



《爬虫与数据分析》期末作业

|  |  |
| --- | --- |
| 题 目： | 爬虫期末作业 |
| 二级学院： | 数学学院 |
| 班 级： | 16计算机科学与技术三班 |
| 姓 名： | 王霜霜 |
| 学 号： | 16219111324 |
| 完成日期： | 2019年06月13日 |

温州大学瓯江学院教务部

二○一二年十一月制

目录

[第一章 主要内容 1](#_Toc11500291)

[1.1 静态页面的爬取 1](#_Toc11500292)

[1.2 动态页面的爬取 1](#_Toc11500293)

[1.3 淘宝书包定向爬取 1](#_Toc11500294)

[1.4 12306自动登录 1](#_Toc11500295)

[1.5 Scrapy爬取豆瓣电影 2](#_Toc11500296)

[1.6 分布式爬取 2](#_Toc11500297)

[1.7 深广度优先递归爬虫(百度百科) 2](#_Toc11500298)

[1.8 Github链接 2](#_Toc11500299)

[第二章 代码实现 2](#_Toc11500300)

[第三章 结果截图 4](#_Toc11500301)

# 主要内容

## 静态页面的爬取

1. **豆瓣Top250电影**
2. 通过网址的规律性用requests向http发送请求获取豆瓣top250网页信息
3. etree和xpath结合使用解析网页内容获取需要的标签（标题）。
4. 连接mysql数据库，通过循环使用游标 cur 操作 execute()方法将获得的数据插入数据库表中。
5. 采用Django展示数据库中存储的内容，利用bootstrap实现页面的美化。
6. **天气预报的爬取**

获取数据：通过使用 urllib.request 模块模拟浏览器的一个请求发起过程获取网页内容；BeautifulSoup 和 lxml 解析网页查找网页中的元素获得日期，天气，温度等信息。

其他爬取过程类似豆瓣Top250电影。

## 动态页面的爬取

1. **京东手机的爬取**

通过使用 selenium(无头模式 headless)模拟浏览器驱动火狐浏览器完成抓取页面获取标签、下载手机图片并模拟鼠标翻页，将动态网页爬取变成静态网页爬取。

数据存储至mysql及展示如豆瓣电影。

## 淘宝书包定向爬取

Requests库获取网页，re库(正则表达式)对内容进行筛选提取，保存至数据库并显示

## 12306自动登录

使用Selenium驱动火狐浏览器获取用户名，密码框并输入，获取验证码图片路径并保存，将验证码post到解析验证码的网站上获取结果，并模拟验证码的点击，最后登陆。

## Scrapy爬取豆瓣电影

通过使用Scrapy框架爬取豆瓣Top250电影信息并存储到mysql数据库中展示在Django中，同时生成csv文件。

## 分布式爬取

通过使用Redis来进行分布式爬取，master.py负责获取网站中所有图片的链接地址，并且加入Redis数据库的队列中；slave.py负责从Redis服务器中读取队列中的图片链接，将图片下载下来并保存在硬盘中。

## 深广度优先递归爬虫(百度百科)

分别使用深度优先搜索和广度优先搜索获取百度百科的fenlei下的所有及条链接。

## Github链接

https://github.com/OhYeahhhhh/Spider.git

# 代码实现

此处列举部分代码，详细代码见github

**12306核心代码：**

driver=webdriver.Firefox()

driver.maximize\_window()

def getImgCode():

headers = {"User-Agent": "Mozilla/5.0 (Windows; U; Windows NT 6.0 x64; en-US; rv:1.9pre)Gecko/2008072421 Minefield/3.0.2pre"}

imgName="imgCode.jpg"

src1 = driver.find\_element\_by\_class\_name("imgCode").get\_attribute("src")

req = urllib.request.Request(src1, headers=headers)

resp = urllib.request.urlopen(req, timeout=100)

data = resp.read()

if os.path.isfile(imgName):

os.remove(imgName)

fobj = open(imgName, "wb")

fobj.write(data)

fobj.close()

#验证码解析

def parse\_img():

url="http://littlebigluo.qicp.net:47720/"

response=requests.request("POST",url,data={"type":"1"},files={'pic\_xxfile':open('imgCode.jpg','rb')})

result=[]

for i in re.findall("<B>(.\*)</B>",response.text)[0].split(" "):

result.append(int(i)-1)

return result

#实现验证码自动点击

def move():

result = parse\_img()

try:

