网络爬虫

目录

- 认识爬虫-为什么要学习爬虫
- 爬虫原理
- Scrapy爬虫框架
- 创建爬虫项目
- 电影数据爬取
- 新的问题-特征提取
- 网络爬虫与正则表达式
- cookie的使用

认识爬虫

爬虫

网络爬虫（Web Spider）

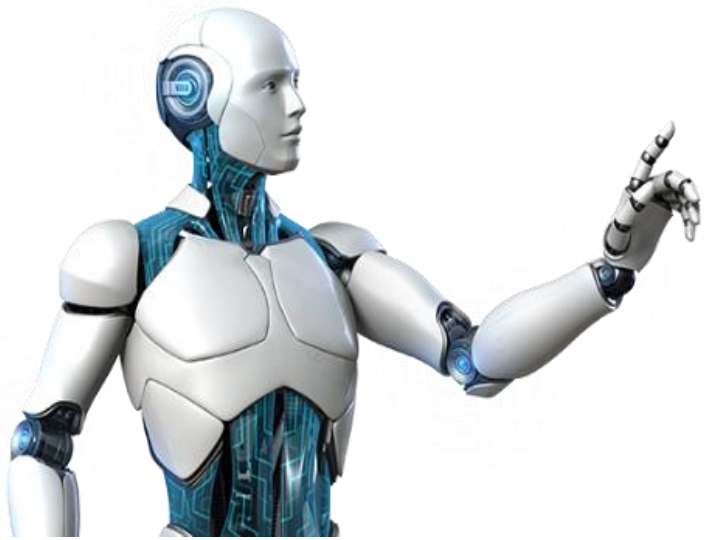
-又叫网页蜘蛛、网络机器人

按照一定的规则，自动地抓取万维网信息的程序或者脚本。它是搜索引擎重要的组成部分

将庞大的互联网看做是一张大网，而我们要做的就是用代码去构造一个类似于爬虫的实体，在这张大网上爬取我们需要的数据。

为什么要学习爬虫


BIG
DATA
ANALYTICS
COMPLEX
SOCIAL
INTERNET
STORAGE
MANAGEMENT
SETS
PETABYTES
SYSTEMS
INCLUDE
SHARED
EXAMPLES
EVERY
DISK
TARGET
APPLIED
SENSOR
DEFINITION
MOVING
WITHIN
PRACTITIONERS
CAPTURE
BUSINESS
TERABYTES
ABILITY
SIZE
TOLERABLE
PARALLEL
SAN
MASSIVELY
GROW
SEARCH
DATABASES
NETWORKS
HUNDREDS
RECORDS
CAPACITY
BIOLOGICAL
PROCESSING
RELATIONAL
CONTINUES
CITATION
LARGER
TENS
ALSO
TOOLS
DISTRIBUTED
DIFFICULTY
GARTNER
WORKING
ELAPSED
CASE
MAY
CURRENT



requests

Requests介绍

Requests-HTTP FOR HUMANS

实现python的网络连接

- ✓ 完美替代python的urllib2模块
- ✓ 更多的自动化
- ✓ 更友好的用户体验
- ✓ 更完善的功能

requests

Requests安装

Windows:pip install requests

Linux:pip install requests

第一个网络爬虫

把豆瓣电影主页的所有内容全部爬取下来

https://movie.douban.com/top250

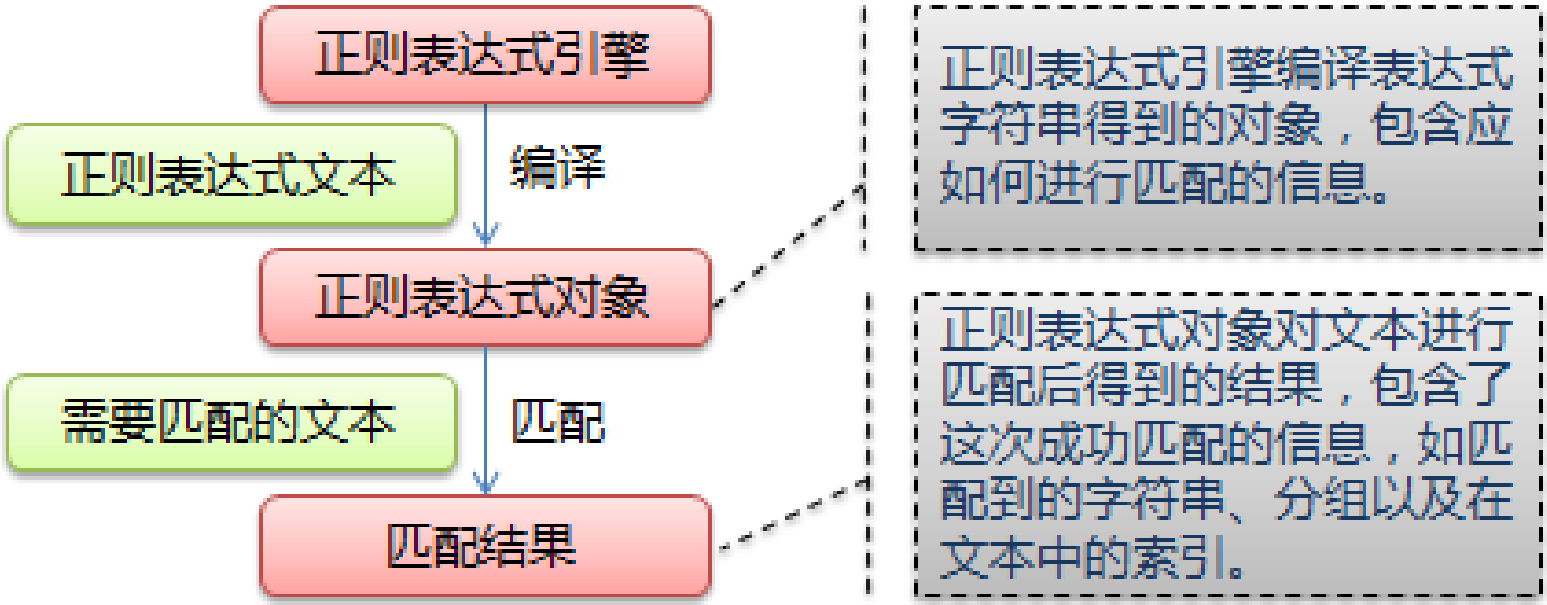
网络爬虫与正则表达式

正则表达式

- 正则表达式是一个特殊的字符序列，它能帮助你方便的检查一个字符串是否与某种模式匹配。
- 如果说网页爬虫爬取的网页信息是数据大海的话，正则表达式就是我们进行“大海捞针”的工具。

正则表达式

正则表达式匹配流程



正则表达式重要符号

符号	描述	符号	描述
\w	匹配字母、数字、下划线	.	匹配任意字符，包括汉字
\W	匹配不是字母、数字、下划线的字符	[m]	匹配单个字符串
\s	匹配空白字符	[m1m2...n]	匹配多个字符串
\S	匹配不是空白的字符	[m-n]	匹配m到n区间内的数字、字母
\d	匹配数字	[^m]	匹配除m以外的字符串
\D	匹配非数字的字符	()	对正则表达式进行分组，一对圆括号表示一组
*	重复0或N次	{m}	重复m次
+	重复1或N次	{m,n}	该限定符的意思是至少有 m 个重复，至多到 n 个重复
?	重复0或1次		

正则表达式重要符号

网络爬虫与正则表达式

- .：匹配任意字符，**换行\n除外**
- *：匹配前一个字符0次或无限次
- ？：匹配前一个字符0次或一次
- .*：贪心算法
- .*?：非贪心算法
- （）内的数据作为结果输出

Python re模块

- re 模块也提供了与这些方法功能完全一致的函数，这些函数使用一个模式字符串做为它们的第一个参数。
- re 模块使 Python 语言拥有全部的正则表达式功能。

```
import re
```

Python re模块

1.re.findall函数

Python 的 re 模块提供了re.findall用于匹配所有符合规律的内容，返回包含结果的列表

```
re.findall(pattern, string, flags=0)
```

参数	描述
pattern	匹配的正则表达式
string	要被查找的原始字符串。
flags	标志位，用于控制正则表达式的匹配方式，如：是否区分大小写，多行匹配等。。

Python re模块

练习：

1、豆瓣电影中，使用re查找所有电影的评价人数

2、找出所有电影名称

1

肖申克的救赎 / The Shawshank Redemption / 月黑高飞(港) / 刺激1995(台) [可播放]

导演: 弗兰克·德拉邦特 Frank Darabont 主演: 蒂姆·罗宾斯 Tim Robbins /...
1994 / 美国 / 犯罪 剧情

★★★★★ 9.6 943723人评价

“ 希望让人自由。 ”

2

霸王别姬 / 再见，我的妾 / Farewell My Concubine [可播放]

导演: 陈凯歌 Kaige Chen 主演: 张国荣 Leslie Cheung / 张丰毅 Fengyi Zha...
1993 / 中国大陆 香港 / 剧情 爱情 同性

★★★★★ 9.5 683484人评价

“ 风华绝代。 ”

Python re模块

正则表达式修饰符 - 可选标志

修饰符	描述
re.I	使匹配对大小写不敏感
re.L	做本地化识别（locale-aware）匹配
re.M	多行匹配，影响 ^ 和 $
re.S	使 . 匹配包括换行在内的所有字符
re.U	根据Unicode字符集解析字符。这个标志影响 \w, \W, \b, \B.
re.X	该标志通过给予你更灵活的格式以便你将正则表达式写得更易于理解。

