# 爬虫测试

## 创建爬虫项目

**scrapy startproject 项目名**

C:\Users\lucifer\Desktop\learnpython\python\_advan\stage05\作业>**scrapy startproject sproject**

New Scrapy project 'sproject', using template directory 'c:\\users\\lucifer\\appdata\\local\\programs\\python\\python36\\lib\\site-packages\\scrapy\\templates\\project', created in:

C:\Users\lucifer\Desktop\learnpython\python\_advan\stage05\作业\sproject

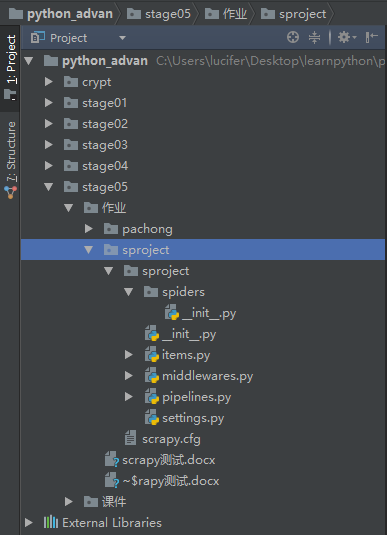
You can start your first spider with:

cd sproject

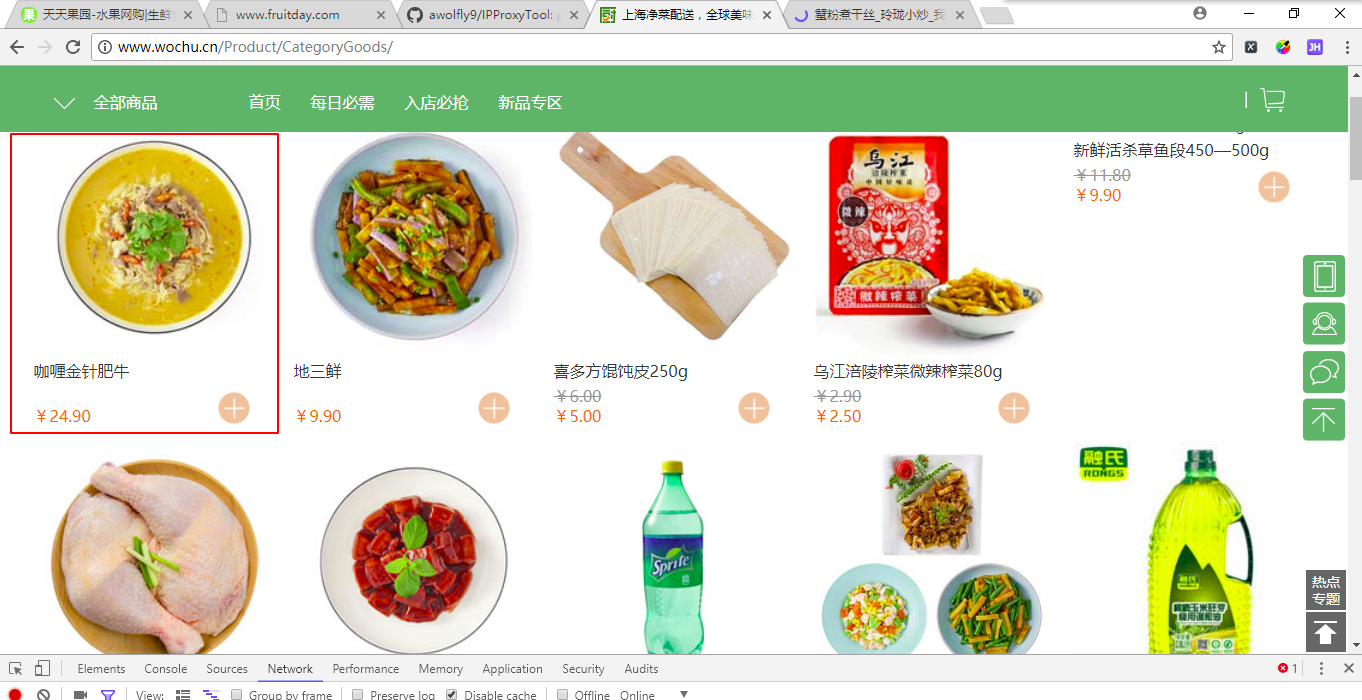
scrapy genspider example example.com

C:\Users\lucifer\Desktop\learnpython\python\_advan\stage05\作业>

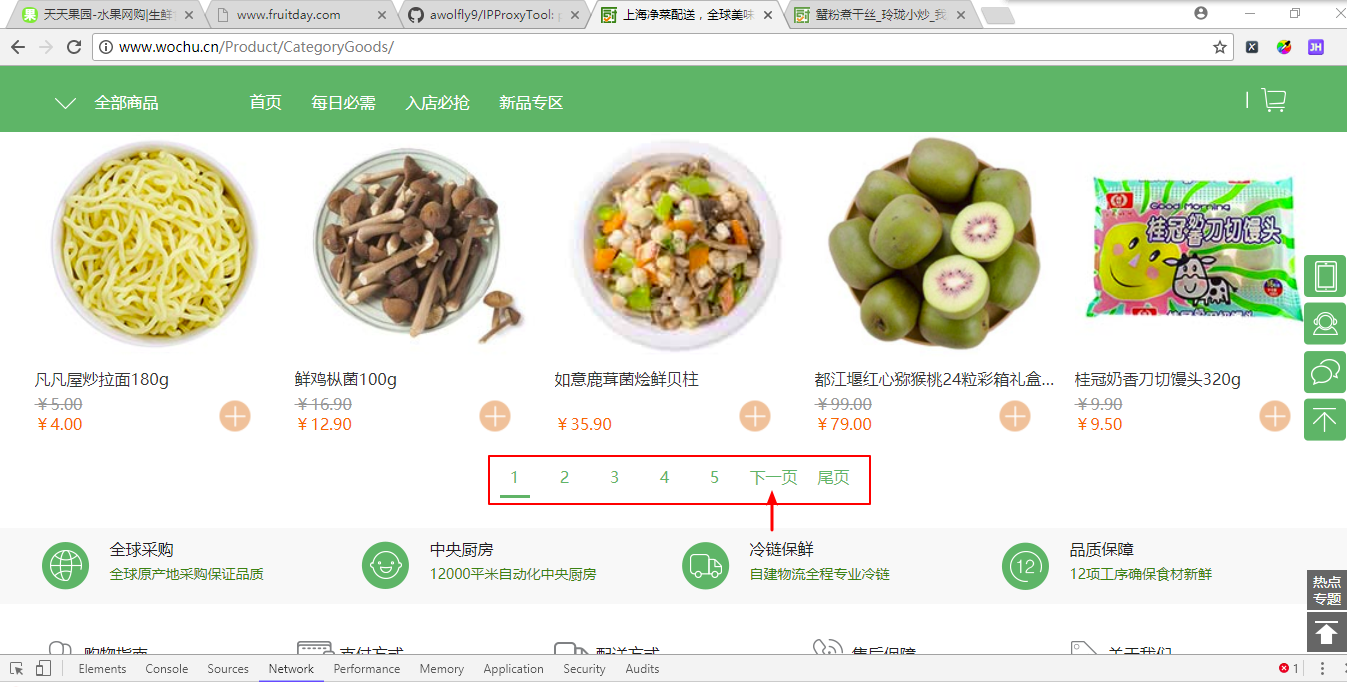
可以看到生成了项目目录



然后需要知道要爬去什么数据，需要到网站踩点，

