第 1 章　运行环境和开发环境


运行环境和开发环境

知识目标

- 掌握计算机硬件、软件的基本概念；
- 掌握程序开发软硬件环境概念和在软件开发中的作用；
- 了解 Python 语言的发展历史；
- 熟悉程序编码、运行的过程；
- 了解程序开发工具的作用和用法。

实践目标

- 能够安装 Python 和 PyCharm 两个软件；
- 能够通过命令行进入 Python 编程环境；
- 能够通过 PyCharm 进入 Python 编程环境；
- 能够熟练建立工程，并成功运行示例程序；
- 能够完成综合训练，并熟练填写任务工单。

素养目标

- 理解信息化进程中创新的作用，培养创新精神；
- 养成认真仔细、精益求精的软件开发理念；
- 培养团队协作、有效沟通的能力；
- 培养爱岗敬业、履职尽责的职业精神。

1.1　Python 语言简介

程序是指计算机能够识别的一组指令，按照一定顺序和规则组合在一起；程序设计语言是用来书写指令的计算机语言，由一组记号和一组规则组成。计算机作为当前人类社会的一种重要辅助工具，其设计目的是提高人们的工作效率。计算机系统由硬件和软件组成，其中软件最主要的组成部分就是程序，即指挥计算机操作的指令集合，这些指令以计算机硬件能够“理解”的方式书写。也就是说，人们按照一定规则，应用程序设计语言设计出计算机软件，从而实现人和计算机之间的沟通交互，提高工作效率。人类借助程序语言设计程序从而指挥计算机，计算机通过执行程序完成工作。

1.1.1 程序设计语言简介

语言是人类用来沟通交流的工具之一，人与人之间使用语言进行交流，人类文明成果通过语言文字进行保存。

为什么不直接采用自然语言设计程序呢？一方面，自然语言种类繁多、千差万别，不利于计算机识别，据统计地球上有五千多种自然语言，显然让计算机识别每一种自然语言在当前阶段是不现实的；另一方面，自然语言语义在很多时候存在二义性和模糊性，一些词语无法作为计算机执行的指令，如汉语中一些“可能、也许”之类的词语，计算机系统无法给出具体执行结果。因此，必须单独设计出一套通用的指令系统，用于设计计算机程序，指挥计算机系统完成工作。

1. 机器语言

机器语言是最早、最原始的程序语言，也称为第一代程序语言。机器语言用二进制代码表示机器指令集合，与每台计算机的 CPU 等硬件有直接关系，难以理解、难以记忆，也难以掌握，很少有人能直接使用机器语言。但是机器语言是计算机芯片唯一能直接“理解”的语言，其他语言都需要翻译成机器语言。

2. 汇编语言

汇编语言是在机器语言之后出现的第二代程序设计语言，它将机器语言中指令符号化，使机器指令容易记忆，并且能够直接与符号相对应。尽管汇编语言已经采用符号来帮助人们记忆枯燥的指令，但是仍然难以记忆、难学难用。然而由于汇编语言可以面向计算机硬件系统直接编程，并且效率较高，因此在一些特殊的场合会利用汇编语言来进行程序设计。

3. 高级语言

在机器语言和汇编语言的基础上，又发展出了第三代程序语言，也就是我们通常所说的高级程序设计语言。第三代程序语言在形式上接近算术语言和自然语言，在概念上接近人们通常使用的语言，因此高级语言具有易学易用、便于记忆、通用性强、应用广泛等特点。高级语言的一个命令可以代替几条、几十条甚至几百条汇编语言指令，书写起来比较精简，大大提高了程序设计效率。高级程序语言主要包括以下几种。

（1）C 语言。C 语言是一种跨平台语言，既具有高级语言的特点，又具有汇编语言的特点，有时候人们把 C 语言称为中级程序语言。C 语言至今仍在广泛应用，也经常作为教学语言使用，并且现在流行的 Java 语言和 .NET 都以 C 语言语法为基础。

