

第 1 章



初识网络爬虫

在这个大数据的时代里，网络信息量变得越来越大、越来越多，此时如果通过人工的方式筛选自己所感兴趣的信息是一件很麻烦的事情，爬虫技术便可以自动高效地获取互联网中的指定信息，因此网络爬虫在互联网中的地位变得越来越重要。

本章将介绍什么是网络爬虫？网络爬虫都有哪些分类、网络爬虫的基本原理以及爬虫环境的搭建工作。

1.1 网络爬虫概述

网络爬虫（又被称为网络蜘蛛、网络机器人，在某社区中经常被称为网页追逐者），可以按照指定的规则（网络爬虫的算法）自动浏览或抓取网络中的信息，通过 Python 可以很轻松地编写爬虫程序或者是脚本。

在生活中网络爬虫经常出现，搜索引擎就离不开网络爬虫。例如，百度搜索引擎的爬虫名字叫作百度蜘蛛（Baiduspider）。百度蜘蛛，是百度搜索引擎的一个自动程序。它每天都会在海量的互联网信息中进行爬取，收集并整理互联网上的网页、图片视频等信息。然后当用户在百度搜索引擎中输入对应的关键词时，百度将从收集的网络信息中找出相关的内容，按照一定的顺序将信息展现给用户。百度蜘蛛在工作过程中，搜索引擎会构建一个调度程序，来调度百度蜘蛛的工作，这些调度程序都是需要使用一定算法来实现的，采用不同的算法，爬虫的工作效率也会有所不同，爬取的结果也会有所差异。所以，在学习爬虫时不仅需要了解爬虫的实现过程，还需要了解一些常见的爬虫算法。在特定的情况下，还需要开发者自己制定相应的算法。

1.2 网络爬虫的分类

网络爬虫按照实现的技术和结构可以分为通用网络爬虫、聚焦网络爬虫、增量式网络爬虫。在实际的网络爬虫中，通常是这几类爬虫的组合物，下面分别介绍。

1. 通用网络爬虫

通用网络爬虫又叫作全网爬虫（Scalable Web Crawler），通用网络爬虫的爬行范围和数量巨大，正

是由于其爬取的数据是海量数据，所以对于爬行速度和存储空间要求较高。通用网络爬虫在爬行页面的顺序要求上相对较低，同时由于待刷新的页面太多，通常采用并行工作方式，所以需要较长时间才可以刷新一次页面。所以存在着一定的缺陷，这种网络爬虫主要应用于大型搜索引擎中，有着非常高的应用价值。通用网络爬虫主要由初始 URL 集合、URL 队列、页面爬行模块、页面分析模块、页面数据库、链接过滤模块等构成。

2. 聚焦网络爬虫

聚焦网络爬虫（Focused Crawler）也叫主题网络爬虫（Topical Crawler），是指按照预先定义好的主题，有选择的进行相关网页爬取的一种爬虫。它和通用网络爬虫相比，不会将目标资源定位在整个互联网中，而是将爬取的目标网页定位在与主题相关的页面中。极大地节省了硬件和网络资源，保存的页面也由于数量少而更快了，聚焦网络爬虫主要应用在对特定信息的爬取，为某一类特定的人群提供服务。

3. 增量式网络爬虫

增量式网络爬虫（Incremental Web Crawler），所谓增量式，对应着增量式更新。增量式更新指的是在更新时只更新改变的地方，而未改变的地方则不更新。所以增量式网络爬虫，在爬取网页时，只会在需要的时候爬行新产生或发生更新的页面，对于没有发生变化的页面，则不会爬取。这样可有效减少数据下载量，减小时间和空间上的耗费，但是在爬行算法上增加了一些难度。

1.3 网络爬虫的基本原理

一个通用网络爬虫的基本工作流程，如图 1.1 所示。

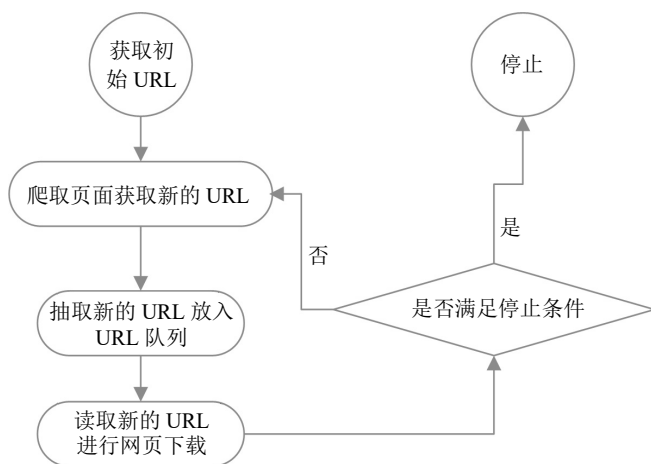


图 1.1 通用网络爬虫的基本工作流程

网络爬虫的基本工作流程如下。

- (1) 获取初始的 URL，该 URL 地址是用户自己制定的初始爬取的网页。
- (2) 爬取对应 URL 地址的网页时，获取新的 URL 地址。
- (3) 将新的 URL 地址放入 URL 队列。
- (4) 从 URL 队列中读取新的 URL，然后依据新的 URL 爬取网页，同时从新的网页中获取新的 URL 地址，重复上述的爬取过程。
- (5) 设置停止条件，如果没有设置停止条件，那么爬虫会一直爬取下去，直到无法获取新的 URL 地址为止。设置了停止条件后，爬虫将会在满足停止条件时停止爬取。

1.4 搭建开发环境

1.4.1 安装 Anaconda

Anaconda 是一个完全免费的大规模数据处理、预测分析和科学计算工具。该工具中不仅集成了 Python 解析器，还有很多用于数据处理和科学计算的第三方模块，其中也包含许多网络爬虫所需要使用的模块，如 requests 模块、Beautiful Soup 模块、lxml 模块等。

