

第

10

章

# 数据获取（爬虫）

## 第1节 HTML简介

## 第2节 json和Xpath简介

## 第3节 Scrapy库的介绍

## 第4节 静态页面的数据获取

## 第5节 动态页面的数据获取



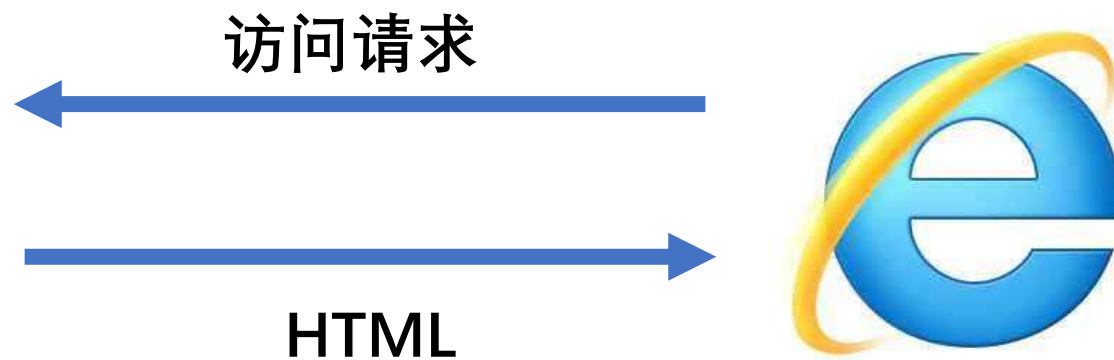
- HTML（HyperText Markup Language）称为超文本标记语言，是一种标识性的语言。它包括一系列标签，通过这些标签来标记要显示的网页中的各个部分。
- 网页文件本身是一种文本文件，通过在文本文件中添加标记符，可以告诉浏览器如何显示其中的内容（如：文字如何处理，画面如何安排，图片如何显示等）。