Action=ActionChains(driver)

imgCode=driver.find\_element\_by\_class\_name("imgCode")

coordinate=[[-105,-20],[-35,-20],[40,-20],[110,-20],[-105,50],[-35,50],[40,50],[110,50]]

for i in result:

Action.move\_to\_element(imgCode).move\_by\_offset(coordinate[i][0],coordinate[i][1]).click()

Action.perform()

except Exception as e:

print(e)

def login(userName,password):

try:

driver.get('https://kyfw.12306.cn/otn/resources/login.html')

time.sleep(3)

account=driver.find\_element\_by\_class\_name("login-hd-account")

account.click()

username=driver.find\_element\_by\_id('J-userName')

username.send\_keys(userName)

driver.find\_element\_by\_id("J-password").send\_keys(password,Keys.ENTER)

time.sleep(3)

getImgCode()

move()

driver.find\_element\_by\_class\_name('login-btn').click()

time.sleep(3)

except:

print("error")

**Scrapy核心代码：**

Itcast.py:

class ItcastSpider(scrapy.Spider):

name = 'itcast'

#allowed\_domains = ['itcast.cn']

start\_urls = ['https://movie.douban.com/top250']

i=1

def parse(self, response):

item=MovieItem()

movies=response.xpath('//\*[@id="content"]/div/div[1]/ol/li')

for movie in movies:

item["id"]=self.i

self.i+=1

item["name"]=movie.xpath('div/div[2]/div[1]/a/span[1]/text()').extract\_first()

item["score"]=movie.xpath('div/div[2]/div[2]/div/span[2]/text()').extract\_first()

yield item

nextPage=response.xpath('//\*[@id="content"]/div/div[1]/div[2]/span[3]/a/@href').extract\_first()

if nextPage:

yield scrapy.Request(url = self.start\_urls[0]+nextPage,callback = self.parse)

pipelines.py

class MyspiderPipeline(object):

def process\_item(self, item, spider):

self.insertDB(item)

return item

def insertDB(self,item):

conn=MySQLdb.connect(host='localhost',user='root',passwd='123456',db='scrapy', charset="utf8")

cur=conn.cursor()

sql="insert into movie (id,name,score)values(%s,'%s','%s')" %(item['id'],item['name'],item['score'])

cur.execute(sql)

cur.close()

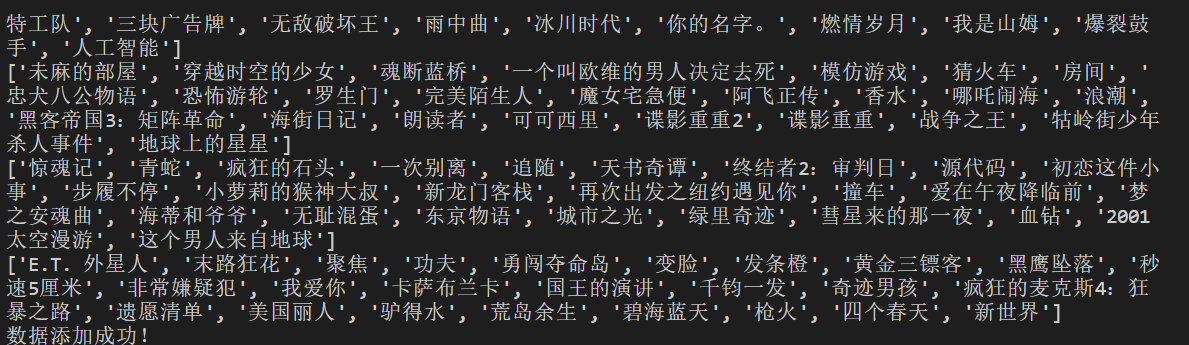
conn.commit()

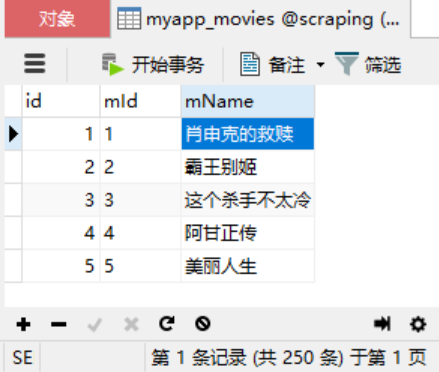
conn.close()

