



温州大学瓯江学院

WENZHOU UNIVERSITY OUJIANG COLLEGE

《爬虫》设计

题 目： 爬虫期末大作业

二级学院： 数学与信息工程学院

班 级： 16 计算机科学与技术三

姓 名： 黄银萍

学 号： 16219111328

完成日期： 2019 年 6 月 20 日

温州大学瓯江学院教务部

二〇一二年十一月制

目录

摘要	1
第 1 章 知识点罗列	2
第 2 章 首页介绍	3
第 3 章 爬取豆瓣 Top250	4
第 4 章 静态爬取城市天气	6
第 5 章 selenium 爬取京东手机	8
第 6 章 淘宝商品信息定向爬虫	11
第 7 章 Scrapy 爬取豆瓣电影	13
第 8 章 selenium 自动登陆 12306	14
第 9 章 分布式爬虫	16
第 10 章 深度优先和广度优先	18
10.1 深度优先的递归爬虫	18
10.2 广度优先的多线程爬虫	19

摘要

网络爬虫是一种按照一定的规则，自动的抓取万维网信息的程序或者脚本。另外一些不常使用的名字还有蚂蚁，自动索引，模拟程序或者蠕虫。

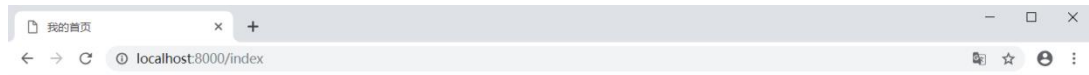
网络搜索功能起源于互联网内容爆炸性发展所带来的对内容检索的需求。搜索引擎不断的发展，人们的需求也在不断的提高，网络信息搜索已经成为人们每天都要进行的内容，如何使搜索引擎能是可满足人们的需求。最初的搜索功能通过索引站的方式实现，而有了网络机器人，及网络爬虫这个技术之后，搜索引擎的时代便开始一发不可收拾了。

第1章 知识点罗列

- 1、静态网页爬取 TOP250 电影数据，并将相关数据存储于 MySQL 中，通过 Django 将数据显示到网页上。
- 2、静态网页爬取北京、上海、广州和深圳一周内的天气数据，并将相关数据存储于 MySQL 中，通过 Django 将数据显示到网页上。
- 3、通过 Selenium 动态网页爬取京东手机，通过 Django 和 bootstrap 结合将数据显示到网页上。
- 4、淘宝商品信息定向爬虫，将数据通过 Django 显示在网页上
- 5、通过 Scrapy 爬取豆瓣 Top250 数据。
- 6、通过 Selenium 与验证码结合自动登陆 12306 网站。通过“反反爬虫”技术成功跳转到 12306 首页。
- 7、客户端分布式爬虫+多个服务器分布式爬虫
- 8、深度优先爬取百度百科和广度优先爬取百度百科

第2章 首页介绍

Django 首页：



我的爬虫项目

爬取豆瓣TOP250数据

爬取京东商城网站数据

爬取天气预报数据

淘宝商品信息定向爬虫

第3章 爬取豆瓣 Top250

此部分爬取了电影的中文名、英文名、别名、导演、主演、上映年份、地区、类型、评分、评价数和电影海报。

```
for li in lis:
    div=li.find("div",attrs={"class":"info"})
    hd=div.find("div",attrs={"class":"hd"})
    spans=hd.find_all("span",attrs={"class":"title"})
    mTitle=spans[0].text.replace("\n","").strip() if len(spans)>0
else ""
    mNative=spans[1].text.replace("\n","").strip() if len(spans)>1
else ""

mNickname=hd.find("span",attrs={"class":"other"}).text.replace("\n","").strip()

    sdiv=li.find("div",attrs={"class":"star"})

mPoint=sdiv.find("span",attrs={"class":"rating_num"}).text.replace("\n","").strip()

mComment=sdiv.find_all("span")[-1].text.replace("\n","").strip()
    bd=div.find("div",attrs={"class":"bd"})
    p=bd.find("p")
    res=self.splitItems(p)
    mDirectors=res[0] if len(res)>0 else ""
    mActors=res[1] if len(res)>1 else ""
    mTime=res[2] if len(res)>2 else ""
    mCountry=res[3] if len(res)>3 else ""
    mType=res[4] if len(res)>4 else ""
    img=li.find("div",attrs={"class":"pic"}).find("img")
    src=urllib.request.urljoin(url,img["src"])
    self.count += 1
```

