

项目说明

16计算机二班

朱永鑫

目录

静态页面爬取

动态页面爬取

Django项目

爬取静态页面 豆瓣网

- 1.首先观察网站的编写特点，来抓取我们所需要的信息。
如：我们所需要的信息都放在了li标签下，所以我们要先获取到所有li标签，再通过遍历和正则表达式获取到每个li标签里的我们所需要的信息。
- 2.编写插入数据库的方法，将获取到的信息插入到数据库中保存下来。

```
soup = BeautifulSoup(r.text, 'lxml').find_all('div', class_='item')
```

```
for i in range(len(soup)):
    name = re.findall(r'<span class="title">(.*?)</span>', str(soup[i]))[0]
    UnMovies=['三块广告牌', '海蒂和爷爷', '功夫', '疯狂的麦克斯4: 狂暴之路', '奇迹男孩']
    if(name in UnMovies):
        Director = re.findall(r'导演:\s(.*?)\s', str(soup[i]))[0]
        Star = re.findall(r'<span class="rating_num" property="v:average">(.*?)</span>', str(soup[i]))[0]
        Comment = '无'
        Pic = re.findall(r'img alt=".*?" class="" src="(.*?)" width=".*?"', str(soup[i]))[0]
        # str1=name+Director+Star+Comment
        insertMovie(name,Pic,Director,Star,Comment)
    else:
        Director = re.findall(r'导演:\s(.*?)\s', str(soup[i]))[0]
        Star = re.findall(r'<span class="rating_num" property="v:average">(.*?)</span>', str(soup[i]))[0]
        Comment = re.findall(r'<span class="inq">(.*?)</span>', str(soup[i]))[0]
        Pic = re.findall(r'img alt=".*?" class="" src="(.*?)" width=".*?"', str(soup[i]))[0]
```

```
def insertMovie(name, Pic, Director, Star, Comment):  
    db = pymysql.connect(host="localhost", user="root", password="admin", db="t_movie", port=3306)  
    cur = db.cursor()  
    # print(Director)  
    sql="insert into movie (id,pic,name,director,star,comment) values(null,'%s','%s','%s','%s','%s')"%(name,Pic,Director,Star,Comment)  
    # print(sql)  
    try:  
        cur.execute(sql)  
        db.commit()  
    except Exception as e:  
        print(e)  
    finally:  
        db.close()
```