练习

练习：

1、获取豆瓣电影页面中如下公司的信息


```
<span id="icp" class="fleft gray-link">
    &copy; 2005 - 2018 douban.com, all rights reserved 北京豆网科技有限公司
</span>
```

Python re模块

2.re.search函数

- re.search 扫描整个字符串并返回第一个成功的匹配。
- 注意要用re.search().group(1)获取具体值

```
re.search(pattern, string, flags=0)
```

参数	描述
pattern	匹配的正则表达式
string	要匹配的字符串。
flags	标志位，用于控制正则表达式的匹配方式，如：是否区分大小写，多行匹配等。

Python re模块

3.re.sub函数

Python 的 re 模块提供了re.sub用于替换字符串中的匹配项。返回替换后的值

```
re.sub(pattern, repl, string, count=0, flags=0)
```

参数	描述
pattern	匹配的正则表达式
repl	替换的字符串，也可为一个函数
string	要被查找替换的原始字符串。
count	模式匹配后替换的最大次数，默认 0 表示替换所有的匹配。

Python re模块

练习：在豆瓣电影中，评分人数的格式如下所示，现只想提取出数字，使用正则表达式如何实现？

来自：豆瓣电影

战狼2

★★★★☆ 7.4 (393283人评价)

导演: 吴京

主演: 吴京 / 弗兰克·格里罗 / 吴刚

类型: 动作

制片国家/地区: 中国大陆

年份: 2017

Python re模块

练习：有如下一段话，请使用正则表达式提取出电话号码?

“我的电话号码是：010-6737-2234，请保存好了”

xpath

- Xpath是是一门在 XML 文档中查找信息的语言
- Xpath支持HTML
- Xpaht通过元素和属性进行导航

- ✓ Xpath可以用来提取信息
- ✓ Xpaht比正则表达式厉害
- ✓ Xpath比正则表达式简单

Xpath安装

- 安装lxml库

  pip install lxml

- from lxml import etree
- Selector=etree.HTML(网页源代码)

Xpath与HTML结构

- 树状结构
- 逐层展开
- 逐层定位
- 寻找独立节点

Xpath提取内容

- //定位根节点
- /往下层寻找
- 提取文本内容：/text()
- 提取属性内容：/@xxxx

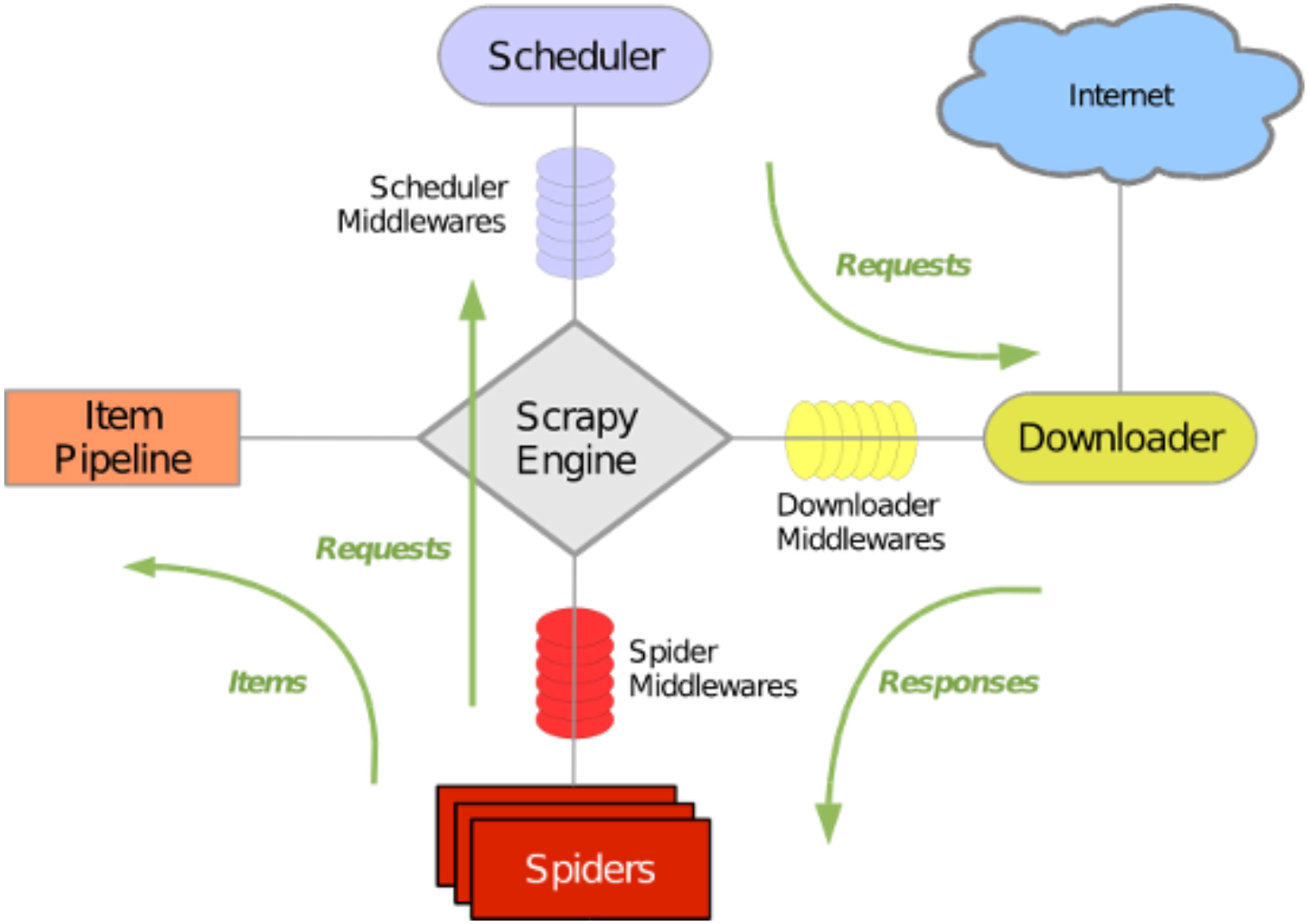
Scrapy爬虫框架

Scrapy爬虫框架

Scrapy = Scratch python

- Scrapy是一个为了爬取网站数据，提取结构性数据而编写的应用框架。 可以应用在包括数据挖掘，信息处理或存储历史数据等一系列的程序中。
- 基于python的快速、高层次的屏幕抓取和Web抓取框架，用于抓取Web站点并从页面中提取结构化的数据
- Scrapy用途广泛，可以用于数据挖掘、监测和自动化测试。

Scrapy爬虫框架



Scrapy爬虫框架

- 引擎(Scrapy Engine)：用来处理整个系统的数据流处理，触发事务。
- 调度器(Scheduler)：用来接受引擎发过来的请求，压入队列中，并在引擎再次请求的时候返回。
- 下载器(Downloader)：用于下载网页内容，并将网页内容返回给蜘蛛。
- 爬虫(Spiders)：*爬虫是主要干活的，用于从特定的网页中提取自己需要的信息，即所谓的实体(Item)。用户也可以从中提取出链接，让Scrapy继续抓取下一个页面*
- 项目管道(Pipeline)：负责处理有蜘蛛从网页中抽取的项目，他的主要任务是清晰、验证和存储数据。当页面被蜘蛛解析后，将被发送到项目管道，并经过几个特定的次序处理数据。

Scrapy爬虫框架

- 下载器中间件(Downloader Middlewares)：位于Scrapy引擎和下载器之间的钩子框架，主要是处理Scrapy引擎与下载器之间的请求及响应。
- 爬虫中间件(Spider Middlewares)：介于Scrapy引擎和蜘蛛之间的钩子框架，主要工作是处理蜘蛛的响应输入和请求输出。
- 调度中间件(Scheduler Middewares)：介于Scrapy引擎和调度之间的中间件，从Scrapy引擎发送到调度的请求和响应。

