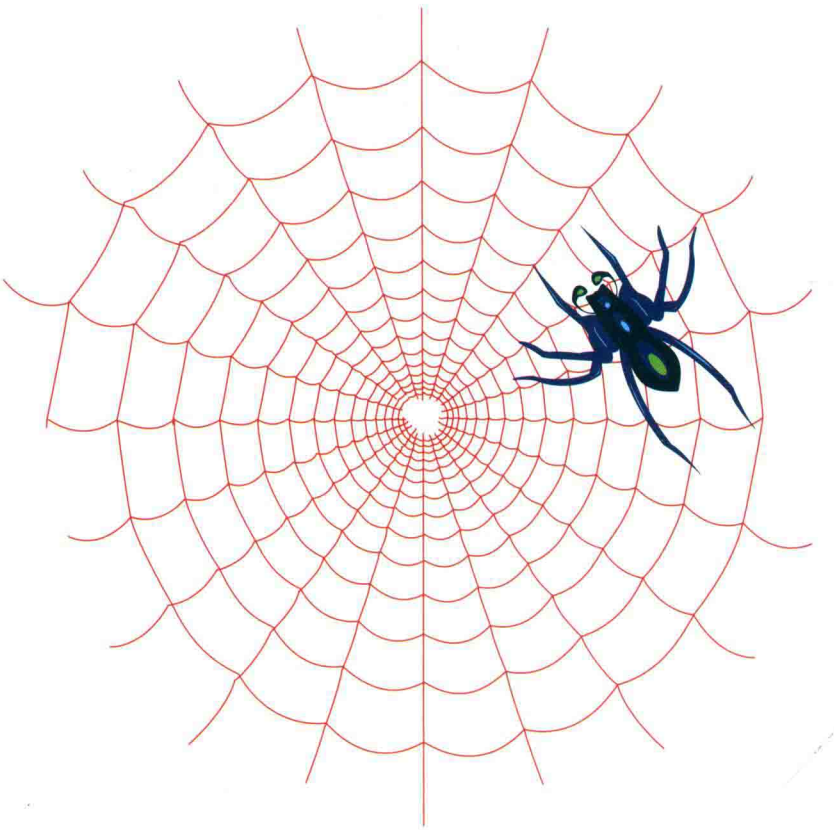
基于Python 3.7，快速掌握网络爬虫编程

- 针对想直接切入Python 3网络爬虫编程的读者
- 围绕爬虫技术、爬虫实战，提高你的数据获取能力
- 学习过程中贯穿大小示例，配合教学视频，便于读者掌握爬虫编程技巧

