1.学习 Python 包并实现基本的爬虫过程

2.了解非结构化数据的存储

3.学习scrapy，搭建工程化爬虫

4.学习数据库知识，应对大规模数据存储与提取

5.掌握各种技巧，应对特殊网站的反爬措施

6.分布式爬虫，实现大规模并发采集，提升效率

 - ❶ -

**学习 Python 包并实现基本的爬虫过程**

大部分爬虫都是按**“发送请求——获得页面——解析页面——抽取并储存内容”**这样的流程来进行，这其实也是模拟了我们使用浏览器获取网页信息的过程。

Python中爬虫相关的包很多：urllib、requests、bs4、scrapy、pyspider 等，**建议从requests+Xpath 开始**，requests 负责连接网站，返回网页，Xpath 用于解析网页，便于抽取数据。

如果你用过 BeautifulSoup，会发现 Xpath 要省事不少，一层一层检查元素代码的工作，全都省略了。这样下来基本套路都差不多，**一般的静态网站根本不在话下，豆瓣、糗事百科、腾讯新闻等基本上都可以上手了**。

当然如果你需要爬取异步加载的网站，可以学习浏览器**抓包分析真实请求或者学习Selenium来实现自动化，这样，知乎、时光网、猫途鹰这些动态的网站也可以迎刃而解**。

 - ❷ -

**了解非结构化数据的存储**

爬回来的数据可以直接用文档形式存在本地，也可以存入数据库中。

开始数据量不大的时候，你可以直接**通过 Python 的语法或 pandas 的方法将数据存为csv这样的文件**。

当然你可能发现爬回来的数据并不是干净的，可能会有缺失、错误等等，你还需要对数据进行清洗，可以**学习 pandas 包的基本用法来做数据的预处理**，得到更干净的数据。

- ❸ -

**学习 scrapy，搭建工程化的爬虫**

掌握前面的技术一般量级的数据和代码基本没有问题了，但是在遇到非常复杂的情况，可能仍然会力不从心，这个时候，强大的 scrapy 框架就非常有用了。

scrapy 是一个功能非常强大的爬虫框架，它不仅能便捷地构建request，还有强大的 selector 能够方便地解析 response，然而它最让人惊喜的还是它超高的性能，让你可以将爬虫工程化、模块化。

**学会 scrapy，你可以自己去搭建一些爬虫框架，你就基本具备爬虫工程师的思维了。**

- ❹ -

**学习数据库基础，应对大规模数据存储**

爬回来的数据量小的时候，你可以用文档的形式来存储，一旦数据量大了，这就有点行不通了。所以掌握一种数据库是必须的，学习目前比较主流的 MongoDB 就OK。

**MongoDB 可以方便你去存储一些非结构化的数据，比如各种评论的文本，图片的链接等等。**你也可以利用PyMongo，更方便地在Python中操作MongoDB。

因为这里要用到的数据库知识其实非常简单，主要是**数据如何入库、如何进行提取**，在需要的时候再学习就行。

- ❺ -

**掌握各种技巧，应对特殊网站的反爬措施**

当然，爬虫过程中也会经历一些绝望啊，比如被网站封IP、比如各种奇怪的验证码、userAgent访问限制、各种动态加载等等。

遇到这些反爬虫的手段，当然还需要一些高级的技巧来应对，常规的比如**访问频率控制、使用代理IP池、抓包、验证码的OCR处理等等**。

往往网站在高效开发和反爬虫之间会偏向前者，这也为爬虫提供了空间，**掌握这些应对反爬虫的技巧，绝大部分的网站已经难不到你了**。

 - ❻ -

**分布式爬虫，实现大规模并发采集**

爬取基本数据已经不是问题了，你的瓶颈会集中到爬取海量数据的效率。这个时候，相信你会很自然地接触到一个很厉害的名字：**分布式爬虫**。

分布式这个东西，听起来很恐怖，**但其实就是利用多线程的原理让多个爬虫同时工作，需要你掌握 Scrapy + MongoDB + Redis 这三种工具**。

Scrapy 前面我们说过了，用于做基本的页面爬取，MongoDB 用于存储爬取的数据，Redis 则用来存储要爬取的网页队列，也就是任务队列。

所以有些东西看起来很吓人，但其实分解开来，也不过如此。当你能够写分布式的爬虫的时候，那么你可以去尝试打造一些基本的爬虫架构了，实现一些更加自动化的数据获取。

你看，这一条学习路径下来，你已然可以成为老司机了，非常的顺畅。所以在一开始的时候，**尽量不要系统地去啃一些东西，找一个实际的项目（开始可以从豆瓣、小猪这种简单的入手），直接开始就好**。