id	name	pic	director	star	comment
1	肖申克的救赎	"https://img3.doubanio.co	弗兰克·德拉邦特	9.6	希望让人自由。
2	霸王别姬	https://img3.doubanio.co	陈凯歌	9.6	风华绝代。
3	这个杀手不太冷	https://img3.doubanio.co	吕克·贝松	9.4	怪蜀黍和小萝莉不得不说的故事。
4	阿甘正传	https://img1.doubanio.co	罗伯特·泽米吉斯	9.4	一部美国近现代史。
5	美丽人生	https://img3.doubanio.co	罗伯托·贝尼尼	9.5	最美的谎言。
6	泰坦尼克号	https://img3.doubanio.co	詹姆斯·卡梅隆	9.3	失去的才是永恒的。
7	千与千寻	https://img3.doubanio.co	宫崎骏	9.3	最好的宫崎骏，最好的久石让。
8	辛德勒的名单	https://img3.doubanio.co	史蒂文·斯皮尔伯格	9.5	拯救一个人，就是拯救整个世界。
9	盗梦空间	https://img3.doubanio.co	克里斯托弗·诺兰	9.3	诺兰给了我们一场无法盗取的梦。
10	忠犬八公的故事	https://img3.doubanio.co	莱塞·霍尔斯道姆	9.3	永远都不能忘记你所爱的人。
11	机器人总动员	https://img3.doubanio.co	安德鲁·斯坦顿	9.3	小瓦力，大人生。
12	三傻大闹宝莱坞	https://img3.doubanio.co	拉库马·希拉尼	9.2	英俊版憨豆，高情商版谢耳朵。
13	海上钢琴师	https://img1.doubanio.co	朱塞佩·托纳多雷	9.2	每个人都要走一条自己坚定了的路，就算是粉身碎骨。
14	放牛班的春天	https://img3.doubanio.co	克里斯托夫·巴拉蒂	9.3	天籁一般的童声，是最接近上帝的存在。
15	楚门的世界	https://img3.doubanio.co	彼得·威尔	9.2	如果再也不能见到你，祝你早安，午安，晚安。
16	大话西游之大圣娶亲	https://img3.doubanio.co	刘镇伟	9.2	一生所爱。
17	星际穿越	https://img3.doubanio.co	克里斯托弗·诺兰	9.2	爱是一种力量，让我们超越时空感知它的存在。
18	龙猫	https://img3.doubanio.co	宫崎骏	9.2	人人心中都有个龙猫，童年就永远不会消失。
19	教父	https://img3.doubanio.co	弗朗西斯·福特·科波拉	9.2	千万不要记恨你的对手，这样会让你失去理智。
20	熔炉	https://img3.doubanio.co	黄东赫	9.3	我们一路奋战不是为了改变世界，而是为了不让世界改变我们。
21	无间道	https://img3.doubanio.co	刘伟强	9.1	香港电影史上永不过时的杰作。
22	疯狂动物城	https://img1.doubanio.co	拜伦·霍华德	9.2	迪士尼给我们营造的乌托邦就是这样，永远善良勇敢，永远出乎意料。
23	当幸福来敲门	https://img3.doubanio.co	加布里尔·穆奇诺	9.0	平民励志片。
24	怦然心动	https://img3.doubanio.co	罗伯·莱纳	9.0	真正的幸福是来自内心深处。
25	触不可及	https://img3.doubanio.co	奥利维·那卡什	9.2	满满温情的高雅喜剧。
26	乱世佳人	https://img3.doubanio.co	维克多·弗莱明	9.2	Tomorrow is another day.

爬取动态页面

使用常规的request方法只能获得静态页面的数据，但是动态页面的数据大多数是用js来编写的，所以一旦我们用request方法来爬取，是爬取不到我们所需要的信息。

这次我爬的网站的图片的地址就是渲染后的地址，使用普通的request得到的地址并不是初始地址。

这次只是简单爬取图片地址以及发表图片用户的名字，并存入数据库中。

1.首先分析页面特点，我们所需要的信息都在li标签中，所有我们要先获取到所有li标签，再通过遍历，获取到我们所需要的信息。

2.将获取到的信息插入到数据库中。

月球先民
@58 MINS AGO

4232227
@天船：京张高铁×沙城铺轨基地，有幸拍了一次三机，虽然现在大家都已经不屑于拍三机，开始拍四机了。

[\[查看原图\]](#)



[举报] [OO](#) [9] [XX](#) [4] 吐槽 [0]

胖次喵喵叫
@2 HOURS AGO

4232174
快递开箱图
[\[查看原图\]](#)



