温州大学瓯江学院理工分院

《python 爬虫项目》 实验报告

实验名称：	实验 2 移动、云及数字化工具				
班　　级：	16 计算机 2 班	姓　　名：	喻文军	学　　号：	16219111210
实验地点：		日　　期：	2019 年 4 月 22 日		

一、实验目的：

[实验目的和要求]

1. 静态与动态爬虫
2. DJANGO
3. 12306 验证码识别登陆

二、实验环境：

1、python,pychram,mysql

三、实验内容和要求：

1.静态爬虫

2.动态爬虫

3.制作 DJANGO 网站

四、实验步骤与结果：

（对实验步骤的说明应该能够保证根据该说明即可重复完整的实验内容，得到正确结果。）

1.

步骤：

1） 确认要爬取的网站，确认 URL

2） 使用 requests 与 beautifiulsoup 库获取网页源代码

3） 使用浏览器查看源码，分析需要爬取的标题等的特征

4） 获取数据存入列表中

5） 将得到的数据存入数据库

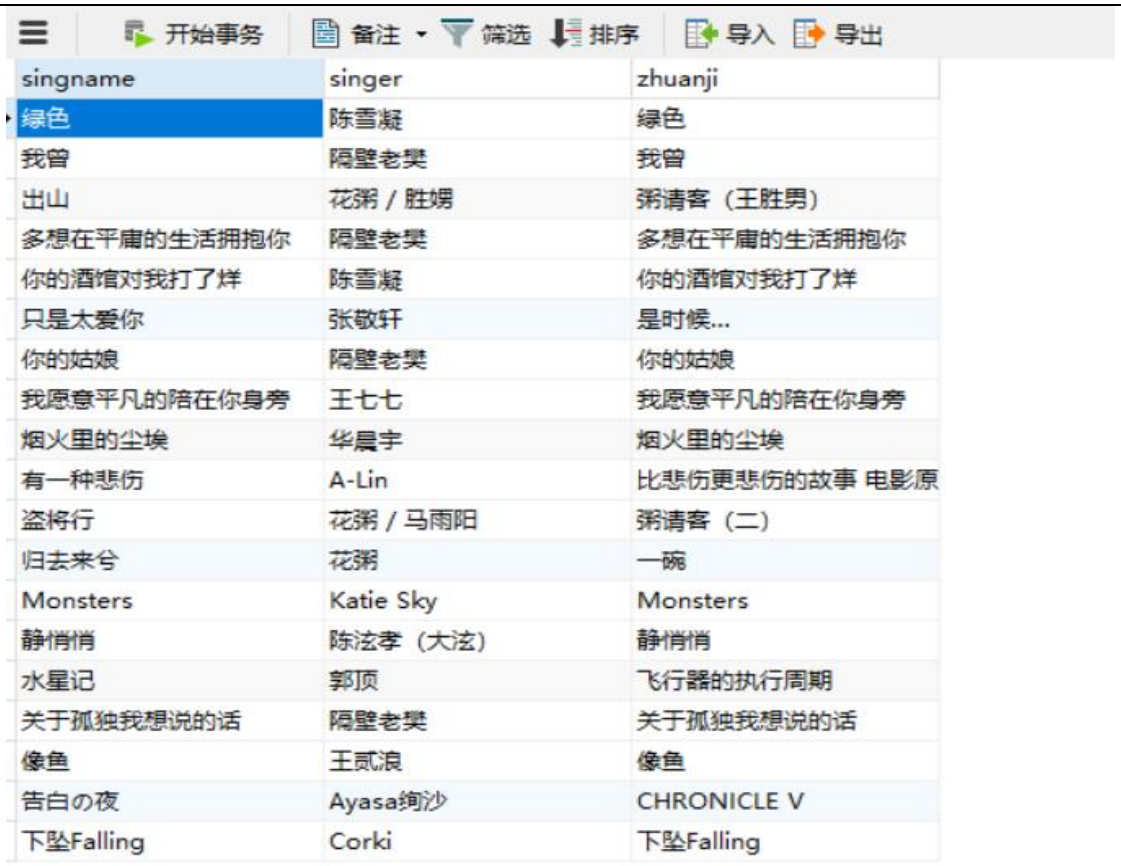
截图：

静态爬虫 （网易云热歌榜）

爬虫代码：

```
import requests
import re
from bs4 import BeautifulSoup
import pymysql
singer=[]
zhuanji=[]
```

```
dict={
    '歌名':'',
    '专辑':'',
    '歌手':''
}
def get_html(url):
    header = {
        'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.86 Safari/537.36'}
    try:
        response=requests.get(url,headers=header)
        if response.status_code==200:
            html = response.text.encode(response.encoding).decode('utf-8', 'ignore')
            return html
        if response.status_code==302:
            print('302')
    except ConnectionError:
        return get_html(url)
def get_info():
    html=get_html("http://music.163.com/discover/toplist?id=3778678")
    soup=BeautifulSoup(html,'lxml')
    tag=soup.find_all('ul',attrs=('class','f-hide'))
    singname=re.findall(r'<li><a href="/song.*?id=\d+">(.*?)</a></li>',str(tag))