## 互联网网站

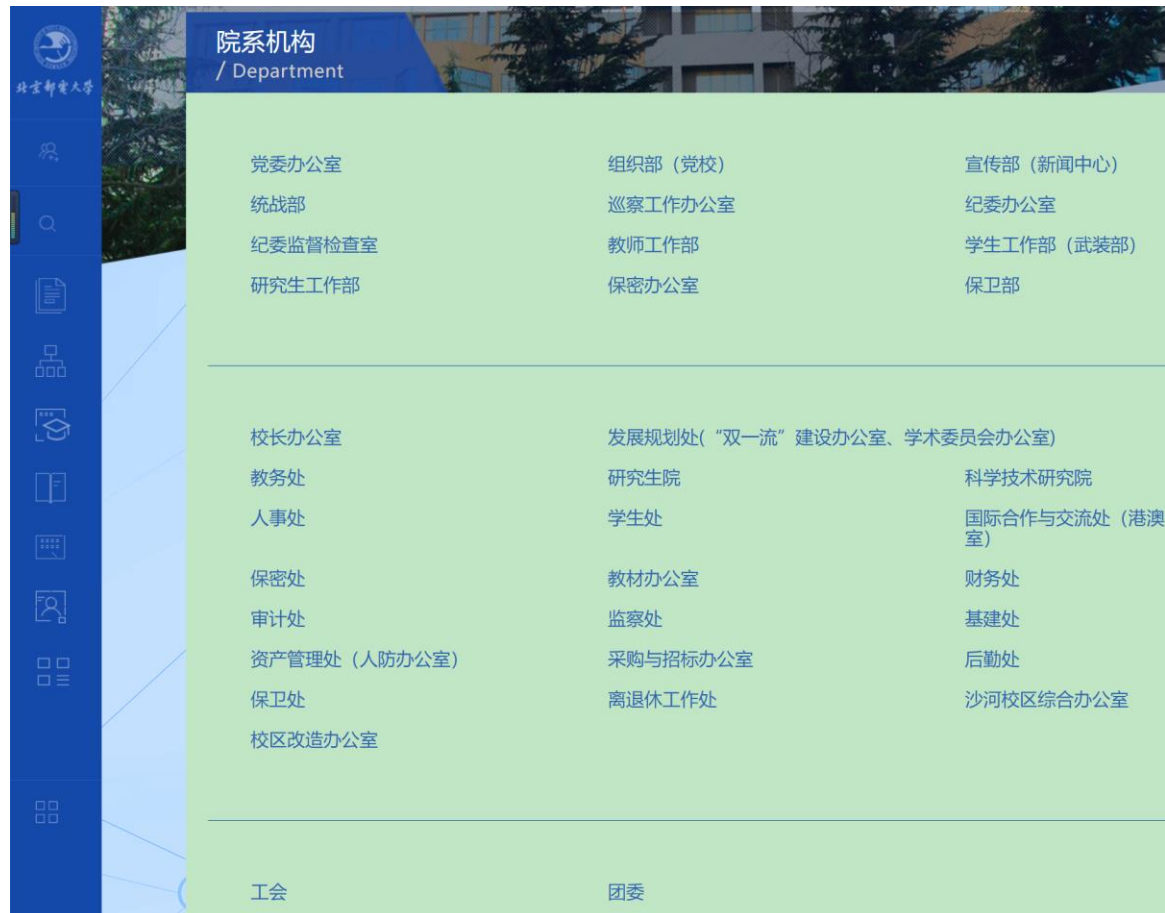


## 浏览器



# 1.HTML简介

## 静态网页

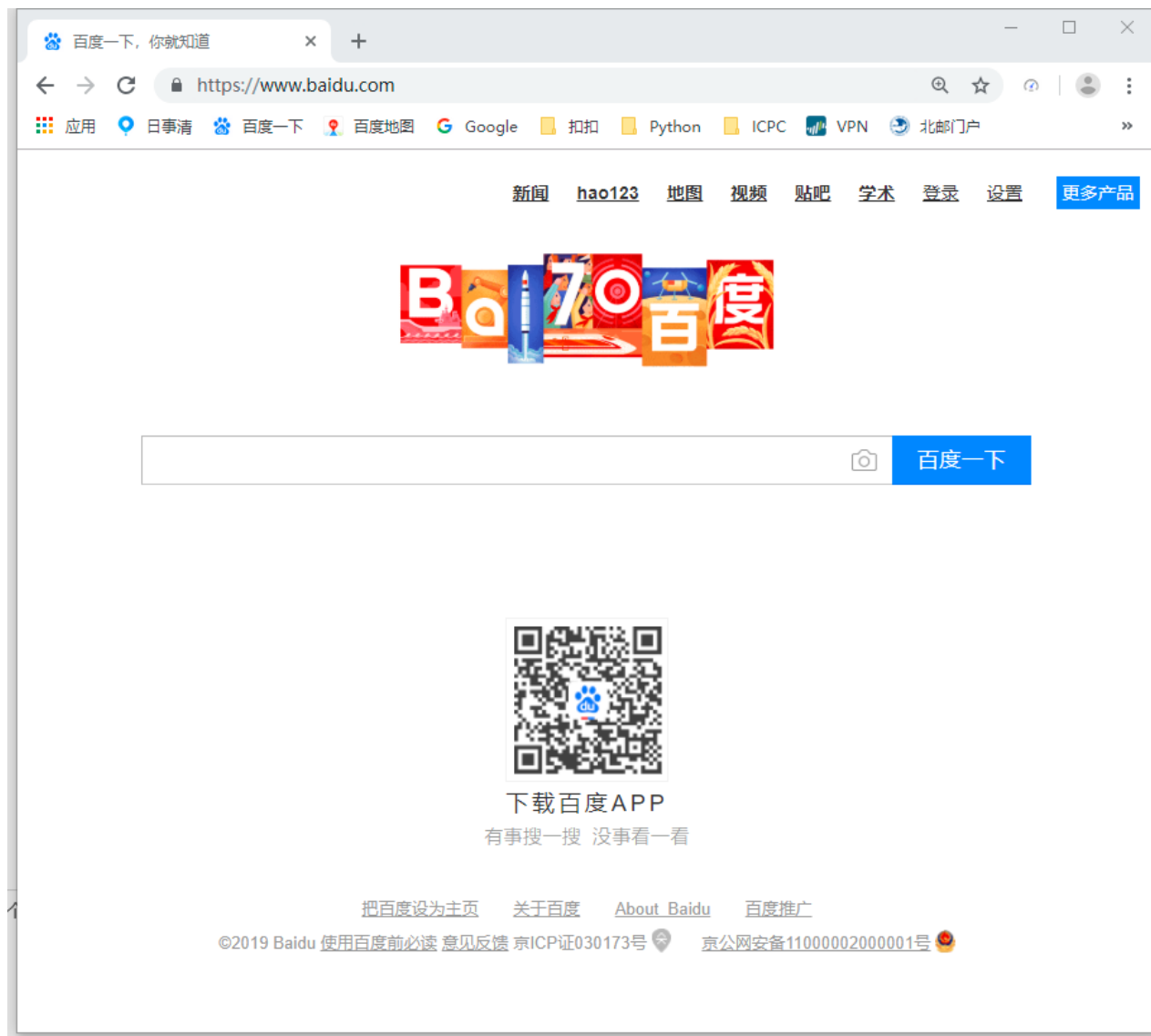


## 动态网页



# 1.HTML简介

## 百度首页



```
<html>
```

```
<head>
```

```
    <title> 标题 </title> （浏览器标题）
```

```
</head>
```

```
<body>
```

```
    内容主体 （网页具体内容）
```

```
</body>
```

```
</html>
```



## 常用标签

`<h1>`

`<p>`

`<ul>`

`<li>`

`<a href>`

`<table> <tr> <td>`





# DIV+CSS



# 1.HTML简介

搜 狐  
SOHU.com

70 国庆  
1949-2019

大家都在搜：建立个人破产制度

Q

搜狐号 | 搜狗搜索 | 搜狐邮箱

新闻 军事 专题 体育 NBA 无人机 娱乐 视频 电视剧 时尚 旅游 母婴 美食 文化 历史 邮箱 浏览器 博客 手机 微门户 公益

财经 宏观 理财 房产 二手房 家居 汽车 买车 新能源 科技 教育 健康 星座 动漫 游戏 地图 输入法 彩票 畅游 17173 政务



我和我的儿女们  
HEART-TO-HEART  
DIALOGUE

聚焦再生家庭痛点 探讨父母子女相处新模式  
正在热播  
tv.sohu.com

网络监督专区  
欢迎监督 如实举报  
学习强国  
XUEXI.QIANGUO.CN



阅兵集训揭秘：整齐划一的队形怎么练



85国佳丽角逐“地球小姐”选美大赛



美媒：波音737MAX遗留“重要保护机制”



美75岁古董B17战机坠毁 现场黑烟滚滚



为保一方平安 民警的国庆一天这样度过



视频

全景CG大片：史诗70年

习近平同比利时国王菲利普就旅比大熊猫诞下幼崽互致贺电

奋斗的史诗是怎样炼成的 | 理上网来

《在庆祝中华人民共和国成立70周年大会上的讲话》出版 | 家国70

这，就是新中国 | 这，就是中国共产党

国庆假期第二天|公路运行平稳 铁路继续增加运力

一图读懂70年减贫成绩单 | 回眸70年积蓄再出发的磅礴伟力

较强冷空气影响北方 多地气温将迎断崖式下降

国庆大会7万气球会带来污染？你担忧的阿中哥都想到了

广州楼市打响促销战，二手房也跌了

贵州铜仁地震已造成4000多户人家房屋受损

国庆西湖断桥变“人桥”，游客：把wifi信号都挤断了

被份子钱掏空的年轻人：7天长假8个婚礼，我太“南”了！

英公布“终极”脱欧方案！首相：谈不拢就硬脱欧

朝中社：朝鲜成功试射最新潜射导弹“北极星”-3

WTO史上最大罚单：允许美向75亿美元欧盟产品加税

韩国华城系列杀人案嫌犯供认杀人强奸40多起犯罪事实

普京谈16岁瑞典环保少女：她是被成人利用的无知青年

惊悚！美国一老式战斗机坠毁致多人死亡

两网民侮辱阅兵式官兵被拘：“打起来才好看”

家政需求持续上涨，月嫂平均月薪达9795元

北京城事 | 搜狐精选 换一换



急诊PCI学术周2019 | 十大心脏病“高危”职业：明星、医生上榜

央视新闻独家采访张艺谋：揭秘巨幅国旗是如何升起的

独立优雅的“大女人”怎么穿？Akris秀场上的马伊琍给你答案

山东游客突发疾病晕倒 特警及时救助脱险

急诊PCI学术周2019 | 心梗救治就是与时间赛跑！专家带您“亲临”手术现场

急诊PCI学术周2019 | 突发心梗，除了拨打120，还需牢记这四点

☐ 经验20倍

☐ 金钱10倍

☒ 金钱20倍

☐ 爆率2倍

☐ 爆率5倍

☒ 爆率10倍

保存数值



经验20倍  
金钱10倍  
金钱20倍  
爆率2倍  
爆率5倍  
爆率10倍  
传送戒指  
治疗戒指  
元宝

用户  
反馈



# DIV+CSS示例代码



第1节 HTML简介

第2节 JSON和XPath简介

第3节 Scrapy库的介绍

第4节 静态页面的数据获取

第5节 动态页面的数据获取



JSON(JavaScript Object Notation, JS 对象表示法), 是一种轻量级的数据交换格式。它基于 ECMAScript (欧洲计算机协会制定的JS规范)的一个子集, 采用完全独立于编程语言的文本格式来存储和表示数据。简洁和清晰的层次结构使得 JSON 成为理想的数据交换语言。易于人阅读和编写, 同时也易于机器解析和生成, 并有效地提升网络传输效率。



- 数据使用键/值（key-value）对表示。
- 使用大括号保存对象，每个名称后面跟着一个 ‘:’（冒号），多个键/值对之间使用，（逗号） 隔开。

```
{  
    "id":1,  
    "language": "Python",  
    "edition": "third",  
    "price": 35.50  
}
```



### 数组结构

```
{  
  "books": [  
    {  
      "id":1,  
      "language": "Python",  
      "edition": "third",  
      "price": 35.50  
    },  
    {  
      "id":2,  
      "language": "C++",  
      "edition": "second",  
      "price": 29.80  
    }  
  ]  
}
```



## 2.JSON

```
<html>
<body>
<h2>JSON Object Creation in JavaScript</h2>

<p>
ID: <span id="j_id"></span><br>
Language: <span id="j_language"></span><br>
Edition: <span id="j_edition"></span><br>
Price: <span id="j_price"></span><br>
</p>

<script>
var JSONObject = {
    "id":1,
    "language": "Python",
    "edition": "third",
    "price": 35.50};
```

```
document.getElementById("j_id").innerHTML=JSONObject.id

document.getElementById("j_language").innerHTML=JSONObject.language

document.getElementById("j_edition").innerHTML=JSONObject.edition

document.getElementById("j_price").innerHTML=JSONObject.price

</script>

</body>
</html>
```