nai nai
@3 HOURS AGO

[\[查看原图\]](#)




```
▼<ol class="commentlist" style="list-style-type: none;">
  ▶<li id="comment-4232227">...</li>
  ▶<li id="comment-4232174">...</li>
  ▶<li id="comment-4232153">...</li>
  <span class="break"></span>
  ▶<li class="row">...</li>
  ▶<li id="comment-4232147">...</li>
  ▶<li id="comment-4232099">...</li>
  ▶<li id="comment-4232088">...</li>
  ▶<li id="comment-4232074">...</li>
  ▶<li id="comment-4232038">...</li>
  ▶<li id="comment-4232026">...</li>
  ▶<li id="comment-4232017">...</li>
  ▶<li id="comment-4232015">...</li>
  ▶<li id="comment-4232008">...</li>
  ▶<li id="comment-4232002">...</li>
  ▶<li id="comment-4231992">...</li>
  ▶<li id="comment-4231981">...</li>
  ▶<li id="comment-4231966">...</li>
  ▶<li id="comment-4231957">...</li>
</ol>
```

```
▼<div class="author">
  <strong title="防伪码: ab51ca831cdef01ae5e6e849d1d6f0b4c3e3aa5a" class">月球先民</strong>
  <br>
  ▶<small>...</small>
</div>
```

```
▼<li id="comment-4232227">
  ▼<div>
    ▼<div class="row">
      ▶<div class="author">...</div>
      ▼<div class="text">
        ▶<span class="righttext">...</span>
        ▼<p == $0
          "@天船: 京张高铁×沙城铺轨基地, 有幸拍了一次三机, 虽然现在大家都已经不屑于拍三机, 开始拍四机了。"
          <br>
          <a href="//wx4.sinainimg.cn/large/538123191y1g2e13v9qv3j20sg0mr1bd.jpg" target="_blank" class=
            "view_img_link" refererpolicy="no-referrer">[查看原图]</a>
          <br>
          
          </p>
        </div>
      ▶<div class="jandan-vote">...</div>
    </div>
  </div>
</li>
```

使用request方法爬取到的图片链接，并非是初始图片地址。链接如下

```
['//img.jandan.net/img/blank.gif']  
['//img.jandan.net/img/blank.gif']  
['//img.jandan.net/img/blank.gif']  
['//img.jandan.net/img/blank.gif']  
['//img.jandan.net/img/blank.gif']  
['//img.jandan.net/img/blank.gif']  
['//img.jandan.net/img/blank.gif']  
['//img.jandan.net/img/blank.gif']  
['//img.jandan.net/img/blank.gif']  
['//img.jandan.net/img/blank.gif']  
['//img.jandan.net/img/blank.gif']  
['//img.jandan.net/img/blank.gif']  
['//img.jandan.net/img/blank.gif']  
['//img.jandan.net/img/blank.gif']  
['//img.jandan.net/img/blank.gif']
```

使用selenium爬取到的图片链接和发表图片的用户名如下：

```
wahaha //ws2.sinaimg.cn/mw600/687614ddgy1g2gi88njqsj21ls0winpd.jpg  
来个C //wx4.sinaimg.cn/thumb180/0076BSS5ly1g2gh3myuvxg304x09qx6p.gif  
月球先民 //wx4.sinaimg.cn/mw600/53812319ly1g2el3v9qv3j20sg0mr1bd.jpg  
胖次喵喵叫 //wx2.sinaimg.cn/mw600/007b1SRbly1g2gdxu53hoj31k0160npd.jpg  
nai nai //ws3.sinaimg.cn/mw600/bba87e2ely1g2gcxbrfqij20m813itmq.jpg  
lemonlhz //wx1.sinaimg.cn/mw600/007Bhwjoly1g2gct3nleij30y019chcq.jpg  
恨风一瞬而过 //wx3.sinaimg.cn/mw600/006qL5utly1g2garnwswnj315o1jku0x.jpg  
benshen //wx1.sinaimg.cn/mw600/d499889aly1g2g9ycbf9fj21jk15ohdt.jpg  
Ccpy //wx2.sinaimg.cn/mw600/006wH16pgy1g2g8z4q7t6j31eo120b29.jpg  
nai nai //wx2.sinaimg.cn/mw600/bba87e2ely1g2g71ijdi4j20u01v04qp.jpg  
来个C //wx4.sinaimg.cn/mw600/0076BSS5ly1g2g6md6j0wj30zk0nnp42.jpg  
CGWS8888 //ws4.sinaimg.cn/mw600/c4665261gy1g2g62qq250j20pt0waq5i.jpg  
来个C //wx1.sinaimg.cn/mw600/0076BSS5ly1g2g5zryq8vj30mt0y746v.jpg  
shuais //ws3.sinaimg.cn/mw600/6758b73aly1g2g5rdzn5gj215o1jke81.jpg  
来个C //wx1.sinaimg.cn/mw600/0076BSS5ly1g2g5e85waoj30u0196tz2.jpg  
来个C //wx1.sinaimg.cn/mw600/0076BSS5ly1g2g4t2ikhhkj30u011iu0x.jpg  
来个C //wx4.sinaimg.cn/mw600/0076BSS5ly1g2g45fi8bhj30b20gomzl.jpg  
来个C //wx3.sinaimg.cn/mw600/0076BSS5ly1g2g3k6s5oaj30xc0m844j.jpg  
来个C //wx2.sinaimg.cn/mw600/0076BSS5ly1g2g2xom9suj30jg0t6gyj.jpg
```

```

def insertMovie(name, Pic):
    db = pymysql.connect(host='localhost', user='root', password='admin', db='t_movie', port=3306)
    cur = db.cursor()
    # print(Director)
    sql='insert into comment (id, name, pic) values(null, '%s', '%s')'%(name, Pic)
    # print(sql)
    try:
        cur.execute(sql)
        db.commit()
    except Exception as e:
        print(e)
    finally:
        db.close()