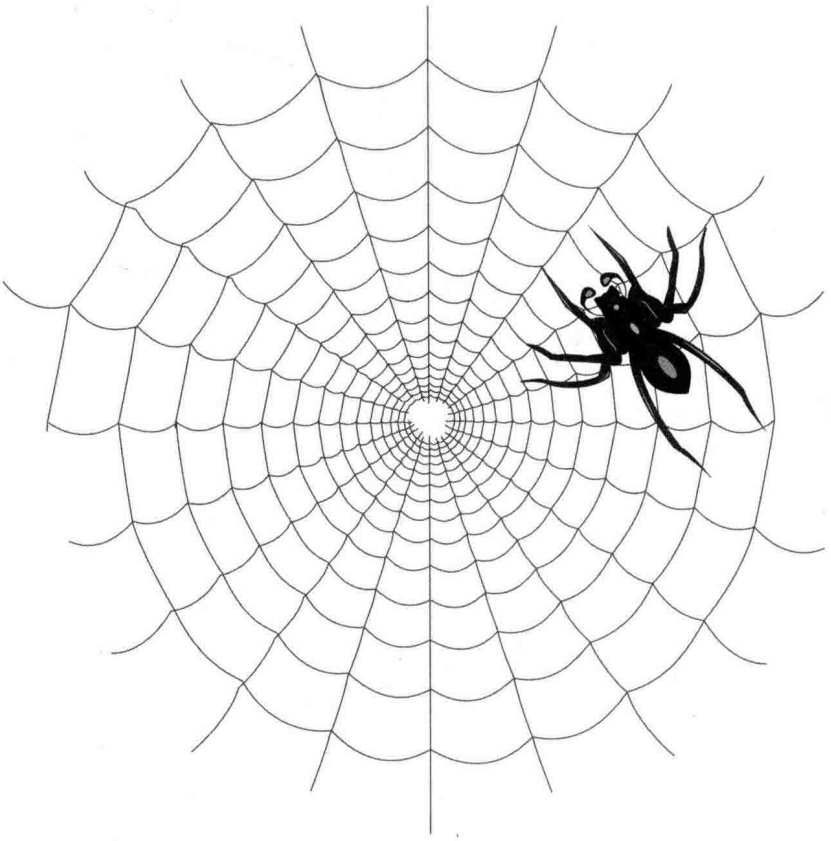


Python 3.7网络爬虫快速入门

王启明 编著

源代码及教学视频

清華大學出版社



Python 3.7网络爬虫快速入门

王启明 编著

清華大學出版社
北京

内 容 简 介

Python 3.7 正在成为目前流行的编程语言，而网络爬虫又是 Python 网络应用中的重要技术，二者的碰撞产生了巨大的火花。本书在这个背景下编写而成，详细介绍 Python 3.7 网络爬虫技术。

本书分为 11 章，分别介绍 Python 3.7 爬虫开发相关的基础知识、lxml 模块、BeautifulSoup 模块、正则表达式、文件处理、多线程爬虫、图形识别、Scrapy 框架、PyQuery 模块等。基本上每一章都配有众多小范例程序与一个大实战案例。作者还为每一章分别录制教学视频供读者自学参考。

本书内容详尽、示例丰富，是有志于学习 Python 网络爬虫技术初学者必备的参考书，同时也可作为 Python 爱好者拓宽知识领域、提升编程技术的参考书。

图书在版编目（CIP）数据

Python 3.7 网络爬虫快速入门/王启明编著. — 北京：清华大学出版社，2019
ISBN 978-7-302-53647-5

I. ①P… II. ①王… III. ①软件工具－程序设计 IV. ①TP311.561

中国版本图书馆 CIP 数据核字（2019）第 186648 号

责任编辑：夏毓彦
封面设计：王　翔
责任校对：闫秀华
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
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn

印 装 者：北京鑫海金澳胶印有限公司
经　　销：全国新华书店
开　　本：190mm×260mm　　**印　　张**：13.25　　**字　　数**：339 千字
版　　次：2019 年 10 月第 1 版　　**印　　次**：2019 年 10 月第 1 次印刷
定　　价：49.00 元

产品编号：082462-01

前 言

Python 是简练的语言

使用像 Python 这样的动态类型语言编写的代码往往比用其他主流语言编写的代码更加简短。这意味着，在编程的过程中会有更少的录入工作，而且会更容易记住算法并真正领会算法的原理。

Python 是易读的语言

Python 不时被人们指为“可执行的伪代码”。虽然很明显这是夸大之词，但是它表明大多数有经验的程序员可以读懂 Python 代码并领会代码所要表达的意图。

Python 是易安装的语言

要搭建 Python 的环境非常容易，不管是 Windows、Linux 还是 Mac 系统，只要配置好 Python 的环境，只需要 easy_install XX 或者 pip install XX 就可以安装所需要的第三方扩展包。

Python 是易扩展的语言

Python 附属了很多标准库，涉及数据函数、XML 解析以及网页下载、RSS 解析、SQLLite 等，可以解决现实中遇到的各种问题。

为什么用 Python 实现网络爬虫

基于上述优点，加上抓取网页文档的接口更简洁；相比其他动态脚本语言，如 Perl、Shell，Python 的 urllib2 包提供了较为完整的访问网页文档的 API，以及抓取后的处理方法，比如筛选 HTML 标签、提取文本等。Python 的相关扩展可以用极短的代码完成大部分文档的处理。

本书涉及的技术或框架

Python 基本语法	正则表达式	线程（Thread）
Python 函数	XML	进程（Process）
lxml 模块	CSV	图形识别验证码
XPath 语法	MySQL	Scrapy
BeautifulSoup	PyQuery	