# 结果截图

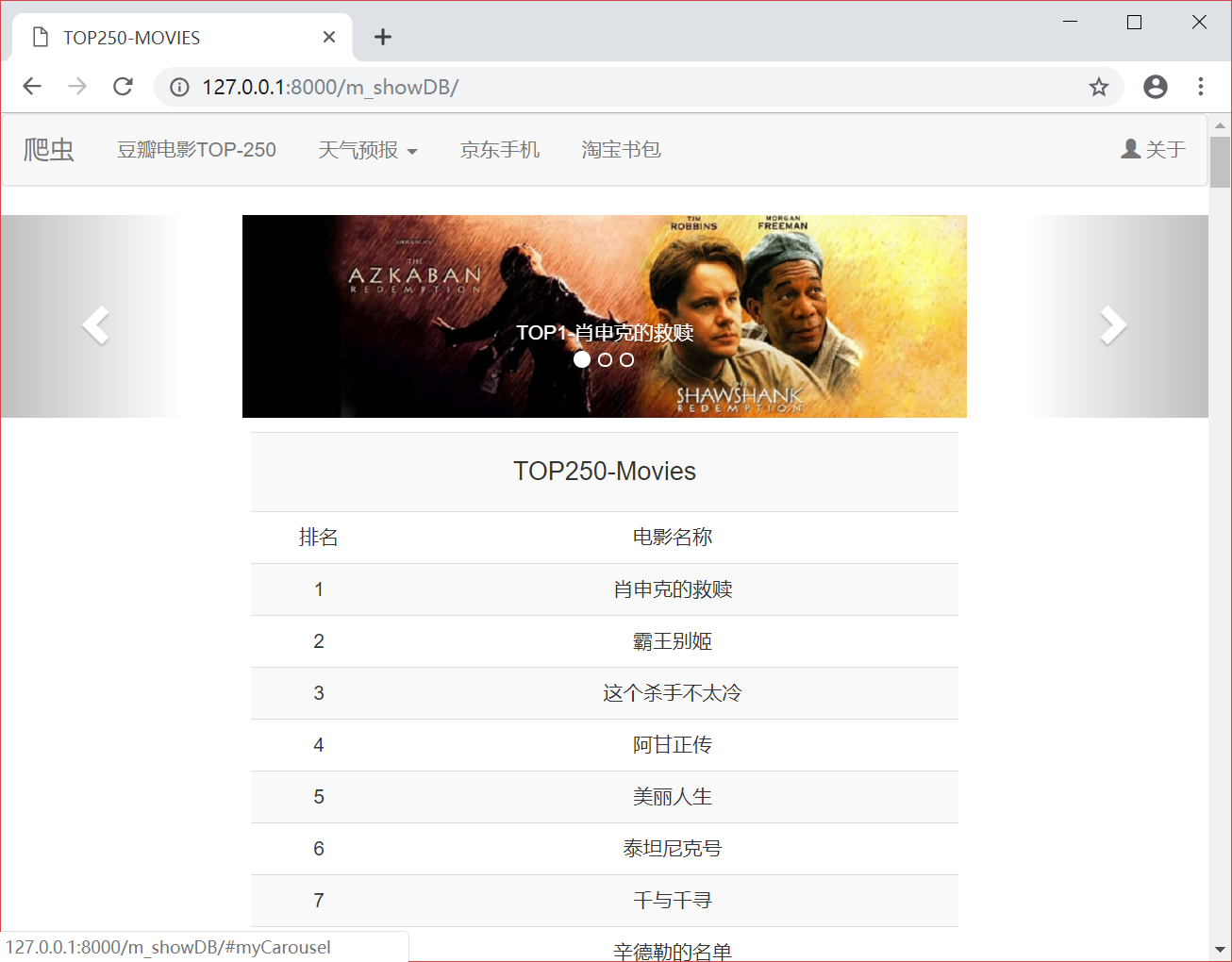
1. **豆瓣Top250电影**

添加数据：



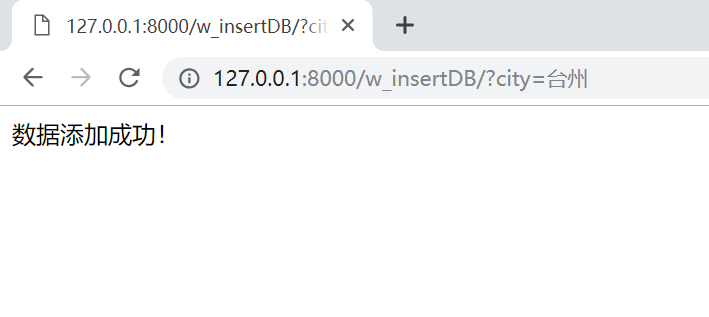