使用Scrapy可以很方便的完成网上数据的采集工作，它为我们完成了大量的工作，而不需要自己费大力气去开发

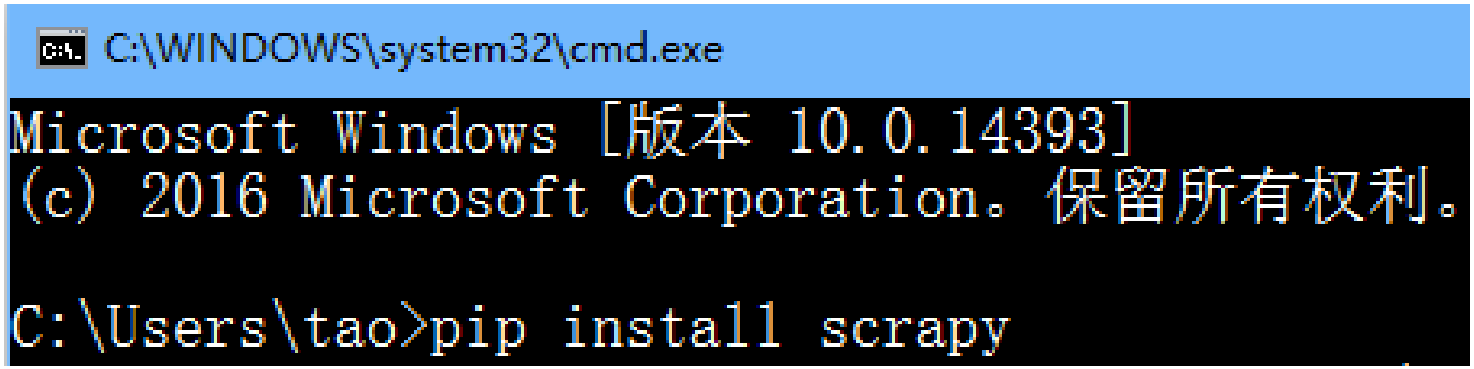
Scrapy爬虫框架

Scrapy运行流程

创建爬虫项目

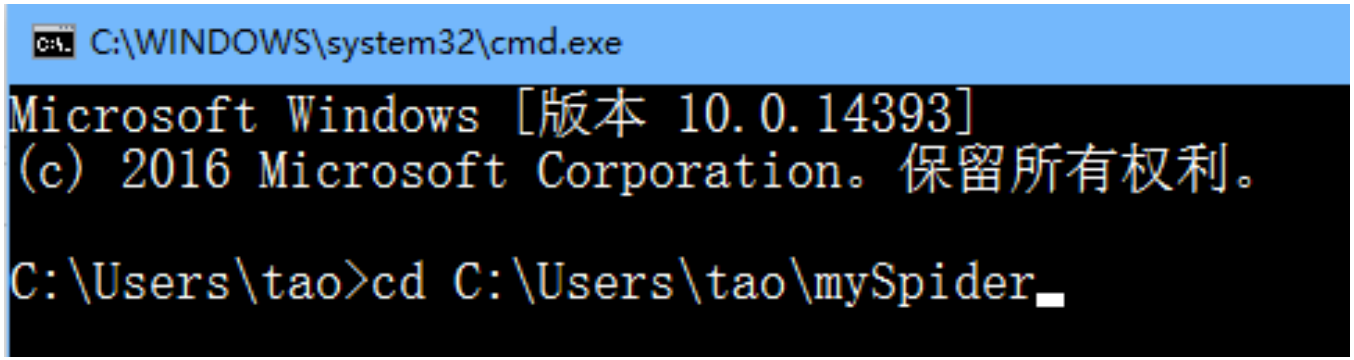
scrapy安装

使用pip安装scrapy：pip install scrapy



创建项目

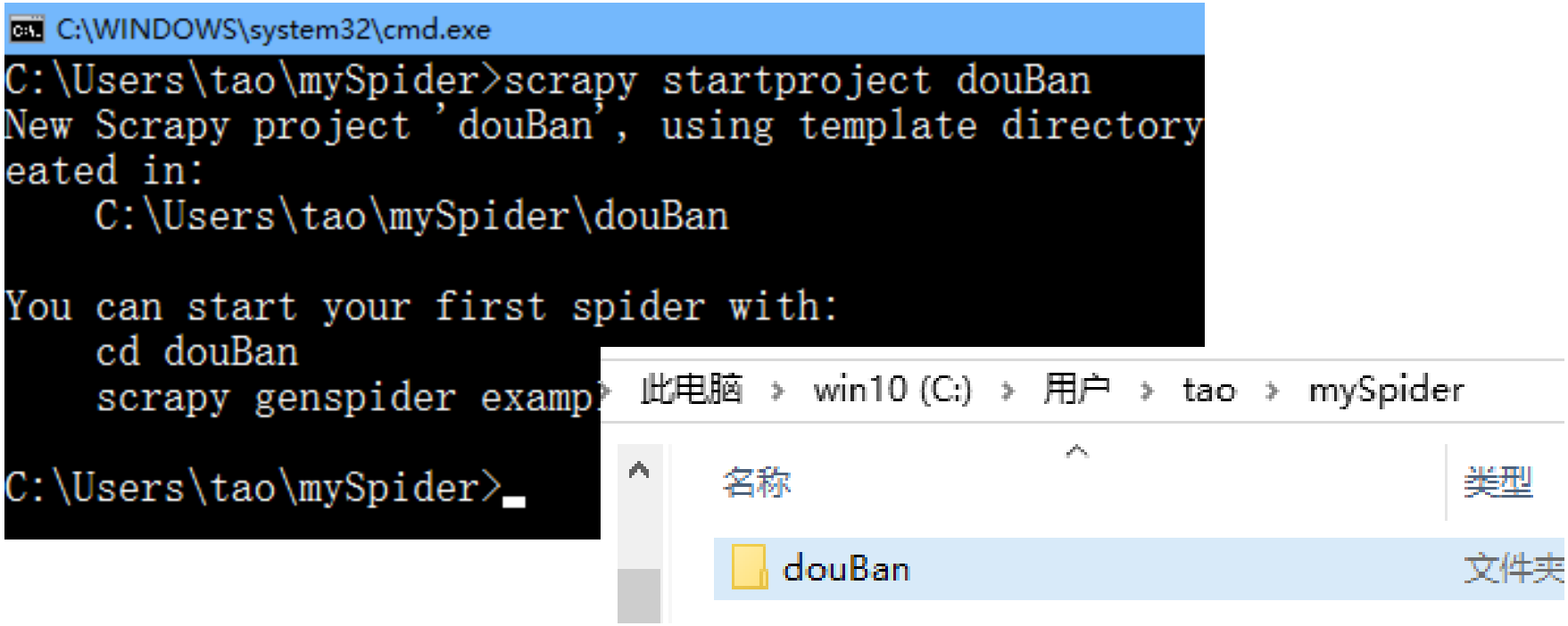
1.使用命令进入存储项目的目录



创建项目

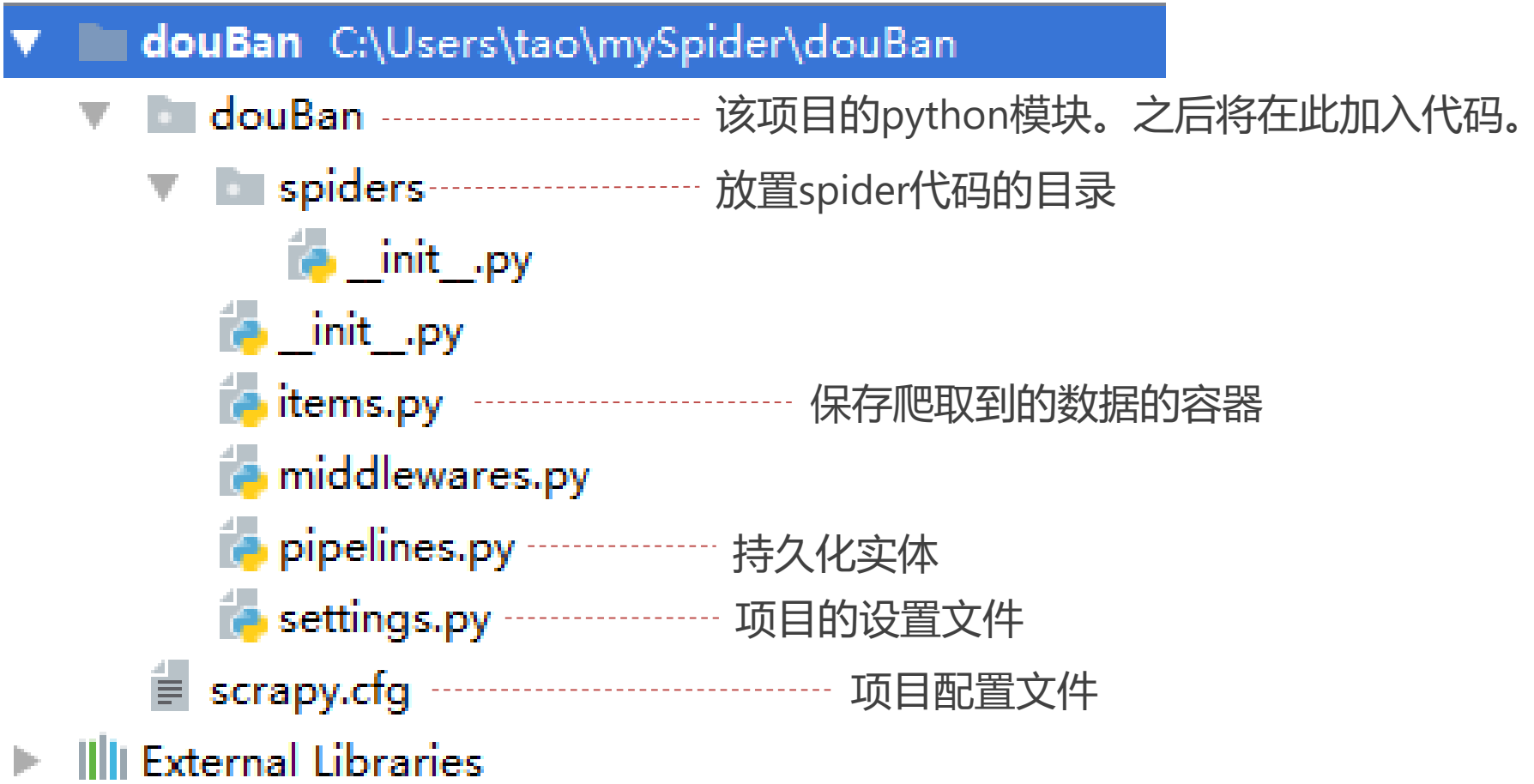
2.运行如下命令，生成一个项目

scrapy startproject 项目名



创建项目

3.运行PyCharm，打开项目



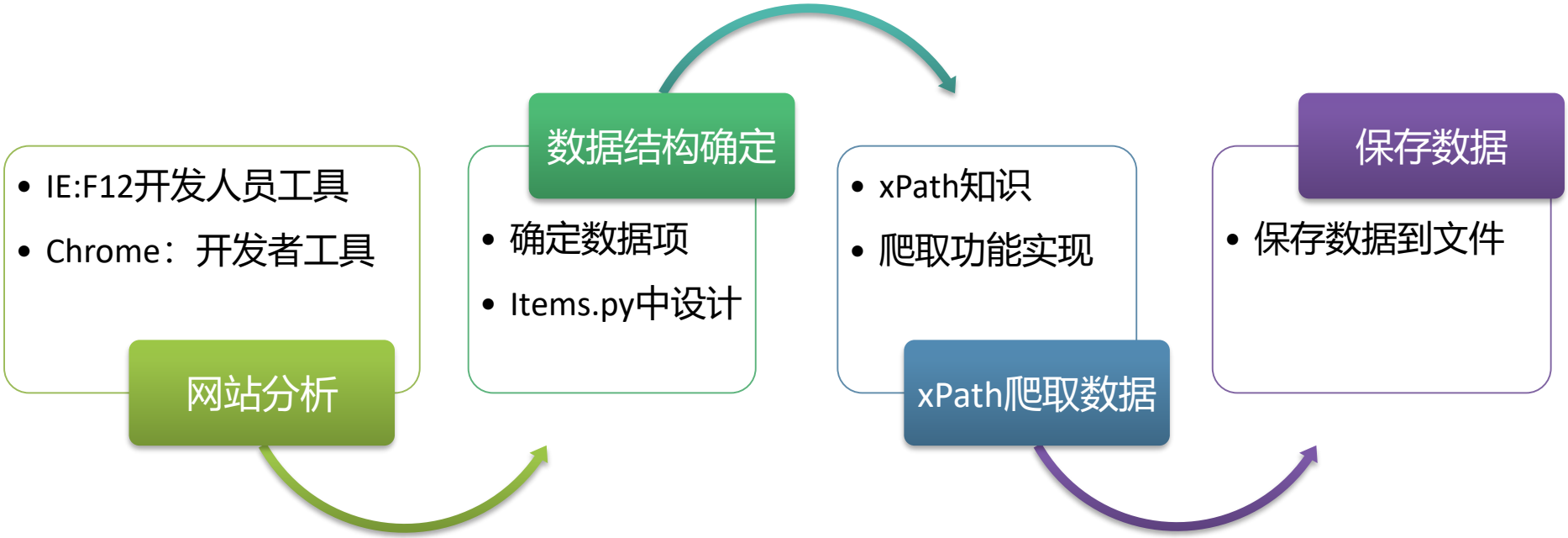
豆瓣电影信息爬取