在 Windows 系统下的浏览器中打开 Anaconda 的官方地址 (<https://www.anaconda.com/distribution/>) 下载对应的安装文件，如图 1.2 所示。

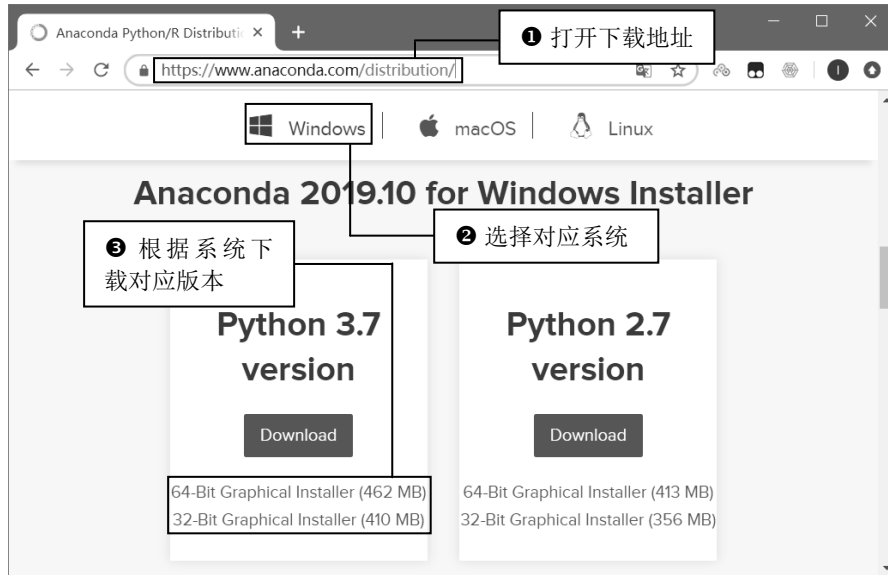


图 1.2 下载 Anaconda

这里笔者所选择的是 Windows（64-Bit Graphical Installer 为当时的最新版本），下载完成后直接双击运行下载的文件，在 Welcome to Anaconda3（自己下载的版本）窗口中直接单击 Next 按钮，如图 1.3 所示。

在 License Agreement 窗口中直接单击 I Agree 按钮，如图 1.4 所示。

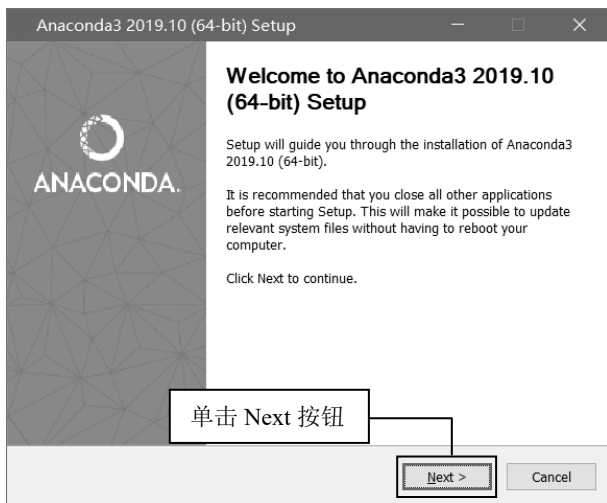


图 1.3 Welcome to Anaconda3 窗口

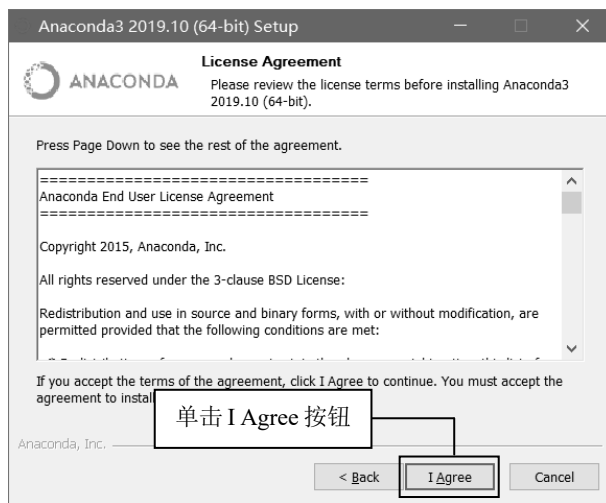


图 1.4 License Agreement 窗口

在 Select Installation Type 窗口内选中 All Users(requires admin privileges)单选按钮，然后单击 Next 按钮，如图 1.5 所示。

在 Choose Install Location 窗口中选择自己的安装路径（建议不要使用中文路径），这里笔者选择一个自定义的安装路径，然后单击 Next 按钮，如图 1.6 所示。