    singid=re.findall(r'<li><a href="/song.*?id=(.*?)">.*?</a></li>',str(tag))
    for i in range(len(singid)):
        newhtml=get_html("https://music.163.com/song?id="+singid[i])
        newsoup=BeautifulSoup(newhtml,'lxml')
        tag2=newsoup.find_all("p",attrs=('class','des s-fc4'))
        singer.append(re.findall(r'<span title="(.*?)">',str(tag2)))
        zhuanji.append(re.findall(r'<a class="s-fc7" href="/album\?id=\d+">(.*?)</a>',str(tag2)))
        dict['专辑']=zhuanji[i]
        dict['歌手']=singer[i]
        s="".join(singname[i].split())
        dict['歌名'] =s
        get_db()
def get_db():
    conn = pymysql.connect(
        host='localhost',
        port=3306,
        user='root',
        passwd='123456',
        db='sy',
    )
    cur = conn.cursor()

    # 插入一条数据
    sql = "INSERT INTO `bbb_yinyue`(singname,singer,zhuanji) values (%s,%s,%s)"
    cur.execute(sql, [dict["歌名"], dict['歌手'], dict['专辑']])
    conn.commit()
    cur.close()
    conn.close()


def main():
    get_info()
main()
```



singname	singer	zhuanji
绿色	陈雪凝	绿色
我曾	隔壁老樊	我曾
出山	花粥 / 胜男	粥请客（王胜男）
多想在平庸的生活拥抱你	隔壁老樊	多想在平庸的生活拥抱你
你的酒馆对我打了烊	陈雪凝	你的酒馆对我打了烊
只是太爱你	张敬轩	是时候...
你的姑娘	隔壁老樊	你的姑娘
我愿意平凡的陪在你身旁	王七七	我愿意平凡的陪在你身旁
烟火里的尘埃	华晨宇	烟火里的尘埃
有一种悲伤	A-Lin	比悲伤更悲伤的故事 电影原
盗将行	花粥 / 马雨阳	粥请客（二）
归去来兮	花粥	一碗
Monsters	Katie Sky	Monsters
静悄悄	陈泫孝（大泫）	静悄悄
水星记	郭顶	飞行器的执行周期
关于孤独我想说的话	隔壁老樊	关于孤独我想说的话
像鱼	王贰浪	像鱼
告白の夜	Ayasa绚沙	CHRONICLE V
下坠Falling	Corki	下坠Falling

