**安徽工程大学**

**（豆瓣电影数据采集）**

|  |  |
| --- | --- |
| 院 系 | 计算机与信息学院 |
| 专业班级 | 数据科学202 |
| 姓 名 | 刘起杰 |
| 学 号 | 3200706208 |
| 指导老师 | 杨磊、冯富霞、陈富春、杨丹 |

2022年 6月 日

# 摘 要

近年来，随着网络应用的逐渐扩展和深入，如何高效的获取网上数据成为了无数公司和个人的追求，在大数据时代，谁掌握了更多的数据，谁就可以获得更高的利益，而网络爬虫是其中最为常用的一种从网上爬取数据的手段。

网络爬虫，即Web Spider，是一个很形象的名字。如果把互联网比喻成一个蜘蛛网，那么Spider就是在网上爬来爬去的蜘蛛。网络蜘蛛是通过网页的链接地址来寻找网页的。从网站某一个页面〈通常是首页）开始，读取网页的内容，找到在网页中的其它链接地址，然后通过这些链接地址寻找下一个网页，这样一直循环下去，直到把这个网站所有的网页都抓取完为止。如果把整个互联网当成一个网站，那么网络蜘蛛就可以用这个原理把互联网上所有的网页都抓取下来。

既然网络爬虫有着如此快捷方便的特点，我们该如何实现它呢?在众多面向对象的语言中，首选python，因为python是一种“解释型的、面向对象的、带有动态语义的”高级程序，可以使人在编程时保持自己的风格，并且编写的程序清晰易懂，有着很广阔的应用前景。

本课题研究的是通过爬虫技术，爬取豆瓣网站中，当地院线正在上映的电影情况，帮助用户享受更好的观影体验。

**关键词：**python；爬虫；数据采集；数据预处理

# 引言

随着网络的迅速发展，万维网成为大量信息的载体，如何有效地提取并利用这些信息成为一个巨大的挑战。搜索引擎（例如传统的通用搜索引擎AltaVista，Yahoo!和Google等)作为一个辅助人们检索信息的工具成为用户访问万维网的入口和指南。但是，这些通用性搜索引擎也存在着一定的局限性，如:

1. 不同领域、不同背景的用户往往具有不同的检索目的和需求，通用搜索引擎所返回的结果包含大量用户不关心的网页。
2. (2)通用搜索引擎的目标是尽可能大的网络覆盖率，有限的搜索引擎服务器资源与无限的网络数据资源之间的矛盾将进一步加深。
3. 3）万维网数据形式的丰富和网络技术的不断发展，图片、数据库、音频/视频多媒体等不同数据大量出现，通用搜索引擎往往对这些信息含量密集且具有一定结构的数据无能为力，不能很好地发现和获取。
4. (4)通用搜索引擎大多提供基于关键字的检索，难以支持根据语义信息提出的查询。

为了解决上述问题，定向抓取相关网页资源的聚焦爬虫应运而生。聚焦爬虫是一个自动下载网页的程序，它根据既定的抓取目标，有选择的访问万维网上的网页与相关的链接，获取所需要的信息。与通用爬虫(generalpurpose web crawler)不同，聚焦爬虫并不追求大的覆盖，而将目标定为抓取与某一特定主题内容相关的网页，为面向主题的用户查询准备数据资源。

目 录

[摘 要 1](#_Toc105099737)

[引言 2](#_Toc105099738)

[1 绪论 1](#_Toc105099739)

[1.1 功能要求 1](#_Toc105099740)

[1.2 完成目标 1](#_Toc105099741)

[1.3 研究意义 1](#_Toc105099742)

[2 网络爬虫的工作原理 2](#_Toc105099743)

[2.1 工作原理 2](#_Toc105099744)

[2.2 工作流程及数据爬取过程分析 2](#_Toc105099745)

[3 界面设计 3](#_Toc105099746)

[3.1.1 作品展示 3](#_Toc105099747)

[4 爬虫程序的具体是实现 4](#_Toc105099748)

[4.1 界面开发工具及思路架构 4](#_Toc105099749)

[4.1.1 开发工具 4](#_Toc105099750)

[4.1.2 程序具体思路 4](#_Toc105099751)

[4.2 获取ip地址 4](#_Toc105099752)

[4.2.1 代码解释 5](#_Toc105099753)

[4.3 构建IP代理池 6](#_Toc105099754)

[4.3.1 什么是IP代理池？ 6](#_Toc105099755)

[4.3.2 IP代理池的作用 6](#_Toc105099756)

[4.3.3 主要代码 6](#_Toc105099757)

[4.4 爬取豆瓣网院线电影信息 8](#_Toc105099758)

[4.4.1 代码解释-基本数据 8](#_Toc105099759)

[4.4.2 代码解释-封面爬取 9](#_Toc105099760)

[4.5 tkinter构建GUI交互界面 10](#_Toc105099761)

[4.5.1 什么是tkinter？ 10](#_Toc105099762)

[4.5.2 代码解释 10](#_Toc105099763)

[4参考文献 13](#_Toc105099764)