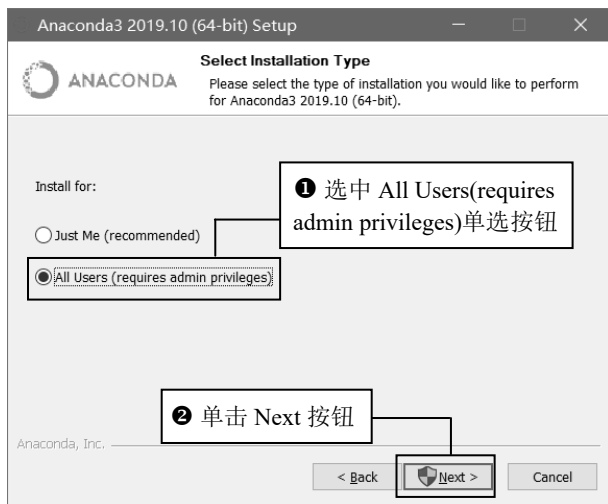


图 1.5 选中 All Users(requires admin privileges)单选按钮

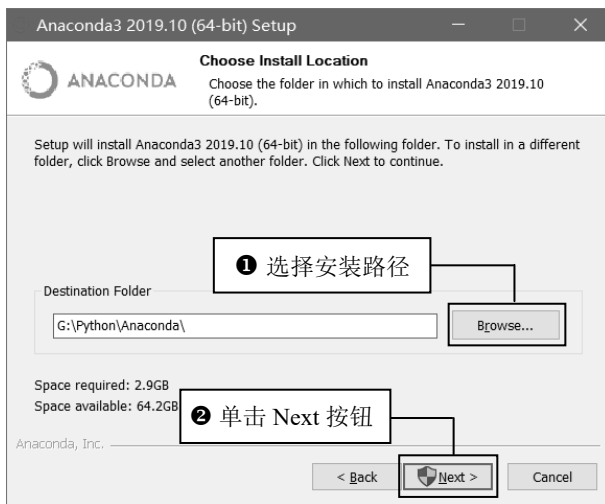


图 1.6 选择安装路径

在 Advanced Installation Options 窗口中，选中第一个复选框，将 Anaconda 加入环境变量，然后单击 Install 按钮进行安装，如图 1.7 所示。

由于 Anaconda 中包含的模块较多，所以在安装过程中需要等待的时间较长，安装进度如图 1.8 所示。

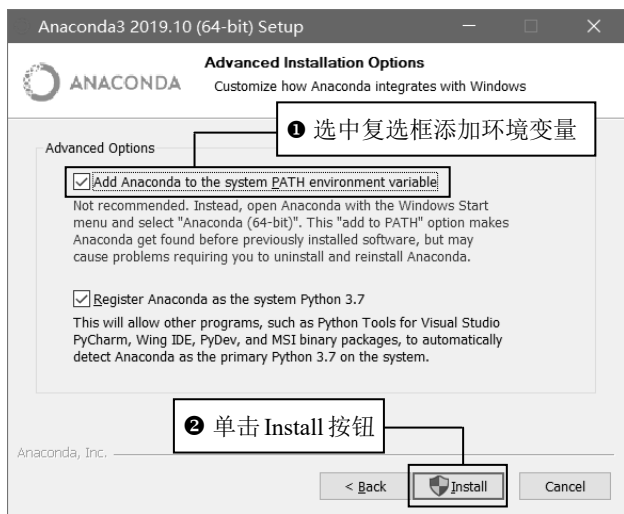


图 1.7 将 Anaconda 加入环境变量

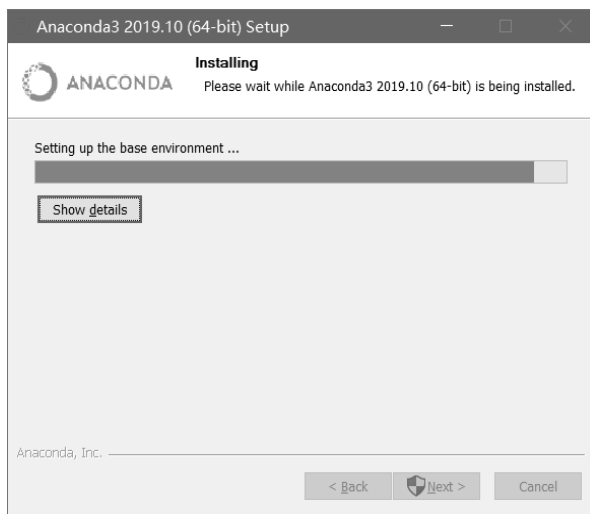


图 1.8 安装进度

安装进度完成以后，将进入 Installation Complete 窗口中，在该窗口中直接单击 Next 按钮，如图 1.9 所示。

由于 Anaconda 与 JetBrains 为合作关系，所以官方推荐使用 PyCharm 开发工具，在该窗口中直接单击 Next 按钮，如图 1.10 所示。



图 1.9 安装完成

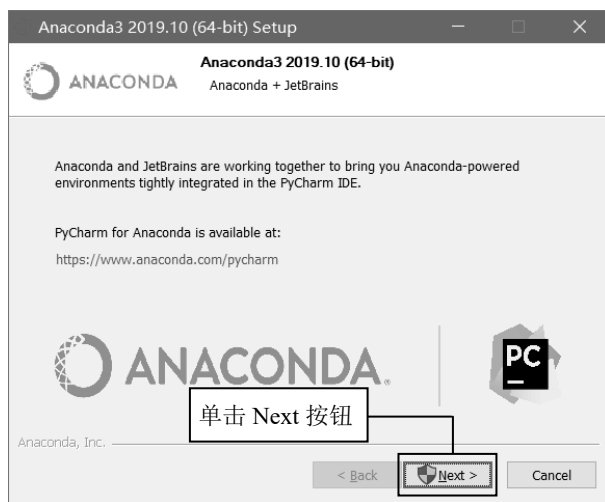


图 1.10 PyCharm 开发工具提示

最后在“Thanks for installing Anaconda3!”窗口中根据个人需求，选中或取消选中（笔者选择取消选中）两个复选框，再单击 Finish 按钮，如图 1.11 所示。

将 Anaconda 安装完成以后并保证已经添加系统环境变量的情况下，打开“命令提示符”窗口，然后输入“conda list”后按 Enter 键，即可查看当前 Anaconda 已经安装好的所有模块，如图 1.12 所示。

