## 豆瓣 Top250

代码：

```
# -*-coding:utf-8 -*-
import io
import sys
#改变标准输出的默认编码
sys.stdout=io.TextIOWrapper(sys.stdout.buffer,encoding='utf8')
import requests
import MySQLdb
from lxml import etree

def get_page(start_num):
    url = 'https://movie.douban.com/top250?start=%s&filter='%start_num
    res = requests.get(url)

    tree = etree.HTML(res.text)
    top250 = tree.xpath('//span[@class="title"][1]/text()')
    print(top250)
    return top250

top250 = get_page(0)
print(top250)

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

    return result



topMovies = []
topMovies= get_all_page(0,10)
url = 'https://movie.douban.com/top250?start=%s&filter='
conn                                                                    =
MySQLdb.connect(host='localhost',user='root',passwd='123456',db='world'
,charset='utf8')
cur = conn.cursor()
for i in topMovies:
    cur.execute("INSERT          INTO          testmodle_test(id,content)
VALUES(%s, %s)",(id,str(i)))




cur.close()
```

```
conn.commit()
conn.close()
```

数据库信息：

| id | content |
| --- | --- |
| https://movie.douba | 2001太空漫游 |
| https://movie.douba | 7号房的礼物 |
| https://movie.douba | E.T. 外星人 |
| https://movie.douba | V字仇杀队 |
| https://movie.douba | 一一 |
| https://movie.douba | 一个叫欧维的男人决 |
| https://movie.douba | 一次别离 |
| https://movie.douba | 七宗罪 |
| https://movie.douba | 七武士 |
| https://movie.douba | 三傻大闹宝莱坞 |
| https://movie.douba | 三块广告牌 |
| https://movie.douba | 上帝之城 |
| https://movie.douba | 东京物语 |
| https://movie.douba | 东邪西毒 |
| https://movie.douba | 两杆大烟枪 |
| https://movie.douba | 乱世佳人 |
| https://movie.douba | 二十二 |
| https://movie.douba | 人工智能 |
| https://movie.douba | 低俗小说 |
| https://movie.douba | 你的名字。 |
| https://movie.douba | 你看起来好像很好吃 |
| https://movie.douba | 侧耳倾听 |
| https://movie.douba | 借东西的小人阿莉埃 |
| https://movie.douba | 倩女幽魂 |
| https://movie.douba | 傲慢与偏见 |

# 天气状况爬取：

```
from bs4 import BeautifulSoup
from bs4 import UnicodeDammit
import urllib.request
import sqlite3
# -*-coding:utf-8 -*-
import io
import sys
#改变标准输出的默认编码
sys.stdout=io.TextIOWrapper(sys.stdout.buffer,encoding='utf8')
import requests
import MySQLdb
from lxml import etree
def insert(city,date,weather,temp):
    try:
        conn                                                    =
MySQLdb.connect(host='localhost',user='root',passwd='123456',db='world'
,charset='utf8')
        cur = conn.cursor()
```

```
        cur.execute("INSERT                                                INTO
testmodle_weathers(City,Date,weather,Teap)
VALUES(%s,%s,%s,%s)",(city,date,weather,temp))
        cur.close()
        conn.commit()
        conn.close()
    except Exception as err:
        print(err)




class WeatherForecast:
    def __init__(self):
        self.headers  =  {"User-Agent":"Mozilla/5.0(Windows;U;Windows  NT
6.0 x64;en-US;rv:1.9pre) Gecko/2008072421 Minefield/3.0.2pre"}
        self.cityCode  =  {"北京":"101010100","南宁":"101300101","上海
":"101020100","广州":"101280101"}

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


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")
            n=0
            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
                    print(city,date,weather,temp)
                    insert(city,date,weather,temp)
                    n=n+1
                except Exception as err:
                    print(err)
```

```
        except Exception as err:
            print(err)


    def process(self,cities):
        for city in cities:
            self.forecastCity(city)

ws = WeatherForecast()
ws.process(["北京","南宁","上海","广州"])
```

