深圳信息职业技术大学

教 案

2025 年/2026 年第 2 学期

专业大类 电子信息

课程名称 Python 网络爬虫程序技术

课程类别 专业课

教学模式（课堂教学/混合式教学） 混合式教学

任课班级 24 软件技术选课班

任课教师 黄锐军

所属院部 软件学院

2026 年 3 月 2 日

说明：

项目	教案数
项目 1 爬取外汇网站数据	5
项目 2 爬取名言网站数据	5
项目 3 爬取天气网站数据	3
项目 4 爬取旅游网站数据	4
项目 5 爬取图书网站数据	7
项目 6 爬取商城网站数据	6
	合计：30

教学单元	项目 1-1 搭建爬虫程序开发环境				
周次		时数	2	授课班级	

教学目标

1. 知识目标

- 理解爬虫程序的定义、核心功能与工作流程
- 掌握 Python 语言的核心特点与主流 Python 开发环境类型
- 掌握 Python + PyCharm 安装与配置流程

2. 技能目标

- 独立完成 Python 与 PyCharm 安装、环境变量配置
- 能在 PyCharm 中创建项目、运行简单 Python 代码
- 区分四种 Python 开发环境的适用场景

3. 情感目标

- 激发对 Python 爬虫开发的学习兴趣
- 养成规范安装、配置开发环境的编程习惯

教学重点：

- 爬虫程序定义与核心流程
- Python 安装与环境变量配置
- PyCharm 安装、项目创建与代码运行

教学难点

- Python 环境变量配置
- PyCharm 解释器关联
- 理解爬虫“请求→获取→解析→存储”逻辑

实验仪器及教具：

计算机、因特网

教　学　过　程	备注

第一部分：爬虫程序认知（20 分钟）

1. 导入新课（5 分钟）

- 问题引入：如何自动获取外汇、新闻、电商数据？
- 引出主题：网络爬虫 —— 自动抓取网页数据的程序
- 案例展示：外汇汇率网页 HTML 源码，直观展示数据位置

2. 理论讲解（10 分钟）

- 爬虫定义：客户端程序，访问 Web 服务器→获取网页代码→提取数据→存入数据库
- 核心流程：**访问服务器 → 获取 HTML → 解析数据 → 存储数据**
- 实例：抓取外汇汇率页面→提取汇率表格→保存数据

3. 课堂提问（5 分钟）

- 爬虫可以获取哪些网页数据？
- 爬虫与手动复制数据的区别是什么？

第二部分：Python 语言与开发环境（20 分钟）

1. 导入（5 分钟）

- 爬虫主流开发语言：Python（简洁、库丰富）

2. 理论讲解（10 分钟）

- Python 特点：开源免费、语法缩进、库丰富、解释型、易维护
- 四大开发环境：

o Python 自带：命令行 + 简易 IDE，仅测试用
o PyCharm：Python 专用，功能强，社区版免费
o VS Code：轻量、插件多
o Anaconda：自带 Python，科学计算方便

- 课程选用：**PyCharm 社区版**

3. 小结（5 分钟）

- 环境对比：轻便性、功能、适用场景

第三部分：Python 安装与配置（20 分钟）

1. 下载（5 分钟）

- 官网：python.org → 下载 Windows/macOS 最新版

2. 安装步骤（10 分钟）

- Windows：

1. 双击安装包
2. **勾选 Add Python.exe to PATH**（关键）
3. 选择 Install now 或自定义安装

- macOS：直接安装，默认配置

3. 验证安装（5 分钟）

- 打开 CMD / 终端，输入：

python --version 或 python3 --version

- 出现版本号即安装成功
- 常见问题：未配置 PATH 导致命令不识别

第四部分：PyCharm 安装与项目创建（15 分钟）

1. 下载安装（2 分钟）

- 官网：jetbrains.com/pycharm/
- 下载 **Community（社区版）**→ 一路下一步安装

教师现场演示

现场演示 + 学生同步操作

2. 首次配置（3 分钟） • 启动 PyCharm → New Project • Location：设置项目路径 • Interpreter：选择 Previously configured interpreter（关联已装 Python） • 取消创建 main.py → Create 3. 运行测试代码（10 分钟） • 新建 spider_test.py • 输入代码： print("Hello Spider") • 右键 Run → 查看控制台输出 • 成功标志：控制台打印文字 **第五部分：课堂总结与答疑（5 分钟）** 1. 核心回顾 • 爬虫定义与流程 • Python + PyCharm 安装配置 • PyCharm 运行第一个程序 2. 常见问题解答 • 环境变量未配置 • PyCharm 找不到解释器 • 命令行无法识别 python 命令 教学资源 • Python 官网、PyCharm 官网 • PyCharm 界面截图、HTML 源码示例 • 安装步骤图文教程	
课后作业： 1. 完成 Python 与 PyCharm 安装，在 PyCharm 运行一段测试代码并截图 2. 整理表格：对比 Python 自带、PyCharm、VS Code、Anaconda 的优缺点 3. 预习：requests 库基础用法与网页请求原理	

（以下内容为选填，供线上课程及线上线下混合式课程填报）

教学资源列表与访问路径	备　注
中国大学 MOOC 网站： https://www.icourse163.org/course/ZIIT-1002925008 https://www.icourse163.org/course/ZIIT-1002924007	
在线教学过程（学生自主学习指引）	
学生在讨论区提问 教师在讨论区回答 学生提交作业，讲师讲评	

在线辅导与答疑安排	
学生观看网络视频 教师在线回答问题	
线上作业： 复习本次课程内容 预习下次课程内容	

教学单元	项目 1-2 Flask 创建 Web 网站				
周次		时数	2	授课班级	

教学目标

1. 知识目标

掌握 Flask 框架的安装方法

理解 Flask 搭建 Web 服务器的核心流程

掌握 urllib.request 获取网页 HTML 代码的原理与步骤

理解 Flask 路由、模板、数据渲染的基本机制

2. 技能目标

能独立安装 Flask 并创建 Web 项目

能编写 Flask 服务器读取 CSV 数据并生成网页表格

能用 urllib.request 编写爬虫获取本地网页 HTML

能完成服务器与客户端程序联调

3. 情感目标

建立“先搭站、再爬取”的爬虫学习思维

培养规范编码、分步调试的编程习惯

提升 Web 与爬虫结合的实践兴趣

教学重点：

Flask 安装与项目搭建

Flask 路由与服务器编写

urllib.request 获取 HTML

CSV 数据转 HTML 表格

教学难点

Flask 路由机制理解

二进制数据解码与编码处理

爬虫“请求 - 获取 - 解析”前置逻辑

实验仪器及教具：

计算机、因特网

教　学　过　程	备注

教学过程	时间分配	备注
一、课程导入（5 分钟） 1. 回顾：爬虫核心流程 “请求→获取→解析→存储” 2. 提出问题：直接爬取外网易报错，如何安全练习？ 3. 引出任务：用 Flask 搭建本地模拟外汇网站，再编写爬虫抓取	5 分钟	激发学习动机
二、新知讲授：Flask 框架安装（10 分钟） 1. Flask 定位：轻量级 Python Web 框架，快速建站 2. 安装命令：pip install flask 3. 验证：出现 Successfully installed 表示成功 4. 常见问题：pip 报错、环境变量问题处理	10 分钟	命令行实操
三、实操任务 1：创建模拟外汇网站（25 分钟） 1. 项目结构：创建 project1 文件夹，新建 rates.csv 2. 编写 rates.csv：外汇汇率表头与 9 种货币数据 3. 编写 server.py： - 导入 flask→创建 app→定义路由 /→读取 CSV→拼接 HTML 表格→返回 - 启动：app.run (debug=True) 4. 运行服务器，浏览器访问 http://127.0.0.1:5000 查看效果	25 分钟	分步编码、逐行讲解
四、新知讲授：urllib.request 获取 HTML（10 分钟） 1. 核心步骤：打开网址→读取二进制→解码为字符串 2. 关键函数：urlopen ()、read ()、decode () 3. 编码要点：默认 utf-8，乱码需切换编码	10 分钟	原理 + 代码对应
五、实操任务 2：编写爬虫获取网页 HTML（15 分钟） 1. 保持服务器运行 2. 编写客户端代码：请求、读取、解码、打印 3. 运行查看输出 HTML，对比浏览器源码 4. 验证：HTML 包含完整汇率表格数据	15 分钟	联调实操
六、知识拓展：Flask 网页模板基础（10 分钟） 1. 模板作用：数据与视图分离 2. 目录结构：templates 文件夹 + html 文件 3. 变量渲染 {{title}}、循环 {% for %} 4. 简单示例：传递标题与汇率列表渲染表格	10 分钟	为下节课铺垫
七、课堂总结与答疑（5 分钟） 1. 核心要点：Flask 安装→服务器编写→爬虫获取 HTML 2. 关键流程：搭站→运行→请求→获取→解析	5 分钟	梳理知识脉络

3. 集中答疑			

课后作业：

完整复现本节课 Flask 外汇网站与爬虫代码，运行截图提交

拓展：在 rates.csv 添加 2 种货币，重新运行服务器并爬取

预习：1.4 Flask 模板循环、条件语句用法

（以下内容为选填，供线上课程及线上线下混合式课程填报）

教学资源列表与访问路径	备　注
中国大学 MOOC 网站： https://www.icourse163.org/course/ZIIT-1002925008 https://www.icourse163.org/course/ZIIT-1002924007	
在线教学过程（学生自主学习指引）	
学生在讨论区提问 教师在讨论区回答 学生提交作业，讲师讲评	

在线辅导与答疑安排	
学生观看网络视频 教师在线回答问题	
线上作业： 复习本次课程内容 预习下次课程内容	

教学单元	项目 1-3　GET 与 POST 方法访问网页				
周次		时数	2	授课班级	

教学目标

1. 知识目标

理解 GET、POST 两种请求方式的原理与区别

掌握 GET 传参格式、中文编码规则

掌握 POST 传参格式与 data 参数使用方法

掌握 Flask 服务器接收 GET/POST 参数的方法

理解路由 methods 参数作用

2. 技能目标

能编写 GET 客户端传递参数并获取指定外汇数据

能编写 POST 客户端传递参数并获取指定外汇数据

能编写支持 GET/POST 的 Flask 服务器

能完成客户端与服务器联调，实现精准查询

3. 情感目标

培养前后端数据交互思维

养成规范编码、异常处理、分步调试习惯

提升爬虫参数传递与接口对接实践能力

教学重点：

GET 传参格式与中文编码

POST 传参与 data 参数使用

Flask 接收参数：request.values.get ()

路由 methods=[GET,POST] 设置

教学难点

GET 与 POST 传参本质区别

中文参数编码处理

Flask 同时支持两种请求的配置

按条件筛选 CSV 数据并返回

实验仪器及教具：

计算机、因特网

教　学　过　程

教学过程	备注
一、课程导入（5 分钟） 回顾：Flask 搭建外汇网站，返回全部汇率 提出问题：如何只查询某一种货币汇率？ 引出任务：使用 GET/POST 向服务器传参，实现精准查询	复习旧知，引入新课
二、新知讲授：GET 方法传递数据（15 分钟） 1. GET 传参格式：URL? 参数名 = 值 2. 中文必须编码：urllib.parse.quote () 3. 客户端代码结构：导入库→输入货币→编码→请求→读取→解码→输出 4. 异常处理 try-except	理论 + 代码演示
三、实操任务 1：编写 GET 查询客户端（15 分钟） 1. 编写 GET 客户端程序 2. 运行服务器，再运行客户端 3. 输入 “美元”“欧元” 测试查询结果	学生实操
四、新知讲授：POST 方法传递数据（15 分钟） 1. POST 特点：参数放在 data 中，不显示在地址栏 2. 格式：data = 值.encode () 3. 与 GET 区别：地址干净、适合传敏感数据	对比教学
五、实操任务 2：编写 POST 查询客户端（10 分钟） 1. 编写 POST 客户端程序 2. 对比 GET 与 POST 运行效果一致	强化理解
六、新知讲授：服务器接收参数（15 分钟） 1. 路由配置：methods=["GET","POST"] 2. 获取参数：request.values.get (参数名，默认值) 3. 逻辑：保留表头 + 筛选匹配货币行 4. 完整服务器代码讲解	核心难点突破
七、综合联调与测试（3 分钟） 1. 启动服务器 2. 分别用 GET/POST 客户端查询不同货币 3. 观察返回结果是否正确	完整验证
八、课堂总结（2 分钟） 1. GET：地址传参，需中文编码 2. POST：data 传参，更安全 3. 服务器：接收参数→筛选数据→返回结果	梳理知识

（以下内容为选填，供线上课程及线上线下混合式课程填报）

教学资源列表与访问路径	备　注
中国大学 MOOC 网站： https://www.icourse163.org/course/ZIIT-1002925008 https://www.icourse163.org/course/ZIIT-1002924007	
在线教学过程（学生自主学习指引）	
学生在讨论区提问 教师在讨论区回答 学生提交作业，讲师讲评	
在线辅导与答疑安排	
学生观看网络视频 教师在线回答问题	
线上作业： 复习本次课程内容 预习下次课程内容	

教学单元	项目 1-4 正则表达式匹配数据				
周次		时数	2	授课班级	

教学目标

1. 知识目标

理解正则表达式的作用与适用场景

掌握 re 库的基本使用：import re、r""、re.search ()

掌握常用元字符：\d、+、*、?、.、 、\s、\w、[]、()、^、$ 等

理解从 HTML 表格中提取数据的匹配逻辑

2. 技能目标

能编写简单正则表达式匹配指定字符串

能使用 search () 匹配并截取目标内容

能循环匹配<td>标签，从 HTML 表格中提取汇率数据

能完成 “请求→获取→正则解析” 完整爬虫流程

3. 情感目标

建立数据解析思维，理解爬虫 “解析” 环节

培养严谨匹配、分步调试的编程习惯

提升从杂乱 HTML 中提取结构化数据的能力

教学重点：

re 库基础用法

常用正则元字符含义

re.search () 匹配与位置截取

循环提取<td>内数据

教学难点

正则元字符的区别：+、*、?

