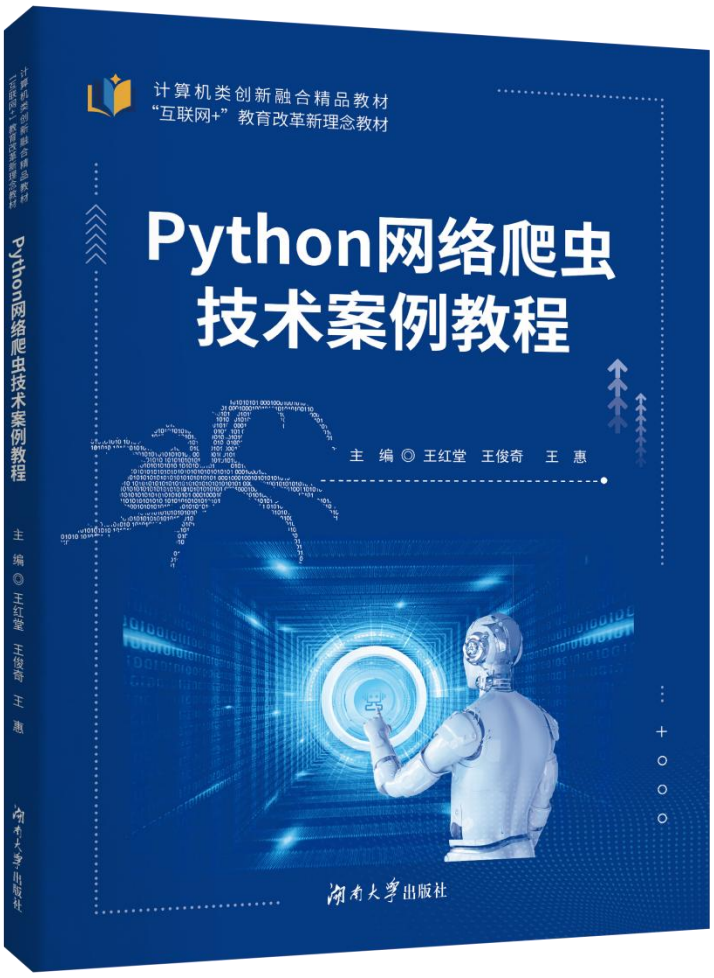
python 网络爬虫技术案例教程



类目：计算机类
书名：python 网络爬虫技术案例教程
主编：王红堂 王俊奇 王 惠
出版社：湖南大学出版社
开本：大 16 开
书号：978-7-5667-4853-9
使用层次：通用
出版时间：2026 年 7 月
定价：49.80 元
印刷方式：双色
是否有资源：有



Python网络爬虫技术案例教程

责任编辑：黄　旺
封面设计：旗语书装

ISBN978-7-5667-1853-9
定价：49.80元

计算机类创新融合精品教材
"互联网+"教育改革新理念教材

Python网络爬虫技术案例教程

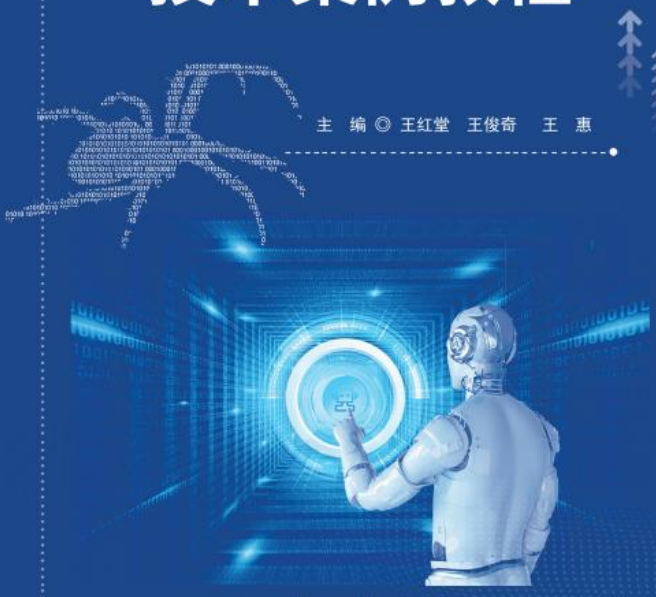
主　编◎王红堂　王俊奇　王　惠

湖南大学出版社

计算机类创新融合精品教材
"互联网+"教育改革新理念教材

Python网络爬虫技术案例教程

主　编◎王红堂　王俊奇　王　惠



湖南大学出版社

计算机类创新融合精品教材
“互联网 +”教育改革新理念教材

Python 网络爬虫技术案例教程

主　编 ◎ 王红堂　王俊奇　王　惠
副主编 ◎ 张娜娜　魏晓辉　王雨晴
张巧珍　李明宇



湖南大学出版社
·长 沙·

图书在版编目（CIP）数据

Python 网络爬虫技术案例教程 / 王红堂，王俊奇，王惠主编 . -- 长沙：湖南大学出版社，2026.7. -- ISBN 978-7-5667-4853-9

Ⅰ. TP312.8

中国国家版本馆 CIP 数据核字第 2026NL1463 号

Python 网络爬虫技术案例教程

PYTHON WANGLUO PACHONG JISHU ANLI JIAOCHENG

主　　编：王红堂　王俊奇　王　惠

责任编辑：黄　旺

印　　装：河北龙大印务有限公司

开　　本：889 mm×1194 mm　1/16　**印　　张：**14　**字　　数：**393 千字

版　　次：2026 年 7 月第 1 版　**印　　次：**2026 年 7 月第 1 次印刷

书　　号：ISBN 978-7-5667-4853-9

定　　价：49.80 元

出 版 人：李文邦

出版发行：湖南大学出版社

社　　址：湖南·长沙·岳麓山　**邮　　编：**410082

电　　话：0731-88822559（营销部）　88821174（编辑室）　88821006（出版部）

传　　真：0731-88822264（总编室）

网　　址：http://press.hnu.edu.cn

电子邮箱：xiaoshulianwenhua@163.com

版权所有,盗版必究

图书凡有印装差错,请与营销部联系

前言
PREFACE

“十五五”规划建议指出，“健全数据要素基础制度，建设开放共享安全的全国一体化数据市场，深化数据资源开发利用。促进实体经济和数字经济深度融合，实施工业互联网创新发展工程。”在数据驱动的时代，网络作为全球最大的信息载体，蕴藏着海量有价值的资源，而网络爬虫技术，正是解锁这份资源宝库的核心钥匙。从互联网内容聚合、市场舆情分析，到科研数据采集、商业决策支撑，爬虫技术已渗透到各行各业，成为信息技术领域不可或缺的实用技能，也成为许多学习者踏入数据领域的入门必修课。

然而，网络爬虫技术的学习之路并非坦途。初学者往往会面临诸多困惑：如何快速掌握爬虫的核心原理与流程？如何规避法律风险，做到合规爬取？如何选择合适的开发工具与技术栈，高效实现数据爬取、解析与存储？如何应对反爬机制，突破爬取瓶颈？如何将爬取的数据转化为可视化成果，发挥数据的实际价值？为解答这些疑问，帮助学习者系统、高效地掌握网络爬虫技术，我们结合多年的实战经验与教学心得编写了本书。

本书从初学者角度出发，通过通俗易懂的语言、丰富的实例，详细介绍了使用 Python 实现网络爬虫开发应掌握的核心技术。全书共分 10 个项目，内容涵盖网络爬虫技术、Web 前端基础、Requests 库的使用、网页解析的核心技术与方法、IP 代理、异步爬虫、分布式爬虫技术、数据清洗技术、数据存储、数据可视化等，内容层层递进、环环相扣，引导学习者逐步掌握爬虫技术的核心能力。

本书具有以下特色：

1. 项目驱动，实战赋能

全书采用“项目—任务”的架构，每个项目聚焦一个核心能力领域，任务源自真实爬虫应用场景，让学习者在解决实际问题的过程中掌握爬虫技能，实现“学以致用”。

2. 结构精巧，体系完整

本书精心设置了 10 个核心项目，每个项目拆解为多个具体任务，构建“入门—实战—进阶—应用”的完整学习闭环，涵盖爬虫全流程知识。从基础概念、环境搭建到前端铺垫，再到核心爬取、反爬突破，最终落地数据处理与可视化，模块衔接流畅、无知识断层，既提升学习的针对性，又助力学习者构建系统的知识体系。

3. 合规引领，规范养成

书中贯彻网络爬虫合规开发理念——如专门剖析爬虫合法性边界、明确 IP 代理合理使用场景、引导规避爬取禁忌，同时融入规范的开发流程，帮助初学者从一开始就建立合规、规范、可落地的爬虫开发思维，避免因技术滥用引发风险。

4. 核心聚焦，梯度适配

全书覆盖网络爬虫的核心领域——基础预备、爬取工具、解析方法、反爬技术、分布式架构、数据处理与存储、可视化呈现，既保证知识广度，又不忽视核心知识点的深度。同时兼顾零基础与进阶学习者需求，基础内容降低入门门槛，进阶内容聚焦企业级应用难点，实现深度与广度、基础与进阶的双重兼顾。

学习爬虫技术，不仅需要掌握相关的技术知识，更需要坚持实践、勇于探索。希望读者在阅读本书的过程中，能够结合书中的案例与技巧，多动手、多尝试，将理论知识转化为实战能力，逐步成长为具备独立开发爬虫能力的学习者。本书适合相关专业学生作为教材，也可作为行业相关工作者提升工作技能的参考书。同时，由于技术更新迭代较快，书中难免存在疏漏与不足，恳请广大读者批评指正，我们将不断完善，为大家提供更优质的学习内容。

编者

目 录

CONTENTS

项目 1

网络爬虫技术 …… 1

任务 1.1 揭开面纱：基本概念解析 …… 2
任务 1.2 步步为营：网络爬虫的标准流程 …… 10
任务 1.3 法律边缘：网络爬虫的合法性问题剖析 …… 11
任务 1.4 工具选择：开发语言与开发环境的选择 …… 13

项目 2

Web 前端基础 …… 25

任务 2.1 网络通信：HTTP 基本原理详解 …… 26
任务 2.2 网页架构：HTML 语言精髓 …… 31
任务 2.3 颜值担当：CSS 层叠样式表的奥秘 …… 36
任务 2.4 锦上添花：动态脚本语言的魅力 …… 40

项目 3

Requests 库的使用 …… 46

任务 3.1 从零开始：安装 Requests 库的正确姿势 …… 47
任务 3.2 精准出击：使用 Requests 库发送请求的技巧 …… 50
任务 3.3 洞察细节：查看 Requests 库响应内容的方法 …… 52

任务 3.4　定制专属：定制请求头部 Headers 的策略 …… 56
任务 3.5　前端调试：夸克浏览器开发者工具面板探秘 …… 58

项目 4
网页解析的核心技术与方法 …… 70

任务 4.1　解析利器：lxml 库的高效应用 …… 71
任务 4.2　简单易用：Beautifulsoup4 库的解析之道 …… 76
任务 4.3　文本匹配：正则表达式的强大威力 …… 83
任务 4.4　数据留存：存储数据至文件的实用技巧 …… 88

项目 5
IP 代理 …… 94

任务 5.1　隐藏身份：IP 代理的关键作用解析 …… 95
任务 5.2　得心应手：IP 代理的正确使用方法 …… 97
任务 5.3　资源储备：搭建 IP 池的实用策略 …… 99
任务 5.4　专业之选：付费 IP 代理的使用指南 …… 103

项目 6
异步爬虫 …… 107

任务 6.1　概念先行：异步爬虫的基本概念与优势 …… 108
任务 6.2　循序渐进：串行下载多个视频的基础实践 …… 111
任务 6.3　并行加速：使用线程池下载多个视频的高效方法 …… 112
任务 6.4　轻量并发：使用协程下载多个视频的进阶技巧 …… 115

项目 7
分布式爬虫技术 122
任务 7.1 分布奥秘：分布式爬虫的原理与优势 123
任务 7.2 强强联合：Scrapy-Redis 分布式爬虫的实现 127
任务 7.3 便捷部署：使用 Scrapyd 部署分布式爬虫的方法 135

项目 8
数据清洗技术 143
任务 8.1 流程把控：数据分析的完整流程概述 144
任务 8.2 去伪存真：数据清洗的概念与详细流程 146
任务 8.3 实用技法：数据清洗的常用方法与技巧 147

项目 9
数据存储 163
任务 9.1 简单存储：文件的存取方法与技巧 164
任务 9.2 轻量存储：SQLite 数据库的应用 172
任务 9.3 数据存储：MySQL 数据库的管理与使用 175