### JSON

```
"car" : {  
  "company": Volkswagen,  
  "brand": " Jetta ",  
  "price": 200000  
}
```

### XML

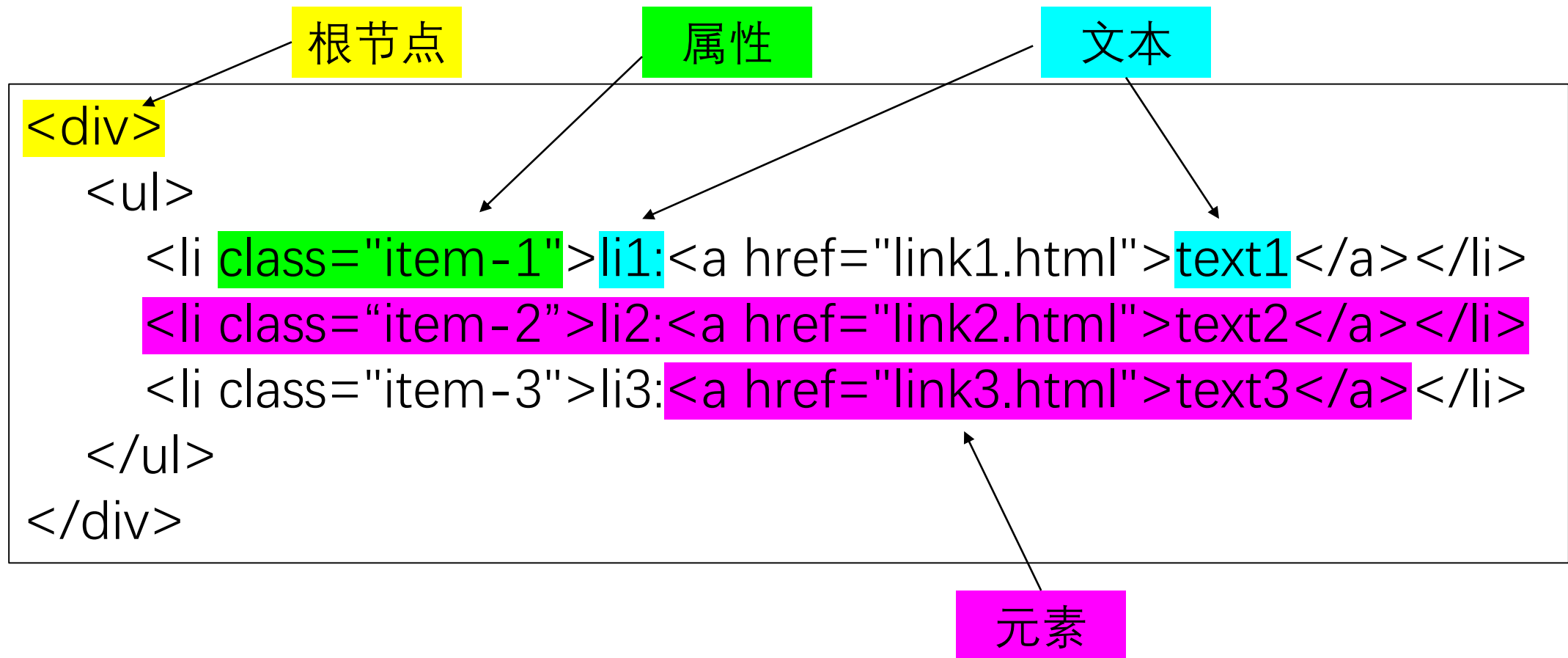
```
<car>  
  <company>Volkswagen</company>  
  <brand>Jetta</brand>  
  <price>200000</price>  
</car>
```



XPath 是 W3C (World Wide Web Consortium) 的一种标准, 是一门在 XML 文档 (包含HTML文档) 中查找信息的语言。可以让我们以路径的形式访问 html 网页中的各个元素。

使用Xpath可以帮助我们很方便地定位复杂网页中的各个元素, 使得我们有针对性的去获取网页中的特定信息。





表达式	描述
nodename	选取当前节点的所有子节点中, 节点名称为nodename的节点
/	从根节点选取
//	选择所有节点
.	选取当前节点。
..	选取当前节点的父节点。
@	选取属性。



## 2.XPath

```
<div class="class-0" id="id0"> div0
  <div class="class-1">div1
    <ul>
      <li class="item-1">li1:<a href="link1.html">text1</a></li>
      <li class="item-2">li2:<a href="link2.html">text2</a></li>
      <li class="item-3">li3:<a href="link3.html">text3</a></li>
    </ul>
  </div>
  <div class="class-2">div2
    <ul>
      <li class="item-1">li4:<a href="link4.html">text4</a></li>
      <li class="item-2">li5:<a href="link5.html">text5</a></li>
      <li class="item-3">li6:<a href="link6.html">text6</a></li>
    </ul>
  </div>
</div>
```

xpath("/div")

xpath("//div")

xpath("div")



## 2.XPath

```
<div class="class-0" id="id0"> div0
  <div class="class-1">div1
    <ul>ul1
      <li class="item-1">li1:<a href="link1.html">text1</a></li>
      <li class="item-2">li2:<a href="link2.html">text2</a></li>
      <li class="item-3">li3:<a href="link3.html">text3</a></li>
    </ul>
  </div>
  <div class="class-2">div2
    <ul>ul2
      <li class="item-1">li4:<a href="link4.html">text4</a></li>
      <li class="item-2">li5:<a href="link5.html">text5</a></li>
      <li class="item-3">li6:<a href="link6.html">text6</a></li>
    </ul>
  </div>
</div>
```

xpath("//@class")



在标签后面还可以加上谓词，用于更为精准的选择。  
例如div[1], li[last()], li[price>=80]等

谓词	描述
[1]	选取第一个子节点
[last()]	选取最后一个子节点
[position()<6]	选取前5个子节点
[@class]	选取有属性名为class的子节点
[@class='0']	选取有属性名位class，且值为0的子节点
[price>80]	选取元素为price而且大于80的子节点



## 2.XPath

```
<div class="class-0" id="id0"> div0
```

```
<div class="class-1">div1
```

```
<ul>
```

```
<li class="item-1">li1:<a href="link1.html">text1</a></li>
```

```
<li class="item-2">li2:<a href="link2.html">text2</a></li>
```

```
<li class="item-3">li3:<a href="link3.html">text3</a></li>
```

```
</ul>
```

```
</div>
```

```
<div class="class-2">div2
```

```
<ul>
```

```
<li class="item-1">li4:<a href="link4.html">text4</a></li>
```

```
<li class="item-2">li5:<a href="link5.html">text5</a></li>
```

```
<li class="item-3">li6:<a href="link6.html">text6</a></li>
```

```
</ul>
```

```
</div>
```

```
</div>
```

xpath('div[1]')

xpath('//div[1]')