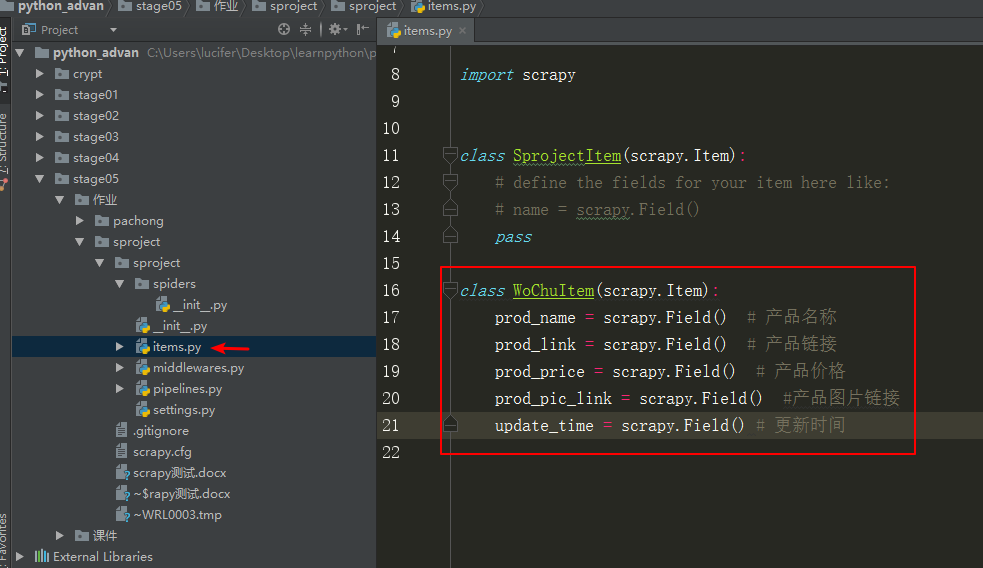


以我厨为例，可以看到 有产品图片，产品链接，产品价格，产品名称



在分页尾部也有下一页，这样的网站方便一个一个下一页爬去

## 在item.py 建立数据模型



源代码如下：

*class* WoChuItem(scrapy.Item):  
 prod\_name = scrapy.Field() # 产品名称  
 prod\_link = scrapy.Field() # 产品链接  
 prod\_price = scrapy.Field() # 产品价格  
 prod\_pic\_link = scrapy.Field() #产品图片链接  
 update\_time = scrapy.Field() # 更新时间

## 在项目目录下创建爬虫

**scrapy genspider 爬虫名 爬虫允许的域名**

C:\Users\lucifer\Desktop\learnpython\python\_advan\stage05\作业\sproject>

**scrapy genspider wochu wochu.cn**

Created spider 'wochu' using template 'basic' in module:

sproject.spiders.wochu

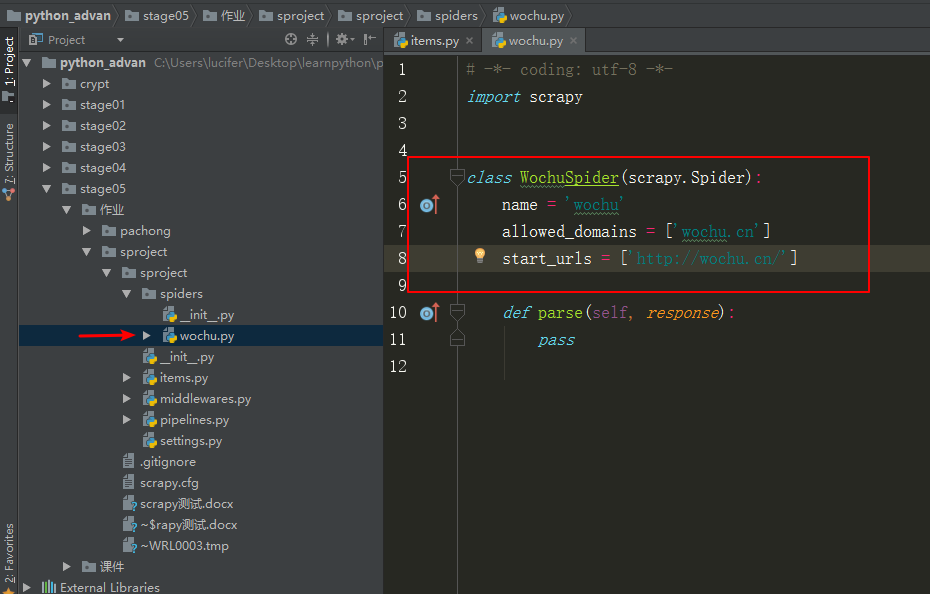
C:\Users\lucifer\Desktop\learnpython\python\_advan\stage05\作业\sproject>