因为爬虫这种技术，既不需要你系统地精通一门语言，也不需要多么高深的数据库技术，**高效的姿势就是从实际的项目中去学习这些零散的知识点，你能保证每次学到的都是最需要的那部分。**



**《Python爬虫：入门+进阶》大纲**

**第一章：Python 爬虫入门**

**1、什么是爬虫**

网址构成和翻页机制

网页源码结构及网页请求过程

爬虫的应用及基本原理

**2、初识Python爬虫**

Python爬虫环境搭建

创建第一个爬虫：爬取百度首页

爬虫三步骤：获取数据、解析数据、保存数据

**3、使用Requests爬取豆瓣短评**

Requests的安装和基本用法

用Requests 爬取豆瓣短评信息

一定要知道的爬虫协议

**4、使用Xpath解析豆瓣短评**

解析神器Xpath的安装及介绍

Xpath的使用：浏览器复制和手写

实战：用 Xpath 解析豆瓣短评信息

**5、使用pandas保存豆瓣短评数据**

pandas 的基本用法介绍

pandas文件保存、数据处理

实战：使用pandas保存豆瓣短评数据

**6、浏览器抓包及headers设置（案例一：爬取知乎）**

爬虫的一般思路：抓取、解析、存储

浏览器抓包获取Ajax加载的数据

设置headers 突破反爬虫限制

实战：爬取知乎用户数据

**7、数据入库之MongoDB（案例二：爬取拉勾）**

MongoDB及RoboMongo的安装和使用

设置等待时间和修改信息头

实战：爬取拉勾职位数据

将数据存储在MongoDB中

补充实战：爬取微博移动端数据

**8、Selenium爬取动态网页（案例三：爬取淘宝）**

动态网页爬取神器Selenium搭建与使用

分析淘宝商品页面动态信息

实战：用Selenium 爬取淘宝网页信息

**第二章：Python爬虫之Scrapy框架**

**1、爬虫工程化及Scrapy框架初窥**

html、css、js、数据库、http协议、前后台联动

爬虫进阶的工作流程

Scrapy组件：引擎、调度器、下载中间件、项目管道等

常用的爬虫工具：各种数据库、抓包工具等

**2、Scrapy安装及基本使用**

Scrapy安装

Scrapy的基本方法和属性

开始第一个Scrapy项目

**3、Scrapy选择器的用法**

常用选择器：css、xpath、re、pyquery

css的使用方法

xpath的使用方法

re的使用方法

pyquery的使用方法

**4、Scrapy的项目管道**

Item Pipeline的介绍和作用

Item Pipeline的主要函数

实战举例：将数据写入文件

实战举例：在管道里过滤数据

**5、Scrapy的中间件**

下载中间件和蜘蛛中间件

下载中间件的三大函数

系统默认提供的中间件

**6、Scrapy的Request和Response详解**

Request对象基础参数和高级参数

Request对象方法

Response对象参数和方法

Response对象方法的综合利用详解

**第三章：Python爬虫进阶操作**

**1、网络进阶之谷歌浏览器抓包分析**

http请求详细分析

网络面板结构

过滤请求的关键字方法

复制、保存和清除网络信息

查看资源发起者和依赖关系

**2、数据入库之去重与数据库**

数据去重

数据入库MongoDB

**第四章：分布式爬虫及实训项目**

**1、大规模并发采集——分布式爬虫的编写**

分布式爬虫介绍

Scrapy分布式爬取原理

Scrapy-Redis的使用

Scrapy分布式部署详解

**2、实训项目（一）——58同城二手房监控**

**3、实训项目（二）——去哪儿网模拟登陆**

**4、实训项目（三）——京东商品数据抓取**