项目 10
数据可视化 187
任务 10.1 可视利器：Matplotlib 的功能与概述 188
任务 10.2 细节把控：图表的常用设置与优化 190
任务 10.3 多样呈现：常用图表的绘制方法与技巧 202

参考文献 216

项目 1

网络爬虫技术

项目导读

本项目聚焦网络爬虫技术，系统介绍其基本原理及在大数据时代的重要作用。内容涵盖爬虫的基本概念、主要类型、标准工作流程，并强调遵守 Robots 协议等合法性问题。同时，项目梳理了开展爬虫开发所需的预备知识，包括 URL、HTTP/HTTPS 协议、HTML 结构等，并推荐使用 Python 语言及其 Requests、Selenium 等第三方库进行高效开发，为后续实践奠定理论基础。

学习目标

知识目标

1. 了解大数据时代数据来源的多样性。
2. 掌握网络爬虫的基本概念、类型和工作原理。
3. 熟悉网络爬虫的标准流程。
4. 理解网络爬虫的合法性问题。
5. 掌握网络爬虫的必备预备知识。

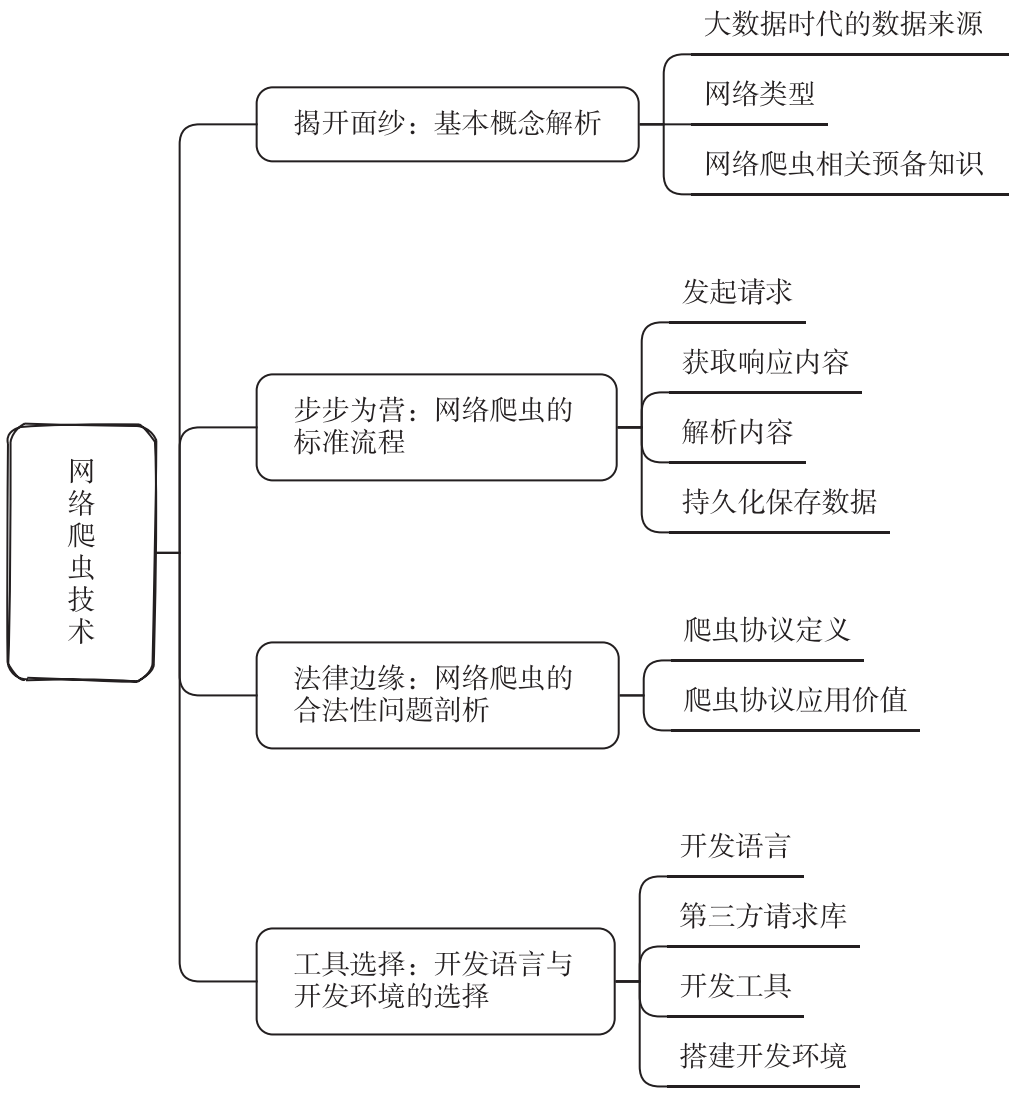
技能目标

1. 能够使用 Python 编程语言编写简单的网络爬虫程序。
2. 能够使用 Requests 库和 Selenium 库进行网页请求和解析。
3. 能够使用正则表达式和网页解析库提取网页中的数据。
4. 能够将爬取到的数据保存到文本文件、数据库或特定格式的文件中。

素质目标

1. 培养学生获取信息、分析问题和解决问题的能力。
2. 增强学生的逻辑思维能力和编程能力。

知识图谱



任务 1.1 揭开面纱：基本概念解析

任务描述

在大数据时代浪潮下，某科技公司数据部门迎来了一位名叫小林的实习生。为了帮助小林快速融入并掌握核心技能，部门资深工程师陈工决定带领他从零开始，系统性地学习和实践网络爬虫技术。陈工将小林叫到白板前，开始了他的第一课："小林，在动手写代码之前，我们必须先弄清楚'我们到底要做什么'以及'为什么这么做'。今天你的任务是，全面了解网络爬虫在大数据生态中的角色、它的不同类型以及核心价值。你需要搞明白，当我们在谈论爬虫时，我们究竟在谈论什么。"

任务分析

本任务是整个项目的基石，为小林建立宏观认知。陈工引导小林认识到，数据是新时代的石油，而爬虫是重要的开采工具之一。他们一起分析了数据的四大来源，并重点探讨了在无法通过 API 或数据交易市场获取数据时，爬虫作为自主采集手段的必要性。接着，陈工详细拆解了四种爬虫类型，并用搜索引擎作为通用爬虫的实例，让小林理解不同业务场景对爬虫策略的选择。这一步确保了小林后续的所有操作都建立在清晰的目标和正确的理念之上，避免了盲目编码。

相关知识

在云计算与大数据技术蓬勃发展的当下，大数据分析技术已在各行业得到广泛应用，数据的采集与获取作为大数据分析的初始环节愈发重要。越来越多的数据搜索与存储，对搜索引擎的性能提出了更为严苛的要求。网络爬虫作为用户从互联网上高效获取网络数据的关键工具，是搜索引擎不可或缺的组成部分，它直接决定了整个系统内容的丰富程度以及信息的即时性与时效性。

1.1.1　大数据时代的数据来源

大数据时代的数据主要来源于以下几个方面。

（1）企业产生的用户数据，如百度指数平台、新浪微博指数平台、网易指数平台等所积累的数据。

（2）通过数据交易平台购买的数据，例如淘数据、数据堂、数多多交易平台、中关村数海大数据交易平台等所提供的数据资源。

（3）政府或机构公开的数据，包括中华人民共和国国家统计局发布的数据、世界银行公开的数据以及联合国公布的数据等。

（4）数据管理咨询公司所掌握的数据，如埃森哲（中国）有限公司、科尔尼（上海）企业咨询有限公司、IBM 咨询等所拥有的数据资源。

倘若在数据交易市场上未能寻得所需数据，或者出于成本考虑不愿购买，那么网络爬虫便是一个理想的选择。

网络爬虫，又称网络蜘蛛、网络蚂蚁、网络机器人等，是按预设规则自动从互联网爬取信息或数据的程序或脚本。它会扫描并爬取目标页面特定信息，直至处理完所有能正常打开的网页。互联网的快速发展催生了爬取和提交网络信息的需求，推动了网络爬虫等应用的诞生。在互联网搜索中，网络爬虫可帮我们获取所需信息，尤其在访问大量非结构化数据时，它是爬取数据的得力助手。网络爬虫能替代人工自动完成数据信息的采集与整理。而且，随着大数据时代到来，网络爬虫在互联网中的地位将更加重要。

从原则上讲，只要是浏览器（即客户端）能够处理的事务，爬虫同样能够胜任。利用网络爬虫对数据信息进行自动采集，可广泛应用于以下领域：搜索引擎中对网站进行爬取与收录、数据分析与挖掘中对数据的采集、金融分析中对金融数据的采集、舆情监测与分析以及目标客户数据的收集等。

1.1.2　网络类型

根据系统结构和运作原理的不同，网络爬虫大致可分为四种类型：通用网络爬虫、聚焦网络爬虫、

增量式网络爬虫和深层网络爬虫。我们常用的搜索引擎，如 Baidu、Yahoo 和 Google 等，本质上都是一种大型复杂的网络爬虫，属于通用网络爬虫的范畴。



什么是网络爬虫

1. 通用网络爬虫

通用网络爬虫又称为全网爬虫，其爬取的对象由一批种子 URL 扩充至整个 Web，主要由搜索引擎或大型 Web 服务提供商使用。这类爬虫爬取的范围和数量都非常大，对于爬取速度及存储空间的要求都比较高，而对于爬取网页的顺序要求比较低，通常采用并行工作的方式来应对大量待爬取网页。

通用网络爬虫比较适合用于搜索引擎去搜索广泛的主题。

2. 聚焦网络爬虫

聚焦网络爬虫又称为主题网络爬虫，其最大的特点是选择性地爬取与预设主题相关的网页。与通用网络爬虫相比，聚焦爬虫只爬取与主题相关的网页，极大地节省硬件及网络资源，能更快地更新保存的页面，更好地满足特定人群对特定领域信息的需求。

3. 增量式网络爬虫

增量式网络爬虫的核心工作机制为仅对已爬取网页执行增量式更新操作，或仅定向爬取新生成的网页与内容发生变更的网页。该机制可在一定程度上确保爬取所得网页内容的时效性。

相较于周期性爬虫与页面刷新式爬虫，增量式网络爬虫的显著特征为按需爬取——仅在必要时对新生成或内容更新的网页发起爬取请求，对内容无变动的网页则不重复爬取。这一特性既可有效降低数据爬取总量，又能实现已爬取网页内容的及时更新，进而显著减少时间成本与存储空间的无效消耗。需要注意的是，此类爬虫的算法逻辑设计与工程实现难度相对较高。

4. 深层网络爬虫

网页按存在方式可分为表层网页和深层网页。表层网页指传统搜索引擎可索引的页面，以通过超链接即可到达的静态网页为主。深层网页则指大部分内容隐藏在搜索表单之后，需用户提交关键词才能动态生成的网页，例如学术数据库的检索结果页、电商平台的商品筛选页等。深层网页蕴含的信息量可达表层网页的数百倍，是当前互联网上增长最快、规模最大的新型信息资源。

为有效获取这类资源，深层网络爬虫需配备一套专门处理表单交互与页面解析的机制，其核心组件包括：URL 列表，存放待抓取的深层网页入口地址；LVS 列表，即标签 / 数值集合，为表单填充提供数据源；爬行控制器，负责整体抓取流程的调度；解析器，完成网页内容的解析；LVS 控制器，管理标签 / 数值集合的选取与使用；表单分析器，提取网页中的表单结构及字段信息；表单处理器，将数据填充至表单并完成提交；响应分析器，对提交后返回的页面进行解析，从中提取新的 URL 供后续爬取。

深层网络爬虫的表单自动填写主要采用两类策略。一是基于领域知识的表单填写，通过预先构建特定领域的关键词库，借助语义分析选择合适的关键词，与表单字段匹配后填入。二是基于网页结构分析的表单填写，多用于领域知识有限的情形，通过分析网页的 DOM 结构、表单元素布局及属性等信息，自动推断字段语义并完成填充。

知识拓展

聚焦网络爬虫中的主题筛选算法与网页评价机制

在聚焦网络爬虫（主题网络爬虫）的应用中，核心难点在于如何精准判断一个网页是否与预设

主题相关，并决定爬取的优先级。教材可进一步拓展介绍网页评价算法，例如基于关键词匹配的简单方法，以及更复杂的基于链接分析的算法（如 PageRank、HITS 算法）和基于机器学习的分类算法。通过引入主题过滤和相似度计算，爬虫能够动态调整抓取策略，优先爬取主题相关度高且质量好的网页，从而在节省资源的同时提高数据采集的精准度。