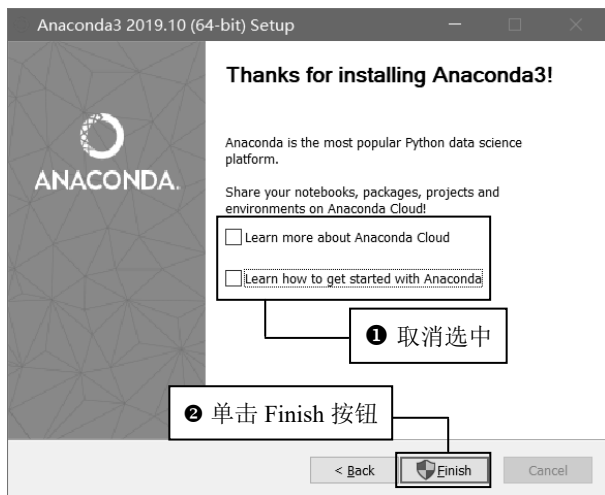


图 1.11 安装结束



图 1.12 查看当前 Anaconda 已经安装好的所有模块

1.4.2 PyCharm 的下载与安装

PyCharm 是由 JetBrains 公司开发的 Python 集成开发环境，由于其具有智能代码编辑器，可实现自动代码格式化、代码完成、智能提示、重构、单元测试、自动导入和一键代码导航等功能，目前已成为 Python 专业开发人员和初学者使用的有力工具。

打开 PyCharm 官网的下载地址 (<https://www.jetbrains.com/pycharm/download/>)，然后选择下载 PyCharm 的操作系统平台为 Windows，单击开始下载社区版 PyCharm (Community)，如图 1.13 所示。



图 1.13 PyCharm 环境与版本下载选择页面

双击 PyCharm 安装包进行安装，在欢迎界面单击 Next 按钮进入软件安装路径设置界面，如图 1.14 所示。

在 Choose Install Location 窗口中选择一个需要安装的路径，这里不建议将安装路径设置在默认的 C 盘中，笔者选择自定义安装路径，确认安装路径后单击 Next 按钮，如图 1.15 所示。



图 1.14 PyCharm 欢迎界面

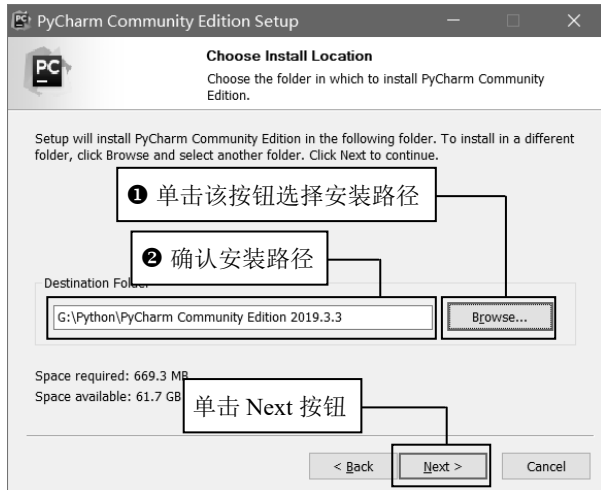


图 1.15 设置 PyCharm 安装路径

在 Installation Options 窗口中首先在桌面快捷方式 (Create Desktop Shortcut) 中设置 PyCharm 程序的快捷方式，笔者系统为 64 位，所以选中 64-bit launcher 复选框，然后设置关联文件 (Create Associations)，选中 “.py” 复选框，这样以后再打开.py (.py 文件是 Python 脚本文件，接下来编写的很多程序都是后缀名为.py 的文件) 文件时，会默认调用 PyCharm 打开，如图 1.16 所示。

在 Choose Start Menu Folder 窗口中直接单击 Install 按钮，如图 1.17 所示。

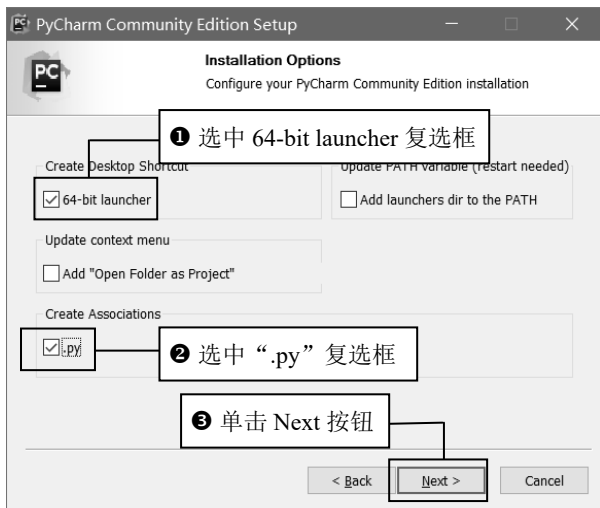


图 1.16 设置快捷方式和关联

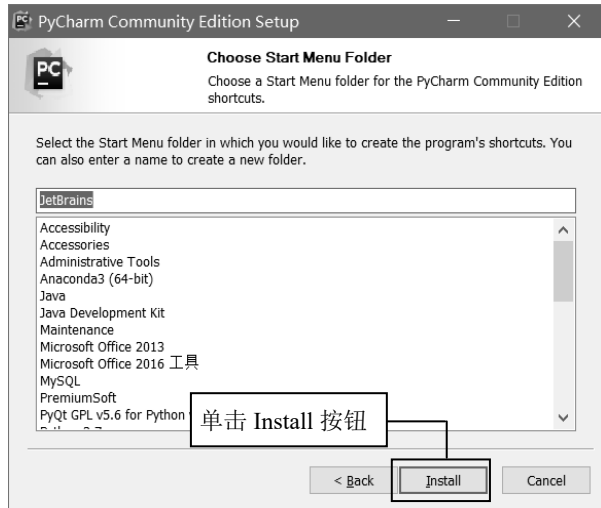


图 1.17 选择开始菜单文件夹窗口

安装进度完成以后，在 Completing PyCharm Community Edition Setup 窗口中，在不直接运行 PyCharm 开发工具的情况下，单击 Finish 按钮即可，如图 1.18 所示。

