

# Python 爬虫作业

班级：2018211313 班

学号：2018211366

姓名：蒋潇逸

版本：1.0

更新：November 17, 2020

本文档是 Python 爬虫作业报告，具体代码详见附录。

## 目录

1	作业目的	3
2	作业要求及内容	3
3	实验设备环境	3
4	爬取学堂在线	3
4.1	确定 items 类中元素	3
4.2	分析出数据接口	4
4.3	编写 spider.py	4
4.4	编写 pipelines.py	6
4.5	运行结果 (截取前 50 条)	7
5	爬取链家官网二手房	9
5.1	确定 items 类中元素	9
5.2	获取 xpath	9

5.3	编写 spider.py . . . . .	10
5.4	编写 pipelines.py . . . . .	12
5.5	运行结果 (截取前 50 条) . . . . .	12
6	作业总结	15

# 1 作业目的

本作业要求学生完成两个爬虫练习，进一步加深学生对于爬虫的理解和认识。

# 2 作业要求及内容

- 爬取学堂在线的计算机类课程页面内容。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。
- 爬取链家官网二手房的数据。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。

# 3 实验设备环境

Windows 10 专业版      PyCharm 2019.3.3 x64      编程语言 Python

# 4 爬取学堂在线

由于学堂在线是动态网页，如果使用和静态网页一样的方法爬取信息，返回将会是一个空值，这是由于动态网页的信息是在页面加载后，由下面的 `javascript` 脚本向服务器获取信息，然后再转化为 `HTML` 语句插入到网页中的。而默认的 `scrapy` 并不支持运行 `javascript` 脚本。所以我们要找出 `javascript` 脚本使用的数据接口。绕过网页，直接获取数据。

## 4.1 确定 `items` 类中元素

由于作业要求爬取课程名称、老师、所属学校和选课人数信息，故在 `XuetangxItem` 类中声明四个元素 `name`, `teachers`, `school`, `count` 分别代表课程名称、老师、所属学校、选课人数，代码如下所示。

```
1 import scrapy
2
3
4 class XuetangxItem(scrapy.Item):
5     # define the fields for your item here like:
```

```

6     name = scrapy.Field()
7     teachers = scrapy.Field()
8     school = scrapy.Field()
9     count = scrapy.Field()

```

## 4.2 分析出数据接口

打开学堂在线网站，利用 F12 浏览器调试者工具，观察元素的加载，发现了在?page=1 中含有课程信息。

Name	Status	Type	Initiator	Size	Time	Waterfall
?page=1	200	xhr	9_edc81cc.....	60.1 kB	162 ms	
captcha-pre-verify.html	200	docu...	TCaptcha.js:1	2.7 kB	24 ms	
pingd?dm=www.xuetang...	200	text/h...	stats.js?v2.0...	239 B	55 ms	
js?id=UA-164784773-1	(pend...	script	search?quer...	0 B	Pendi...	
9_edc81cc57938d3cfc7a.js	200	script	search?quer...	(mem...	1 ms	
119_6987c4b118057b67...	200	script	search?quer...	(mem...	0 ms	
tex-cthtml.js	200	script	search?quer...	(mem...	0 ms	
0.745d5dea18d788f581ff...	200	styles...	search?quer...	(mem...	0 ms	
15967950432640.png?im...	200	png	9_edc81cc.....	(mem...	0 ms	
15967951282740.png?im...	200	png	9_edc81cc.....	(mem...	0 ms	
119.6987c4b118057b670...	200	styles...	search?quer...	(disk ...	6 ms	
15938509762199.jpg?im...	200	jpeg	9_edc81cc.....	(mem...	0 ms	
15785319191453.png?im...	200	png	9_edc81cc.....	(mem...	0 ms	
15786192391381.png?im...	200	png	9_edc81cc.....	(mem...	0 ms	
15785318522032.png?im...	200	png	9_edc81cc.....	(mem...	0 ms	
tcaptcha-frame.21565e8...	200	script	TCaptcha.js:1	(mem...	0 ms	
73_d9e463083640724b9...	200	script	manifest_ec...	(disk ...	10 ms	
15730291622149.jpg?im...	200	jpeg	9_edc81cc.....	(mem...	0 ms	
15682771241816.jpg?im...	200	jpeg	9_edc81cc.....	(mem...	0 ms	
73.d9e463083640724b9c...	200	styles...	manifest_ec...	(disk ...	2 ms	
9784c471384d49f775d46...	200	font	0.745d5de.....	(mem...	0 ms	
data:font/woff;base...	200	font	0.745d5de.....	(mem...	0 ms	
TCaptcha.js	200	script	search?quer...	(mem...	0 ms	
44 requests   63.0 kB transferred   6.7 MB resources   Finish: 1.32 s   DOMContentLoaded: 618 ms						