1.1.3 网络爬虫相关预备知识

网络爬虫是通过模拟人类在浏览器中的操作以获取有效信息的技术手段，其核心是借助 HTTP 协议从网站中提取数据的过程。因此，要掌握网络爬虫技术，需先了解 Web 前端、Web 站点、HTTP 协议等基础概念。

一个基本的爬虫通常包含数据采集（即网页下载）、数据处理（即网页解析）和数据存储（即将有用信息持久化）三部分内容。网络爬虫的流程，如图 1-1 所示，首先向起始 URL 发送请求并获取响应，随后对响应内容进行提取；若提取到新的 URL，则继续发送请求以获取响应；若提取到数据，则将数据进行持久化保存。

下面对网络爬虫的预备知识做一些简单的补充。

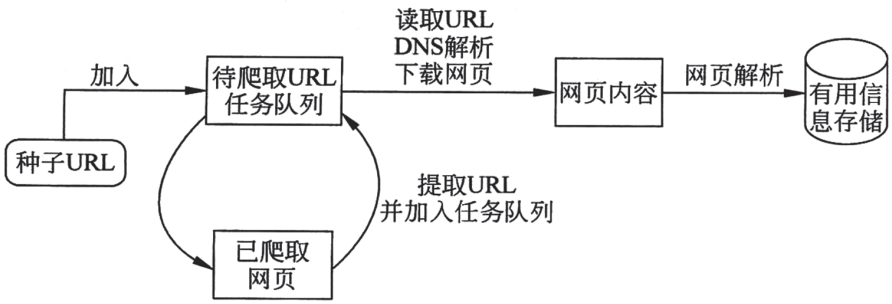


图 1-1　网络爬虫流程

1．统一资源定位器

在万维网（world wide web，WWW）中，每一个信息资源均拥有统一且在万维网范围内唯一的标识地址，该地址被称为统一资源定位器（uniform resource locator，URL）。URL 的表现形式既可以是包含字母、数字及特定符号的域名（例如 baidu.com），也可以是互联网协议（IP）地址，如 202.108.22.5。由于域名的字符组合形式相较于纯数字构成的 IP 地址更便于记忆，因此绝大多数用户访问网站时倾向于通过域名进行操作。

传统形式的 URL 格式为 scheme://host:port/path?query#fragment。

scheme：协议，例如 HTTP、HTTPS、FTP 等，常见的 scheme 名称及含义见表 1-1。

host：域名或者 IP 地址。

port：端口，HTTP 默认端口为 80，可以省略。

path：路径，例如“/abc/a/b/c”。

query：查询参数，例如“token = sdfs2223fds2&name = sdffaf”。

fragment：锚点，用于定位网页的某个位置。

下面是两个符合规则的 URL：

http://www.baidu.com/java/web?flag = 1#function。

http://www.runoob.com/html/html-tutorial.html。

表 1-1　常见的 URL scheme

scheme	访问	用途
http	超文本传输协议	以 http:// 开头的普通网页，不加密
https	安全超文本传输协议	安全网页，加密所有信息交换
ftp	文件传输协议	用于将文件下载或上传至网站
file	访问计算机上的文件	访问电脑本地文件

在 URL 中，如上述符合规则的 URL，有一些特殊字符，如“#”“？”“&”“/”“+”等，这些特殊字符代表不同含义。“#”代表网页中的一个位置，其后面的字符是该位置的标识符。例如“http://www.example.com/index.html#print”中的“#”代表网页 index.html 的“print”位置，浏览器读取这个 URL 后，会自动将“print”位置滚动至可视区域。对 URL 各种字符及其含义作了简单说明，见表 1-2。

表 1-2　URL 各字符及其含义

字符	含义十六	进制
+	表示空格（URL 中不能使用空格）	%2B
空格	URL 中的空格可以用 + 或者编码	%20
/	分隔目录和子目录	%2F
?	分隔实际的 URL 和参数	%3F
#	表示书签，代表网页中的一个位置	%23
&	URL 中指定的参数间的分隔符	%26
=	URL 中指定的参数的值	%3D

在 URL 中，空格和单引号都会被编码，例如访问 URL“https://www.baidu.com/?id = ”抓包时，单引号、双引号、中文和空格都会被编码为如下字符串：

“https://www.baidu.com/?id = 1&name = %27admin%27&pass = %27123456%27”

2. 超文本传输协议

超文本传输协议（hyper text transfer protocol，HTTP）是一种用于传输超媒体文档（如 HTML 文档）的应用层网络协议。该协议最初主要面向 Web 浏览器与 Web 服务器之间的通信场景设计，同时也支持其他各类数据传输需求。

HTTP 遵循经典的客户端—服务器架构模式，客户端主动建立网络连接并发送请求后，进入等待状态，直至接收服务器返回的响应。

HTTP 属于无状态协议，其“无状态”特性具体表现为：两次通信连接之间不存在关联关系，每次通信均为独立的全新连接；服务器端不会记录前后两次请求的相关上下文信息，也不会在多个请求之间保留任何数据状态。

（1）典型的 HTTP 会话。

在 HTTP 的客户端—服务器协议架构中，一次完整的通信会话包含三个核心阶段，具体流程如下。

①连接建立阶段，客户端与服务器端建立 TCP 连接（若传输层未采用 TCP 协议，可选用其他适配的传输层连接方式，以满足通信需求）。

②请求发送阶段，客户端通过已建立的连接向服务器端发送请求消息，并进入等待响应的状态。

③响应返回阶段，服务器端接收并完成请求处理后，向客户端返回响应消息，该响应消息由状态码和对应的数据内容两部分组成。

自 HTTP/1.1 版本起，协议引入长连接机制。完成上述第三阶段后，连接默认保持不关闭。基于该持久连接，客户端可按需再次发起新的请求，此时第二阶段（请求发送）与第三阶段（响应返回）可连续执行多次，无需重复建立连接。

（2）建立连接。

在客户端 / 服务器端协议中，连接是由客户端发起并建立的。在 HTTP 中打开连接意味着在底层传输层启动连接，通常是 TCP。

使用 TCP 时，HTTP 服务器的默认端口号为 80，另外还有 8000 和 8080 也很常用。网页的 URL 会包含域名和端口号，但当端口号为 80 时可以省略。

（3）发送客户端请求。

一旦连接建立，用户代理就可以发送请求，用户代理通常是 Web 浏览器，但也可以是其他的程序，例如爬虫程序。客户端请求是由一系列文本指令组成的，被划分为三个块。

①第一行包含请求方法及请求参数，具体内容如下：一是不包含协议和域名、仅为文档路径的绝对路径 URL；二是所使用的 HTTP 协议版本。

②接下来的每一行均代表一个 HTTP 头部，用于向服务器传递关于所需数据的信息，例如语言、MIME 类型，或是一些可改变请求行为的数据，例如数据已被缓存的情况。这些 HTTP 头部共同构成一个数据块，并以一个空行作为结束。

③最后一块是可选数据块，包含更多数据。

【例 1-1】访问 developer. 摩托拉 .org 的根页面“http://developer. 摩托拉 .org/”，并告诉服务器用户代理该网页需要使用法语展示，指令如下：

```
GET/HTTP/1.1
Host:developer.摩托拉.org
Accept-language:fr

```

注意最后的空行，它把头部与数据块隔开。由于在 HTTP 头部中没有 Content-Length，数据块是空的，所以服务器可以在收到代表头部结束的空行后就开始处理请求。

【例 1-2】发送表单的结果，代码如下：

```
POST/contact_form.php HTTP/1.1
Host:developer.mozilla.org
Content-Length:64
Content-Type:application/x-www-form-urlencoded
name = Joe%20User&request = Send%20me%20one%20of%20your%20catalogue
```

（4）请求方法。

HTTP 定义了一组请求方法，用于指定对目标资源的操作行为。其中，最常用的请求方法为 GET 和 POST。

GET 方法用于请求指定资源，且仅被用于获取数据。

POST 方法用于向服务器发送数据，会导致服务器状态发生改变，该方法常用于 HTML 表单中。

（5）服务器响应结构。

当收到用户代理发送的请求后，Web 服务器将对其进行处理，并最终返回一个响应。与客户端请求类似，服务器响应由一系列文本指令构成，分为三个不同的部分：

①第一行是状态行，包含所使用的 HTTP 协议版本、状态码以及状态描述。

②紧随其后的每一行代表一个 HTTP 头部，用于向客户端提供关于所发送数据的相关信息，例如数据类型、数据大小、采用的压缩算法、缓存指示等。与客户端请求的头部部分类似，这些 HTTP 头部组成一个整体，并以一个空行结束。

③最后一部分是数据块，包含响应的数据内容，也可能不包含任何数据。

【例 1-3】成功的服务器响应网页代码如下：

```
HTTP/1.1 200 OK
Date:Sat,09 Oct 2010 14:28:02 GMT
Server:Apache
Last-Modified:Tue,01 Dec 2009 20:18:22 GMT
ETag:"51142bc1-7449-479b075b2891b"
Accept-Ranges:bytes
Content-Length:29769
Content-Type:text/html

<!DOCTYPE html……
```

【例 1-4】请求资源已被永久移动的网页响应代码如下：

```
HTTP/1.1 301 Moved Permanently
Server:Apache/2.2.3(Red Hat)
Content-Type:text/html;charset = iso-8859-1
Date:Sat,09 Oct 2010 14:30:24 GMT
Location:https://developer.摩托拉.org/(目标资源新地址，服务器期望用户代理去访问它)
Keep-Alive:timeout = 15,max = 98
Accept-Ranges:bytes
Via:Moz-Cache-zlb05
Connection:Keep-Alive
X-Cache-Info:caching
X-Cache-Info:caching
Content-Length:325(如果用户代理无法转到新地址，就显示一个默认网页)

<!DOCTYPEHTMLPUBLIC"-//IETF//DTD HTML2.0//EN">
<html><head>
<title>301 Moved Permanently</title>
</head><body>
<h1>Moved Permanently</h1>
<p>The document has moved<a href = "https://developer.摩托拉.org/">here</a></p>
<hr>
<address>Apache/2.2.3(RedHat)Server at developer.mozilla.org Port80</address>
</body></html>
```

【例 1-5】请求资源不存在的网页响应代码如下：

```
HTTP/1.1 404 Not Found
Date:Sat,09 Oct 2010 14:33:02 GMT
Server:Apache
Last-Modified:Tue,01 May 2007 14:24:39 GMT
ETag:"499fd34e-29ec-42f695ca96761;48fe7523cfcc1"
Accept-Ranges:bytes
Content-Length:10732
Content-Type:text/html

<!DOCTYPE html……
```

指令包含了一个站点自定义的 404 网页。

（6）响应状态码。

HTTP 响应状态码用来表示一个 HTTP 请求是否成功完成。响应被分为 5 种类型：信息型响应、成功响应、重定向、客户端错误和服务器 URL 错误。常见的响应代码如下：

① 200:OK，表示请求成功。

② 301:Moved Permanently，表示请求资源的 URL 已被改变。

③ 404:Not Found，表示服务器无法找到请求的资源。

（7）HTTPS。

HTTPS（安全超文本传输协议）作为 HTTP 协议的加密版本，通常借助 SSL 或 TLS 协议，对客户端与服务器之间的所有通信进行加密处理。通过建立安全链接，客户端能够与服务器安全地交换敏感数据，像网上银行交易、在线商城购物等涉及金钱往来的操作，均依赖这一安全机制来保障数据传输的安全性。

课堂探讨

小李：老师刚才讲到 HTTPS 是 HTTP 的加密版本，那是不是说我们平时上网，只要网址前面是“https://”就代表安全了？

小王：对，基本上是这样。HTTPS 会在客户端和服务器之间建立一个加密通道，用的是 SSL 或 TLS 协议。比如我们在淘宝买东西、登录网银的时候，浏览器地址栏会有个小锁图标，那就是用了 HTTPS。

小张：那 SSL 和 TLS 有什么区别啊？我好像经常看到这两个词混着用。

小李：我记得老师提过，SSL 是早期的协议，现在基本被 TLS 取代了。TLS 其实是 SSL 的升级版，更安全。不过大家习惯性还是说“SSL 证书”，其实技术上用的大多是 TLS。