将爬取到的所有数据存储于 MySQL 中：

对象	testmodel_movie @root (root ...)										
	<div>开始事务</div> <div>备注</div> <div>筛选</div> <div>排序</div> <div>导入</div> <div>导出</div>										
id	mTitle	mNative	mNickname	mDirecors	mActors	mTime	mCountry	mType	mPoint	mComment	mFile
351	肖申克的救 / The Shawsh	月黑高飞(港)	弗兰克·德拉邦特	蒂姆·罗宾斯	Ti 1994		美国	犯罪 剧情	9.6	1401550人评价	https://im
352	霸王别姬		再见，我的妾	陈凯歌	Kaige C 张国荣 Leslie	1993	中国大陆 香港	剧情 爱情 同性	9.6	1038128人评价	https://im
353	这个杀手不太 / Léon	杀手莱昂 / 终极	吕克·贝松	Luc B. 让·雷诺 Jean I	1994		法国	剧情 动作 犯	9.4	1279733人评价	https://im
354	阿甘正传 / Forrest Gun	福雷斯特·冈普	罗伯特·泽米基斯	汤姆·汉克斯	T. 1994		美国	剧情 爱情	9.4	1103796人评价	https://im
355	美丽人生 / La vita è be	一个快乐的传说(罗伯托·贝尼尼	R 罗伯托·贝尼尼	1997		意大利	剧情 喜剧 爱	9.5	646317人评价	https://im
356	泰坦尼克号 / Titanic	铁达尼号(港) / 台	詹姆斯·卡梅隆	J. 莱昂纳多·迪卡	1997		美国	剧情 爱情 灾	9.3	1044273人评价	https://im
357	千与千寻 / 千と千尋の神	神隐少女(台)	宫崎骏	Hayao M 柊瑠美 Rumi I	2001		日本	剧情 动画 奇	9.3	1029533人评价	https://im
358	辛德勒的名 / Schindler's	舒特拉的名单(港)	史蒂文·斯皮尔伯	连姆·尼森	Liar 1993		美国	剧情 历史 战	9.5	576354人评价	https://im
359	盗梦空间 / Inception	潜行凶间(港)	克里斯托弗·诺兰	莱昂纳多·迪卡	2010		美国 英国	剧情 科幻 暴	9.3	1109996人评价	https://im
360	忠犬八公的 / Hachi: A Do	忠犬小八(台)	莱塞·霍尔斯道姆	理查·基尔	Rich 2009		美国 英国	剧情	9.3	731581人评价	https://im
361	机器人总动员 / WALL·E	瓦力(台)	安德鲁·斯坦顿	A 本·贝尔特	Ben 2008		美国	爱情 科幻 动	9.3	735345人评价	https://im
362	三傻大闹宝 / 3 Idiots	三个傻瓜(台)	拉库马·希拉尼	R. 阿米尔·汗	Aan 2009		印度	剧情 喜剧 爱	9.2	996838人评价	https://im
363	海上钢琴师 / La leggenda	声光伴我飞(港)	朱塞佩·托纳多雷	蒂姆·罗斯	Tim 1998		意大利	剧情 音乐	9.2	818150人评价	https://im
364	放牛班的春 / Les choriste	歌声伴我心(港)	克里斯托夫·巴拉	热拉尔·朱尼奥	2004		法国 瑞士 德国	剧情 音乐	9.3	690160人评价	https://im
365	楚门的世界 / The Truman	真人Show(港)	彼得·威尔	Peter 金·凯瑞	Jim C. 1998		美国	剧情 科幻	9.2	762482人评价	https://im
366	大话西游之 / 西遊記大結局	西遊記完結篇(台)	刘镇伟	Jeffrey L 周星驰	Steph 1995		香港 中国大陆	喜剧 爱情 奇	9.2	771266人评价	https://im
367	星际穿越 / Interstellar	星际启示录(港)	克里斯托弗·诺兰	马修·麦康纳	N 2014		美国 英国 加拿	剧情 科幻 冒	9.2	791518人评价	https://im
368	龙猫 / とねりのトトロ	邻居托托罗 / 邻	宫崎骏	Hayao M 日高法子	Nori 1988		日本	动画 奇幻 冒	9.2	681728人评价	https://im
369	教父 / The Godfat	Mario Puzo's T	弗朗西斯·福特·科	马龙·白兰度	M 1972		美国	剧情 犯罪	9.3	500179人评价	https://im
370	熔炉 / 도가니	无声呐喊(港)	黄东赫	Dong-hy. 孔侑	Yoo Gor 2011		韩国	剧情	9.3	446123人评价	https://im
371	无间道 / 無間道	Infernal Affairs	刘伟强 / 麦兆辉	刘德华 / 梁朝	2002		香港	剧情 犯罪 暴	9.1	633194人评价	https://im

+

-

✖

↺

↻


SELECT * FROM `testmodel_movie` LIMIT 0, 1000

第 1 条记录 (共 250 条) 于第 1 页

最后通过 Django+CSS 将数据显示在网页上:

豆瓣电影

localhost:8000/moviedb




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

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 1401550人评价

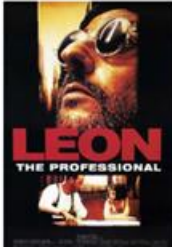


霸王别姬/ 再见，我的妾 / Farewell My Concubine

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

9.6 1038128人评价



这个杀手不太冷/ Léon/ 杀手莱昂 / 终极追杀令(台)

导演: 吕克·贝松 Luc Besson 主演: 让·雷诺 Jean Reno / 娜塔莉·波特曼 ...