匹配位置截取：start ()、end ()

多层循环截取<tr>与<td>内容

HTML 表格数据的正则解析思路

实验仪器及教具：

计算机、因特网

教 学 过 程

教学过程	备注
一、课程导入（5 分钟） 1. 回顾：GET/POST 请求获取 HTML 源码 2. 问题：HTML 杂乱，如何提取有效汇率？ 3. 引出：正则表达式是爬虫数据解析核心工具	承上启下
二、新知讲授：正则表达式基础（20 分钟） 1. 作用：匹配、查找、提取字符串 2. 使用步骤：导包→写规则→search ()→获取结果 3. 重点规则讲解： ·\d 匹配数字 ·+/*/? 重复次数 ·. 任意字符 ·\s 空白、\w 字母数字下划线 ·[] 任选、() 分组、^/$ 开头结尾 4. 简单示例演示	讲练结合
三、实操任务 1：基础正则匹配练习（15 分钟） 1. 匹配数字、字母、指定格式 2. 对比 +、*、? 效果差异 3. 观察 match 对象的 span、start、end	巩固语法
四、新知讲授：正则爬取 HTML 数据（15 分钟） 1. 思路：先截<tr>内容→再循环截<td>内容 2. 关键函数：search ()、start ()、end () 3. 代码逻辑： ·截<tr>…</tr>中间内容 ·循环截<td>…</td>内容 ·输出并截断字符串	核心难点
五、实操任务 2：解析外汇表格数据（15 分钟） 1. 运行 Flask 服务器 2. 用 GET 获取 HTML 3. 用正则提取货币名称、汇率等字段 4. 输出格式化结果	完整爬虫流程

六、课堂总结（3 分钟） 1. 正则：规则 + search ()+ 位置截取 2. 解析 HTML：先截行，再截列 3. 完整爬虫：请求→获取→解析	梳理体系
七、答疑（2 分钟） 集中解答正则语法与代码问题	查漏补缺

（以下内容为选填，供线上课程及线上线下混合式课程填报）

教学资源列表与访问路径	备　注
中国大学 MOOC 网站： https://www.icourse163.org/course/ZIIT-1002925008 https://www.icourse163.org/course/ZIIT-1002924007	
在线教学过程（学生自主学习指引）	
学生在讨论区提问 教师在讨论区回答 学生提交作业，讲师讲评	
在线辅导与答疑安排	
学生观看网络视频 教师在线回答问题	
线上作业： 复习本次课程内容 预习下次课程内容	

教学单元	项目 1-5 爬取外汇网站数据				
周次		时数	2	授课班级	

教学目标

1. 知识目标

掌握 Flask 搭建模拟外汇网站完整流程

掌握正则表达式解析 HTML 表格数据方法

掌握 SQLite3 数据库创建、建表、插入、查询操作

理解 “请求→获取→解析→存储” 完整爬虫架构

掌握带属性标签（ ）的通用匹配方法

2. 技能目标

能独立编写并运行 Flask 外汇服务器

能使用 re 解析 HTML 并提取汇率数据

能编写爬虫类实现数据入库与控制台展示

能完成模拟网站 + 真实招行外汇网双项目爬取

3. 情感目标

建立完整爬虫工程思维

养成模块化、类封装、异常处理的编程习惯

提升从本地模拟到真实网站的迁移实战能力

教学重点：

完整爬虫流程：请求→解析→入库

SQLite3 数据库操作

正则循环提取<tr>、<td>数据

match () 函数匹配带属性标签

教学难点

多层循环截取 HTML 内容

带 class 属性的<td>标签通用匹配

数据类型转换与字段对应

真实网站 HTML 结构定位与过滤

实验仪器及教具：

计算机、因特网

教学过程	备注
一、课程导入（5 分钟）	承上启下，明确目标

1. 回顾：Flask 建站、GET/POST、正则解析 2. 提出本节总任务：完整爬虫项目落地 3. 明确双线任务：模拟外汇网站 → 真实招行外汇网	
二、任务 1：搭建模拟外汇网站（10 分钟） 1. 准备 rates.csv 数据文件 2. 编写 server.py：读取 CSV→生成 HTML 表格 3. 运行服务器，访问 http://127.0.0.1:5000 验证	快速复现环境
三、任务 2：数据库设计与封装（15 分钟） 1. SQLite3 建库建表：Currency、TSP、CSP、TBP、CBP 2. 类封装：openDB/closeDB/insertDB/show 3. 要点：主键约束、异常处理、事务提交	数据存储核心
四、任务 3：编写完整爬虫类（20 分钟） 1. 爬取流程：请求→解码→截取<tr>→循环取<td> 2. 跳过标题行，数据类型转 float 3. 逐行插入数据库并展示 4. 代码逐段讲解与调试	核心逻辑讲解
五、综合联调：模拟网站爬取（10 分钟） 1. 启动 server.py 2. 运行 spider.py 3. 查看控制台输出与数据库结果	完整流程验证
六、实战提升：爬取招行真实外汇网（15 分钟） 1. 定位： 2. 难点：带属性<td>匹配→编写 match () 函数 3. 增加 Time 字段，调整字段索引 4. 运行实战爬虫，观察真实数据	迁移实战
七、课堂总结（3 分钟） 1. 万能爬虫公式：请求→截取→循环解析→入库 2. 模拟站→真实站的思路迁移 3. 代码规范：类封装、函数单一职责	体系化梳理
八、答疑（2 分钟）	查漏补缺

（以下内容为选填，供线上课程及线上线下混合式课程填报）

教学资源列表与访问路径	备　注
中国大学 MOOC 网站： https://www.icourse163.org/course/ZIIT-1002925008 https://www.icourse163.org/course/ZIIT-1002924007	

在线教学过程（学生自主学习指引）	
学生在讨论区提问 教师在讨论区回答 学生提交作业，讲师讲评	
在线辅导与答疑安排	
学生观看网络视频 教师在线回答问题	
线上作业： 复习本次课程内容 预习下次课程内容	

教学单元	项目 2-1　BeautifulSoup 装载 HTML 文档				
周次		时数	2	授课班级	

教学目标

1. 知识目标

理解 BeautifulSoup 的作用与优势

掌握 Flask 模拟名言网站的搭建方法

掌握 bs4 与 lxml 的安装方法

理解 HTML 文档树结构

掌握 BeautifulSoup 加载、修复、格式化 HTML 的方法

2. 技能目标

能独立创建 Flask 模拟名言网站

能安装 bs4、lxml 库

能使用 BeautifulSoup 加载 HTML 并输出格式化树结构

能处理不完整 HTML 并观察自动修复效果

3. 情感目标

建立 “解析器优先于正则” 的高级爬虫思维

养成规范搭建模拟环境、分步调试的习惯

提升复杂 HTML 解析的自信心

教学重点：

Flask 模拟名言网站搭建

bs4 安装与导入

BeautifulSoup (html, 'lxml') 创建对象

soup.prettify () 格式化输出

教学难点

理解 HTML 文档树层级结构

BeautifulSoup 自动修复 HTML 规则

模拟网站与真实网站结构对应关系

实验仪器及教具：

计算机、因特网

教学过程	备注
一、课程导入（5 分钟） 1. 回顾：正则爬取简单网页，复杂网页难写规则 2. 引入：BeautifulSoup 高效解析器 3. 任务：搭建模拟名言站→解析 HTML→输出文档树	对比引入，明确方向
二、任务 1：创建模拟名言网站（15 分钟） 1. 项目结构：project2/templates/quotes.html 2. 编写 HTML 模板：名言、作者、标签样式 3. 编写 server.py：Flask 渲染模板 4. 运行服务器，访问 http://127.0.0.1:5000 验证	环境搭建
三、任务 2：安装 BeautifulSoup 环境（10 分钟） 1. 安装命令：pip install bs4 lxml 2. 导入测试：from bs4 import BeautifulSoup 3. 常见问题：pip 报错、解释器不匹配	环境配置
四、任务 3：BeautifulSoup 装载 HTML（20 分钟） 1. 步骤：请求→解码→创建 soup 对象 2. 核心代码：BeautifulSoup (html, 'lxml') 3. prettify () 输出标准文档树 4. 逐行分析树结构：html→head→body→div→span	核心知识点
五、任务 4：HTML 缺失修复实验（15 分钟） 1. 编写残缺 HTML 代码 2. 使用 BeautifulSoup 加载 3. 观察自动修复：补全标签、转小写、闭合标签 4. 理解解析器容错优势	加深理解
六、综合演示：爬取真实名言网站（10 分钟） 1. 访问 http://quotes.toscrape.com/ 2. 用同样代码解析真实网站 HTML 3. 对比模拟站与真实站结构	实战衔接
七、课堂总结与答疑（5 分钟） 1. 流程：搭建模拟站→安装库→加载→格式化→修复 2. 核心：文档树 + 自动修复 3. 下节预告：查找标签、提取内容	梳理体系

（以下内容为选填，供线上课程及线上线下混合式课程填报）

教学资源列表与访问路径	备　注

中国大学 MOOC 网站： https://www.icourse163.org/course/ZIIT-1002925008 https://www.icourse163.org/course/ZIIT-1002924007	
在线教学过程（学生自主学习指引）	
学生在讨论区提问 教师在讨论区回答 学生提交作业，讲师讲评	
在线辅导与答疑安排	
学生观看网络视频 教师在线回答问题	
线上作业： 复习本次课程内容 预习下次课程内容	