```

id	name	pic
1	wahaha	//ws2.sinaimg.cn/mw600/687614ddgy1g2gi88njsj21ls0winpd.jpg
2	来个C	//wx4.sinaimg.cn/thumb180/0076BSS5ly1g2gh3myuvxg304x09qx6p.gif
3	月球先民	//wx4.sinaimg.cn/mw600/53812319ly1g2el3v9qv3j20sg0mr1bd.jpg
4	胖次喵喵叫	//wx2.sinaimg.cn/mw600/007b1SRbly1g2gdxu53hoj31k0160npd.jpg
5	nai nai	//ws3.sinaimg.cn/mw600/bba87e2ely1g2gcxbrfqij20m813itm.jpg
6	lemonlh	//wx1.sinaimg.cn/mw600/007Bhwjoly1g2gct3n1eij30y019chcq.jpg
7	恨风一瞬而	//wx3.sinaimg.cn/mw600/006qL5utly1g2garnwswnj315o1jku0x.jpg
8	benshen	//wx1.sinaimg.cn/mw600/d499889aly1g2g9ycbf9fj21jk15ohdt.jpg
9	Ccpy	//wx2.sinaimg.cn/mw600/006wH16pgy1g2g8z4q7t6j31eo120b29.jpg
10	nai nai	//wx2.sinaimg.cn/mw600/bba87e2ely1g2g71jdi4j20u01v04qp.jpg
11	来个C	//wx4.sinaimg.cn/mw600/0076BSS5ly1g2g6md6j0wj30zk0npn42.jpg
12	CGWS888	//ws4.sinaimg.cn/mw600/c4665261gy1g2g62qq250j20pt0waq5i.jpg
13	来个C	//wx1.sinaimg.cn/mw600/0076BSS5ly1g2g5zryq8vj30mt0y746v.jpg
14	shuais	//ws3.sinaimg.cn/mw600/6758b73aly1g2g5rdzn5gj215o1jke81.jpg
15	来个C	//wx1.sinaimg.cn/mw600/0076BSS5ly1g2g5e85waoj30u0196tz2.jpg
16	来个C	//wx1.sinaimg.cn/mw600/0076BSS5ly1g2g4t2ikhkj30u011iu0x.jpg
17	来个C	//wx4.sinaimg.cn/mw600/0076BSS5ly1g2g45fi8bhj30b20gomzl.jpg
18	来个C	//wx3.sinaimg.cn/mw600/0076BSS5ly1g2g3k6s5oaj30xc0m844j.jpg
19	来个C	//wx2.sinaimg.cn/mw600/0076BSS5ly1g2g2xom9suj30jg0t6gyj.jpg