1994/法国/剧情 动作 犯罪

9.4 1279733人评价

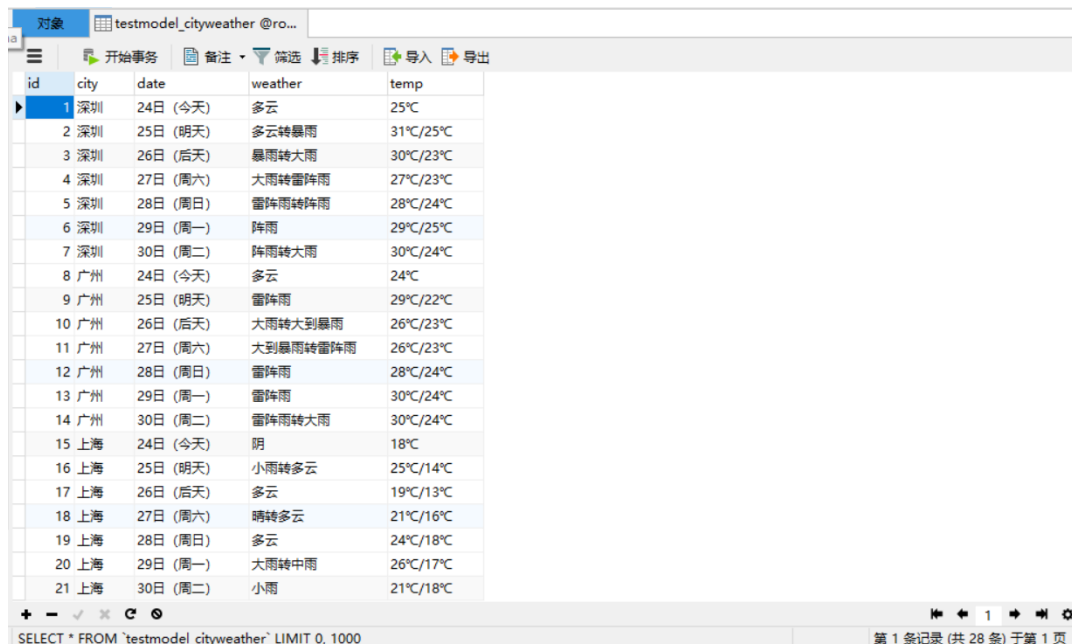
5

第4章 静态爬取城市天气

爬取了北京、上海、广州和深圳的一周的天气数据

```
for li in lis:
    try:
        date=li.select('h1')[0].text
        weather=li.select("p[class='wea']")[0].text
        if n>0:
            temp=li.select("p[class='tem']
span")[0].text+"/"+li.select("p[class='tem'] i")[0].text
        else:
            temp=li.select("p[class='tem'] i")[0].text
        cursor.execute("insert into
testmodel_cityweather(city,date,weather,temp)
values(%s,%s,%s,%s)",(city,date,weather,temp))
        n=n+1
    except Exception as err:
```

将爬取到的数据保存在数据库中:



id	city	date	weather	temp
1	深圳	24日 (今天)	多云	25°C
2	深圳	25日 (明天)	多云转暴雨	31°C/25°C
3	深圳	26日 (后天)	暴雨转大雨	30°C/23°C
4	深圳	27日 (周六)	大雨转雷阵雨	27°C/23°C
5	深圳	28日 (周日)	雷阵雨转阵雨	28°C/24°C
6	深圳	29日 (周一)	阵雨	29°C/25°C
7	深圳	30日 (周二)	阵雨转大雨	30°C/24°C
8	广州	24日 (今天)	多云	24°C
9	广州	25日 (明天)	雷阵雨	29°C/22°C
10	广州	26日 (后天)	大雨转大到暴雨	26°C/23°C
11	广州	27日 (周六)	大到暴雨转雷阵雨	26°C/23°C
12	广州	28日 (周日)	雷阵雨	28°C/24°C
13	广州	29日 (周一)	雷阵雨	30°C/24°C
14	广州	30日 (周二)	雷阵雨转大雨	30°C/24°C
15	上海	24日 (今天)	阴	18°C
16	上海	25日 (明天)	小雨转多云	25°C/14°C
17	上海	26日 (后天)	多云	19°C/13°C
18	上海	27日 (周六)	晴转多云	21°C/16°C
19	上海	28日 (周日)	多云	24°C/18°C
20	上海	29日 (周一)	大雨转中雨	26°C/17°C
21	上海	30日 (周二)	小雨	21°C/18°C

数据库中的数据最终通过 Django 显示于网页

城市	日期	天气	温度
深圳	24日 (今天)	多云	25°C
深圳	25日 (明天)	多云转暴雨	31°C/25°C
深圳	26日 (后天)	暴雨转大雨	30°C/23°C
深圳	27日 (周六)	大雨转雷阵雨	27°C/23°C
深圳	28日 (周日)	雷阵雨转阵雨	28°C/24°C
深圳	29日 (周一)	阵雨	29°C/25°C
深圳	30日 (周二)	阵雨转大雨	30°C/24°C
广州	24日 (今天)	多云	24°C
广州	25日 (明天)	雷阵雨	29°C/22°C
广州	26日 (后天)	大雨转大到暴雨	26°C/23°C
广州	27日 (周六)	大到暴雨转雷阵雨	26°C/23°C
广州	28日 (周日)	雷阵雨	28°C/24°C
广州	29日 (周一)	雷阵雨	30°C/24°C
广州	30日 (周二)	雷阵雨转大雨	30°C/24°C
上海	24日 (今天)	阴	18°C
上海	25日 (明天)	小雨转多云	25°C/14°C
上海	26日 (后天)	多云	19°C/13°C
上海	27日 (周六)	晴转多云	21°C/16°C
上海	28日 (周日)	多云	24°C/18°C
上海	29日 (周一)	大雨转中雨	26°C/17°C
上海	30日 (周二)	小雨	21°C/18°C
北京	24日 (今天)	小雨	6°C
北京	25日 (明天)	多云	18°C/6°C
北京	26日 (后天)	晴转多云	21°C/11°C
北京	27日 (周六)	小雨	18°C/8°C
北京	28日 (周日)	多云	22°C/10°C
北京	29日 (周一)	多云转小雨	25°C/13°C
北京	30日 (周二)	多云	26°C/14°C