教学单元	项目 2-2　BeautifulSoup 查找 HTML 元素				
周次		时数	2	授课班级	

教学目标

1. 知识目标

理解 BeautifulSoup 文档树查找原理

掌握 find ()/find_all () 用法与参数含义

掌握获取元素文本 text 与属性 attr 的方法

理解递归查找 recursive 作用

掌握自定义过滤函数高级查找

2. 技能目标

能使用 find () 查找单个指定元素

能使用 find_all () 批量获取元素列表

能提取元素文本与属性值

能链式查找精确定位数据

能用自定义函数实现复杂条件查找

3. 情感目标

建立结构化解析思维

养成先定位、再提取、后验证的爬虫习惯

提升复杂网页数据提取能力

教学重点：

find () 单个查找

find_all () 批量查找

获取文本 .text

获取属性［属性名］

按 class/attrs 精准筛选

教学难点

递归查找 recursive=True/False 区别

多级元素链式嵌套查找

自定义过滤函数编写

class 属性以列表形式判断

实验仪器及教具：

计算机、因特网

教学过程	备注
一、课程导入（5 分钟） 1. 回顾：BeautifulSoup 加载 HTML、文档树、prettify 格式化 2. 提出问题：如何从树中精准取出名言、作者、标签？ 3. 本节任务：学习四大核心查找方法	承上启下
二、知识点 1：find () 查找单个元素（15 分钟） 1. 用法：find (标签名，attrs={}, recursive=True) 2. 返回：第一个匹配元素 或 None 3. 示例：查找 span、a、div.tags 4. 链式查找：find ().find () 5. recursive 参数：True 全树查找，False 只查子层	讲练结合
三、知识点 2：获取元素文本与属性（10 分钟） 1. 文本：元素.text 2. 属性：元素 ["href"]、元素 ["class"] 3. 示例：提取名言、作者、链接地址	实操提取
四、知识点 3：find_all () 批量查找（15 分钟） 1. 用法：返回列表，可遍历、索引 [-1] 2. 示例：查找所有 a、所有 quote 块、所有标签 3. 双层循环：批量提取所有名言与标签	重点训练
五、知识点 4：高级查找（自定义过滤函数）（15 分钟） 1. 函数规则：传入 tag，返回 True/False 2. 常用判断：tag.name、tag.text、tag.has_attr () 3. 示例：查找文本为 thinking 的 a 标签、查找所有 tag 标签、含 page 的链接	难点突破
六、综合实操：完整爬取模拟名言站（15 分钟） 1. 启动模拟网站 2. 依次提取：2 条名言、作者、链接、所有标签 3. 控制台格式化输出结果	完整应用
七、课堂总结与答疑（5 分钟） 1. 查找套路：先块→再行→再字段 2. 方法对比：find () 取一个，find_all () 取全部 3. 高级：自定义函数应对复杂规则	体系梳理

（以下内容为选填，供线上课程及线上线下混合式课程填报）

教学资源列表与访问路径	备　注

中国大学 MOOC 网站： https://www.icourse163.org/course/ZIIT-1002925008 https://www.icourse163.org/course/ZIIT-1002924007	
在线教学过程（学生自主学习指引）	
学生在讨论区提问 教师在讨论区回答 学生提交作业，讲师讲评	
在线辅导与答疑安排	
学生观看网络视频 教师在线回答问题	
线上作业： 复习本次课程内容 预习下次课程内容	

教学单元	项目 2-3 BeautifulSoup 遍历文档元素				
周次		时数	2	授课班级	

教学目标

1. 知识目标

理解 HTML 文档树节点关系（父、子、子孙、兄弟）

掌握 parent /children/descendants /sibling 遍历方法

区分 Tag、NavigableString、Comment 三种节点类型

2. 技能目标

能获取任意元素的所有父节点

能遍历直接子节点并过滤空白文本

能递归遍历所有子孙节点提取数据

能通过兄弟节点批量获取并列标签

能识别并处理文本节点与注释节点

3. 情感目标

建立完整的 HTML 文档树结构思维

养成 “无特征标签用遍历、有特征标签用查找” 的解题习惯

提升复杂网页无标识数据提取能力

教学重点：

find () 单个查找

find_all () 批量查找

获取文本 .text

获取属性［属性名］

按 class/attrs 精准筛选

教学难点

递归查找 recursive=True/False 区别

多级元素链式嵌套查找

自定义过滤函数编写

class 属性以列表形式判断

实验仪器及教具：

计算机、因特网

教学过程	备注
一、课程导入（5 分钟） 1. 回顾：BeautifulSoup find /find_all 按特征查找 2. 问题：遇到**无 class、无 id、无属性**的标签怎么提取？ 3. 引出：用**节点遍历**实现精准定位	承上启下
二、知识点 1：获取父节点 parent（10 分钟） 1. 用法：tag.parent 2. 层级：元素 → 父 → 祖父 → … → html → [document] 3. 案例：获取第一个 span 的所有父节点	快速入门
三、知识点 2：获取直接子节点 children（15 分钟） 1. 用法：for child in tag.children 2. 节点类型：Tag / NavigableString / Comment 3. 文本节点用 .string 获取，需过滤 \n 与空白 4. 案例：遍历作者信息 span 的所有直接子节点	重点讲解
四、知识点 3：获取所有子孙节点 descendants（15 分钟） 1. 用法：for d in tag.descendants 2. 作用：递归遍历所有下级节点（子、孙、重孙…） 3. 案例：提取.tags 下所有文本与标签	深度遍历
五、知识点 4：获取兄弟节点（15 分钟） 1. 前兄弟：previous_sibling 2. 后兄弟：next_sibling 3. 技巧：循环跳过文本节点，只取 Tag 元素 4. 案例：批量获取所有标签<a>	难点突破
六、综合实操：遍历提取完整名言数据（15 分钟） 1. 启动模拟网站 2. 用遍历方式提取：名言、作者、所有标签 3. 格式化输出，过滤空白与换行 4. 对比 find 与遍历两种方法优劣	巩固应用
七、课堂总结与答疑（5 分钟） 1. 遍历四件套：父、子、子孙、兄弟 2. 文本节点必须过滤 \n 3. 遍历适合无特征标签，查找适合有特征标签	体系梳理

（以下内容为选填，供线上课程及线上线下混合式课程填报）

教学资源列表与访问路径	备　注

中国大学 MOOC 网站： https://www.icourse163.org/course/ZIIT-1002925008 https://www.icourse163.org/course/ZIIT-1002924007	
在线教学过程（学生自主学习指引）	
学生在讨论区提问 教师在讨论区回答 学生提交作业，讲师讲评	
在线辅导与答疑安排	
学生观看网络视频 教师在线回答问题	
线上作业： 复习本次课程内容 预习下次课程内容	

教学单元	项目 2-4　BeautifulSoup 使用 CSS 语法查找				
周次		时数	2	授课班级	

教学目标

1. 知识目标

理解 CSS 选择器在 BeautifulSoup 中的作用与优势

掌握 select () 与 select_one () 的用法区别

掌握层级、属性、子节点、兄弟节点 CSS 写法

理解 ^=、$=、*= 模糊匹配规则

2. 技能目标

能使用 CSS 选择器快速定位网页元素

能通过 class、id、属性精准筛选内容

能使用子孙、直接子节点、兄弟节点写法

能使用 select_one () 提取单个元素

3. 情感目标

提升高效爬虫编写思维，简化复杂查找逻辑

养成前端结构与爬虫解析结合的编程习惯

教学重点：

select () /select_one () 用法

.class、#id、标签 [属性] 写法

空格（子孙）、>（直接子节点）

~（后续兄弟）、+（紧邻兄弟）

教学难点

CSS 层级关系区分（子孙 / 直接子节点 / 兄弟）

属性模糊匹配（^= $= *=）

nth-of-type (n) 索引定位

实验仪器及教具：

计算机、因特网

教学过程	备注
一、课程导入（5 分钟） 1. 回顾：find /find_all 节点遍历 2. 引入：CSS 选择器更简洁、前端通用 3. 任务：用 CSS 语法快速提取名言、作者、标签	对比引入
二、知识点 1：select () 与 select_one ()（10 分钟） 1. soup.select (css) → 返回列表 2. soup.select_one (css) → 返回第一个元素 3. 简写：.quote 代替 class='quote'	基础入门
三、知识点 2：CSS 层级查找（15 分钟） 1. 空格：子孙节点 div span a 2. > ：直接子节点 div > span > a 3. 案例：提取名言块中的作者链接	重点讲解
四、知识点 3：属性匹配与模糊查询（15 分钟） 1. [属性]、[属性 = 值] 2. ^= 开头、$= 结尾、*= 包含 3. 案例：提取所有包含 page 的 href 链接	高频实用
五、知识点 4：兄弟节点查找（15 分钟） 1. + ：紧邻下一个兄弟 2. ~ ：后面所有兄弟 3. 案例：批量提取标签栏所有 a 标签	难点突破
六、综合实操：CSS 语法爬取全部名言（15 分钟） 1. 启动模拟网站 2. 用 CSS 提取：名言、作者、所有标签 3. 对比 find 写法与 CSS 写法代码量 4. 输出格式化结果	巩固应用
七、课堂总结与答疑（5 分钟） 1. CSS 选择器 = 简洁高效 2. 层级：空格 (子孙) >(直接子) ~(兄弟) +(紧邻) 3. 模糊匹配：^ $ *	梳理体系

（以下内容为选填，供线上课程及线上线下混合式课程填报）

教学资源列表与访问路径	备　注
中国大学 MOOC 网站： https://www.icourse163.org/course/ZIIT-1002925008 https://www.icourse163.org/course/ZIIT-1002924007	

在线教学过程（学生自主学习指引）	
学生在讨论区提问 教师在讨论区回答 学生提交作业，讲师讲评	
在线辅导与答疑安排	
学生观看网络视频 教师在线回答问题	
线上作业： 复习本次课程内容 预习下次课程内容	

教学单元	项目 2-5　爬取名言网站数据				
周次		时数	2	授课班级	

1. 知识目标

掌握 Flask 模拟名言网站完整搭建流程

掌握双表结构数据库设计（名言表 + 标签表）

掌握 CSS 选择器批量提取数据

掌握分页爬取与自动停止逻辑

2. 技能目标

能独立搭建模拟名言网站

能使用 BeautifulSoup 爬取名言、作者、链接、标签

能完成 SQLite3 双表创建、插入、关联查询

能实现多页爬取并自动判断最后一页停止

3. 情感目标

建立完整爬虫项目工程思维

养成模块化、类封装、健壮性编程习惯

提升从模拟环境到真实网站的迁移能力

教学重点：

模拟网站搭建

BeautifulSoup CSS 选择器数据提取

SQLite3 双表设计与外键关联

分页循环爬取与结束判断

教学难点

数据库 ID 关联存储标签数据

分页爬取循环控制

真实网站结构适配与异常处理

实验仪器及教具：

计算机、因特网

教学过程	备注
一、课程导入（5 分钟） 1. 回顾：BeautifulSoup 查找、遍历、CSS 选择器 2. 本节任务：完成**模拟站爬取**+**真实站多页爬取**两大项目 3. 目标：爬取→解析→入库→展示 全流程打通	项目式驱动
二、任务 1：搭建模拟名言网站（10 分钟） 1. 项目结构：templates/quotes.html + server.py 2. 运行服务器，访问 http://127.0.0.1:5000 3. 分析页面结构：.quote .text .author .tags	环境准备
三、任务 2：数据库设计（10 分钟） 1. 主表 quotes：ID、名言、作者、链接 2. 从表 tags：ID、qID (关联)、标签、标签链接 3. 函数：openDB、insertQuote、insertTag、closeDB	数据存储
四、任务 3：模拟站爬虫编写与运行（20 分钟） 1. 使用 select/select_one 提取内容 2. 循环爬取每条名言 3. 插入主表→获取 ID→插入从表 4. show () 函数格式化输出	核心逻辑
五、任务 4：实战项目 —— 真实名言网站爬取（25 分钟） 1. 分析真实站结构：分页 URL 规则 2. 循环构造 page/n/ 3. 判断 next 按钮，无则停止 4. 爬取 10 页共 100 条数据入库 5. 菜单式交互：爬取 / 显示 / 退出	实战提升
六、课堂总结（5 分钟） 1. 万能爬虫公式：请求→解析→存储→展示 2. 模拟站→真实站只需修改 URL 与选择器 3. 分页爬取：循环 + 判断结束标志	体系梳理
七、答疑（5 分钟）	解决问题

（以下内容为选填，供线上课程及线上线下混合式课程填报）

教学资源列表与访问路径	备　注
中国大学 MOOC 网站： https://www.icourse163.org/course/ZIIT-1002925008 https://www.icourse163.org/course/ZIIT-1002924007	

在线教学过程（学生自主学习指引）	
学生在讨论区提问 教师在讨论区回答 学生提交作业，讲师讲评	
在线辅导与答疑安排	
学生观看网络视频 教师在线回答问题	
线上作业： 复习本次课程内容 预习下次课程内容	