2. 动态网页：

动态爬虫 （淘宝手机畅销的店铺）

爬虫代码：

```
from selenium import webdriver
from lxml import etree
import time
import pymysql

driver = webdriver.Chrome()
driver.maximize_window()
if __name__ == "__main__":
    url = 'https://login.taobao.com/member/login.jhtml'
    driver.get(url)
    driver.implicitly_wait(10)

    # 让程序休眠10 秒，在这期间，弹出登录界面之后，使用你的手机扫码登录淘宝
    time.sleep(10)
    # 定位搜索框，并将其清除
    driver.find_element_by_id('q').clear()
    # 在搜索框内输入“充电宝”
    driver.find_element_by_id('q').send_keys('手机')
    # 休息两秒，免得被发现为爬虫
    time.sleep(2)
    # 点击搜索按钮
    driver.find_element_by_xpath('//*[@id="J_TSearchForm"]/div[1]/button').click()
    # 休息两秒
    time.sleep(2)
```

```
    # 点击来源于“ 天猫 ” 按钮
    driver.find_element_by_xpath('//*[@id="tabFilterMall"]').click()
    # 休息两秒
    time.sleep(2)
    # 点击 “ 销量最高按钮 ”
    driver.find_element_by_xpath('//*[@id="J_relative"]/div[1]/div/ul/li[2]/a').click()
    # 休息两秒
    time.sleep(2)
    # 打印当前页面的 URL
    print(driver.current_url)
    # 解析网页
    html = etree.HTML(driver.page_source)
    # 利用 xpath 寻找大循环。
    items = html.xpath('//div[@class="item J_MouserOnverReq    "]')

    la = []
    for item in items:
        # 利用 xpath 进行小循环，打印销量排名靠前的店家名称
        shop = item.xpath('div[2]/div[3]/div[1]/a/span[2]/text()')[0]
        #将数据加到 la 列表
        la.append(shop)
    #将列表 la 中重复的数据去除
    def deleteDuplicatedElementFromList(listA):
        return sorted(set(listA), key=listA.index)
    a = deleteDuplicatedElementFromList(la)
    print(a)
    for i in range(len(a)):
        conn = pymysql.connect(host='localhost', user='root', password='123456', db='sy', charset="utf8")
        cur = conn.cursor()
        sql = "insert into bbb_shouji (name)values('{0}')".format(a[i])
        cur.execute(sql)
        conn.commit()
        cur.close()
        conn.close()
```

name
小米官方旗舰店
荣耀官方旗舰店
华为官方旗舰店
苏宁易购官方旗舰店
vivo官方旗舰店
oppo官方旗舰店
能良数码官方旗舰店
三际数码官方旗舰店
魅族官方旗舰店
三星官方旗舰店
中国移动官方旗舰店
中国电信官方旗舰店
美图旗舰店
vivo欧曙专卖店
广东电信亿品汇专卖店
茂鹏数码专营店
vivo航鹰专卖店
久汇数码专营店
绿森数码官方旗舰店

3.django

步骤：

1） 使用 pycharm 创建项目

2） 在 setting.py 文件中通过 DATABASES 选项进行数据库配置，在_init_.py 中配置

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',    # 数据库引擎
        'NAME': 'sy',           # 你要存储数据的库名，事先要创建之
        'USER': 'root',         # 数据库用户名
        'PASSWORD': '123456',     # 密码
        'HOST': 'localhost',    # 主机
        'PORT': '3306',         # 数据库使用的端口
    }
}
```

```
import pymysql
pymysql.install_as_MySQLdb()
```

3） 在 urls.py 中写两个 PATH 一个跳转到网易云，一个跳转到淘宝

```
"""
from django.contrib import admin
from django.urls import path
from bbb import views
from django.conf.urls import url,include

urlpatterns = [
    path('admin/',admin.site.urls),
    path('index/',views.index),
    path('taobao/',views.taobao),
]
```

4） 在 models.py 中建立两张表

```python
from django.db import models

# Create your models here.
class Yinyue(models.Model):
    singname=models.CharField(max_length=150)
    singer=models.CharField(max_length=150)
    zhuanji=models.CharField(max_length=150)

class Shouji(models.Model):
    name=models.CharField(max_length=150)
```

然后在数据库中生成数据表，进行数据迁移

Python manage.py makemigrations

Python manage.py migrate

5） 编写 views.py

Views.py

