Python 开发环境，然后介绍 Python 第三方开发工具，最后要求访问官网下载基于 Windows 平台的 Python 安装包，安装并配置 Python 环境，用不同方式编程并运行 Python 程序。

## 1.2 技术准备



Python 的诞生与发展

### 1.2.1 Python 的诞生与发展

人与计算机交互所使用的语言称为计算机程序设计语言。计算机程序设计语言按其发展的过程是机器语言→汇编语言→高级语言。机器语言是计算机唯一能够识别并能直接执行的语言，实际上机器语言就是计算机中二进制代码指令的集合，用机器语言编写程序十分烦琐，工作量大，且写出的程序可靠性差，除了计算机生产厂家的专业人员外，绝大多数程序员已经不再学习机器语言了。汇编语言是用英文助记符表示的符号语言，克服了机器语言的难读、难改的缺点，保持了机器语言的相应优点，但它和机器语言一样都是面向机器的低级语言，使用时必须对计算机内部结构有较为深入的了解。在汇编语言中，用助记符代替机器指令的操作码，

用地址符号或标号代替指令或操作数的地址，例如用 ADD 表示加法，从而增强程序的可读性并降低编程难度。汇编语言和机器自身的编程环境息息相关，因此仍然存在通用性差、可读性差的缺点。

高级语言接近自然语言，因此编写的程序易学、易读、易改，通用性好，而且不依赖机器，是一种面向过程、面向对象的程序设计语言。常见的高级语言有 Java、C、C++、Python、Swift、JavaScript、BASIC 和 Scratch 等。高级语言编写的程序不能被计算机识别，必须经过转换才能被执行，按转换方式可将它们分为两类：编译型和解释型。编译型语言将源代码一次性转换成目标代码，执行编译过程的程序称为编译器，如 C、C++、Java 等。解释型语言将源代码逐条转换成目标代码同时逐条运行，执行解释过程的程序称为解释器。

Python 是一种解释型、面向对象的高级程序设计语言，它是一款易于学习且功能强大的开放源代码的编程语言，可以快速帮助人们完成各种编程任务，并且能够把用其他语言制作的各种模块很轻松地联结在一起。使用 Python 编写的程序可以在绝大多数平台顺利运行。近年来 Python 的影响逐年扩大，而且整体呈上升趋势，反映出 Python 的应用越来越广泛，也越来越得到业内的认可。

Python 的创始人是荷兰人吉多·范罗苏姆 (Guido van Rossum)。1989 年，吉多·范罗苏姆开始开发一个新的脚本解释程序，作为 ABC 语言的一种继承，也就是 Python 的编译器。吉多希望这个称作 Python 的语言能符合他的理想——创造一种处于 C 和 Shell 之间，功能全面、易学易用、可拓展的语言。

1991 年，Python 公开发布第一个版本。由于其功能强大且采用开源方式发行，Python 发展迅猛，用户越来越多，逐渐形成了一个强大的社区力量。2000 年 10 月 16 日，Python 2.0 发布，实现了完整的垃圾回收功能，并且支持 Unicode，同时其开发过程更加透明。2008 年 12 月 3 日，Python 3.0 发布。

目前 Python 的主流版本主要有 Python 2.x 和 Python 3.x 系列。与 Python 2.x 系列相比，Python 3.x 系列在语法层面和解释器内部都做了很多重大的改进，同时在语句输出、编码、运算和异常等方面也做了一些调整，因此 Python 3.x 系列版本的代码无法向下兼容 Python 2.x 系列。从总体趋势而言，会有越来越多的开发者选择 Python 3.x，放弃 Python 2.x。我们在使用 Python 编制程序时，也要发扬这种精益求精的工匠精神，不断调整优化代码。此外，围绕 Python 3.x 的第三方库也会逐渐丰富起来。建议选用目前流行的 Python 3.x 系列版本进行开发（本书是以 Python 3.x 为默认运行环境撰写的）。

