爬虫期中作业：

姓名：李雪萌

班级:16 计算机科学与技术三班

学号：16219111337

项目内容：

Django 项目+爬虫代码+数据库+实验报告

涉及知识：

Python 语言的基本语法以及第三方库的使用；

静态网页的爬取;

网页的解析；

动态网页的爬取；

模拟无头浏览器爬取动态网页；

爬虫与反爬虫

涉及其他：

正则表达式

Django 的 **mvc** 模型

Selenium 的安装调用

豆瓣爬取：

```python
import requests
from lxml import etree
import MySQLdb


con= MySQLdb.connect(host='localhost',user='root',password='root',db='test',charset="utf8")
cursor=con.cursor()
cursor.execute("create table test(No varchar(32) ,content char(255))")

#静态网页抓取：request爬虫实践：top250电影数据
def get_page(start_num):
    url='https://movie.douban.com/top250?start=%s&filter=' %start_num
    print(url)

    response=requests.get(url)
    tree=etree.HTML(response.text)
    title=tree.xpath('//span[@class="title"][1]/text()')
    return title

def get_all_page(start,end):
    result=[]
    for i in range(start,end-start):
        title_list=get_page(i*25)
        result+=title_list
        print(result)
    return result

if __name__=="__main__":
    result=get_all_page(0,10)

    for i in range(len(result)):
        cursor.execute("INSERT INTO test(No,content) values(%d,'%s')" %(i+1,result[i]))
    cursor.close()
    con.commit()
    con.close()
```

天气爬取：

```python
from bs4 import BeautifulSoup
from bs4 import UnicodeDammit
import urllib.request
import MySQLdb

class weatherDB:
    def openDB(self):
        self.con= MySQLdb.connect(host='localhost',user='root',password='root',db='test',charset="utf8")
        self.cursor=self.con.cursor()
        try:
            self.cursor.execute("create table weathers1(wCity varchar(16),wDate varchar(16),wWeather varchar(64),wTemp varchar(32),constraint pk_we
        except:
            self.cursor.execute("delete from weathers")


    def closeDB(self):
        self.con.commit()
        self.con.close()


    def insert(self,city,date,weather,temp):
        try:
            self.cursor.execute("insert into weathers1(wCity,wDate,wWeather,wTemp) values(?,?,?,?)",(city,date,weather,temp))
        except Exception as err:
            print(err)


    def show(self):
        self.cursor.execute("select * from weathers1")
        rows=self.cursor.fetchall()
        print("%-16s%-16s%-32s%-16s"%("city","date","weather","temp"))
        for row in rows:
            print("%-16s%-16s%-32s%-16s"%(row[0],row[1],row[2],row[3]))


class weatherForecast:
    def __init__(self):
        self.headers={"User-Agent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103 Safari/53
        self.cityCode={"北京":"101010100","上海":"101020100","广州":"101280101","深圳":"101280601"}

    def forecastCity(self,city):
        if city not in self.cityCode.keys():
            print(city+"code cannot be found")
            return
```

```python
        url="http://www.weather.com.cn/weather/"+self.cityCode[city]+".shtml"
        try:
            req=urllib.request.Request(url,headers=self.headers)
            data=urllib.request.urlopen(req)
            data=data.read()
            dammit=UnicodeDammit(data,["utf-8","gbk"])
            data=dammit.unicode_markup
            soup=BeautifulSoup(data,"lxml")
            lis=soup.select("ul[class='t clearfix'] li")
            for li in lis:
                try:
                    date=li.select('h1')[0].text
                    weather=li.select('p[class="wea"]')[0].text
                    temp=li.select('p[class="tem"]   span')[0].text+"/"+li.select('p[class="tem"] i')[0].text
                    print(city,data,weather,temp)
                    self.db.insert(city,date,weather,temp)
                except Exception as err:
                    print(err)
        except Exception as err:
            print(err)

    def process(self,cities):
        self.db=weatherDB()
        self.db.openDB()
        for city in cities:
            self.forecastCity(city)

        #self.db.show()
        self.db.closeDB()

ws=weatherForecast()
ws.process(["北京","上海","深圳","广州"])
print("completed")
```

京东爬取：

```
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
import urllib.request
import threading
import MySQLdb
import os
import datetime



class MySpider:
    header={"User-Agent: Mozilla/5 (Windows NT 10 ; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36"}
    imagePath="download"

    def startUp(self,url,key):
        try:
            self.con=MySQLdb.connect(host='localhost',user='root',password='root',db='test',charset="utf8")
            self.cursor=self.con.cursor()
            try:
                self.cursor.execute("drop table phones")

            except:
                pass
            try:
                sql="create table phones(mNo varchae(32) primary key,mMark varchar(256),mPrice varchar(32),mNote varchar(1024),mFile varchar(256))"
                self.cursor.execute(sql)
            except:
                pass
        except Exception as err:
            print(err)

        chrome_option=Options()
        chrome_option.add_argument('--headless')
        chrome_option.add_argument('--disable-gpu')

        self.driver=webdriver.Chrome(chrome_options=chrome_option)
        self.threads=[]
        self.No=0
        self.imgNo=0


        try:
            if not  os.path.exists(MySpider.imagePath):
                os.mkdir(MySpider.imagePath)
            images=os.listdir(MySpider.imagePath)
            for img in images:
```

```
            for img in images:
                s=os.path.join(MySpider.imagePath,img)
                os.remove(s)

        except Exception as err:
            print(err)
        try:
            if not os.path.exists(MySpider.imagePath):
                os.mkdir(MySpider)
            images=os.listdir(MySpider.imagePath)
            for img in images:
                s=os.path.join(MySpider.imagePath,img)
                os.remove(s)
        except Exception as err:
            print(err)
        self.driver.get(url)
        keyInput=self.driver.find_element_by_id("key")
        keyInput.send_keys(key)
        keyInput.send_keys(Keys.ENTER)

    def closeUp(self):
        try:
            self.con.commit()
            self.con.close()
            self.driver.close()
        except Exception as err:
            print(err)

    def insertDB(self,mNo,mMark,mPrice,mNote,mFile):
        try:
            sql="insert into phones(mNo,mMark,mPrice,mNote,mFile)values(?,?,?,?,?)"
            self.cursor.execute(sql,(mNo,mMark,mPrice,mNote,mFile))
        except Exception as err:
            print(err)

    def showDB(self):
        try:
            con= MySQLdb.connect(host='localhost',user='root',password='root',db='test',charset="utf8")
            cursor=con.cursor()
            print("%-8s %-16s %-8s %-16s %s"%("No","Mark","Price","Image","Note"))
            cursor.execute("select mNo,mMark,mPrice,mNote from phones order by mNo")
            rows=cursor.fetchall()
            for row in rows:
                print("%-8s %-16s %-8s %-16s %s"%(row[0],row[1],row[2],row[3],row[4]))
            con.close()
```

```
        except Exception as err:
            print(err)
    def download(self,src1,src2,mFile):
        data=None
        if src1:
            try:
                req=urllib.request.Request(src1,headers=MySpider.headers)
                resp=urllib.request.urlopen(req,timeout=400)
                data=resp.read()
            except:
                pass
        if not data and src2:
            try:
                req=urllib.request.Request(src2,headers=MySpider.headers)
                resp=urllib.request.urlopen(req,timeout=400)
                data=resp.read()
            except:
                pass
        if data:
            fobj=open(MySpider.imagePath+"\\"+mFile,"wb")
            fobj.write(data)
            fobj.close()
            print("download",mFile)

    def processSpider(self):
        try:
            time.sleep(10)
            print(self.driver.current_url)
            lis=self.driver.find_element_by_xpath("//div[@id='J_goodsList']//li[@class='gl-item']")
            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:
                    price="0"
                try:
                    note=li.find_element_by_xpath(".//div[@class='p-price']//i").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(",","")
                    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.urljion(self.driver.current_url,src1)
                    p=src1.rfind(".")
                    mFile=no+src1[p:]
                elif src2:
                    src2=urllib.request.urljion(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.setDeamon(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()
        except Exception as err:
            print(err)

    def executeSpider(self,url,key):
        starttime=datetime.datetime.now()
        print("Spider starting.....")
        self.startUp(url,key)
        self.processSpider()
        self.closeUp()
        for t in self.threads:
            t.join()
        print("Spider completed......")
        endtime=datetime.datetime.now()
        elapsed=(endtime-starttime).seconds
        print("Total",elapsed,"second elapsed")
```

```
            t.join()
        print("Spider completed......")
        endtime=datetime.datetime.now()
        elapsed=(endtime-starttime).seconds
        print("Total",elapsed,"second elapsed")

url='http://www.jd.com'
spider=MySpider()
while True:
    print("1.爬取")
    print("2.显示")
    print("3.退出")
    s=input("请输入选择（1, 2, 3）:")
    if s=="1":
        spider.executeSpider(url,"手机")
    elif s=="2":
        spider.showDB()
    elif s=="3":
        break
```