查看爬虫列表

C:\Users\lucifer\Desktop\learnpython\python\_advan\stage05\作业\sproject>**scrapy list**

wochu

可以看到生成了爬虫文件

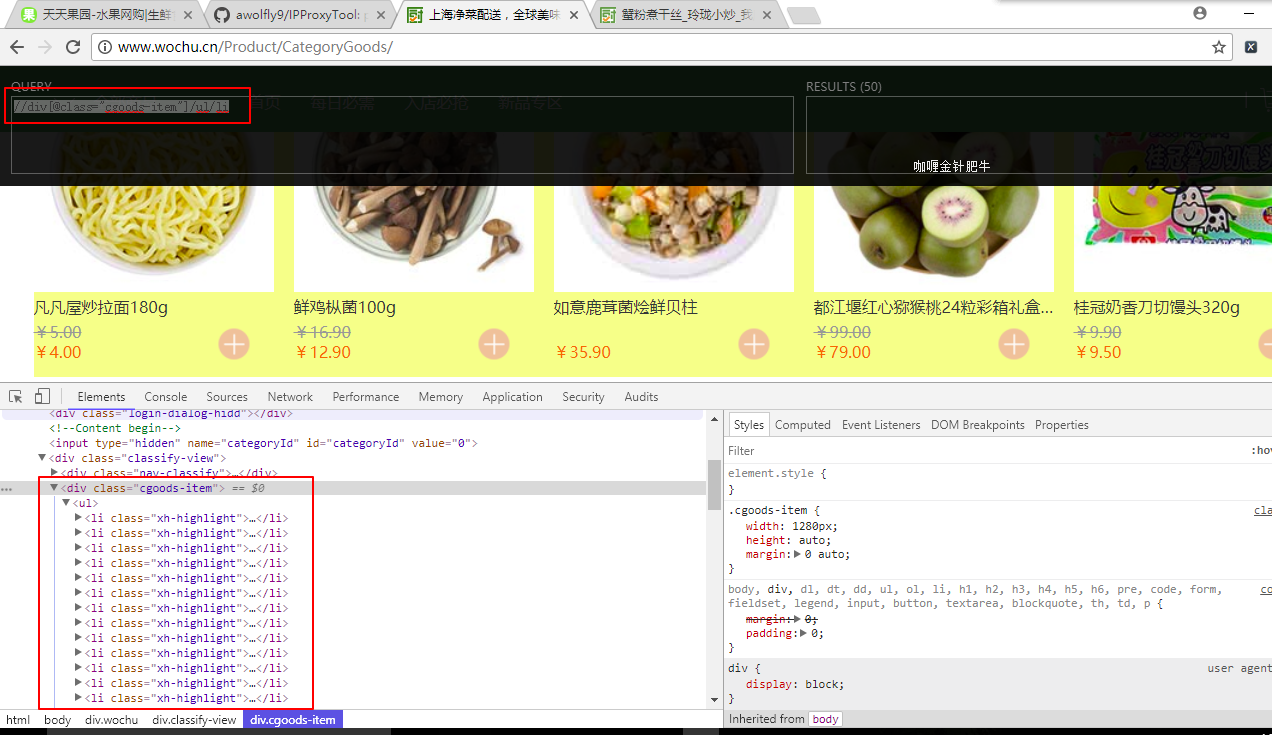


源代码如下：

*import* scrapy  
  
  
*class* WochuSpider(scrapy.Spider):  
 name = 'wochu'  
 allowed\_domains = ['wochu.cn']  
 start\_urls = ['http://wochu.cn/']  
  
 *def* parse(self, *response*):  
 *pass*

## 编写处理函数 parse

通过观察页面特征获取xpath



//div[@class="cgoods-item"]/ul/li

先写简单代码看看爬虫能不能获取到数据

*def* parse(self, *response*):  
 node\_list = *response*.xpath('//div[@class="cgoods-item"]/ul/li')  
 print(node\_list)

运行爬虫

C:\Users\lucifer\Desktop\learnpython\python\_advan\stage05\作业\sproject>**scrapy crawl wochu**

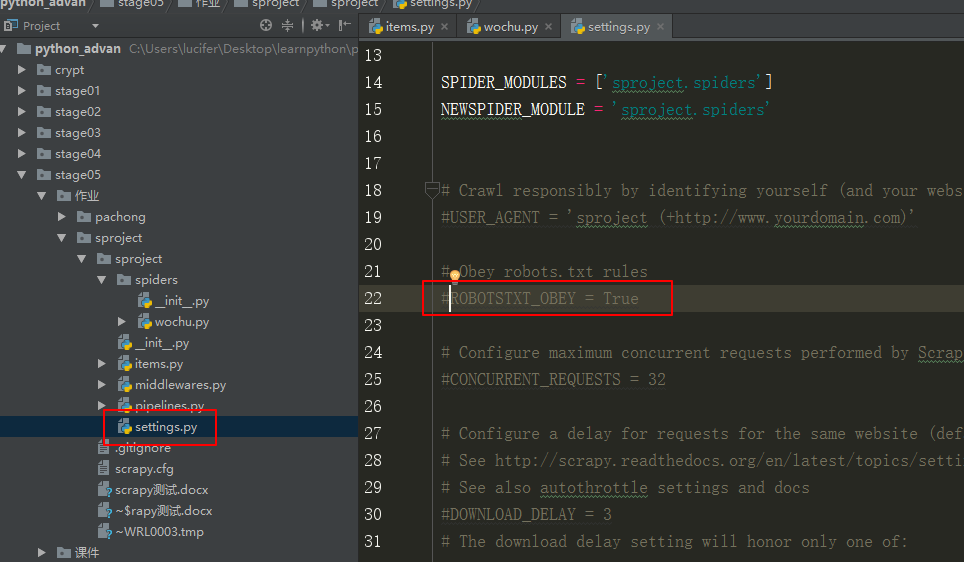
报错

ModuleNotFoundError: No module named 'win32api'