### 1.2.2 Python 的特点

Python 简单易懂，初学者学习 Python 不但入门容易，而且通过深入学习，可以编写一些功能非常复杂的程序。

#### 1. Python 的优点

(1) 简单易学。作为初学 Python 的人员，直接的感觉就是 Python 非常简单，非常适合阅读，阅读一个良好的 Python 程序就像是在读英语文章一样。Python 虽然是用 C 语言编写的，但是它摒弃了 C 语言中非常复杂的指针，简化了 Python 的语法结构。

(2) 免费开源。Python 是自由/开放源代码软件 (Free/Libre and Open Source Software, FLOSS) 之一。简单地说，用户可以自由地发布这个软件的备份、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 的开发者希望 Python 能让更多优秀的人参与创造并经常改进。

(3) 面向对象。Python 既支持面向过程的函数编程，也支持面向对象的抽象编程。面向对象的程序设计更加接近人类的思维方式，是对现实世界中的客观实体进行结构和行为的模拟。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。与其他主要的语言（如 C++ 和 Java）相比，Python 以一种非常强大又简单的方式实现面向对象编程。

(4) 移植性强。由于 Python 具有开源的本质，因此它已经被移植到许多平台上（经过改动能够工作在不同平台上）。如果开发者能小心地避免使用 Python 依赖于系统的特性，那么几乎所有 Python 程序无须修改就可以在下述平台运行，包括 Linux、Windows、FreeBSD、Macintosh、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，甚至还有 PocketPC、Symbian 以及 Google 基于 Linux 开发的 Android 平台。

(5) 强大的生态系统。Python 自身具有丰富和强大的库，同时还拥有数量众多的第三方扩展库，这使人们通过编程实现相应的功能变得非常简单，这也是 Python 得以流行的原因之一。在实际应用中，Python 的用户群体中的绝大多数并非专业的开发者，而是其他领域的爱好者。对于这一部分用户来说，他们学习 Python 的目的不是去做专业的程序开发，而仅仅是使用现成的类库去解决实际工作中的问题。Python 极其庞大的生态系统刚好能够满足这些用户的需求，同时其丰富的生态系统也给专业开发者带来了极大的便利。Python 大量成熟的第三方库可以直接使用，专业开发者只需要使用很少的语法结构就可以编写出功能强大的代码，缩短了开发周期，提高了开发效率。常用的 Python 第三方库包括 Matplotlib（数据可视化库）、NumPy（数值计算功能库）、SciPy（数学、科学、工程计算功能库）、pandas（数据分析高层次应用库）、Scrapy（网络爬虫功能库）、BeautifulSoup（HTML 和 XML 的解析库）、Django（Web 应用框架）、Flask（Web 应用微框架）等。

(6) 编程规范。Python 通过强制缩进来体现语句之间的逻辑关系，使代码可读性强，进而增强了 Python 程序的可维护性。常用的 Python 集成式开发环境都具有自动缩进的机制，例如输入“:”之后，按 Enter 键会自动进行缩进。我们编制程序要遵循语法规则，没有规矩，不成方圆，应牢固树立规则意识。

为了提高程序的可读性，可在程序的适当位置加上必要的注释。Python 中的注释有两种：行注释和块注释。

行注释：以“#”开头，可以单独成行，也可以跟在某行代码的后边。

块注释：也称多行注释，用 3 个单引号（'''）或者 3 个双引号（"""）将多行注释括起来，通常用于对函数、类等的大段说明。

## 2. Python 的缺点

(1) 运行速度慢。由于 Python 是解释型语言，所以它的速度会比 C、C++、Java 稍微慢一些。不过对一般用户而言，由于现在的硬件配置都非常高，硬件性能的提升可以弥补软件性能的不足，因此机器上运行速度的因素是可以忽略的。如果用户有速度要求，那么也有解决办法，例如可以嵌入 C 语言程序以提高运行速度。

(2) 不能加密。Python 的开源性使 Python 语言不能加密。我们在发布 Python 程序时，实际上就是发布源代码。但是目前国内市场纯粹靠编写软件卖给客户的情况越来越少，网站和