教学单元	项目 3-1　lxml 装载 HTML 文档				
周次		时数	2	授课班级	

教学目标

1. 知识目标

理解天气网站爬取项目的整体任务

掌握 Flask 搭建模拟天气网站方法

掌握 lxml 库的作用与 HTML 装载流程

理解 lxml 自动修复残缺 HTML 的特性

对比 lxml 与 BeautifulSoup 解析特点

2. 技能目标

能独立搭建模拟天气网站并运行

能安装并导入 lxml 库

能使用 etree.HTML () 加载 HTML

能使用 tostring () 格式化输出文档树

能处理并观察自动修复效果

3. 情感目标

建立“先模拟、后实战”的爬虫工程思维

养成多解析器对比学习习惯

提升高效解析 HTML 的能力

教学重点：

模拟天气网站搭建

lxml 安装与导入

etree.HTML () 装载 HTML

etree.tostring () 格式化输出

HTML 自动修复特性

教学难点

lxml 对残缺 HTML 的自动修复规则

HTML 文档树结构理解

编码与格式化输出设置

实验仪器及教具：

计算机、因特网

教学过程	备注
一、课程导入（5 分钟） 1. 介绍项目：爬取中国天气网 7 天天气数据 2. 学习路线：先搭模拟天气站 → 学习 lxml 解析 → 再爬真实站 3. 本节目标：搭建模拟站 + 掌握 lxml 装载 HTML	项目引入
二、任务 1：创建模拟天气网站（15 分钟） 1. 项目结构：project3/templates/weather.html 2. 编写 HTML 模板：日期、天气、温度、风力 3. 编写 server.py：Flask 渲染模板 4. 运行访问 http://127.0.0.1:5000	环境搭建
三、任务 2：lxml 库安装与基础使用（10 分钟） 1. 安装：pip install lxml 2. 导入：from lxml import etree 3. 核心：root = etree.HTML (html) 4. 对比 BeautifulSoup 解析方式	快速上手
四、任务 3：lxml 格式化输出与 HTML 修复（20 分钟） 1. tostring () 参数：unicode、pretty_print、html 2. 实验：加载残缺 HTML → 观察自动补全 3. 效果：补全 html/head/body、标签小写、闭合标签	核心知识点
五、任务 4：解析模拟天气网站 HTML（20 分钟） 1. 请求网页获取 HTML 2. lxml 加载并构建 DOM 树 3. 格式化输出完整结构 4. 分析 li、h5、wea、tem、win 等结构位置	实操巩固
六、课堂总结与答疑（10 分钟） 1. lxml 流程：请求→HTML→etree.HTML→格式化 2. 核心优势：速度快、自动修复、支持 XPath 3. 下节预告：XPath 提取天气数据	梳理体系

（以下内容为选填，供线上课程及线上线下混合式课程填报）

教学资源列表与访问路径	备　注
中国大学 MOOC 网站： https://www.icourse163.org/course/ZIIT-1002925008 https://www.icourse163.org/course/ZIIT-1002924007	
在线教学过程（学生自主学习指引）	

学生在讨论区提问 教师在讨论区回答 学生提交作业，讲师讲评	
在线辅导与答疑安排	
学生观看网络视频 教师在线回答问题	
线上作业： 复习本次课程内容 预习下次课程内容	

教学单元	项目 3-2 P xpath 查找 HTML 元素				
周次		时数	2	授课班级	

1. 知识目标
理解 lxml 解析后的 Element 对象类型
掌握 XPath 路径语法：///. 的区别
掌握文本、属性、条件、位置、父子兄弟节点写法
理解 XPath 与 BeautifulSoup 查找方式的差异
2. 技能目标
能使用 XPath 精准提取日期、天气、温度、风力
能使用属性筛选、contains 模糊匹配
能获取文本 text () 和属性 @attr
能遍历父节点、子节点、兄弟节点
3. 情感目标
建立高效解析思维，掌握工业级主流爬虫写法
养成结构化定位数据的习惯

教学重点：
XPath 路径：/ 直接子节点 // 所有子孙
获取文本：text ()
获取属性：@属性名
条件筛选：[@class='xx']、contains ()
位置：position ()=1

教学难点
// 与 / 的区别与使用场景
多条件复合筛选与 contains 匹配

兄弟节点、父节点、子节点遍历

实验仪器及教具：

计算机、因特网

教学过程	备注
一、课程导入（5 分钟） 1. 回顾：lxml 装载 HTML、构建文档树 2. 引入：XPath 是工业级最常用定位语言 3. 目标：用 XPath 提取天气网站所有数据	承上启下
二、知识点 1：lxml 元素与 XPath 返回类型（10 分钟） 1. root：etree._Element 2. xpath () 返回列表 3. 列表中每个元素是 Element 对象 4. 示例：//ul 返回结果分析	理解基础
三、知识点 2：XPath 路径语法（10 分钟） 1. /：从根节点 / 直接子节点查 2. //：全文档查找所有子孙 3. .：从当前节点往下查 4. 示例对比：//ul 和 /ul	核心语法
四、知识点 3：提取文本与属性（10 分钟） 1. 获取文本：/text () 或 .text 2. 获取属性：/@class /@title /@href 3. 示例：提取日期、天气描述	高频实用
五、知识点 4：条件筛选与模糊匹配（15 分钟） 1. 精准匹配：[@class='sky'] 2. 模糊匹配：contains (@class,'sky') 3. 多条件：and、or、[] 叠加 4. 位置筛选：position ()=1	重点难点
六、知识点 5：父子兄弟节点遍历（15 分钟） 1. 父节点：parent:: *2. 兄节点：preceding-sibling::* 3. 弟节点：following-sibling::* 4. 孩子节点：getchildren ()	进阶技能
七、综合实操：爬取全部 3 天天气数据（10 分钟） 1. 提取：日期、天气、温度、风力 2. 格式化输出 3. 对比 BeautifulSoup 写法	巩固应用
八、课堂总结与答疑（5 分钟） 1. XPath 万能公式：路径 + 条件 + 取值 2. 优先用 // + contains 定位 3. 文本用 text ()，属性用 @xxx	梳理体系

（以下内容为选填，供线上课程及线上线下混合式课程填报）

<table>
<tr><td>教学资源列表与访问路径</td><td>备　注</td></tr>
<tr><td>中国大学 MOOC 网站：
https://www.icourse163.org/course/ZIIT-1002925008
https://www.icourse163.org/course/ZIIT-1002924007</td><td></td></tr>
<tr><td>在线教学过程（学生自主学习指引）</td><td rowspan="2"></td></tr>
<tr><td>学生在讨论区提问
教师在讨论区回答
学生提交作业，讲师讲评</td></tr>
<tr><td>在线辅导与答疑安排</td><td rowspan="2"></td></tr>
<tr><td>学生观看网络视频
教师在线回答问题</td></tr>
<tr><td colspan="2">线上作业：
复习本次课程内容
预习下次课程内容</td></tr>
</table>

教学单元	项目 3-3　爬取天气网站数据				
周次		时数	2	授课班级	

教学目标

1. 知识目标

掌握模拟天气网站完整搭建流程

掌握 XPath 提取日期、天气、温度、风向、风力

掌握 SQLite3 数据库表设计与数据存储

掌握真实网站结构分析与代码迁移方法

掌握温度数据可视化绘图原理

2. 技能目标

能搭建并运行模拟天气网站

能使用 XPath 批量爬取 7 天天气数据

能完成数据库创建、插入、查询

能爬取中国天气网真实数据

能使用 matplotlib 绘制温度曲线图

3. 情感目标

建立工业级爬虫项目开发思维

养成结构化解析、模块化编程习惯

提升从模拟到真实网站的迁移实战能力

教学重点：

XPath 精准提取天气数据

SQLite3 数据库设计与操作

真实网站 HTML 结构分析

菜单式交互爬虫程序

教学难点

多级 XPath 定位与列表取值

温度字符串处理与数值提取

真实网站 class 多值模糊匹配

实验仪器及教具：

计算机、因特网

教学过程	备注
一、课程导入（5 分钟）	项目驱动

1. 回顾：lxml 与 XPath 基础语法 2. 本节任务：**模拟站爬取+真实天气网爬取+绘图展示** 3. 目标：打通请求→解析→存储→展示全流程	
二、任务 1：搭建模拟天气网站（10 分钟） 1. 项目结构：templates/weather.html + server.py 2. 运行服务器，访问 http://127.0.0.1:5000 3. 分析结构：ul.clearfix → li → h5 /p.wea/p.tem/p.win	环境准备
三、任务 2：数据库设计（10 分钟） 1. 库：weathers.db 2. 表：weathers（城市、日期、天气、最高温、最低温、风向、风力） 3. 函数：openDB、insert、closeDB、show	数据存储
四、任务 3：模拟站爬虫编写与运行（20 分钟） 1. 使用 XPath 循环提取每日数据 2. 批量插入数据库 3. 运行并输出 7 天天气	核心逻辑
五、任务 4：实战项目 —— 爬取真实天气网站（25 分钟） 1. 分析中国天气网结构：ul [contains (@class,'t')] 2. 修改 XPath 适配真实页面（h1 替换 h5） 3. 菜单式交互：爬取 / 显示 / 绘图 / 退出 4. 正则提取温度数字并绘制曲线图	实战提升
六、课堂总结（5 分钟） 1. 爬虫万能流程：请求→解析→存储→展示 2. 模拟站→真实站：只改 URL 与少量 XPath 3. 数据可视化增强项目效果	体系梳理
七、答疑（5 分钟）	解决问题

（以下内容为选填，供线上课程及线上线下混合式课程填报）

教学资源列表与访问路径	备　注
中国大学 MOOC 网站： https://www.icourse163.org/course/ZIIT-1002925008 https://www.icourse163.org/course/ZIIT-1002924007	
在线教学过程（学生自主学习指引）	

学生在讨论区提问 教师在讨论区回答 学生提交作业，讲师讲评	
在线辅导与答疑安排	
学生观看网络视频 教师在线回答问题	
线上作业： 复习本次课程内容 预习下次课程内容	

教学单元	项目 4-1　网站树的爬取路径				
周次		时数	2	授课班级	

教学目标

1. 知识目标

掌握模拟天气网站完整搭建流程

掌握 XPath 提取日期、天气、温度、风向、风力

掌握 SQLite3 数据库表设计与数据存储

掌握真实网站结构分析与代码迁移方法

掌握温度数据可视化绘图原理

2. 技能目标

能搭建并运行模拟天气网站

能使用 XPath 批量爬取 7 天天气数据

能完成数据库创建、插入、查询

能爬取中国天气网真实数据

能使用 matplotlib 绘制温度曲线图

3. 情感目标

建立工业级爬虫项目开发思维

养成结构化解析、模块化编程习惯

提升从模拟到真实网站的迁移实战能力

教学重点：

XPath 精准提取天气数据

SQLite3 数据库设计与操作

真实网站 HTML 结构分析

菜单式交互爬虫程序

教学难点

多级 XPath 定位与列表取值

温度字符串处理与数值提取

真实网站 class 多值模糊匹配

实验仪器及教具：

计算机、因特网

教学过程	备注
一、课程导入（5 分钟） 1. 回顾：lxml 与 XPath 基础语法 2. 本节任务：**模拟站爬取+真实天气网爬取+绘图展示** 3. 目标：打通请求→解析→存储→展示全流程	项目驱动
二、任务 1：搭建模拟天气网站（10 分钟） 1. 项目结构：templates/weather.html + server.py 2. 运行服务器，访问 http://127.0.0.1:5000 3. 分析结构：ul.clearfix → li → h5 /p.wea/p.tem/p.win	环境准备
三、任务 2：数据库设计（10 分钟） 1. 库：weathers.db 2. 表：weathers（城市、日期、天气、最高温、最低温、风向、风力） 3. 函数：openDB、insert、closeDB、show	数据存储
四、任务 3：模拟站爬虫编写与运行（20 分钟） 1. 使用 XPath 循环提取每日数据 2. 批量插入数据库 3. 运行并输出 7 天天气	核心逻辑
五、任务 4：实战项目 —— 爬取真实天气网站（25 分钟） 1. 分析中国天气网结构：ul [contains (@class,'t')] 2. 修改 XPath 适配真实页面（h1 替换 h5） 3. 菜单式交互：爬取 / 显示 / 绘图 / 退出 4. 正则提取温度数字并绘制曲线图	实战提升
六、课堂总结（5 分钟） 1. 爬虫万能流程：请求→解析→存储→展示 2. 模拟站→真实站：只改 URL 与少量 XPath 3. 数据可视化增强项目效果	体系梳理
七、答疑（5 分钟）	解决问题

（以下内容为选填，供线上课程及线上线下混合式课程填报）

教学资源列表与访问路径	备　注
中国大学 MOOC 网站： https://www.icourse163.org/course/ZIIT-1002925008 https://www.icourse163.org/course/ZIIT-1002924007	
在线教学过程（学生自主学习指引）	