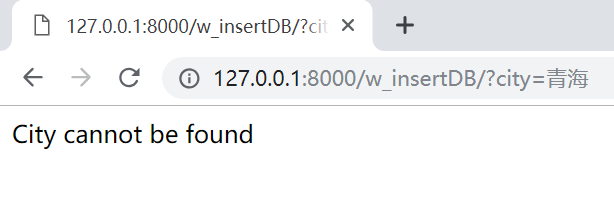
显示数据：



1. **天气预报**

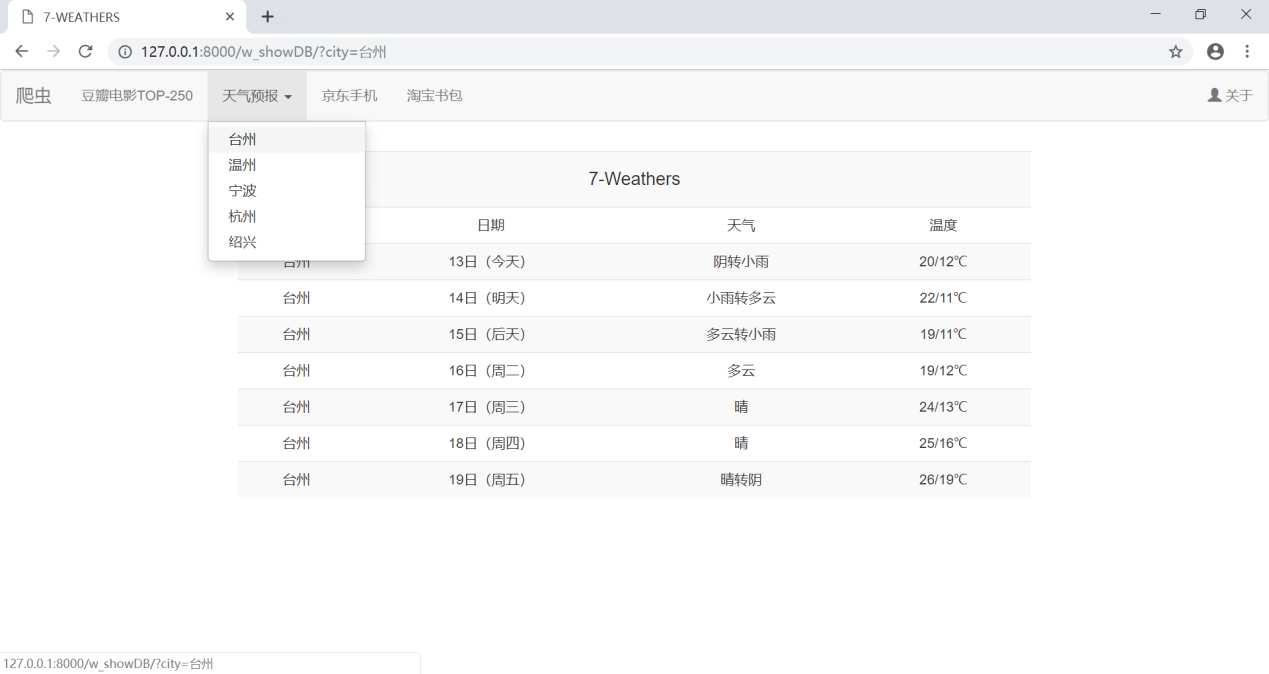