[5 附录 14](#_Toc105099765)

[5.1 获取IP地址 14](#_Toc105099766)

[5.2 构建IP代理池 14](#_Toc105099767)

[5.3 爬取豆瓣网院线电影信息 15](#_Toc105099768)

[5.4 Tkinter构建GUI交互界面 17](#_Toc105099769)

# 绪论

## 功能要求

爬取豆瓣网中当地电影院的电影上映情况，爬取剧情、喜剧动作、爱情、科幻等标签下的名字、上映时间、评分、导演等信息，对数据进行预处理，统计分析，并得到对应的分析结果。

分析电影评论，分析不同类型电影的所占比例是多少，并将结果用条形图展示；爬取当地上映电影的详细信息，包括封面、上映时间、是否有票进行展示。

## 完成目标

1. 网络编程：开发相应的后台爬取程序，如数据爬取，存储数据等。
2. GUI 编程：利用tkinter库开发前台GUI姐买你，便于用户操作。
3. 文件操作：在系统的开发中，在核实的功能实现的时候，对文件进行读写和更新。

## 研究意义

深入学习python和爬虫编程，自己动手实现由良好扩展性的网路爬虫，提高学习新技术的能力。

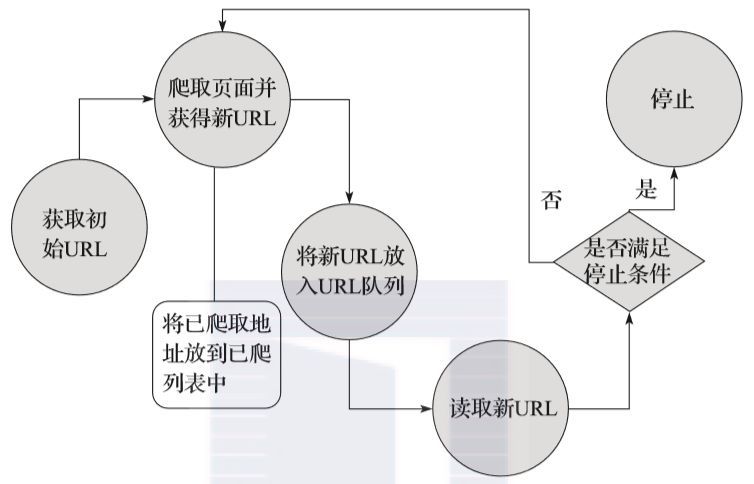
利用软件工程学的思想，探索简单易扩展的设计方案，让我们的爬虫系统做到功能完善，提高扩展性。

# 网络爬虫的工作原理

## 工作原理

爬虫首先要做的工作是获取网页的源代码，源代码里包含了网页的部分有用信息；之后爬虫构造一个请求并发送给服务器，服务器接收到响应并将其解析出来。实际上，获取网页——分析网页源代码——提取信息，便是爬虫工作的三部曲，如图1-1所示

如何提取信息？最通用的方法是采用正则表达式。网页结构有一定的规则，还有一些根据网页节点属性、CSS选择器或XPath来提取网页信息的库，如Requests、pyquery、lxml等，使用这些库，便可以高效快速地从中提取网页信息，如节点的属性、文本值等，并能简单保存为TXT文本或JSON文本，这些信息可保存到数据库。