## 4.3 编写 spider.py

通过?page=1 中的信息编写代码如下

```

1 import scrapy
2 import json
3
4 from test2.items import XuetangxItem
5
6

```

```

7 class XuetangxSpider(scrapy.spiders.Spider):
8     name = "xuetangx"
9     allowed_domains = ["xuetangx.com/"]
10
11     url_pat = 'https://www.xuetangx.com/api/v1/lms/get_product_list/?
12         page={}'
13     data = '{"query":"","chief_org":[],"classify":["1"],"selling_type
14         ":[]","status":[],"appid":10000}'
15     # temp = json.dumps(data)
16     # print(temp)
17     headers = {
18         'authority': 'www.xuetangx.com',
19         'accept': 'application/json, text/plain, */*',
20         'django-language': 'zh',
21         'x-client': 'web',
22         'accept-language': 'zh',
23         'xtbz': 'xt',
24         'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
25             AppleWebKit/537.36 (KHTML, like Gecko) Chrome
26             /84.0.4147.125 Safari/537.36',
27         'content-type': 'application/json',
28         'origin': 'https://www.xuetangx.com',
29         'sec-fetch-site': 'same-origin',
30         'sec-fetch-mode': 'cors',
31         'sec-fetch-dest': 'empty',
32         'referer': 'https://www.xuetangx.com/search?query=&org=&
33             classify=1&type=&status=&page=1',
34         'cookie': 'provider=xuetang; django_language=zh',
35     }
36
37     # 不使用start_urls列表，因为要创建POST请求
38     def start_requests(self):
39         for page in range(1, 7):
40             yield scrapy.FormRequest(
41                 url=self.url_pat.format(page),
42                 headers=self.headers,
43                 method='POST',
44                 body=self.data,
45                 callback=self.parse

```

```

41         )
42
43     def parse(self, response):
44         j = json.loads(response.body)
45         for product in j['data']['product_list']:
46             item = XuetangxItem()
47             item['name'] = product['name']
48
49             teacher_name_list = []
50             for teacher in product['teacher']:
51                 teacher_name_list.append(teacher['name'])
52             item['teachers'] = ','.join(teacher_name_list)
53
54             item['school'] = product['org']['name']
55
56             item['count'] = product['count']
57             print(item)
58             yield item

```

## 4.4 编写 pipelines.py

```

1  import csv
2
3
4  class XuetangxPipeline:
5      def open_spider(self, spider):
6          try:
7              self.file = open('xuetangx.csv', 'w', newline='')
8              self.csv = csv.writer(self.file)
9          except Exception as err:
10             print(err)
11
12     def process_item(self, item, spider):
13         self.csv.writerow(list(item.values()))
14         return item
15
16     def close_spider(self, spider):
17         self.file.close()