Django项目 豆瓣网

前端使用bootstrap框架

后端使用Django框架

实现了用户的登陆注册注销功能

实现了电影信息列表显示

实现了分页效果

实现了电影信息的详情信息功能

实现了模糊搜索功能

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



肖申克的救赎

排名：1

导演：弗兰克·德拉邦特

评分：9.6

名言：希望让人自由。



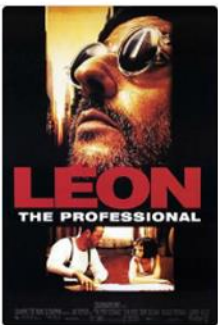
霸王别姬

排名：2

导演：陈凯歌

评分：9.6

名言：风华绝代。



这个杀手不太冷

排名：3

导演：吕克·贝松

评分：9.4

名言：怪蜀黍和小萝莉不得不说的故事。


```

def movie_list(request):
    movies=Movie.objects.all()
    p=Paginator(movies,10)
    if p.num_pages <= 1: # 如果文章不足一页
        article_list =movies # 直接返回所有文章
        data = '' # 不需要分页按钮
    else:
        page = int(request.GET.get('page', 1)) # 获取请求的文章页码，默认为第一页
        article_list = p.page(page) # 返回指定页码的页面
        left = [] # 当前页左边连续的页码号，初始值为空
        right = [] # 当前页右边连续的页码号，初始值为空
        left_has_more = False # 标示第 1 页页码后是否需要显示省略号
        right_has_more = False # 标示最后一页页码前是否需要显示省略号
        first = False # 标示是否需要显示第 1 页的页码号。
        # 因为如果当前页左边的连续页码号中已经含有第 1 页的页码号，此时就无需再显示第 1 页的页码号，
        # 其它情况下第一页的页码是始终需要显示的。
        # 初始值为 False
        last = False # 标示是否需要显示最后一页的页码号。
        total_pages = p.num_pages
        page_range = p.page_range
        if page == 1: # 如果请求第1页
            right = page_range[page:page + 2] # 获取右边连续号码页
            print(total_pages)
            if right[-1] < total_pages - 1: # 如果最右边的页码号比最后一页的页码号减去 1 还要小，
                # 说明最右边的页码号和最后一页的页码号之间还有其它页码，因此需要显示省略号，通过 right_has_more 来指示。
                right_has_more = True
            if right[-1] < total_pages: # 如果最右边的页码号比最后一页的页码号小，说明当前页右边的连续页码号中不包含最后一页的页码
                # 所以需要显示最后一页的页码号，通过 last 来指示
                last = True

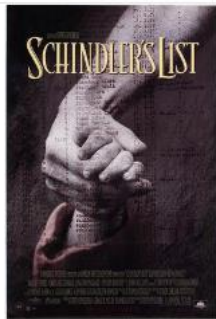
```

```

    if right[-1] < total_pages: # 如果最右边的页码号比最后一页的页码号小，说明当前页右边的连续页码号中不包含最后一页的页码
        # 所以需要显示最后一页的页码号，通过 last 来指示
        last = True
elif page == total_pages: # 如果请求最后一页
    left = page_range[(page - 3) if (page - 3) > 0 else 0:page - 1] # 获取左边连续号码页
    if left[0] > 2:
        left_has_more = True # 如果最左边的号码比2还要大，说明其与第一页之间还有其他页码，因此需要显示省略号，通过 left_has_more 来指示
    if left[0] > 1: # 如果最左边的页码比1要大，则要显示第一页，否则第一页已经被包含在其中
        first = True
else: # 如果请求的页码既不是第一页也不是最后一页
    left = page_range[(page - 3) if (page - 3) > 0 else 0:page - 1] # 获取左边连续号码页
    right = page_range[page:page + 2] # 获取右边连续号码页
    if left[0] > 2:
        left_has_more = True
    if left[0] > 1:
        first = True
    if right[-1] < total_pages - 1:
        right_has_more = True
    if right[-1] < total_pages:
        last = True
data = { # 将数据包含在data字典中
    'left': left,
    'right': right,
    'left_has_more': left_has_more,
    'right_has_more': right_has_more,
    'first': first,
    'last': last,
    'total_pages': total_pages,
    'page': page
}
return render(request, 'movie_list.html', context={
    'article_list': article_list, 'data': data
})