移动应用不需要给客户源代码，所以这个问题也就可以忽略了。

Python 的优点有很多，是程序设计语言领域近年来最重要的成果之一，在产业界被广泛使用，是国际上最流行的程序设计语言。我们在使用 Python 语言进行编程时，也要学会正确认识自己，明白自身的优缺点，发挥特长，补齐短板。

### 1.2.3 Python 的应用领域

Python 的应用领域非常广泛，概括起来有如下 9 大类。

#### 1. Web 开发

Python 是目前 Web 开发的主流语言之一，其类库丰富，使用方便，能够为一个需求提供多种方案。常用的 Web 开发框架有 Django、Flask、Tornado、Web2py 等，这些框架能够让用户快速方便地构建功能完善的高质量网站。目前很多大型网站如豆瓣、YouTube 等都是用 Python 开发的。

#### 2. 网络爬虫

除了 Python 自身的标准库 urllib 外，还有众多的第三方扩展库，如 Requests、BeautifulSoup 以及一些网络爬虫框架如 Scrapy，这些库使利用 Python 进行爬虫开发更加方便高效。

#### 3. 人工智能

Python 生态圈拥有大量用于机器学习、深度学习、图像识别、自然语言处理等人工智能领域的第三方扩展库，如 sklearn、TensorFlow、PyTorch、NLTK 等。

#### 4. 云计算

目前最流行、最知名的云计算框架是 OpenStack，它正是由 Python 开发的。Python 流行的很大一部分原因就是云计算的发展。

#### 5. 自动化运维

Python 是一种脚本语言，本身提供了一些能够调用系统功能的库，可编写脚本程序来控制系统，实现自动化运维。目前常用的系统自动化运维工具如 Ansible、Airflow、Celery、Paramiko 等都是用 Python 开发的。

#### 6. 科学计算

从 1997 年开始，美国航空航天局（National Aeronautics and Space Administration, NASA）就大量使用 Python 进行各种复杂的科学计算。随着 NumPy、SciPy、Matplotlib 和 Enthought librarys 等众多程序库的开发，使 Python 越来越适合于做科学计算、绘制高质量的 2D 和 3D 图像。与科学计算领域最流行的商业软件 MATLAB 相比，Python 是一门通用的程序设计语言，比 MATLAB 所采用的脚本语言的应用范围更广泛。

#### 7. 游戏开发

使用 Python 可以用更少的代码描述游戏业务逻辑，可以大大缩减大型游戏项目的代码量，因此很多游戏开发者先利用 Python 来编写游戏逻辑代码，再使用 C++ 编写图形显示等对性能要求较高的模块。Python 的 Pygame 模块可以制作 2D 游戏。

#### 8. 多媒体应用

Python 的 PIL、Piddle、ReportLab 等模块可方便地处理图像、声音、视频、动画等，并可动态生成统计分析图表，同时还可处理 2D 和 3D 图像，因此 Python 也被广泛应用于多媒体处理中。

## 9. 金融分析

目前, Python 是金融分析、量化投资领域中使用最多的开发语言。量化投资就是采用计算机技术及数据挖掘模型, 实现自己的投资理念或投资方法的一种过程。很多公司编写的分析程序、高频交易软件就是用的 Python 语言。

### 1.2.4 Python 开发环境的搭建

Python 可以用于多种平台, 包括 Windows、Linux 和 Mac OS 等。学习 Python 首先需要安装开发环境。安装 Python 后会得到 Python 解释器, 它负责运行 Python 程序。Python 可以在命令行交互环境或集成开发环境下运行。IDLE 是 Python 自带的集成开发环境, 其界面简洁, 使用简单方便, 适合小型项目的开发和初学者使用。当安装好 Python 以后, IDLE 就自动安装好了, 不需要另外安装。请到 Python 官网 (<https://www.python.org/>) 下载与自己计算机操作系统匹配的安装包并安装。本书采用的操作系统是 Windows, 使用 Python 3.x 运行环境。