```

## 4.5 运行结果 (截取前 50 条)

```
1 C++语言程序设计基础,"郑莉,李超,徐明星",清华大学,424160
2 数据结构(上),邓俊辉,清华大学,410762
3 数据结构(下),邓俊辉,清华大学,358568
4 Java程序设计,郑莉,清华大学,195235
5 操作系统,"向勇,陈渝",清华大学,193026
6 网络技术与应用,"沈鑫剡,俞海英,李兴德,许继恒,钱万正,徐海斌,魏涛,宋以胜",
  中国人民解放军陆军工程大学,175775
7 C++语言程序设计进阶,"郑莉,李超,徐明星",清华大学,117453
8 C程序设计案例教程(基础),张莉,中国农业大学,113430
9 C程序设计案例教程(进阶),张莉,中国农业大学,110251
10 数据挖掘:理论与算法,袁博,清华大学,107902
11 大学计算机教程,"张莉,马钦",中国农业大学,62256
12 Web前端攻城狮,"刘强,吴亮,赵文博",清华大学,57691
13 Office办公软件应用,"史巧硕,朱怀忠,刘洪普,李娟",河北工业大学,57525
14 汇编语言程序设计,"张悠慧,翟季冬",清华大学,55227
15 计算机应用基础,"宋承继,李莹,李龙龙",陕西工业职业技术学院,55111
16 微软亚洲研究院大数据系列讲座,"洪小文,宋睿华,谢幸,郑宇,张洪宇",
  Microsoft,52831
17 算法设计与分析,王振波,清华大学,51345
18 学做小程序——基础篇,"刘强,小程序慕课讲师",清华大学,51061
19 大学计算机——计算思维的视角,郝兴伟,山东大学,49725
20 面向对象程序设计(C++),"黄震春,徐明星",清华大学,49054
21 数据科学导论,"袁博,何隽",清华大学,48674
22 大学计算机基础,卫春芳,湖北大学,46903
23 基于Linux的C++,乔林,清华大学,45901
24 大学计算机基础,"徐红云,刘欣欣,曹晓叶",华南理工大学,45116
25 计算机网络,"袁华,杜广龙,张凌",华南理工大学,44637
26 JAVA程序设计进阶,许斌,清华大学,42072
27 Web开发技术,"陈静,王成良,祝伟华",重庆大学,38317
28 计算几何,邓俊辉,清华大学,38151
29 程序设计基础(上),"赵宏,闫晓玉,李妍,王恺,李敏",南开大学,36495
30 大数据机器学习,袁春,清华大学,35741
31 ARM微控制器与嵌入式系统,"曾鸣,薛涛,龚光华",清华大学,35457
32 R语言数据分析,艾新波,北京邮电大学,35425
33 单片机原理及应用,"杨居义,王颖丽,蒲敏,向兵",绵阳职业技术学院,34772
34 程序设计基础(下),"赵宏,李妍,闫晓玉,王恺,李敏",南开大学,32428
35 高级大数据系统,王智,清华大学,31637
```