```

分页效果



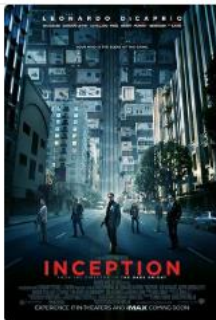
辛德勒的名单

排名: 8

导演: 史蒂文·斯皮尔伯格

评分: 9.5

名言: 拯救一个人, 就是拯救整个世界。



盗梦空间

排名: 9

导演: 克里斯托弗·诺兰

评分: 9.3

名言: 诺兰给了我们一场无法盗取的梦。



忠犬八公的故事

排名: 10

导演: 莱塞·霍尔斯道姆

评分: 9.3

名言: 永远都不能忘记你所爱的人。



龙猫

排名: 18

导演: 宫崎骏

评分: 9.2

名言: 人人心中都有个龙猫, 童年就永远不会消失。



教父

排名: 19

导演: 弗朗西斯·福特·科波拉

评分: 9.2

名言: 千万不要记恨你的对手, 这样会让你失去理智。



熔炉

排名: 20

导演: 黄东赫

评分: 9.3

名言: 我们一路奋战不是为了改变世界, 而是为了不让世界改变我们。

登录页面

豆瓣电影

剧情

喜剧

犯罪

爱情

动作

战争

灾难

动画

Search



登录

注册

登录 [没有账号？注册](#)

用户名/手机/邮箱

zyx

密码

.....

[忘记密码？](#)

登录


```
def loginCheck(request):  
    if request.method == 'POST':  
        username = request.POST.get('username')  
        password = request.POST.get('password')  
        try:  
            user = User.objects.get(username=username)  
        except:  
            return render(request, 'login.html')  
        # 查询用户是否在数据库中  
        if User.objects.filter(username=username, password=password).exists():  
            return render_to_response('movie_list.html.html', context={'username':username})
```

登录后的页面

豆瓣电影

剧情

喜剧

犯罪

爱情

动作

战争

灾难

动画

Search



欢迎你,zyx

注销

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



肖申克的救赎

排名: 1

导演: 弗兰克·德拉邦特

评分: 9.6

名言: 希望让人自由。



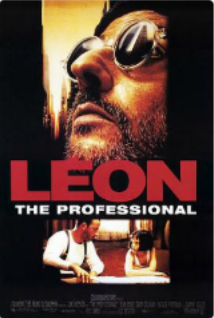
霸王别姬

排名: 2

导演: 陈凯歌

评分: 9.6

名言: 风华绝代。



这个杀手不太冷

排名: 3

导演: 吕克·贝松

评分: 9.4

名言: 怪蜀黍和小萝莉不得不说的故事。

肖申克的救赎



肖申克的救赎

排名：1

导演：弗兰克·德拉邦特

评分：9.6

名言：希望让人自由。

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



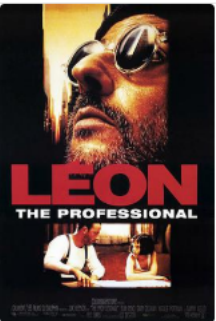
霸王别姬

排名：2

导演：陈凯歌

评分：9.6

名言：风华绝代。



这个杀手不太冷

排名：3

导演：吕克·贝松

评分：9.4

名言：怪蜀黍和小萝莉不得不说的故事。

```
def logout(request):  
    if not request.session.get('is_login', None):  
        # 如果本来就未登录，也就没有登出一说  
        return redirect("movie_list")  
    request.session.flush()  
    # 或者使用下面的方法  
    # del request.session['is_login']  
    # del request.session['user_id']  
    # del request.session['user_name']  
    return redirect("movie_list")
```