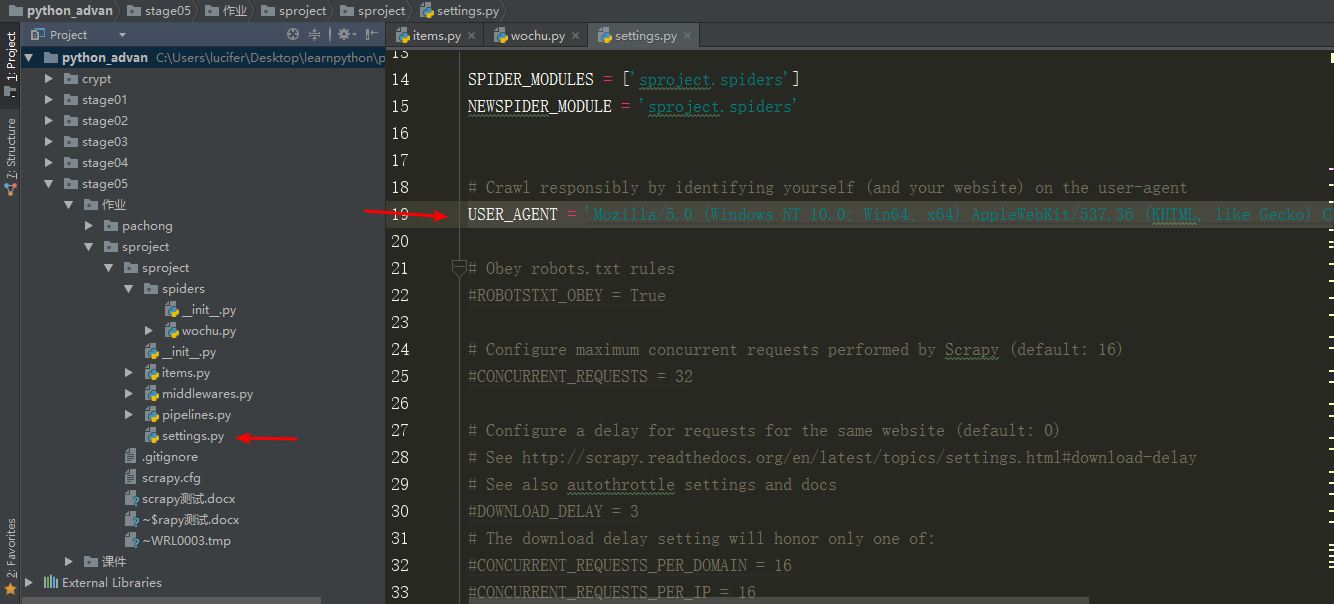
网上找了下解决方案

C:\Users\lucifer\Desktop\learnpython\python\_advan\stage05\作业\sproject>pip install pypiwin32

在settings.py中禁用robot协议

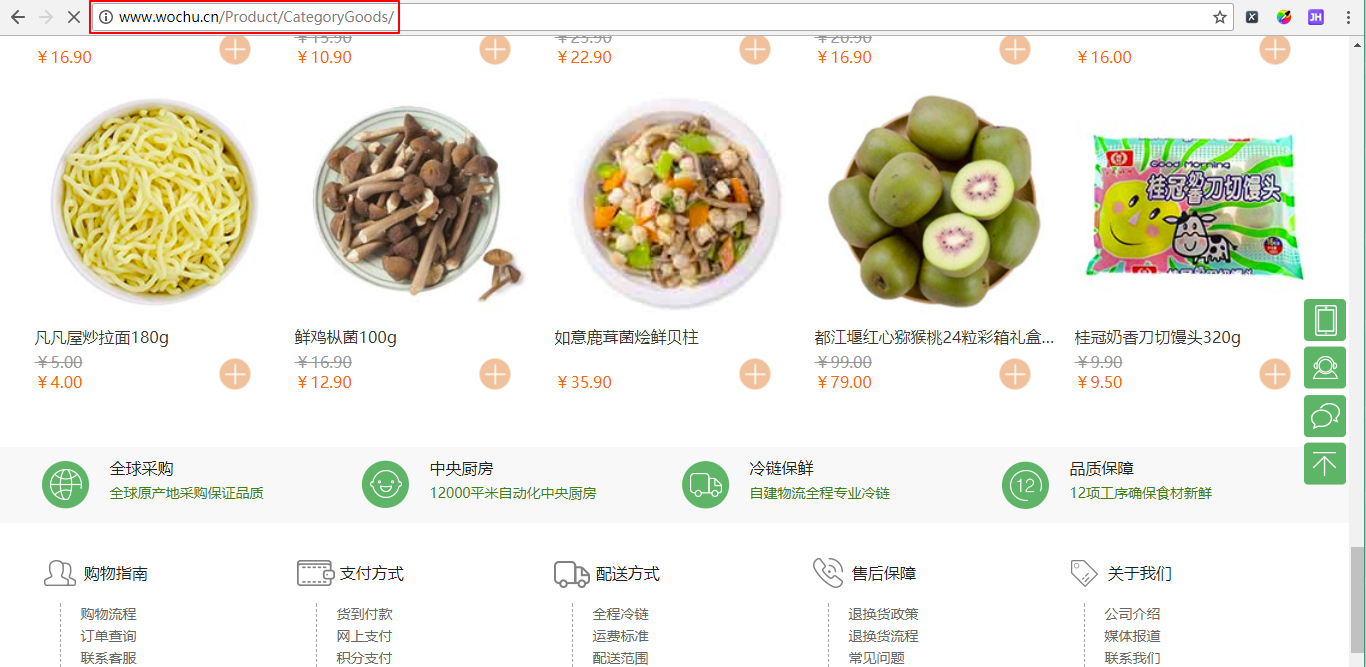


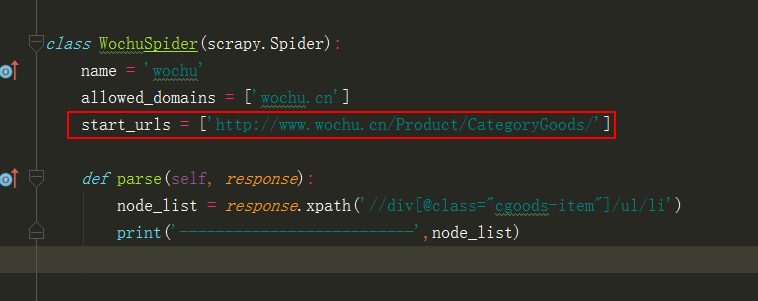
顺便伪装成良民



USER\_AGENT = 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/62.0.3202.75 Safari/537.36'