学生在讨论区提问 教师在讨论区回答 学生提交作业，讲师讲评	
在线辅导与答疑安排	
学生观看网络视频 教师在线回答问题	
线上作业： 复习本次课程内容 预习下次课程内容	

教学单元	项目 4-2　爬取多页面网站数据				
周次		时数	2	授课班级	

教学目标

1. 知识目标

理解旅游网站多页爬取项目需求

理解网站结构可抽象为树结构

掌握深度优先（DFS） 爬取原理（递归 / 栈）

掌握广度优先（BFS） 爬取原理（队列）

理解两种爬取策略的顺序与适用场景

2. 技能目标

能搭建 Flask 分页模拟网站

能编写递归爬虫实现深度优先爬取

能使用栈 / 队列实现非递归深度 / 广度爬取

能自动追踪链接并去重访问

3. 情感目标

建立全站爬取工程思维

养成去重、递归、非递归规范编程习惯

提升复杂网站自动遍历能力

教学重点：

网站树结构理解

递归深度优先爬取

栈实现深度优先

队列实现广度优先

URL 去重机制

教学难点

递归爬取流程与回溯原理

深度与广度爬取顺序区别

链接拼接与去重逻辑

实验仪器及教具：

计算机、因特网

教学过程	备注
一、课程导入（5 分钟） 1. 回顾：单页爬取、BeautifulSoup、XPath 2. 新问题：如何爬取**多页、带链接、全站数据**？ 3. 引出：网站 = 树；爬取 = 遍历；两种策略：DFS/BFS	引入新课
二、知识点 1：网站树结构（10 分钟） 1. 演示：books.html 多级链接结构 2. 结构：首页→分类→子页面 3. 抽象：**网站 = 多叉树** 4. 爬取本质：遍历这棵树	建立模型
三、知识点 2：递归深度优先爬取（15 分钟） 1. 原理：一路到底，再回溯 2. 步骤：访问→提取链接→递归新链接 3. 全局 urls 列表去重 4. 代码演示与结果分析	核心方法
四、知识点 3：栈实现非递归深度优先（15 分钟） 1. 栈：后进先出 2. 入栈、出栈、判空 3. 逆序入栈保证顺序 4. 运行结果与递归完全一致	等价实现
五、知识点 4：队列实现广度优先（15 分钟） 1. 队列：先进先出 2. 按层遍历、一层一层爬 3. 适合获取最新 / 首页优先数据 4. 结果顺序对比：层序输出	重点对比
六、综合演示：模拟旅游网站爬取（10 分钟） 1. 启动 Flask 模拟网站 2. 分别用 DFS/BFS 爬取标题 3. 观察输出顺序差异 4. 讲解适用场景	巩固应用
七、课堂总结与答疑（5 分钟） 1. 深度优先：递归 / 栈，纵向深入 2. 广度优先：队列，横向按层 3. 必须 URL 去重	体系梳理

（以下内容为选填，供线上课程及线上线下混合式课程填报）

教学资源列表与访问路径	备　注

中国大学 MOOC 网站： https://www.icourse163.org/course/ZIIT-1002925008 https://www.icourse163.org/course/ZIIT-1002924007	
在线教学过程（学生自主学习指引）	
学生在讨论区提问 教师在讨论区回答 学生提交作业，讲师讲评	
在线辅导与答疑安排	
学生观看网络视频 教师在线回答问题	
线上作业： 复习本次课程内容 预习下次课程内容	

教学单元	项目 4-3　Python 实现多线程				
周次		时数	2	授课班级	

教学目标

1. 知识目标

理解多线程的作用与优势

掌握守护线程（Daemon）的概念

掌握线程等待 join () 的用法

掌握线程锁解决资源竞争问题

2. 技能目标

能够使用 threading 创建线程

能够设置守护线程

能够使用 join () 控制线程执行顺序

能够使用 RLock 保证多线程安全访问数据

3. 情感目标

建立并发编程思想

养成数据安全、规范编码的意识

教学重点：

threading.Thread 创建与启动

setDaemon () 守护线程

join () 线程等待

threading.RLock 线程锁

教学难点

守护线程与主线程生命周期

join () 阻塞原理

多线程资源竞争与锁机制

实验仪器及教具：

计算机、因特网

教学过程	备注
一、课程导入（5 分钟） 1. 回顾：单线程爬虫爬取速度慢 2. 引入：多线程可同时执行多个任务，大幅提高爬取效率 3. 本节目标：学习 Python 多线程基础用法	引入新课
二、知识点 1：线程创建与启动（15 分钟） 1. 导入 threading 库 2. 创建线程：Thread (target = 函数名) 3. 启动线程：start () 4. 示例演示：创建子线程执行打印任务	基础用法
三、知识点 2：守护线程 Daemon（15 分钟） 1. 非守护线程：主线程结束，子线程继续执行 2. 守护线程：主线程结束，子线程立即结束 3. 设置方法：setDaemon (True) 4. 对比两个示例运行结果	重点讲解
四、知识点 3：线程等待 join ()（15 分钟） 1. join () 作用：等待指定线程执行完毕再继续 2. 主线程等待子线程 3. 线程嵌套等待 4. 示例演示顺序执行效果	核心控制
五、知识点 4：线程锁与资源安全（15 分钟） 1. 问题：多线程同时修改数据导致混乱 2. 解决方法：线程锁 RLock 3. 加锁：acquire () 4. 释放：release () 5. 示例演示安全排序	难点突破
六、综合实操与总结（10 分钟） 1. 运行多线程示例 2. 观察并发执行效果 3. 总结：创建→守护→等待→锁 4. 答疑	巩固提升
七、课堂总结（5 分钟） 1. 多线程提高程序效率 2. 守护线程控制生命周期 3. join () 控制执行顺序 4. 锁保证数据安全	梳理体系

（以下内容为选填，供线上课程及线上线下混合式课程填报）

教学资源列表与访问路径	备　注
中国大学 MOOC 网站： https://www.icourse163.org/course/ZIIT-1002925008 https://www.icourse163.org/course/ZIIT-1002924007	
在线教学过程（学生自主学习指引）	
学生在讨论区提问 教师在讨论区回答 学生提交作业，讲师讲评	
在线辅导与答疑安排	
学生观看网络视频 教师在线回答问题	
线上作业： 复习本次课程内容 预习下次课程内容	

教学单元	项目 4-4　爬取旅游网站图像				
周次		时数	2	授课班级	

教学目标

1. 知识目标

掌握单线程与多线程图片爬取的效率差异

理解多线程图片下载原理与优势

掌握数据库存储旅游信息结构设计

掌握分页递归爬取与详情页内容获取

掌握真实网站请求头伪装技术

2. 技能目标

能搭建带延迟的模拟旅游网站

能编写多线程图片批量下载程序

能将旅游信息（ID、标题、日期、内容、图片）存入数据库

能爬取中国日报真实旅游网站全部数据

3. 情感目标

建立高效并发爬虫思维

养成工程化、模块化、可视化项目开发习惯

提升真实网站爬取实战能力

教学重点：

多线程图片下载（threading）

分页递归爬取

详情页内容爬取

SQLite 数据库存储旅游信息

真实网站 headers 请求头

教学难点

多线程并发安全与 join 等待

图片 URL 拼接与本地保存

真实网站结构解析与翻页识别

实验仪器及教具：

计算机、因特网

教学过程	备注
一、课程导入（5 分钟） 1. 回顾：多线程、分页爬取 2. 问题：单线程下载图片慢、易阻塞 3. 本节任务：**多线程爬图 + 数据库存储 + 真实网站实战**	项目驱动
二、任务 1：搭建带延迟的模拟旅游网站（10 分钟） 1. 修改模板：图片路径指向 /getImage 2. 服务器增加随机延迟 0-3 秒 3. 运行验证：图片加载缓慢	模拟慢网络环境
三、任务 2：单线程与多线程爬图对比（15 分钟） 1. 单线程：依次下载，耗时约 200 秒 + 2. 多线程：同时启动下载线程，耗时仅 2－3 秒 3. 核心代码：threading.Thread + 线程列表 + join	效率对比演示
四、任务 3：综合项目 —— 数据入库 + 图片下载（20 分钟） 1. 数据库表设计：ID、标题、日期、内容、扩展名 2. 爬取流程：分页→列表→详情→内容→入库 3. 多线程下载图片到文件夹 4. 菜单交互程序：爬取 / 显示 / 退出	核心综合项目
五、任务 4：实战项目 —— 爬取中国日报旅游网（20 分钟） 1. 分析真实网站结构 2. 添加 headers 请求头伪装浏览器 3. 解析列表、标题、日期、图片、详情内容 4. 自动翻页爬取 111 页全部数据	真实网站实战
六、运行演示与总结（10 分钟） 1. 运行模拟项目：71 条数据 + 图片快速下载 2. 运行真实项目：2200 + 条数据入库 3. 总结：多线程提速、递归翻页、结构解析、数据存储	成果展示

（以下内容为选填，供线上课程及线上线下混合式课程填报）

教学资源列表与访问路径	备　注
中国大学 MOOC 网站： https://www.icourse163.org/course/ZIIT-1002925008 https://www.icourse163.org/course/ZIIT-1002924007	
在线教学过程（学生自主学习指引）	

学生在讨论区提问 教师在讨论区回答 学生提交作业，讲师讲评	
在线辅导与答疑安排	
学生观看网络视频 教师在线回答问题	
线上作业： 复习本次课程内容 预习下次课程内容	

教学单元	项目 5-1　scrapy 创建爬虫程序				
周次		时数	2	授课班级	

教学目标

1. 知识目标

理解图书爬取项目需求与分页结构

掌握 Scrapy 框架作用、安装与项目结构

理解异步爬虫、回调函数、yield 生成器原理

掌握 Scrapy 请求与响应流程

2. 技能目标

能搭建 Flask 图书模拟网站

能安装 Scrapy 并创建项目

能编写基础 Scrapy 爬虫并运行

能使用 run.py 启动爬虫并关闭日志

3. 情感目标

建立工程化、框架化爬虫思维

养成规范项目结构开发习惯

教学重点：

Scrapy 安装与项目创建

爬虫类：name、start_requests、parse

yield 生成器与异步请求

run.py 启动爬虫

教学难点

yield 工作机制

异步回调 parse 函数

Scrapy 项目目录结构理解

实验仪器及教具：

计算机、因特网

教学过程	备注
一、课程导入（5 分钟） 1. 回顾：多线程、requests、BeautifulSoup 单机爬虫 2. 新需求：大批量、高效率、分布式爬取当当图书 3. 引入：**Scrapy 工业级爬虫框架** 4. 本节目标：搭建环境 + 编写第一个 Scrapy 爬虫	场景引入
二、任务 1：图书项目需求与模拟网站（10 分钟） 1. 项目目标：爬取 480 本图书信息 + 封面图片 2. 数据来源：books.csv + images 3. 分页结构：120 页自动翻页 4. 搭建简易 Flask 测试服务器	需求分析
三、任务 2：Scrapy 安装与项目创建（15 分钟） 1. 安装：pip install scrapy 2. 创建项目：scrapy startproject demo 3. 目录结构讲解：items、pipelines、settings、spiders 4. PyCharm 打开项目	环境搭建
四、任务 3：编写第一个 Scrapy 爬虫（20 分钟） 1. 爬虫类继承 scrapy.Spider 2. name = "mySpider" 3. start_requests 发送请求 4. parse 解析响应数据 5. run.py 启动爬虫	核心代码
五、任务 4：yield 与异步机制讲解（15 分钟） 1. yield 生成器返回可迭代对象 2. 异步：不等待，回调执行 3. start_urls 简写方式 4. 对比：同步 vs 异步	原理理解
六、运行调试与总结（15 分钟） 1. 运行 run.py 爬取测试页 2. 查看输出：url 与 body 3. 关闭日志：LOG_ENABLED=False 4. 常见错误与解决	实战验证
七、课堂总结（5 分钟） 1. 框架优势：高效、异步、工程化 2. 固定流程：创建→编写→运行 3. 下节：Items 与 Pipeline 数据存储	体系梳理

（以下内容为选填，供线上课程及线上线下混合式课程填报）

教学资源列表与访问路径	备　注
中国大学 MOOC 网站： https://www.icourse163.org/course/ZIIT-1002925008 https://www.icourse163.org/course/ZIIT-1002924007	
在线教学过程（学生自主学习指引）	
学生在讨论区提问 教师在讨论区回答 学生提交作业，讲师讲评	
在线辅导与答疑安排	
学生观看网络视频 教师在线回答问题	
线上作业： 复习本次课程内容 预习下次课程内容	