（2）BASIC 语言。BASIC 语言是一种结构简单但是功能强大的程序语言，其简单易学而且执行方式灵活，尤其是早期微软公司的大多数软件是用 BASIC 语言设计的，推动了 BASIC 语言的发展。随着程序设计技巧发展进步，在 BASIC 语言基础上发展出了 VB、VB.NET 等程序设计语言，很多计算机语言也借鉴了 BASIC 语言的语法结构。

（3）APT 语言。APT 语言是第一个专用语言，用于数控机床程序设计。

（4）Fortran 语言。Fortran 语言是第一个广泛使用的高级语言，其特点使其非常适合科学计算使用，为广大科研人员使用。在计算机语言发展初期，使用率非常高，但随着计

算机深入社会各个行业，Fortran 语言逐步被淘汰。

（5）COBOL 语言。COBOL 语言是一种面向商业的通用语言，是 20 世纪商业程序设计使用最广泛的语言，现在很多流行的程序语言是以它为原型演化出来的，COBOL 语言适用于数据处理。

（6）Pascal 语言。Pascal 语言是一种重要的结构化程序设计语言，非常适合教学，曾被各大高校作为程序设计教学语言使用，但现在基本上已被淘汰。

（7）Perl 语言。Perl 语言是一种广泛应用于 UNIX/Linux 系统管理的脚本语言，至今仍被广泛使用。

（8）C++ 语言。C++ 语言是在 C 语言基础上发展起来的一种面向对象的程序设计语言，便于构建大型软件，并且具有较高的效率，有利于项目管理和开发。

（9）Java 语言。Java 语言是 SUN 公司在 C 语言语法基础上发展起来的面向对象的程序设计语言，它不仅是一种语言，同时包括了一系列软件开发技术和软件设计思想。Java 语言是一种跨操作系统的语言，具有高通用性、高安全性和易开发等特性，是目前最流行的程序语言之一。

1.1.2　Python 语言

Python 语言是一种解释型、面向对象的程序设计语言，具有简洁易学、应用广泛等特点，适合零基础入门学习，不少地区中小学开始 Python 语言学习，普及程序设计基础，锻炼程序设计思维。

Python 语言同时是开源的、免费的、解释型高级动态编程语言，多次跃居编程语言排行榜首位，已成为当今最为流行的开发语言之一。程序设计方面具有语法灵活、模块众多和跨平台等优点，能够广泛应用于科学计算、Web 开发、大数据及人工智能等领域。随着 Python 语言的普及，不同领域不断丰富的拓展库也极大增强了 Python 的功能，反过来进一步推动了 Python 的发展，使其几乎渗透到了所有的领域和学科，被称为“胶水语言”。它能够将不同语言编写的程序融合在一起，集成不同语言工具优点，满足人们实际应用要求，当前越来越多的院校已采用 Python 语言作为程序设计教学语言。

1. Python 语言发展历史

Python 语言诞生于 1990 年，是由著名计算机科学家吉多·范罗苏姆（Guido van Rossum）设计并领导开发。诞生 30 多年来，Python 语言主要形成了两大版本系列，即 Python 2.X 和 Python 3.X，二者比较相似，但也存在一定差异。

（1）Python 2.X。2000 年 10 月，Python 2.X 系列开始发布，目前最新版本为 Python 2.7，并且不再进行重大更新，主要适用于历史遗留 Python 代码。

（2）Python 3.X。2008 年 12 月，Python 3.X 系列开始发布，在语法和解释器内部都进行了较多改进，是当前流行的主要版本。本书采用 Python 3 语法进行示例。

2. Python 语言特点

Python 语言除了可以解释执行之外，还支持将其源代码通过伪编译方式得到字节

码，以提高程序运行速度，同时在一定程度上对源代码进行加密，起到源代码保护作用。Python 有命令式和函数式两种编程模式，支持面向对象程序设计，便于构建多模块、多功能和多环境软件架构，并且具有较高开发效率，有利于软件项目管理和开发，具有较强通用性。其主要特点总结如下。