再次爬去之前发现 start\_urls写错了 ，我们要的是产品页，所以修改以下

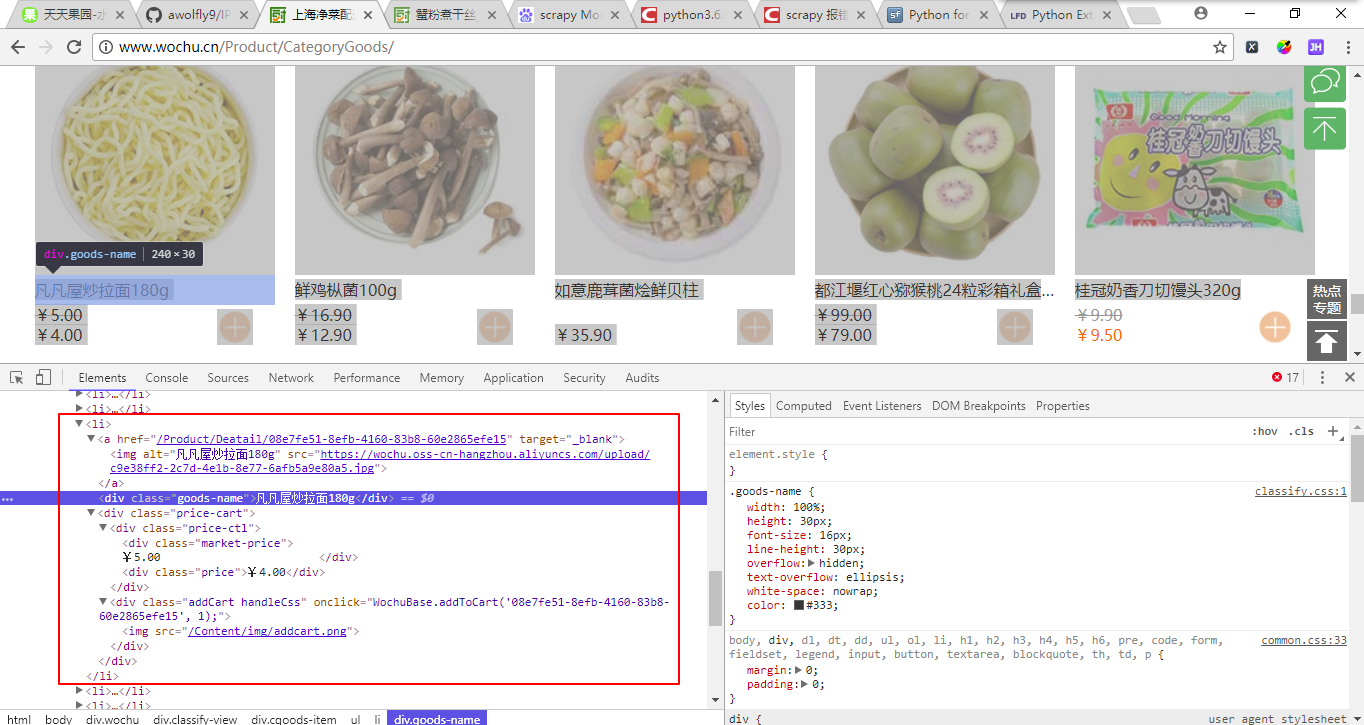


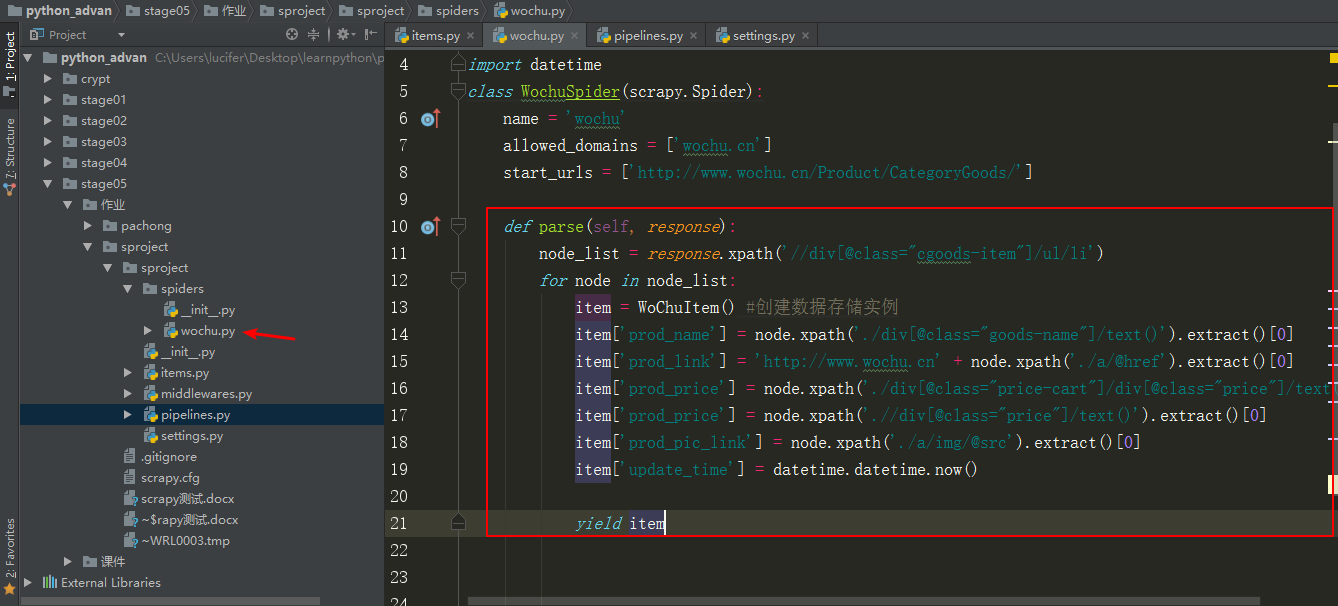


可以看到已经获取到数据了，接下来就是解析数据了



接下来对li的商品结构进行分析





源代码

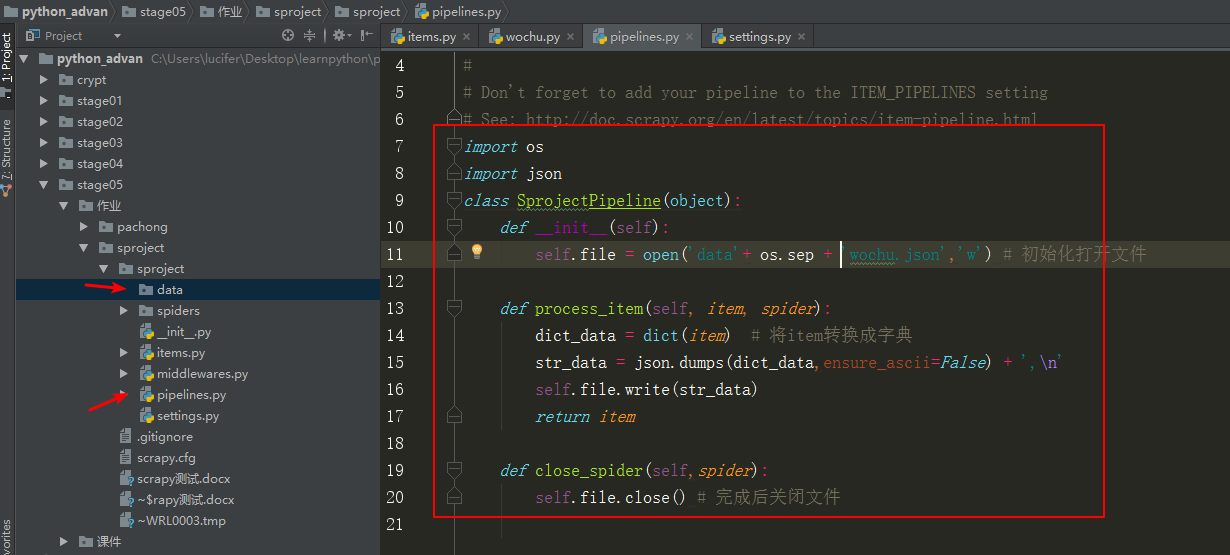
*def* parse(self, *response*):  
 node\_list = *response*.xpath('//div[@class="cgoods-item"]/ul/li')  
 *for* node *in* node\_list:  
 item = WoChuItem() #创建数据存储实例  
 item['prod\_name'] = node.xpath('./div[@class="goods-name"]/text()').extract()[0]  
 item['prod\_link'] = 'http://www.wochu.cn' + node.xpath('./a/@href').extract()[0]  