教学单元	项目 5-2　scrapy 通过 BeautifulSoup 爬取数据				
周次		时数	2	授课班级	

教学目标

1. 知识目标

掌握 Scrapy 与 BeautifulSoup 混合解析方法

理解 response.body 获取 HTML 原理

掌握绝对 URL 拼接 response.urljoin ()

掌握 Scrapy 异步下载图片机制

2. 技能目标

能搭建 Flask 图书模拟网站

能在 Scrapy 中使用 BS4 解析书名、作者、价格等信息

能异步爬取并保存图书封面图片

3. 情感目标

建立框架 + 解析器协同爬取思维

养成结构化解析、异步下载的工程习惯

教学重点：

Scrapy + BeautifulSoup 联合解析

response.urljoin 拼接绝对地址

异步下载图片 yield Request

download 回调保存文件

教学难点

Scrapy 异步请求与回调流程

相对路径转绝对路径

多回调协作爬取

实验仪器及教具：

计算机、因特网

教学过程	**备注**
一、课程导入（5 分钟）	衔接旧知

1. 回顾：Scrapy 项目结构、start_requests、parse 2. 问题：Scrapy 自带选择器不熟悉，如何沿用 BS4 技能？ 3. 方案：**Scrapy 负责请求，BeautifulSoup 负责解析** 4. 目标：爬取图书详情 + 封面图片	
二、任务 1：搭建图书模拟网站（10 分钟） 1. 准备 images 文件夹与 000002.jpg 2. 编写 book.html 模板（标题、作者、出版社、价格、简介、图片） 3. Flask 服务器配置静态资源路径 4. 运行访问 http://127.0.0.1:5000	环境准备
三、任务 2：Scrapy 中使用 BS4 解析数据（20 分钟） 1. response.body.decode () 获取 HTML 2. soup = BeautifulSoup(html, "lxml") 3. 依次提取：书名、作者、出版社、日期、价格、简介 4. 打印输出验证	核心解析
四、任务 3：图片地址处理与异步下载（20 分钟） 1. 提取 img 的 src 相对路径 2. response.urljoin (src) 转为绝对地址 3. yield scrapy.Request 异步下载 4. download 回调函数保存图片到 download 文件夹	图片爬取
五、任务 4：完整爬虫编写与运行（15 分钟） 1. 编写 MySpider.py 完整代码 2. 编写 run.py 启动器 3. 运行爬虫，查看文本输出与图片保存 4. 常见错误排查	综合调试
六、课堂总结与答疑（10 分钟） 1. Scrapy 只管请求调度，BS4 只管解析提取 2. 图片必须用绝对 URL + 异步下载 3. 下节预告：Items + Pipeline 数据存储	体系梳理

（以下内容为选填，供线上课程及线上线下混合式课程填报）

教学资源列表与访问路径	备　注
中国大学 MOOC 网站： https://www.icourse163.org/course/ZIIT-1002925008 https://www.icourse163.org/course/ZIIT-1002924007	
在线教学过程（学生自主学习指引）	

学生在讨论区提问 教师在讨论区回答 学生提交作业，讲师讲评	
在线辅导与答疑安排	
学生观看网络视频 教师在线回答问题	
线上作业： 复习本次课程内容 预习下次课程内容	

教学单元	项目 5-3　scrapy 爬取关联网页数据				
周次		时数	2	授课班级	

教学目标

1. 知识目标

理解关联网站树状结构

掌握深度优先、广度优先、Scrapy 异步三种爬取策略

理解 Scrapy 异步非阻塞、自动去重、分布式特性

2. 技能目标

能搭建带随机延迟的关联分页图书模拟网站

能使用 XPath 提取链接并递归爬取

能对比三种爬取顺序并解释差异原因

3. 情感目标

建立分布式爬虫思想

理解工业级爬虫效率优势

教学重点：

关联网站链接爬取

深度优先与广度优先原理

Scrapy 异步爬取与自动去重

response.xpath + yield Request 递归爬取

教学难点

Scrapy 爬取顺序随机性原因

异步非阻塞与分布式特性理解

多链接递归爬取去重机制

实验仪器及教具：

计算机、因特网

教学过程	**备注**
一、课程导入（5 分钟）	引入场景

1. 回顾：Scrapy XPath、extract/extract_first 2. 新场景：**多页面互相关联、带链接跳转** 3. 问题：如何爬取整站？顺序如何？ 4. 目标：对比三种爬取策略，掌握 Scrapy 全站爬取	
二、任务 1：搭建关联图书模拟网站（12 分钟） 1. 创建 7 个互链接页面：books、database、program 等 2. Flask 服务器添加 0~2 秒随机延迟 3. 网站结构呈现**树状层级** 4. 运行验证页面跳转与延迟	环境准备
三、任务 2：深度优先爬取（12 分钟） 1. 原理：一条路走到尽头再回溯 2. 代码：递归 + URL 去重列表 3. 输出顺序：计算机→数据库→MySQL→程序设计→…	讲解 + 演示
四、任务 3：广度优先爬取（12 分钟） 1. 原理：按层遍历，先同级再下级 2. 代码：队列 + URL 去重 3. 输出顺序：计算机→数据库→程序设计→网络→…	讲解 + 演示
五、任务 4：Scrapy 异步爬取（20 分钟） 1. 爬虫：parse 中提取链接→yield 新 Request 2. 特点：**自动去重、异步、非阻塞、顺序随机** 3. 运行观察：每次爬取顺序可能不同 4. 优势：不被慢页面阻塞，效率极高	核心难点
六、三种爬取策略对比（13 分钟） 1. 深度优先：递归、自上而下 2. 广度优先：队列、逐层爬取 3. Scrapy：异步、谁快先取谁、分布式 4. 总结：Scrapy 最适合工业级爬取	总结对比
七、课堂答疑（6 分钟）	查漏补缺

（以下内容为选填，供线上课程及线上线下混合式课程填报）

教学资源列表与访问路径	备　注
中国大学 MOOC 网站： https://www.icourse163.org/course/ZIIT-1002925008 https://www.icourse163.org/course/ZIIT-1002924007	
在线教学过程（学生自主学习指引）	

学生在讨论区提问 教师在讨论区回答 学生提交作业，讲师讲评	
在线辅导与答疑安排	
学生观看网络视频 教师在线回答问题	
线上作业： 复习本次课程内容 预习下次课程内容	

教学单元	项目 5-4　scrapy 通过 xpath 查找元素				
周次		时数	2	授课班级	

教学目标

1. 知识目标

掌握 Scrapy 与 BeautifulSoup 混合解析方法

理解 response.body 获取 HTML 原理

掌握绝对 URL 拼接 response.urljoin ()

掌握 Scrapy 异步下载图片机制

2. 技能目标

能搭建 Flask 图书模拟网站

能在 Scrapy 中使用 BS4 解析书名、作者、价格等信息

能异步爬取并保存图书封面图片

3. 情感目标

建立框架 + 解析器协同爬取思维

养成结构化解析、异步下载的工程习惯

教学重点：

Scrapy + BeautifulSoup 联合解析

response.urljoin 拼接绝对地址

异步下载图片 yield Request

download 回调保存文件

教学难点

Scrapy 异步请求与回调流程

相对路径转绝对路径

多回调协作爬取

实验仪器及教具：

计算机、因特网

教学过程	备注
一、课程导入（5 分钟）	衔接旧知

1. 回顾：Scrapy 项目结构、start_requests、parse 2. 问题：Scrapy 自带选择器不熟悉，如何沿用 BS4 技能？ 3. 方案：**Scrapy 负责请求，BeautifulSoup 负责解析** 4. 目标：爬取图书详情 + 封面图片	
二、任务 1：搭建图书模拟网站（10 分钟） 1. 准备 images 文件夹与 000002.jpg 2. 编写 book.html 模板（标题、作者、出版社、价格、简介、图片） 3. Flask 服务器配置静态资源路径 4. 运行访问 http://127.0.0.1:5000	环境准备
三、任务 2：Scrapy 中使用 BS4 解析数据（20 分钟） 1. response.body.decode () 获取 HTML 2. soup = BeautifulSoup(html, "lxml") 3. 依次提取：书名、作者、出版社、日期、价格、简介 4. 打印输出验证	核心解析
四、任务 3：图片地址处理与异步下载（20 分钟） 1. 提取 img 的 src 相对路径 2. response.urljoin (src) 转为绝对地址 3. yield scrapy.Request 异步下载 4. download 回调函数保存图片到 download 文件夹	图片爬取
五、任务 4：完整爬虫编写与运行（15 分钟） 1. 编写 MySpider.py 完整代码 2. 编写 run.py 启动器 3. 运行爬虫，查看文本输出与图片保存 4. 常见错误排查	综合调试
六、课堂总结与答疑（10 分钟） 1. Scrapy 只管请求调度，BS4 只管解析提取 2. 图片必须用绝对 URL + 异步下载 3. 下节预告：Items + Pipeline 数据存储	体系梳理

（以下内容为选填，供线上课程及线上线下混合式课程填报）

教学资源列表与访问路径	备　注
中国大学 MOOC 网站： https://www.icourse163.org/course/ZIIT-1002925008 https://www.icourse163.org/course/ZIIT-1002924007	
在线教学过程（学生自主学习指引）	

学生在讨论区提问 教师在讨论区回答 学生提交作业，讲师讲评	
在线辅导与答疑安排	
学生观看网络视频 教师在线回答问题	
线上作业： 复习本次课程内容 预习下次课程内容	

教学单元	项目 5-5 scrapy 通过 xpath 爬取数据				
周次		时数	2	授课班级	

教学目标

1. 知识目标

掌握 Scrapy 中 XPath 完整提取流程

掌握相对 XPath .// 写法与循环提取

掌握 meta 参数传值、图片异步下载、SQLite3 存储

理解 response.urljoin 拼接绝对路径原理

2. 技能目标

能搭建 Flask 图书模拟网站

能使用 XPath 批量提取书名、作者、出版社、价格、简介

能异步下载图片并保存到本地

能将数据存入 SQLite 数据库

3. 情感目标

建立工业级爬虫工程化思维

养成结构化解析、异步下载、数据持久化规范

教学重点：

XPath 循环提取列表数据

相对路径 .// 定位子元素

meta 传参传递图片名

数据库建表、插入、查询

教学难点

循环内使用相对 XPath

meta 字典传值机制

异步下载与数据入库顺序控制

实验仪器及教具：

计算机、因特网

教学过程	备注
一、课程导入（5 分钟） 1. 回顾：Scrapy 爬虫结构、Selector、XPath 基础 2. 任务：爬取图书信息 + 封面图片 → 存入数据库 3. 流程：请求 → XPath 解析 → 图片下载 → 数据存储	项目式教学
二、任务 1：搭建图书模拟网站（10 分钟） 1. 准备 book.html 模板与 images 图片 2. 编写 Flask 服务器，配置静态资源路径 3. 运行访问 http://127.0.0.1:5000 验证页面	环境准备
三、任务 2：XPath 循环提取图书数据（20 分钟） 1. 获取所有 li：//ul/li 2. 循环内使用 **相对 XPath** .// 提取信息 3. 提取书名、作者、出版社、日期、价格、简介 4. extract_first ().strip () 清洗数据	核心解析
四、任务 3：图片异步下载与 meta 传参（15 分钟） 1. 提取 img/@src → urljoin 转绝对路径 2. 使用 Request.meta 传递图片名称 3. 回调 download 函数保存图片到 download 文件夹	图片爬取
五、任务 4：数据库设计与数据存储（15 分钟） 1. 创建 books 表：ID、书名、作者、出版社、日期、价格、简介、图片名 2. 封装 openDB、insertDB、closeDB、showDB 3. 逐条插入数据并控制台输出	数据持久化
六、完整爬虫运行与调试（10 分钟） 1. 启动 Flask 服务器 2. 运行 Scrapy 爬虫 3. 查看数据库与下载图片，排查错误	综合验证
七、课堂总结与答疑（5 分钟） 1. Scrapy 爬取流程：请求 → 解析 → 下载 → 入库 2. 关键技巧：相对 XPath、meta 传参、urljoin、异步下载	体系梳理