### 1.2.5 Python 第三方开发工具

Python 的第三方开发工具有很多, 其中比较常用的有 Anaconda、PyCharm、Eclipse+PyDev 等。

#### 1. Anaconda

Anaconda 是一个用于科学计算的 Python 发行版本, 支持 Linux、Mac、Windows 系统, 包含了众多流行的科学计算、数据分析的 Python 包, 在数据分析与数据挖掘方面具有优势, 是数据科学家和数据分析人员的首选开发环境。Anaconda 中包含 NumPy、pandas、Matplotlib 等库, 利用 Anaconda 可以让用户免于将过多的精力花在环境搭建上, 从而快速进入 Python、数据分析、机器学习等领域的探索当中。

#### 2. PyCharm

PyCharm 是一款功能强大的 Python 编辑器。PyCharm 拥有一般的集成开发环境应该具备的功能, 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。另外, PyCharm 还提供了一些很好的功能用于 Django 开发, 而且还支持 Google App Engine。

#### 3. Eclipse+PyDev

Eclipse 是一款基于 Java 的可扩展开发平台, 包括 Java SE、Java EE、Java ME 等诸多版本。除此之外, Eclipse 还可以通过安装插件的方式进行诸如 Python、Android、PHP 等语言的开发。PyDev 是一个功能强大的 Eclipse 插件, 能够将 Eclipse 当作 Python IDE, 使用户可以利用 Eclipse 进行 Python 应用程序的开发和调试。Eclipse+PyDev 对 IDLE 进行了封装, 提供了强大的功能, 非常适合开发大型项目。

## 1.3 项目分解



安装 Python 开发环境

### 任务 1: 安装 Python 开发环境

Python 是一个轻量级的软件, 我们可以在其官网上下载安装程序。

运行下载的 python-3.8.7-amd64.exe 安装包, 出现如图 1.1 所示界面, 提示有两种不同的

安装方式，如果要采用系统默认路径安装，则直接选择 **Install Now** 选项；如果想指定安装路径，则 **Customize installation** 选项。无论选用哪种安装方式，要注意勾选 **Add Python 3.8 to PATH** 复选框，这样省去了手动配置环境变量的麻烦。



图 1.1 Python 安装界面

Python 安装完成后，打开 Windows 系统的 cmd 命令提示符窗口，输入 `python` 后按 **Enter** 键，如果出现图 1.2 所示界面，则表明 Python 开发环境安装成功。

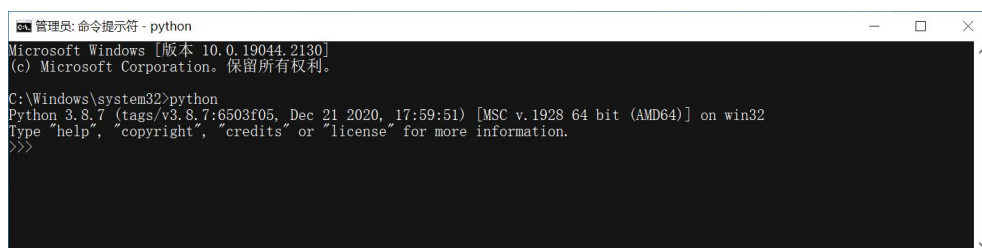


图 1.2 测试 Python 开发环境是否安装成功

## 任务 2：编写交互式代码

交互式编程不需要创建脚本文件，而是通过 Python 解释器的交互模式来编写代码的，图 1.2 所示的界面就是一个交互式执行环境。在交互模式中，每次只能执行一条语句。可以在 Python 命令提示符“>>>”后面输入 Python 代码，按 **Enter** 键后就会立即看到执行结果。例如，输入 `print("Hello World")`，运行效果如图 1.3 所示。