3.功能实现：爬取豆瓣电影中的电影票房信息

https://www.douban.com/doulist/1295618/

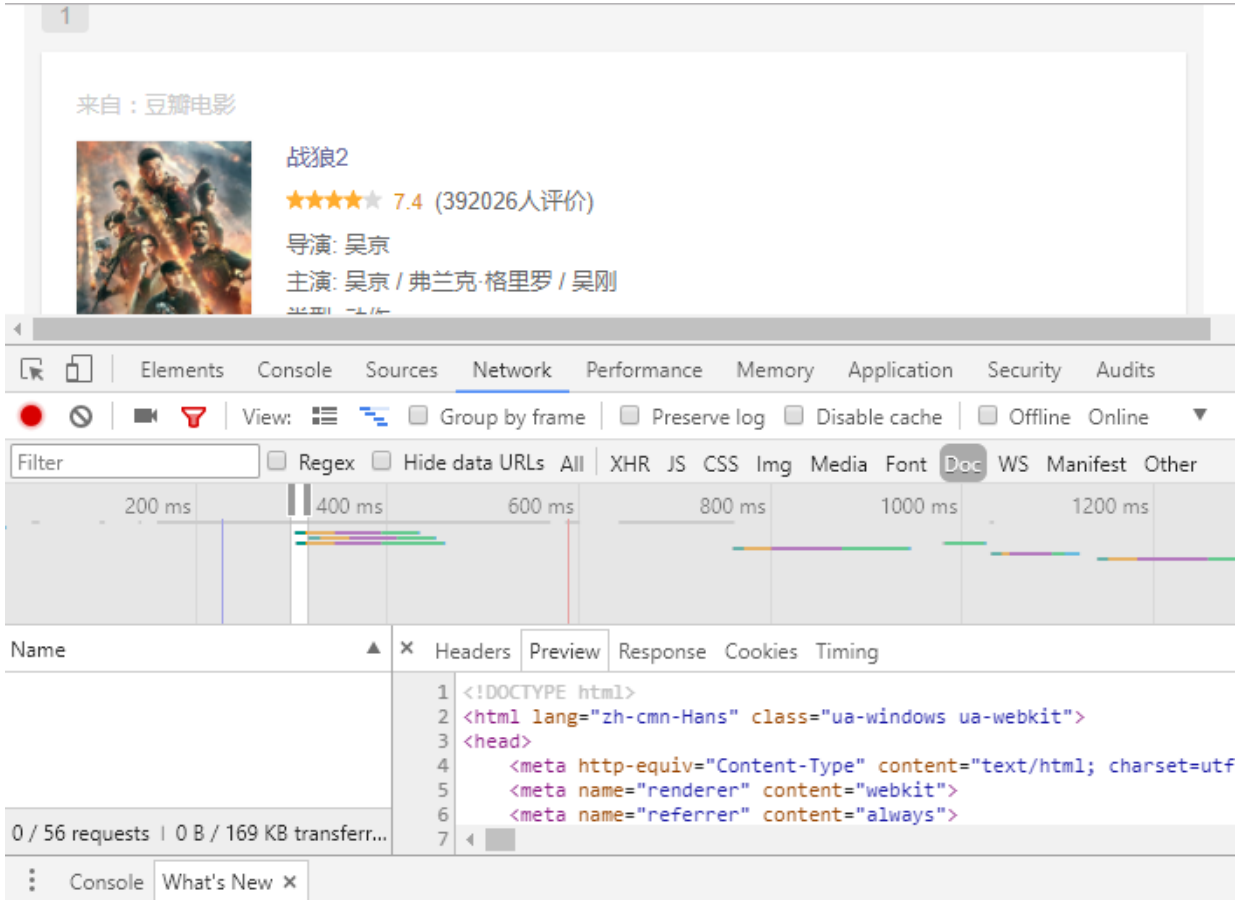
豆瓣电影信息爬取

功能实现步骤



豆瓣电影-网站分析

网站分析-使用开发者工具



豆瓣电影-数据结构

数据结构确定

- 排名
- 电影名称
- 导演
- 主演
- 类型
- 国家
- 年份
- 评分
- 总票房

豆瓣电影-数据结构

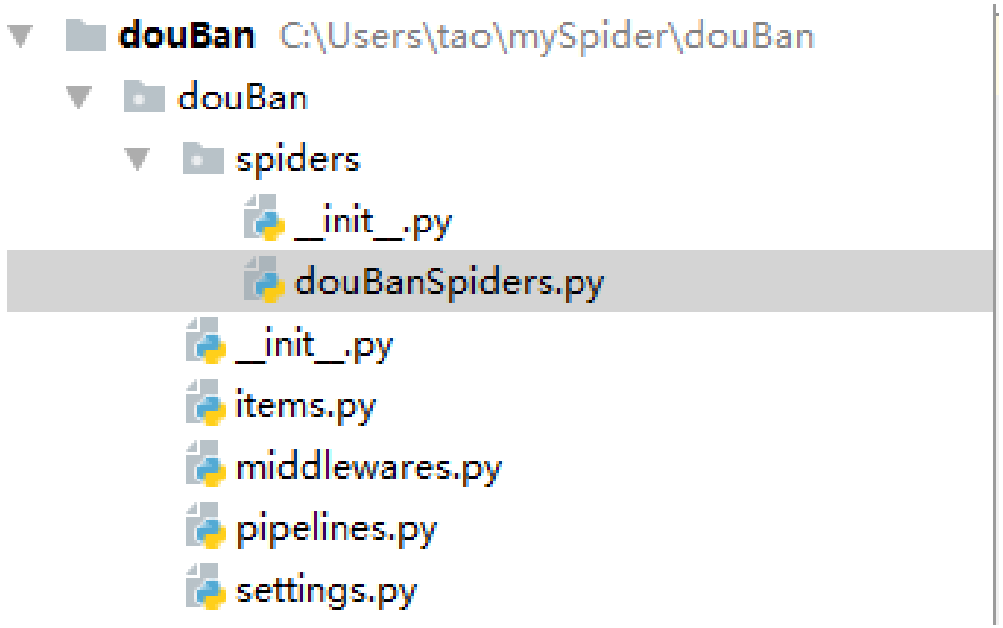
数据结构确定

```
import scrapy

class DoubanMovieItem(scrapy.Item):
    # 排名
    ranking = scrapy.Field()
    # 电影名称
    movie_name = scrapy.Field()
    # 评分
    score = scrapy.Field()
    ......
```