添加数据：

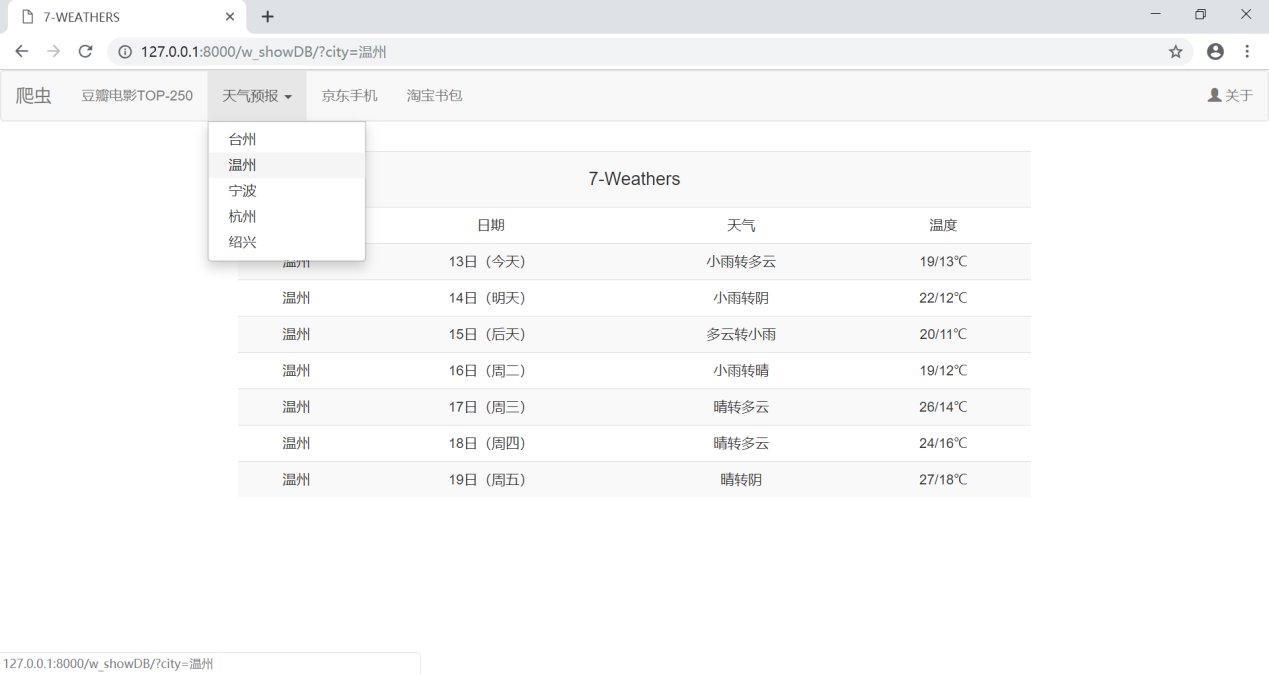




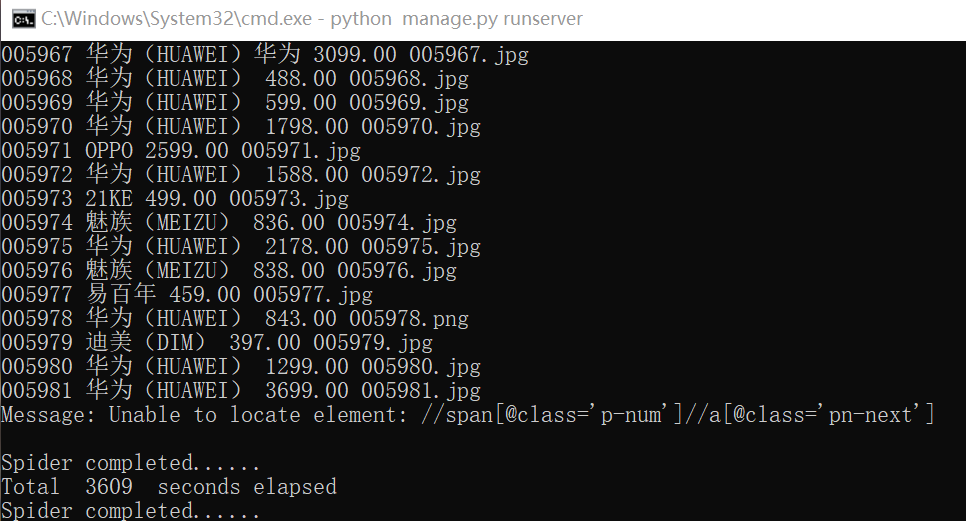
