法律声明

- 本课件包括：演示文稿，示例，代码，题库，视频和声音等，小象学院拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利。

- 课程详情请咨询
  - 微信公众号：大数据分析挖掘
  - 新浪微博：ChinaHadoop

第四讲

网络数据的获取与表示

--梁斌

目录

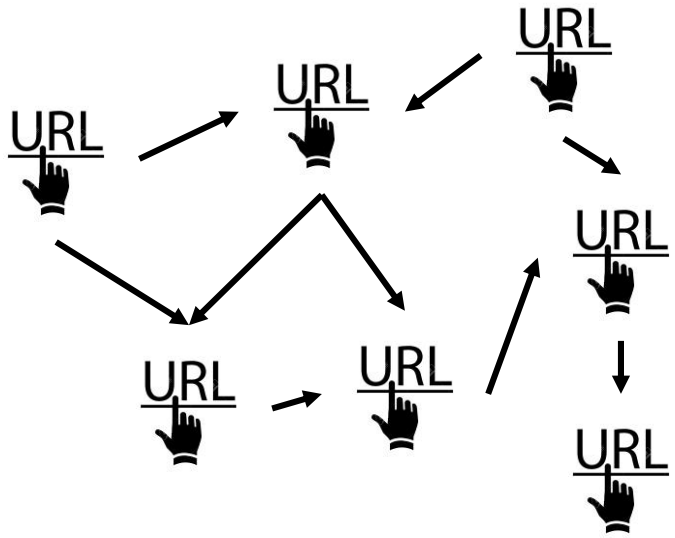
- 爬虫简介
- BeautifulSoup解析网页
- 爬虫框架Scrapy基础
- 实战案例：获取电商网站的商品信息

目录

爬虫简介

爬虫

- 自动抓取互联网信息的程序
- 利用互联网数据进行分析、开发产品



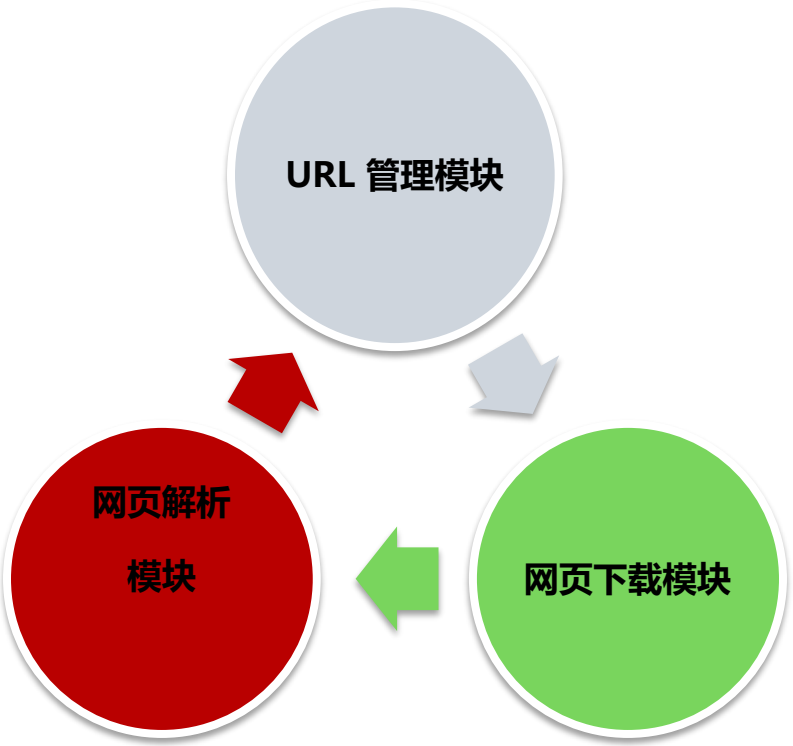
爬虫简介

爬虫基本架构

- URL 管理模块
  - 对计划爬取的或已经爬取的URL进行管理
- 网页下载模块
  - 将URL管理模块中指定的URL进行访问下载
- 网页解析模块
  - 解析网页下载模块中的URL，处理或保存数据
  - 如果解析到要继续爬取的URL，返回URL管理模块继续循环