36 Python程序设计基础,许志良,深圳信息职业技术学院,31303

37 大学计算机基础,"李敏,高裴裴,郭蕴,赵宏,刘哲理,路明晓,李妍,闫晓玉,宋丽培",南开大学,30510

38 移动快速应用开发,"唐贤传,盛鸿宇,王婷婷,汤恒,胡万福",芜湖职业技术学院,27496

39 人工智能,罗会兰,江西理工大学,26872

40 数据库技术与程序设计,"高裴裴,李敏,袁晓洁,赵宏,路明晓,康介恢,闫晓玉,李妍,郭蕴",南开大学,26780

41 大数据技术与应用,李军,清华大学,97952

42 软件工程,"刘强,刘璘",清华大学,93703

43 计算机文化基础,"李秀,姚瑞霞,安颖莲,全成斌",清华大学,83710

44 程序设计基础,"徐明星,王瑀屏,邬晓钧",清华大学,80789

45 组合数学,马昱春,清华大学,74391

46 大数据系统基础,"王建民,徐葳,陈康,陈文光",清华大学,74244

47 VC++面向对象与可视化程序设计(上): Windows编程基础,黄维通,清华大学,69319

48 VC++面向对象与可视化程序设计(下): MFC编程基础,黄维通,清华大学,67918

49 人工智能原理,王文敏,北京大学,63686

50 大数据平台核心技术,"武永卫,姚文辉,陶阳宇,冯骁,谢德军",清华大学,62945

	A	B	C	D
1	C++语言程序设计基础	郑莉,李超,徐明星	清华大学	424160
2	数据结构(上)	邓俊辉	清华大学	410762
3	数据结构(下)	邓俊辉	清华大学	358568
4	Java程序设计	郑莉	清华大学	195235
5	操作系统	向勇,陈渝	清华大学	193026
6	网络技术与应用	沈鑫,刘,俞海英,李兴德,许继恒,钱万正,徐海斌,魏清,宋以胜	中国人民解放军陆军工程大学	175775
7	C++语言程序设计进阶	郑莉,李超,徐明星	清华大学	117453
8	C程序设计案例教程(基础)	张莉	中国农业大学	113430
9	C程序设计案例教程(进阶)	张莉	中国农业大学	110251
10	数据挖掘:理论与算法	袁博	清华大学	107902
11	大学计算机教程	张莉,马钦	中国农业大学	62256
12	Web前端攻城狮	刘强,吴亮,赵文博	清华大学	57691
13	Office办公软件应用	史巧硕,朱怀忠,刘洪磊,李娟	河北工业大学	57525
14	汇编语言程序设计	张悠慧,崔李冬	清华大学	55227
15	计算机应用基础	宋承继,李莹,李龙龙	陕西工业职业技术学院	55111
16	微软亚洲研究院大数据系列讲座	洪小文,宋睿华,谢幸,郑宇,张洪宇	Microsoft	52831
17	算法设计与分析	王振波	清华大学	51345
18	学做小程序——基础篇	刘强,小程序慕课讲师	清华大学	51061
19	大学计算机——计算思维的视角	郝兴伟	山东大学	49725
20	面向对象程序设计(C++)	黄震春,徐明星	清华大学	49054
21	数据科学导论	袁博,何勇	清华大学	48674
22	大学计算机基础	卫春芳	湖北大学	46903
23	基于Linux的C++	齐林	清华大学	45901
24	大学计算机基础	徐红云,刘欣欣,曹晓叶	华南理工大学	45116
25	计算机网络	袁华,杜广龙,张凌	华南理工大学	44637
26	JAVA程序设计进阶	许斌	清华大学	42072
27	Web开发技术	陈静,王成良,祝伟华	重庆大学	38317
28	计算几何	邓俊辉	清华大学	38151
29	程序设计基础(上)	赵宏,闫晓玉,李妍,王恺,李敏	南开大学	36495
30	大数据机器学习	袁春	清华大学	35741
31	ARM微控制器与嵌入式系统	曾鸣,薛涛,袁光华	清华大学	35457
32	R语言数据分析	艾新波	北京邮电大学	35425
33	单片机原理及应用	杨居义,王颖丽,唐敏,向兵	绵阳职业技术学院	34772
34	程序设计基础(下)	赵宏,李妍,闫晓玉,王恺,李敏	南开大学	32428
35	高级大数据系统	王智	清华大学	31637
36	Python程序设计基础	许志良	深圳信息职业技术学院	31303
37	大学计算机基础	李敏,高裴裴,郭蕴,赵宏,刘哲理,路明晓,李妍,闫晓玉,宋丽培	南开大学	30510
38	移动快速应用开发	唐贤传,盛鸿宇,王婷婷,汤恒,胡万福	芜湖职业技术学院	27496
39	人工智能	罗会兰	江西理工大学	26872
40	数据库技术与程序设计	高裴裴,李敏,袁晓洁,赵宏,路明晓,康介恢,闫晓玉,李妍,郭蕴	南开大学	26780
41	大数据技术与应用	李军	清华大学	97952
42	软件工程	刘强,刘璘	清华大学	93703
43	计算机文化基础	李秀,姚瑞霞,安颖莲,全成斌	清华大学	83710
44	程序设计基础	徐明星,王瑀屏,邬晓钧	清华大学	80789
45	组合数学	马昱春	清华大学	74391
46	大数据系统基础	王建民,徐葳,陈康,陈文光	清华大学	74244
47	VC++面向对象与可视化程序设计(上):	黄维通	清华大学	69319
48	VC++面向对象与可视化程序设计(下):	黄维通	清华大学	67918
49	人工智能原理	王文敏	北京大学	63686
50	大数据平台核心技术	武永卫,姚文辉,陶阳宇,冯骁,谢德军	清华大学	62945



## 5 爬取链家官网二手房

### 5.1 确定 items 类中元素

由于作业要求爬取楼盘名称、总价、平米数、单价，故在 *MyItem* 类中声明四个元素 *name*, *totalPrice*, *area*, *unitPrice* 分别代表楼盘名称、总价、平米数、单价，代码如下所示。

```
1 import scrapy
2
3 class MyItem(scrapy.Item):
4     name = scrapy.Field()
5     totalPrice = scrapy.Field()
6     area = scrapy.Field()
7     unitPrice = scrapy.Field()
```