（1）简单且易学。Python 语言是一种解释型编程语言，遵循简单、明确、高效的设计原则，程序语法清晰易读，是程序设计初学者的入门首选。

（2）免费且开源。Python 语言是当前主流的开源程序设计语言之一，既允许自由下载、阅读和修改，也允许将其包含于其他符合开源协议的软件中发布。

（3）高级且易开发。Python 语言是一门高级程序设计语言，用户无须考虑底层实现细节，并且内置包含列表、元组和字典等高级数据结构，方便进行程序开发。

（4）面向对象且易移植。Python 语言支持面向过程和面向对象程序开发，提供类继承、重载和多态等功能，具有较强可复用性。Python 语言也支持跨平台开发，在主流 Windows、Linux 和 Mac 等环境下均可部署和运行，易于程序在不同平台上移植。

（5）功能丰富且易拓展。Python 语言提供了功能丰富的标准库，支持通过 pip 等命令安装第三方自定义库，方便不同领域应用需求。Python 语言也提供了丰富 API 和工具，便于软件拓展和系统集成，便于不同领域程序员协同开发。

（6）可嵌入其他语言。Python 语言具有灵活的可嵌入性，如将其嵌入 C/C++ 程序中，以提高其他程序语言脚本化编程能力。

基于以上特点，不少高校采用 Python 作为程序设计入门教学语言，不仅计算机相关专业大类学生学习 Python 语言，不少非计算机专业大类学生也选择学习 Python 语言，以获取程序设计思维，帮助其他专业课学习。一旦熟练掌握 Python 语言，不仅能够帮助学生掌握程序设计思维，并且对于学习其他程序语言以及大数据和人工智能算法等课程，都有非常大的帮助。

下面先来看一段用 Python 语言设计的程序源代码。

【例 1-1】 计算某选修课平均成绩并输出。

程序代码如下：

```
#  选修课原始成绩列表
v=[65,90,77,80,76,55,64,88,92,83,72,93,50,74,49,54,71,87,63,67]
#  使用 len 函数求出人数
n=len(v)
#  初始化平均数
avg=0
for i in range(n):
#  根据原始成绩求出成绩总和
    avg=avg+v[i]
#  求出平均成绩
avg=avg/n
#  输出计算结果
print('平均成绩  ',avg)
```

这段程序运行结果如下：

```
平均成绩：  72.5
```

在这段程序中，以“#”开头的行为注释，注释是为了方便阅读程序，并不被计算机执行。其余为正式程序，又称为源代码，用来实现相应功能。那么这些源代码该放到哪里执行才能够得到想要的结果呢？怎样才能让计算机执行这些代码或者指令呢？或者说，要经过一个什么样的过程，才能得到我们想要的结果呢？例 1-1 中计算了 20 个人的平均成绩，如果不是 20 个人呢？平均成绩经过计算后，又将结果输出到哪里呢？如何查看结果？

通常情况下，可在设计环境中设计好源代码，并在相应运行环境（包括软硬件环境）中运行。接下来，我们将逐步了解运行 Python 程序的相关基础知识和操作方法，熟悉运行环境、开发环境配置，掌握程序设计知识和方法。

1.2　运 行 环 境

运行一个程序，不仅需要计算机、网络、存储等硬件设备支持，同时需要考虑操作系统、配套软件、数据库软件等软件兼容等问题。程序运行所需要的硬件设备和软件共同构成了程序运行环境，称为硬件运行环境和软件运行环境。

1.2.1　硬件运行环境

硬件运行环境是指能够支持软件运行的硬件系统，随着计算机软件的发展，运行软件对硬件的要求越来越高。例如，2GB 内存可以支持 Windows XP 运行，但是对于 Windows 7 或 Windows 10 就捉襟见肘了。Python 基础开发对计算机硬件配置的要求并不高，普通的四核 CPU、4GB 内存、100GB 硬盘即可满足基础的开发需求。但是 Python 也经常用在一些复杂场合中，如大数据、人工智能等，这些开发场景可能会对内存、显卡等配置有特殊的要求。下面重点从程序与内存的角度来介绍程序硬件运行环境。

存储器是组成计算机的重要部分，其主要功能就是存储程序和数据，正是因为有了存储器，计算机才能具备“记忆”功能，是计算机智能化基础条件之一。计算机中主要存储器包括硬盘和内存，例 1-1 中提到的源代码，正是通过键盘输入（键盘和鼠标为计算机系统的输入设备），完成输入的源代码以文件形式存储在计算机硬盘中。因为硬盘中的数据即使在断电情况下也可以保存，因此完成输入的源代码不会丢失，设计人员下次既可以继续编辑，也可以通过 U 盘等外界存储设备复制到不同计算机上进行编辑。