淘宝爬取：

```
import requests
import re
import MySQLdb


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])
    except:
        print("")


def printGoodList(ilt):
    con= MySQLdb.connect(host='localhost',user='root',password='root',db='test',charset="utf8")
    cursor=con.cursor()
    cursor.execute("select * from shubao")
    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]))
        cursor.execute("insert into shubao(id,Price,Title) values('"+count+"','"+g[0]+"','"+g[1]+"')")
    cursor.close()
    con.commit()
    con.close()


def main():
    goods='书包'
    depth=3
```

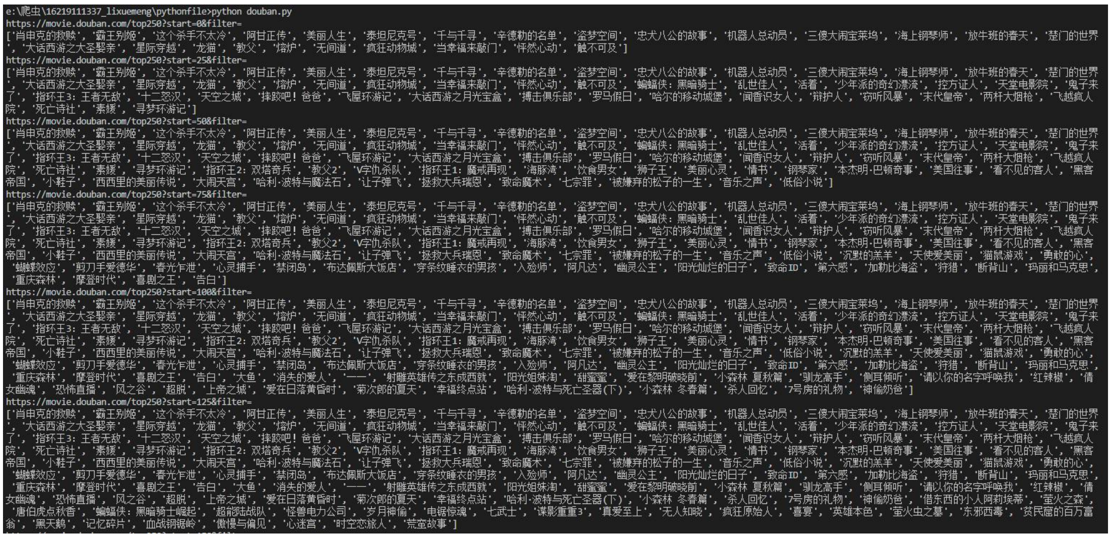
```
    depth=3
    start_url='https://s.taobao.com/search?q='+goods
    infoList=[]
    for i in range(depth):
        try:
            url=start_url+'&s='+str(44*i)
            html=getHTMLText(url)
            parsePage(infoList,html)
        except:
            continue
    printGoodList(infoList)


main()
```

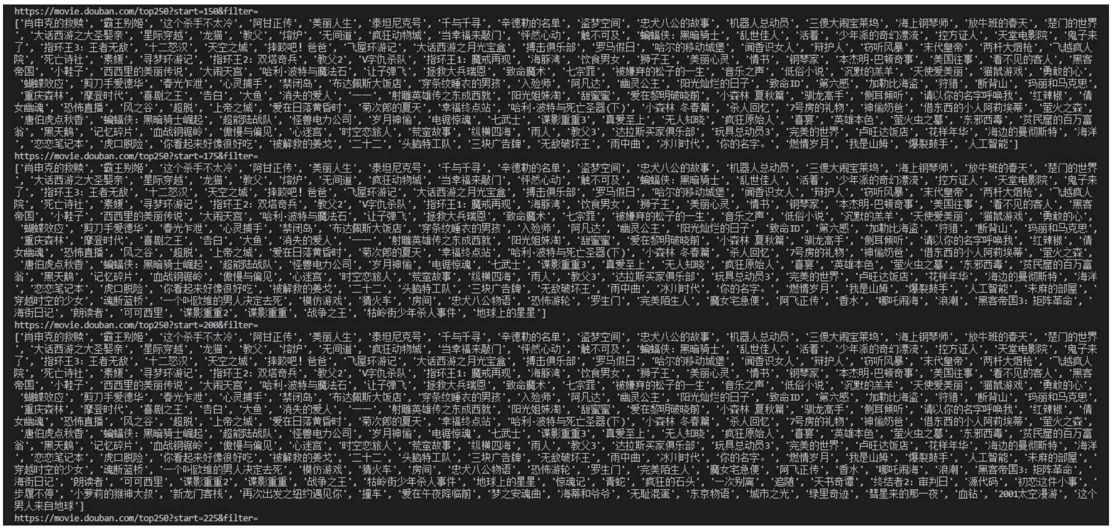
爬取结果展示：

```
e:\爬虫\16219111337_lixuemeng\pythonfile>python tq.py
list index out of range
北京 26日（明天） 晴转多云 21℃10℃
北京 27日（后天） 多云 18℃7℃
北京 28日（周日） 多云 21℃9℃
北京 29日（周一） 多云转小雨 24℃13℃
北京 30日（周二） 多云 26℃14℃
北京 1日（周三） 多云 24℃14℃
list index out of range
上海 26日（明天） 多云转晴 18℃12℃
上海 27日（后天） 多云 18℃14℃
上海 28日（周日） 小雨转多云 23℃17℃
上海 29日（周一） 中雨转阴 24℃17℃
上海 30日（周二） 阴转小雨 22℃17℃
上海 1日（周三） 小雨转多云 23℃17℃
list index out of range
广州 26日（明天） 中雨转中到大雨 28℃24℃
广州 27日（后天） 中到大雨转雷阵雨 28℃24℃
广州 28日（周日） 雷阵雨 29℃25℃
广州 29日（周一） 雷阵雨转中雨 30℃25℃
广州 30日（周二） 中雨转大到暴雨 30℃22℃
广州 1日（周三） 大到暴雨转多云 26℃20℃
list index out of range
深圳 26日（明天） 中雨转大雨 29℃24℃
深圳 27日（后天） 大雨转雷阵雨 27℃23℃
深圳 28日（周日） 雷阵雨 28℃24℃
深圳 29日（周一） 雷阵雨 30℃25℃
深圳 30日（周二） 雷阵雨转暴雨 30℃23℃
深圳 1日（周三） 暴雨转阵雨 27℃23℃
完成!
```