注册

已有账号? [登录](#)

用户名

密码

手机号

注册即代表您同意[服务条款](#)

	id	username	password	phone
▶	2	zyx	96zhuyongxin	15558966862
	3	admin	123	15558966862
	4	16219111217	96zhuyongxin	15558966862
	5	lan	96zhuyongxin	15558966862

注册页面


```
def registerForm(request):  
    if request.method == 'POST':  
        username = str(request.POST.get('username', ''))  
        password = str(request.POST.get('password', ''))  
        phone=str(request.POST.get('phone', ''))  
        new_user = User.objects.create()  
        new_user.username=username  
        new_user.password=password  
        new_user.phone=phone  
        new_user.save()  
        return redirect('movie_list')  
    return render(request, 'movie_list.html', locals())
```

详情信息页面

豆瓣电影

剧情

喜剧

犯罪

爱情

动作

战争

灾难

动画

Search



登录

注册

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

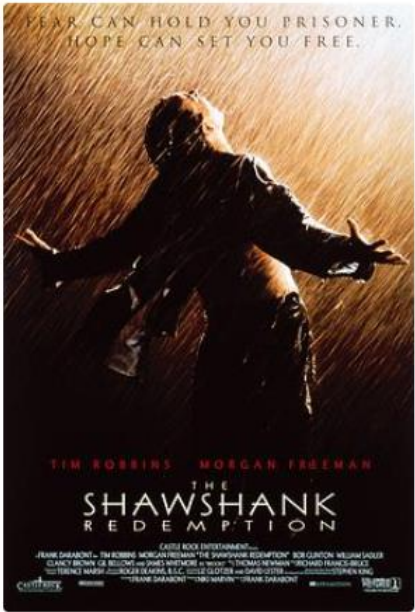
泰坦尼克号

千与千寻

辛德勒的名单

盗梦空间

忠犬八公的故事



肖申克的救赎

排名: 1

导演: 弗兰克·德拉邦特

评分: 9.6

名言: “希望让人自由。”