数据爬取-新建爬取源文件

1.新建数据爬取源文件



新建douBanSpider.py

定义Spiders类

使用选择器提取数据

保存数据

数据爬取-定义Spider类

2.定义spider类

douBanSpiders.py

```
1  #encoding:utf-8
2  import scrapy
3
4  class DouBanMovieTopAllSpider(scrapy.Spider):
5      name = 'douban_movie'
6      def start_requests(self): #初始请求(request)
7          pass
8      def parse(self, response): #处理响应（response)的内容
9          pass
```

继承 **scrapy.Spider** 基类

- name:定义爬虫名，必须定义
- start_requests():包含了spider用于爬取的第一个Request
- parse():负责处理response并返回处理的数据以及(/或)跟进的URL

数据爬取-start_request()方法

2.定义spider类- start_requests（）方法

```
def start_requests(self): #初始请求(request)
    url="https://www.douban.com/doulist/1295618/" #初始网址
    # yield:类似return，不同之处在于，yield返回的是一个生成器
    yield  scrapy.Request(url)
```

数据爬取- start_request()方法

反爬虫措施：有些网站有反爬虫措施，因此运行程序就会出错

```
2017-10-18 14:11:58 [scrapy.spidermiddlewares.httperror] INFO: Ignoring response <40
3 https://www.douban.com/doulist/1295618/>: HTTP status code is not handled or not a
llowed
2017-10-18 14:11:58 [scrapy.core.engine] INFO: Closing spider (finished)
2017-10-18 14:11:58 [scrapy.statscollectors] INFO: Dumping Scrapy stats:
```

settings.py

```
13
14  SPIDER_MODULES = ['douBan.spiders']
15  NEWSPIDER_MODULE = 'douBan.spiders'
16  USER_AGENT = 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_3) ' \
17               'AppleWebKit/536.5 (KHTML, like Gecko) ' \
18               'Chrome/19.0.1084.54 Safari/536.5'
```

数据爬取-parse（）方法

2.定义spider类- parse（）方法

- parse 负责处理response并返回处理的数据以及(/或)跟进的URL
- Spider 对其他的Request的回调函数也有相同的要求
- 该方法及其他的Request回调函数必须返回一个包含 Request 及(或) Item 的可迭代的对象。

参数:response (Response) – 用于分析的response

```
def parse(self, response): #处理响应（response)的内容
    #实现数据爬取功能
    pass
```

数据爬取-parse（）方法

3. 提取数据-选择器

我们如何从HTML中提取出我们想要的数据呢?

Scrapy选择器(seletors)

通过特定的 XPath 或者 CSS 表达式来“选择” HTML文件中的某个部分。

数据爬取-parse（）方法

3. 提取数据-选择器

练习一：获取豆瓣电影中title中的内容


```
def parse(self, response): #处理响应（response)的内容
    #获取title中的数据信息
    title = response.xpath("//title/text()").extract()
    print title
```

数据爬取-parse（）方法

3. 提取数据-选择器

练习二：获取豆瓣电影中所有电影的名称

```
▼<div class="bd doulist-subject">
    <div class="source">
          来自：豆瓣电影
        </div>
  ▶<div class="post">…</div>
  ▼<div class="title">
      <a href="https://movie.douban.com/subject/26363254/" target="_blank">
            战狼2
          </a>
    </div>
```

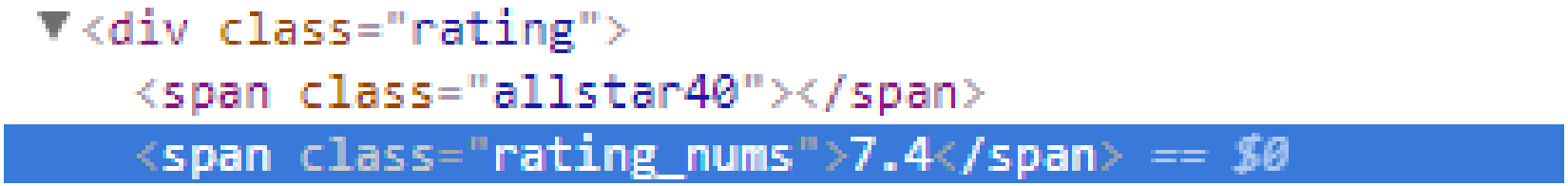
```
items = DoubanItem() #定义DoubanItem对象，用于保存数据
names = response.xpath("//div[@class='title']/a/text()").extract()
for index in range(len(names)):
```

观察一下结果，有什么问题，如何解决？

数据爬取-parse（）方法

3. 提取数据-选择器

练习三：获取豆瓣电影中所有电影评分




```
#所有的评分
scores = response.xpath('//span[@class="rating_nums"]/text()').extract()
for index in range(len(names)):
    items["score"] = scores[index]
```

数据爬取-parse（）方法

3. 提取数据-选择器

练习四：获取豆瓣电影中所有电影评分人数

```
▼<div class="rating">
    <span class="allstar40"></span>
    <span class="rating_nums">7.4</span> == $0
```

来自：豆瓣电影

战狼2

★★★★☆ 7.4 (392220人评价)

导演: 吴京

主演: 吴京 / 弗兰克·格里罗 / 吴刚

类型: 动作

制片国家/地区: 中国大陆

获取的结果是“(181180人评价)”，
而不是具体的数字

数据爬取-parse（）方法

3. 提取数据-选择器

练习五：获取豆瓣电影中导演、主演、类型、国家、年份

来自：豆瓣电影

战狼2

★★★★☆ 7.4 (392220人评价)

导演: 吴京

主演: 吴京 / 弗兰克·格里罗 / 吴刚

类型: 动作

制片国家/地区: 中国大陆

年份: 2017

```
<div class="abstract"> == $0
  "
          导演: 吴京
          "
  <br>
  "
          主演: 吴京 / 弗兰克·格里罗 / 吴刚
          "
  <br>
  "
          类型: 动作
          "
  <br>
  "
          制片国家/地区: 中国大陆
          "
  <br>
  "
          年份: 2017
```

1.如何获取到主演?

2.如何只获取第一个主演?

数据爬取-parse（）方法

3. 提取数据-选择器

练习六：获取豆瓣电影中总票房

战狼2

★★★★☆ 7.4 (392220人评价)

导演: 吴京

主演: 吴京 / 弗兰克·格里罗 / 吴刚

类型: 动作

制片国家/地区: 中国大陆

年份: 2017

评语：总票房：567647万元 | 上映日期：2017年07月27日（20:00） | 发行类别：国产（上映中……）

1.如何获取票房数据?

2.如何只取数字，即去掉文字“万元”？

数据爬取-parse（）方法

我们已经顺利获取第一页25条电影的数据

问题：如何获取豆瓣电影后续每一页的所有电影信息

<前页 1 2 3 4 5 6 7 8 9 ... 35 36 后页>

分析：

（1）使用循环，自动完成

（2）本页执行完后，需要知道下一页的网址

数据爬取-parse（）方法

3. 提取数据-选择器

练习七：实现获取豆瓣电影后续每一页的所有电影信息

思路：

（1）获取下一页的网址

<前页 1 2 3 4 5 6 7 8 9 ... 35 36 后页>

```
▼<span class="next">
   <link rel="next" href="https://www.douban.com/doulist/1295618/?start=75&sort=seq&sub_type=">
   <a href="https://www.douban.com/doulist/1295618/?start=75&sort=seq&sub_type=">后页></a>
 </span>
```