在 Python 语言中，编辑好的源代码以源文件形式存放在硬盘上，此时该源代码无法直接运行，只有当设计人员下达运行指令时，代码从计算机硬盘中调入内存中，由 Python 解释器将其转换为字节码，再由 Python 解释器来执行这些字节码，从而得到运行结果。如图 1-1 所示，程序运行过程中将内存作为载体，程序启动时，需要将源代码从硬盘加载到内存，并由内存为源代码运行提供基础环境。源代码执行完毕，如果需要输出结果，则可将结果通过显示器或打印机等设备输出。

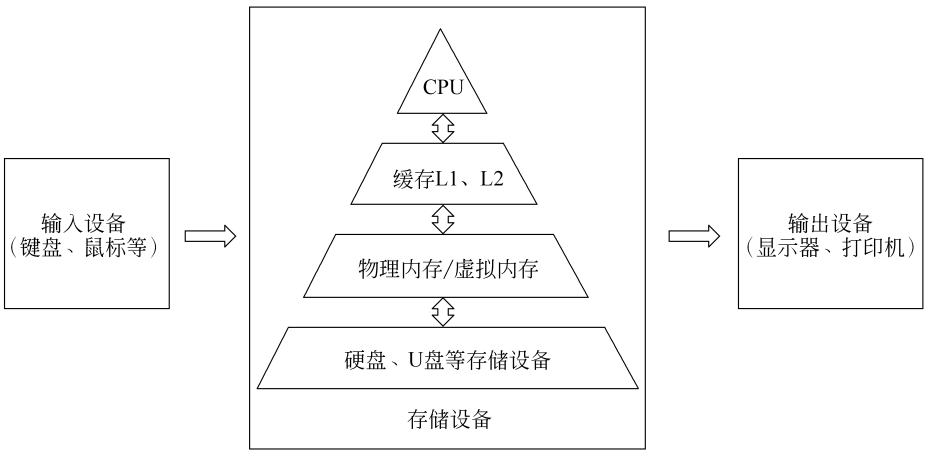

图 1-1　程序存储与运行

从图 1-1 中可以看出，程序是一组计算机能够识别的指令，这些指令按照一定的顺序组合，并以文件的形式保存。运行该程序时，计算机可以依据这些指令，有条不紊地进行工作。为了使计算机系统能实现各种功能，需要不同的成千上万的程序，这些程序既可以“同时”运行，也可以按照程序设计人员的意志依次执行。

1.2.2　软件运行环境

软件运行环境即操作系统平台和程序运行支撑软件（软件相关知识见本章电子活页）。不同程序设计语言往往需要不同软件运行环境的支持，如早期的 BASIC、Fortran、.NET 系列等语言只能运行在 Windows 操作系统上，在 Linux 上无法运行（现在可以通过安装一些支撑软件来运行）。后来出现了一些跨平台程序设计语言，如 C 语言、Java 语言，但是 Java 语言实际上需要安装 JVM 支持，从而忽视操作系统。程序设计语言运行环境和一些常见软件一样，需要考虑不同操作系统，如 Windows 32、Windows 64、Linux 和 macOS 等之间的差异。

Python 语言也是跨平台开发语言，其软件运行环境对操作系统平台没有要求，但需要安装 Python 解释器，下面我们以 Window 10 操作系统为例进行演示。

1. 下载文件

如图 1-2 所示，访问官网，单击 Download Python 3.10.2 按钮，下载得到如图 1-3 所示安装文件。注意由于 Python 处于持续发展阶段，所以不同时期下载的 Python 版本存在差异。此外，如果使用其他操作系统，如 Linux、macOS 等，也可按页面提示选择对应下载链接进行切换。

2. 安装下载好的文件

如图 1-3 所示，下载得到的安装文件为 exe 格式，与其他 Windows 软件安装过程类似，可直接双击此 exe 文件从而进入安装向导。为便于操作，此处尽量按软件默认选项进行安装，具体过程如下。