## 2.XPath

```
<div class="class-0" id="0"> div0
```

```
  <div class="class-1">div1
```

```
    <ul>
```

```
      <li class="item-1">li1:<a href="link1.html">text1</a></li>
```

```
      <li class="item-2">li2:<a href="link2.html">text2</a></li>
```

```
      <li class="item-3">li3:<a href="link3.html">text3</a></li>
```

```
    </ul>
```

```
  </div>
```

xpath('//li[1]')

```
<div class="class-2">div2
```

```
  <ul>
```

```
    <li class="item-1">li4:<a href="link4.html">text4</a></li>
```

```
    <li class="item-2">li5:<a href="link5.html">text5</a></li>
```

```
    <li class="item-3">li6:<a href="link6.html">text6</a></li>
```

```
  </ul>
```

```
</div>
```

xpath('//a[1]')

```
</div>
```



## 2.XPath

```
<div class="class-0" id="0"> div0
```

```
  <div class="class-1">div1
```

```
    <ul>
```

xpath('//li[1]')

```
      <li class="item-1">li1:<a href="link1.html">text1</a></li>
```

```
      <li class="item-2">li2:<a href="link2.html">text2</a></li>
```

```
      <li class="item-3">li3:<a href="link3.html">text3</a></li>
```

```
    </ul>
```

```
  </div>
```

xpath('//li[last()]')

```
<div class="class-2">div2
```

```
  <ul>
```

```
    <li class="item-1">li4:<a href="link4.html">text4</a></li>
```

```
    <li class="item-2">li5:<a href="link5.html">text5</a></li>
```

xpath('//a[1]')

```
    <li class="item-3">li6:<a href="link6.html">text6</a></li>
```

```
  </ul>
```

```
</div>
```

```
</div>
```



通配符	描述
//*	选取文档中的所有节点
/*	选取当前节点下的所有子节点
[@*]	选取带有属性的任何节点



使用Chrome浏览器，打开网页后，右键->检查->copy xpath，在网页中精准找到要定位的信息。



第1节 HTML简介

第2节 json和Xpath简介

第3节 Scrapy库的介绍

第4节 静态页面的数据获取

第5节 动态页面的数据获取

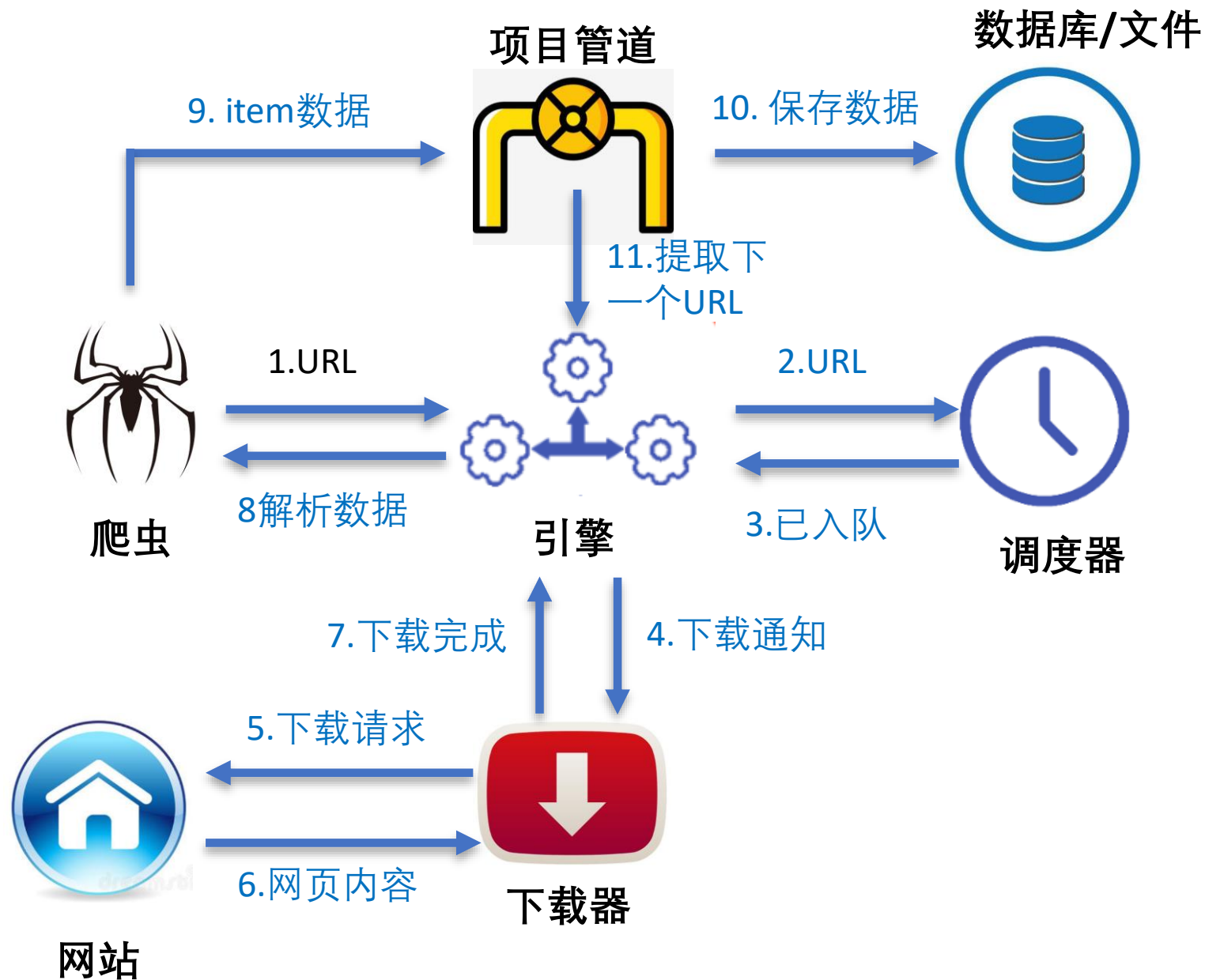


### 3. scrapy

- 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 请求，还有强大的 selector 能够方便地解析 response 响应。
- 可以使用这个工具将爬虫程序工程化、模块化。
- scrapy 是一个基于 Twisted 的异步处理框架，是纯 python 实现的爬虫框架，其架构清晰，模块之间的耦合程度低，可扩展性也很强。



### 3. scrapy的工作过程



### 3. scrapy的主要模块

模块	说明
Scrapy Engine引擎	整个框架的核心，负责控制数据流在系统中的各个组件之间的传送，并在相应动作发生时触发事件
Scheduler调度器	接受引擎发过来的请求，并将其加入url队列当中，并默认完成去掉重复的url的工作
Downloader 下载器	负责下载 Engine 发送的所有 Requests 请求，并将其获取到的 responses 回传给 Scrapy Engine
Spider 爬虫	负责解析response，从中提取数据赋给Item的各个字段。并将需要继续进一步处理的url提交给引擎，再次进入Scheduler(调度器)
Item Pipeline 项目管道	处理Spider中获取到的Item，并进行后期的处理。例如清理HTML 数据、验证爬取的数据（检查 item 包含某些字段）、查重（并丢弃）、爬取数据的持久化（写入文件或者存入数据库等）
Downloader Middlewares 下载中间件	是 Engine 和 Downloader 的枢纽。负责处理 Downloader 传递给 Engine 的 responses；它还支持自定义扩展。
Spider Middlewares 爬虫中间件	可以自定义扩展和操作引擎和Spider中间通信的功能组件（比如进入Spider的Responses;和从Spider出去的Requests）负责下载 Engine 发送的所有 Requests 请求，并将其获取到的 responses 回传给 Scrapy Engine