爬虫简介

爬虫基本架构



爬虫简介

URL管理模块

- 防止重复爬取或循环指向
- 实现方式
  - Python的set数据结构，原因？
  - 数据库中的数据表，how?
  - 缓存数据库Redis，适用于大型互联网公司

爬虫简介

URL下载模块

- 将URL对应的网页下载到本地或读入内存(字符串)
- 实现方式
  - urllib2，Python官方基础模块
  - requests或其他第三方的模块
- 通过URL直接下载

```
response = urllib2.urlopen(url)
response.getcode()
response.read()
```

示例代码： 01_crawl_basic.ipynb

爬虫简介

URL下载模块 (续)

- 通过Request访问下载

```
request = urllib2.Request(url)
request.add_heard()
request.add_data()
response = urllib2.urlopen(request)
```

示例代码： 01_crawl_basic.ipynb

爬虫简介

URL下载模块 (续)

- 通过Cookie访问下载
- 使用cookielib模块
- cookie_jar = cookielib.CookieJar()

  opener = urllib2.build_opener()

  urllib2.install_opener(opener)

  response = urllib2.urlopen(url)

示例代码： 01_crawl_basic.ipynb

爬虫简介

网页解析模块

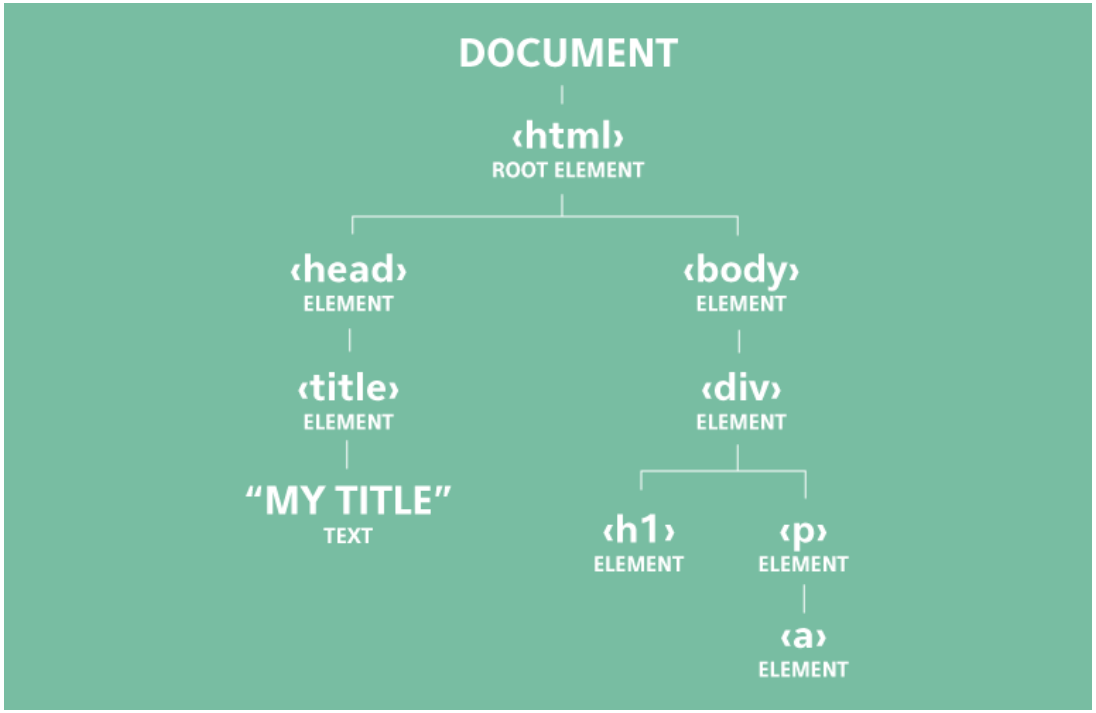
- 从已下载的网页中爬取数据
- 实现方式
  - 正则表达式，字符串的模糊匹配
  - html.parser
  - BeautifulSoup，结构化的网页解析
  - lxml