本书涉及的范例和案例

爬取豆瓣网的内容
爬取电影天堂网的内容
爬取猫眼电影网的内容
爬取腾讯招聘网的内容
使用 BeautifulSoup 爬取电影天堂的内容
使用正则表达式爬取糗事百科的内容
爬取鼠绘漫画的图片
使用多线程爬取豆瓣的内容
使用 Tesseract 识别验证码的图片
使用 Scrapy 框架爬取豆瓣网的内容
使用 PyQuery 爬取微博热搜的内容

本书特点

（1）语言通俗易懂。对于没有基础的读者，最忌讳的就是讲一些艰深晦涩的理论，让人难于理解。本书则尽量使用通俗易懂的语言来介绍 Python，让大家更容易理解各种知识点，从而将相应的知识变成自己的能力。

（2）结合范例程序来讲解知识点。为了讲明各个知识点，基本上每个知识点都通过相关的范例程序来说明。通过范例程序及实际的执行效果，让大家学以致用，在理解领会的基础上进一步掌握相关知识、相应模块的方法。

（3）插图配合教学视频。为了保证本书的范例程序均能够成功执行，每个范例程序不仅有相应的程序代码，还有程序执行后的效果图。大家可以通过效果图来对比程序执行的结果，确保学习质量。同时每章还配有一个教学视频供读者自学参考。

（4）案例丰富。为了向读者说明 Python 爬虫程序的效果，书中选择的被爬取的网站都是国内热门的网站，比如豆瓣电影、猫眼电影、电影天堂、微博热搜等。这些网站大家喜闻乐见。通过这些案例，大家可以轻松地掌握相关模块的使用方法，举一反三，将相应技术应用于其他同类的网站中。

代码与教学视频下载

本书示例源代码与教学视频下载地址请扫描右边二维码获得。

如果下载有问题，请联系 booksaga@163.com，邮件主题为“Python 3.7 网络爬虫快速入门”。

本书读者

- 有志于学习 Python 爬虫编程的初学者
- 对 Python 网络爬虫技术有兴趣的开发人员
- 各类综合信息网站的站长或技术人员
- 高校和培训学校相关专业的师生

编　者

2019 年 7 月

目 录

第 1 章

◂ 简识Python ▸

Python 语言是一种面向对象的计算机程序设计语言，一经发行便受到众多计算机开发者的喜爱。经过多年的完善补充，Python 语言越发显现出它的优点，成为大数据技术人员热衷学习研究的热门语言。Python 简洁清晰的编程风格，易于和计算机其他应用领域巧妙结合，使得 Python 适应面非常广。

读者通过本章的学习，可以初步了解 Python 的语言魅力，以及如何将 Python 用来解决一些数学上的小问题。但是要想完全掌握好一门语言，则需要不断地使用，同时还要在使用过程中不断地积累编程技巧，最后达到融会贯通。

本章主要涉及的知识点有：

- 认识 Python：Python 是一种解释型、交互式、面向对象、对初学者友好的语言
- 编程：通过编程实现和计算机的对话，用 Python 编程实现一些简单的功能
- 了解主流的开发环境：借助完善的开发环境进行编程，事半功倍
- Hello，World！：编写第一个小程序，并学会处理程序中可能出现的问题

1.1 了解 Python

本节首先介绍 Python 的基本概念，这些概念是学习和使用 Python 编程的前提。理解了 Python 的这些基础概念之后才能为学好 Python 编程打下坚实的基础。

1.1.1 Python 的概念

首先来看看 Python 的定义。Python 是一种解释型、交互式、面向对象的程序设计语言，也是对初学者友好的一种程序设计语言。

- Python 是解释型语言

Python 作为一种解释型语言，意味着在开发过程中可以没有编译这个环节，相较于 C 语言这门中级程序设计语言而言，Python 是一种高级程序设计语言，编程者运用起来更加容易