### 3. scrapy的工作过程

- 1、spider爬虫将初始url请求发给引擎;
- 2、引擎将初始请求发给调度器, 调度器将该url放入队列;
- 3、调度器回复引擎url已经入队;
- 4、通知下载器进行下载;
- 5、下载器向目标网站发出下载请求;
- 6、获得网页内容;
- 7、下载器通知引擎已经下载完成;
- 8、引擎将response发给spider, spider解析数据、提取item;
- 9、spider将获取到的数据给到引擎, 并通知引擎把新的url给到调度器进入队列, 同时把item数据发送给Item Pipelines进行保存;
- 10、Item Pipelines将提取到的数据加工并保存到数据库或者文件中
- 11、保存完毕后通知引擎进行下一个url的提取;
- 12、循环1-11步, 直到调度器中没有新的url, 结束整个过程。



### 3. scrapy的安装

- scrapy依赖的模块较多，例如wheel、lxml、Twisted、pywin32等，只有这些都成功安装之后，才能安装scrapy。可以使用pip工具依次安装，但这样比较麻烦，推荐使用pycharm来安装更加简单方便。



### 3. scrapy的安装

- 安装好pycharm后，新建一个工程（例如工程名称为myscrapy），打开菜单File->Setting->Project: spider>Project Interpreter，找到右上角的加号符号，选择添加scrapy。安装过程中可能会出现提示，提示缺少Microsoft Visual C++ 14.0。这时可以使用以下网址安装：  
<http://go.microsoft.com/fwlink/?LinkId=691126>。
- 安装好Microsoft Visual Build Tools之后，便可以成功安装scrapy了。



第1节 HTML简介

第2节 json和Xpath简介

第3节 Scrapy库的介绍

第4节 静态页面的数据获取

第5节 动态页面的数据获取



### 3. 使用scrapy的步骤

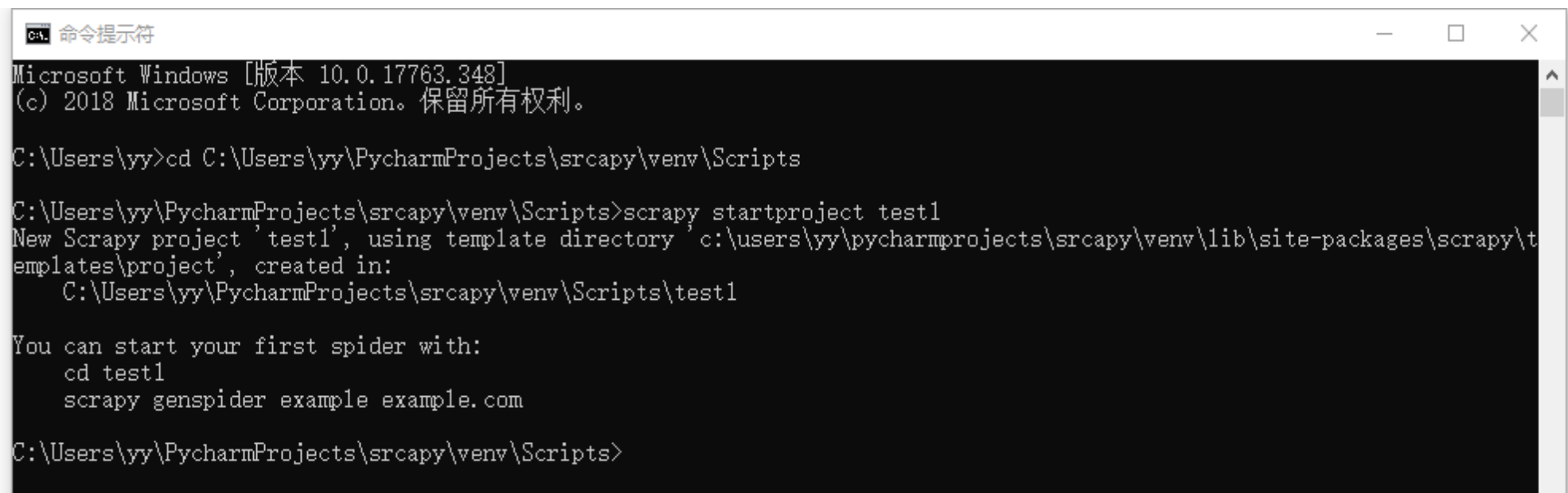
1. 新建项目 (`scrapy startproject xxx`): 新建一个新的爬虫项目
2. 确定目标 (编写`items.py`) : 明确你想要抓取的目标
3. 制作爬虫 (`spiders/xxspider.py`) : 制作爬虫开始爬取网页
4. 存储内容 (`pipelines.py`) : 设计管道存储爬取内容



### step1

创建一个Scrapy项目:

先找到安装scrapy的目录: 打开cmd命令行, 先用cd命令转到该目录下的venv\scripts\, 再键入命令: scrapy startproject test1, 即可创建一个新的项目。



```
命令提示符
Microsoft Windows [版本 10.0.17763.348]
(c) 2018 Microsoft Corporation. 保留所有权利。

C:\Users\yy>cd C:\Users\yy\PycharmProjects\srcapy\venv\Scripts

C:\Users\yy\PycharmProjects\srcapy\venv\Scripts>scrapy startproject test1
New Scrapy project 'test1', using template directory 'c:\users\yy\pycharmprojects\srcapy\venv\lib\site-packages\scrapy\templates\project', created in:
  C:\Users\yy\PycharmProjects\srcapy\venv\Scripts\test1

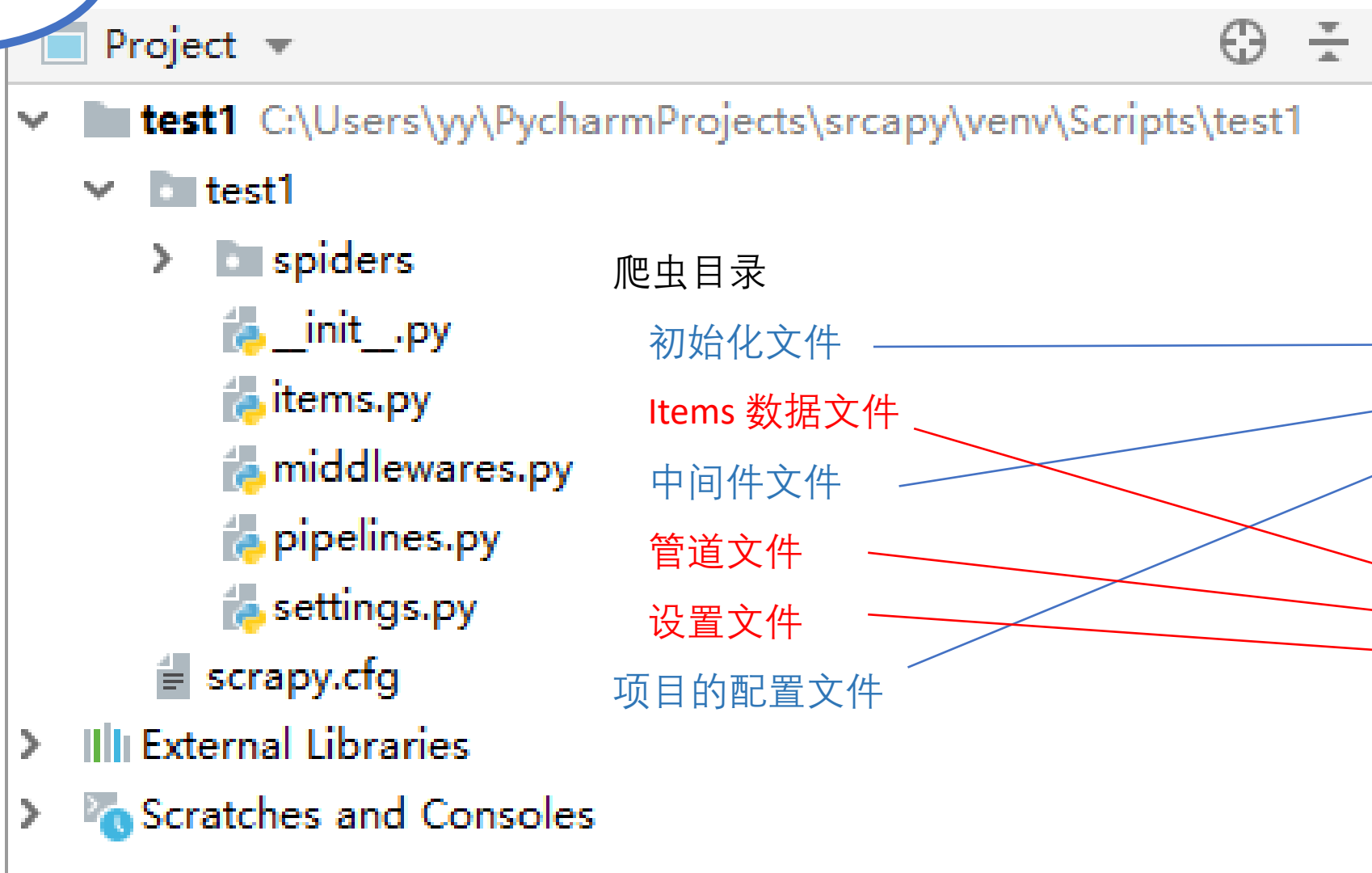
You can start your first spider with:
  cd test1
  scrapy genspider example example.com

C:\Users\yy\PycharmProjects\srcapy\venv\Scripts>
```