示例代码： 01_crawl_basic.ipynb

爬虫简介

网页解析模块 (续)

- 结构化解析
- DOM (Document Object Model)，树形结构



目录

BeautifulSoup解析网页

BeautifulSoup

- 用于解析HTML或XML
- conda install -c asmeurer beautiful-soup=4.3.2
- import bs4
- 步骤

  1. 创建BeautifulSoup对象
  2. 查询节点

     find，找到第一个满足条件的节点

     find_all, 找到所有满足条件的节点

BeautifulSoup解析网页

创建对象

- 创建BeautifulSoup对象
- bs = BeautifulSoup(

    url,

    html_parser, 指定解析器

    enoding 指定编码格式（确保和网页编码格式一致）

)

示例代码： 02_bs4_basic.ipynb

BeautifulSoup解析网页

查找节点

- next page
- 可按节点类型、属性或内容访问
- 按类型查找节点
  - bs.find_all('a')
- 按属性查找节点
  - bs.find_all('a', href='a.html')
  - bs.find_all('a', href='a.html', string='next page')
  - bs.find_all('a', class_='a_link')
    - 注意：是class_

示例代码： 02_bs4_basic.ipynb

BeautifulSoup解析网页

获取节点信息

- node是已查找到的节点
- node.name
  - 获取节点标签名称
- node['href']
  - 获取节点href属性
- node.get_text()
  - 获取节点文字

异常处理

- 网络资源或URL是经常变动的
- 需要处理异常

示例代码： 02_bs4_basic.ipynb

BeautifulSoup解析网页

BeautifulSoup 进阶

- 使用CSS方式、正则表达式查找节点
- 保存解析的内容
- DOM树形结构
  - children 只返回“孩子”节点
  - desecdants 返回所有“子孙”节点
  - next_siblings 返回下一个“同辈”节点
  - previous_siblings 返回上一个“同辈”节点
  - parent 返回“父亲”节点

示例代码： 03_bs4_advanced.ipynb

BeautifulSoup解析网页

BeautifulSoup 进阶（续）

- 正则表达式
- 简单的字符串匹配可以使用字符串方法完成
- 复杂、模糊的字符串匹配使用正则表达式
  - 如：电子邮箱格式匹配
- 通过使用单个字符串描述匹配一系列符合某个语法规则的字符串
- 字符串操作的逻辑公式
- 常用语处理文本数据
- 匹配过程：依次拿出表达式和文本中的字符作比较，如果每个字符都能匹配，则匹配成功；否则失败

示例代码： 03_bs4_advanced.ipynb

BeautifulSoup解析网页

BeautifulSoup 进阶 (续)

- 正则表达式
- import re
- pattern = re.compile(‘str’) 返回pattern对象
  - 推荐使用 r’str’ 无需考虑转义字符
- pattern.match()
- 基本语法
  - https://msdn.microsoft.com/zh-cn/library/ae5bf541(v=vs.90).aspx

示例代码： 03_bs4_advanced.ipynb

目录

爬虫框架Scrapy基础

Scrapy简介

- 开源的爬虫框架
- 快速强大，只需编写少量代码即可完成爬取任务
- 易扩展，添加新的功能模块
- 用户群
  - https://scrapy.org/companies/