## 数据库信息：

| id | City | Date | Weather | Teap |
|---|---|---|---|---|
| 1 | 北京 | 24日（今天） | 小雨 | 9℃ |
| 2 | 北京 | 25日（明天） | 多云转晴 | 19/8℃ |
| 3 | 北京 | 26日（后天） | 晴转多云 | 21/12℃ |
| 4 | 北京 | 27日（周六） | 小雨 | 18/8℃ |
| 5 | 北京 | 28日（周日） | 多云转晴 | 22/10℃ |
| 6 | 北京 | 29日（周一） | 多云 | 25/13℃ |
| 7 | 北京 | 30日（周二） | 小雨转阴 | 26/14℃ |
| 8 | 南宁 | 24日（今天） | 多云 | 25℃ |
| 9 | 南宁 | 25日（明天） | 多云转小雨 | 32/23℃ |
| 10 | 南宁 | 26日（后天） | 小雨转中雨 | 30/23℃ |
| 11 | 南宁 | 27日（周六） | 中雨转多云 | 29/23℃ |
| 12 | 南宁 | 28日（周日） | 多云 | 30/23℃ |
| 13 | 南宁 | 29日（周一） | 小雨转多云 | 31/24℃ |
| 14 | 南宁 | 30日（周二） | 小雨转中雨 | 32/24℃ |
| 15 | 上海 | 24日（今天） | 小雨转阴 | 17℃ |
| 16 | 上海 | 25日（明天） | 多云 | 25/15℃ |
| 17 | 上海 | 26日（后天） | 多云 | 20/14℃ |
| 18 | 上海 | 27日（周六） | 多云 | 20/18℃ |
| 19 | 上海 | 28日（周日） | 多云转小雨 | 24/19℃ |
| 20 | 上海 | 29日（周一） | 中雨 | 28/19℃ |
| 21 | 上海 | 30日（周二） | 小雨转大雨 | 23/15℃ |
| 22 | 广州 | 24日（今天） | 多云 | 25℃ |
| 23 | 广州 | 25日（明天） | 中雨转阴 | 30/23℃ |
| 24 | 广州 | 26日（后天） | 暴雨转大雨 | 29/25℃ |
| 25 | 广州 | 27日（周六） | 大雨转雷阵雨 | 27/24℃ |
| 26 | 广州 | 28日（周日） | 雷阵雨 | 29/24℃ |

## 使用 Selenium 爬取动态网页：

```
#coding:utf-8
from selenium import webdriver
import os
import time
import pymysql.cursors

class Crawler(object):
    def __init__(self):
        self.Chromedriver = "C:/Program Files (x86)/Google/Chrome/Application/chromedriver.exe"
        os.environ["webdriver.chrome.driver"] = self.Chromedriver
        self.Chrome = webdriver.Chrome(self.Chromedriver)
```

```
        # Connect to the database
        self.MySql = pymysql.connect(host='localhost',
                                     user='root',
                                     password='123456',
                                     db='world',
                                     charset='utf8',
                                     cursorclass=pymysql.cursors.DictCursor)

    def crawling36kr(self):
        url = "https://www.36kr.com/information/web_news"
        # 添加 cookie 前必须先打开一次网站
        self.Chrome.get(url)
        cookie = { "name" : "new_user_guidance", "value" : "true",
"domain" : ".36kr.com"}
        self.Chrome.add_cookie(cookie)
        self.Chrome.get(url)
        time.sleep(3)

        item_list = self.Chrome.find_elements_by_class_name("kr-shadow-
content")
        print("item_list = ", len(item_list))
        index = 0
        while index < len(item_list) - 1:
            item = item_list[index]
            index += 1
            item.location_once_scrolled_into_view
            imgSrc                                                    =
item.find_element_by_class_name("scaleBig").get_attribute("src")
            title = item.find_element_by_class_name("article-item-title")
            path = title.get_attribute("href")
            print(index, " len = ", len(item_list), " img = " , imgSrc ,
" path = " , path, " title = ", title.text)

            self.news_detail_36kr(index, item)
            item_list    =    self.Chrome.find_elements_by_class_name("kr-
shadow-content")
            time.sleep(2)

            if index == len(item_list) - 1:
                loading_more_button                                   =
self.Chrome.find_element_by_class_name("kr-loading-more-button")
                loading_more_button.location_once_scrolled_into_view
                time.sleep(3)
                item_list = self.Chrome.find_elements_by_class_name("kr-
shadow-content")
                if index == len(item_list) - 1:
                    loading_more_button.click()
                    time.sleep(3)
```

```
                item_list = self.Chrome.find_elements_by_class_name("kr-shadow-content")


    def news_detail_36kr(self, index, item):
        imgSrc = item.find_element_by_class_name("scaleBig").get_attribute("src")
        title = item.find_element_by_class_name("article-item-title")
        path = title.get_attribute("href")

        title.click()
        # select second page
        num = self.Chrome.window_handles
        self.Chrome.switch_to_window(num[1])
        time.sleep(5)
        # get content
        article_title = self.Chrome.find_element_by_class_name("article-title")
        author = self.Chrome.find_element_by_class_name("title-icon-item")
        summary = self.Chrome.find_element_by_class_name("summary")
        p_list = self.Chrome.find_elements_by_xpath("//*[@id='app']/div/div/div/div/div/div/div/div/div/div/div/div/div/div/p")
        print("article_title == ", article_title.text, " author = ", author.text, " summary = ", summary.text, " len(p_list) = ", len(p_list))
        content = ""
        for p in p_list:
            content += p.text
            print(content)

        # insert Data to db
        # try:
        try:
            with self.MySql.cursor() as cursor:
                # Create a new record
                sql = "INSERT INTO `testmodle_dontai` (`title`, `url`, `img`, `author`, `summary`, `content`) VALUES (%s, %s, %s, %s, %s, %s)"
                cursor.execute(sql, (article_title.text, path, imgSrc, author.text, summary.text, content))
            # connection is not autocommit by default. So you must commit to save your changes.
            self.MySql.commit()
        except Exception as e:
            print(e)

        time.sleep(3)
        self.Chrome.close()
        self.Chrome.switch_to_window(num[0])
```

```
if __name__ == "__main__":
    crawler = Crawler()
    crawler.crawling36kr()
```