### 3. 使用scrapy的步骤

#### step1 使用pycharm打开工程test1



爬虫目录

初始化文件

Items 数据文件

中间件文件

管道文件

设置文件

项目的配置文件

不需要修改

需要修改



### 3. 使用scrapy的步骤

#### step1

在test1工程之下，新建一个begin.py文件(与scrapy.cfg在同一级目录下)，内容如下：

```
from scrapy import cmdline  
  
cmdline.execute("scrapy crawl bupt".split())  
  
#bupt为爬虫的名字，在spider.py中定义
```





### 3. 使用scrapy的步骤

step2

修改items.py文件:

```
import scrapy  
  
class MyItem(scrapy.Item):  
    # define the fields for your item here like:  
  
    school = scrapy.Field()  
  
    link = scrapy.Field()
```



#### step3

新建一个spider.py文件(在spider目录下)

```
import scrapy

from test1.items import MyItem #从items.py中引入MyItem对象

class mySpider(scrapy.spiders.Spider):

    name = "bupt" #爬虫的名字是bupt

    allowed_domains = ["bupt.edu.cn/"] #允许爬取的网站域名

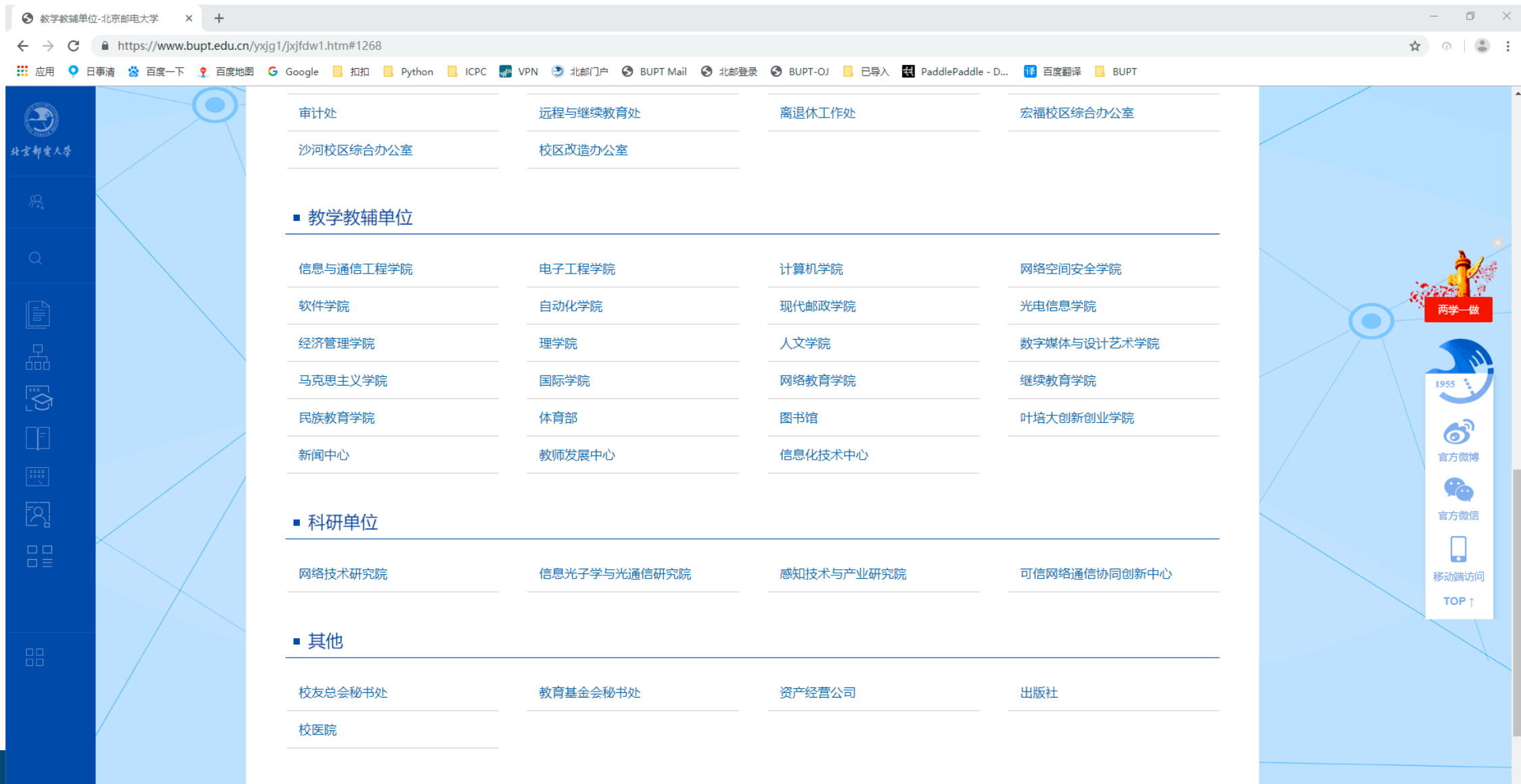
    start_urls = ["https://www.bupt.edu.cn/yxjg1.htm"]

    #初始URL, 即爬虫爬取的第一个URL
```



# 4.静态网页数据抓取

<https://www.bupt.edu.cn/yxjg1.htm>



## step3

办公室

远程与继续教育处

离退休工作处

校区改造办公室

 $1105.36 \times 378$ 

■ #000000

Font 20px "Microsoft YaHei"