图表 1-1

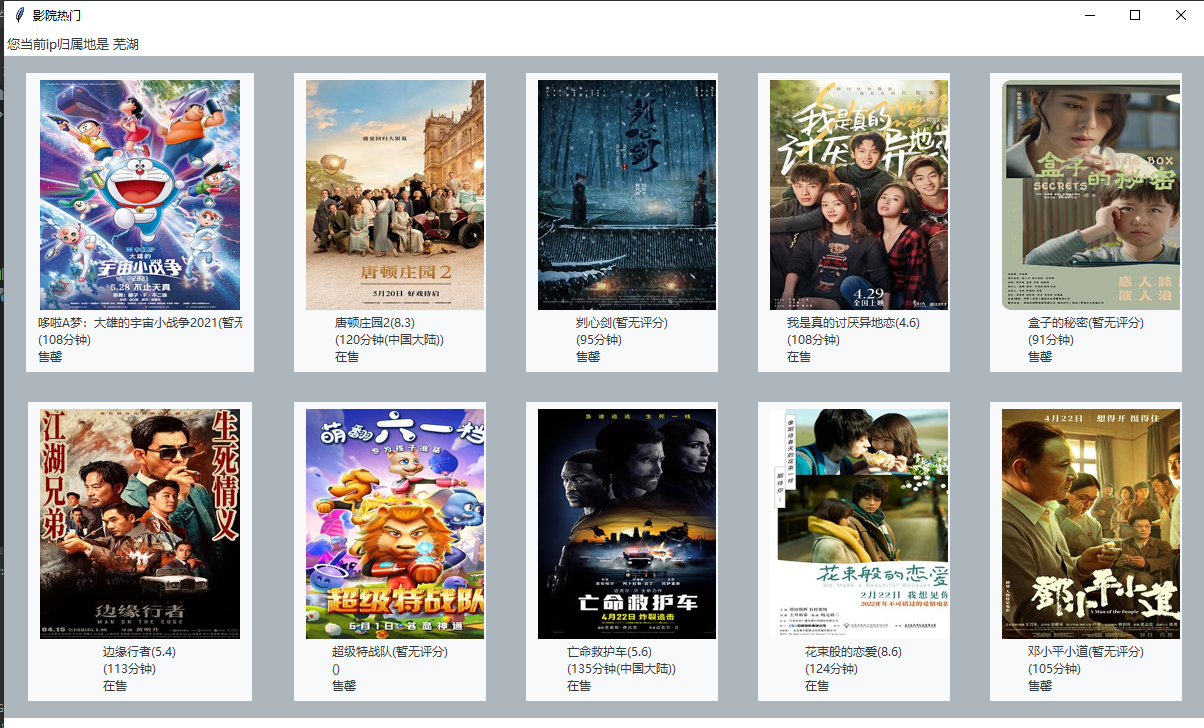
## 工作流程及数据爬取过程分析

与普通的用户使用浏览器的工作原理类似，爬虫工作爬取网页的过程也是在于对Request和Response的处理。

通过HTTP库向目标站点发起请求，也就是发送一个Request请求，其中可以包含header等信息，等待服务器相应。如果服务器能正常相应，会得到一个Response，我们要爬取的数据便在Response中，数据类型可能是HTML、Json字符串、二进制数据（图片或视频）等类型。保存的形式多样，可以村委文本，也可以保存到数据库等。

# 界面设计

### 作品展示



程序会首先获取当前电脑的ip地址，从而通过地理位置爬取当地影院上线电影情况。如果想查看详细信息的话，可以点击图片查看。

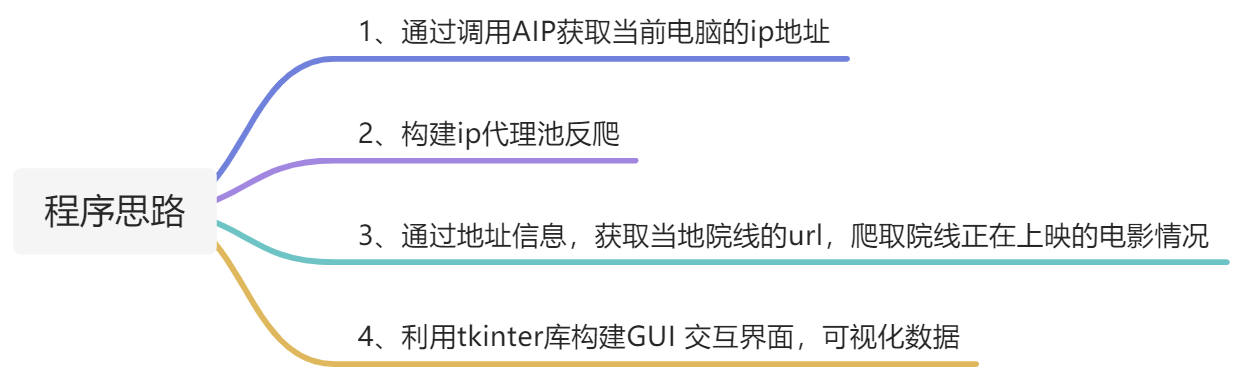
# 爬虫程序的具体实现

## 界面开发工具及思路架构

### 开发工具

* 开发语言：python
* 操作系统：windows10
* 开发工具：Pychram（Community Edition）
* 文本解析库：Xpath
* 文档编写工具：Word