第5章 selenium 爬取京东手机

通过动态页面爬取京东的全部手机信息，包括：名称、具体内容、价格和海报。

```
        for li in lis:
            try:
                src1 =
li.find_element_by_xpath("../../../div[@class='p-img']/a/img").get_attribute("src")
            except:
                src1=""
            try:
                src2 =
li.find_element_by_xpath("../../../div[@class='p-img']/a/img").get_attribute("data-lazy-img")
            except:
                src2=""
            try:
                price =
li.find_element_by_xpath("../../../div[@class='p-price']/i").text
            except:
                price="0"
            try:
                note = li.find_element_by_xpath("../../../div[@class='p-name p-name-type-2']/em").text
                mark = note.split(" ")[0]
                mark = mark.replace("爱心东东\n", "")
                mark = mark.replace(", ", "")
                note = note.replace("爱心东东\n", "")
                note = note.replace(", ", "")
            except:
                note=""
                mark=""
            self.No = self.No + 1
            no = str(self.No)
            while len(no) < 6:
                no = "0" + no
            print(no,mark,price)
            if src1:
                src1=urllib.request.urljoin(self.driver.current_url,src1)
                p = src1.rfind(".")
```

```

        mFile = no + src1[p:]
    elif src2:
        src2=urllib.request.urljoin(self.driver.current_url,src2)
        p = src2.rfind(".")
        mFile = no + src2[p:]
    if src1 or src2:
        T = threading.Thread(target=self.download,
args=(src1,src2,mFile))
        T.setDaemon(False)
        T.start()
        self.threads.append(T)
    else:
        mFile = ""
        self.insertDB(no, mark, price, note, mFile)
try:
self.driver.find_element_by_xpath("//span[@class='p-num']/a[@class='pn-next
-disabled']")
except:
    nextPage =
self.driver.find_element_by_xpath("//span[@class='p-num']/a[@class='pn-next
']")
    nextPage.click()
    self.processSpider()

```

将数据存储于数据库中，可以查看到一共获取到 4185 条数据

对象

testmodel_phones @root (r...

开始事务

备注

筛选

排序

导入

导出

id	mNo	mMark	mPrice	mNote	mFile
1	000001	【预售】魅族	3198.00	【预售】魅族 16s 骁龙855: 000001.jpg	
2	000002	Apple	5698.00	Apple iPhone XR (A2108) 000002.jpg	
3	000003	【KPL官方比赛用机】vivo	3298.00	【KPL官方比赛用机】 vivo i 000003.jpg	
4	000004	华为	3988.00	华为 HUAWEI P30 超感光# 000004.jpg	
5	000005	荣耀8X	1299.00	荣耀8X 千元屏霸 91%屏占比 000005.jpg	
6	000006	小米	1199.00	小米 红米Redmi Note7 幻 000006.jpg	
7	000007	荣耀10青春版	1299.00	荣耀10青春版 幻彩渐变 24C 000007.jpg	
8	000008	vivo	799.00	vivo U1 水滴全面屏 AI智慧 000008.jpg	
9	000009	联想Z6	2999.00	联想Z6 Pro 8GB+128GB # 000009.jpg	
10	000010	小米	799.00	小米 红米6 4GB+64GB 铂 000010.jpg	
11	000011	荣耀V20	2799.00	荣耀V20 胡歌同款 麒麟980 000011.jpg	
12	000012	荣耀畅玩8C两天一充	899.00	荣耀畅玩8C两天一充 莱赛尔 000012.jpg	
13	000013	小米8SE	1399.00	小米8SE 全面屏智能游戏拍! 000013.jpg	
14	000014	小米9	3299.00	小米9 4800万超广角三摄 8 000014.jpg	
15	000015	小米8青春版	1499.00	小米8青春版 镜面渐变AI双摄 000015.jpg	
16	000016	vivo	1598.00	vivo Z3 6GB+64GB 极光蓝 000016.jpg	
17	000017	三星	6999.00	三星 Galaxy S10+ 8GB+12 000017.jpg	
18	000018	小米	799.00	小米 红米Redmi 7 AI双摄: 000018.jpg	
19	000019	Apple	6199.00	Apple iPhone X (A1865) 6 000019.jpg	
20	000020	vivo	3598.00	vivo X27 8GB+256GB大内 000020.jpg	
21	000021	小米	649.00	小米 红米6A AI美颜 3GB+ 000021.jpg	

+ - × ↺ ↻


SELECT * FROM `testmodel_phones` LIMIT 0, 1000

第 1 条记录 (共 1000 条)