47	289.00	牛津布双肩包女2019新款韩版时尚尼龙书包电脑包双肩旅行防盗背包
48	69.00	双肩包女背包2018新款韩版潮牛津布帆布时尚百搭书包旅行小包2019
49	69.00	双肩包女背包2018新款韩版潮牛津布帆布时尚百搭书包旅行小包2019
50	59.00	男士背包电脑旅游休闲韩版时尚潮流高中学生书包大容量旅行双肩包
51	59.90	绝地求生三级包吃鸡装备包男双肩包制激战场高中初中小学生书包
52	228.00	anello双肩包女日本乐天ins超火旅行男背包书包离家出走包alleno
53	149.00	鳄鱼男士双肩包大容量商务休闲电脑背包旅行简约学生时尚潮流书包
54	169.00	鳄鱼男士双肩包大容量旅行简约电脑背包商务休闲学生时尚潮流书包
55	699.00	鳄鱼男士双肩包真皮大容量商务休闲青年电脑背包时尚潮流牛皮书包
56	109.00	七匹狼双肩包男 新款商务男士旅行背包女休闲大容量电脑学生书包
57	139.00	赫曼尔双肩包男士潮流背包男韩版旅行包时尚休闲学生书包电脑包潮
58	129.00	MIUI/小米小米双肩包简约休闲多功能书包男女笔记本电脑包背包
59	139.00	uek小学生书包男孩女生1-3-6年级护脊双肩背包6-12岁轻便儿童书包
60	228.00	anello双肩包女妈妈咪包日本乐天背包男帆布书包离家出走包alleno
61	29.00	定期改完旅行背包休闲双肩包女初中小学生中学生书包男士时尚潮流
62	159.00	双肩包男士背包大容量17寸旅行包时尚潮流电脑包商务多功能书包男
63	149.00	七匹狼双肩包男士背包女商务电脑包旅行包中学生书包时尚潮流帆布
64	158.00	专柜正品JanSport杰斯伯经典双肩背包男女同款学生书包T501纯色
65	89.00	稻草人休闲双肩包女简约韩版中学生书包大容量旅行背包男时尚潮流
66	116.00	香港正品Doughnut甜甜圈双肩包学生书包旅行包电脑双肩背包女书包
67	149.00	双肩包男潮牌休闲男士背包15.6寸电脑包旅行包学生书包男时尚潮流
68	298.00	CK双肩包男时尚潮流背包大学生书包休闲简约牛津布旅行包新品
69	79.00	日本乐天sun earth双肩包女2019新款书包男男包旅行商家出走包
70	49.00	双肩包男潮简约大容量休闲旅行背包男时尚潮流书包女高初中学生包
71	139.00	追德双肩包男士大容量休闲电脑包韩版学生书包旅行包时尚潮流背包
72	358.00	香港潮牌2019新款女生真皮大号旅行书包双肩包羊皮拼接多用背包女
73	142.00	七匹狼商务双肩包男士背包女韩版中学生书包休闲男包旅行包电脑包
74	158.00	休闲双肩包男简约韩版时尚潮流大学生书包大容量电脑旅行男士背包
75	119.00	双肩背包男休闲多功能男士商务15.6寸双肩电脑包。时尚潮流女书包
76	189.00	七匹狼双肩包男士潮流背包时尚中学生商务大容量休闲旅行电脑书包
77	249.00	苹天2019新款双肩包女帆布大容量商务电脑包牛津布旅行背包女书包
78	124.00	男士双肩包韩版大学生书包男大容量运动背包男时尚潮流电脑旅行包
79	69.90	追卡侬官网户外双肩包登山旅行包男书包初中高中学生休闲背包QUBP
80	49.90	古着感少女书包女韩版原宿ulzzang大学生森系百搭高中双肩包ins潮
81	169.00	瑞士军刀双肩包瑞士大容量初中生书包男中学生运动旅游旅行背包女
82	29.00	小米双肩包米家小背包男女通用运动包日常休闲双肩包学生书包
83	168.00	FAl童拉杆书包小学生女孩6-12周岁大童1-3-5-6年级防水爬坡拖拉
84	129.00	米熙背包男双肩包男士大容量学生书包休闲商务电脑包女旅游旅行包
85	138.00	尼龙背包男简约休闲旅行包电脑双肩包时尚潮流男包大学生书包潮牌
86	228.00	anello双肩包女日本乐天ins超火帆布背包男书包离家出走包alleno
87	25.89	束口袋抽绳双肩包男女小学生书包轻便运动布袋简易补习背包补课包
88	380.00	Samsonite/新秀丽双肩包女小包 商务电脑包男书包休闲旅行背包BP2
89	68.00	迪士尼书包小学生1-3-4二四年级男童女童米奇减负轻便儿童双肩包6
90	399.00	FJCN/菲驼潮新款双肩包旅行包 女士拉链印花背包青年防水书包小包
91	274.62	上海迪士尼包包达菲熊书包雪梨纹背包帆布包星黛露双肩包莫妮同款
92	59.00	背包男双肩包旅行包户外轻便旅游行李包休闲时尚大容量登山书包
93	98.00	休闲背包男双肩包时尚潮流青年男士旅行大学生书包大容量电脑包潮
94	188.00	书包小学生男童1-3-4-6年级牛津大学儿童6-12周岁一减负护脊女孩5
95	168.00	卡帝乐鳄鱼正品休闲时尚双肩包男书包女大容量旅行背包学生电脑包
96	89.00	超火双肩包女2019新款旅游旅行电脑背包大学生书包韩版高中ins风
97	19.90	追卡侬旗舰店双肩包新款男书包旅行迷你运动小背包轻便女包QUBP


98 69.00 双肩包女2018新款韩版潮百搭时尚书包女士pu软皮小背包女包包2019
99 79.00 中学生书包女双肩包2019新款潮韩版原宿ulzzang 高中简约初中背包
100 209.00 莱夫书包女韩版时尚双肩包小帆布包百搭尼龙2019新款潮牛津布背包
101 49.00 双肩包女2019新款韩版潮牛津布小背包女士尼龙百搭时尚帆布书包包
102 79.90 森马双肩包女韩版 休闲校园简约学院背包潮旅行包高中学生书包男
103 168.00 正品冠军champion双肩包女韩版学生书包时尚潮牌背包帆布旅行包男
104 139.00 King and kiss背包男双肩包时尚潮流皮质书包个性韩版百搭简约
105 169.00 瑞士军刀双肩包瑞士中学生书包休闲女商务男士旅行大容量电脑背包
106 69.00 酷仔王子儿童拉杆书包3-5-6年级女童6-12周岁小学生书包三轮拖拉
107 59.00 书包男中学生初中生高中生小学男生背包青少年校园双肩包时尚潮流
108 59.00 背包男韩版时尚潮流商务双肩包旅行休闲男士初中高中学生书包
109 128.00 双肩包男时尚潮流个性街头背包休闲韩版旅行包简约高中大学生书包
110 148.00 旅行双肩包男时尚潮流休闲大学生书包电脑大容量简约轻便男士背包
111 258.00 JanSport旗舰店官网杰斯伯双肩包女学生书包纯色男士女士背包T501
112 698.00 Herschel Supply Retreat经典色时尚潮流男女双肩包书包背包10066
113 38.00 小学生书包男孩1-3-4-6年级双肩减负护脊儿童书包男轻便男童背包
114 1.50 运动防水抽绳双肩包拉绳培训学生补习班袋子 印字logo 束口袋定制
115 169.90 迪卡侬运动双肩包电脑背包书包男女健身包旅行包休闲商务KIPSTA
116 148.00 电脑双肩包男时尚潮流学生书包运动旅行轻便休闲简约商务男士背包
117 118.00 日本CILOCALA双肩包女妈咪包旅行背包果冻包亲子包儿童书包情侣包
118 15.90 帆布包女2019新款单肩ins中包大学生上课书包韩版百搭斜挎手提袋
119 298.00 CK正品双肩包男时尚潮流背包大学生书包休闲防水男包韩版旅行包男
120 148.00 香港正品Doughnut甜甜圈双肩包新款韩版旅行包背包男女学生书包
121 158.00 裂狼休闲双肩包男士韩版潮流皮背包男时尚学生书包旅行包电脑男包
122 59.90 绝地求生三级包吃鸡3级背包男双肩包刺激战场高中初中小学生书包
123 35.90 幼儿园书包男女孩1-3-6岁宝宝可爱卡通韩版潮儿童小书包双肩背包5
124 49.00 书包小学生6-12周岁 可爱公主双肩包3-5年级女童背包 1-3年级女孩
125 99.00 三级包吃鸡3级背包男绝地求生同款双肩包时尚潮流初中小学生书包
126 318.00 香港IT双肩包女士2019新款韩版潮背包软皮休闲时尚旅行大容量书包
127 52.00 LEQUEEN防水牛津布学生双肩包书包旅行背包大容量男女离家出走包
128 118.00 随歌绝地求生同款三级包吃鸡3级背包男双肩包高中初中小学生书包
129 169.00 瑞士军刀瑞戈双肩背包男瑞士学生书包女大容量商务休闲电脑旅行包
130 100.80 儿童小学生书包女孩拉杆书包1-3-5-6年级6-12周岁三轮大轮爬楼梯
131 69.00 迪士尼书包小学生男童4-6年级1-3-5三四五六女双肩包儿童米奇书包
132 179.00 迪德双肩包男时尚潮流背包男士旅行包韩版休闲青年学生书包电脑包
133 936.00 美国正品代购tory burch女包TB真皮双肩包2017新款书包旅行背包
134 39.00 ins超火的双肩包女韩版背包学院高中学生书包男电脑旅行包大容量
135 29.90 双肩包女士韩版2019新款时尚pu软皮潮包百搭休闲背包女大容量书包
136 800.00 新款正品 coach蔻驰男包真皮男士商务背包休闲旅行书包电脑双肩包