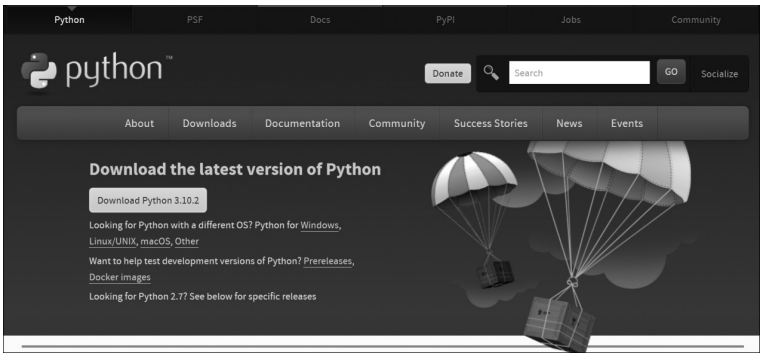

图 1-2　Python 下载页面

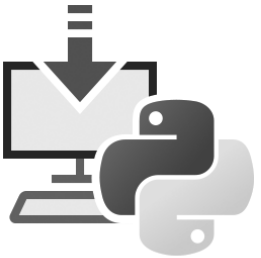

图 1-3　Python 安装文件

（1）如图 1-4 所示，选中底部 Add Python 3.10 to PATH 复选框，这样能自动配置 Python 路径到操作系统中，自动完成环境变量设置，便于系统指令找到执行程序。

图 1-4　设置安装选项

（2）单击 Install Now 选项，执行安装，如图 1-5 所示。

（3）如图 1-6 所示，等待安装完成。

（4）如图 1-6 所示，部分计算机由于操作系统配置原因，可能会出现 Disable path length limit 的提示项，此时可单击此提示以完成最后的配置，最终结果如图 1-7 所示。

单击右下角的 Close 按钮，即可完成安装。根据前面的默认配置，安装目录为 C:\ 用户 \ Administrator\AppData\Local\Programs\Python\Python310，进入此目录可查看已安装的文件信息，如图 1-8 所示。

至此，Python 基础运行环境安装完毕，类似于例 1-1 这样的简单程序可以在该环境下运行。我们先对运行环境安装结果进行检验。进入 cmd 窗口（在操作系统搜索框中输入 cmd，然后按 Enter 键，即可进入 cmd 窗口），在命令提示符下输入 python 命令，按 Enter 键就可以得到如图 1-9 所示界面，这里显示了所安装的 Python 版本、版权等信息。