Django 结合 bootstrap 通过分页将数据显示于网页中：


京东手机 × +

localhost:8000/phonesdb




¥3198.00

【预售】 魅族 16s 骁龙855全金属拍照游戏手机, 6GB+128GB 暗夜黑 全网通移动联通电信4G 双卡双待




¥5698.00

Apple iPhone XR (A2108) 128GB 黑色 移动联通电信4G手机, 双卡双待




¥3298.00

【XPL官方比赛手机】 vivo X27 Pro 44W超快充电 8GB+128GB电竞蓝 全网通拍照手机, 骁龙855电竞版 全网通4G




¥3888.00

华为 HUAWEI P30 超感光镜头+三摄麒麟800AI智能芯片全网通屏内指纹手机, 8GB+64GB亮黑色全网通4G双卡




¥1299.00

荣耀X 千元屏霸 91%屏占比 2000万AI双摄 4GB+64GB 记录黑 移动联通电信4G全网通 双卡双待




¥1199.00

小米 Redmi Note 7 5G版骁龙710 4GB+64GB 梦幻蓝 全网通4G 双卡双待 全网全网通拍照游戏智能




¥1299.00

荣耀10青春版 全网通 2400万AI自拍 全网通4GB+64GB 梦幻蓝 移动联通电信4G全网通 双卡双待




¥799.00

vivo U1 3.0英寸屏 AI智能拍照手机, 3GB+32GB 阳光色 移动联通电信4G




¥2999.00

荣耀20 Pro 8GB+128GB 黑色 骁龙855 4800万AI双摄 4000mAh大电池 P-CPU双摄双摄像头 游戏 全网通4G 双卡双待




¥799.00

小米 红米6 4GB+64GB 魅影灰 全网通4G手机, 双卡双待




¥2799.00

荣耀V20 骁龙855 麒麟810芯片 魅蓝全网通 4800万双摄手机, 6GB+128GB 记录黑 移动联通电信4G全网通




¥899.00

荣耀8X Pro 全网通 全网通4GB+32GB 记录黑 移动联通电信4G全网通 双卡双待




¥1399.00

小米9SE 全网通智能拍照游戏手机, 6GB+64GB 灰色 骁龙710处理器 全网通4G 双卡双待




¥3299.00

小米8 4800万超广角三摄 8GB+128GB 金色 麒麟810 全网通4G 双卡双待 全网全网通拍照游戏智能




¥1499.00

小米9 Pro 5G版 骁龙855 AI双摄 6GB+64GB 梦幻蓝 全网通4G 双卡双待 全网全网通拍照游戏智能




¥599.00

荣耀 畅玩7 2GB+16GB 金色 全网通4G手机, 双卡双待




¥1398.00

荣耀 Note9 全网通智能拍照手机, 4GB+64GB 记录黑 全网通移动联通电信4G 双卡双待




¥3688.00

华为 HUAWEI P30 Pro 全网通三摄拍照手机, 6GB+128GB 亮黑色 全网通移动联通电信4G 双卡双待




¥899.00

HUAWEI 华为畅享3 3GB+32GB 全网通 全网通AI长续航 全网通4G 移动联通电信4G




¥2399.00

OPPO R17 2000万像素超广角 4.8英寸高清屏 光感屏指纹 6G+128GB 全网通 全网通移动联通电信4G 双卡双待




¥2999.00

Apple iPhone 8s Plus (A1899) 128G 玫瑰金色 移动联通电信4G手机




¥1399.00

OPPO K1 全网通智能拍照 全网通拍照手机, 4GB+64GB 墨晶黑 全网通 移动联通电信4G 双卡双待




¥4499.00

华为 HUAWEI Mate20X 麒麟980芯片全网通超广角摄像头超广角半三摄8GB+128GB全网通双卡双待4G双卡双待



¥2499.00

小米8 Pro 5G版 8GB+128GB 黑色 全网通4G 双卡双待 全网全网通拍照游戏智能手机



¥949.00

荣耀 N10 Pro 全网通手机, 4GB+64GB 曜黑 全网通移动联通电信4G手机, 双卡双待

上一页 1 2 3 4 5 6 7 8 9 10 下一页

第6章 淘宝商品信息定向爬虫

爬取了淘宝“书包”的价格和名称

```
def getHTMLText(url):
    try:
        r=requests.get(url,timeout=30)
        r.raise_for_status()
        r.encoding=r.apparent_encoding
        return r.text
    except:
        return ""

def parsePage(ilt,html):
    try:
        plt=re.findall(r'"view_price"\:("[\d\.]*)"',html)
        tlt=re.findall(r'"raw_title"\:("[\.\*?\"]*)"',html)
        for i in range(len(plt)):
            price=eval(plt[i].split(':')[1])
            title=eval(tlt[i].split(':')[1])
            ilt.append([price,title])
            cursor.execute("insert into testmodel_taobao(title,price)
values(%s,%s)",(title,price))
    except:
        print("")

def printGoodsList(ilt):
    tplt="{:4}\t{:8}\t{:16}"
    print(tplt.format("序号","价格","商品名称"))
    count=0
    for g in ilt:
        count=count+1
        print(tplt.format(count,g[0],g[1]))
```

我们可以看到 testmodel_taobao 表中共爬取了 92 条信息

对象 testmodel_taobao @root (r...		
<div> <div>开始事务</div> <div>备注</div> <div>筛选</div> <div>排序</div> <div>导入</div> <div>导出</div> </div>		
id	price	title
1	45.80	小学生书包男生1-3-4-6年级6-12周岁儿童
2	39.90	迪卡侬双肩包运动背包男女健身包书包儿童学生户外旅行包KIPSTA
3	119.00	kk树书包小学生女孩6-12周岁儿童1-3-6年级女童双肩背包护脊减负
4	499.00	Fjallraven/北极狐双肩包kanken classic书包女户外旅行背包23510
5	129.00	小米双肩包简约休闲多功能书包男女笔记本电脑包时尚潮流旅行背包
6	258.00	电视剧款JanSport旗舰店官网杰斯伯双肩包时尚女书包背包男大容量
7	348.00	爆款anello官方旗舰店日本ins潮流双肩女背包男离家出走包书包
8	199.00	小米 米兔儿童书包 6-12岁男女小学生潮双肩背包幼儿园大容量背包
9	79.00	双肩包男士背包大容量旅行包电脑休闲女时尚潮流高中初中学生书包
10	109.00	七匹狼商务双肩包男书包中学生女电脑包旅行包休闲男士背包大容量
11	148.00	佑一良品男士背包双肩包男韩版大学生书包男时尚潮流大容量旅行包
12	69.00	巴布豆旗舰店书包1-3年级护脊减负儿童书包男4-6小学生书包轻便
13	299.00	BOPAI博牌电脑背包男户外旅行休闲双肩包商务书包出差多功能男包
14	49.00	小学生书包6-12周岁 女童双肩包 3-5年级女童背包 1-3年级女孩
15	45.80	儿童书包小学生男童1-3年级6-12周岁4-6年级男孩双肩背包轻便减负