小王：没错！而且这个加密不只是防别人偷看数据，还能防止中间人篡改内容。比如有人想在你和银行服务器之间插入假信息，HTTPS 也能检测出来。

小张：那是不是所有网站都应该用 HTTPS 啊？我看有些小网站还是 http 开头。

小李：理论上是的。现在主流浏览器都会把 HTTP 网站标记为“不安全”。而且像 Google 这些搜索引擎还会给 HTTPS 网站更高的排名，鼓励大家用加密连接。

小王：对，尤其是涉及账号密码、支付信息的网站，不用 HTTPS 根本没法保障用户隐私。咱们做网页开发的时候，也得记得部署 SSL/TLS 证书。

小张：明白了！所以 HTTPS 不只是“加个锁”那么简单，它背后是一整套安全机制，保护我们在线交易和通信的安全。

小李：没错，可以说没有 HTTPS，就没有今天可信的互联网服务！

任务 1.2 步步为营：网络爬虫的标准流程

任务描述

在建立了理论基础后，陈工给小林布置了第二个任务：“理论是骨架，现在我们要为它填充血肉。你的任务是，亲手走一遍一个标准爬虫的完整生命周期。从向服务器发起请求开始，到最终把数据安全地存下来为止。每一步都要清晰、严谨，这是所有爬虫程序的通用范式。”

任务分析

本任务将抽象概念转化为具体可执行的步骤。陈工指导小林将爬虫工作流程分解为四个核心环节：发起请求、获取响应、解析内容和持久化保存。同时，陈工特别强调了工程实践中的关键点：优先检查网站 API、准确定义所需字段、进行数据流量分析以规划爬取路径。这不仅教会了小林“怎么做”，更培养了他“如何做得好”的工程化思维，为后续的开发打下了坚实的操作基础。

相关知识

1.2.1 发起请求

HTTP 库负责向目标网站发起请求，请求中可包含请求头部、请求参数等附加信息，随后等待服务器对该访问请求的响应。这一过程与人工操作浏览器的流程类似：人工通过浏览器地址栏输入目标网址 URL 并按下回车键确认，本质上是将浏览器作为客户端，向服务器发送访问请求的过程。

1.2.2 获取响应内容

当服务器处于正常运行状态且能够及时作出响应时，我们便能够获取相应的响应内容。这些响应内容正是我们指定要爬取的目标数据，其数据类型丰富多样：既可以是 HTML 文档类型，这类文档通常包含网页的结构与文本信息；也可以是二进制数据类型，例如图片、音频、视频等多媒体类型数据，它们以特定的二进制编码形式存在；还可以是 JSON 字符串，这种格式的数据便于在计算机程序之间进行数据交换与解析。

1.2.3 解析内容

如果爬取的响应内容是 HTML 文档类型的文件，可以使用正则表达式和网页解析库对内容进行解析。爬取的内容也可能是 JSON 字符串，可以直接将其转换成 JSON 对象进行解析。爬取的内容也可能是二进制数据，如图片、音频、视频等，可以保存或进一步处理。

1.2.4 持久化保存数据

爬取的数据需要根据原始数据类型与格式进行持久化保存，文本类数据可保存为 TXT、JSON、CSV 等文本文件，结构化数据可存入数据库；二进制资源（如图片、音频、视频）则需按照其原生格式保存为 JPG、MP3、MP4 等文件。这个过程与人工在网页上下载、保存文件的原理一致。

任务 1.3　法律边缘：网络爬虫的合法性问题剖析

任务描述

当小林对技术流程充满热情时，陈工严肃地提醒他：“技术是把双刃剑，小林。我们的第三个任务至关重要：深入研究网络爬虫的道德与法律边界。你必须学会如何做一个‘守规矩’的爬虫开发者，而不是一个‘破坏者’。”

任务分析

本任务聚焦于合规性与职业道德，是项目中不可或缺的风险控制环节。陈工带领小林重点学习了 Robots 协议，将其比喻为网站与爬虫之间的“君子协定”。他们一起分析了淘宝、腾讯等大厂的 robots.txt 文件，解读了 User-Agent、Disallow、Allow 等指令的含义。陈工强调，尊重 robots.txt 不仅是行业共识，更是规避法律风险、保护目标网站服务器稳定性的基本要求。

相关知识

1.3.1　爬虫协议定义

网络爬虫的作用具有显著的双面性：它既为数据采集提供了高效便捷的途径，也对网络信息安全构成了潜在威胁。部分不法主体借助爬虫技术非法搜集网民个人信息，或发起恶意爬取攻击，造成目标网站服务器过载、系统瘫痪等严重后果。基于此，互联网领域形成了一项针对爬虫行为的公认道德规范，即爬虫协议。

爬虫协议，又称 Robots 协议，全称为“网络爬虫排除标准”（robots exclusion protocol）。该协议是网站向搜索引擎爬虫传递爬取规则的核心方式，用于明确告知爬虫程序网站中哪些网页允许爬取、哪些网页禁止爬取。作为国际互联网界通行的行业道德规范，Robots 协议虽未被纳入法律强制约束范畴，但所有爬虫的设计与运行均应遵循此项协议。

知识拓展

Robots 协议的产生

robots.txt 并不是某一个公司制定的，而是早在 1993 年就已出现，当时还没有 Google。真实 Robots 协议的起源是在互联网从业人员的公开邮件组里面讨论并诞生的。即便是今天，互联网领域的相关问题仍然是在一些专门的邮件组中讨论并产生（主要是在美国）。

1994 年 6 月 30 日，在经过搜索引擎人员以及被搜索引擎抓取的网站站长共同讨论后，一份行业规范被正式发布，即 robots.txt 协议。在此之前，相关人员一直在起草这份文档，并在世界互联网技术邮件组发布后，这一协议被几乎所有的搜索引擎采用，包括最早的 altavista、infoseek，后来的 Google、bing，以及中国的百度、搜搜、搜狗等公司也相继采用并严格遵循。

Robot，又称 Spider，是搜索引擎自动获取网页信息的电脑程序的通称。Robots 协议的核心思想就是要求 Robot 程序不要去检索那些站长们不希望被直接搜索到的内容。将约束 Robot 程序的具体方法规范成格式代码，就成了 Robots 协议。一般来说，网站是通过 Robots.txt 文件来实现 Robots 协议的。

自有搜索引擎之日起，Robots 协议就是一种最有效的维持着网站与搜索引擎之间平衡的方式，让网站和搜索引擎之间的利益不致过度倾斜。它就像一个钟摆，让互联网上的搜索与被搜索和谐相处。

（资料来源：https://blog.csdn.net/u012965373/article/details/44563999，有删改）

1.3.2 爬虫协议应用价值

Robots 协议的应用价值体现在多个方面：其一，可对网站内占用带宽较高的资源文件（如图片、音乐、视频等）进行爬取限制，降低服务器的运行负载；其二，能够屏蔽站点内的死链接，提升搜索引擎爬虫的爬取效率；其三，可通过配置网站地图链接，引导爬虫程序按照合理路径遍历网站内容。

针对网站中部分访问时资源消耗较高、不希望被搜索引擎爬取的网页，技术人员可将其相关路径写入网站根目录下的 robots.txt 文件中。通常情况下，绝大多数网站都会配置 robots.txt 文件，用于界定搜索引擎的爬取范围与执行规则。若网站在开发与维护阶段未部署 robots.txt 文件，爬虫程序则可获取站点内所有未加密的公开数据，即理论上能够爬取网站的全部公开网页内容。

Robots 协议文件的一般写法如下：

```
User-Agent:*          这里的*代表所有的搜索引擎种类，*是一个通配符;
Disallow:admin/       禁止爬取admin目录下面的目录及文件;
Disallow:require/     禁止爬取require目录下面的目录及文件;
Disallow:ABC/         禁止爬取ABC目录下面的目录及文件;
Disallow:cgi-bin/*.htm    禁止爬取cgibin/目录下的所有以“.htm”为后缀的URL(包含子目录);
Disallow:/*?*         禁止爬取网页中所有包含问号(?)的网址;
Disallow:/.jpg$       禁止爬取网页中所有的.jpg格式的图片;
Disallow:ab/adc.html      禁止爬取ab目录下面的adc.html文件;
Allow:cgibin/         允许爬取cgibin目录下面的目录;
Allow:tmp             允许爬取整个tmp目录;
Allow:htm $           允许爬取以“.htm”为后缀的URL;
Allow:gif $           允许爬取网页和gif格式图片;
Sitemap:          网站地图，告诉爬虫这个网页是网站地图。
```

【例 1-6】禁止所有搜索引擎访问网站任何部分的文件写法。

```
User-Agent:*
Disallow:/
```

【例 1-7】查看百度的爬虫协议，在浏览器地址栏中输入访问网址“https://www.Baidu.com/robots.txt”，代码如下：

```
User-Agent:Baiduspider
Allow:/
User-Agent:Baiduspider-image
```

```
Allow:/
User-Agent:Baiduspider-video
Allow:/
User-Agent:Baiduspidernews
Allow:/
User-Agent:Googlebot
Allow:/
User-Agent:MSNBot
Allow:/
User-Agent:YoudaoBot
Allow:/
User-Agent:Sogou web spider
Allow:/
User-Agent:Sogou inst spider
Allow:/
User-Agent:Sogou spider2
Allow:/
User-Agent:Sogou blog
Allow:/
User-Agent:Sogou News Spider
Allow:/
User-Agent:Sogou Orion spider
Allow:/
User-Agent:Jikespider
Allow:/
User-Agent:Sosospider
Allow:/
User-Agent:*
Disallow:/
```

任务 1.4　工具选择：开发语言与开发环境的选择

任务描述

在完成了所有前置知识的学习后，陈工终于进入了实操阶段："万事俱备，只欠东风。现在的任务是选择我们的武器。小林，你需要了解为什么我们选择 Python 作为开发语言，并熟悉我们将要使用的开发工具链。"

任务分析

本任务是理论通往实践的桥梁，明确了项目的具体技术栈。陈工向小林解释了选择 Python 的核

心原因：其简洁的语法、丰富的第三方库以及强大的数据处理能力，使其成为爬虫开发的首选。同时，他们确定了开发环境和可能用到的数据库。这个任务不仅为小林指明了接下来的编码方向，也让他理解了技术选型背后的逻辑——即在效率、成本和功能之间找到最佳平衡点。

相关知识

1.4.1 开发语言

本书的程序开发工作均基于 Python 编程语言完成。Python 是一门高层次的脚本编程语言，该语言同时融合了解释性、编译性、交互性与面向对象的核心特性。在语言设计层面，Python 秉持着高可读性的设计原则，其语法结构具备鲜明的自身特色，相较于其他编程语言具备突出的辨识度与易用性。

Python 爬虫是用 Python 编写的网络爬虫技术，主要用于互联网数据的定向爬取与处理。与其他编程语言相比，Python 优势在于内置大量功能完备的类库，能低成本、高效率实现爬虫核心功能。Python 爬虫应用广泛，可用于搜索引擎数据爬取、公开数据采集、广告过滤等基础场景，也可作为数据分析前置环节，在网络数据爬取中很重要。网页结构遵循语法规则，Python 生态有基于网页节点属性等的解析类库，调用它能高效精准爬取目标信息。此外，Python 标准库内置网页爬取模板，依托该语言开发爬虫程序，有以下核心技术优势：

（1）相较于 Java、C#、C++ 等编译型编程语言开发的网页文档爬取程序，Python 开发的同类型程序具备界面友好、代码简洁的特点；同时在功能实现上，Python 可借助 Requests、Mechanize 等第三方扩展类库，轻松模拟人工访问浏览器的行为逻辑，使相关功能的开发与实现更为便捷。

（2）爬虫程序在获取网页源码数据后，需对数据进行标签过滤、有效文本提取等清洗处理操作。针对各类文本数据的处理需求，Python 展现出了优异的数据处理能力，对于绝大多数的网页文档数据，仅需编写精简的代码脚本，即可完成全部的处理流程与需求。

小提示

Python 是开发网络爬虫的理想语言。Python 语法简洁、高可读性，能降低编程门槛。Python 拥有丰富且功能完备的第三方类库（如 Requests），能轻松模拟浏览器行为，高效处理数据。最后，借助其强大的解析库，可实现精准定向的数据抓取。

1.4.2 第三方请求库

1. Requests 简介

Requests 是用 Python 语言编写的，是 Python 语言的第三方库，采用 Apache 2.0 Licensed 开源协议的 HTTP 库。Requests 是一个很实用的 Python HTTP 客户端库，编写爬虫程序和测试服务器响应数据时经常会用到。

2. Selenium 简介

Selenium 是一款用于 HTML 网页的 UI 自动化测试工具，支持夸克、Safari、Firefox 等主流界面式浏览器，也支持多种语言开发，比如 Java、C、Python 等。Selenium 本质是通过驱动浏览器，完全模拟对浏览器的各种操作，比如跳转、输入、单击、下拉等，来获得网页渲染之后的结果。