## 数据库信息：

| d | title | url | img | author | summary |
|---|---|---|---|---|---|
| 1 | 谁吃掉了我们创造的 | https://36kr.com/p/ | https://pic.36krcnd. | 造就 | "理性只能把你 |
| 2 | 再见，复仇者 | https://36kr.com/p/ | https://pic.36krcnd. | 极客公园 | 十一年来，这一 |
| 3 | 为什么游戏第一股不 | https://36kr.com/p/ | https://pic.36krcnd. | 刺猬公社 | 整体来讲，直播 |
| 4 | 36氪「融资合伙人」 | https://36kr.com/p/ | https://pic.36krcnd. | 张达 | 顶级专家团线下 |
| 5 | 虎扑最新本地新闻推 | https://36kr.com/p/ | https://img.36krcdn | 全媒派 | 要抓住数字时代 |
| 6 | 华为大改革：2万CN | https://36kr.com/p/ | https://img.36krcdn | 36氪的朋友们 | '刀尖'上跳舞， |
| 7 | 最前线丨星巴克一口 | https://36kr.com/p/ | https://pic.36krcnd. | 吴筱 | 频频创新的星巴 |
| 8 | 36氪独家丨连咖啡完 | https://36kr.com/p/ | https://pic.36krcnd. | 彭倩 | 连咖啡会是下一 |
| 9 | 贝壳找房熟了 | https://36kr.com/p/ | https://img.36krcdn | 椅镜深网 | 从链家到贝壳， |
| 10 | 《权力的游戏》里的 | https://36kr.com/p/ | https://img.36krcdn | 三文娱 | "最基础的是要 |
| 11 | 成本节约70%，酒店 | https://36kr.com/p/ | https://pic.36krcnd. | 裴裴 | 芯片植入、夜间 |
| 12 | 恋爱综艺，滤镜下的 | https://36kr.com/p/ | https://img.36krcdn | 吴烁烁 | 恋爱综艺发生了 |
| 13 | QuestMobile：短视频 | https://36kr.com/p/ | https://pic.36krcnd. | 时氪分享 | 短视频也从差异 |
| 14 | 推酒店共享预订平台 | https://36kr.com/p/ | https://pic.36krcnd. | 茉小莉 | 下一站，改造有 |
| 15 | 明知IP不赚钱还追IP | https://36kr.com/p/ | https://img.36krcdn | 冯伞 | 做 IP，终归还 |
| 16 | 智能可穿戴的时尚身 | https://36kr.com/p/ | https://pic.36krcnd. | 脑极体 | 谷歌表示不服！ |
| 17 | 跨越7个时区，百亿 | https://36kr.com/p/ | https://pic.36krcnd. | 石亚琼 | 两个国家，一个 |
| 18 | 干思聪加持IG再夺冠 | https://36kr.com/p/ | https://pic.36krcnd. | 36氪的朋友们 | 相关俱乐部、电 |

五、实验结果与分析（含程序、数据记录及分析和实验总结等）：

## Django 的使用

## Settings.py：

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'TestModle',
]

T
```

```
BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [BASE_DIR+"/templates",],
        'APP_DIRS': True,
```

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'world',
```

```
        'USER': 'root',
        'PASSWORD':'123456',
        'HOST':'localhost',
        'PORT':'3306',
    }
```