 item['prod\_price'] = node.xpath('./div[@class="price-cart"]/div[@class="price"]/text()')  
 item['prod\_price'] = node.xpath('.//div[@class="price"]/text()').extract()[0]  
 item['prod\_pic\_link'] = node.xpath('./a/img/@src').extract()[0]  
 item['update\_time'] = str(datetime.datetime.now())  
  
 *yield* item

数据成功过获取到了



新建一个data文件夹用来保存数据

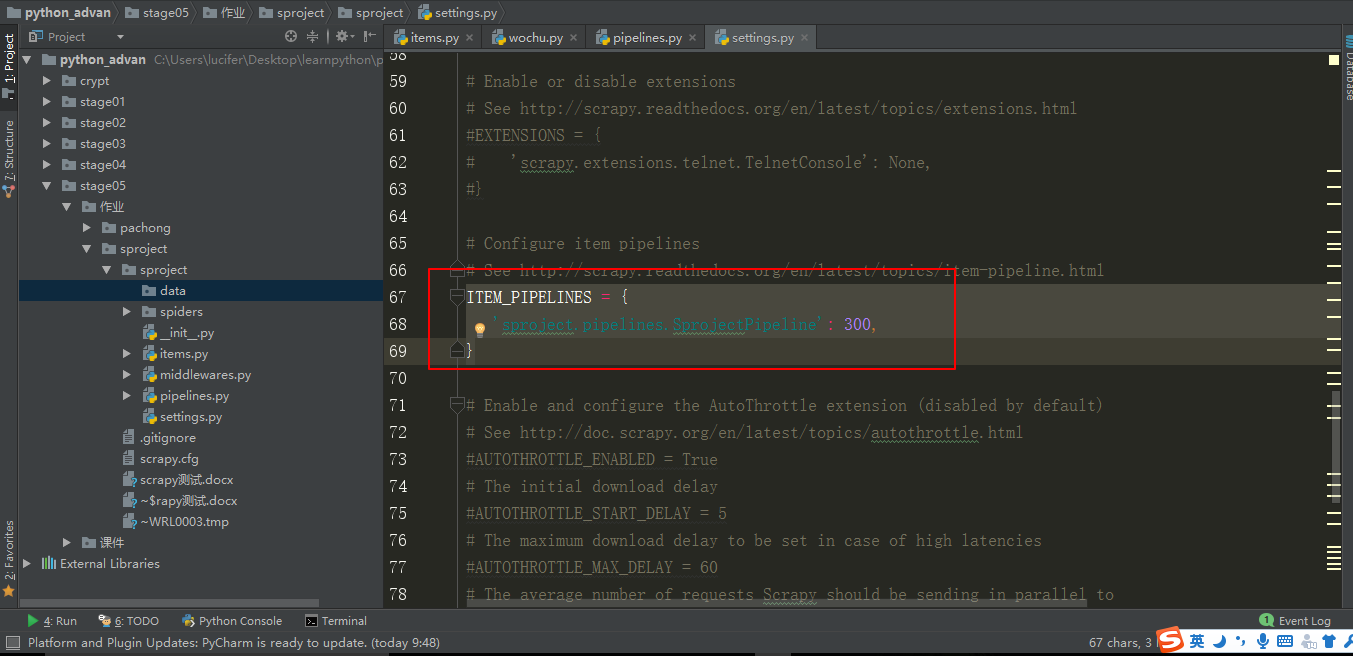
然后在修改pipelines.py文件夹 保存数据



具体代码如下：

*import* json  
*class* SprojectPipeline(object):  
 *def* \_\_init\_\_(self):  
 self.file = open('.\\sproject\\data\\'+ 'wochu.json','w',encoding='utf-8') # 初始化打开文件  
  