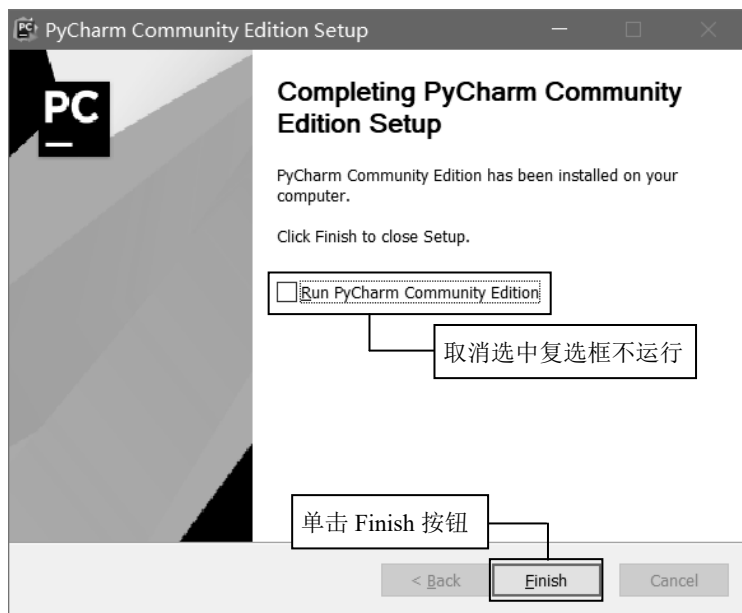


图 1.18 完成安装

1.4.3 配置 PyCharm

双击 PyCharm 桌面快捷方式，启动 PyCharm 程序。选择是否导入开发环境配置文件，这里选择不导入，单击 OK 按钮，进入阅读协议页，如图 1.19 所示。

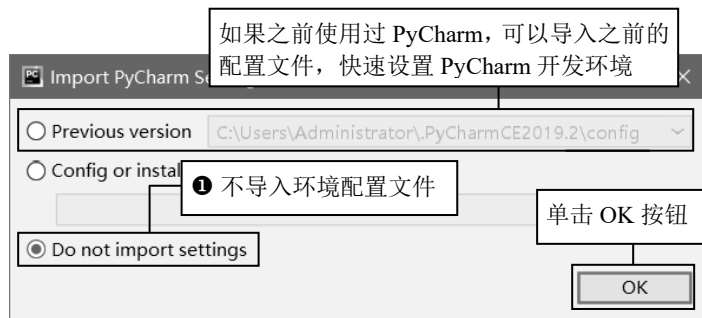


图 1.19 环境配置文件窗口

在 Set UI theme 窗口中可以根据个人需求选择开发工具的主题样式，笔者这里选中 Light，使用白色的主题颜色，然后单击 Next:Featured plugins 按钮，如图 1.20 所示。

在 Download featured plugins 窗口中，直接单击 Start using PyCharm 按钮，如图 1.21 所示，此时程序将进入欢迎界面。