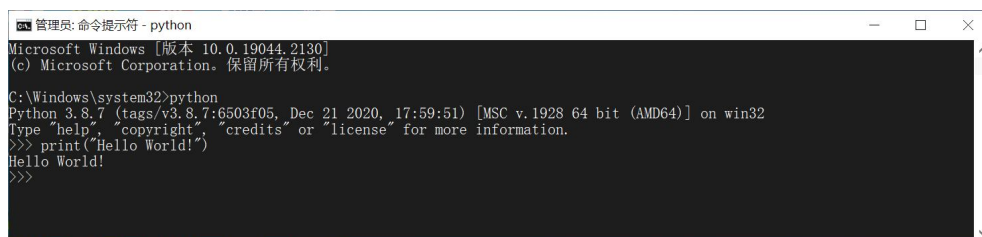


图 1.3 Python 交互式编程

### 任务 3: 运行程序文件

下面运行一个简单的 Python 程序文件。如在 Windows 系统的 D 盘根目录下已经存在一个程序文件 `example01-01.py`，该文件中只有如下一行代码：

```
print("Hello World")
```

打开 Windows 系统的 cmd 命令提示符窗口，切换至 `hello.py` 所在目录，并在命令提示符后面输入如下语句：

```
python D:\example01-01.py
```

运行结果如图 1.4 所示。



编写交互模式代码、  
运行程序文件

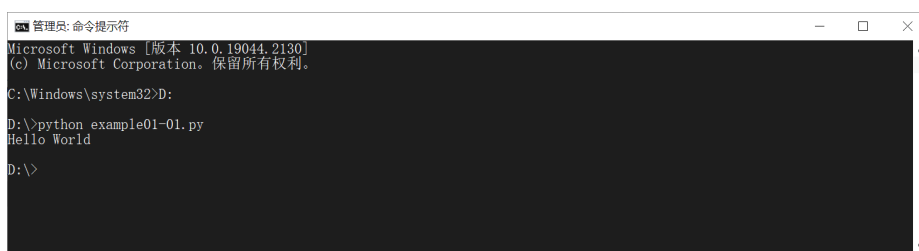


图 1.4 example01-01.py 运行结果

### 任务 4: 使用 IDLE 编写代码

集成开发环境（Integrated Development and Learning Environment, IDLE）是一个 Python Shell，程序开发人员可以利用 Python Shell 与 Python 交互。在 Windows 系统的“开始”菜单中找到 IDLE(Python 3.8 64-bit)，或者在 Windows 搜索框中输入 `idle`，也可找到 IDLE，打开进入 IDLE 主窗口，如图 1.5 所示。IDLE 启动后默认进入的是交互式模式，“`>>>`”为提示符，可以在提示符后面输入 Python 代码。普通语句输入完成后直接按 Enter 键就可执行该语句，而一些复合语句需要按两次 Enter 键才能执行。