+

-

✓

✕

↺

↻

⏮

⏪

1

⏩

⏭

⚙

SELECT * FROM `testmodel_taobao` LIMIT 0, 1000

第 1 条记录 (共 92 条) 于第 1 页

Django 页面展示图如下：

淘宝书包信息	
标题：小学生书包男生1-3-4-6年级6-12周岁儿童	价格：45.80
标题：迪卡侬双肩包运动背包男女健身包书包儿童学生户外旅行包KIPSTA	价格：39.90
标题：kk树书包小学生女孩6-12周岁儿童1-3-6年级女童双肩背包护脊减负	价格：119.00
标题：Fjallraven/北极狐双肩包kanken classic书包女户外旅行背包23510	价格：499.00
标题：小米双肩包简约休闲多功能书包男女笔记本电脑包时尚潮流旅行背包	价格：129.00
标题：电视剧款JanSport旗舰店官网杰斯伯双肩包时尚女书包背包男大容量	价格：258.00

第7章 Scrapy 爬取豆瓣电影

此处爬取了电影的名称、评分、评分人数和排名

```
def parse(self, response):
    item = DoubanItem()
    movies = response.xpath('//ol[@class="grid_view"]/li')
    for movie in movies:
        item['ranking'] =
movie.xpath('.//div[@class="pic"]/em/text()').extract()[0]
        item['movie_name'] =
movie.xpath('.//div[@class="hd"]/a/span[1]/text()').extract()[0]
        item['score'] =
movie.xpath('.//div[@class="star"]/span[@class="rating_num"]/text()').extract()[0]
        item['score_num'] =
movie.xpath('.//div[@class="star"]/span/text()').extract()[0]
        yield item

    next_url = response.xpath('//span[@class="next"]/a/@href').extract()
    if next_url:
        next_url = 'https://movie.douban.com/top250' + next_url[0]
        yield Request(next_url, headers=self.headers)
```

最后将爬取的数据存储在 movie.csv 中

score	score_num	ranking	movie_name
9.6	9.6	1	肖申克的救赎
9.6	9.6	2	霸王别姬
9.4	9.4	3	这个杀手不太冷
9.4	9.4	4	阿甘正传
9.5	9.5	5	美丽人生
9.3	9.3	6	泰坦尼克号
9.3	9.3	7	千与千寻
9.5	9.5	8	辛德勒的名单
9.3	9.3	9	盗梦空间
9.3	9.3	10	忠犬八公的故事
9.3	9.3	11	机器人总动员
9.2	9.2	12	三傻大闹宝莱坞
9.2	9.2	13	海上钢琴师
9.3	9.3	14	放牛班的春天

第8章 selenium 自动登陆 12306

从页面获取到验证码图片：

```
def getVerifyImage(self):
    try:

        img_element =WebDriverWait(self.driver, 100).until(
            EC.presence_of_element_located((By.ID, "J-loginImg"))
        )

    except Exception as e:
        print("网络开小差,请稍后尝试")
        base64_str=img_element.get_attribute("src").split(",")[-1]
        imgdata=base64.b64decode(base64_str)
        with open('verify.jpg','wb') as file:
            file.write(imgdata)
        self.img_element=img_element
```

然后通过上传图片查找出正确的物品

```
def getVerifyResult(self):
    url="http://littlebigluo.qicp.net:47720/"
    response=requests.request("POST",url,data={"type":"1"},files={'pic_x
xfile':open('verify.jpg','rb')})
    result=[]
    print(response.text)
    for i in re.findall("<B>(.*?)</B>",response.text)[0].split(" "):
        result.append(int(i)-1)
    self.result=result
    print(result)
```