### 程序具体思路



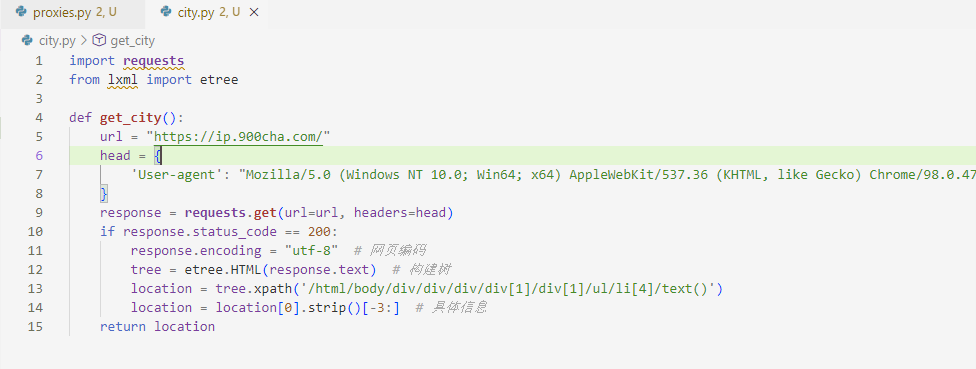
## 获取ip地址

程序会首先通过爬取IP地址查询网站，获取当前电脑的IP地址如图2。



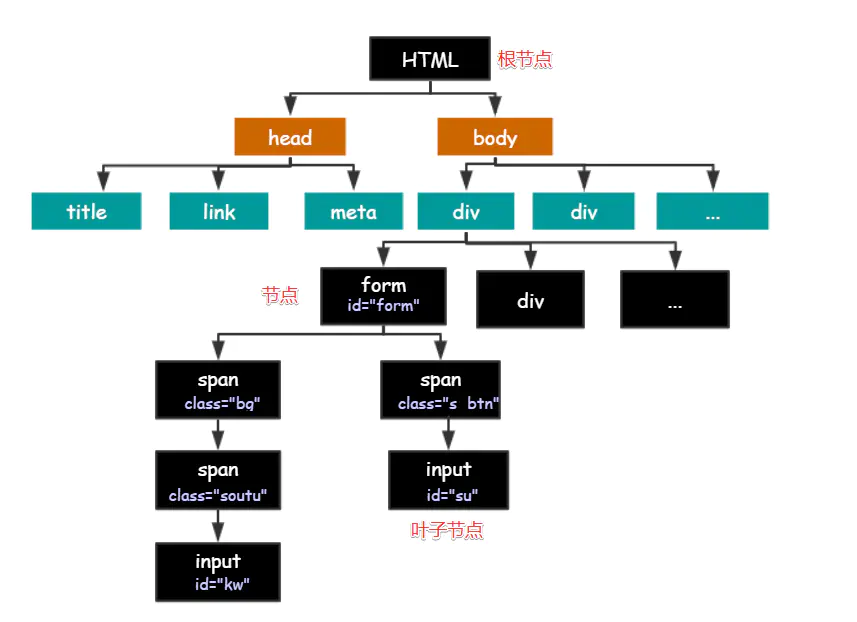
图表 2

### 代码解释



有些网站为了防止爬虫程序恶意访问，在检测到爬虫程序多次访问服务器，会禁止该程序访问。所以，在获取数据之前，需要将爬虫程序伪装成一个浏览器，以实现网页的适量爬取。通过添加header参数即可实现伪装浏览器这一步骤。

在成功爬虫网页数据后，需要对网页数据进行解析，xpath是一个网页解析的利器。什么是Xpath？Xpath是一种用在XML文档定位元素的语言，通过也支持HTML元素解析。我们通过下面图3来了解HTML的结构。



图表 3

HTML 的结构就是树形结构，HTML 是根节点，所有的其他元素节点都是从根节点发出的。其他的元素都是这棵树上的节点Node，每个节点还可能有属性和文本。而路径就是指某个节点到另一个节点的路线。

通过浏览器的开发者工具，我们可以拷贝到Xpath的路径代码。



Xpath通过路径便可以快速的获取相对应的数据。

## 构建IP代理池

### 什么是IP代理池？

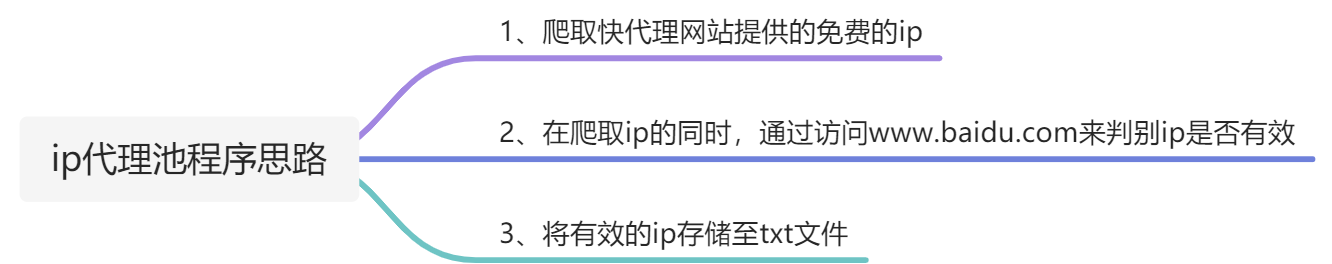
ip代理充当您的计算机和网站之间的中介，帮助您将请求路由到网站并显示其自己的IP地址，同时隐藏您自己的IP地址。

### IP代理池的作用

1. 有时访问网站找不着网址并不是错误或网址关闭，只是你的访问请求被限制。比如国外的网址国内的网络是不可以访问，因此一些网站登录不上。使用代理IP，改成别的地区的IP有时就可以访问了。
2. 当大量爬取一个网页的数据时，网站会禁止该ip访问服务器，使用代理IP访问网站的时候，另一方服务器查出的IP是代理IP的信息，并不是我们真正的IP，可以很好的隐藏我们真正的IP地址，从而防止在爬取的过程是出现报错。

### 主要代码

#### 代码思路



#### 爬取ip代理代码解释





首先通过爬取<https://free.kuaidaili.com/free>提供的免费的ip代理，利用xpath获取数据。



其次，构建简单的爬虫程序，IP设置为爬取的IP，访问<https://www.baidu.com/>来判断IP是否有效，如果有效则保存至proxies.txt文件中，方便后面程序的调用。