### 5.2 获取 xpath

打开链家网站，利用 F12 浏览器调试者工具，找出若干个房源信息所在的 xpath, 结果如下表所示

表 1: xpath 结果

元素	xpath 结果
第一个房源的名称	//*[@id="content"]/div[1]/ul/li[1]/div[1]/div[2]/div/a[1]/text()
第二个房源的名称	//*[@id="content"]/div[1]/ul/li[2]/div[1]/div[2]/div/a[1]/text()
第三个房源的名称	//*[@id="content"]/div[1]/ul/li[3]/div[1]/div[2]/div/a[1]/text()
第一个房源的总价	//*[@id="content"]/div[1]/ul/li[1]/div[1]/div[6]/div[1]/span/text()
第二个房源的总价	//*[@id="content"]/div[1]/ul/li[2]/div[1]/div[6]/div[1]/span/text()
第三个房源的总价	//*[@id="content"]/div[1]/ul/li[3]/div[1]/div[6]/div[1]/span/text()
第一个房源的平米数	//*[@id="content"]/div[1]/ul/li[1]/div[1]/div[3]/div/text()
第二个房源的平米数	//*[@id="content"]/div[1]/ul/li[2]/div[1]/div[3]/div/text()
第三个房源的平米数	//*[@id="content"]/div[1]/ul/li[3]/div[1]/div[3]/div/text()
第一个房源的单价	//*[@id="content"]/div[1]/ul/li[1]/div[1]/div[6]/div[2]/span/text()
第二个房源的单价	//*[@id="content"]/div[1]/ul/li[2]/div[1]/div[6]/div[2]/span/text()
第三个房源的单价	//*[@id="content"]/div[1]/ul/li[3]/div[1]/div[6]/div[2]/span/text()

从表中可以轻易地发现元素所对应的 xpath 的规律，如下表所示 (其中 x 表示该页面的第 x 个房源)

表 2: xpath 结果

元素	xpath 结果
房源的名称	<code>//*[@id="content"]/div[1]/ul/li[x]/div[1]/div[2]/div/a[1]/text()</code>
房源的总价	<code>//*[@id="content"]/div[1]/ul/li[x]/div[1]/div[6]/div[1]/span/text()</code>
房源的平米数	<code>//*[@id="content"]/div[1]/ul/li[x]/div[1]/div[3]/div/text()</code>
房源的单价	<code>//*[@id="content"]/div[1]/ul/li[x]/div[1]/div[6]/div[2]/span/text()</code>

### 5.3 编写 spider.py

将项目名称设为 `lianjia`，设置初始 url 为 `https://bj.lianjia.com/ershoufang/dongcheng/`。根据之前发现的 xpath 的规律，在 `parse` 函数中对每个 xpath 为 `//*[@id="content"]/div[1]/ul/li` 的元素进行遍历，通过上表所总结的房源信息对应的 xpath，将信息写入 item 对应的元素中去。

由于网页的信息太过于冗杂，其中要涉及字符串的处理，题目要求获取房源的面积，可网页将面积置于“1 室 1 厅 | 43.78 平米 | 南 | 精装 | 中楼层 (共 12 层) | 2009 年建 | 板塔结合”结构中，该程序利用字符串的 `split` 函数，对字符串以“|”进行分割，然后获取其第二个元素即为面积；题目还要求获取房源的价格，可网页将 390 和万置于两个元素中，故要分别获取并利用 `join` 函数合并。

作业还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），刚开始我将这 4 个区的前五页的 url 共计 30 个都写在 `start_urls` 中，代码也能运行，后来在 CSDN 论坛上了解了可以利用 `for` 循环结合 `Request` 函数可以轻松取得同样的效果，具体操作是：利用循环中 `i` 值的变化，使得 url 更新，再利用 `Request` 函数发生跳转，最后回调 `parse` 函数做相同的处理。具体代码如下

```
1 import scrapy
2 from test1.items import MyItem
3 from scrapy.http import Request
4
5
6 class MySpider(scrapy.Spider):
7     name = "lianjia"
8     start_urls = ["https://bj.lianjia.com/ershoufang/dongcheng/"] #
    初始url
```

```

9
10 def parse(self, response):
11     item = MyItem()
12     for each in response.xpath('//*[@id="content"]/div[1]/ul/li'):
13         item['name'] = each.xpath("div[1]/div[2]/div/a[1]/text()")
14         .extract()
15         str1 = ''.join((str(each.xpath("div[1]/div[6]/div[1]/span
16         /text()").extract()[0]),
17         str(each.xpath("div[1]/div[6]/div[1]/text
18         ()").extract()[0])))
19         item['totalPrice'] = str1.split('|')[0:1]
20         str1 = ''.join(each.xpath("div[1]/div[3]/div/text()").
21         extract())
22         item['area'] = str1.split('|')[1:2]
23         item['unitPrice'] = each.xpath("div[1]/div[6]/div[2]/span
24         /text()").extract()
25         if item['name'] and item['totalPrice'] and item['
26         unitPrice'] and item['area']:
27             yield (item)
28
29     for i in range(2, 6):
30         url = 'https://bj.lianjia.com/ershoufang/dongcheng/pg{}/'
31         .format(str(i))
32         yield Request(url, callback=self.parse) # 回调
33     for i in range(1, 6):
34         url = 'https://bj.lianjia.com/ershoufang/xicheng/pg{}/'
35         .format(str(i))
36         yield Request(url, callback=self.parse) # 回调
37     for i in range(1, 6):
38         url = 'https://bj.lianjia.com/ershoufang/haidian/pg{}/'
39         .format(str(i))
40         yield Request(url, callback=self.parse) # 回调
41     for i in range(1, 6):
42         url = 'https://bj.lianjia.com/ershoufang/chaoyang/pg{}/'
43         .format(str(i))
44         yield Request(url, callback=self.parse) # 回调