使用 IDLE 编写代码

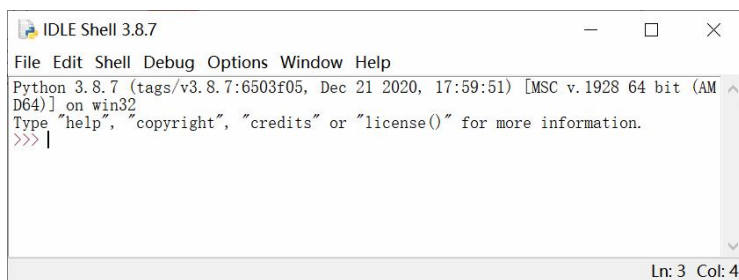


图 1.5 IDLE 主窗口

如果要创建一个代码文件，那么可以执行如下步骤。

(1) 新建文件。在菜单中选择 `File`→`New File` 选项打开一个新的 IDLE 窗口，在窗口中输入程序，如图 1.6 所示。

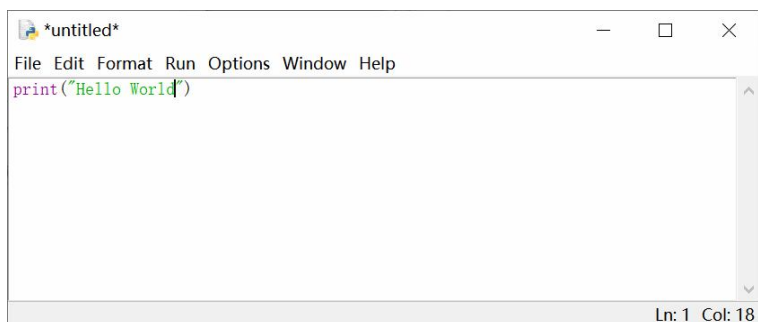


图 1.6 在 IDLE 窗口中创建程序

(2) 保存程序。在 IDLE 窗口中编写完程序后，在菜单里依次选择 File→Save 选项（或者用 Ctrl+S 组合键）来进行保存，首次保存时会弹出文件对话框，要求用户输入保存的文件名。此时保存的文件名为 example01-01.py。

(3) 运行程序。文件编辑完成后，可以按 F5 键运行程序，或选择 Run→Run Module 选项，程序运行结果如图 1.7 所示。

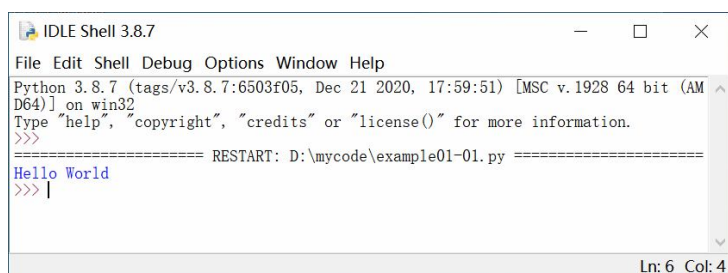


图 1.7 程序运行结果

在 IDLE 中，除了撤销（Ctrl+Z）、全选（Ctrl+A）、复制（Ctrl+C）、粘贴（Ctrl+V）、剪切（Ctrl+X）等常规组合键之外，其他比较常用的组合键见表 1.1。

表 1.1 IDLE 常用组合键

组合键	功能说明
Alt+P	浏览历史命令（上一条）
Alt+N	浏览历史命令（下一条）
Ctrl+F6	重启 Shell，之前定义的对象和导入的模块全部失效
F1	打开 Python 帮助文档
Alt+/ Ctrl+] ]	自动补全前面曾经出现过的单词，如果之前有多个单词具有相同前缀，则在多个单词中循环选择
Ctrl+] ]	缩进代码块
Ctrl+[ [	取消代码块缩进
Alt+3	注释代码块
Alt+4	取消代码块注释
Tab	补全单词

## 任务 5: 搭建 Python 集成开发环境

PyCharm 是一款功能强大的 Python 编辑器,具有跨平台性。进入 PyCharm 官方下载页面 (<https://www.jetbrains.com/pycharm/download/#section=windows>) 下载 Community 版(社区版)。注意 Professional 版(专业版)是收费的,Community 社区版是完全免费的。双击打开下载的安装包,安装界面如图 1.8 所示。