## 爬取豆瓣网院线电影信息

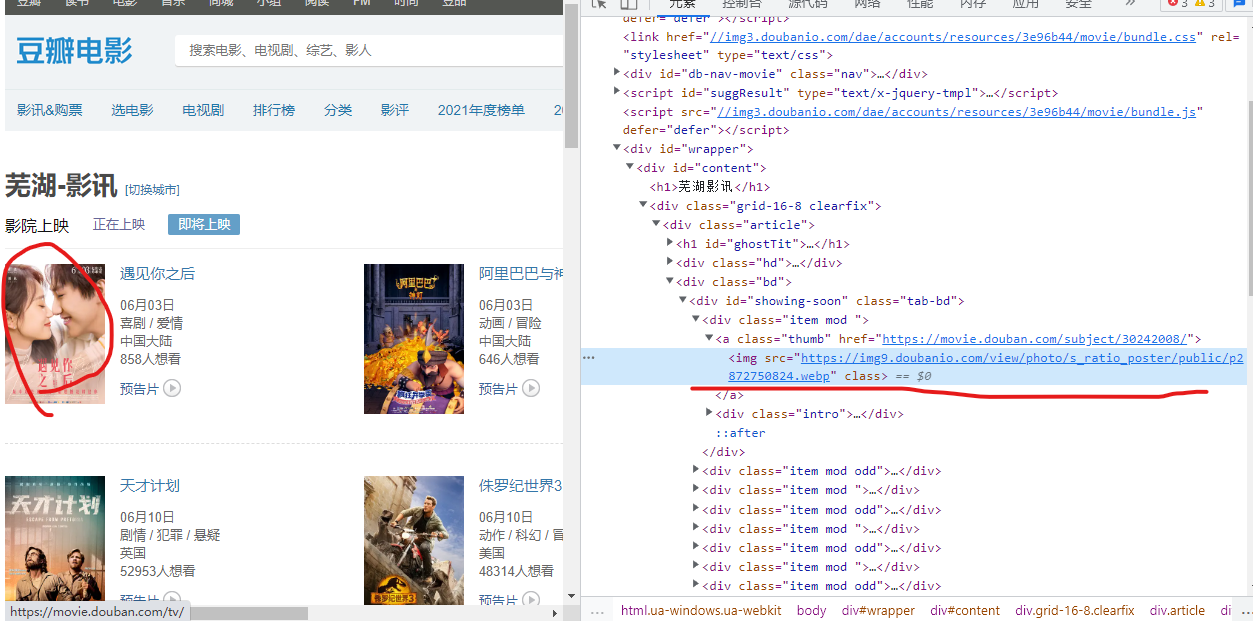


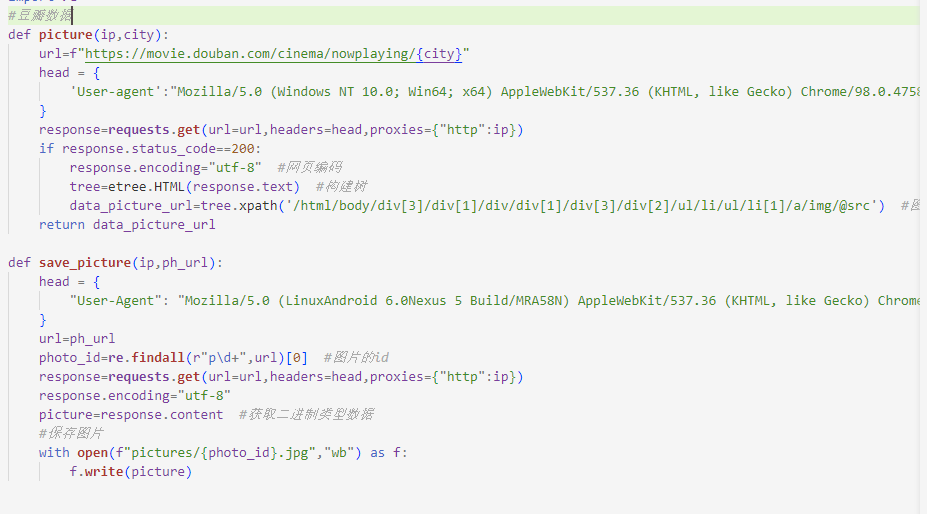
### 代码解释-基本数据



在获取到IP地址后，修改url为需要访问院线网页，通过get请求获取相应数据，利用Xpath解析器对数据进行筛选如：电影名称、评分、时长、是个有票等信息，最后将数据存储至字典中。

### 代码解释-封面爬取





通过浏览器的开发者工具确定图片的url，通过爬虫程序获取图片二进制数据，保存至对应文件夹中。

## tkinter构建GUI交互界面

### 什么是tkinter？

 Tkinter模块是Python 的标准GUI工具包的接口 .Tk 和 Tkinter 可以在大多数的Unix平台下使用,同样可以应用在Windows和Macintosh系统里。Tk8.0 的后续版本可以实现本地窗口风格,并良好地运行在绝大多数平台中。