```python
from django.shortcuts import render
import pymysql
from bbb.models import Yinyue
from bbb.models import Shouji
from django.shortcuts import HttpResponse
from django.core.paginator import Paginator,PageNotAnInteger,EmptyPage


def index(request):
        yinyue_list = Yinyue.objects.all().order_by("id")  # 一定要排序
        paginator = Paginator(yinyue_list, 12)  # 每页 15 条记录
        page = request.GET.get('page')  # 获取第一页数据，从 1 开始
        try:
            customer=paginator.page(page)
        except PageNotAnInteger:
            customer=paginator.page(1)
        except EmptyPage:
            customer=paginator.page(paginator.num_pages)

        return render(request, 'index.html',{"yinyue_list":customer} )

def taobao(request):
    shouji_list = Shouji.objects.all().order_by("id")  # 一定要排序
    paginator = Paginator(shouji_list, 12)  # 每页 15 条记录
    page = request.GET.get('page')  # 获取第一页数据，从 1 开始
    try:
        customer = paginator.page(page)
    except PageNotAnInteger:
        customer = paginator.page(1)
    except EmptyPage:
        customer = paginator.page(paginator.num_pages)
```

```
        return render(request, 'taobao.html', {"shouji_list": customer})
```

6） 再详细页中使用获取的字典，显示数据

html 文件

index.html(网易云)

```
{% load staticfiles %}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>

    <link href="{% static 'css/bootstrap.min.css' %}" rel="stylesheet">
    <link href="{% static 'css/style.css' %}" rel="stylesheet">
</head>

<body>
    <div class="container" style="margin-top:50px;">
     <h1 align="center"><b>云音乐热歌榜</b></h1>
        <table class="table">
            <thead class="thead-dark">
                <tr>
                    <th scope="col">排名</th>
                    <th scope="col">歌曲名称</th>
                    <th scope="col">歌手</th>
                    <th scope="col">专辑</th>
                </tr>
            </thead>
            {% for music in yinyue_list %}
            <tr>
                <td scope="row">{{music.id}}</td>
                <td>{{music.singname}}</td>
                <td>{{music.singer}}</td>
                <td>{{music.zhuanji}}</td>
            </tr>
            {% endfor %}
        </table>

        <div class="row justify-content-center">
            <ul class="pagination">
                {% for pg in yinyue_list.paginator.page_range %}
                    {% if yinyue_list.number == pg %}
                        <li class="page-item"><a class="page-link" href="?page={{ pg }}">{{ pg }}</a></li>
                    {% else %}
                        <li><a class="page-link" href="?page={{ pg }}">{{ pg }}</a></li>
                    {% endif %}
                {% endfor %}
            </ul>
```

```
        </div>

    </div>
    <link href="{% static 'js/jquery-3.3.1.min.js' %}" rel="stylesheet">
    <link href="{% static 'js/bootstrap.min.js' %}" rel="stylesheet">
</body>
</html>
```

taobao.html（淘宝界面）

```
{% load staticfiles %}
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <title>taobao</title>

    <link href="{% static 'css/bootstrap.min.css' %}" rel="stylesheet">
    <link href="{% static 'css/style.css' %}" rel="stylesheet">
</head>

<body>
    <div class="container" style="margin-top:50px;">
    <h1 align="center"><b>手机销量店铺排名</b></h1>
        <table class="table">
            <thead class="thead-dark">
                <tr>
                    <th scope="col">排名</th>
                    <th scope="col">店铺名</th>
                </tr>
            </thead>
            {% for i in shouji_list %}
            <tr>
                <td scope="row">{{i.id}}</td>
                <td>{{i.name}}</td>
```

```
            </tr>
            {% endfor %}
        </table>

        <div class="row justify-content-center">
            <ul class="pagination">
                {% for pg in shouji_list.paginator.page_range %}
                    {% if shouji_list.number == pg %}
                        <li class="page-item"><a class="page-link" href="?page={{ pg }}">{{ pg }}</a></li>
                    {% else %}
                        <li><a class="page-link" href="?page={{ pg }}">{{ pg }}</a></li>
                    {% endif %}
                {% endfor %}
            </ul>
        </div>
    </div>
    <link href="{% static 'js/jquery-3.3.1.min.js' %}" rel="stylesheet">
    <link href="{% static 'js/bootstrap.min.js' %}" rel="stylesheet">
</body>