```

## 5.4 编写 pipelines.py

pipelines.py 是写如何存储数据的程序，将 item 生成字典对象，然后生成 json 串，最后将 json 串写入文件，具体代码如下

```
1 import json
2
3
4 class MyPipeline(object):
5     def open_spider(self, spider):
6         try: # 打开json文件
7             self.file = open('MyData.json', "w", encoding="utf-8")
8         except Exception as err:
9             print(err)
10
11     def process_item(self, item, spider):
12         dict_item = dict(item) # 生成字典对象
13         json_str = json.dumps(dict_item, ensure_ascii=False) + "\n"
14         # 生成json串
15         # self.file = open('MyData.json', "w", encoding="utf-8")
16         self.file.write(json_str) # 将json串写入到文件中
17         return item
18
19     def close_spider(self, spider):
20         self.file.close() # 关闭文件
```

## 5.5 运行结果 (截取前 50 条)

```
1 {"name": ["太华公寓 "], "totalPrice": ["925万"], "area": [" 150.92平
   米 "], "unitPrice": ["单价61291元/平米"]}
2 {"name": ["华龙美晟 "], "totalPrice": ["390万"], "area": [" 43.78平米
   "], "unitPrice": ["单价89082元/平米"]}
3 {"name": ["安贞苑50号院 "], "totalPrice": ["540万"], "area": [" 49.29
   平米 "], "unitPrice": ["单价109556元/平米"]}
4 {"name": ["青年湖北里 "], "totalPrice": ["1000万"], "area": [" 97.28
   平米 "], "unitPrice": ["单价102797元/平米"]}
5 {"name": ["中海紫御公馆 "], "totalPrice": ["1100万"], "area": ["
   91.22平米 "], "unitPrice": ["单价120588元/平米"]}
```

```

6 {"name": ["景泰西里西区 "], "totalPrice": ["389万"], "area": [" 56.38
   平米 "], "unitPrice": ["单价68997元/平米"]}
7 {"name": ["广渠门内大街 "], "totalPrice": ["638万"], "area": [" 76.59
   平米 "], "unitPrice": ["单价83301元/平米"]}
8 {"name": ["新景家园西区 "], "totalPrice": ["615万"], "area": [" 58.45
   平米 "], "unitPrice": ["单价105219元/平米"]}
9 {"name": ["民安小区东羊管胡同 "], "totalPrice": ["780万"], "area": ["
   70.01平米 "], "unitPrice": ["单价111413元/平米"]}
10 {"name": ["西总布胡同 "], "totalPrice": ["501万"], "area": [" 50.36平
   米 "], "unitPrice": ["单价99484元/平米"]}
11 {"name": ["六铺炕 "], "totalPrice": ["736万"], "area": [" 68.8平米
   "], "unitPrice": ["单价106977元/平米"]}
12 {"name": ["东直门外大街(东城) "], "totalPrice": ["616万"], "area":
   [" 66.23平米 "], "unitPrice": ["单价93010元/平米"]}
13 {"name": ["东方银座 "], "totalPrice": ["590万"], "area": [" 76.4平米
   "], "unitPrice": ["单价77226元/平米"]}
14 {"name": ["望陶园小区 "], "totalPrice": ["588万"], "area": [" 89.66平
   米 "], "unitPrice": ["单价65582元/平米"]}
15 {"name": ["永定门东街东里 "], "totalPrice": ["549万"], "area": [" 68
   平米 "], "unitPrice": ["单价80736元/平米"]}
16 {"name": ["双玉中街43号院 "], "totalPrice": ["579万"], "area": ["
   61.22平米 "], "unitPrice": ["单价94577元/平米"]}
17 {"name": ["新裕家园 "], "totalPrice": ["831万"], "area": [" 76.76平米
   "], "unitPrice": ["单价108260元/平米"]}
18 {"name": ["朝阳门北小街 "], "totalPrice": ["815万"], "area": [" 69平
   米 "], "unitPrice": ["单价118116元/平米"]}
19 {"name": ["华龙美晟 "], "totalPrice": ["590万"], "area": [" 85.45平米