进入 PyCharm 欢迎页，单击 Create New Project，创建一个新工程文件，如图 1.22 所示。

在 New Project 窗口中，首先选择工程文件保存的路径，然后单击 Create 按钮，如图 1.23 所示。

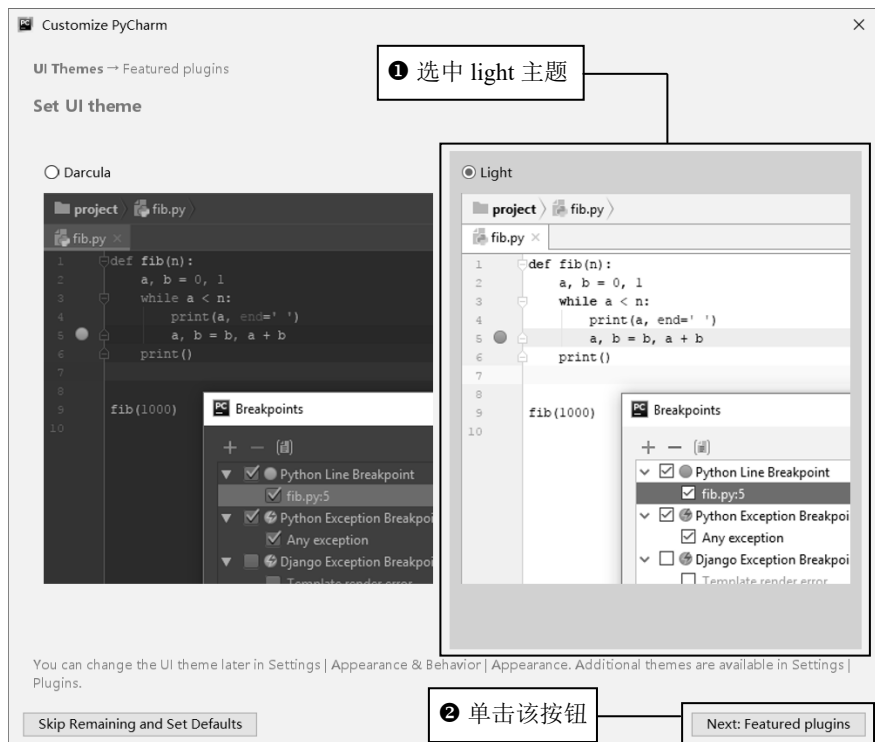


图 1.20 选择主题颜色

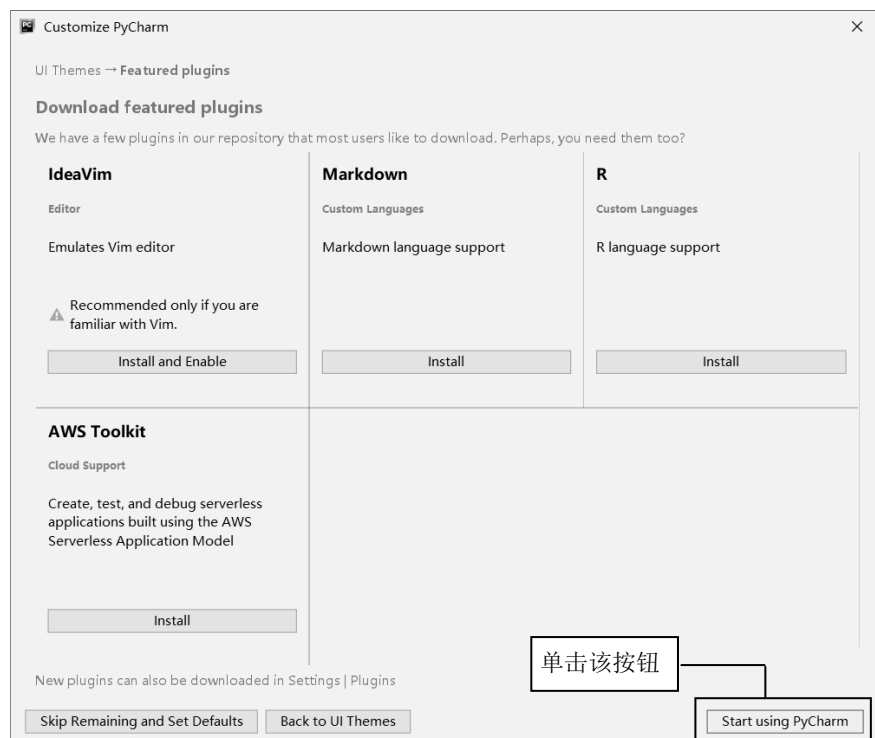


图 1.21 下载特色插件

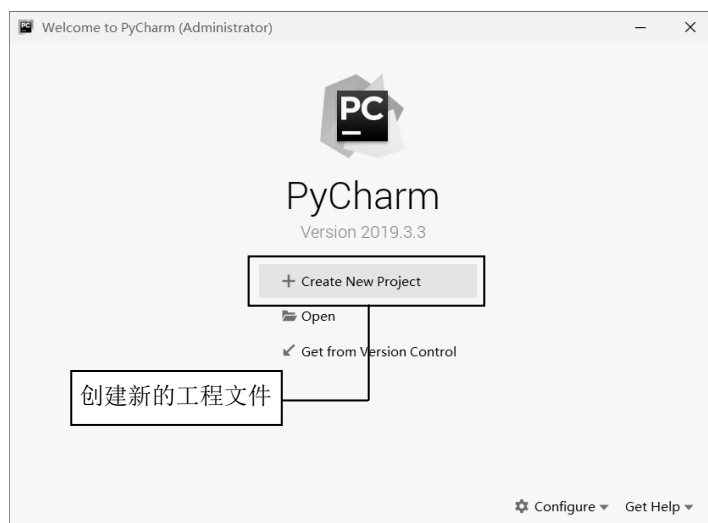


图 1.22 PyCharm 欢迎界面



图 1.23 设置 Python 存储路径

工程创建完成以后，关闭 Tip of the Day 窗口，然后依次选择 File→Settings 选项，如图 1.24 所示。

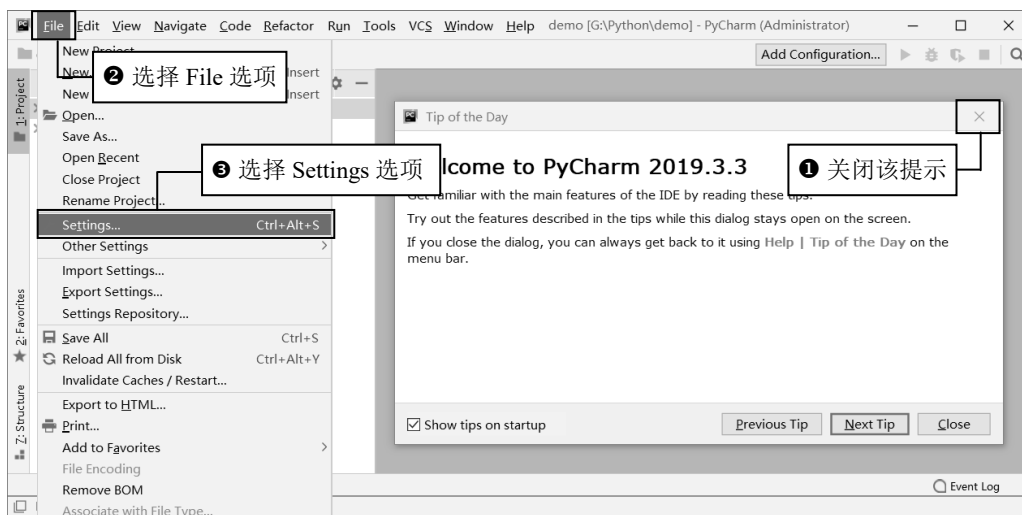


图 1.24 打开设置窗口

在 Settings 窗口中依次选择 Project:demo（demo 为自己编写的工程名称）→Project Interpreter，然后在右侧的下拉列表中选择 Show All...，将打开 Project Interpreters 窗口，如图 1.25 所示。