理解和便捷。

- Python 是交互式语言

这意味着，开发者在开发环境中写入代码，即刻就可以得到回馈的结果。在这个过程中，程序无须先进行整体的编译处理，而是逐条执行程序语句给出运行的结果。

- Python 是面向对象的程序设计语言

这意味着 Python 支持将代码封装成对象的面向对象的程序设计方式，使得 Python 程序能够充分发挥面向对象程序设计技术的长处。

- Python 是对初学者友好的语言

对初学者来说，Python 语言清晰简洁的编程方式，对语法的要求比其他语言更加宽松且具有丰富的扩展功能，便于初学编程者学习和掌握，即学习曲线更短。

1.1.2 有趣的 Python 程序

在深入学习 Python 语言之前，先来感受一下 Python 语言实现的一些有趣的小程序。

【范例程序 1-1】在屏幕上画一条蛇（Python 这单词的英文意思就是蟒蛇）

范例程序 1-1 的代码

```
import turtle          # 导入 turtle 包
turtle.setup(650,350,200,200)
turtle.penup()         # 提起笔移动，不绘图
turtle.fd(-250)        # 前进
turtle.pendown()       # 边移动边绘图
turtle.pensize(25)
turtle.pencolor("green")
turtle.seth(-40)
for I in range(3):
      turtle.circle(40,80)
      turtle.circle(-40,80)
turtle.circle(40,80/2)
turtle.fd(10)
turtle.circle(50,100)
turtle.circle(-25,130)
turtle.fd(20*6/5)
turtle.done()
```

执行以上代码，结果如图 1.1 所示。相信大多数读者通过运行这个范例程序代码，已经看到了这条可爱的小青蛇在屏幕上一点点画出来的样子，是不是很有趣呢？

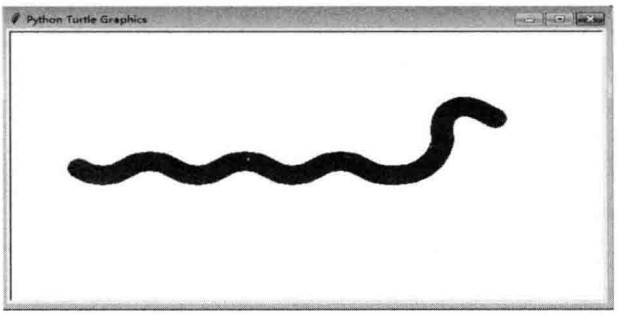


图 1.1　画蛇

同时，有些读者可能会对这个范例程序到底写了什么感到好奇。不用着急，请读者循序渐进、一点一点地弄明白这个范例程序代码中各种符号代表的含义。通过本书后面章节的学习，大家很快就可以自己轻松地画出想要的图形了。

Python 除了能够实现在屏幕上画图之外，还可以轻松解决一些数学问题。

【范例程序 1-2】实现简单的温度转换

目前有两种度量温度的标准，分别是摄氏度和华氏度，这个范例程序可以通过简单地输入一种温度值，然后经过换算得到另一种温度值。

范例程序 1-2 的代码

```
Temp = input("请输入带有符号的温度值：")
if Temp[-1]in['F','f']:
    value = (eval(Temp[0:-1]) - 32 )/1.8
    print("转换后的温度值为{:.2f}C".format(value))
elif Temp[-1]in['C','c']:
    value = eval(Temp[0:-1])*1.8 +32
    print("转换后的温度值为{:.2f}F".format(value))
else:
    print("输入温度值格式有误")
```

尝试一下将这些程序代码复制到你的开发环境中，运行一下这个范例程序，试一试温度值的转换。范例程序 1-2 的运行结果如图 1.2 与图 1.3 所示。

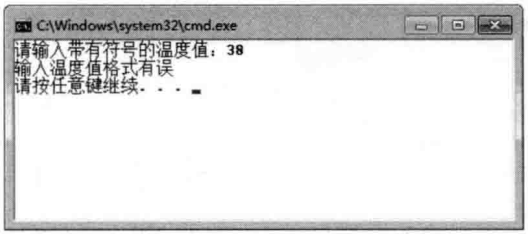


图 1.2　实现温度转换 I

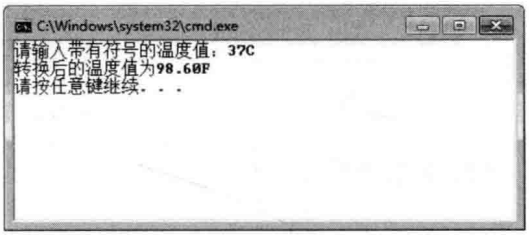


图 1.3　实现温度转换 II

有些读者可能还没有一个可以运行 Python 程序的集成开发环境，没关系，我们在下一节就指导读者安装和建立一个 Python 的集成开发环境。

1.2 集成开发环境

在上一节，大家看到了两个用 Python 语言编写的范例程序，心中一定充满了好奇，到底 Python 是怎么做到的呢？从本节开始，我们就正式开始 Python 的学习。俗话说，“工欲善其事，必先利其器。”所以在学习之前，我们需要了解一下 Python 的集成开发环境。