请上传一张12306验证码图片

未选择任何文件

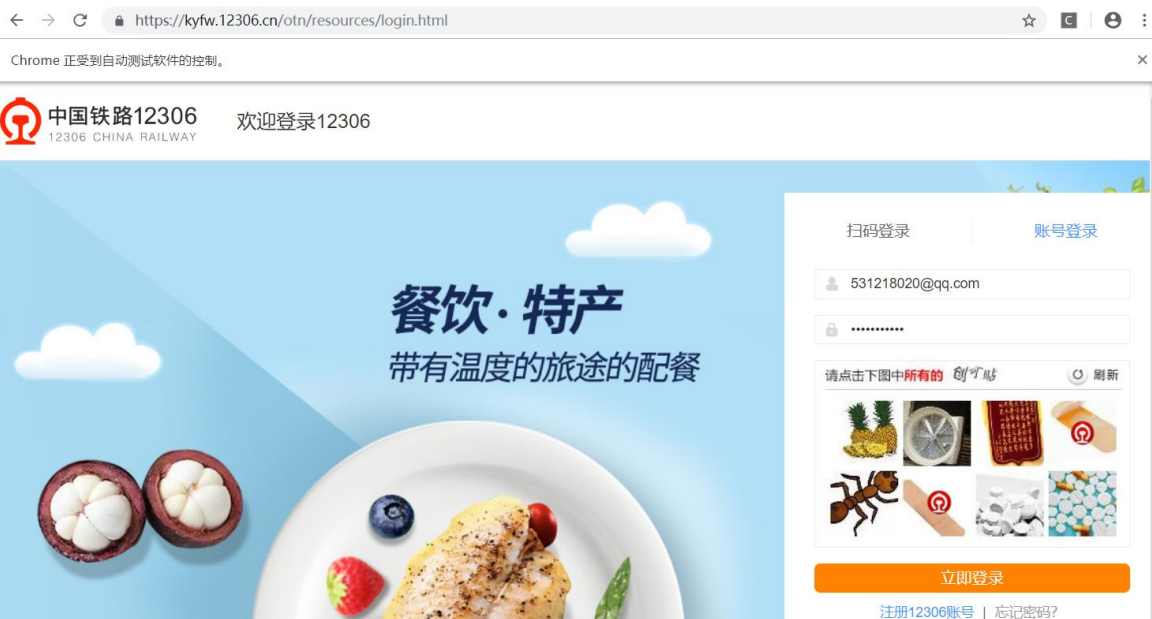
使用方法：

- 1-打开12306网站登录界面：[点击这里打开12306](#)
- 2-点击12306页面顶部**登录**按钮，然后点击**账号登陆**，鼠标右键点击**页面中间验证码图片**
- 3-选择**图片另存为**保存验证码图片，并重命名以.jpg结尾
- 4-然后点击本页面**选择文件**按钮选择刚刚保存的图片
- 5-然后点击本页面**上传**按钮查看结果

上传非标准12306图片验证码文件，本系统会拒绝连接

本破解基于深度学习算法实现：[点击这里查看详情](#)

有意见或建议？？欢迎交流：3490699170@qq.com



第9章 分布式爬虫

master 主人与 slave 奴隶的区别:

1、 master 主人

```
if __name__ == '__main__':  
    this_machine='master'  
    print('开始分布式爬虫')  
    if this_machine=='master':  
        push_redis_list()  
    else:  
        get_img()
```

```
PS F:\大三下\爬虫> cd 'f:\大三下\爬虫'; ${env:PYTHONIOENCODING}='UTF-8'; ${env:PYTHONIOENCODING}='UTF-8'  
ns\ms-python.python-2019.4.12954\pythonFiles\ptvsd_launcher.py' '--default' '--cli  
开始分布式爬虫  
[b'img url']  
加入的图片url: //www.baidu.com/img/bd_logo1.png  
加入的图片url: //www.baidu.com/img/bd_logo1.png?qua=high  
加入的图片url: //www.baidu.com/img/baidu_jgylogo3.gif  
加入的图片url: //www.baidu.com/img/baidu_resultlogo@2.png  
现在图片链接的个数为 855  
加入的图片url: //mat1.gtimg.com/pingjs/ext2020/qqindex2018/dist/img/qq_logo_2x.png  
加入的图片url: //mat1.gtimg.com/pingjs/ext2020/test2017/netwatch.png  
加入的图片url: //img1.gtimg.com/ninja/2/2018/10/ninja153907290259802.png  
加入的图片url: //img1.gtimg.com/ninja/2/2018/10/ninja153907291410277.png  
加入的图片url: //inews.gtimg.com/newsapp_ls/0/9024800937_640330/0  
加入的图片url: //inews.gtimg.com/newsapp_ls/0/9041252153_640330/0  
加入的图片url: //img1.gtimg.com/ninja/2/2019/05/ninja155840091740722.jpg
```

2、 slave 奴隶

```
if __name__ == '__main__':  
    this_machine='slave'  
    print('开始分布式爬虫')  
    if this_machine == 'master':  
        push_redis_list()  
    else:  
        get_img()
```



第10章 深度优先和广度优先

以下均爬取百度百科

10.1 深度优先的递归爬虫

```
def scrappy(url,depth=1):
    global g_writecount
    try:
        headers={'User-Agent':'Mozilla/5.0 (Windows;U;Windows NT 6.1;en-US;rv:1.9.1.6) Gecko/20091201 Firefox/3.5.6'}
        r=requests.get("https://baike.baidu.com/"+url,headers=headers)
        html=r.content.decode("utf-8")
    except Exception as e:
        print('Failed downloading and saving',url)
        print(e)
        exist_url.append(url)
        return None

    exist_url.append(url)
    if(depth==1):
        link_list=re.findall('<a href="/fenlei/([^:#=<>]*?)".*?</a>',html)
    else:
        link_list=re.findall('<a href="/([^:#=<>]*?)".*?</a>',html)
    unique_list=list(set(link_list)-set(exist_url))

    for eachone in unique_list:
        g_writecount+=1
        output="No."+str(g_writecount)+"\t Depth:"+str(depth)+"\t"+url+' --> '+eachone+'\n'
        print(output)
        with open('title.txt',"a+",encoding="utf-8") as f:
            f.write(output)
            f.close()
        if depth<2:
            scrappy("fenlei/"+eachone,depth+1)
```