e:\爬虫\16219111337_lixuemeng\pythonfile>python douban.py
https://movie.douban.com/top250?start=0&filter=
['肖申克的救赎', '霸王别姬', '这个杀手不太冷', '阿甘正传', '美丽人生', '泰坦尼克号', '千与千寻', '辛德勒的名单', '盗梦空间', '忠犬八公的故事', '机器人总动员', '三傻大闹宝莱坞', '海上钢琴师', '放牛班的春天', '楚门的世界', '大话西游之大圣娶亲', '星际穿越', '龙猫', '教父', '熔炉', '无间道', '疯狂动物城', '当幸福来敲门', '怦然心动', '触不可及']
https://movie.douban.com/top250?start=25&filter=
['肖申克的救赎', '霸王别姬', '这个杀手不太冷', '阿甘正传', '美丽人生', '泰坦尼克号', '千与千寻', '辛德勒的名单', '盗梦空间', '忠犬八公的故事', '机器人总动员', '三傻大闹宝莱坞', '海上钢琴师', '放牛班的春天', '楚门的世界', '大话西游之大圣娶亲', '星际穿越', '龙猫', '教父', '熔炉', '无间道', '疯狂动物城', '当幸福来敲门', '怦然心动', '触不可及', '蝙蝠侠：黑暗骑士', '乱世佳人', '活着', '少年派的奇幻漂流', '控方证人', '天堂电影院', '鬼子来了', '指环王3：王者无敌', '十二怒汉', '天空之城', '摔跤吧！爸爸', '飞屋环游记', '大话西游之月光宝盒', '搏击俱乐部', '罗马假日', '哈尔的移动城堡', '闻香识女人', '辩护人', '窃听风暴', '末代皇帝', '两杆大烟枪', '飞越疯人院', '死亡诗社', '素媛', '寻梦环游记']
https://movie.douban.com/top250?start=50&filter=
https://movie.douban.com/top250?start=75&filter=
https://movie.douban.com/top250?start=100&filter=
https://movie.douban.com/top250?start=125&filter=


https://movie.douban.com/top250?start=150&filter=
https://movie.douban.com/top250?start=175&filter=
https://movie.douban.com/top250?start=200&filter=
https://movie.douban.com/top250?start=225&filter=