   "], "unitPrice": ["单价69047元/平米"]}
20 {"name": ["广渠门外南街 "], "totalPrice": ["370万"], "area": [" 46.8
   平米 "], "unitPrice": ["单价79060元/平米"]}
21 {"name": ["安德路47号院 "], "totalPrice": ["586万"], "area": [" 59.75
   平米 "], "unitPrice": ["单价98076元/平米"]}
22 {"name": ["景泰西里西区 "], "totalPrice": ["410万"], "area": [" 60.78
   平米 "], "unitPrice": ["单价67457元/平米"]}
23 {"name": ["国瑞城中区 "], "totalPrice": ["870万"], "area": [" 90.84平
   米 "], "unitPrice": ["单价95773元/平米"]}
24 {"name": ["海晟名苑南区 "], "totalPrice": ["1730万"], "area": ["
   144.26平米 "], "unitPrice": ["单价119923元/平米"]}
25 {"name": ["富莱茵花园 "], "totalPrice": ["619万"], "area": [" 91.26平

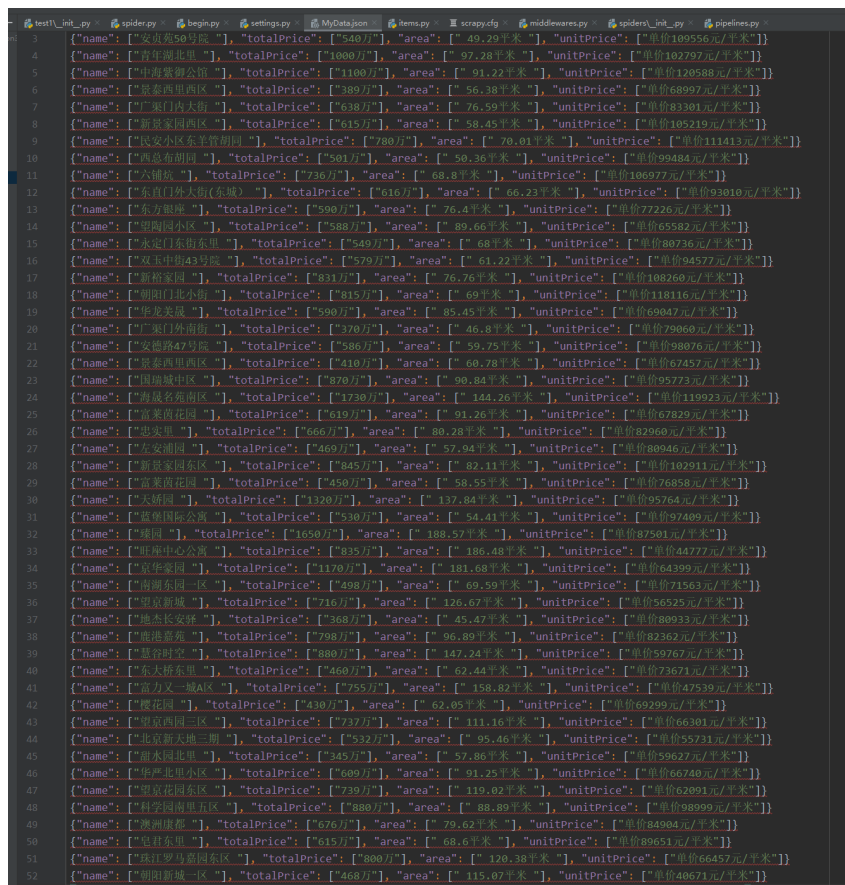
```

```

    米 "], "unitPrice": ["单价67829元/平米"]}]
26 {"name": ["忠实里 "], "totalPrice": ["666万"], "area": [" 80.28平米
    "], "unitPrice": ["单价82960元/平米"]}]
27 {"name": ["左安浦园 "], "totalPrice": ["469万"], "area": [" 57.94平米
    "], "unitPrice": ["单价80946元/平米"]}]
28 {"name": ["新景家园东区 "], "totalPrice": ["845万"], "area": [" 82.11
    平米 "], "unitPrice": ["单价102911元/平米"]}]
29 {"name": ["富莱茵花园 "], "totalPrice": ["450万"], "area": [" 58.55平
    米 "], "unitPrice": ["单价76858元/平米"]}]
30 {"name": ["天娇园 "], "totalPrice": ["1320万"], "area": [" 137.84平米
    "], "unitPrice": ["单价95764元/平米"]}]
31 {"name": ["蓝堡国际公寓 "], "totalPrice": ["530万"], "area": [" 54.41
    平米 "], "unitPrice": ["单价97409元/平米"]}]
32 {"name": ["臻园 "], "totalPrice": ["1650万"], "area": [" 188.57平米
    "], "unitPrice": ["单价87501元/平米"]}]
33 {"name": ["旺座中心公寓 "], "totalPrice": ["835万"], "area": ["
    186.48平米 "], "unitPrice": ["单价44777元/平米"]}]
34 {"name": ["京华豪园 "], "totalPrice": ["1170万"], "area": [" 181.68平
    米 "], "unitPrice": ["单价64399元/平米"]}]
35 {"name": ["南湖东园一区 "], "totalPrice": ["498万"], "area": [" 69.59
    平米 "], "unitPrice": ["单价71563元/平米"]}]
36 {"name": ["望京新城 "], "totalPrice": ["716万"], "area": [" 126.67平
    米 "], "unitPrice": ["单价56525元/平米"]}]
37 {"name": ["地杰长安驿 "], "totalPrice": ["368万"], "area": [" 45.47平
    米 "], "unitPrice": ["单价80933元/平米"]}]
38 {"name": ["鹿港嘉苑 "], "totalPrice": ["798万"], "area": [" 96.89平米
    "], "unitPrice": ["单价82362元/平米"]}]
39 {"name": ["慧谷时空 "], "totalPrice": ["880万"], "area": [" 147.24平
    米 "], "unitPrice": ["单价59767元/平米"]}]
40 {"name": ["东大桥东里 "], "totalPrice": ["460万"], "area": [" 62.44平
    米 "], "unitPrice": ["单价73671元/平米"]}]
41 {"name": ["富力又一城A区 "], "totalPrice": ["755万"], "area": ["
    158.82平米 "], "unitPrice": ["单价47539元/平米"]}]
42 {"name": ["樱花园 "], "totalPrice": ["430万"], "area": [" 62.05平米
    "], "unitPrice": ["单价69299元/平米"]}]
43 {"name": ["望京西园三区 "], "totalPrice": ["737万"], "area": ["
    111.16平米 "], "unitPrice": ["单价66301元/平米"]}]
44 {"name": ["北京新天地三期 "], "totalPrice": ["532万"], "area": ["
    95.46平米 "], "unitPrice": ["单价55731元/平米"]}]