Margin 0px 0px 0px -32.1875px

信息与通信工程学院

电子工程学院

计算机学院

软件学院

自动化学院

现代邮政学院

经济管理学院

理学院

人文学院

马克思主义学院

网络教育学院

民族教育学院

教育部

图书馆

新闻中心

教师发展中心

信息化技术中心

## ■ 科研单位

网络技术研究院

信息光子学与光通信研究院

感知技术与产业研究院

## 快捷入口

[网站地图](#)

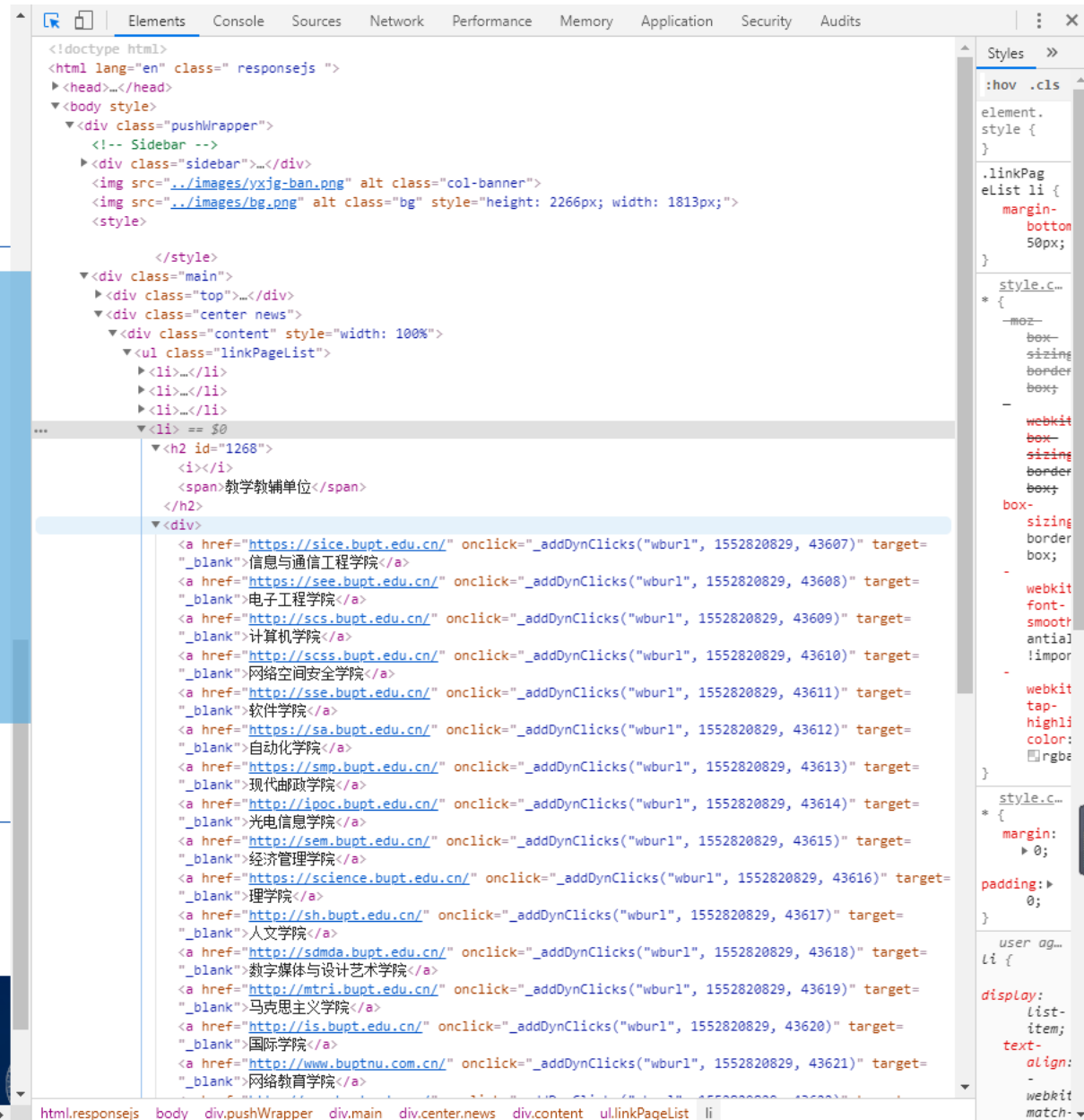
## 风光人文

## 校内通讯录

北邮校区

西土城路校区

贵州省 长顺县



### step3

在spider.py文件中的class mySpider中，添加parse函数

```
def parse(self, response): #解析爬取的内容

    item = MyItem() #生成一个在items.py中定义好的Myitem对象,用于接收爬取的数据

    for each in response.xpath("/html/body/div/div[2]/div[2]/div/ul/li[4]/div/*"):
        #用xpath来解析html， div标签中的数据，就是我们需要的数据。

        item['school'] = each.xpath("text()").extract() #学院名称在text中
        item['link'] = each.xpath("@href").extract() #学院链接在href中

        if(item['school'] and item['link']): #去掉值为空的数据

            yield(item) #返回item数据给到pipelines模块
```

### step4

### 修改pipelines.py

```
import json

class MyPipeline(object):
    def open_spider(self, spider):
        try: #打开json文件
            self.file = open('MyData.json', "w", encoding="utf-8")
        except Exception as err:
            print(err)

    def process_item(self, item, spider):
        dict_item = dict(item) #生成字典对象
        json_str = json.dumps(dict_item, ensure_ascii=False) + "\n" #生成json串
        self.file.write(json_str) #将json串写入到文件中
        return item

    def close_spider(self, spider):
        self.file.close() #关闭文件
```