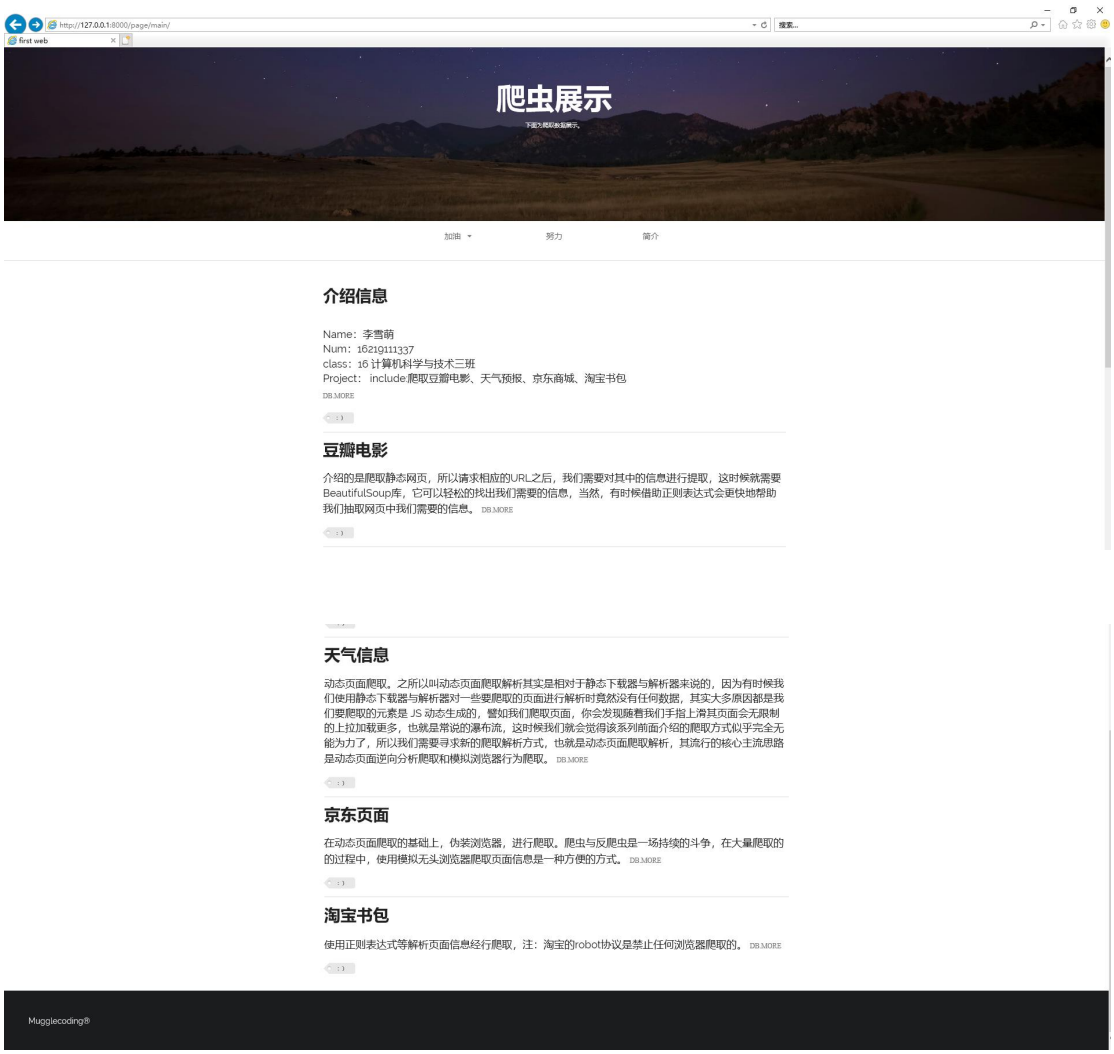
网站思想：

1.运通 mvc 框架（简单用 mvt）

2.Template 模板页用于显示 view 层的信息

3.Modle 层是 django 特有的数据库形式，以类的方式引用

4.页面使用 html+css（semantic ui）+简单 js+django 特有编辑语言



代码：

web：

```
<!DOCTYPE html>
{% load staticfiles %}
<html>
    <head>
        <meta charset="utf-8">
        <title>first web</title>
        <link rel="stylesheet" href="{% static 'css/semantic.css'%}"  media="screen" title="no title" charset="utf-8">
        <link href="https://fonts.googleapis.com/css?family=Oswald|Raleway" rel="stylesheet">


        <style type="text/css">
            h1 {
                font-family:'Oswald', sans-serif!important;
                font-size:40px;
            }

            body {
                font-family: 'Raleway', sans-serif;
            }
            p {
                font-family: 'Raleway', sans-serif;
                font-size:18px;
            }
            .ui.vertical.segment.masthead {
                height: 300px;
                background-image: url("{% static 'images/star_banner.jpg'%}");
                background-size: cover;
                background-position: 100% 80%;
            }

            .ui.container.segment {
                width: 800px;
            }

            .ui.center.aligned.header.blogslogon {
                margin-top: 40px;
            }

            .ui.center.aligned.header.blogslogon p {
                margin-top: 10px;
                color: □white;
                font-size: 10px;
            }
            .ui.container.nav {
                width: 500px;
            }
```

```
        </style>
        {% block css %}{% endblock%}

    </head>
    <body>
        <div class="ui inverted vertical  segment masthead">

            <h1 class="ui center aligned header blogslogon" style="font-size:50px;font-family: 'Raleway', sans-serif!important;">
                爬虫展示
                <p class="ui sub header">
                    下面为爬取数据展示。
                </p>

            </h1>
        </div>
        <div class="ui container nav">
            <div class="ui borderless text three item menu ">
                <div class="ui simple dropdown  item">
                    加油
                    <i class="dropdown icon"></i>
                    <div class="menu">
                        <a class="item" href="">简介</a>
                        <a class="item" href="">简介</a>
                    </div>
                </div>
                <a class="item">
                    努力
                </a>
                <a class="item">
                    简介
                </a>

            </div>
        </div>

        <div class="ui divider"></div>

        {% block header %}{% endblock%}

        <div class="ui  vertical segment">
            <div class="ui container vertical segment">
                <a href="#">
                    <h1 class="ui header">
                        介绍信息
                    </h1>
                </a>
```

```
        </a>

        <i class="icon grey small unhide"></i>
        <p>
            <form >
                <div  class="six wide column">
                    <p><label >Name: 李雪萌</label></p>
                </div>
                <div >
                    <p><label >Num: 16219111337</label></p>
                </div>
                <div >
                    <p><label >class: 16 计算机科学与技术三班</label></p>
                </div>
                <div >
                    <p>
                        <label>
                        Project:
                            include:爬取豆瓣电影、天气预报、京东商城、淘宝书包
                        </label>
                    </p>
                </div>
            </form>
            <a href="#">
                <i class="angle tiny double grey right icon">DB.MORE</i>
            </a>
        </p>

        <div class="ui mini  tag label">
            : )
        </div>
    </div>
    <div class="ui container vertical segment">
        <a href="#">
            <h1 class="ui header">
                豆瓣电影
            </h1>
        </a>

        <i class="icon grey small unhide"></i>
        <p>
            介绍的是爬取静态网页，所以请求相应的URL之后，我们需要对其中的信息进行提取，这时候就需要BeautifulSoup库，它可以轻松的找出我们需要的信息
            <a href="http://127.0.0.1:8000/page/db">
                <i class="angle tiny double grey right icon">DB.MORE</i>
            </a>
        </p>
```

```
        </p>

        <div class="ui mini  tag label">
            : )
        </div>
    </div>

<div class="ui container vertical segment">
    <a href="#">
        <h1 class="ui header">
            天气信息
        </h1>
    </a>

    <i class="icon grey small unhide"></i>
    <p>
        动态页面爬取。之所以叫动态页面爬取解析其实是相对于静态下载器与解析器来说的，因为有时候我们使用静态下载器与解析器对一些要爬取的页面进行解
        <a href="http://127.0.0.1:8000/page/tq">
            <i class="angle tiny double grey right icon">DB.MORE</i>
        </a>
    </p>

    <div class="ui mini  tag label">
        : )
    </div>
</div>
<div class="ui container vertical segment">
        <a href="#">
            <h1 class="ui header">
                京东页面
            </h1>
        </a>

        <i class="icon grey small unhide"></i>
        <p>
            在动态页面爬取的基础上，伪装浏览器，进行爬取。爬虫与反爬虫是一场持续的斗争，在大量爬取的的过程中，使用模拟无头浏览器爬取页面信息是一种方
            <a href="http://127.0.0.1:8000/page/jd">
                <i class="angle tiny double grey right icon">DB.MORE</i>
            </a>
        </p>

        <div class="ui mini  tag label">
            : )
        </div>
</div>
<div class="ui container vertical segment">
```

```
                    : )
                </div>
            </div>
            <div class="ui container vertical segment">
                <a href="#">
                    <h1 class="ui header">
                    淘宝书包
                    </h1>
                </a>

                <i class="icon grey small unhide"></i>
                <p>
                    使用正则表达式等解析页面信息经行爬取，注：淘宝的robot协议是禁止任何浏览器爬取的。
                    <a href="http://127.0.0.1:8000/page/tb">
                        <i class="angle tiny double grey right icon">DB.MORE</i>
                    </a>
                </p>

                <div class="ui mini  tag label">
                    : )
                </div>
            </div>


        </div>

        {% block body  %}{% endblock%}

        <div class="ui inverted  vertical very padded  segment">
            Mugglecoding®
        </div>

        {% block foot %}{% endblock%}

    </body>
</html>
```

web_db:

```
!DOCTYPE html>
{% load staticfiles %}
<html>
    <head>
        <meta charset="utf-8">
        <title>first web</title>
        <link rel="stylesheet" href="{% static 'css/semantic.css'%}"  media="screen" title="no title" charset="utf-8">
        <link href="https://fonts.googleapis.com/css?family=Oswald|Raleway" rel="stylesheet">


        <style type="text/css">
            h1 {
                font-family:'Oswald', sans-serif!important;
                font-size:40px;
            }

            body {
                font-family: 'Raleway', sans-serif;
            }
            p {
                font-family: 'Raleway', sans-serif;
                font-size:18px;
            }
            .ui.vertical.segment.masthead {
                height: 300px;
                background-image: url("{% static 'images/star_banner.jpg'%}");
                background-size: cover;
                background-position: 100% 80%;
            }

            .ui.container.segment {
                width: 800px;
            }

            .ui.center.aligned.header.blogslogon {
                margin-top: 40px;
            }

            .ui.center.aligned.header.blogslogon p {
                margin-top: 10px;
                color: □white;
                font-size: 10px;
            }
            .ui.container.nav {
                width: 500px;
            }
```

```
        }

    </style>
    {% block css %}{% endblock%}

</head>
<body>
    <div class="ui inverted vertical  segment masthead">

        <h1 class="ui center aligned header blogslogon" style="font-size:50px;font-family: 'Raleway', sans-serif!important;">
            爬虫展示
            <p class="ui sub header">
                下面为爬取数据展示。
            </p>

        </h1>
    </div>
    <div class="ui container nav">
        <div class="ui borderless text three item menu ">
            <div class="ui simple dropdown  item">
                加油
                <i class="dropdown icon"></i>
                <div class="menu">
                    <a class="item" href="">简介</a>
                    <a class="item" href="">简介</a>
                </div>
            </div>
            <a class="item">
                努力
            </a>
            <a class="item">
                简介
            </a>

        </div>
    </div>

    <div class="ui divider"></div>

    {% block header %}{% endblock%}

    <div  class="ui container nav" >

            <br>

            <br>
```

```
            <br>

            <h1>豆瓣电影</h1>

            <br>

            <br>

            <h3><a href='http://127.0.0.1:8000/page/main'>回到页面</a></h3>

            <br>

            <br>

            {% for content in content_list %}

            <h3>
                编号：{{content.id}} <br>

                名字：{{content.content}}<br>
            </h3>

            <br>

            <br><br><br>

            {% endfor %}




            {% if data %}

            <ul id="pages" class="pagination pagination-sm pagination-xs">

                    {% if data.first %}

                    <li><a href="?page=1">1</a></li>

                    {% endif %}

                    {% if data.left %}

                    {% if data.left_has_more %}
```

```
                {% if data.left_has_more %}
                <li><span>...</span></li>
                {% endif %}
                {% for i in data.left %}
                <li><a href="?page={{i}}">{{i}}</a></li>
                {% endfor %}
                {% endif %}
                <li class="active"><a href="?page={{data.page}}">{{data.page}}</a></li>
                {% if data.right %}
                {% for i in data.right %}
                <li><a href="?page={{i}}">{{i}}</a></li>
                {% endfor %}
                {% if data.right_has_more %}
                <li><span>...</span></li>
                {% endif %}
                {% endif %}
                {% if data.last %}
                <li><a href="?page={{data.total_pages}}">{{data.total_pages}}</a></li>
                {% endif %}
            </ul>
            {% endif %}
        </div>
        {% block body  %}{% endblock%}
        <div class="ui inverted  vertical very padded  segment">
        </div>
        {% block body  %}{% endblock%}
        <div class="ui inverted  vertical very padded  segment">
            Mugglecoding®
        </div>
        {% block foot %}{% endblock%}
    </body>
</html>
```