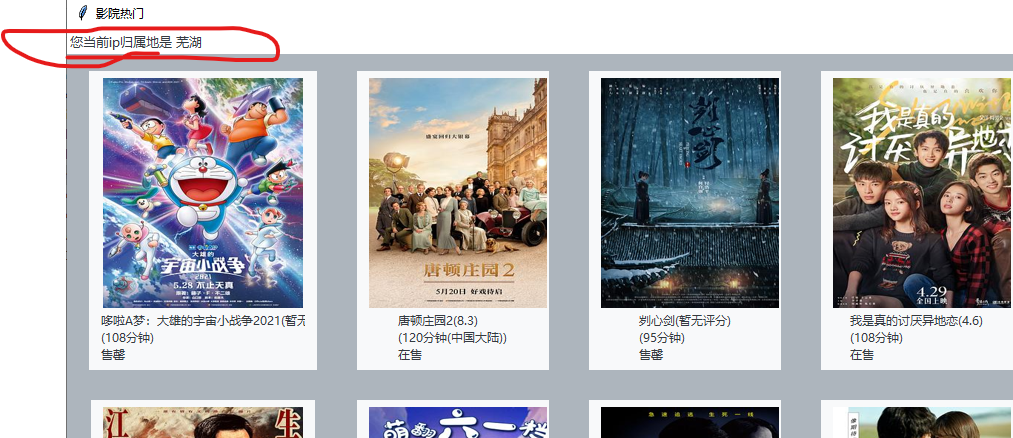
### 代码解释



为了防止图片储存错乱问题，在程序运行时利用os库会将原来的图片删除，然后创建交互界面的窗口，设置界面标题。



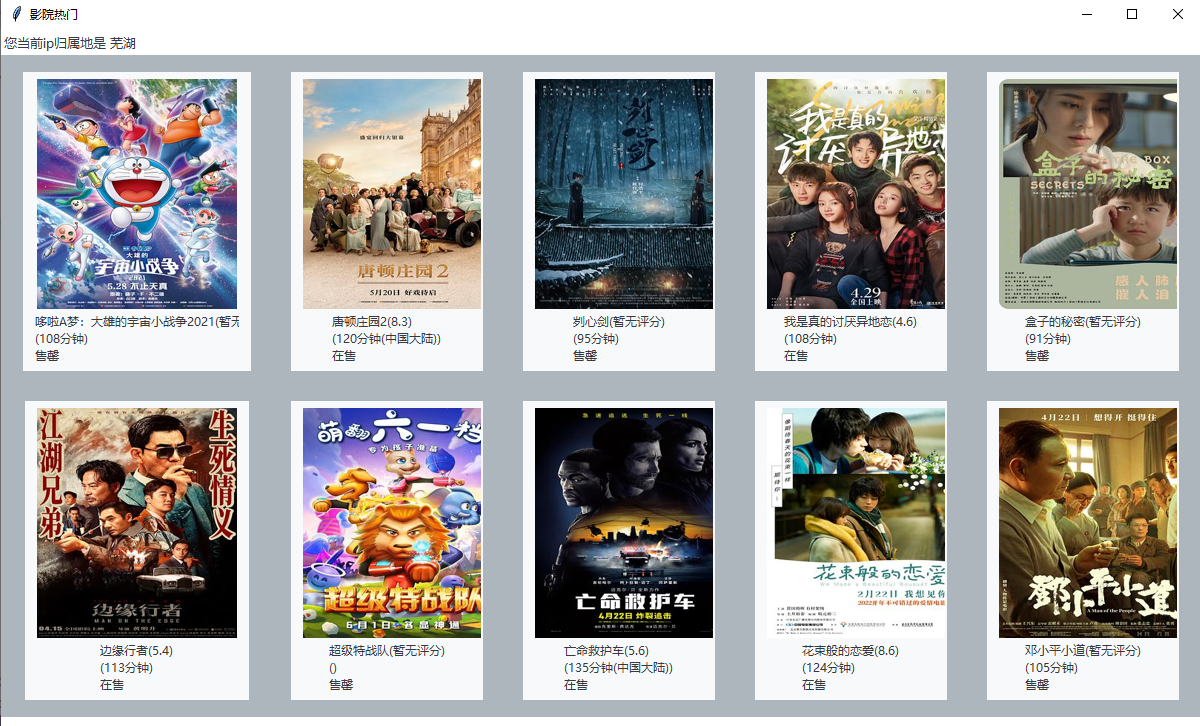
显示IP地址，将获取到的IP地址显示在交互界面的上边界，如下图4



图表 4



显示电影的基本信息，包括封面、时长、是否在售。



# 4参考文献

1. [《信息记录材料》| 2021年第007期](https://www.zhangqiaokeyan.com/academic-journal-cn_information-recording-materials_thesis/0201290519740.html)
2. [《计算机仿真》| 2021年第006期](https://www.zhangqiaokeyan.com/academic-journal-cn_computer-simulation_thesis/0201290367527.html)
3. [《智能计算机与应用》 | 2021 年第 004 期](https://www.zhangqiaokeyan.com/academic-journal-cn_intelligent-computer-applications_thesis/0201290470432.html)
4. [《现代计算机（专业版）》 | 2021 年第 005 期](https://www.zhangqiaokeyan.com/academic-journal-cn_modern-computer_thesis/0201288868997.html)
5. [《电脑知识与技术》 | 2018 年第 034 期](https://www.zhangqiaokeyan.com/academic-journal-cn_digital-community-smart-home_thesis/0201269992126.html)
6. [《软件导刊》 | 2016 年第 008 期](https://www.zhangqiaokeyan.com/academic-journal-cn_software-guide_thesis/0201242531321.html)