### step4

修改setting.py

添加ITEM\_PIPELINES = {'test1.pipelines.MyPipeline': 300,}

修改ROBOTSTXT\_OBEY = False

- 参数是分配给每个类的整型值，确定了它们运行的顺序，item按数字从低到高的顺序，通过pipeline。
- 通常将这些数字定义在0-1000范围内。



## 4.静态网页数据抓取

step4

运行begin.py

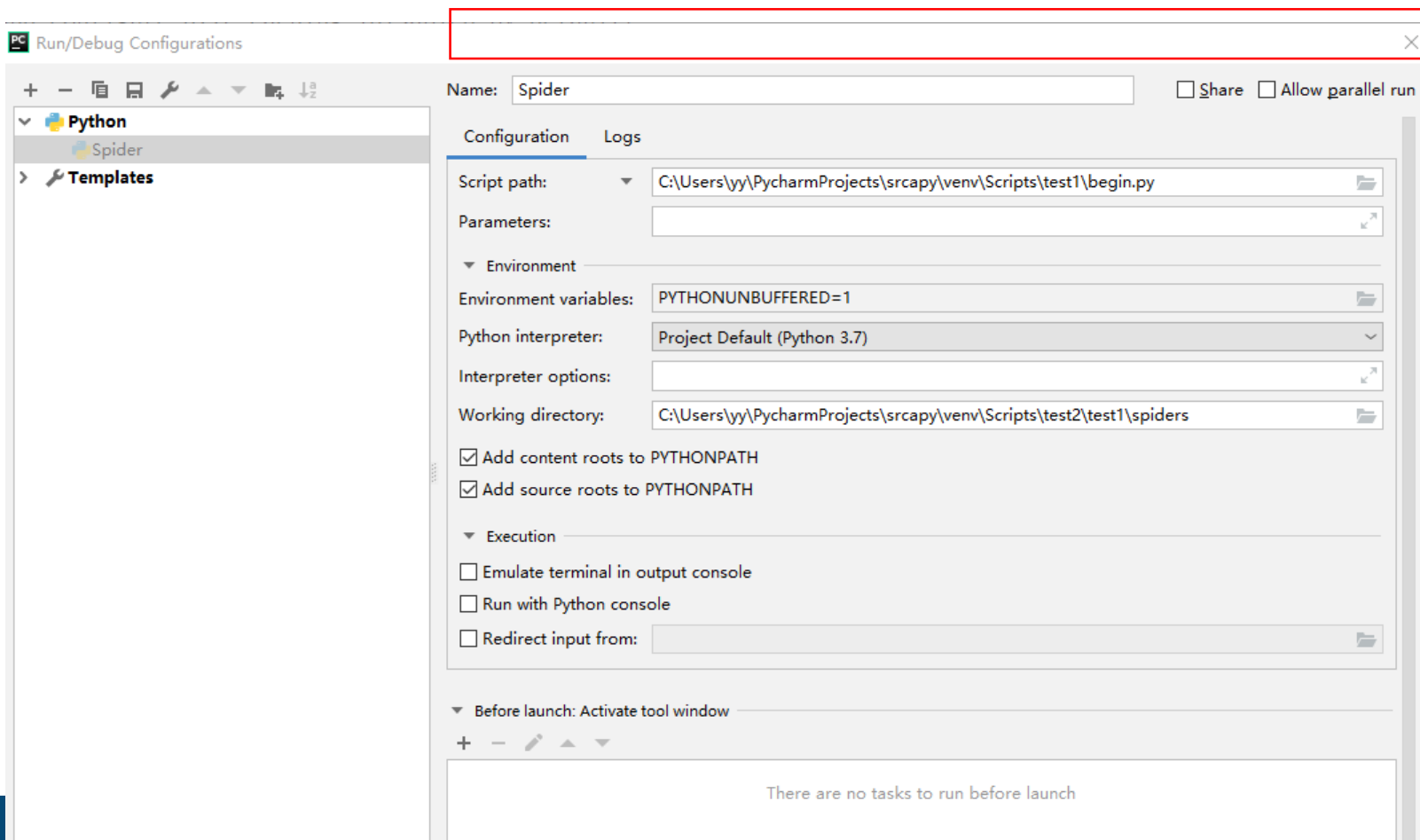




## 4.静态网页数据抓取

step4

或者运行spider.py，并将其运行时的Script path配置项修改为begin.py



### step4

结果

```
{ "school": ["信息与通信工程学院"], "link": ["https://sice.bupt.edu.cn/"] }
{ "school": ["电子工程学院"], "link": ["https://see.bupt.edu.cn/"] }
{ "school": ["计算机学院"], "link": ["http://scs.bupt.edu.cn/"] }
{ "school": ["网络空间安全学院"], "link": ["http://scss.bupt.edu.cn/"] }
{ "school": ["软件学院"], "link": ["http://sse.bupt.edu.cn/"] }
{ "school": ["自动化学院"], "link": ["https://sa.bupt.edu.cn/"] }
{ "school": ["现代邮政学院"], "link": ["https://smp.bupt.edu.cn/"] }
{ "school": ["光电信息学院"], "link": ["http://ipoc.bupt.edu.cn/"] }
{ "school": ["经济管理学院"], "link": ["http://sem.bupt.edu.cn/"] }
{ "school": ["理学院"], "link": ["https://science.bupt.edu.cn/"] }
{ "school": ["人文学院"], "link": ["http://sh.bupt.edu.cn/"] }
{ "school": ["数字媒体与设计艺术学院"], "link": ["http://sdmda.bupt.edu.cn/"] }
{ "school": ["马克思主义学院"], "link": ["http://mtri.bupt.edu.cn/"] }
{ "school": ["国际学院"], "link": ["http://is.bupt.edu.cn/"] }
{ "school": ["网络教育学院"], "link": ["http://www.buptnu.com.cn/"] }
{ "school": ["继续教育学院"], "link": ["http://sce.bupt.edu.cn/"] }
{ "school": ["民族教育学院"], "link": ["http://seme.bupt.edu.cn/"] }
{ "school": ["体育部"], "link": ["http://ped.bupt.edu.cn/"] }
{ "school": ["图书馆"], "link": ["https://lib.bupt.edu.cn/index.html"] }
```



第1节 HTML简介

第2节 json和Xpath简介

第3节 Scrapy库的介绍

第4节 静态页面的数据获取

第5节 动态页面的数据获取



## 5.动态页面的数据获取

1. 新建项目 (`scrapy startproject xxx`): 新建一个新的爬虫项目
2. 确定目标 (编写`items.py`): 明确你想要抓取的目标
3. 制作爬虫 (`spiders/xxspider.py`): 制作爬虫开始爬取网页
4. 存储内容 (`pipelines.py`): 设计管道存储爬取内容



## 5.动态页面的数据获取

- <https://bj.lianjia.com/ershoufang>

默认排序


最新发布

房屋总价

房屋单价

房屋面积

共找到 92969 套北京二手房



试试地图找房



此房为全南向正规一居室 楼层高 采光充... 必看好房

朝丰家园 - 豆各庄

1室1厅 | 49.96平米 | 南 | 简装 | 高楼层(共16层) | 2009年建 | 板楼


87人关注 / 1个月以前发布

VR房源

房本满五年

253万

单价50641元/平米



满五年商品房，南北通透，格局方正，看... 必看好房

悦秀园 - 西三旗

2室1厅 | 73.73平米 | 南北 | 简装 | 低楼层(共6层) | 2000年建 | 板楼

178人关注 / 3个月以前发布

VR房源

房本满五年

468万

单价63475元/平米



西直门 今典花园 花园洋房 两室两厅 满... 必看好房

今典花园 - 小西天

2室2厅 | 90.39平米 | 西南 | 精装 | 中楼层(共25层) | 2002年建 | 塔楼

120人关注 / 13天以前发布

VR房源

房本满五年

随时看房

709万

单价78438元/平米

热门问答

提交了购房资质审核以后，我的资质是否长期有效呢？  
23个回答 / 2016-05-27

买二手房契税按几个点交？ 21个回答 / 2017-05-12

女朋友是京籍，我不是，婚后买房能写我的名字吗？  
17个回答 / 2016-08-11

北京新购房政策中家庭怎么定义？  
17个回答 / 2017-05-12

我在购房资格审核后怀孕了，需要重新做审核吗？  
12个回答 / 2016-06-12

热门百科

更多

满五不唯一的二手住宅需交... ▾

全款买二手房，需要走哪些... ▾

在北京贷款买房有哪些方式... ▾



# 作业

## 1.爬取学堂在线的计算机类课程页面内容

<https://www.xuetangx.com/search?query=&org=&classify=1&type=&status=&page=1>

要求将课程名称、老师、所属学校和选课人数信息，保存到一个csv文件中。

## 2.爬取链家官网二手房的数据

<https://bj.lianjia.com/ershoufang/>

要求爬取北京市东城、西城、海淀和朝阳四个城区的数据（每个区爬取5页），将楼盘名称、总价、平米数、单价保存到json文件中。

以上作业以报告形式提交，需要将核心代码贴在报告中，并在报告中给出最终的csv和json文件内容（截取前50条数据即可）。文件名为学号，文件格式为pdf，提交平台为爱课堂。