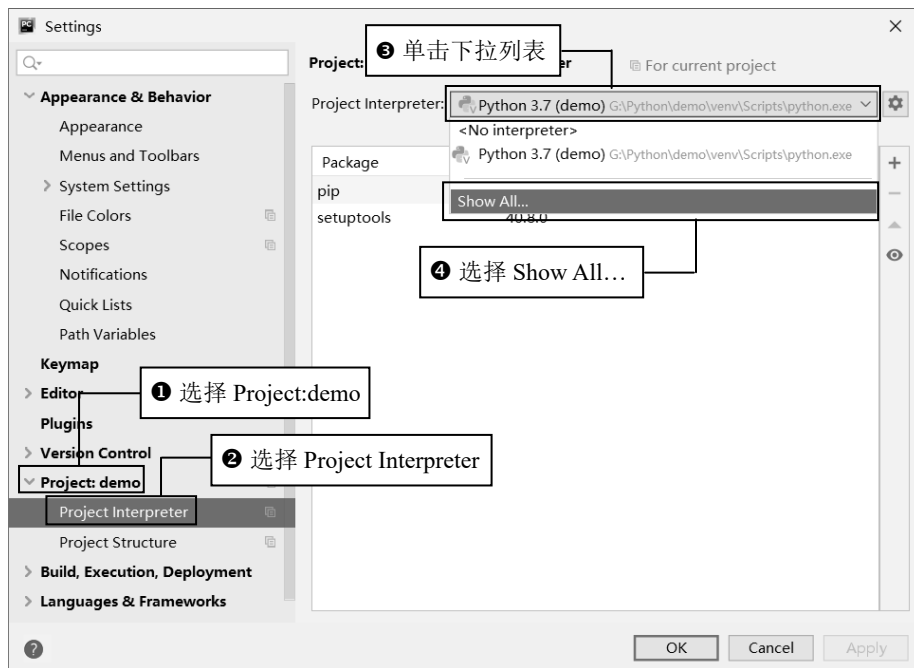


图 1.25 进入设置窗口

在 Project Interpreters 窗口中，单击右侧的“+”按钮，如图 1.26 所示。



图 1.26 单击按钮

在 Add Python Interpreter 窗口中，首先单击左侧的 System Interpreter 选项，然后在右侧的下拉列表中选择 Anaconda 中的 python.exe，最后单击 OK 按钮，如图 1.27 所示。

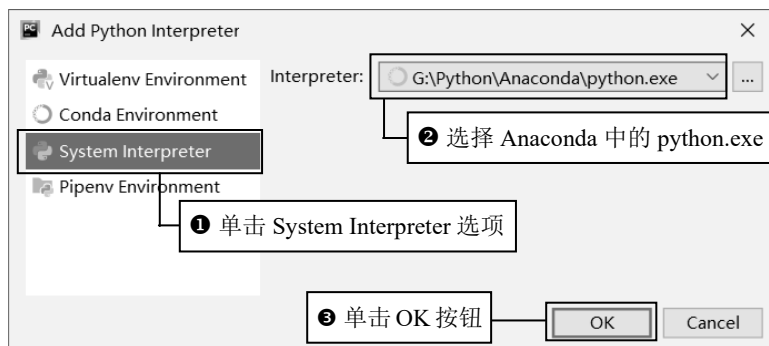


图 1.27 添加 Python 编译器

返回 Project Interpreters 窗口后，选择新添加的 Anaconda 中的 python.exe 编译器，然后单击 OK 按钮，如图 1.28 所示。

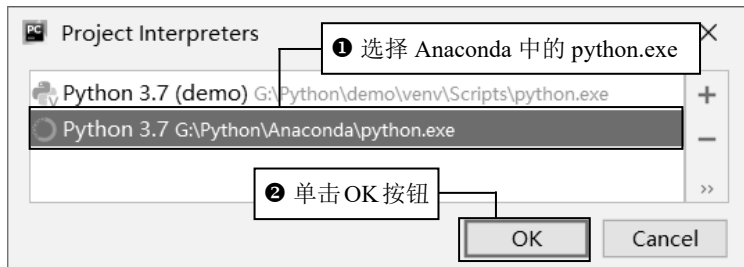


图 1.28 选择 Anaconda 中的 Python 编译器

返回 Settings 窗口，此时窗口中将自动显示出 Anaconda 内已经安装的所有 Python 模块，然后单击 OK 按钮，如图 1.29 所示。

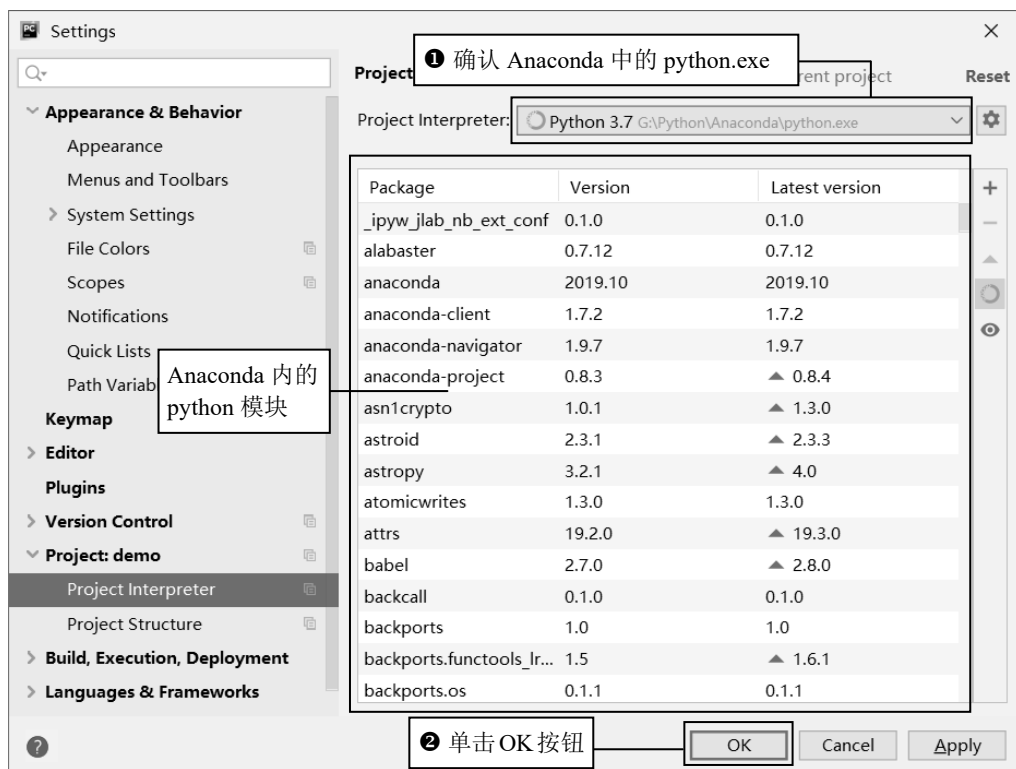


图 1.29 显示 Anaconda 内已经安装的 Python 模块

1.4.4 测试 PyCharm

右击新建好的 demo 项目，在弹出的快捷菜单中选择 New→Python File 命令（一定要选择 Python File 项，这个至关重要，否则无法后续学习），如图 1.30 所示。