什么是集成开发环境？又有哪些可供选择的 Python 集成开发环境呢？

首先，集成开发环境就像画家手中的纸笔、厨师所用的炊具、木匠常用的工具，是开发程序必备的设施。设想一下，如果没了集成开发环境，那么程序要在哪里编写、执行和调试呢？

Python 的集成开发环境有很多，人们可以根据不同的需求来选择使用不同的集成开发环境。对于初学者来说，最适合的集成开发环境当然 Python 官网提供的。

1.2.1 安装 Python 3.7

下面先来安装 Python 3.7。

首先，读者需要知道自己所用的计算机配备的是什么操作系统。

Python 是一种跨平台的语言，可以在 Windows、Mac OS 以及各种 Linux 操作系统上运行。目前，人们在计算机中使用最多的还是微软公司的 Windows 操作系统，因此下面我们主要介绍如何在 Windows 上一步步安装 Python。

（1）下载 Python 安装包。首先使用浏览器打开 Python 官网，网址为 http://www.python.org/downloads/），如图 1.4 所示。然后选择相应的安装包下载即可。

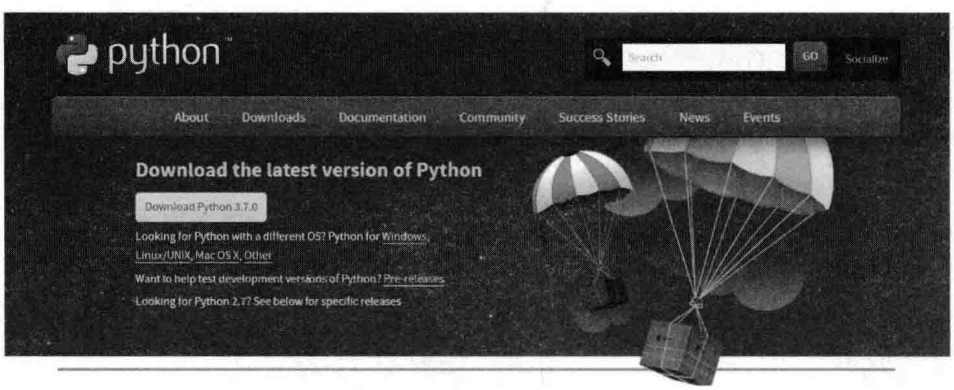


图 1.4 Python 官网

（2）安装下载的 Python 3.7.2 安装包。执行安装包，出现类似图 1.5 所示的安装界面。

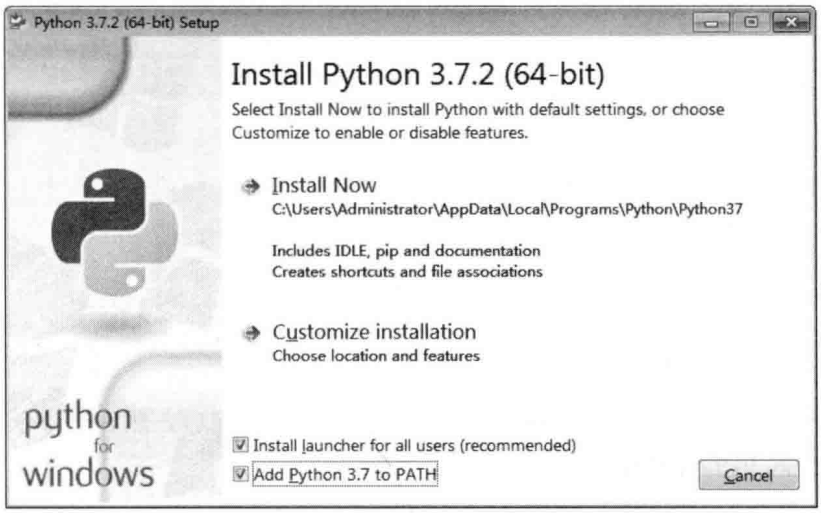


图 1.5 安装 Python 步骤 1

（3）选择图 1.5 中的“Install Now”（开始安装）或者“Customize installation”（自定义安装）选项。在开始安装前需要先勾选“Add Python 3.7 to PATH”（将与运行 Python 3.7 有关的路径加入到系统环境变量 Path 中）选项。之后将出现如图 1.6 所示的安装界面。