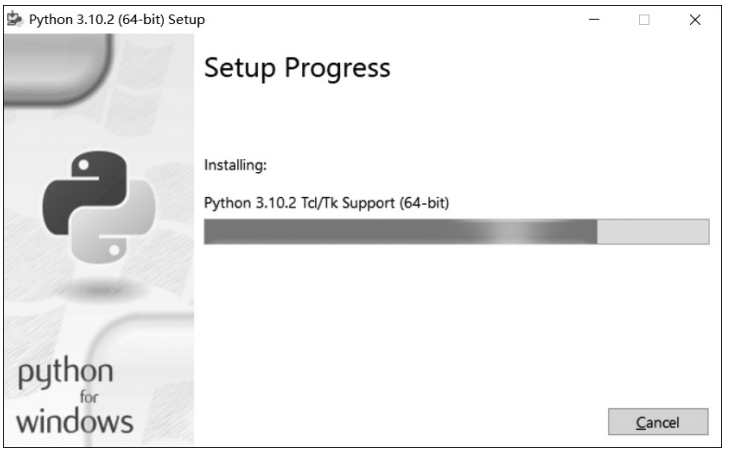

图 1-5　执行安装

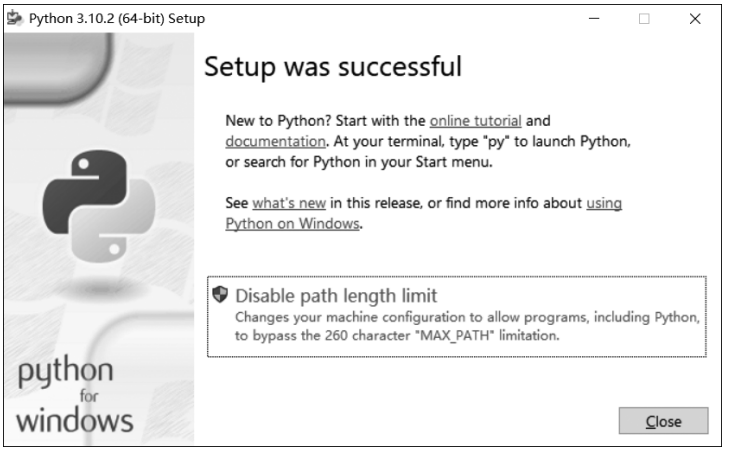

图 1-6　完成安装

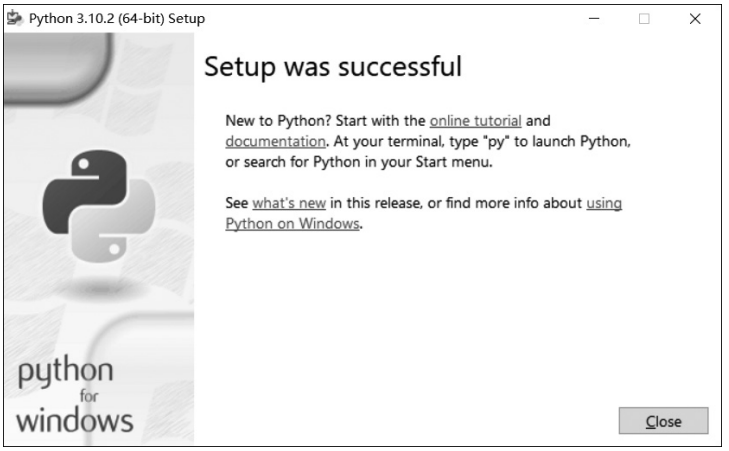

图 1-7　安装成功

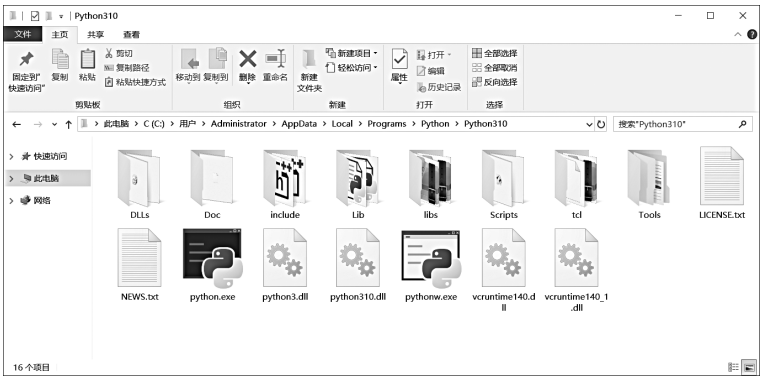

图 1-8　安装目录

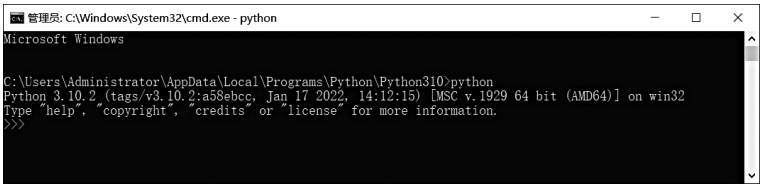

图 1-9　Python 版本、版权信息

注意：当在 cmd 窗口输入 python 命令以后，其提示符变为 >>>，这说明 Python 环境已经启动。在该提示符下，通过 Python 语言自带命令，可以实现运行程序、查看错误、调试程序等功能，在本书后面会进行介绍。

1.3　开 发 环 境

这里的开发环境主要是指软件开发环境（software development environment，SDE），硬件能够支持和运行开发环境所需软件即可，这里就不再单独阐述。软件开发环境是指在基本硬件和软件基础上，为支持系统软件和应用软件工程化的开发和维护而使用的一组软件。也就是用来支持书写和调试源代码的一组软件（一些大型项目工程需要多个软件配合），注意同一个项目、同一种语言，有很多种 SDE 可供选择，不同 SDE 有不同特点，至于如何选择，则主要根据项目特点和程序员的习惯偏好。常见的支持简单 Python 程序设计的 SDE 工具软件主要有以下几种。