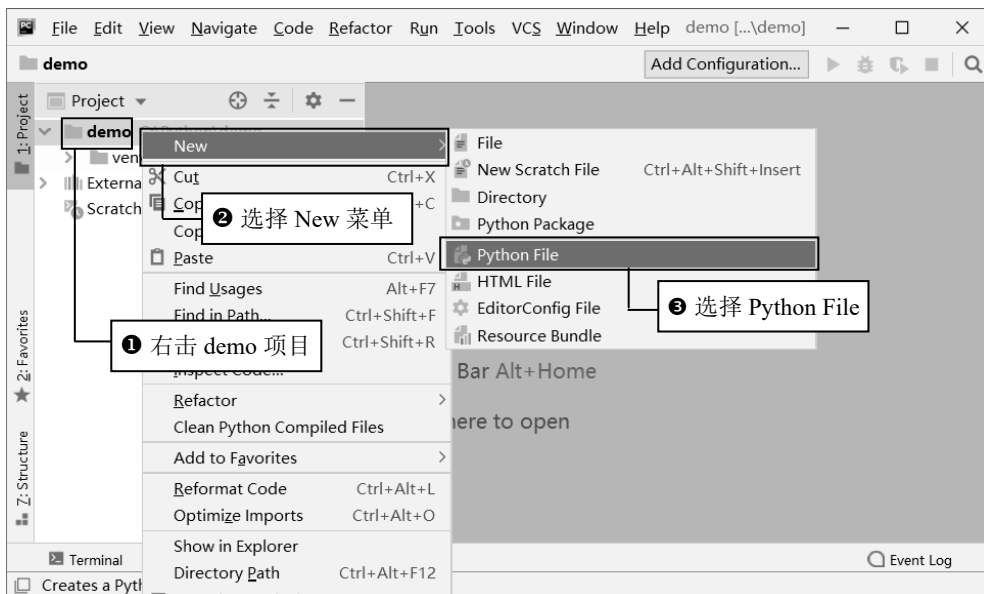


图 1.30 新建 Python 文件

在新建文件对话框输入要建立的 Python 文件名 hello world，如图 1.31 所示。随后按 Enter 键，即可完成新建 Python 文件工作。

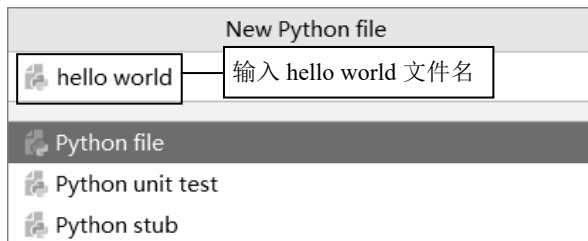


图 1.31 输入新建的 Python 文件名称

在新建文件的代码编辑区输入代码“`print ("hello world!")`”，如图 1.32 所示。

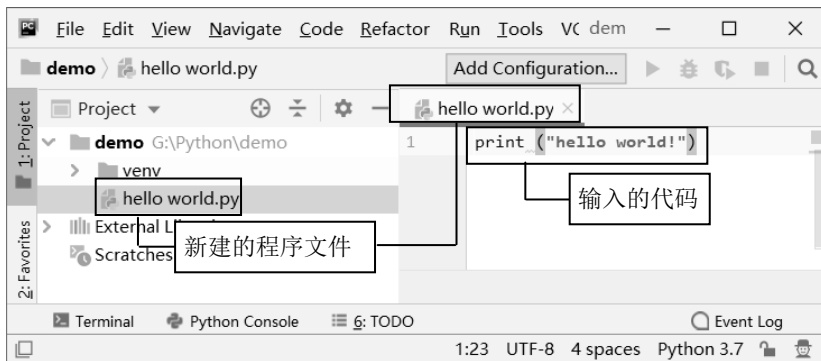


图 1.32 输入代码

在编写代码的区域右击，在弹出的快捷菜单中选择 Run 'hello world'命令，运行测试代码，如图 1.33

所示。



图 1.33 运行 Python 测试代码

如果程序代码没有错误，那么将显示运行结果，如图 1.34 所示。

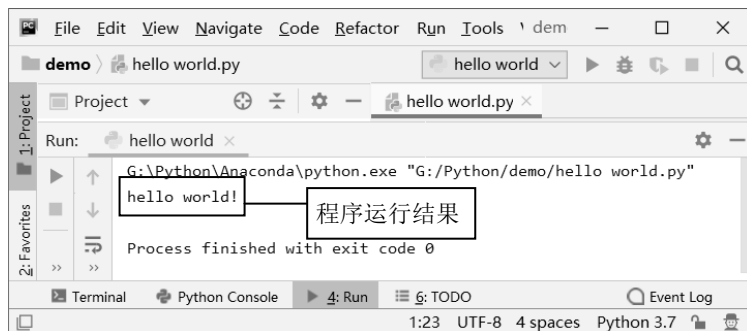


图 1.34 显示程序运行结果

1.5 小 结

本章首先介绍了什么是爬虫，然后介绍了爬虫都有哪些分类（通用爬虫、聚焦爬虫以及增量式爬虫）、爬虫的基本原理，接着学习了如何搭建爬虫的开发环境，这里推荐读者安装 **Anaconda**，这样可以避免频繁地安装很多第三方模块。为了提高开发效率，推荐读者使用 **PyCharm** 开发工具来编写爬虫程序。