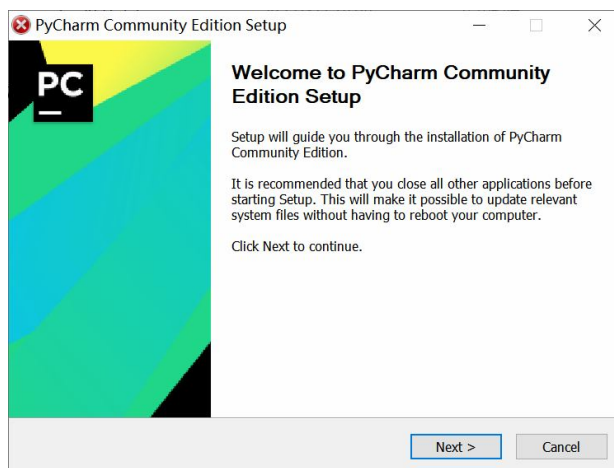


图 1.8 PyCharm 安装界面

单击 Next 按钮,设置 PyCharm 的安装路径,用户按照安装向导的提示逐步安装,安装完成后的界面如图 1.9 所示。

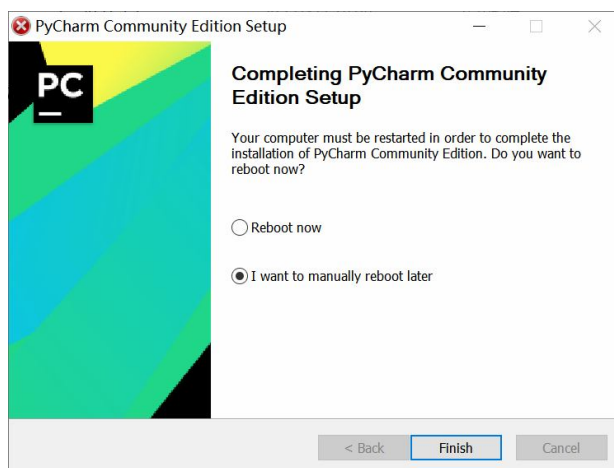


图 1.9 PyCharm 安装完成

用 PyCharm 创建项目并运行 Python 程序的步骤如下。

(1) 运行 PyCharm,选择 File→New Project 选项进入 CreateProject 界面,如图 1.10 所示。这里设置项目存储路径为 C:\Users\39685280\mypython,之后单击 Create 按钮进入项目界面,如图 1.11 所示。

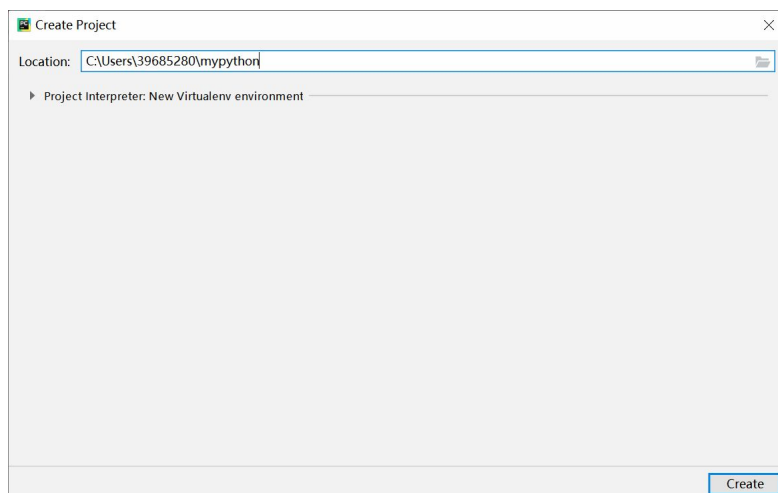


图 1.10 PyCharm 创建项目

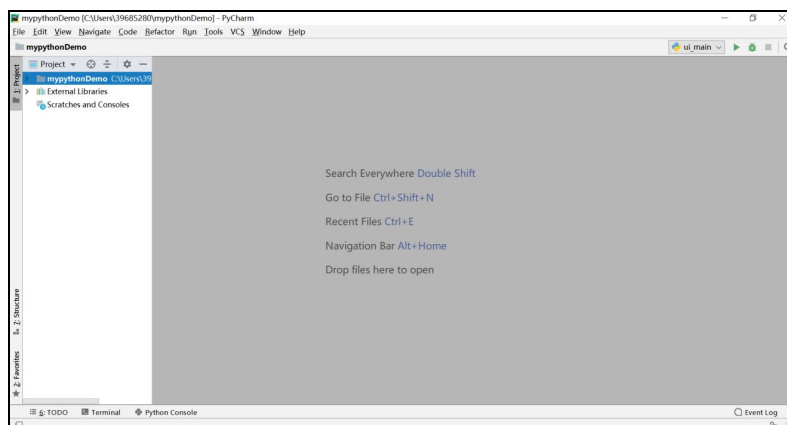


图 1.11 项目界面

(2) 选中项目名称，右击，在弹出的快捷菜单中选择 New→Python File 选项，弹出 New Python file 对话框，在该对话框的 Name 文本框中设置 Python 文件名为 example01-02，如图 1.12 所示。单击 OK 按钮后完成文件的创建，可以编辑 example01-02.py 程序，如图 1.13 所示。