（1）免费开发工具 Pydev+Eclipse。Pydev+Eclipse 是一组非常普遍、免费的 Python 开发工具，同时提供很多强大的功能以支持 Python 程序设计，如 Django 集成、自动代码补全、多语言支持、集成 Python 调试、代码分析等，这些强大功能使得其成为程序员的首选。

（2）一个开源并且遵循 GPL 协议、同样免费的文本编辑器 VIM。VIM 在 Python 程序设计人员中很受喜欢。VIM 不仅是一个功能强大、界面友好的文本编辑器，还是一个轻量级、模块化、快速响应的工具。同时 VIM 是一个支持 Linux 系统的 Python IDE，这满足了 Linux 开发者需求。

（3）专业的 Python 集成开发环境 PyCharm。PyCharm 是专门针对 Python 语言设计的 SDE 工具，是由 JetBrains 打造的一款 Python 集成开发环境，支持跨平台使用且具有强大的功能，包括代码调试、代码跳转、代码补全、智能提示、语法高亮、项目管理、单元测试和版本控制等。PyCharm 也支持插件拓展，提供方便的插件安装和管理功能，得到广大 Python 开发人员的青睐。PyCharm 有两个版本，一个是免费社区版本，另一个是面向企业开发者的专业版本。专业版本要更加高级，支持更多高级功能，如远程开发、数据库支持等。对于普通开发者和 Python 语言初学者而言，免费社区版本提供了足够多功能。

除了上述三种 SDE 以外，还有不少 SDE 能够支持 Python 语言进行程序开发，这里就不一一阐述了。接下来将对 PyCharm 下载、安装、使用过程进行详细介绍，本书大部分案例也是基于 PyCharm 进行开发的，本章介绍如何用 PyCharm 进行例 1-1 的开发，后面章节会逐步介绍其他功能。

1. 下载安装文件

官网提供多种版本的 PyCharm。如图 1-10 所示，本书选择 Windows 10 环境下的社区版 PyCharm 进行下载，得到的安装文件可参见图 1-11。

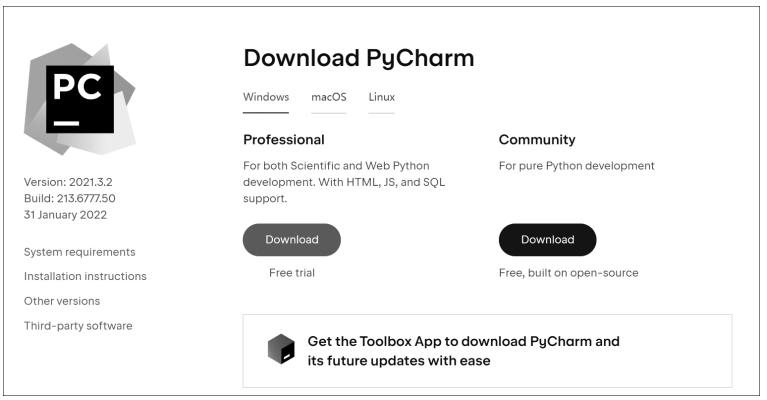

图 1-10　PyCharm 下载页面

图 1-11　PyCharm 安装文件

2. 安装 PyCharm

如图 1-11 所示，下载好的安装文件为 exe 格式，与其他 Windows 软件安装过程类似，可直接双击此exe文件从而进入安装向导。为便于操作，此处尽量按软件默认选项进行安装，具体过程如下所示。

（1）如图 1-12 所示，单击 Next 按钮进行安装。

（2）如图 1-13 所示，配置安装目录，单击 Next 按钮继续安装。

（3）如图 1-14 所示，配置安装选项，建议全部选中以便于使用并自动关联 .py，并单击 Next 按钮继续安装。

（4）如图 1-15 所示，单击 Next 按钮，执行安装。

（5）如图 1-16 所示，等待安装完成。