（以下内容为选填，供线上课程及线上线下混合式课程填报）

教学资源列表与访问路径	备　注
中国大学 MOOC 网站： https://www.icourse163.org/course/ZIIT-1002925008 https://www.icourse163.org/course/ZIIT-1002924007	

在线教学过程（学生自主学习指引）	
学生在讨论区提问 教师在讨论区回答 学生提交作业，讲师讲评	
在线辅导与答疑安排	
学生观看网络视频 教师在线回答问题	
线上作业： 复习本次课程内容 预习下次课程内容	

教学单元	项目 5-6　scrapy 通过管道存储数据				
周次		时数	2	授课班级	

教学目标

1. 知识目标

理解 Scrapy 管道异步存储原理

掌握 Item 数据结构定义

掌握 Pipeline 三大核心方法

掌握 settings 启用管道配置

2. 技能目标

能编写 Item 字段类

能在爬虫中 yield 提交 Item

能编写 Pipeline 完成数据库存储

能配置并运行完整管道项目

3. 情感目标

建立爬虫与存储分离的工程思想

理解异步架构优势

教学重点：

Item 类定义

parse 中 yield item

Pipeline：open_spider、process_item、close_spider

settings 启用 ITEM_PIPELINES

教学难点

异步流程理解

Item 数据传递机制

Pipeline 与数据库生命周期管理

实验仪器及教具：

计算机、因特网

教学过程	**备注**
一、课程导入（5 分钟）	痛点引入

1. 回顾：Scrapy 爬取图书信息 + 图片下载 2. 问题：直接在 parse 里存数据库会阻塞异步 3. 引出：**Scrapy 管道 Pipeline**—— 爬取与存储分离	
二、任务 1：搭建图书模拟网站（10 分钟） 1. 准备 book.html 模板 2. 编写 server.py 读取 books.csv 渲染 10 本图书 3. 运行访问 http://127.0.0.1:5000	环境准备
三、任务 2：编写 Item 数据结构（10 分钟） 1. 修改 items.py 2. 定义 BookItem，包含：书名、作者、出版社、日期、价格、简介、图片名 3. 赋值与取值方法：item ["bTitle"]	数据模型
四、任务 3：编写爬虫并提交 Item（15 分钟） 1. 导入 BookItem 2. XPath 提取数据并封装进 item 3. 使用 yield item 推送给管道 4. 图片下载保持不变	核心逻辑
五、任务 4：编写 Pipeline 管道类（15 分钟） 1. open_spider：打开数据库、建表 2. process_item：接收 item、入库、计数 3. close_spider：提交、关闭数据库、打印总数	存储核心
六、任务 5：配置 settings 启用管道（10 分钟） 1. 设置 ROBOTSTXT_OBEY = False 2. 配置 ITEM_PIPELINES 注册管道 3. 数字 300 表示执行优先级	配置关键
七、运行调试与结果分析（10 分钟） 1. 启动 Flask 2. 运行爬虫 3. 观察输出：open→10 次 process→close 4. 查看数据库与下载图片	完整验证
八、课堂总结（5 分钟） 1. 流程：爬取→Item→Pipeline→数据库 2. 管道三剑客：开启、处理、关闭 3. 优势：异步、解耦、可扩展	体系梳理

（以下内容为选填，供线上课程及线上线下混合式课程填报）

教学资源列表与访问路径	备　注

中国大学 MOOC 网站： https://www.icourse163.org/course/ZIIT-1002925008 https://www.icourse163.org/course/ZIIT-1002924007	
在线教学过程（学生自主学习指引）	
学生在讨论区提问 教师在讨论区回答 学生提交作业，讲师讲评	
在线辅导与答疑安排	
学生观看网络视频 教师在线回答问题	
线上作业： 复习本次课程内容 预习下次课程内容	

教学单元	项目 5-7 爬取图书网站数据				
周次		时数	2	授课班级	

教学目标

1. 知识目标

掌握分页图书网站结构与翻页原理

掌握 Scrapy 全站异步爬取与自动翻页

掌握 Item + Pipeline 完整数据持久化流程

掌握真实网站（当当网）HTML 结构解析

2. 技能目标

能搭建分页模拟图书网站

能编写 Scrapy 爬虫实现自动翻页爬取

能批量下载图片并存储到 SQLite 数据库

能独立爬取当当网 Python 图书数据

3. 情感目标

建立工业级爬虫工程思维

养成模块化、异步化、健壮性编程习惯

教学重点：

分页爬取与自动翻页

XPath 精准提取图书数据

Scrapy 管道存储数据

真实网站结构适配

教学难点

分页链接定位与递归请求

空值判断与异常处理

真实网站编码与反爬应对

实验仪器及教具：

计算机、因特网

教学过程	**备注**
一、课程导入（5 分钟）	项目总览

1. 回顾：Scrapy、Item、Pipeline、XPath、异步下载 2. 本节任务：**模拟站全站爬取 + 当当网实战爬取** 3. 目标：爬取全部图书→入库→下载封面	
二、任务 1：搭建分页模拟图书网站（10 分钟） 1. 编写分页 book.html 模板 2. Flask 读取 books.csv 实现分页（每页 4 本） 3. 运行验证翻页功能	环境准备
三、任务 2：项目架构编写（10 分钟） 1. items.py 定义 BookItem（ID、书名、作者、出版社等） 2. pipelines.py 实现数据库建表、插入、关闭 3. settings.py 启用管道	架构搭建
四、任务 3：爬虫核心 —— 翻页 + 爬取（25 分钟） 1. XPath 提取每本书信息 2. 自动翻页：获取 “下一页” 链接并递归请求 3. 封面图片异步下载 4. 封装 Item 并提交管道	核心逻辑
五、任务 4：模拟站全站爬取运行（10 分钟） 1. 启动 Flask 2. 运行爬虫，爬取约 120 页、480 本图书 3. 查看数据库与下载图片	综合验证
六、任务 5：实战 —— 爬取当当图书网（15 分钟） 1. 分析当当网列表页结构 2. 适配 XPath、处理编码（gbk）、处理空值 3. 实现自动翻页，爬取数千条数据	真实实战
七、课堂总结与答疑（5 分钟） 1. 万能爬虫流程：请求→解析→翻页→下载→入库 2. 模拟站→真实站只需修改 URL 与 XPath 3. Scrapy 优势：异步、高效、可扩展	体系梳理

（以下内容为选填，供线上课程及线上线下混合式课程填报）

教学资源列表与访问路径	备　注
中国大学 MOOC 网站： https://www.icourse163.org/course/ZIIT-1002925008 https://www.icourse163.org/course/ZIIT-1002924007	
在线教学过程（学生自主学习指引）	

学生在讨论区提问 教师在讨论区回答 学生提交作业，讲师讲评	
在线辅导与答疑安排	
学生观看网络视频 教师在线回答问题	
线上作业： 复习本次课程内容 预习下次课程内容	

教学单元	项目 6-1　selenium 编写爬虫程序				
周次		时数	2	授课班级	

教学目标

1. 知识目标

理解静态网页与动态网页的区别

掌握 Selenium 作用与原理

掌握 ChromeDriver 环境配置

理解普通爬虫无法爬取 JS 数据的原因

2. 技能目标

能搭建 JS 动态模拟网站

能安装配置 Selenium + ChromeDriver

能编写无头浏览器爬虫

能爬取 AJAX/JS 动态渲染内容

3. 情感目标

建立动态网站爬虫思维

掌握工业级真实网站爬取方案

教学重点：

Selenium 安装与 ChromeDriver 配置

无头模式（headless）启动

driver.get/page_source 获取渲染后 HTML

爬取 JS + Ajax 动态数据

教学难点

Chrome 与 ChromeDriver 版本匹配

动态数据渲染等待原理

静态爬取与动态爬取的本质区别

实验仪器及教具：

计算机、因特网

教学过程	备注
一、课程导入（5 分钟） 1. 回顾：requests/urllib/BeautifulSoup 只能爬静态网页 2. 问题：京东、淘宝数据由 JS 渲染，普通爬虫爬不到 3. 解决方案：**Selenium 模拟浏览器执行 JS**	痛点引入
二、任务 1：搭建 JS 动态网页测试网站（12 分钟） 1. 编写 phone.html（包含静态值、JS 赋值、Ajax 请求） 2. Flask 编写 server.py 3. 运行验证：页面显示 3 组消息	环境准备
三、任务 2：普通爬虫爬取失败对比（10 分钟） 1. 使用 urllib 爬取网页 2. 只能获取静态 HTML，无法获取 jMsg、sMsg 3. 得出结论：普通爬虫不执行 JS	对比演示
四、任务 3：Selenium 环境安装（13 分钟） 1. 安装：pip install selenium 2. 查看 Chrome 版本 3. 下载对应 ChromeDriver 并配置环境 4. 测试驱动是否可用	环境配置
五、任务 4：Selenium 无头浏览器爬虫（20 分钟） 1. 设置 headless 无界面模式 2. driver.get (url) 访问网页 3. driver.page_source 获取渲染后页面 4. BeautifulSoup 解析全部 3 个数据	核心代码
六、任务 5：运行与结果分析（15 分钟） 1. 运行 Selenium 爬虫 2. 成功爬取：Html Message / JS Message / Server Message 3. 原理讲解：Selenium = 隐形浏览器	验证结论
七、课堂总结与答疑（5 分钟） 1. 静态爬虫→拿源码；动态爬虫→拿渲染结果 2. Selenium 流程：启动→访问→获取→关闭 3. 下节：爬取 JS 分页商城网站	体系梳理

（以下内容为选填，供线上课程及线上线下混合式课程填报）

教学资源列表与访问路径	备　注

中国大学 MOOC 网站： https://www.icourse163.org/course/ZIIT-1002925008 https://www.icourse163.org/course/ZIIT-1002924007	
在线教学过程（学生自主学习指引）	
学生在讨论区提问 教师在讨论区回答 学生提交作业，讲师讲评	
在线辅导与答疑安排	
学生观看网络视频 教师在线回答问题	
线上作业： 复习本次课程内容 预习下次课程内容	

教学单元	项目 6-2　selenium 查找 HTML 元素				
周次		时数	2	授课班级	

教学目标

1. 知识目标

掌握 Selenium 八大元素定位方式

掌握 text 获取文本、get_attribute 获取属性

理解单个 / 多个元素查找区别

理解复合 class 定位规则

2. 技能目标

能搭建手机商城模拟网站

能使用 XPath/CSS/ID/NAME 等定位数据

能获取文本、属性、HTML 内容

3. 情感目标

建立精准定位思维

提升动态网页爬取能力

教学重点：

XPath 定位、CSS 定位、ID/NAME 定位

text 获取文本、get_attribute 获取属性

find_element /find_elements 区别

教学难点

相对 XPath 写法

复合 class 定位处理

空属性与不存在元素判断

实验仪器及教具：

计算机、因特网

教学过程	备注
一、课程导入（5 分钟） 1. 回顾：Selenium 可渲染 JS 页面 2. 新问题：拿到页面后**如何精准取数据**？ 3. 本节目标：学习 Selenium 八大定位法，爬取手机信息	承上启下
二、任务 1：搭建手机商城模拟网站（10 分钟） 1. 编写 phone.html 页面（图片、名称、品牌、价格、简介） 2. Flask 配置静态图片路径 3. 运行访问 http://127.0.0.1:5000	环境准备
三、任务 2：Selenium 基础定位结构（5 分钟） 1. 无头模式启动浏览器 2. driver.get 打开网页 3. By 类导入： ID/NAME/XPATH/CSS/TAG_NAME/LINK_TEXT/PARTIAL_LINK_TEXT/CLASS_NAME	框架搭建
四、任务 3：核心定位方法精讲（35 分钟） 1. XPath 定位（最通用） find_element(By.XPATH, "//h3") 2. ID/NAME 定位（最简单） 3. CSS 定位（最简洁） 4. CLASS_NAME / TAG_NAME 定位 5. LINK_TEXT 链接文本定位 6. 异常捕获：找不到元素抛错处理	核心知识点
五、任务 4：获取数据（文本 / 属性 / HTML）（15 分钟） 1. .text 获取元素文本 2. .get_attribute ("src") 获取属性 3. .get_attribute ("innerHTML/outerHTML") 获取源码	数据提取
六、任务 5：复合 class 定位（5 分钟） 1. class="pl date" 不能直接 CLASS_NAME 查找 2. XPath: contains(@class,'pl') 3. CSS: span.pl.date	难点突破
七、综合运行与总结（5 分钟） 1. 一次性爬取：名称、品牌、价格、图片地址 2. 总结：优先 XPath / CSS 3. 关闭 driver	巩固提升