1.4.3　开发工具

开发平台为 PyCharm、Notebook 均可，适配的数据库包含 SQLite、MySQL、MongoDB 等类型。本书中所使用的浏览器为夸克浏览器，开发平台为 PyCharm。

1.4.4　搭建开发环境

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。

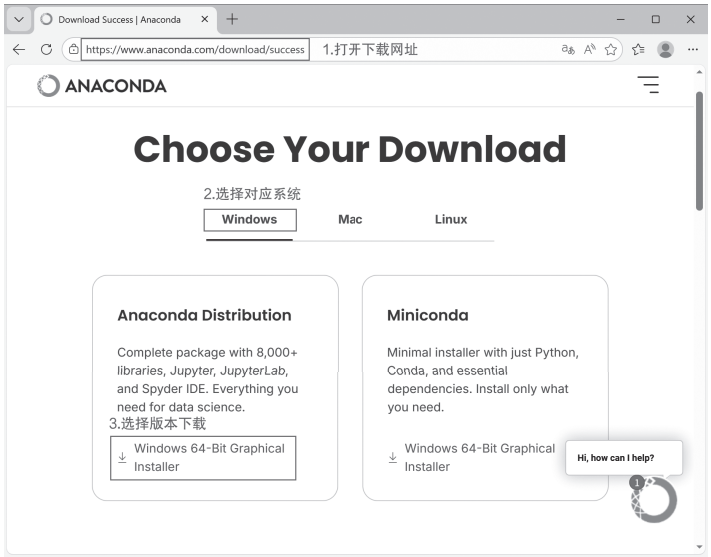


图 1-2　下载 Anaconda

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。



图 1-3　Welcome to Anaconda3 窗口

在 License Agreement 窗口中直接单击“I Agree”按钮，如图 1-4 所示。

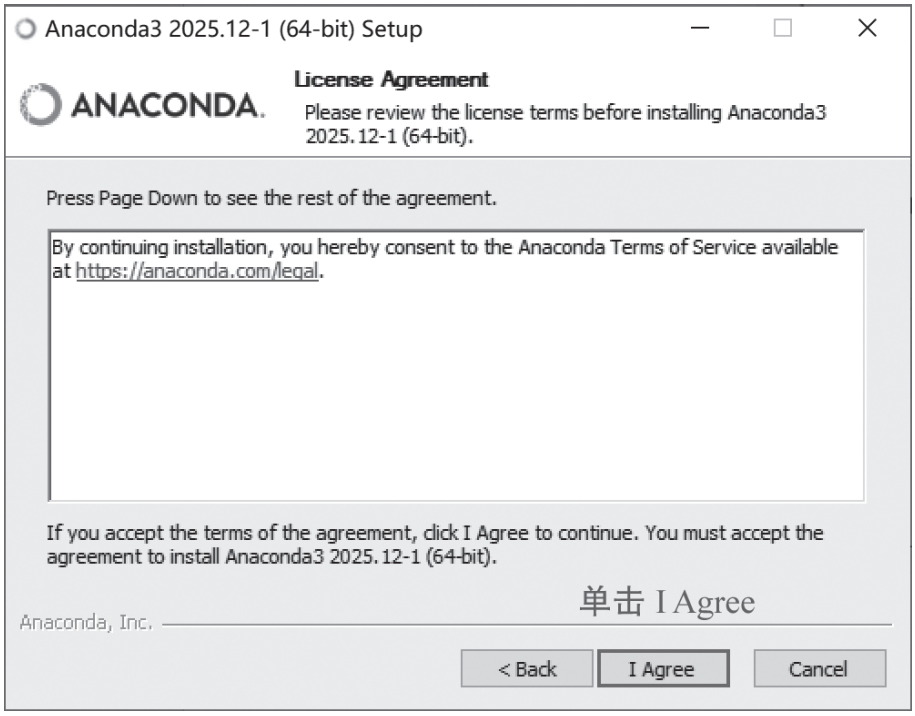


图 1-4　License Agreement 窗口

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。

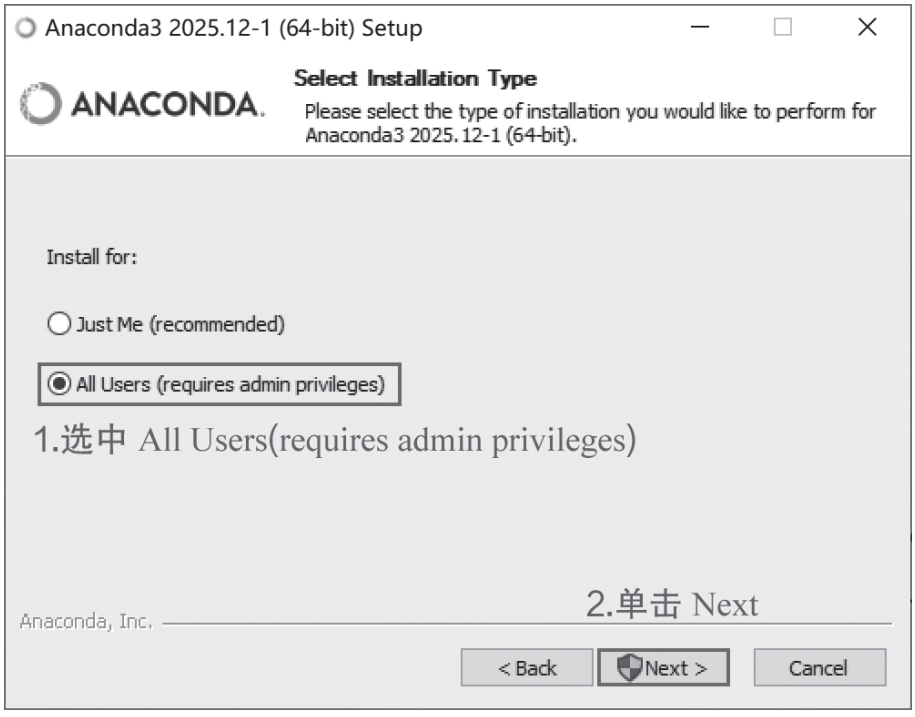


图 1-5　选中 All Users（requires admin privileges）单选按钮

在 Choose Install Location 窗口中选择安装路径（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。

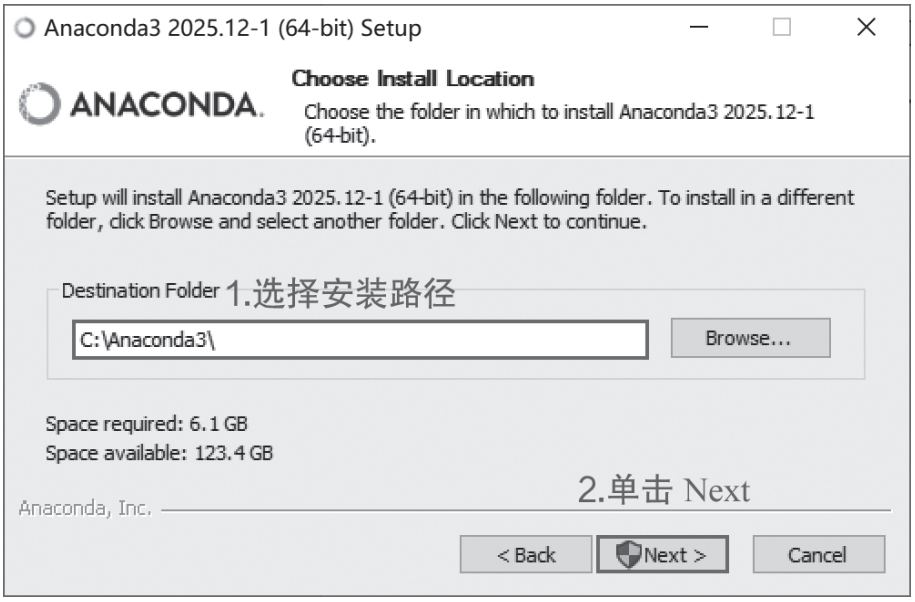


图 1-6　选择安装路径

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。

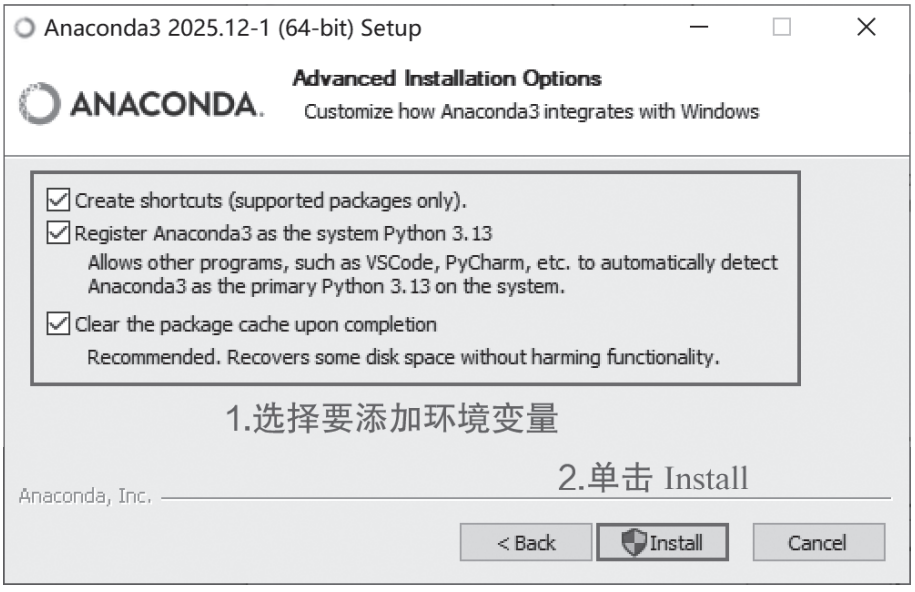


图 1-7　将 Anaconda 加入环境变量

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

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 1-8 所示。

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。

在 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。

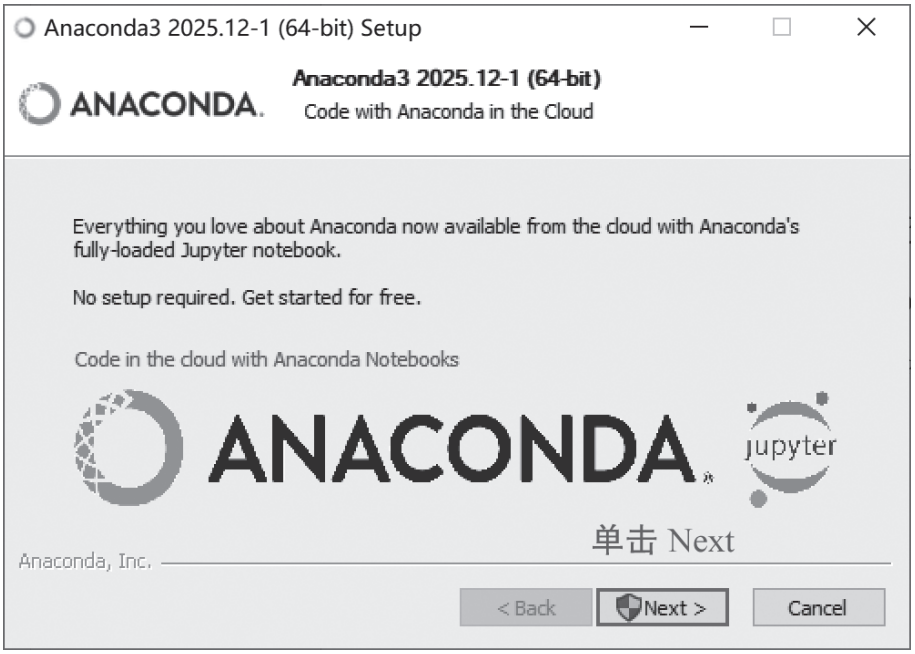


图 1-8　PyCharm 开发工具提示

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，选择对应的版本下载，如图 1-9 所示。



图 1-9　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 1-10 所示。



图 1-10　PyCharm 欢迎界面

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 1-11 所示。

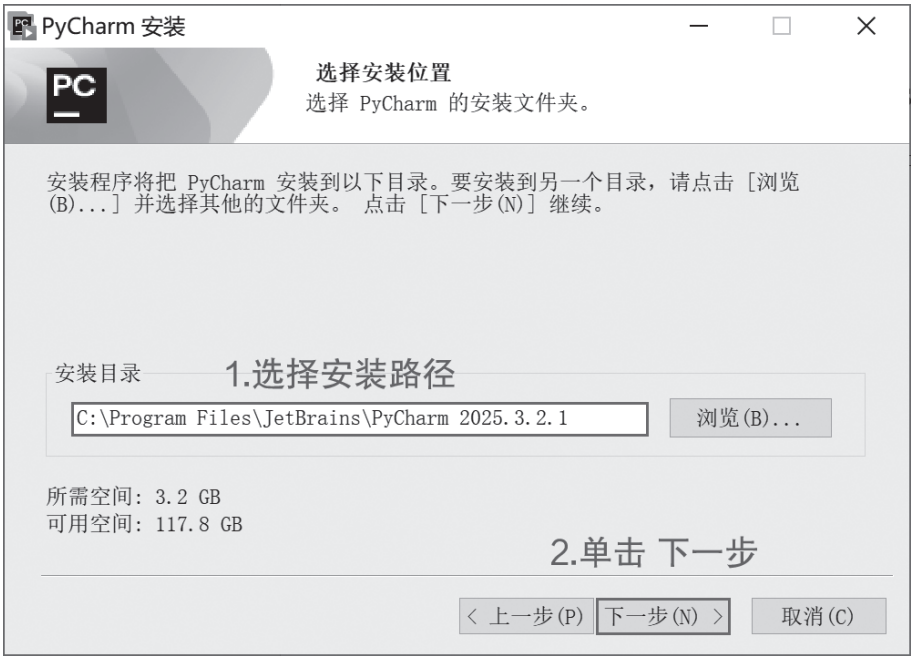


图 1-11　设置 PyCharm 安装路径

在 Installation Options 窗口中首先在桌面快捷方式中设置 PyCharm 程序的快捷方式，然后设置关联文件，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 1-12 所示。