爬虫框架Scrapy基础

Scrapy抓取过程

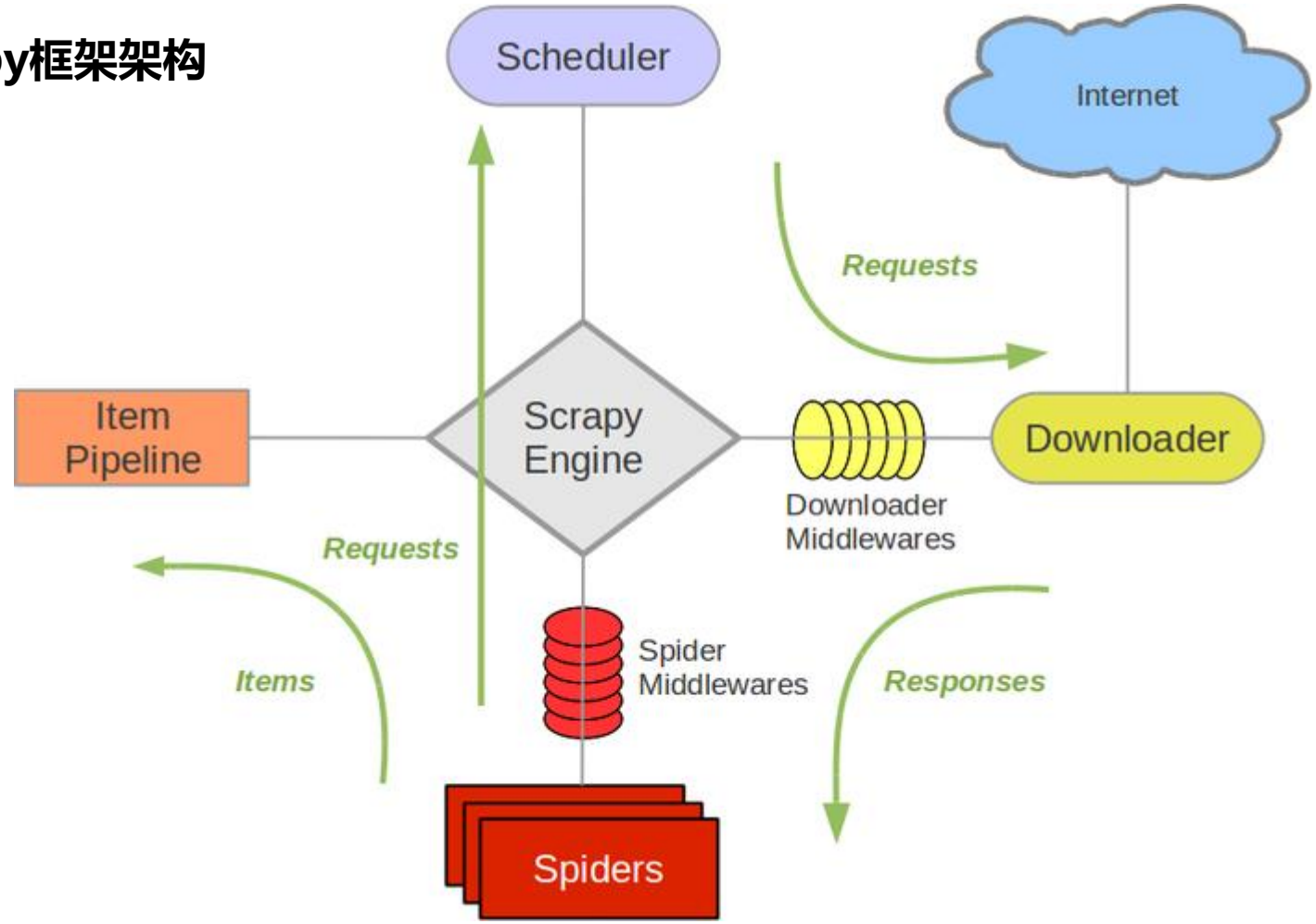
- 使用start_urls作为初始url生成Request，默认将parse作为他的回调函数
- 在parse函数中解析目标url