# 附录

## 获取IP地址

import requests

from lxml import etree

def **get\_city**():

    url = "https://ip.900cha.com/"

    head = {

        'User-agent': "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.102 Safari/537.36"

    }

    response = requests.get(url=url, headers=head)

    if response.status\_code == 200:

        response.encoding = "utf-8"  *# 网页编码*

        tree = etree.HTML(response.text)  *# 构建树*

        location = tree.xpath('/html/body/div/div/div/div[1]/div[1]/ul/li[4]/text()')

        location = location[0].strip()[-3:]  *# 具体信息*

    return location

## 构建IP代理池

import **requests**

from lxml import etree

import **random**

class **get\_ip**:

    def **\_\_init\_\_**(self):

        self.proxies\_link=[]

        self.lei1="ha"  *#高匿代理*

        self.lei2="tr"  *#普通代理*

    '''判断ip是否有效'''

    def **check\_proxy**(ip):

        try:

            url="https://www.baidu.com"

            head = {

                'User-agent':"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.102 Safari/537.36"

            }

            proxies={"http":ip}

            response=**requests**.**get**(url=url,headers=head,proxies=proxies,timeout=8)

            if response.status\_code==200:

*#print(f"{ip}有效")*

                with **open**("proxies.txt","a",encoding="utf-8",newline="") as f:  *#保存有效的ip*

                    f.**write**(ip+"\n")

                return True

        except **Exception** as e:

**print**(f"{ip}无效")

            pass

    '''爬取ip'''

    def **get\_proxy**(self,num\_page):

        try:

            for page in **range**(num\_page):

**print**(f"正在爬取{page}页")

                url="https://www.kuaidaili.com/free/in"+**str**(self.lei1)+"/"+**str**(page+1)+"/"

                head = {

                    'User-agent':"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.102 Safari/537.36"

                }

                response=**requests**.**get**(url=url,headers=head)

                if response.status\_code==200:

                    response.encoding="utf-8"

                    tree = etree.HTML(response.text)

                    ip\_list=tree.xpath('//\*[@id="list"]/table/tbody/tr/td[1]/text()')

                    for i in ip\_list:

                        if **get\_ip**.**check\_proxy**(i)==True:  *#判断ip是否有效*

                            self.proxies\_link.**append**(i)

                    return self.proxies\_link

        except **Exception**  as e:

**print**(e)

## 爬取豆瓣网院线电影信息

import **requests**

from lxml import etree

*#豆瓣数据*

def **get\_data**(ip,city):

    url=f"https://movie.douban.com/cinema/nowplaying/{city}"

    head = {

        'User-agent':"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.102 Safari/537.36"

    }

    response=**requests**.**get**(url=url,headers=head,proxies={"http":ip})

    if response.status\_code==200:

        response.encoding="utf-8"  *#网页编码*

        tree=etree.HTML(response.text)  *#构建树*

        data\_name=tree.xpath('//\*[@id="nowplaying"]/div[2]/ul/li/@data-title')  *#获取名称*

        data\_score=tree.xpath('//\*[@id="nowplaying"]/div[2]/ul/li/@data-score') *#评分*

        for i in **range**(**len**(data\_score)):

            if data\_score[i]=="0":

                data\_score[i]="暂无评分"

        data\_duration=tree.xpath('//\*[@id="nowplaying"]/div[2]/ul/li/@data-duration')  *#时长*

        data\_enough=tree.xpath('//\*[@id="nowplaying"]/div[2]/ul/li/@data-enough')  *#本地电影院是否有票*

**print**(data\_enough)

        for i in **range**(**len**(data\_enough)):

            if data\_enough[i]=="True":

                data\_enough[i]="在售"

            else:

                data\_enough[i]="售罄"

        data\_details\_url=tree.xpath('/html/body/div[3]/div[1]/div/div[1]/div[3]/div[2]/ul/li/ul/li[1]/a/@href')  *#------------------>电影详细信息的url*

*# Todo:*

        dic={

            "name":data\_name,

            "score":data\_score,

            "duration":data\_duration,

            "enough":data\_enough,

            "detail\_url":data\_details\_url,

        }

    return dic

import **requests**

from lxml import etree

import **re**

*#豆瓣数据*

def **picture**(ip,city):

    url=f"https://movie.douban.com/cinema/nowplaying/{city}"

    head = {

        'User-agent':"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.102 Safari/537.36"