图 1-12　设置快捷方式和关联

在 Choose Start Menu Folder 窗口中直接单击 Install 按钮，如图 1-13 所示。



图 1-13　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。

3. 配置 PyCharm

PyCharm 首次启动时，会弹出“Import PyCharm Settings”窗口，此时需选择“Do not import settings”并点击“OK”，随后进入用户协议页面，勾选“I confirm that I have read and accept the terms of this User Agreement”后点击“Continue”，在数据共享页面选择“Don’t send”以避免数据上传，最终进入 PyCharm

的欢迎界面，完成首次启动的基础初始化。

进入欢迎界面后，点击“Customize”→“All Settings”（或启动后通过“File”→“Settings”，快捷键 Ctrl+Alt+S）进入配置面板，首先完成中文语言包的安装，这是提升新手操作体验的关键步骤：在配置面板左侧选择“Plugins”，在搜索框中输入“Chinese（Simplified）Language Pack”，如图 1-14 所示。点击“Install”按钮开始安装，安装完成后点击“Restart IDE”按钮重启 PyCharm，重启后界面将自动切换为简体中文。

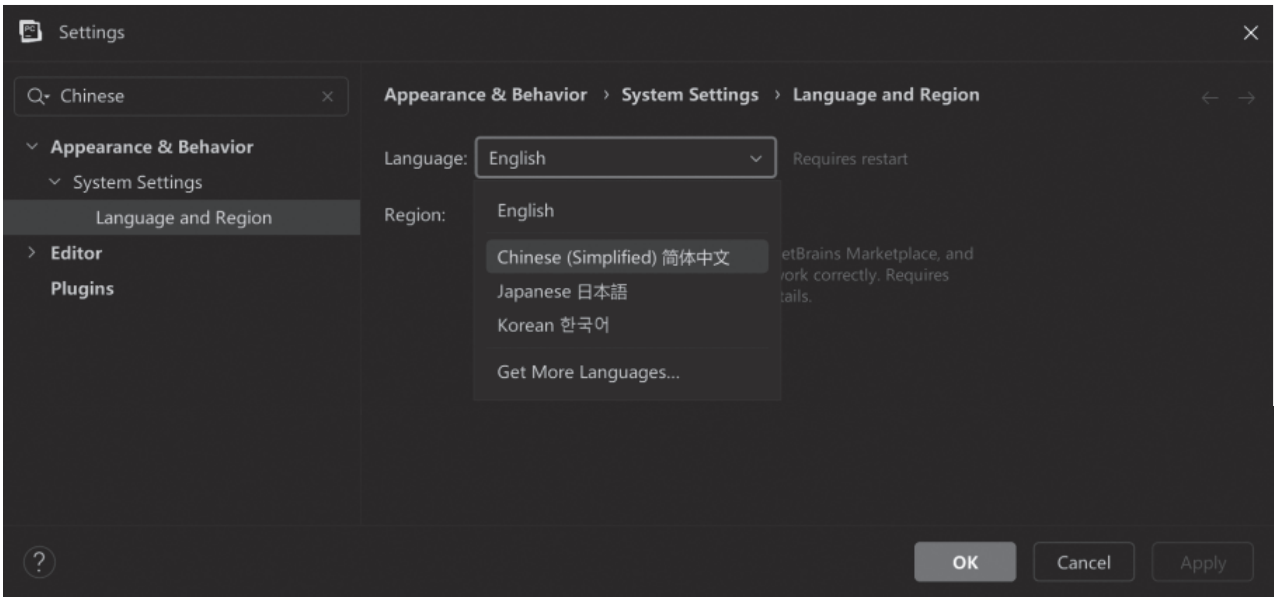


图 1-14　在设置面板搜索框设置

Python 解释器的配置分为全局与项目级两类。全局解释器适用于所有新项目，进入“设置”→“项目”→“Python 解释器”，点击右上角齿轮图标选择“添加”，再选择“系统解释器”，在列表中选中已安装的 Python 版本即可。项目级虚拟环境是教学场景的重点配置，可有效隔离不同项目的依赖，新建项目时点击欢迎界面的“新建项目”，输入项目名称与路径后，勾选“新建虚拟环境”，选择基础解释器为已配置的全局解释器，并勾选“继承全局站点包”，点击“创建”后 PyCharm 会自动生成 venv 虚拟环境文件夹，可通过底部终端输入“pip list”验证，仅显示 pip、setuptools 等基础包即配置成功。

调试环境是教学中的重点内容，配置流程如下：首先编写一段测试代码，例如创建 test.py 文件，写入“a = 10; b = 20; c = a + b; print (f "a + b = {c}")”；随后在需要调试的代码行（如 c = a + b 行）左侧空白处点击，出现红色圆点即完成断点设置；接着点击右上角“运行 / 调试配置”→“编辑配置”，选择“Python”类型，设置脚本路径为测试文件的绝对路径，解释器选择项目的虚拟环境；最后按下 Shift+F9 启动调试，进入调试模式后可通过步进、步过等功能分析代码执行流程，查看变量实时值，帮助理解代码逻辑。

思政园地

数据时代浪潮奔涌，网络爬虫作为数字化转型的重要技术工具，成为连接海量信息与创新应用的关键桥梁，在大数据分析、行业洞察、公共服务等领域发挥着不可替代的作用。然而，技术是一

把双刃剑，爬虫技术的便捷性既为数据采集提供了高效路径，也潜藏着触碰法律边界、侵犯信息安全的风险，这要求每一位技术学习者都必须坚守“科技向善”的初心与底线。

本项目强调的 Robots 协议，不仅是行业自律的“君子协定”，更是技术从业者必须恪守的道德准则与行为规范。它提醒我们，技术创新不能脱离规则约束，数据获取必须建立在合法合规、尊重他人权益的基础之上。在数字经济蓬勃发展的今天，网络空间的秩序需要共同维护，爬虫技术的应用既要追求效率与价值，更要坚守法律红线与道德底线，坚决抵制恶意爬取、非法采集等损害网络安全与他人权益的行为。

当代青年作为数字时代的原住民与技术创新的主力军，生逢科技快速发展的盛世，肩负着推动技术进步与维护网络生态的双重使命。学习网络爬虫技术，不仅要掌握编程技能、解析方法等硬实力，更要锤炼规则意识、责任担当等软实力。正如文档素质目标所倡导的，在提升获取信息、分析问题、解决问题能力的同时，更要树立正确的技术观，将合规合法的理念融入技术实践的每一个环节。

数字化、智能化是国家发展的重要趋势，网络空间的安全与有序是国家治理能力现代化的重要体现。广大青年在钻研爬虫技术、投身数据应用的过程中，要始终牢记技术的初心是服务社会、造福人民，以 Robots 协议为行为指南，以法律法规为行动准绳，用负责任的技术实践助力数字经济健康发展，用坚守底线的担当守护网络空间清朗生态，在科技强国的征程中贡献青春力量，书写无愧于时代的青春答卷。

项目小结

本项目首先揭开了网络爬虫的神秘面纱，系统阐述了爬虫在大数据生态中的核心角色，详细解析了通用、聚焦、增量式及深层网络爬虫的四大类型，并深入讲解了 URL 结构、HTTP/HTTPS 协议原理及请求响应机制等必备基础知识。其次，项目步步为营地拆解了爬虫的标准工作流程，从模拟浏览器发起请求、获取 HTML 或 JSON 等响应内容，到解析数据与持久化存储，构建了完整的工程化开发逻辑。最后，项目重点剖析了网络爬虫的法律边界与道德规范，强调了 Robots 协议的约束力，并指导完成了 Python 开发环境的搭建与配置。通过本项目，学生不仅掌握了网络爬虫的核心技术与开发工具，更树立了合规采集数据的职业意识，为后续进行高效、安全的数据挖掘与分析奠定了坚实的基础。

项目实训

网络爬虫技术项目实训

请按照表 1-3 任务单的内容进行项目实训。

表 1-3　任务单

姓名		学号	
班级		组别	
任务项目	项目 1　网络爬虫技术		

续表

任务名称	网络爬虫技术项目实训
典型工作场景	基于 Python 编程语言，搭建网络爬虫开发环境，遵循 Robots 协议，完成从网页请求、数据解析到持久化保存的全流程实践，开发简易网络爬虫程序
学习目标	1. 掌握网络爬虫的基本概念、类型、标准工作流程及合法性要求； 2. 理解 URL、HTTP/HTTPS 协议、HTML 结构等爬虫必备预备知识； 3. 学会使用 Python 及 Requests、Selenium 等库开发简易爬虫程序； 4. 掌握数据解析（正则表达式、解析库）与持久化保存（文本 / 数据库）方法； 5. 培养合规爬取意识、逻辑思维能力及问题解决能力
任务描述	1. 系统学习网络爬虫的核心理论； 2. 深入理解 URL 结构、HTTP/HTTPS 协议原理、HTML 文档结构； 3. 安装 Anaconda、PyCharm，配置 Python 解释器及相关第三方库； 4. 编写程序实现网页请求、响应解析、数据提取及持久化保存； 5 在实践中遵循 Robots 协议，确保爬取行为合法合规
任务要求	1. 能准确区分通用、聚焦、增量式、深层网络爬虫的适用场景； 2. 能完整描述网络爬虫“发起请求—获取响应—解析内容—持久化保存”的标准流程； 3. 能解读目标网站的 Robots 协议，规避法律风险； 4. 能独立搭建 Python 爬虫开发环境，熟练使用 Requests 库发送请求、Selenium 库模拟浏览器操作； 5. 能使用正则表达式或网页解析库提取目标数据，并保存为文本文件或数据库； 6. 开发的爬虫程序需符合合规性要求，不恶意爬取网站数据
参考资料	1. 网络爬虫核心理论：爬虫类型、工作流程、Robots 协议规范； 2. 预备知识文档：URL 结构解析、HTTP/HTTPS 协议原理、HTML 文档语法； 3. 开发工具手册：Anaconda 安装指南、PyCharm 配置教程； 4.Python 第三方库文档：Requests 库使用手册、Selenium 库 API 参考； 5. 合规性参考：淘宝网、百度等主流网站 Robots 协议实例

制作计划单，帮助学生清晰梳理实训各环节的工作步骤、明确操作注意事项，具体填写见表 1-4。

表 1-4　计划单

任务名称	网络爬虫技术实践
典型工作场景	基于 Python 编程语言，搭建网络爬虫开发环境，遵循 Robots 协议，完成从网页请求、数据解析到持久化保存的全流程实践，开发简易网络爬虫程序
相关资讯描述	
计划制定方式	