我们可以在 title.txt 中查看顺利获取的数据，如图：

```
No. 1      Depth:1      --> 电子产品
No. 2      Depth:2      fenlei/电子产品 --> fenlei/%E6%88%BF%E5%9C%B0%E4%BA%A7
No. 3      Depth:2      fenlei/电子产品 --> fenlei/%E8%83%BD%E6%BA%90
No. 4      Depth:2      fenlei/电子产品 --> fenlei/%E5%8F%B0%E5%8C%97
No. 5      Depth:2      fenlei/电子产品 --> fenlei/%E5%BC%80%E6%94%BE
No. 6      Depth:2      fenlei/电子产品 --> fenlei/OpenGL
No. 7      Depth:2      fenlei/电子产品 --> fenlei/%E6%8A%80%E6%9C%AF
No. 8      Depth:2      fenlei/电子产品 --> fenlei/%E6%92%AD%E6%94%BE%E5%99%A8
No. 9      Depth:2      fenlei/电子产品 --> fenlei/%E7%BD%91%E7%BB%9C
No. 10     Depth:2      fenlei/电子产品 --> fenlei/%E5%9B%BD%E4%BA%A7%E6%89%8B%E6%9C
No. 11     Depth:2      fenlei/电子产品 --> view/1225332.htm
No. 12     Depth:2      fenlei/电子产品 --> fenlei/%E5%8F%AF%E7%88%B1
No. 13     Depth:2      fenlei/电子产品 --> fenlei/MMORPG
No. 14     Depth:2      fenlei/电子产品 --> fenlei/%E9%80%9A%E8%AE%AF
No. 15     Depth:2      fenlei/电子产品 --> view/1154965.htm
No. 16     Depth:2      fenlei/电子产品 --> renwu
No. 17     Depth:2      fenlei/电子产品 --> yishu
No. 18     Depth:2      fenlei/电子产品 --> fenlei/%E7%94%B5%E6%B1%A0
No. 19     Depth:2      fenlei/电子产品 --> lishi
No. 20     Depth:2      fenlei/电子产品 --> view/3163404.htm
No. 21     Depth:2      fenlei/电子产品 --> view/344201.htm
No. 22     Depth:2      fenlei/电子产品 --> fenlei/%E4%B8%AD%E5%9B%BD%E7%94%B5%E4%BF
No. 23     Depth:2      fenlei/电子产品 --> jingji
No. 24     Depth:2      fenlei/电子产品 --> subview/8055576/7964003.htm
No. 25     Depth:2      fenlei/电子产品 --> view/220766.htm
No. 26     Depth:2      fenlei/电子产品 --> fenlei/%E5%93%81%E7%89%8C
No. 27     Depth:2      fenlei/电子产品 --> shehui
No. 28     Depth:2      fenlei/电子产品 --> view/171667.htm
No. 29     Depth:2      fenlei/电子产品 --> view/1224493.htm
No. 30     Depth:2      fenlei/电子产品 --> fenlei/%E4%BC%A0%E6%84%9F%E5%99%A8
No. 31     Depth:2      fenlei/电子产品 --> fenlei/%E5%8D%95%E7%89%87%E6%9C%BA
No. 32     Depth:2      fenlei/电子产品 --> view/42387.htm
No. 33     Depth:2      fenlei/电子产品 --> view/1440081.htm
No. 34     Depth:2      fenlei/电子产品 --> view/1053362.htm
No. 35     Depth:2      fenlei/电子产品 --> subview/1191984/5735305.htm
```

最终获取的 URL 数量为 4426 个。

10.2 广度优先的多线程爬虫

```
def run(self):
    global g_mutex
    global g_writecount
    try:
        print(self.tid,"crawl",self.url)
        headers={'User-Agent':'Mozilla/5.0 (Windows;U;Windows NT 6.1;en-US;
rv:1.9.1.6) Gecko/20091201 Firefox/3.5.6'}
```

```

r=requests.get("https://baike.baidu.com/"+self.url,headers=headers)
    r.encoding="utf-8"
    html=r.text

    link_list2=re.findall('<a href="/([^:#=<>]*?)".*?</a>',html)
    unique_list2=list(set(link_list2))
    for eachone in unique_list2:
        g_writecount+=1
        content2="No."+str(g_writecount)+"\t
Thread"+str(self.tid)+"\t"+self.url+' --> '+eachone+'\n'
        with open('title2.txt',"a+",encoding='utf-8') as f:
            print(content2)
            f.write(content2)
            f.close()
    except Exception as e:
        g_mutex.acquire()
        g_existURL.append(self.url)
        g_mutex.release()
        print('Failed downloading and saving',self.url)
        print(e)
        return None
g_mutex.acquire()
g_pages.append(html)
g_existURL.append(self.url)
g_mutex.release()

```

我们可以在 title2.txt 中查看顺利获取的数据，如图：

No. 1	Thread0	-->
No. 2	Thread0	--> fenlei/植物
No. 3	Thread0	--> fenlei/自然灾害
No. 4	Thread0	--> fenlei/建筑
No. 5	Thread0	--> keji
No. 6	Thread0	--> fenlei/体育设施
No. 7	Thread0	--> fenlei/自然资源
No. 8	Thread0	--> fenlei/文化人物
No. 9	Thread0	--> fenlei/军事
No. 10	Thread0	--> fenlei/历史事件
No. 11	Thread0	--> tiyu
No. 12	Thread0	--> shenghuo
No. 13	Thread0	--> fenlei/自然现象
No. 14	Thread0	--> fenlei/体育奖项
No. 15	Thread0	--> ziran
No. 16	Thread0	--> fenlei/政治
No. 17	Thread0	--> task
No. 18	Thread0	--> city/
No. 19	Thread0	--> fenlei/戏剧
No. 20	Thread0	--> fenlei/民族
No. 21	Thread0	--> fenlei/时尚
No. 22	Thread0	--> item/秒懂星课堂
No. 23	Thread0	--> task/
No. 24	Thread0	--> fenlei/互联网
No. 25	Thread0	--> calendar
No. 26	Thread0	--> shehui
No. 27	Thread0	--> kedou/
No. 28	Thread0	--> fenlei/体育组织
No. 29	Thread0	--> usercenter
No. 30	Thread0	--> fenlei/经济
No. 31	Thread0	--> fenlei/经济人物
No. 32	Thread0	--> renwu
No. 33	Thread0	--> fenlei/电子产品
No. 34	Thread0	--> calendar/
No. 35	Thread0	--> fenlei/舞蹈
No. 36	Thread0	--> item/秒懂五千年
No. 37	Thread0	--> art
No. 38	Thread0	--> item/秒懂大师说
No. 39	Thread0	--> fenlei/历史人物
No. 40	Thread0	--> fenlei/地形地貌

最终获取的 URL 数量为 4929 个，花费的时间为 79 秒