 *def* process\_item(self, *item*, *spider*):  
 dict\_data = dict(*item*) # 将item转换成字典  
 str\_data = json.dumps(dict\_data,ensure\_ascii=*False*) + ',\n'  
 self.file.write(str\_data)  
 *return item  
  
 def* close\_spider(self,*spider*):  
 self.file.close() # 完成后关闭文件

在settings.py中启用pipeline

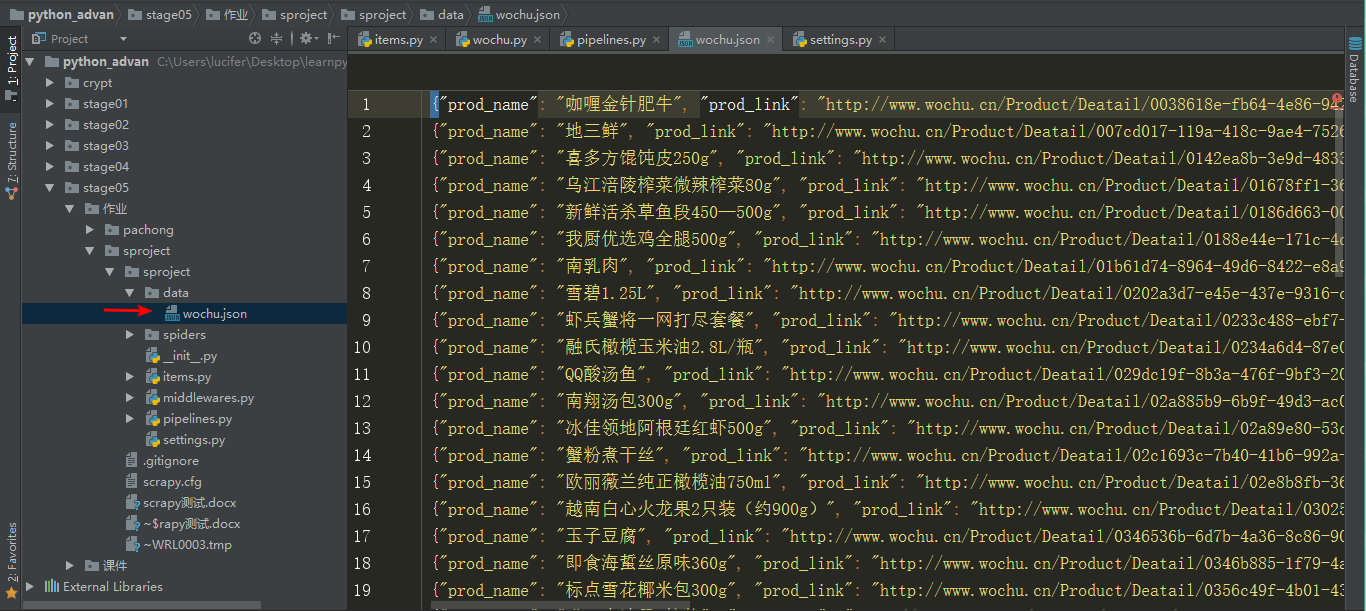


源代码如下

ITEM\_PIPELINES = {  
 'sproject.pipelines.SprojectPipeline': 300,  
}

C:\Users\lucifer\Desktop\learnpython\python\_advan\stage05\作业\sproject>**scrapy crawl wochu**