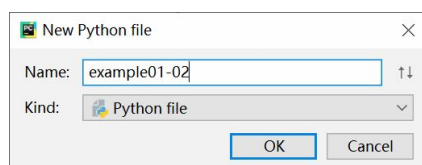


图 1.12 创建 Python 程序

(3) 在 example01-02.py 文件中输入下列代码：

```
print(" I can do it")
```

右击 example01-02.py 文件，在弹出的快捷菜单中选择 Run ‘example01-02’选项运行程序，如图 1.14 所示。

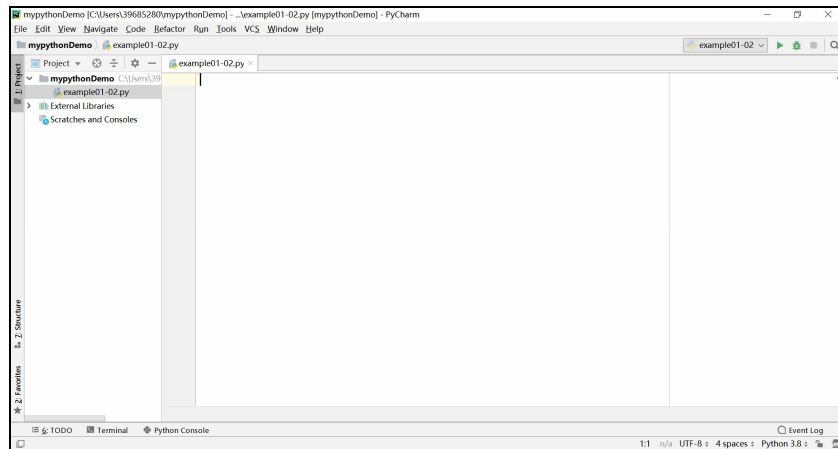


图 1.13 Python 程序编辑界面

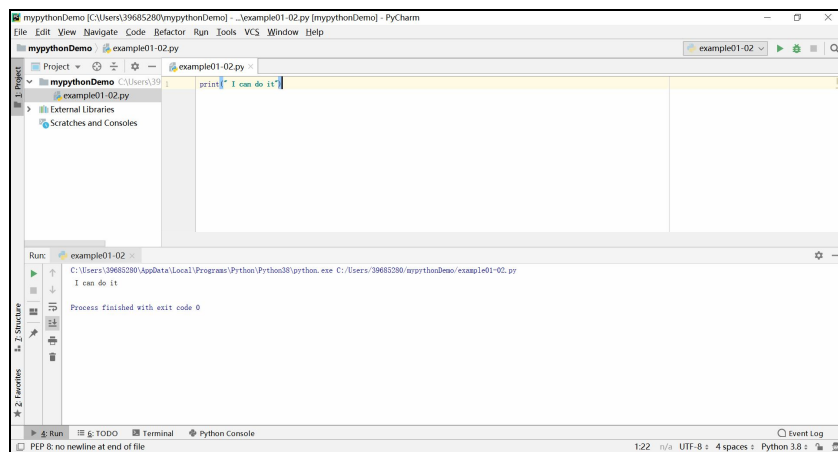


图 1.14 Python 程序运行界面

## 1.4 项目总结

本项目首先介绍了 Python 的诞生与发展、特点和应用领域；其次介绍了 Python 开发环境的搭建；然后简要介绍了常用的 Python 第三方开发工具；最后通过 5 个任务的上机实践使读者熟练掌握了 Python 开发环境的搭建和 Python 程序的运行。

## 1.5 习 题

1. 简述 Python 的特点。
2. 简述 Python 的应用领域。
3. 简述如何在 Python 中进行注释。
4. 简述常见的 Python 第三方开发工具。