```

```
45 {"name": ["甜水园北里 "], "totalPrice": ["345万"], "area": [" 57.86平  
米 "], "unitPrice": ["单价59627元/平米"]}  
46 {"name": ["华严北里小区 "], "totalPrice": ["609万"], "area": [" 91.25  
平米 "], "unitPrice": ["单价66740元/平米"]}  
47 {"name": ["望京花园东区 "], "totalPrice": ["739万"], "area": ["  
119.02平米 "], "unitPrice": ["单价62091元/平米"]}  
48 {"name": ["科学园南里五区 "], "totalPrice": ["880万"], "area": ["  
88.89平米 "], "unitPrice": ["单价98999元/平米"]}  
49 {"name": ["澳洲康都 "], "totalPrice": ["676万"], "area": [" 79.62平米  
"], "unitPrice": ["单价84904元/平米"]}  
50 {"name": ["皂君东里 "], "totalPrice": ["615万"], "area": [" 68.6平米  
"], "unitPrice": ["单价89651元/平米"]}
```



## 6 作业总结

通过这次爬虫作业，我了解了爬虫的基本功能以及如何利用爬虫爬取动态网页和静态网页，虽然这次作业对我个人来说难度较大，东西比较新颖，花费了不少时间查阅资料和博客，但是我的收获无疑是非常大的。