（2）自动获取下一页的数据，直到最后一页止

数据爬取-parse（）方法

152

来自：豆瓣电影

建党伟业

暂无评分

导演: 韩三平 / 黄建新

主演: 刘烨 / 冯远征 / 张嘉译

类型: 剧情 / 历史

制片国家/地区: 中国大陆/香港

年份: 2011

评语：总票房：42297万元 | 上映日期：2011年6月15日 | 发行类别：国产（合拍）

赞 (1) 回复

2011年8月25日

引入try-except出错判断，减少错误

保存数据

1. 使用命令把items中的数据保存到文件中

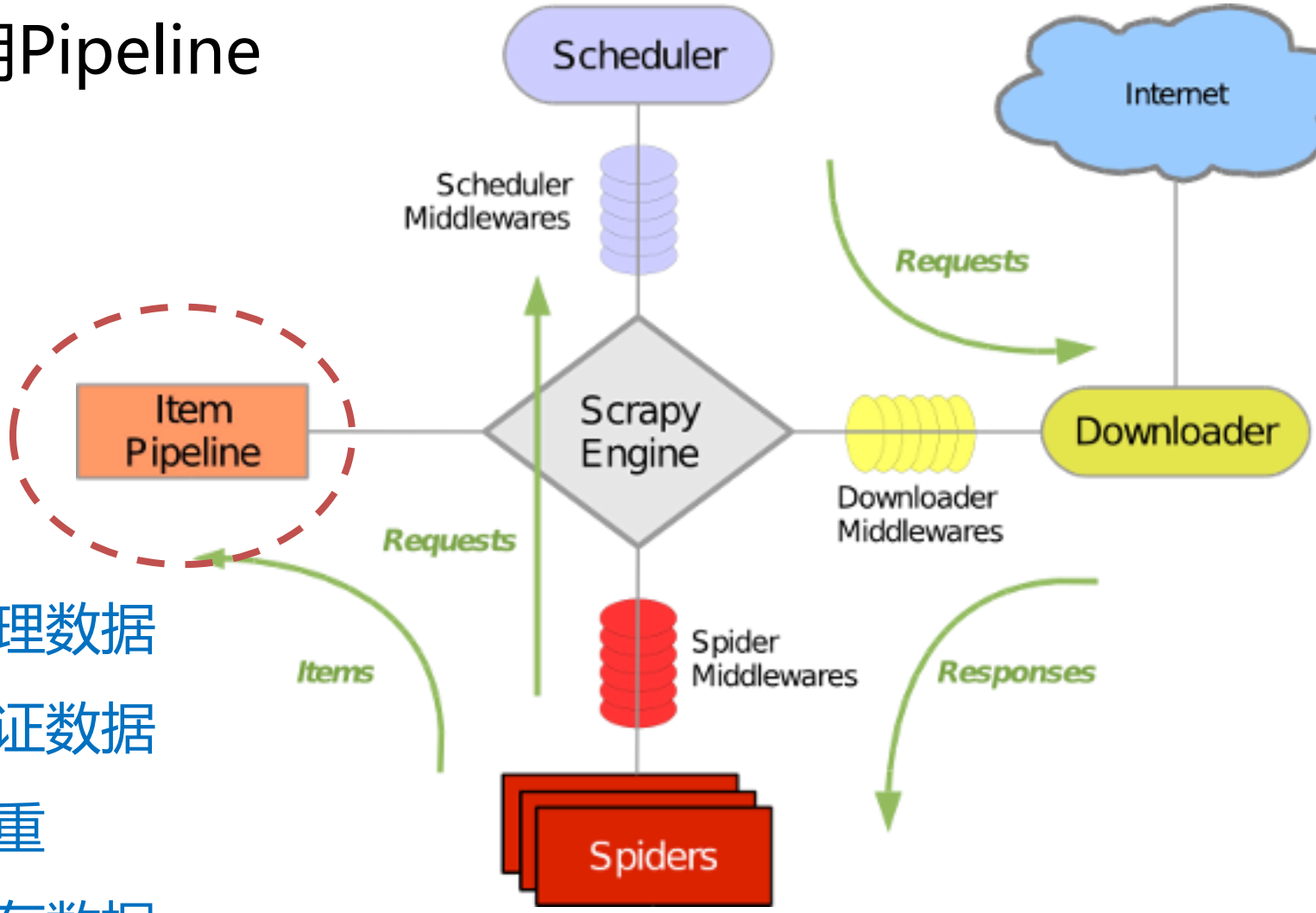
```
scrapy crawl douban_movie -o result.csv
```

支持的格式：

- ✓ Xml
- ✓ Jsonlines
- ✓ Jl
- ✓ Json
- ✓ csv
- ✓ Pickle
- ✓ marshal

保存数据

2. 使用Pipeline



- 清理数据
- 验证数据
- 查重
- 保存数据

保存数据

练习一：将数据保存于文件中，且数据之间使用#作为间隔符

```
class DoubanPipeline(object):
    def process_item(self, item, spider):
        # 获取当前工作目录
        base_dir = os.getcwd()
        fiename = base_dir + '/news1.txt'
```

settings.py

```
67  # Configure item pipelines
68  # See http://scrapy.readthedocs.org/en/latest/to
69  ITEM_PIPELINES = {
70      'douBan.pipelines.DoubanPipeline': 300,
71  }
```

保存数据

练习二：将豆瓣电影数据保存于mysql数据库中

1、安装MySQL数据库

2、下载MySQL数据库接口：MySQLdb

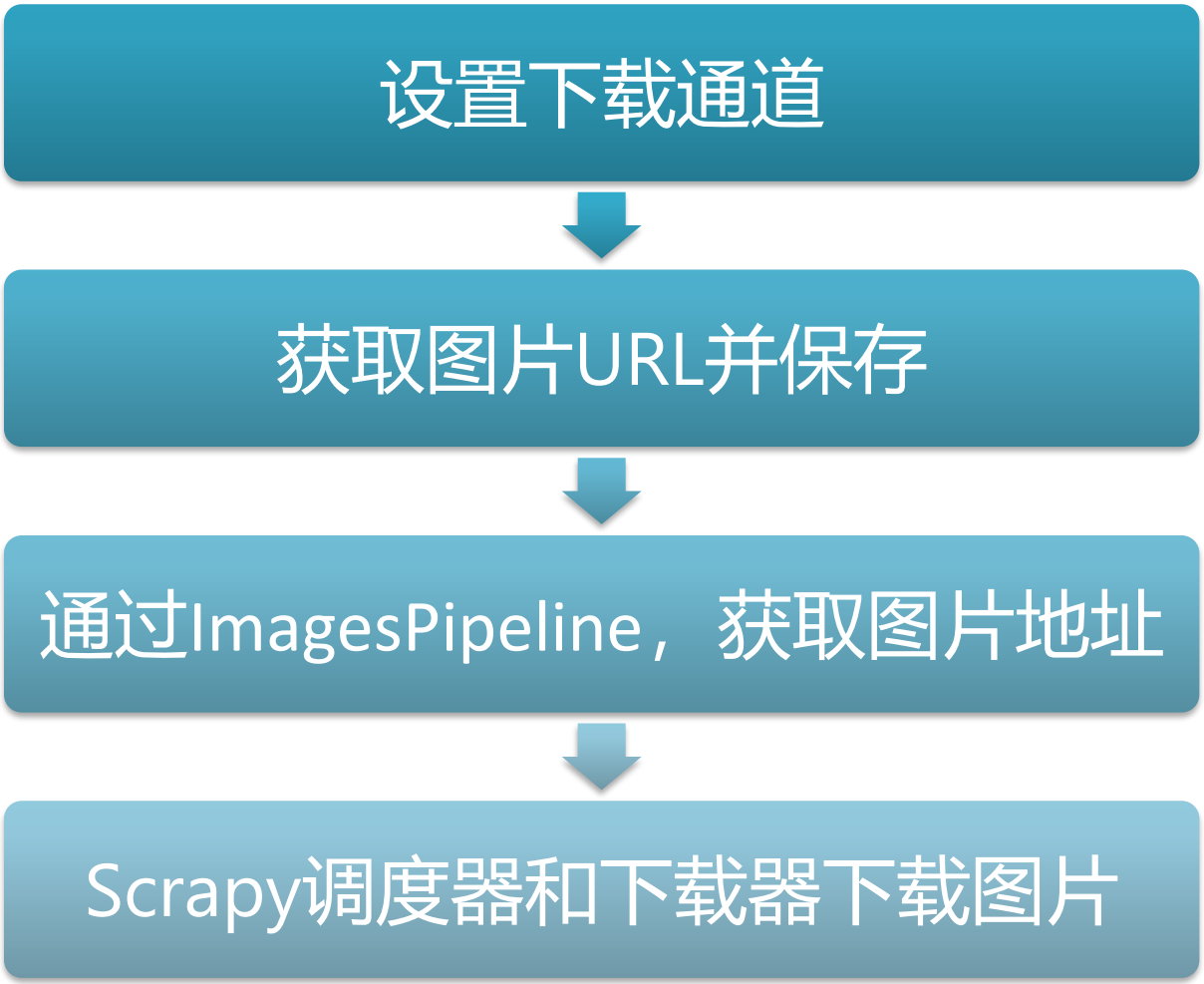
2、安装MySQLdb

3、pipeline中导入MySQLdb

4、将爬取的数据插入到MySQL数据库中

保存数据

练习三：将豆瓣电影中的图片下载到本地



新的问题-特征提取

新的问题

问题：通过爬虫获取的数据，会作为以后机器学习算法的样本数据，这些数据有什么问题呢？

```
movie_act,movie_name,movie_country,movie_director,score,movie_type,box_office,score_num
吴京,战狼2,中国大陆,吴京,7.4,动作,567647,392515
邓超,美人鱼,中国大陆,周星驰,6.8,喜剧,339323,398442
白百何,捉妖记,中国大陆,许诚毅,6.8,喜剧,243817,254229
范·迪塞尔,速度与激情7 Furious 7,美国,温子仁,8.3,动作,241600,256700
艾伦,羞羞的铁拳,中国大陆,宋阳,7.3,喜剧,184642,180336
成龙,功夫瑜伽,中国大陆,唐季礼,5.0,喜剧,175259,98542
陈坤,寻龙诀,中国大陆,乌尔善,7.5,剧情,168036,333120
吴亦凡,西游伏妖篇,中国大陆,徐克,5.6,喜剧,165678,221760
徐峥,港囧,中国大陆,徐峥,5.6,喜剧,161183,211308
```