## Testdb.py：

```
 from django.http import HttpResponse
from TestModle.models import Test
from TestModle.models import weathers
from TestModle.models import dontai
from django.shortcuts import render
import MySQLdb

#数据库操作
def testdb(request):

    allList = Test.objects.all()#获取 top250 电影
    weather = weathers.objects.all()#获取天
    dontais=dontai.objects.all()#获取动态信息

    Test.objects.order_by("id")


    return
render(request,'base1.html',{'allList':allList,'weather':weather,'donta
is':dontais})
```

## Urls.py：

```
from django.urls import path,include
from . import view,testdb

from TestModle import views

urlpatterns = [
    path(r'testdb/', testdb.testdb),
]
```

## Views.py

```
from django.http import HttpResponse
from django.shortcuts import render
import MySQLdb
```

## templates/html：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
</head>
<body>
       <div class="jumbotron text-center" style="margin-bottom:0">
              <h1>Django 显示页面</h1>
            </div>

            <nav class="navbar navbar-inverse">
              <div class="container-fluid">
                <div class="navbar-header">
                  <button   type="button"   class="navbar-toggle"   data-
toggle="collapse" data-target="#myNavbar">
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                  </button>
                  <a class="navbar-brand" href="#">网站名</a>


                </div>

              </div>
            </nav>
       <div class="container">
              <div class="row">
                <div class="col-sm-4">
                  <h2>席子文</h2>
                  <h5></h5>

                  <h3></h3>
                  <p></p>
                  <ul class="nav nav-pills nav-stacked">
                    <li                                  class="active"><a
href="https://github.com/16219111431/Django"
target="_blank">github</a></li>
```

```
                </ul>
                <hr class="hidden-sm hidden-md hidden-lg">
              </div>
        <h2>天气爬取</h2>
        <div class="panel-group" id="accordion">
          <div class="panel panel-default">
            <div class="panel-heading">
              <h4 class="panel-title">
                <a data-toggle="collapse" data-parent="#accordion"
                   href="#collapseOne">
                  七日天气预报
                </a>
              </h4>
            </div>
            <div id="collapseOne" class="panel-collapse collapse">
              <div class="panel-body">
                {% for i in weather %}
                <li   style="font-size:large">{{   i.City   }}   {{i.Date}}
{{i.Weather}} {{i.Teap}}</li>
                <hr />
                {% endfor %}
              </div>
            </div>
          </div>
        <br>
    <h2>豆瓣 Top250 电影</h2>
            <div class="fakeimg">图像</div>
              <div class="container"></div>
              <pre class="pre-scrollable"><!--内容可滚动-->
                <ol >
                  {% for i in allList %}
                  <li>{{ i.content }}</li>
                  {% endfor %}
                </ol>
              </pre>
            </div>
    <h2>动态信息</h2>
            <div class="dt"></div>
            <pre class="pre-scrollable">
                <ol>
                    {% for i in dontais %}
                    <li>{{i.title}}   {{i.url}}   {{i.img}}   {{i.author}}
{{i.summary}} {{i.content}} </li>
                    {% endfor %}
                </ol>
            </pre>
</body>
</html>
```

## TestModle/Models.py：

```
from django.db import models


from django.db import models

class Test(models.Model):
    content = models.CharField(max_length=255,default ="")
    objects = models.Manager()


class weathers(models.Model):
    City=models.CharField(max_length=20)
    Date =models.CharField(max_length=20)
    Weather=models.CharField(max_length=80)
    Teap=models.CharField(max_length=40)
    objects=models.Manager
class dontai(models.Model):
    title=models.CharField(max_length=50)
    url=models.CharField(max_length=80)
    img=models.CharField(max_length=80)
    author=models.CharField(max_length=80)
    summary=models.CharField(max_length=80)
    content=models.CharField(max_length=80)
```

## 页面显示：

# 豆瓣Top250电影

图像

1. 2001太空漫游
2. 7号房的礼物
3. E.T. 外星人
4. V字仇杀队

# 动态信息

1. 谁吃掉了我们创造的数据？ https://36kr.com/p/5197826 https://pic.36krcnd.com/2

2. 再见，复仇者 https://36kr.com/p/5197803 https://pic.36krcnd.com/201904/240238

3. 为什么游戏第一股不是斗鱼？ https://36kr.com/p/5197896 https://pic.36krcnd.com

4. 36氪「融资合伙人」顶级专家团重磅来袭！ https://36kr.com/p/5196745 https://pic