续表

序号	工作步骤	注意事项
1		
2		
3		
4		
5		

计划评价	个人评分	小组评分	指导老师评分

项目小练

1. 网络爬虫在大数据生态中的主要作用是什么？
2. 通用网络爬虫与聚焦网络爬虫的主要区别是什么？
3. 网络爬虫的标准工作流程包含哪四个核心环节？
4. Robots 协议的作用是什么？请列举其三条应用价值。

项目 2

Web 前端基础

项目导读

本项目聚焦 Web 前端开发基础，系统讲解构建现代网页所需的核心技术。内容涵盖 HTTP 协议、HTML 语言、CSS 层叠样式表、JavaScript 脚本语言，阐述它们在实现网页行为与用户交互中的重要作用。通过本项目的学习，可为后续网络爬虫的数据解析和 Web 开发奠定坚实基础。

学习目标

知识目标

1. 理解 HTTP 协议的基本原理。
2. 掌握 HTML 语言的基本语法。
3. 了解 CSS 层叠样式表的作用和基本语法。
4. 了解 JavaScript 脚本语言的基本概念和使用方法。

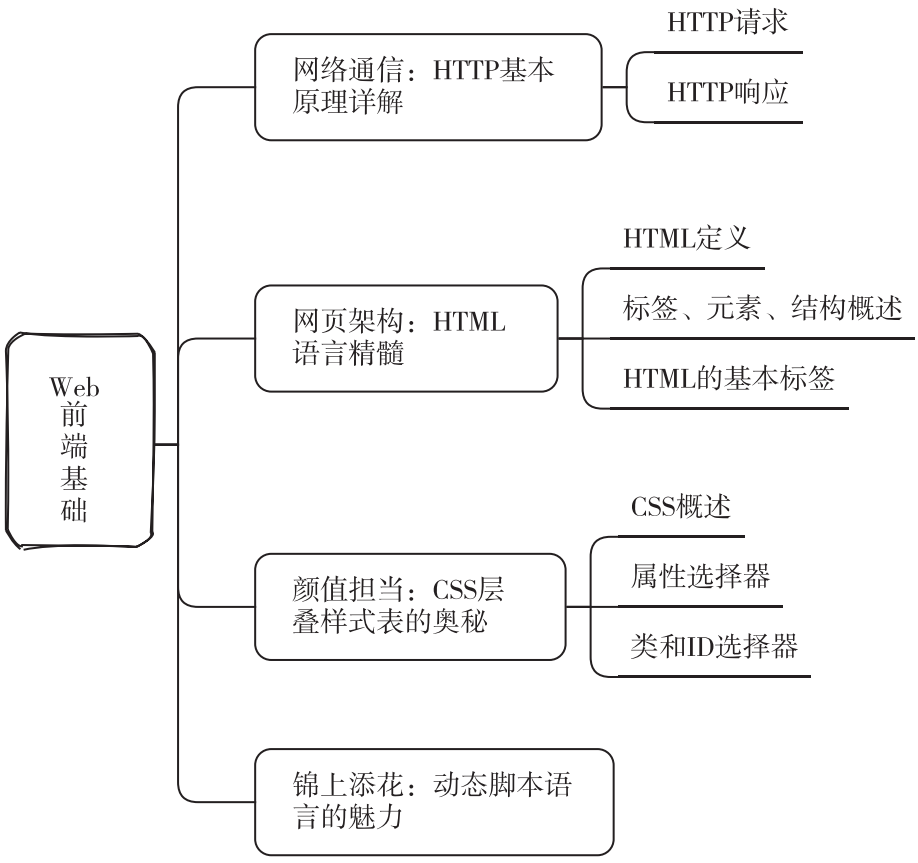
技能目标

1. 能够使用浏览器开发者工具，查看 HTTP 请求和响应信息。
2. 能够编写简单的 HTML 页面，并使用 CSS 美化页面样式。
3. 能够在 HTML 页面中嵌入 JavaScript 代码，实现简单的动态交互功能。

素质目标

1. 培养信息素养，能够获取和利用网络信息资源。
2. 培养逻辑思维能力，能够分析和解决问题。
3. 培养自主学习能力，能够不断学习和更新知识。

知识图谱



任务 2.1 网络通信：HTTP 基本原理详解

任务描述

确定 Python 为爬虫开发的核心语言后，陈工向小林强调，仅掌握 Python 不足以成为优秀的爬虫工程师——爬虫的核心目标是从网页精准获取数据，若不了解网页构造，爬取工作会效率低下且易出现错误。小林需深入学习 Web 前端的三大核心技术 HTML、CSS 和 JavaScript，这是成为专业爬虫工程师的必备基础。陈工打开夸克浏览器，为小林布置了一个学习任务：以“网络侦探”的视角，掌握浏览器开发者工具的使用方法，监听并理解网页加载过程中，浏览器与服务器之间 HTTP 请求和响应的完整流程，同时能够准确识别请求的 URL、请求方法、请求头信息，以及服务器返回的状态码和响应体。

任务分析

本任务让小林从理论走向可视化实践，深刻理解爬虫与服务器交互的核心协议——HTTP。陈工

指导小林通过实际操作，将抽象的 HTTP 概念具象化。他们重点剖析了 HTTP 请求的四大组成部分和 HTTP 响应的三大部分。通过查看百度首页的真实请求记录，小林明白了 User-Agent 是伪装身份的关键，Cookie 用于维持会话，而 200 OK 状态码则是数据成功获取的信号。这为他后续编写能与服务器正常“对话”的爬虫代码奠定了坚实的基础。

相关知识

在浏览器中输入一个 URL 并按下回车键后，网页内容便会显示在浏览器中。实际上，这一过程是浏览器向 Web 服务器发送了一个 HTTP 请求。Web 服务器接收到该请求后进行解析与处理，随后向浏览器返回对应的 HTTP 响应。浏览器再对 HTTP 响应进行解析，最终将网页呈现出来。下面以使用夸克浏览器打开百度搜索页面为例，展示 HTTP 请求与响应的过程。

打开夸克浏览器，在页面空白处单击鼠标右键，在弹出的快捷菜单中选择“检查”选项，打开浏览器的开发者工具窗口，随后点击工具窗口顶部的“网络”选项卡。访问 https://www.baidu.com/，即可在开发者工具的网络面板中查看完整的请求记录，如图 2-1 所示。每一条记录均对应一次 HTTP 请求的发送与响应过程。

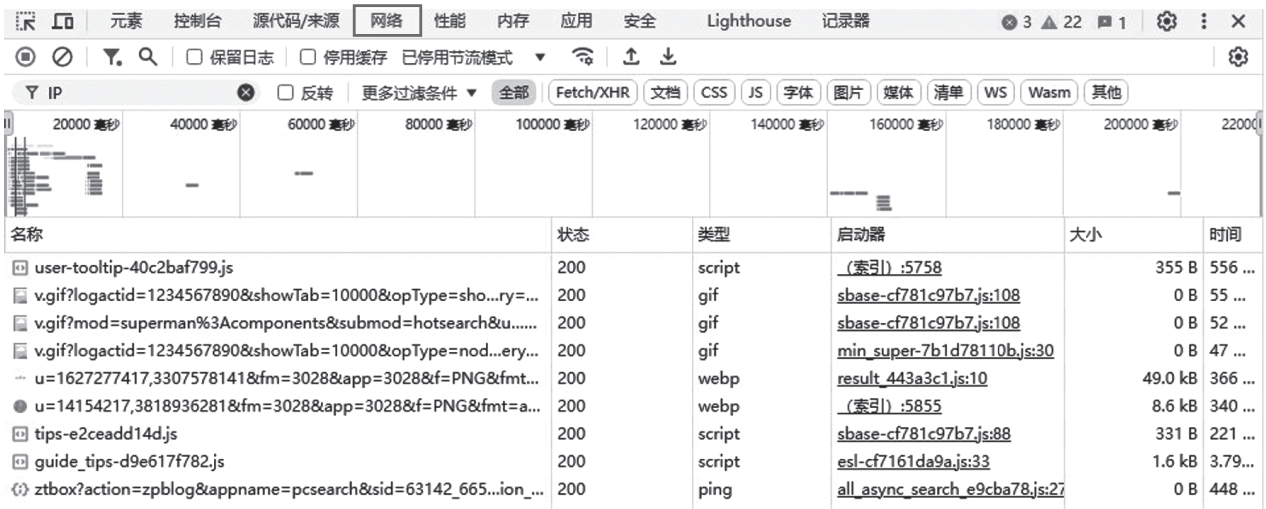


图 2-1　夸克浏览器显示 HTTP 请求记录

在记录列表中，选择“www.baidu.com”选项，然后在上方打开的界面中选择“网络”选项，即可显示 HTTP 请求和响应的详细信息，如图 2-2 所示。

2.1.1　HTTP 请求

HTTP 请求主要由四个核心部分构成，分别为请求的网址（Request URL）、请求方法（Request Method）、请求头（Request Headers）以及请求体（Request Body）。以下对这四个部分展开详细介绍。

1. 请求的网址

请求的网址即统一资源定位符（URL），它在网络世界中扮演着至关重要的角色，能够唯一且精准地确定请求所指向的资源。例如，在图 2-2 中，“Request URL”部分所展示的“https://www.baidu.com/”，就是一个典型的请求网址，通过它，浏览器能够明确知晓要访问的具体网页资源。

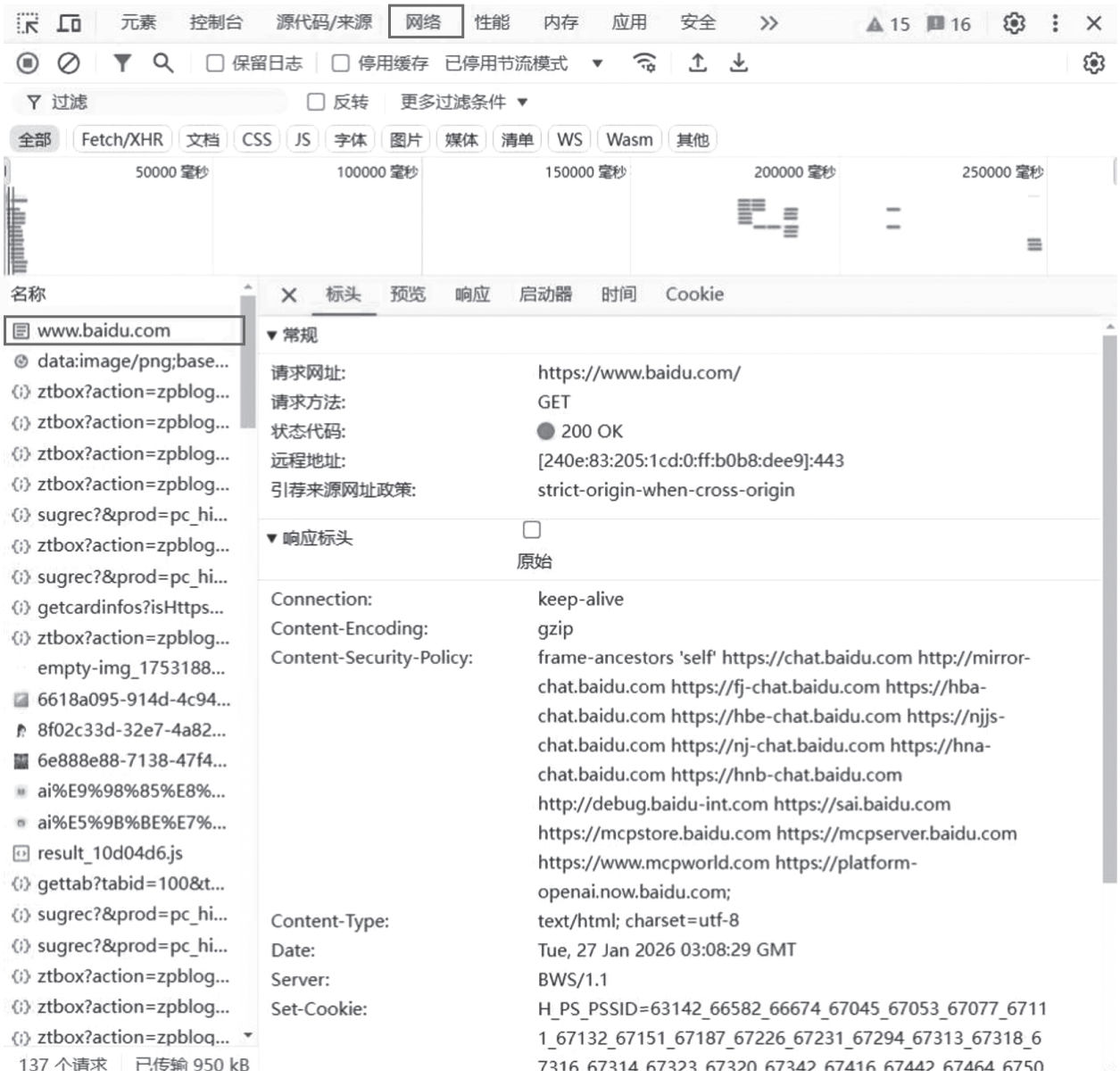


图 2-2　HTTP 请求和响应的详细信息

2. 请求方法

在 HTTP 协议中，常见的请求方法主要有 GET 方法和 POST 方法。

（1）GET 方法的主要功能是请求指定的网页信息，服务器在接收到该请求后，会将对应的网页内容返回给客户端。不过，GET 方法存在一定的局限性，其提交的数据量最多只有 1024 字节。例如，在图 2-2 中，“请求方法：”部分显示为 GET 方法，这表明此次请求是为了获取网页内容。在日常生活中，当我们打开网站时，通常使用的就是 GET 方法，其本质就是向服务器请求特定的页面。