显示数据：

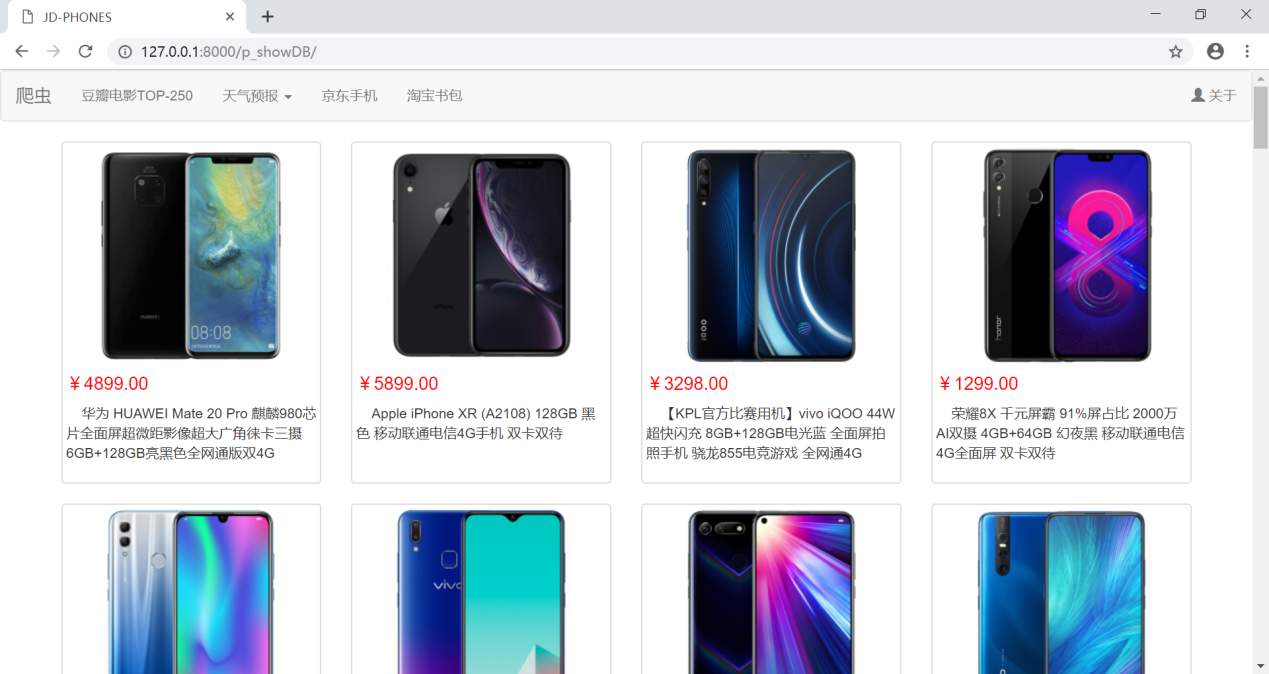


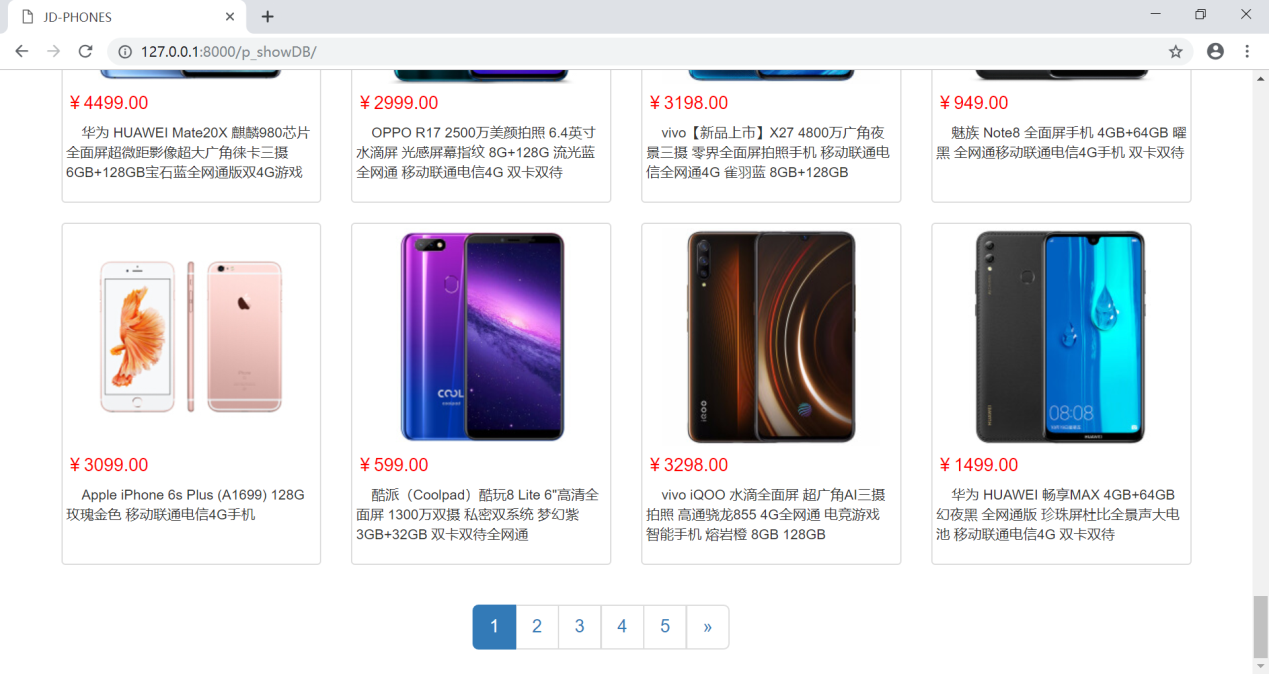