    }

    response=**requests**.**get**(url=url,headers=head,proxies={"http":ip})

    if response.status\_code==200:

        response.encoding="utf-8"  *#网页编码*

        tree=etree.HTML(response.text)  *#构建树*

        data\_picture\_url=tree.xpath('/html/body/div[3]/div[1]/div/div[1]/div[3]/div[2]/ul/li/ul/li[1]/a/img/@src')  *#图片*

    return data\_picture\_url

def **save\_picture**(ip,ph\_url):

    head = {

        "User-Agent": "Mozilla/5.0 (LinuxAndroid 6.0Nexus 5 Build/MRA58N) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.212 Mobile Safari/537.36 Edg/90.0.818.62"

    }

    url=ph\_url

    photo\_id=**re**.**findall**(r"p\d+",url)[0]  *#图片的id*

    response=**requests**.**get**(url=url,headers=head,proxies={"http":ip})

    response.encoding="utf-8"

    picture=response.content  *#获取二进制类型数据*

*#保存图片*

    with **open**(f"pictures/{photo\_id}.jpg","wb") as f:

        f.**write**(picture)

## Tkinter构建GUI交互界面

import ttkbootstrap as **ttk**

from ttkbootstrap.constants import \*  *#添加主题*

from **PIL** import **Image**,**ImageTk**

import **tkinter** as **tk**

import **random**

import **city**

import **douban\_picture**

import **glob**

import **numpy** as **np**

import **douban\_data**

import **os**

pic\_dir=**os**.**listdir**("pictures")

for item in pic\_dir:

**os**.**remove**(**os**.path.**join**("pictures",item))  *#删除picture文件中的图片，更新图片*

'''创建窗口'''

root=**tk**.**Tk**()

root.title("影院热门")

root.geometry("1200x700")  *#设置窗口的大小*

'''部件'''

*#显示标题*

title\_var=**tk**.**StringVar**()

title\_var.**set**("影院热门")

*#显示ip地址*

def **show\_ip**(ip\_list):

    global location

    ip=**random**.choice(ip\_list)

    ip\_var=**tk**.**StringVar**()  *#设置变量*

    ip\_var.**set**(f"您当前ip归属地是{location}")

**tk**.**Label**(root,textvariable=ip\_var,

            anchor=NW,   *#位置在最上面*

            font=("微软雅黑", 10)  *#设置字体和大小*

            ).pack(fill=X)  *#x方向的填充*

*#显示图片和信息*

def  **show\_picture**(ip\_list):

    global photos,location

    ip=**random**.choice(ip\_list)

    picture\_url=**douban\_picture**.**picture**(**random**.choice(ip),location)  *#获取图片的url列表*

*#保存图片*

    for i in **range**(**len**(picture\_url)):

**douban\_picture**.**save\_picture**(ip,picture\_url[i])

*#读取图片*

    photos\_dir= **glob**.**glob**("pictures//\*.jpg")

    photos\_dir=**sorted**(photos\_dir,key=**os**.path.**getmtime**)  *#根据图片存储时间排序*

    photos=[**ImageTk**.**PhotoImage**(**Image**.**open**(item).**resize**((200,230)))  for item in photos\_dir]

**print**(**len**(photos))

*#photos =ImageTk.PhotoImage(get\_photos.picture())*

*#显示图片*

    container=**ttk**.Frame(master=root, padding=2,bootstyle=SECONDARY)

    container.pack(anchor=**tk**.CENTER,fill=X)

    matrix= **np**.**arange**(0,10,1).reshape(2,-1)

*#显示信息（电影名称、时长、是否邮票、评分）*

    data=**douban\_data**.**get\_data**(ip,location)

    try:

        for i in **range**(**len**(matrix)):

            for j in **range**(**len**(matrix[i])):

                index=matrix[i][j]

**print**(photos[index])

*#container.rowconfigure(matrix[i][j], weight=1)*

                container.columnconfigure(matrix[i][j], weight=1)  *#自适应改变大小,放大比例是1    ->>>>>>>*

                photo\_labels = **ttk**.Button(container, image=photos[index],

                                        compound=TOP,    *#设置文字在图片的方位*

                                        text=f"{data.**get**('name')[index]}({data.**get**('score')[index]})\n({data.**get**('duration')[index]})\n{data.**get**('enough')[index]}",

*#font=('微软雅黑',10),  #设置字体*

                                        bootstyle=(LIGHT)

                                        )

                photo\_labels.grid(row=i,column=j,padx=20,pady=15)  *#图片间距*

    except:

        pass

if \_\_name\_\_=="\_\_main\_\_":

    with **open**("proxies.txt","r") as f:

        ip\_list=f.**readlines**()

    location=**city**.**get\_city**()  *#当前电脑的ip地址*

*#显示ip地址*

**show\_ip**(**random**.choice(ip\_list))

*#显示图片、粗略电影信息*

**show\_picture**(ip\_list)

    '''显示窗口'''

    root.**mainloop**()