web_tq:

```
!DOCTYPE html>
{% load staticfiles %}
<html>
    <head>
        <meta charset="utf-8">
        <title>first web</title>
        <link rel="stylesheet" href="{% static 'css/semantic.css'%}"  media="screen" title="no title" charset="utf-8">
        <link href="https://fonts.googleapis.com/css?family=Oswald|Raleway" rel="stylesheet">


        <style type="text/css">
            h1 {
                font-family:'Oswald', sans-serif!important;
                font-size:40px;
            }

            body {
                font-family: 'Raleway', sans-serif;
            }
            p {
                font-family: 'Raleway', sans-serif;
                font-size:18px;
            }
            .ui.vertical.segment.masthead {
                height: 300px;
                background-image: url("{% static 'images/star_banner.jpg'%}");
                background-size: cover;
                background-position: 100% 80%;
            }

            .ui.container.segment {
                width: 800px;
            }

            .ui.center.aligned.header.blogslogon {
                margin-top: 40px;
            }

            .ui.center.aligned.header.blogslogon p {
                margin-top: 10px;
                color: □white;
                font-size: 10px;
            }
            .ui.container.nav {
                width: 500px;
            }
```

```
            }

        </style>
        {% block css %}{% endblock%}

    </head>
    <body>
        <div class="ui inverted vertical  segment masthead">

            <h1 class="ui center aligned header blogslogon" style="font-size:50px;font-family: 'Raleway', sans-serif!important;">
                爬虫展示
                <p class="ui sub header">
                    下面为爬取数据展示。
                </p>

            </h1>
        </div>
        <div class="ui container nav">
            <div class="ui borderless text three item menu ">
                <div class="ui simple dropdown  item">
                    加油
                    <i class="dropdown icon"></i>
                    <div class="menu">
                        <a class="item" href="">简介</a>
                        <a class="item" href="">简介</a>
                    </div>
                </div>
                <a class="item">
                    努力
                </a>
                <a class="item">
                    简介
                </a>

            </div>
        </div>

        <div class="ui divider"></div>

        {% block header %}{% endblock%}

        <div  class="ui container nav" >

            <br>

            <br>
```

```
<h1>天气页面</h1>

<br>

<br>

<h3><a href='http://127.0.0.1:8000/page/main'>回到页面</a></h3>

<br>

<br>

{% for weather in content_list %}

<h3>{{weather.id}}    :    
    {{weather.wCity}}        
    {{weather.wDate}}        
    {{weather.wWeather}}        
    {{weather.wTemp}}</h3>

<br><br><br>

{% endfor %}



{% if data %}
<ul id="pages" class="pagination pagination-sm pagination-xs">

    {% if data.first %}

    <li><a href="?page=1">1</a></li>

    {% endif %}

    {% if data.left %}

    {% if data.left_has_more %}

    <li><span>...</span></li>

    {% endif %}

    {% for i in data.left %}
```

```
    {% if data.left_has_more %}

    <li><span>...</span></li>

    {% endif %}

    {% for i in data.left %}

    <li><a href="?page={{i}}">{{i}}</a></li>

    {% endfor %}

    {% endif %}

    <li class="active"><a href="?page={{data.page}}">{{data.page}}</a></li>

    {% if data.right %}

    {% for i in data.right %}

    <li><a href="?page={{i}}">{{i}}</a></li>

    {% endfor %}

    {% if data.right_has_more %}

    <li><span>...</span></li>

    {% endif %}

    {% endif %}

    {% if data.last %}

    <li><a href="?page={{data.total_pages}}">{{data.total_pages}}</a></li>

    {% endif %}

</ul>

{% endif %}
</div>

{% block body  %}{% endblock%}

<div class="ui inverted  vertical very padded  segment">
```

```
    </div>

    {% block body  %}{% endblock%}

    <div class="ui inverted  vertical very padded  segment">
        Mugglecoding®
    </div>

    {% block foot %}{% endblock%}

  </body>
</html>
```

Web_jd:

```
!DOCTYPE html>
{% load staticfiles %}
<html>
    <head>
        <meta charset="utf-8">
        <title>first web</title>
        <link rel="stylesheet" href="{% static 'css/semantic.css'%}"  media="screen" title="no title" charset="utf-8">
        <link href="https://fonts.googleapis.com/css?family=Oswald|Raleway" rel="stylesheet">


        <style type="text/css">
            h1 {
                font-family:'Oswald', sans-serif!important;
                font-size:40px;
            }

            body {
                font-family: 'Raleway', sans-serif;
            }
            p {
                font-family: 'Raleway', sans-serif;
                font-size:18px;
            }
            .ui.vertical.segment.masthead {
                height: 300px;
                background-image: url("{% static 'images/star_banner.jpg'%}");
                background-size: cover;
                background-position: 100% 80%;
            }

            .ui.container.segment {
                width: 800px;
            }

            .ui.center.aligned.header.blogslogon {
                margin-top: 40px;
            }

            .ui.center.aligned.header.blogslogon p {
                margin-top: 10px;
                color: white;
                font-size: 10px;
            }
            .ui.container.nav {
                width: 500px;
            }
```

```
        }

    </style>
    {% block css %}{% endblock%}

</head>
<body>
    <div class="ui inverted vertical  segment masthead">

        <h1 class="ui center aligned header blogslogon" style="font-size:50px;font-family: 'Raleway', sans-serif!important;">
            爬虫展示
            <p class="ui sub header">
                下面为爬取数据展示。
            </p>

        </h1>
    </div>
    <div class="ui container nav">
        <div class="ui borderless text three item menu ">
            <div class="ui simple dropdown  item">
                加油
                <i class="dropdown icon"></i>
                <div class="menu">
                    <a class="item" href="">简介</a>
                    <a class="item" href="">简介</a>
                </div>
            </div>
            <a class="item">
                努力
            </a>
            <a class="item">
                简介
            </a>

        </div>
    </div>

    <div class="ui divider"></div>

    {% block header %}{% endblock%}

    <div  class="ui container nav" >

            <br>

            <br>
```

```
            <h1>京东手机</h1>

            <br>

            <br>

            <h3><a href='http://127.0.0.1:8000/page/main'>回到页面</a></h3>

            <br>

            <br>

            {% for content in content_list %}

                <h4>

                编号：{{content.mNo}}<br>

                图片：<br>

                <img src="static/download/{{content.mNo}}.jpg" /><br>

                品牌：{{content.mMark}}<br>

                价格：{{content.mPrice}}<br>

                简介：{{content.mNote}}<br>

                </h4>

                <br><br><br>

                {% endfor %}

            {% if data %}
            <ul id="pages" class="pagination pagination-sm pagination-xs">

                {% if data.first %}

                <li><a href="?page=1">1</a></li>

                {% endif %}

                {% if data.left %}

                {% if data.left has more %}
```

```
                {% if data.left_has_more %}
                <li><span>...</span></li>
                {% endif %}
                {% for i in data.left %}
                <li><a href="?page={{i}}">{{i}}</a></li>
                {% endfor %}
                {% endif %}
                <li class="active"><a href="?page={{data.page}}">{{data.page}}</a></li>
                {% if data.right %}
                {% for i in data.right %}
                <li><a href="?page={{i}}">{{i}}</a></li>
                {% endfor %}
                {% if data.right_has_more %}
                <li><span>...</span></li>
                {% endif %}
                {% endif %}
                {% if data.last %}
                <li><a href="?page={{data.total_pages}}">{{data.total_pages}}</a></li>
                {% endif %}
            </ul>
            {% endif %}
        </div>
        {% block body  %}{% endblock%}
        <div class="ui inverted  vertical very padded  segment">
        </div>
        {% block body  %}{% endblock%}
        <div class="ui inverted  vertical very padded  segment">
            Mugglecoding®
        </div>
        {% block foot %}{% endblock%}
    </body>
</html>
```

Web_tb:

```
!DOCTYPE html>
{% load staticfiles %}
<html>
    <head>
        <meta charset="utf-8">
        <title>first web</title>
        <link rel="stylesheet" href="{% static 'css/semantic.css'%}"  media="screen" title="no title" charset="utf-8">
        <link href="https://fonts.googleapis.com/css?family=Oswald|Raleway" rel="stylesheet">


        <style type="text/css">
            h1 {
                font-family:'Oswald', sans-serif!important;
                font-size:40px;
            }

            body {
                font-family: 'Raleway', sans-serif;
            }
            p {
                font-family: 'Raleway', sans-serif;
                font-size:18px;
            }
            .ui.vertical.segment.masthead {
                height: 300px;
                background-image: url("{% static 'images/star_banner.jpg'%}");
                background-size: cover;
                background-position: 100% 80%;
            }

            .ui.container.segment {
                width: 800px;
            }

            .ui.center.aligned.header.blogslogon {
                margin-top: 40px;
            }

            .ui.center.aligned.header.blogslogon p {
                margin-top: 10px;
                color: □white;
                font-size: 10px;
            }
            .ui.container.nav {
                width: 500px;
            }
```

```
            }
            .ui.container.nav {
                width: 500px;
            }

        </style>
        {% block css %}{% endblock%}

    </head>
    <body>
        <div class="ui inverted vertical  segment masthead">

            <h1 class="ui center aligned header blogslogon" style="font-size:50px;font-family: 'Raleway', sans-serif!important;">
                爬虫展示
                <p class="ui sub header">
                    下面为爬取数据展示。
                </p>

            </h1>
        </div>
        <div class="ui container nav">
            <div class="ui borderless text three item menu ">
                <div class="ui simple dropdown  item">
                    加油
                    <i class="dropdown icon"></i>
                    <div class="menu">
                        <a class="item" href="">简介</a>
                        <a class="item" href="">简介</a>
                    </div>
                </div>
                <a class="item">
                    努力
                </a>
                <a class="item">
                    简介
                </a>

            </div>
        </div>

        <div class="ui divider"></div>

        {% block header %}{% endblock%}

        <div  class="ui container nav" >
            <br>
```