样本数据中的文字，需要抽取数值特征，以便支持机器学习的算法

中文汉字的转换

我们需要把非数值数据转换为特征值

吴京：100
邓超：110
范·迪塞尔：120

中国大陆：200
美国：210
发过：320

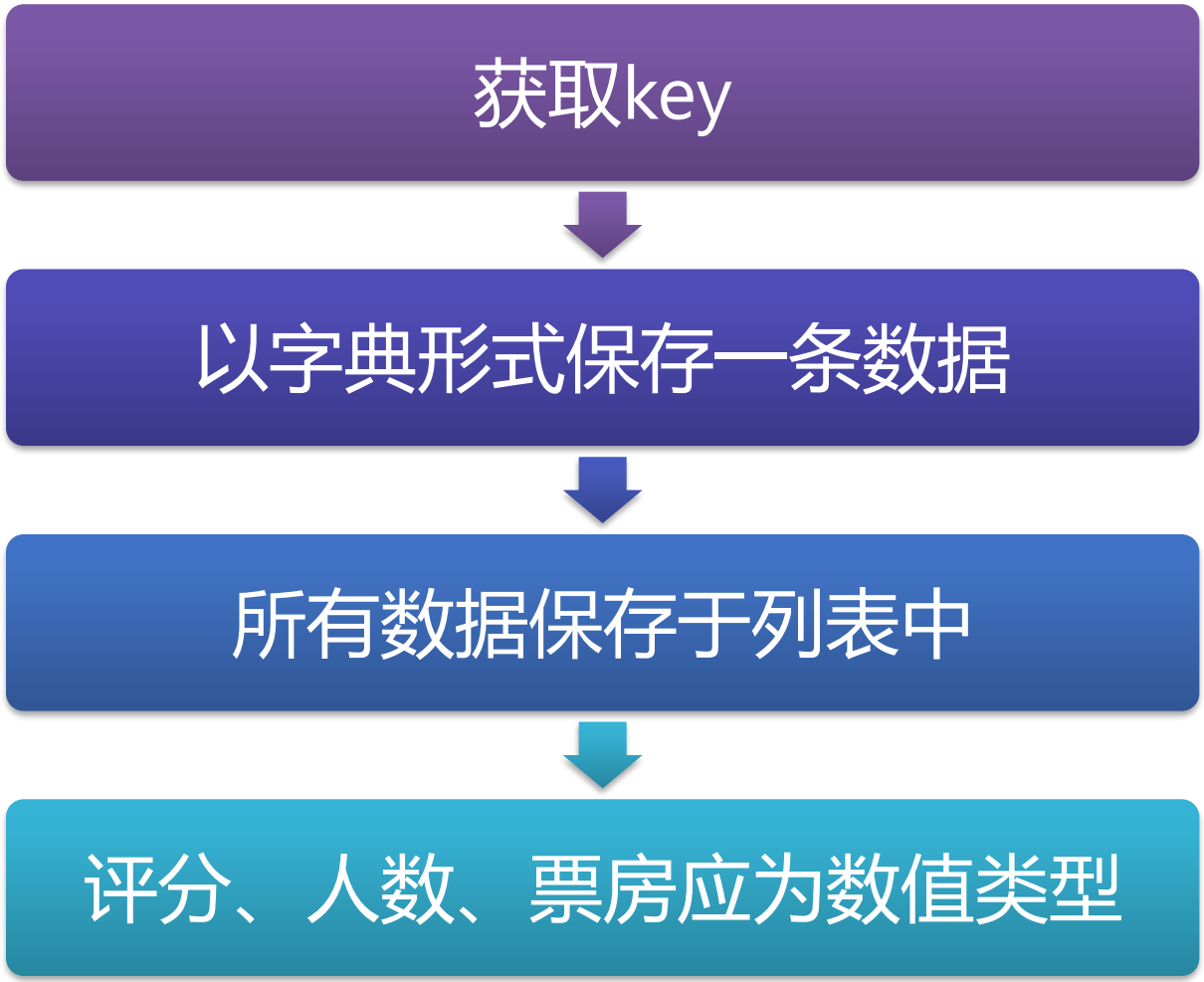
sklearn.feature_extraction模块

sklearn.feature_extraction模块，可以用于从包含文本和图片的数据集中提取特征，以便支持机器学习算法使用。

- 从文件中读取样本数据
- 提取特征
- 新样本数据
- 保存到文件

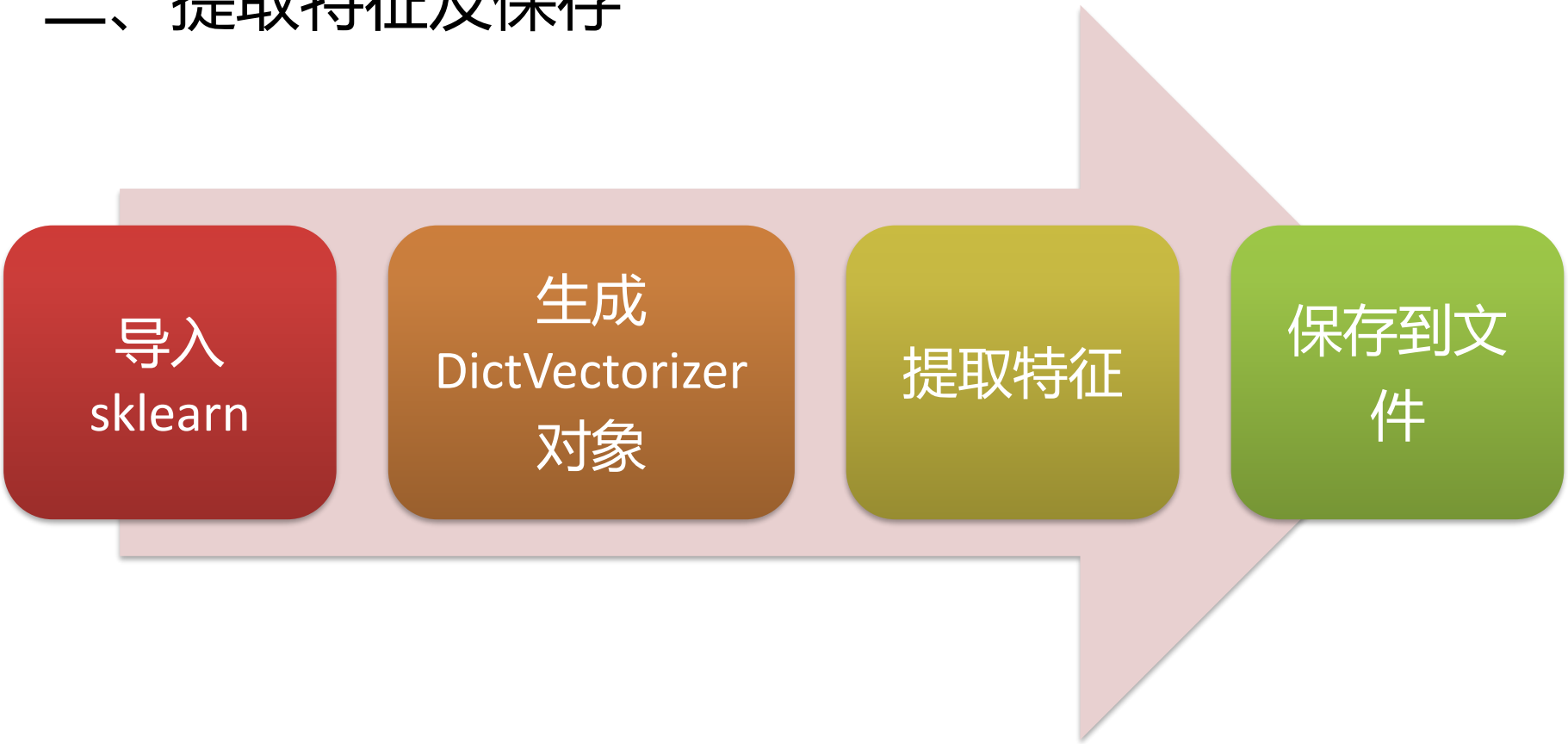
sklearn.feature_extraction模块

一、读取样本数据



sklearn.feature_extraction模块

二、提取特征及保存



练习

1、从网路上爬取本省本年度的天气预报，以csv文件保存

2、从网络上爬取二手房信息，并对文本进行特征提取，保存于mysql数据库中

3、爬取昵图网中所有自然景观的图片，下载保存到本地，地址为：

http://www.nipic.com/photo/jingguan/index.html

cookie的使用

cookie

任务：爬取豆瓣上用户首页的内容，如下图所示

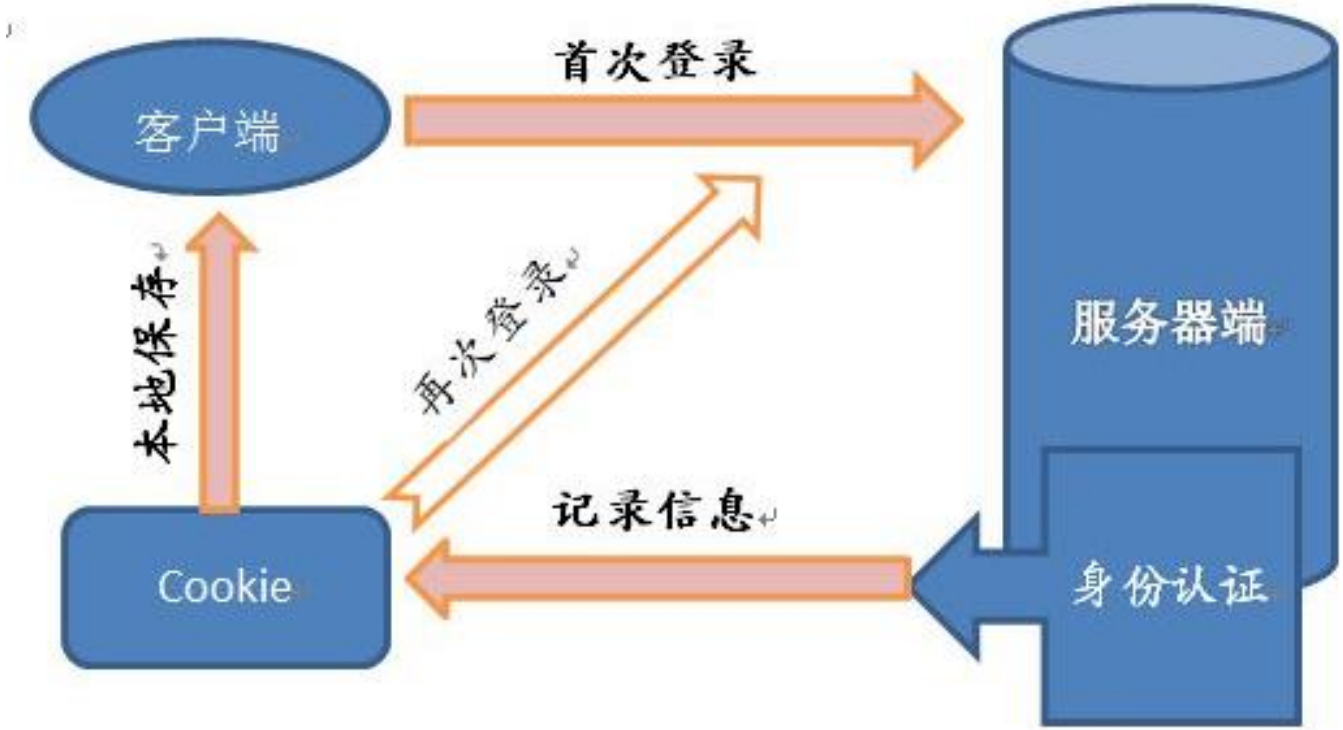
cookie

问题：页面需要登陆后才可以访问，无法爬取

HTTP是无状态的面向连接的协议，为了保持连接状态，引入了Cookie机制

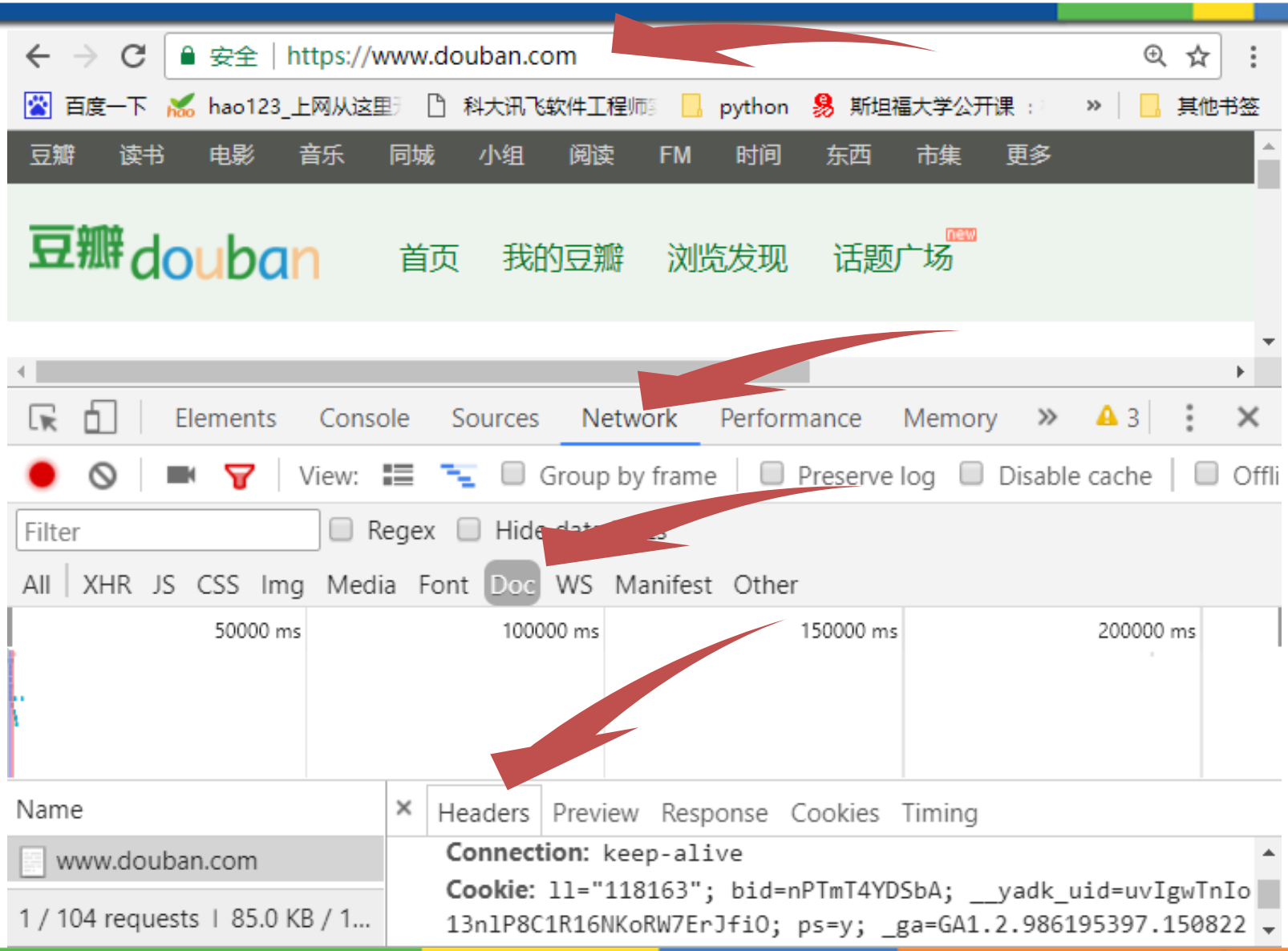
cookie的概念

- Cookie 是由 Web 服务器保存在用户浏览器（客户端）上的小文本文件，它可以包含有关用户的信息。
- 无论何时用户链接到服务器，Web 站点都可以访问 Cookie 信息



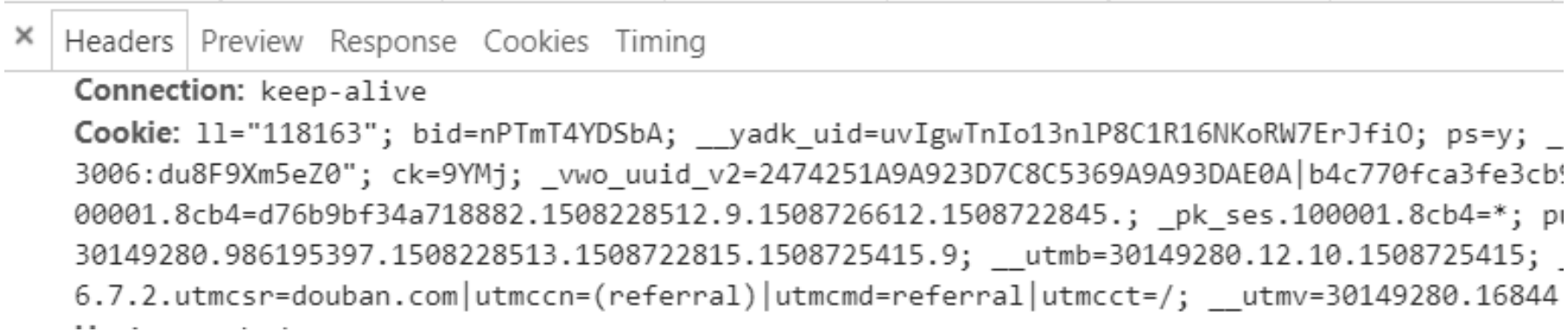