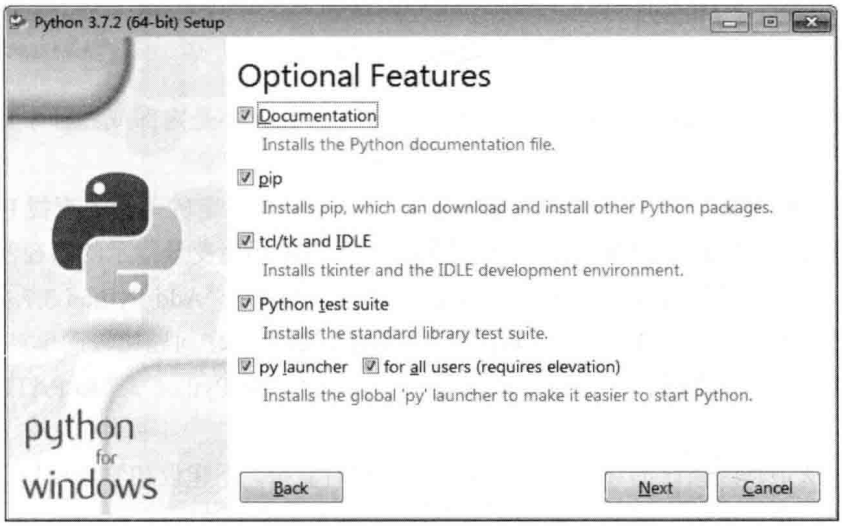


图 1.6　安装 Python 步骤 2

（4）接下来单击【Next】按钮，直到完成。

（5）测试安装是否成功。安装完毕之后需要对安装结果进行测试。首先打开系统的运行窗口，输入 cmd，如图 1.7 所示。

图 1.7　运行窗口

（6）输入完成并单击【确定】按钮，之后会进入命令行窗口。然后在命令行窗口中输入“python”，进入 Python 开发环境，如图 1.8 所示。

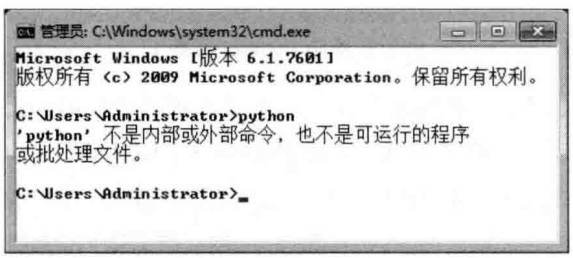


图 1.8　在命令行窗口中输入“python”

（7）查看图 1.8，可以发现这里出现一个错误：‘python’不是内部或外部命令，也不是可运行的程序。

这是为什么呢？原因是 Windows 会根据安装时环境变量设定的路径去查找 Python 程序。出现图 1.8 所示的错误是因为 Windows 通过系统的 PATH 环境变量指示的路径没有找到这个程序。回忆一下，安装过程中是否漏掉了需要勾选的一个选项“Add Python 3.7 to PATH”。这时需要手动把安装的路径添加到 PATH 环境变量中。解决这个问题比较简单：

第一种是最简单的做法，即重新启动安装程序，将“Add Python 3.7 to PATH”选项勾选上。

第二种方法是修改系统的环境变量。这种方法相对复杂，不建议初学者使用。如果还是想尝试这种方法，建议读者去网络上查询修改 Path 环境变量的方法。

如果读者能够顺利地完成上述安装步骤，那么恭喜读者已经获得了一把“趁手的武器”。接下来，就可以借助这个 Python 开发环境来编写 Python 程序了。

1.2.2　从 IDLE 启动 Python

启动 Python 系统自带的、简洁的集成开发环境 IDLE。

1. 通过运行 cmd 在命令行窗口中启动 IDLE

首先，找到 Windows 的“运行”选项，在运行框中输入“cmd”命令，启动命令行窗口，如图 1.9 所示。