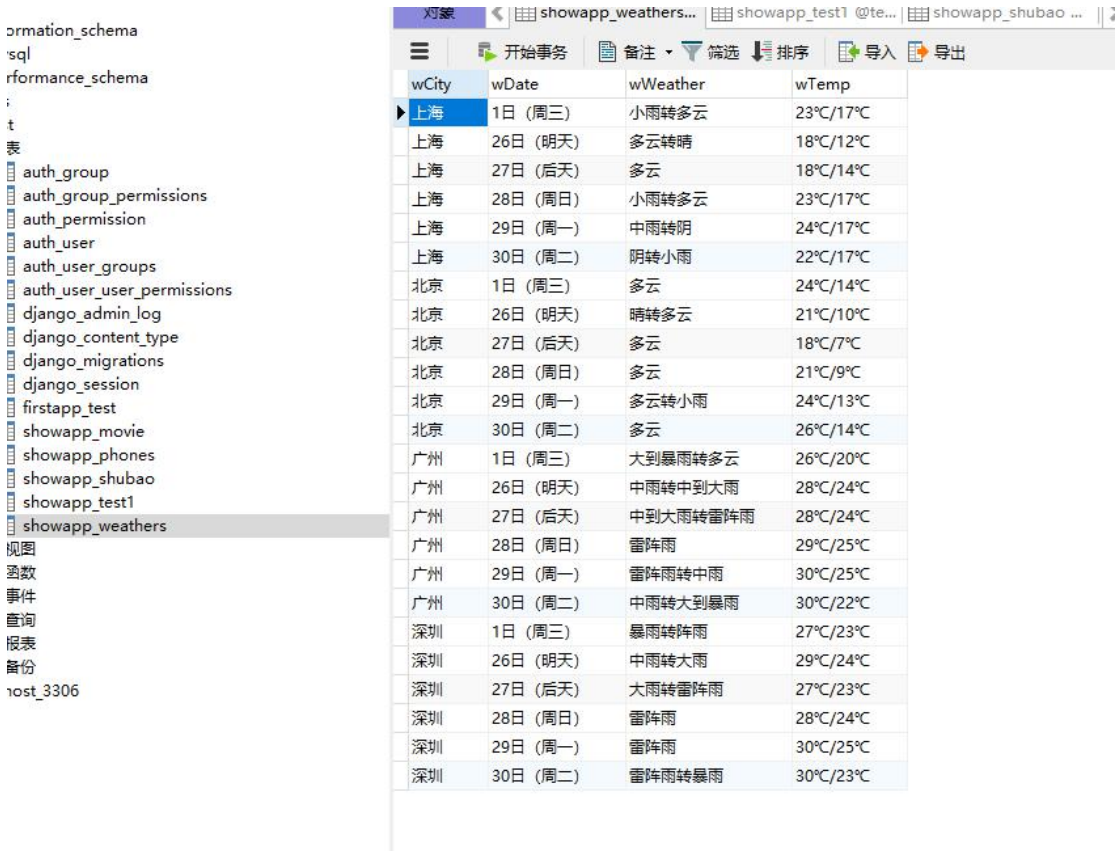
```
<div  class="ui container nav" >
    <br>
    <br>
    <h1>淘宝书包</h1>
    <br>
    <br>
    <h3><a href='http://127.0.0.1:8000/page/main'>回到页面</a></h3>
    <br>
    <br>
    {% for shubao in content_list %}
        <h3>
        编号：{{shubao.id}}<br>
        价格：{{shubao.Price}}<br>
        简介：{{shubao.Title}}</h3>
        <br><br>
        {% endfor %}
    {% if data %}

    <ul id="pages" class="pagination pagination-sm pagination-xs">

        {% if data.first %}

        <li><a href="?page=1">1</a></li>

        {% endif %}

        {% if data.left %}

        {% if data.left_has_more %}

        <li><span>...</span></li>

        {% endif %}

        {% for i in data.left %}

        <li><a href="?page={{i}}">{{i}}</a></li>

        {% endfor %}

        {% endif %}

        <li class="active"><a href="?page={{data.page}}">{{data.page}}</a></li>
```

```
        {% if data.left_has_more %}

        <li><span>...</span></li>

        {% endif %}

        {% for i in data.left %}

        <li><a href="?page={{i}}">{{i}}</a></li>

        {% endfor %}

        {% endif %}

        <li class="active"><a href="?page={{data.page}}">{{data.page}}</a></li>

        {% if data.right %}

        {% for i in data.right %}

        <li><a href="?page={{i}}">{{i}}</a></li>

        {% endfor %}

        {% if data.right_has_more %}

        <li><span>...</span></li>

        {% endif %}

        {% endif %}

        {% if data.last %}

        <li><a href="?page={{data.total_pages}}">{{data.total_pages}}</a></li>

        {% endif %}

    </ul>

    {% endif %}
</div>

{% block body  %}{% endblock%}

<div class="ui inverted  vertical very padded  segment">
```

```
    </div>

    {% block body  %}{% endblock%}

    <div class="ui inverted  vertical very padded  segment">
        Mugglecoding®
    </div>

    {% block foot %}{% endblock%}

  </body>
</html>
```

数据库：

NO	content
1	肖申克的救赎
2	霸王别姬
3	这个杀手不太冷
4	阿甘正传
5	美丽人生
6	泰坦尼克号
7	千与千寻
8	辛德勒的名单
9	盗梦空间
10	忠犬八公的故事
11	机器人总动员
12	三傻大闹宝莱坞
13	海上钢琴师
14	放牛班的春天
15	楚门的世界
16	大话西游之大圣娶亲
17	星际穿越
18	龙猫
19	教父
20	熔炉
21	无间道
22	疯狂动物城
23	当幸福来敲门
24	怦然心动
25	触不可及
26	蝙蝠侠：黑暗骑士
27	乱世佳人
28	活着
29	少年派的奇幻漂流



wCity	wDate	wWeather	wTemp
上海	1日（周三）	小雨转多云	23℃/17℃
上海	26日（明天）	多云转晴	18℃/12℃
上海	27日（后天）	多云	18℃/14℃
上海	28日（周日）	小雨转多云	23℃/17℃
上海	29日（周一）	中雨转阴	24℃/17℃
上海	30日（周二）	阴转小雨	22℃/17℃
北京	1日（周三）	多云	24℃/14℃
北京	26日（明天）	晴转多云	21℃/10℃
北京	27日（后天）	多云	18℃/7℃
北京	28日（周日）	多云	21℃/9℃
北京	29日（周一）	多云转小雨	24℃/13℃
北京	30日（周二）	多云	26℃/14℃
广州	1日（周三）	大到暴雨转多云	26℃/20℃
广州	26日（明天）	中雨转中到大雨	28℃/24℃
广州	27日（后天）	中到大雨转雷阵雨	28℃/24℃
广州	28日（周日）	雷阵雨	29℃/25℃
广州	29日（周一）	雷阵雨转中雨	30℃/25℃
广州	30日（周二）	中雨转大到暴雨	30℃/22℃
深圳	1日（周三）	暴雨转阵雨	27℃/23℃
深圳	26日（明天）	中雨转大雨	29℃/24℃
深圳	27日（后天）	大雨转雷阵雨	27℃/23℃
深圳	28日（周日）	雷阵雨	28℃/24℃
深圳	29日（周一）	雷阵雨	30℃/25℃
深圳	30日（周二）	雷阵雨转暴雨	30℃/23℃