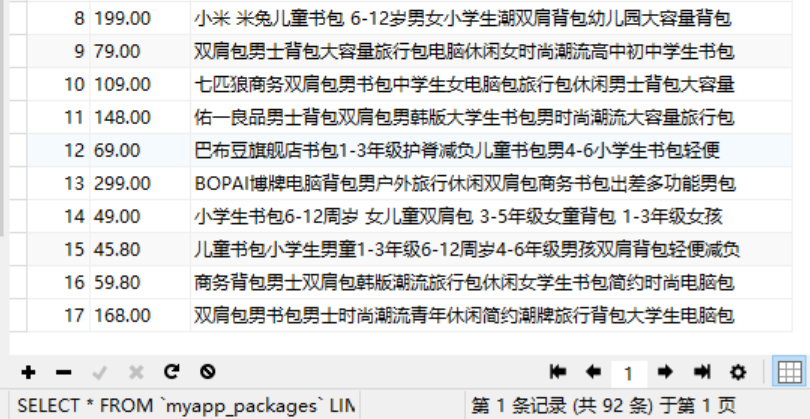
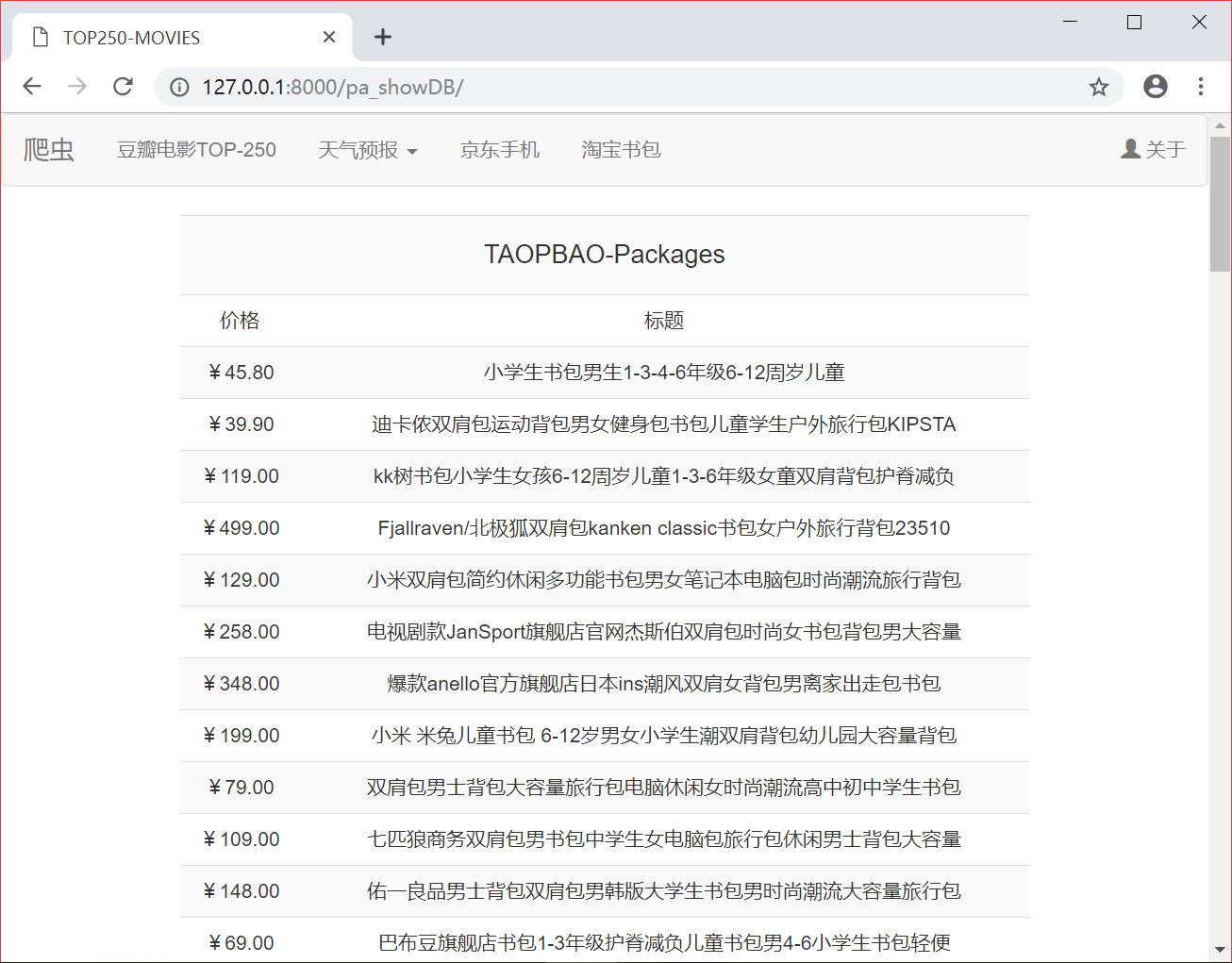
1. **京东手机**