（2）POST 方法主要用于向指定资源提交数据，并请求服务器对该数据进行相应的处理，例如提交表单数据或者上传文件等操作。与 GET 方法不同的是，POST 方法提交的数据都包含在请求体中，而且提交的数据量没有字节限制。比如，当我们登录网站时，就需要向服务器提交用户名、密码等数据，此时就会用到 POST 方法。

除了上述两种常见的请求方法外，HTTP 协议还定义了其他一些请求方法，如 HEAD、PUT、

DELETE、CONNECT、OPTIONS 和 TRACE 等。然而，在实际编写爬虫程序的过程中，这些请求方法的使用频率相对较低，此处将不再介绍。

知识拓展

HEAD 方法

功能：HEAD 方法与 GET 方法类似，也是用于请求服务器发送某个资源。但不同之处在于，服务器在响应 HEAD 请求时，只返回资源的头部信息，而不返回实体的主体部分。

应用场景：HEAD 方法常用于在不获取资源的情况下了解资源的情况，比如判断资源的类型、查看资源的元数据等。此外，通过查看头部信息，还可以测试资源是否被修改过。

3. 请求头

请求头是请求的重要组成部分，在编写爬虫程序时，大部分情况下都需要设定请求头。不同请求的请求头包含的内容不同，应用时应根据实际需求设定。常见的请求头及其说明见表 2-1。

表 2-1　常见的请求头及其说明

请求头	说明
Accept	指定客户端可识别的内容类型
Accpet-Encoding	指定客户端可识别的内容编码
Accept-Language	指定客户端可识别的语言类型
Cookie	网站为了辨别用户身份进行会话跟踪而存储在用户本地的数据，主要功能是维持当前访问会话
Host	指定请求的服务器的域名和端口号
User-Agent	使服务器识别客户端使用的操作系统及版本、浏览器及版本等信息，实现爬虫时加上此信息，可以伪装为浏览器
Content-Type	请求的媒体类型信息
Content-Length	请求的内容长度
Referer	包含一个 URL，用户以该 URL 代表的页面出发访问当前请求页面

其中，“Content-Type”确定了 POST 请求提交数据的方式，如表 2-2 所示。

表 2-2　Content-Type 和 POST 提交数据方式的关系

Content–Type	提交数据的方式
application/x-www-form-urlencoded	表单数据
multipart/form-data	表单文件
application/json	序列化 JSON 数据
text/xml	XML 数据

4. 请求体

请求体中的内容一般是 POST 请求中的表单数据，而 GET 请求的请求体为空。例如，使用账户登录百度网站可看到 POST 请求中的请求体（“Form Data”部分）。

2.1.2 HTTP 响应

HTTP 响应可分为三部分内容：响应状态码（Response Status Code）、响应头（Response Headers）和响应体（Response Body）。

1. 响应状态码

在网络通信和 Web 服务中，当客户端向服务器发送请求时，服务器会返回相应的响应状态码。这些状态码对于开发人员和系统管理员来说至关重要，它们提供了关于请求处理情况的关键信息。通过了解这些状态码，可以快速诊断和解决网络相关的问题，确保系统的正常运行。常见的响应状态码及其含义说明见表 2-3。

表 2-3 常见的响应状态码

状态码	英文名称	说明
100	Continue	服务器已收到请求的一部分，正在等待其余部分，应继续提出请求
200	OK	服务器已成功处理了请求
302	Move Temporarily	服务器要求客户端重新发送一个请求
304	Not Modified	此请求返回的网页未修改，继续使用上次的资源
404	Not Found	服务器找不到请求的网页
500	Internal Server Error	服务器遇到错误，无法完成请求

图 2-2 中的“Status Code”部分即为响应状态码（200 OK）。在爬取网页时，爬虫程序可依据该状态码判断服务器的响应状态：若状态码为 200，则表明数据返回成功，可开展下一步处理。

2. 响应头

响应头包含服务器对请求的应答信息，图 2-2 中的“响应标头”部分即为响应头。常见响应头及其说明见表 2-4。

表 2-4 常见的响应头及其说明

响应头	说明
Content-Encoding	Web 服务器支持的编码类型
Content-Language	响应体的语言
Content-Length	响应体的长度
Content-Type	返回内容的媒体类型
Date	原始服务器消息发出的时间
Expires	响应过期的日期和时间
Last-Modified	请求资源的最后修改时间
Set-Cookie	设置 HTTP Cookie
Location	重定向接收到请求的 URL 位置

3. 响应体

响应体是 HTTP 响应中极为关键的部分，它承载着响应的正文数据。当客户端发起请求获取网页

时，服务器返回的响应体中包含的便是该网页的源代码，浏览器正是通过对这些源代码的解析与渲染，才将我们熟悉的网页呈现出来；而当请求的对象是图片时，响应体中存储的则是图片的二进制数据，这些二进制数据经过特定处理后，能够在客户端正确显示为对应的图片。在图 2-2 中选择“响应”选项，可显示响应体，如图 2-3 所示。

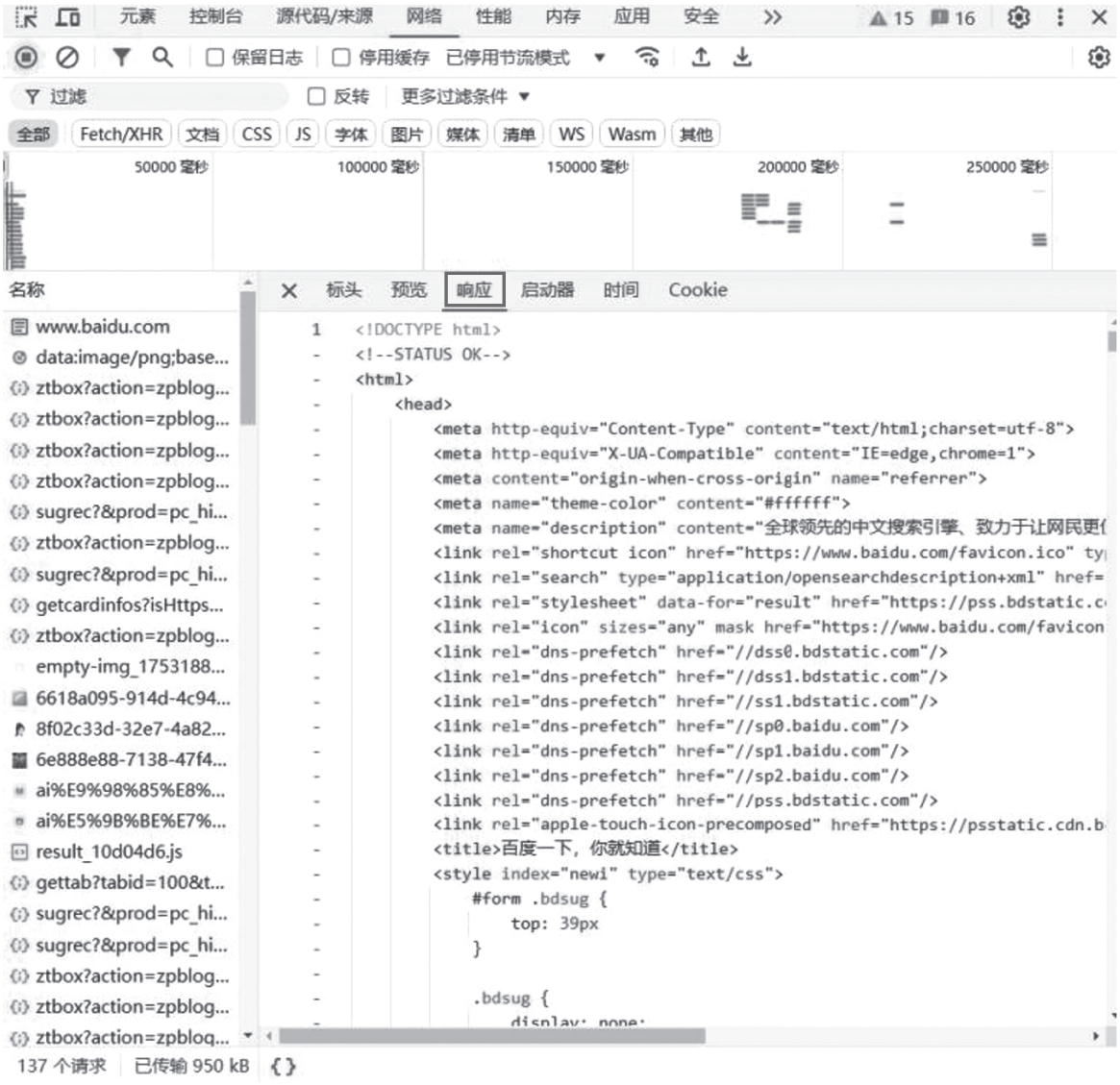


图 2-3　响应体

任务 2.2　网页架构：HTML 语言精髓

任务描述

掌握 HTTP 通信协议的核心原理后，陈工指着浏览器中结构复杂的网页向小林讲解：服务器返回的响应体中，承载的便是网页的“骨架”与“血肉”，也就是 HTML。为此，他为小林布置了第二

个学习任务：学会阅读并理解网页的 HTML 源代码，掌握 HTML 的基本文档结构、核心标签，以及各类元素通过标签嵌套构建完整页面的实现方式。

任务分析

本任务聚焦于网页内容的载体——HTML，这是数据提取的直接对象。陈工让小林动手编写一个简单的个人简历 HTML 页面，并通过浏览器查看效果。通过这个过程，小林直观地理解了 HTML 文档的树状结构：根元素 <html> 包含头部 <head> 和主体 <body>。陈工强调，所有需要爬取的文本、链接、图片等信息，都包裹在特定的 HTML 标签内。掌握这些标签的含义和层级关系，就如同拿到了藏宝图，能让小林在未来精准地定位到目标数据所在的“房间”。

相关知识

2.2.1 HTML 定义

HTML 作为超文本标记语言，属于纯文本类型。用其编写的网页文件同样是标准的纯文本文件，不包含特殊二进制编码。我们既能用任意文本编辑器，如 Windows 的“记事本”，直接打开查看 HTML 源代码，了解网页底层结构，也能在浏览器打开网页时，通过“查看”→“源文件”命令查看代码。而且，HTML 文件无需编译，浏览器可直接解释执行，读取代码、分析语法结构后，将解释结果转化为直观的网页内容呈现给用户。

知识拓展

HTML 的行业发展趋势

当前 HTML 的发展已不再是单一的“标签更新”，而是围绕“跨端兼容、无障碍、性能优化、语义化深化”四大方向，与 CSS、JavaScript、WebAssembly 深度融合：

（1）强化无障碍访问，新增更多面向残障人士的标签 / 属性，满足全球各国的网络无障碍法规要求。

（2）优化性能，精简冗余规范，提升浏览器渲染效率，适配低性能设备（如物联网设备）。

（3）融合 WebAssembly（Wasm），让 C/C++、Rust 等编译型语言可在 HTML 页面中运行，实现网页端高性能计算（如视频剪辑、3D 建模），突破 JavaScript 的性能瓶颈。

（4）适配元宇宙 / 沉浸式网页，新增支持 3D、VR/AR 的标签和 API，让 HTML 成为沉浸式网页的基础载体。

2.2.2 标签、元素、结构概述



HTML 标签的属性

1. HTML 标签

HTML 的标签分为单独出现的标签和成对出现的标签两种类型。大多数标签以成对形式存在，由首标签与尾标签共同构成。首标签的格式为 < 元素名称 >，尾标签的格式