（以下内容为选填，供线上课程及线上线下混合式课程填报）

教学资源列表与访问路径	备　注

中国大学 MOOC 网站： https://www.icourse163.org/course/ZIIT-1002925008 https://www.icourse163.org/course/ZIIT-1002924007	
在线教学过程（学生自主学习指引）	
学生在讨论区提问 教师在讨论区回答 学生提交作业，讲师讲评	
在线辅导与答疑安排	
学生观看网络视频 教师在线回答问题	
线上作业： 复习本次课程内容 预习下次课程内容	

教学单元	项目 6-3　selenium 实现用户登录				
周次		时数	2	授课班级	

1. 知识目标
理解需要登录的网站爬取原理
掌握 Selenium 模拟输入、点击、执行 JS
掌握表单提交与页面跳转逻辑
2. 技能目标
能搭建登录验证网站
能使用 send_keys /click 模拟登录
能用 execute_script 执行 JS 完成登录
能爬取登录后页面数据
3. 情感目标
掌握真实网站爬虫必备技能
提升爬虫工程化能力

教学重点：
find_element 定位表单元素
send_keys 输入账号密码
click 点击登录按钮
execute_script 执行 JS

教学难点
页面加载等待处理
JS 传参 arguments 使用
登录状态保持与跳转

实验仪器及教具：

计算机、因特网

教学过程	备注
一、课程导入（5 分钟） 1. 回顾：Selenium 定位元素、获取文本 / 属性 2. 新场景：**需要登录才能爬取数据** 3. 目标：模拟用户登录→进入后台→爬取手机数据	场景引入
二、任务 1：搭建登录验证网站（12 分钟） 1. 编写 login.html 登录表单（用户、密码、提交） 2. Flask 编写 server.py 验证登录 3. 登录成功跳转 /show 显示手机表格 4. 运行验证登录流程	环境准备
三、任务 2：Selenium 模拟登录（20 分钟） 1. 定位账号输入框：NAME=user 2. send_keys ("xxx") 输入用户名 3. 定位密码框：NAME=pwd 4. send_keys ("123") 输入密码 5. 定位登录按钮并 click () 提交	核心操作
四、任务 3：登录后爬取数据（15 分钟） 1. 等待页面跳转 2. 查找所有 tr 标签 3. 循环读取 td 文本 4. 格式化输出品牌、型号、价格	数据爬取
五、任务 4：JS 方式实现登录（18 分钟） 1. driver.execute_script (js 代码) 2. 直接给 input 赋值：value='xxx' 3. 调用 click () 提交 4. arguments 传参用法	进阶方法
六、综合运行与总结（7 分钟） 1. 完整运行：启动浏览器→登录→爬取→关闭 2. 对比两种登录方式优缺点 3. 总结：模拟操作三步骤：定位→输入→点击	巩固梳理
七、答疑（3 分钟）	解决问题

（以下内容为选填，供线上课程及线上线下混合式课程填报）

教学资源列表与访问路径	备　注

中国大学 MOOC 网站： https://www.icourse163.org/course/ZIIT-1002925008 https://www.icourse163.org/course/ZIIT-1002924007	
在线教学过程（学生自主学习指引）	
学生在讨论区提问 教师在讨论区回答 学生提交作业，讲师讲评	
在线辅导与答疑安排	
学生观看网络视频 教师在线回答问题	
线上作业： 复习本次课程内容 预习下次课程内容	

教学单元	项目 6-4　selenium 爬取 Ajax 网页数据				
周次		时数	2	授课班级	

教学目标

1. 知识目标

理解 Ajax 动态加载原理

理解静态爬虫无法爬取 Ajax 数据的原因

掌握 Selenium 渲染 JS + Ajax 页面机制

2. 技能目标

能搭建 Ajax 动态手机网站

能使用 Selenium 爬取异步加载数据

能模拟点击下拉框切换品牌并批量爬取

3. 情感目标

掌握真实网站主流数据加载爬取方案

教学重点：

Ajax 异步数据加载原理

Selenium 获取渲染后页面

模拟点击 option 切换数据

批量爬取表格数据

教学难点

下拉框选择与 onchange 事件触发

动态页面等待与数据稳定获取

实验仪器及教具：

计算机、因特网

教学过程	**备注**
一、课程导入（5 分钟）	场景引入

1. 回顾：Selenium 登录、元素定位、模拟点击 2. 新场景：**Ajax 异步加载数据**（下拉选择→异步请求→刷新局部） 3. 目标：爬取 3 个品牌所有手机数据	
二、任务 1：搭建 Ajax 动态手机网站（12 分钟） 1. 编写 phone.html（JS+select+Ajax 加载） 2. Flask 编写 server.py 返回 JSON 数据 3. 运行验证：切换下拉框刷新数据	环境准备
三、任务 2：静态爬虫失败对比（10 分钟） 1. urllib 直接获取页面源码 2. 结果：无任何手机数据，只有 JS 代码 3. 结论：静态爬虫无法执行 Ajax 请求	对比演示
四、任务 3：Selenium 爬取 Ajax 页面（18 分钟） 1. 无头模式启动浏览器 2. driver.get 访问页面，自动执行 JS 3. driver.page_source 获取完整渲染页面	核心原理
五、任务 4：模拟选择下拉框并爬取（22 分钟） 1. 定位 select 元素：id=marks 2. 获取所有 option 并循环点击 3. 每次点击触发 Ajax，爬取当前品牌手机 4. 解析 table/tr/td 获取型号、价格	核心逻辑
六、任务 5：运行与结果验证（10 分钟） 1. 运行爬虫，输出华为、苹果、三星全部机型 2. 共 5 条数据，验证爬取完整 3. 关闭浏览器	结果验证
七、课堂总结（3 分钟） 1. Ajax 数据必须用 Selenium 爬取 2. 流程：打开→渲染→定位→模拟操作→提取 3. 下节：爬取京东真实动态手机数据	体系梳理

（以下内容为选填，供线上课程及线上线下混合式课程填报）

教学资源列表与访问路径	备　注
中国大学 MOOC 网站： https://www.icourse163.org/course/ZIIT-1002925008 https://www.icourse163.org/course/ZIIT-1002924007	
在线教学过程（学生自主学习指引）	

学生在讨论区提问 教师在讨论区回答 学生提交作业，讲师讲评	
在线辅导与答疑安排	
学生观看网络视频 教师在线回答问题	
线上作业： 复习本次课程内容 预习下次课程内容	

教学单元	项目 6-5　selenium 等待 HTML 元素				
周次		时数	2	授课班级	

教学目标

1. 知识目标

理解动态网页元素延迟加载原理

掌握强制等待、隐式等待、显式等待的区别

掌握常用 expected_conditions 等待条件

2. 技能目标

能搭建延迟加载模拟网站

能使用三种等待方式解决元素找不到问题

能使用显式等待精准定位元素

3. 情感目标

养成稳定、健壮的爬虫编写习惯

教学重点：

强制等待 time.sleep ()

隐式等待 implicitly_wait ()

显式等待 WebDriverWait + EC

元素出现、可见、可点击判断

教学难点

显式等待原理与使用场景

等待条件选择（出现 / 可见 / 可点击

实验仪器及教具：

计算机、因特网

教学过程	备注
一、课程导入（5 分钟） 1. 回顾：Ajax 动态数据、模拟点击、表单操作 2. 问题：元素未加载完就操作会**报错** 3. 解决方案：Selenium 三大等待机制	痛点引入
二、任务 1：搭建延迟加载网站（10 分钟） 1. 编写 phone.html（Ajax 异步加载下拉框） 2. Flask 服务器加 time.sleep (1) 模拟延迟 3. 运行验证：选项延迟 1 秒后出现	环境准备
三、任务 2：无等待爬虫失败演示（10 分钟） 1. 直接访问页面，获取 option 数量为 0 2. 按钮处于禁用状态 3. 结论：必须等待元素加载完成	失败对比
四、任务 3：三种等待方式精讲（35 分钟） 1. **强制等待**：time.sleep(1.5) 2. **隐式等待**：driver.implicitly_wait(1.5) 3. **显式等待**：WebDriverWait + EC 等待元素出现、可见、可点击	核心知识点
五、任务 4：显式等待综合实战（15 分钟） 1. 等待 option 加载完成 2. 等待提交按钮可点击 3. 爬取所有品牌并输出页面结构	稳定爬取
六、总结与对比（3 分钟） 1. 强制等待：简单但浪费时间 2. 隐式等待：全局设置，不够灵活 3. 显式等待：精准高效，生产推荐	体系梳理
七、答疑（2 分钟）	解决问题

（以下内容为选填，供线上课程及线上线下混合式课程填报）

教学资源列表与访问路径	备　注
中国大学 MOOC 网站： https://www.icourse163.org/course/ZIIT-1002925008 https://www.icourse163.org/course/ZIIT-1002924007	
在线教学过程（学生自主学习指引）	

学生在讨论区提问 教师在讨论区回答 学生提交作业，讲师讲评	
在线辅导与答疑安排	
学生观看网络视频 教师在线回答问题	
线上作业： 复习本次课程内容 预习下次课程内容	

教学单元	项目 6-6　爬取商城网站数据				
周次		时数	2	授课班级	

教学目标

1. 知识目标

掌握 JS 分页动态网站爬取逻辑

掌握多线程图片异步下载

掌握数据库存储商品数据完整流程

理解真实电商网站结构与反爬特点

2. 技能目标

能搭建分页手机商城模拟网站

能使用 Selenium 自动翻页爬取

能批量下载图片并入库

能独立爬取京东手机商品数据

3. 情感目标

建立工业级动态网站爬虫开发思维

养成模块化、健壮性、可扩展编程习惯

教学重点：

Selenium 自动翻页、XPath 解析商品

多线程异步下载图片

SQLite 数据库存储

京东实战结构适配

教学难点

动态按钮判断与翻页逻辑

图片懒加载处理（src/data-lazy-img）

异常处理与等待机制

实验仪器及教具：

计算机、因特网

教学过程	备注
一、课程导入（5 分钟） 1. 回顾：Selenium 定位、等待、模拟操作、Ajax 爬取 2. 本节任务：**模拟商城全站爬取 + 京东真实项目实战** 3. 目标：爬数据 + 存数据库 + 下图片	项目总览
二、任务 1：搭建分页模拟商城（10 分钟） 1. 编写 phone.html 分页模板 2. Flask 读取 phones.csv 实现分页 3. 运行验证：JS 按钮翻页	环境准备
三、任务 2：爬虫架构设计（10 分钟） 1. 面向对象 MySpider 类 2. 初始化：浏览器、数据库、图片文件夹 3. 核心：爬取、翻页、下载、入库	架构搭建
四、任务 3：核心爬取逻辑（20 分钟） 1. XPath 提取手机信息（名称、价格、简介、图片） 2. 自动翻页：判断 “下一页” 是否可用 3. 多线程异步下载图片，提升效率 4. 数据插入 SQLite	核心代码
五、任务 4：模拟商城全站爬取（10 分钟） 1. 运行爬虫，自动多页爬取 2. 查看数据库与 download 图片 3. 验证数据完整性	综合验证
六、任务 5：京东实战项目（15 分钟） 1. 结构分析：商品列表、图片懒加载、JS 翻页 2. 处理：输入关键词、回车搜索、循环翻页 3. 兼容 src /data-lazy-img 4. 批量爬取数千条商品数据	真实实战
七、课堂总结与答疑（5 分钟） 1. 动态网站万能流程：打开→等待→解析→翻页→存储 2. 模拟站→真实站：只改 URL 与 XPath 3. Selenium 是动态爬虫终极方案	体系梳理

（以下内容为选填，供线上课程及线上线下混合式课程填报）

教学资源列表与访问路径	备　注

中国大学 MOOC 网站： https://www.icourse163.org/course/ZIIT-1002925008 https://www.icourse163.org/course/ZIIT-1002924007	
在线教学过程（学生自主学习指引）	
学生在讨论区提问 教师在讨论区回答 学生提交作业，讲师讲评	
在线辅导与答疑安排	
学生观看网络视频 教师在线回答问题	
线上作业： 复习本次课程内容 预习下次课程内容	