获取cookie步骤

获取cookie


安全 | https://www.douban.com
豆瓣 读书 电影 音乐 同城 小组 阅读 FM 时间 东西 市集 更多
douban 首页 我的豆瓣 浏览发现 话题广场
Elements Console Sources Network Performance Memory
Filter Regex
All XHR JS CSS Img Media Font Doc WS Manifest Other
50000 ms 100000 ms 150000 ms 200000 ms
Name
www.douban.com
1 / 104 requests | 85.0 KB / 1...
Headers Preview Response Cookies Timing
Connection: keep-alive
Cookie: ll="118163"; bid=nPTmT4YDSbA; __yadk_uid=uvIgwTnIo
13nlP8C1R16NKoRW7ErJfiO; ps=y; _ga=GA1.2.986195397.150822

转换cookie格式

将cookie字符串转换为字典格式



爬取文件

带着cookie向网站服务器发请求，表明我们是一个已登录的用户

```
def start_requests(self): #初始请求(request)
    # 带着cookie向网站服务器发请求，表明我们是一个已登录的用户
    yield Request(self.start_urls[0], callback=self.parse, cookies=self.myCookie,
                  headers=self.headers, meta=self.meta)
```

练习

1、从知乎中获取各种电影话题的信息，其网址如下：

https://www.zhihu.com/#signin。

2、获取微薄中个人首页的各种微薄信息。

Q&A

Thanks!