剧情: 20世纪40年代末, 小有成就的青年银行家安迪 (蒂姆·罗宾斯 Tim Robbins 饰) 因涉嫌杀害妻子及她的情人而锒铛入狱。在这座名为肖申克的监狱内, 希望似乎虚无缥缈, 终身监禁的惩罚无疑注定了安迪接下来灰暗绝望的人生。未过多久, 安迪尝试接近囚犯中颇有声望的瑞德 (摩根·弗里曼 Morgan Freeman 饰), 请求对方帮自己搞来小锤子。以此为契机, 二人逐渐熟稔, 安迪也仿佛在鱼龙混杂、罪恶横生、黑白混淆的牢狱中找到属于自己的求生之道。他利用自身的专业知识, 帮助监狱管理层逃税、洗黑钱, 同时凭..(展开全部)

```
def movie_detail(request, movie_id):
    movie = get_object_or_404(Movie, pk=movie_id)
    movies = Movie.objects.all()
    p = Paginator(movies, 10)
    if p.num_pages <= 1: # 如果文章不足一页
        article_list = movies # 直接返回所有文章
        data = '' # 不需要分页按钮
    else:
        page = int(request.GET.get('page', 1)) # 获取请求的文章页码，默认为第一页
        print(page)
        article_list = p.page(page) # 返回指定页码的页面
        left = [] # 当前页左边连续的页码号，初始值为空
        right = [] # 当前页右边连续的页码号，初始值为空
        left_has_more = False # 标示第 1 页页码后是否需要显示省略号
        right_has_more = False # 标示最后一页页码前是否需要显示省略号
        first = False # 标示是否需要显示第 1 页的页码号。
        # 因为如果当前页左边的连续页码号中已经含有第 1 页的页码号，此时就无需再显示第 1 页的页码号，
        # 其它情况下第一页的页码是始终需要显示的。
        # 初始值为 False
        last = False # 标示是否需要显示最后一页的页码号。
        total_pages = p.num_pages
        page_range = p.page_range
        if page == 1: # 如果请求第1页
            right = page_range[page:page + 2] # 获取右边连续号码页
            print(total_pages)
            if right[-1] < total_pages - 1: # 如果最右边的页码号比最后一页的页码号减去 1 还要小，
                # 说明最右边的页码号和最后一页的页码号之间还有其它页码，因此需要显示省略号，通过 right_has_more 来指示。
                right_has_more = True
```

```

    if right[-1] < total_pages: # 如果最右边的页码号比最后一页的页码号小，说明当前页右边的连续页码号中不包含最后一页的页码
        # 所以需要显示最后一页的页码号，通过 last 来指示
        last = True
elif page == total_pages: # 如果请求最后一页
    left = page_range[(page - 3) if (page - 3) > 0 else 0:page - 1] # 获取左边连续号码页
    if left[0] > 2:
        left_has_more = True # 如果最左边的号码比2还要大，说明其与第一页之间还有其他页码，因此需要显示省略号，通过 left_has_more 来指示
    if left[0] > 1: # 如果最左边的页码比1要大，则要显示第一页，否则第一页已经被包含在其中
        first = True
else: # 如果请求的页码既不是第一页也不是最后一页
    left = page_range[(page - 3) if (page - 3) > 0 else 0:page - 1] # 获取左边连续号码页
    right = page_range[page:page + 2] # 获取右边连续号码页
    if left[0] > 2:
        left_has_more = True
    if left[0] > 1:
        first = True
    if right[-1] < total_pages - 1:
        right_has_more = True
    if right[-1] < total_pages:
        last = True
data = { # 将数据包含在data字典中
    'left': left,
    'right': right,
    'left_has_more': left_has_more,
    'right_has_more': right_has_more,
    'first': first,
    'last': last,
    'total_pages': total_pages,
    'page': page
}

Context={ 'article_list': article_list, 'data': data, 'movie':movie}
return render_to_response('movie_detail.html',Context)

```

搜索页面

豆瓣电影

剧情

喜剧

犯罪

爱情

动作

战争

灾难

动画

肖申克



欢迎你,zyx

注销

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



肖申克的救赎

排名: 1

导演: 弗兰克·德拉邦特

评分: 9.6

名言: 希望让人自由。



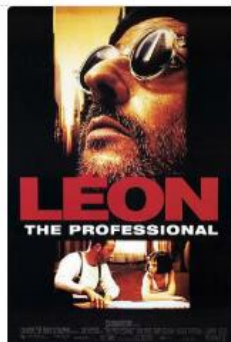
霸王别姬

排名: 2

导演: 陈凯歌

评分: 9.6

名言: 风华绝代。



这个杀手不太冷

排名: 3

导演: 吕克·贝松

评分: 9.4

名言: 怪蜀黍和小萝莉不得不说的故事。

豆瓣电影

剧情

喜剧

犯罪

爱情

动作

战争

灾难

动画

Search



欢迎你,zyx

注销

搜索“肖申克”，找到1个结果

肖申克的救赎

搜索“肖申克的久没”，找到0个结果

没有找到任何东西，换个关键词试试吧

```
def search(request):  
    search_word=request.GET.get('wd','')  
    search_movies=Movie.objects.filter(name__icontains=search_word)  
  
    context={}  
    context['search_word']=search_word  
    context['search_movies']=search_movies  
    context['search_movies_count']=search_movies.count()  
    return render(request,'search.html',context)
```

THANK YOU