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潮风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 女儿童双肩包 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
16	59.80	商务背包男士双肩包韩版潮流旅行包休闲女学生书包简约时尚电脑包
17	168.00	双肩包男书包男士时尚潮流青年休闲简约潮牌旅行背包大学生电脑包
18	69.00	迪士尼书包小学生男女1-3-4-6年级米奇减负背包儿童书包8-10-12岁
19	119.00	巴朗商务双肩包休闲时尚潮流大学生书包15.6寸电脑包男士背包男潮
20	99.00	米熙休闲运动背包双肩包女书包中学生男韩版时尚大容量旅游旅行包
21	195.02	国家地理背包女运动户外时尚双肩包男牛津布旅行防水学生情侣书包
22	79.00	双肩包女士2019新款小韩版百搭时尚书包2019旅行防盗牛津帆布背包
23	89.00	迪士尼书包小学生男童1-3-4五年级6-12周岁女孩儿童减负双肩背包
24	69.00	双肩包女2019新款潮牌韩版时尚百搭女士休闲pu软皮小背包旅行书包
25	138.00	瑞士军刀双肩包男 背包休闲商务旅行大容量瑞士书包电脑男士户外
26	139.00	花花公子男士双肩包时尚潮流休闲电脑旅行书包潮牌大学生帆布背包
27	89.00	HK双肩包男简约个性书包韩版时尚潮流休闲电脑包户外旅行轻便背包
28	49.00	背包男双肩包休闲大容量旅行包时尚潮流韩版高中生初中学生书包男
29	69.00	休闲双肩包男士韩版简约电脑旅行背包女时尚潮流初中高中学生书包

id	mNo	mMark	mPrice	mNote	mFile
1	000001	OPPO	3599.00	OPPO Reno 全面屏拍照手	000001.jp
2	000002	Apple	5899.00	Apple iPhone XR (A2108)	000002.jp
3	000003	【KPL官方比赛用机】vivo	3298.00	【KPL官方比赛用机】vivo i	000003.jp
4	000004	荣耀8X	1299.00	荣耀8X 千元屏霸 91%屏占	000004.jp
5	000005	荣耀10青春版	1299.00	荣耀10青春版 幻彩渐变 24(	000005.jp
6	000006	vivo	799.00	vivo U1 水滴全面屏 AI智慧	000006.jp
7	000007	荣耀V20	2799.00	荣耀V20 胡歌同款 麒麟980	000007.jp
8	000008	vivo	3598.00	vivo X27 8GB+256GB大内	000008.jp
9	000009	OPPO	2999.00	OPPO Reno手机 新品 全面	000009.jp
10	000010	小米	1199.00	小米 红米Redmi Note7 幻	000010.jp
11	000011	荣耀畅玩8C两天一充	899.00	荣耀畅玩8C两天一充 莱茵护	000011.jp
12	000012	小米	799.00	小米 红米6 4GB+64GB 铂	000012.jp
13	000013	黑鲨游戏手机2	3499.00	黑鲨游戏手机2 8GB+128G	000013.jp
14	000014	Apple	6199.00	Apple iPhone X (A1865) 6	000014.jp
15	000015	小米	799.00	小米 红米Redmi 7 幻彩渐	000015.jp
16	000016	小米8SE	1599.00	小米8SE 全面屏智能游戏拍	000016.jp
17	000017	三星	6999.00	三星 Galaxy S10+ 8GB+1	000017.jp
18	000018	Apple	9699.00	Apple iPhone XS Max (A2	000018.jp
19	000019	Apple	3799.00	Apple iPhone 8 (A1863) 6	000019.jp
20	000020	vivo	1598.00	vivo Z3 6GB+64GB 极光蓝	000020.jp
21	000021	华为	3999.00	华为 HUAWEI Mate 20 麒	000021.jp
22	000022	vivo	2298.00	vivo S1 6GB+128GB 宠爱	000022.jp
23	000023	Apple	4699.00	Apple iPhone 8 Plus (A18	000023.jp
24	000024	小米8青春版	1499.00	小米8青春版 镜面渐变AI双	000024.jp
25	000025	三星	1448.00	三星 Galaxy A6s 6GB+64(	000025.jp
26	000026	荣耀10	2199.00	荣耀10 GT游戏加速 AIS手	000026.jp
27	000027	诺基亚	1099.00	诺基亚 NOKIA X6 6GB+64	000027.jp
28	000028	华为	1999.00	华为 HUAWEI nova 4e 32(	000028.jp

补充：

多进程：

```python
#!/usr/bin/env python
# -*- coding=utf-8 -*-

from multiprocessing import Process, Queue

import time
from lxml import etree
import requests


class DouBanSpider(Process):
    def __init__(self, url, q):
        # 重写写父类的__init__方法
        super(DouBanSpider, self).__init__()
        self.url = url
        self.q = q
        self.headers = {
            'Host': 'movie.douban.com',
            'Referer': 'https://movie.douban.com/top250?start=225&filter=',
            'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/59.0.3071.104 Safari/537.36',
        }

    def run(self):
        self.parse_page()

    def send_request(self,url):
        '''
        用来发送请求的方法
        :return: 返回网页源码
        '''
        # 请求出错时，重复请求３次，
        i = 0
        while i <= 3:
            try:
                print (u"[INFO]请求url:"+url)
                return requests.get(url=url,headers=self.headers).content
            except Exception as e:
                print (u'[INFO] %s%s'% (e,url))
                i += 1

    def parse_page(self):
        '''
        解析网站源码，并采用ｘｐａｔｈ提取　电影名称和平分放到队列中
        :return:
        '''
        response = self.send_request(self.url)
        html = etree.HTML(response)
        #  获取到一页的电影数据
        node_list = html.xpath("//div[@class='info']")
        for move in node_list:
            # 电影名称
            title = move.xpath('.//a/span/text()')[0]
            # 评分
            score = move.xpath('.//div[@class="bd"]//span[@class="rating_num"]/text()')[0]

            # 将每一部电影的名称跟评分加入到队列
            self.q.put(score + "\t" + title)


def main():
    # 创建一个队列用来保存进程获取到的数据
    q = Queue()
    base_url = 'https://movie.douban.com/top250?start='
    # 构造所有ｕｒｌ
    url_list = [base_url+str(num) for num in range(0,225+1,25)]

    # 保存进程
    Process_list = []
    # 创建并启动进程
    for url in url_list:
        p = DouBanSpider(url,q)
        p.start()
        Process_list.append(p)

    # 让主进程等待子进程执行完成
    for i in Process_list:
        i.join()

    while not q.empty():
        print (q.get())

if __name__=="__main__":

    start = time.time()
    main()
    print ('[info]耗时：%s'%(time.time()-start))
```

多线程：

```python
#!/usr/bin/env python
# -*- coding=utf-8 -*-

from threading import Thread
from queue import Queue
import time
from lxml import etree
import requests

class DoubanSpider(Thread):
    def __init__(self,url,q):
        super(DoubanSpider,self).__init__()
        self.url=url
        self.q=q
        self.headers={
            'Cookie': 'll="118282"; bid=ctyiEarSLfw; ps=y; __yadk_uid=0Sr85yZ9d4bEeLKhv4w3695OFOPoedzC; dbcl2="155150959:OEu4dds1G1o"; as="https://
            'Host': 'movie.douban.com',
            'Referer': 'https://movie.douban.com/top250?start=225&filter=',
            'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/59.0.3071.104 Safari/537.36',
        }
    def run(self):
        self.parse_page()

    def send_request(self,url):
        '''
        用来发送请求的方法
        :return: 返回网页源码
        '''
        # 请求出错时，重复请求3次，
        i = 0
        while i <= 3:
            try:
                print (u"[INFO]请求url:"+url)
                html = requests.get(url=url,headers=self.headers).content
            except Exception as e:
                print (u'[INFO] %s%s'% (e,url))
                i += 1
            else:
                return html

    def parse_page(self):
        '''
        解析网站源码，并采用xpath提取  电影名称和平分放到队列中
        :return:
        '''
        response = self.send_request(self.url)
```

```python
        response = self.send_request(self.url)
        html = etree.HTML(response)
        #  获取到一页的电影数据
        node_list = html.xpath("//div[@class='info']")
        for move in node_list:
            # 电影名称
            title = move.xpath('.//a/span/text()')[0]
            # 评分
            score = move.xpath('.//div[@class="bd"]//span[@class="rating_num"]/text()')[0]

            # 将每一部电影的名称跟评分加入到队列
            self.q.put(score + "\t" + title)


def main():
    # 创建一个队列用来保存进程获取到的数据
    q = Queue()
    base_url = 'https://movie.douban.com/top250?start='
    # 构造所有url
    url_list = [base_url+str(num) for num in range(0,225+1,25)]

    # 保存线程
    Thread_list = []
    # 创建并启动线程
    for url in url_list:
        p = DouBanSpider(url,q)
        p.start()
        Thread_list.append(p)

    # 让主线程等待子线程执行完成
    for i in Thread_list:
        i.join()

    while not q.empty():
        print (q.get())

if __name__=="__main__":

    start = time.time()
    main()
    print ('[info]耗时: %s'%(time.time()-start))
```

总结：

1.正确爬取爬虫代码，并将其存储在数据库中。

2.使用 django，设计首页，包括样式，链接跳转，子页面。

3.子页面中{{classname：idname}}显示数据库调取的信息。

4.其中，我使用了 semantic ui 框架来作为我页面的美化。

5.但是在分页面中我是用{% extends ‘xxx.html’%}来继承主页面的样式时由于版本的原因，不能经行覆盖。

6.代码上传，编写报告，完成。