Scrapy高级特性

- 内置数据抽取器css/xpath/re
- 交互式控制台用于调试
- 结果输出的格式支持，JSON，CSV，XML等
- 自动处理编码
- 支持自定义扩展

爬虫框架Scrapy基础

Scrapy框架架构

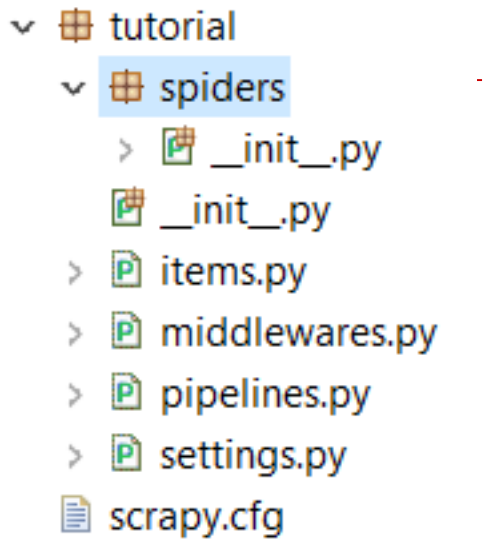


爬虫框架Scrapy基础

Scrapy使用步骤

- 安装：conda install -c anaconda scrapy
- 1．创建工程
- 2．定义Item，构造爬取的对象（可选）
- 3．编写Spider，爬虫主体
- 4．编写配置和Pipeline，用于处理爬取的结果（可选）
- 5．执行爬虫

爬虫框架Scrapy基础



Scrapy使用步骤

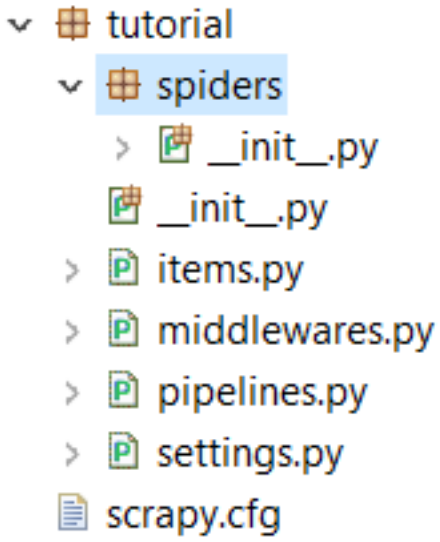
1. 创建工程
   - scrapy startproject tutorial
   - 目录结构
3. 编写Spider
   - scrapy genspider amazon_spider https://www.amazon.cn/%E5%9B%BE%E4%B9%A6/b/ref=topnav_storetab_b?ie=UTF8&node=658390051
5. 运行Spider
   - scrapy crawl amazon_spider

示例代码：lecture04_scrapy.zip

爬虫框架Scrapy基础

Scrapy使用步骤

2. 定义Item
   - scrapy.Field()
3. 编写Spider
   - 调用自定义的Item
4. pipelines
   - 默认return item
5. 运行Spider
   - scrapy crawl amazon_spider



示例代码：lecture04_scrapy.zip

爬虫框架Scrapy基础

Scrapy常用命令

- help: 查看帮助，scrapy --help
- version: 查看版本信息，
  - scrapy version，查看scrapy版本
  - scrapy version –v，查看相关模块的版本
- startproject， 新建工程，scrapy startproject porj_name
- genspider，生成spider模板，scrapy genspider spider_name url

爬虫框架Scrapy基础

Scrapy常用命令 (续)

- list，列出所有的spider，scrapy list
- view，返回网页源代码并在浏览器中打开，scrapy view url
  - 有时页面渲染的结果和查看结果是不同的
- parse，调用工程spider中的parse解析url，scrapy parse url
- shell，进入交互式调试模式，scrapy shell url
- bench，可以用来检测scrapy是否安装成功
- …

目录

实战案例

项目介绍

- 通过Scrapy框架爬取amazon图书销售排行榜

- **项目任务**

1．获取单页面数据

2．手动获取多页面数据

3．自动获取多页面数据

示例代码：lecture04_proj.zip

实战案例

涉及知识点

- Python面向对象编程
- Scrapy框架
- xpath
- 数据保存
  - CSV
  - JSON
  - XML

示例代码：lecture04_proj.zip

参考

- BeautifulSoup

  https://www.crummy.com/software/BeautifulSoup/bs4/doc/
- 正则表达式

  http://www.regexlab.com/zh/regref.htm
- Scrapy

  https://scrapy.org/
- Xpath教程

  http://www.w3school.com.cn/xpath/
- Scrapy命令行

  https://doc.scrapy.org/en/latest/topics/commands.html

疑问

- 问题答疑： http://www.xxwenda.com/
  - 可邀请老师或者其他人回答问题

小象问答 @Robin_TY

联系我们

小象学院：互联网新技术在线教育领航者

- 微信公众号：小象
- 新浪微博：ChinaHadoop