运行完成后正确收到数据

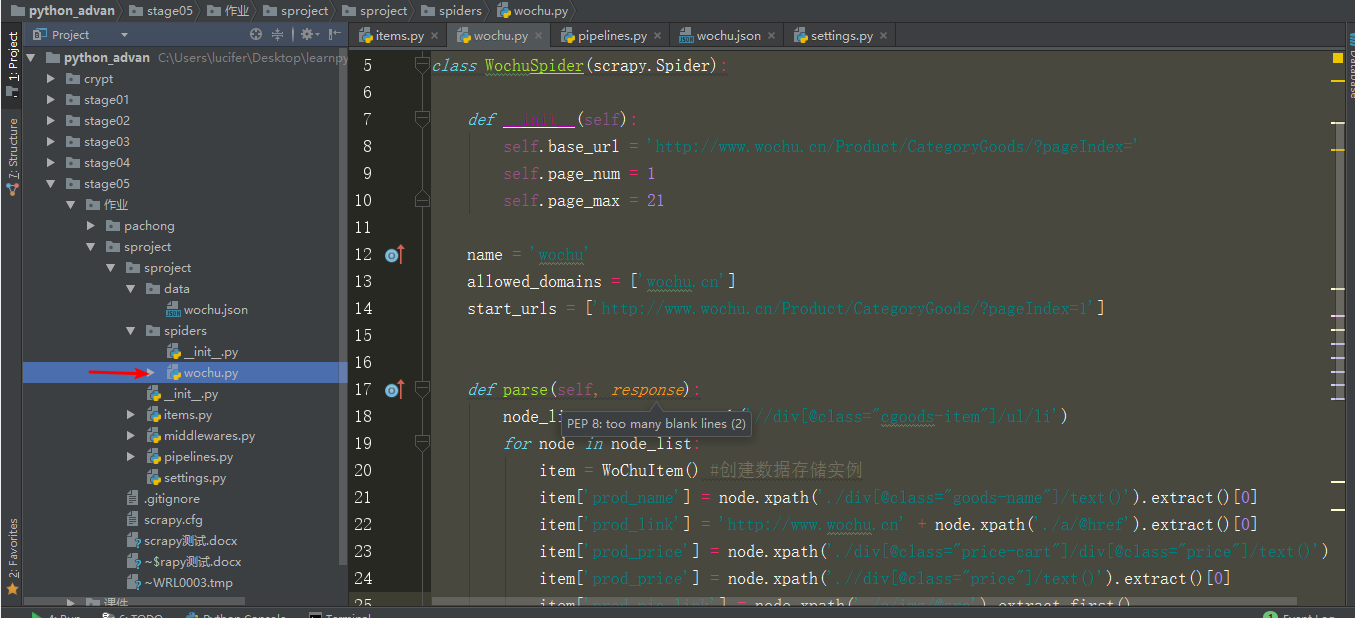
****

以上只是获取了一页数据而已，接下来再做一下分页即可

发现分页是通过url来实现的 一共21页商品



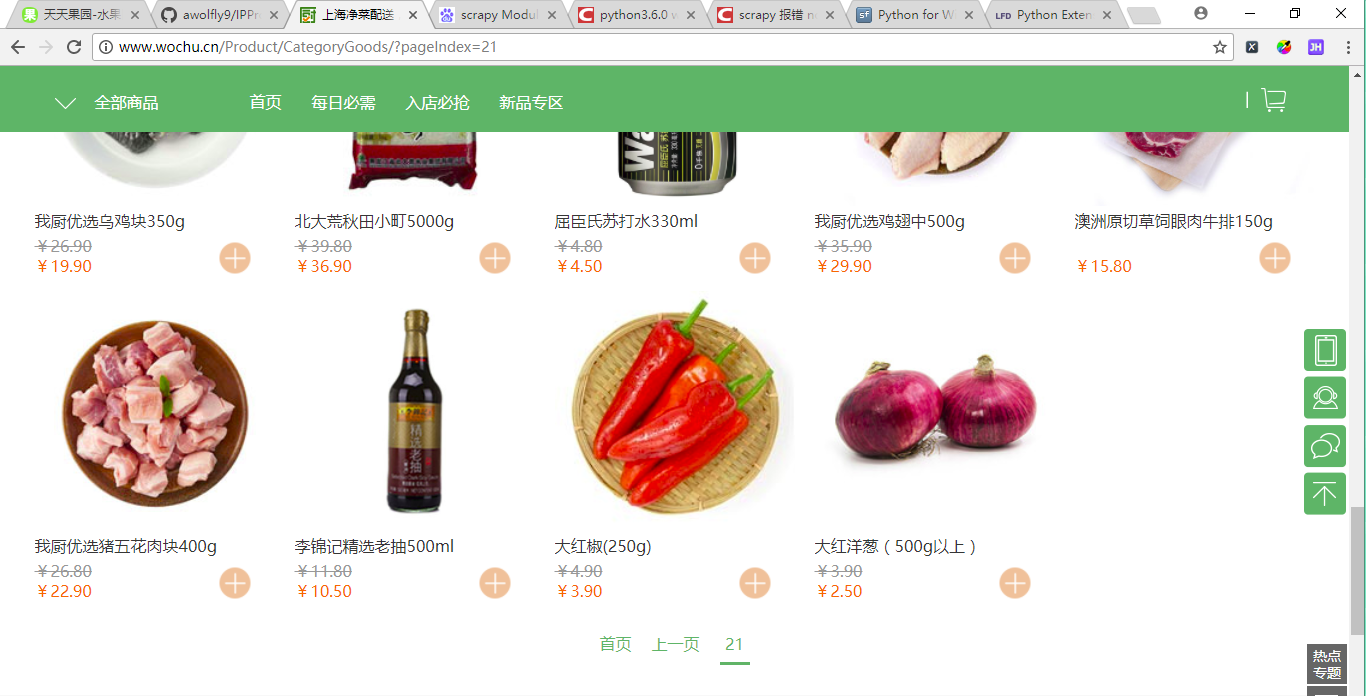
经过一番波折 终于把翻页写完

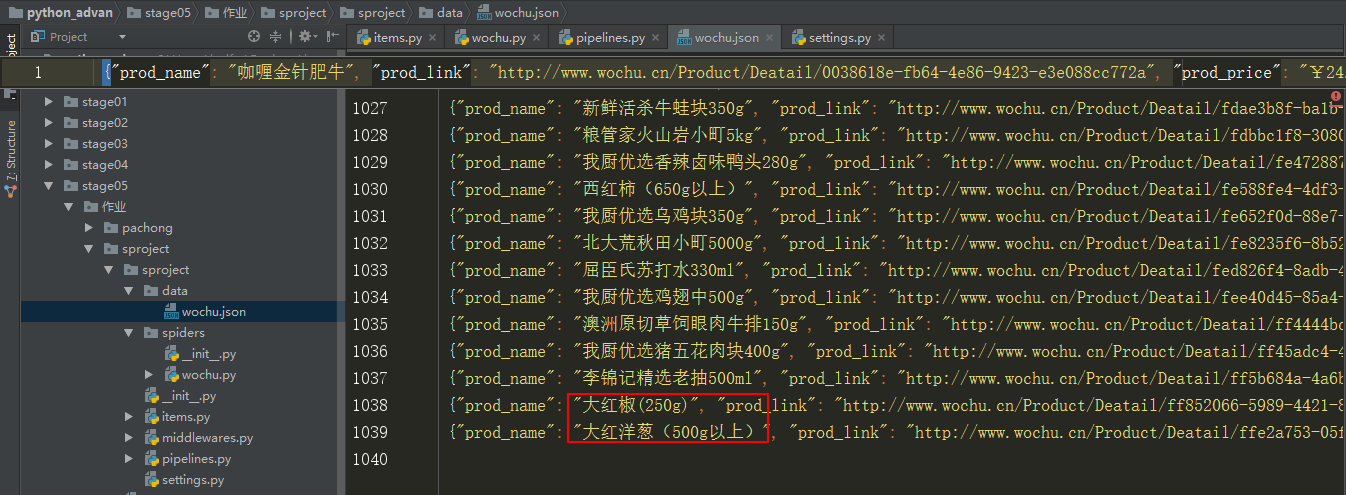


源代码如下

*import* scrapy  
*from* ..items *import* \*  
*import* datetime  
*class* WochuSpider(scrapy.Spider):  
  
 *def* \_\_init\_\_(self):  
 self.base\_url = 'http://www.wochu.cn/Product/CategoryGoods/?pageIndex='  
 self.page\_num = 1  
 self.page\_max = 21  
  
 name = 'wochu'  
 allowed\_domains = ['wochu.cn']  
 start\_urls = ['http://www.wochu.cn/Product/CategoryGoods/?pageIndex=1']  
  
  
 *def* parse(self, *response*):  
 node\_list = *response*.xpath('//div[@class="cgoods-item"]/ul/li')  
 *for* node *in* node\_list:  
 item = WoChuItem() #创建数据存储实例  
 item['prod\_name'] = node.xpath('./div[@class="goods-name"]/text()').extract()[0]  
 item['prod\_link'] = 'http://www.wochu.cn' + node.xpath('./a/@href').extract()[0]  
 item['prod\_price'] = node.xpath('./div[@class="price-cart"]/div[@class="price"]/text()')  
 item['prod\_price'] = node.xpath('.//div[@class="price"]/text()').extract()[0]  
 item['prod\_pic\_link'] = node.xpath('./a/img/@src').extract\_first()  
 item['update\_time'] = str(datetime.datetime.now())  
  
 *yield* item  
  
 *if* self.page\_num <= self.page\_max:  
 self.page\_num +=1  
 next\_url = self.base\_url + str(self.page\_num)  
 *yield* scrapy.Request(next\_url,callback=self.parse)

然后在看一下数据





最后几个产品是一致的，那应该没有啥问题