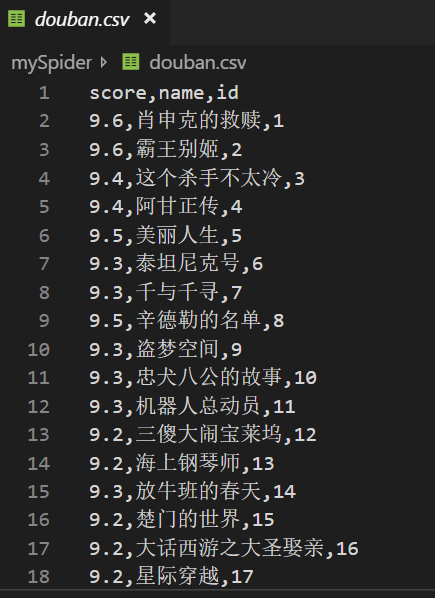
1. **淘宝书包**

1. **12306登录**

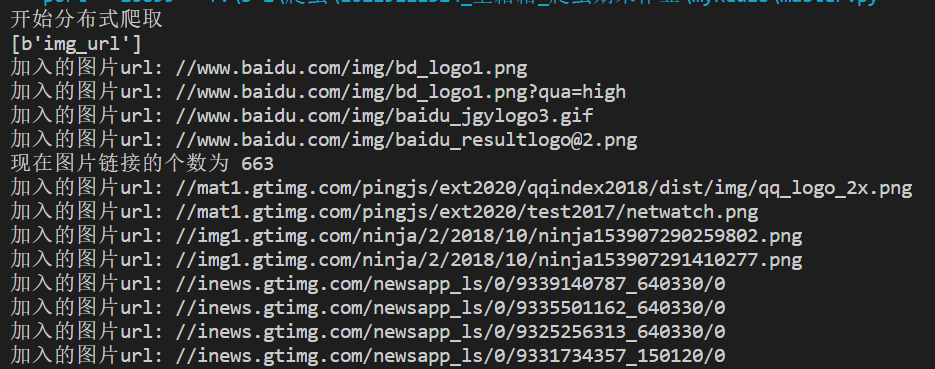


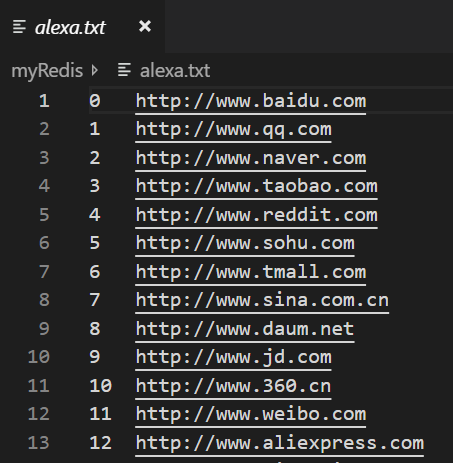
1. **Scrapy豆瓣电影**

****

1. **分布式爬取**

master.py





slave.py



1. **深度优先递归优先爬虫**



1. **广度优先多线程爬虫**