</html>
```

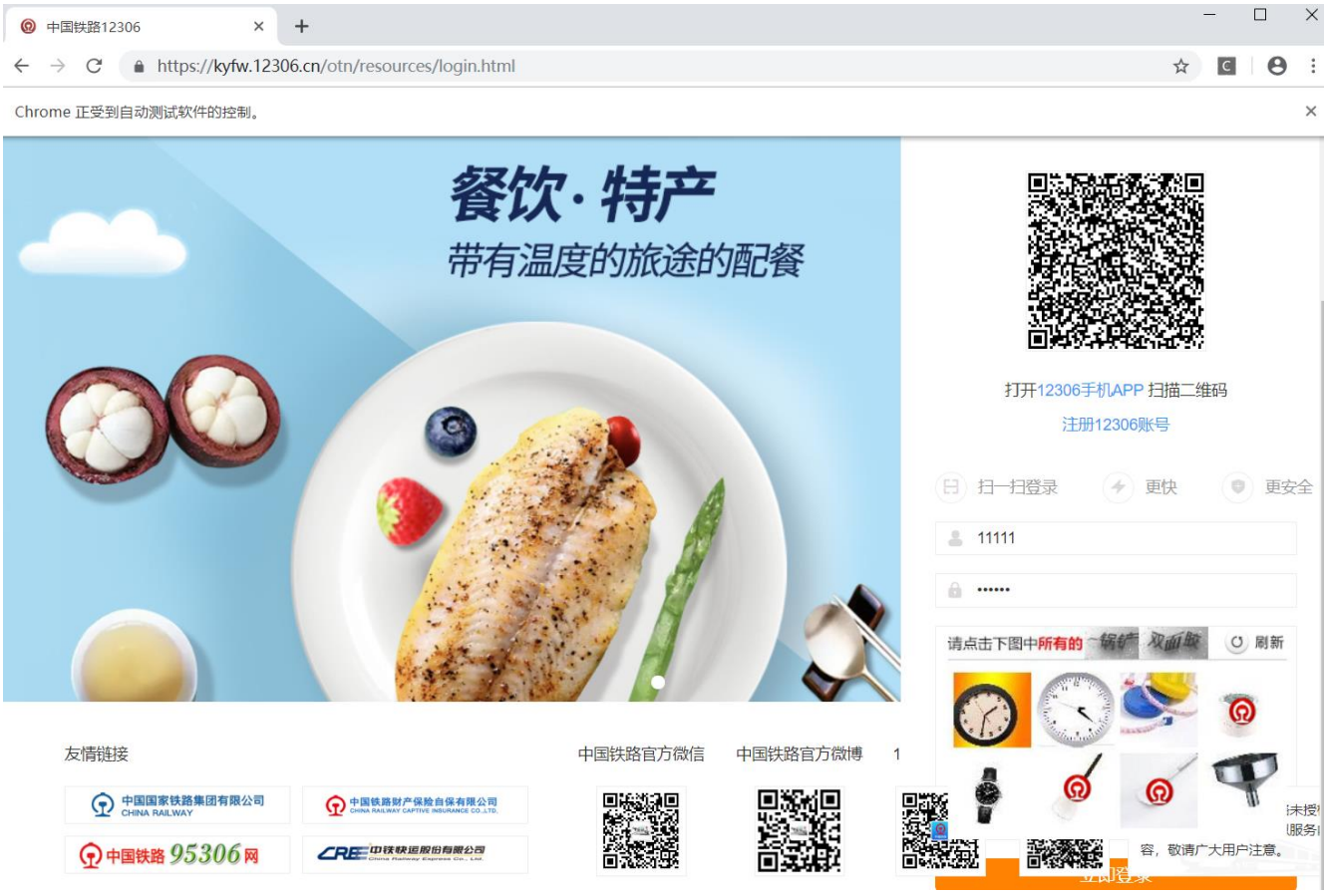
12306 验证码识别登陆

代码

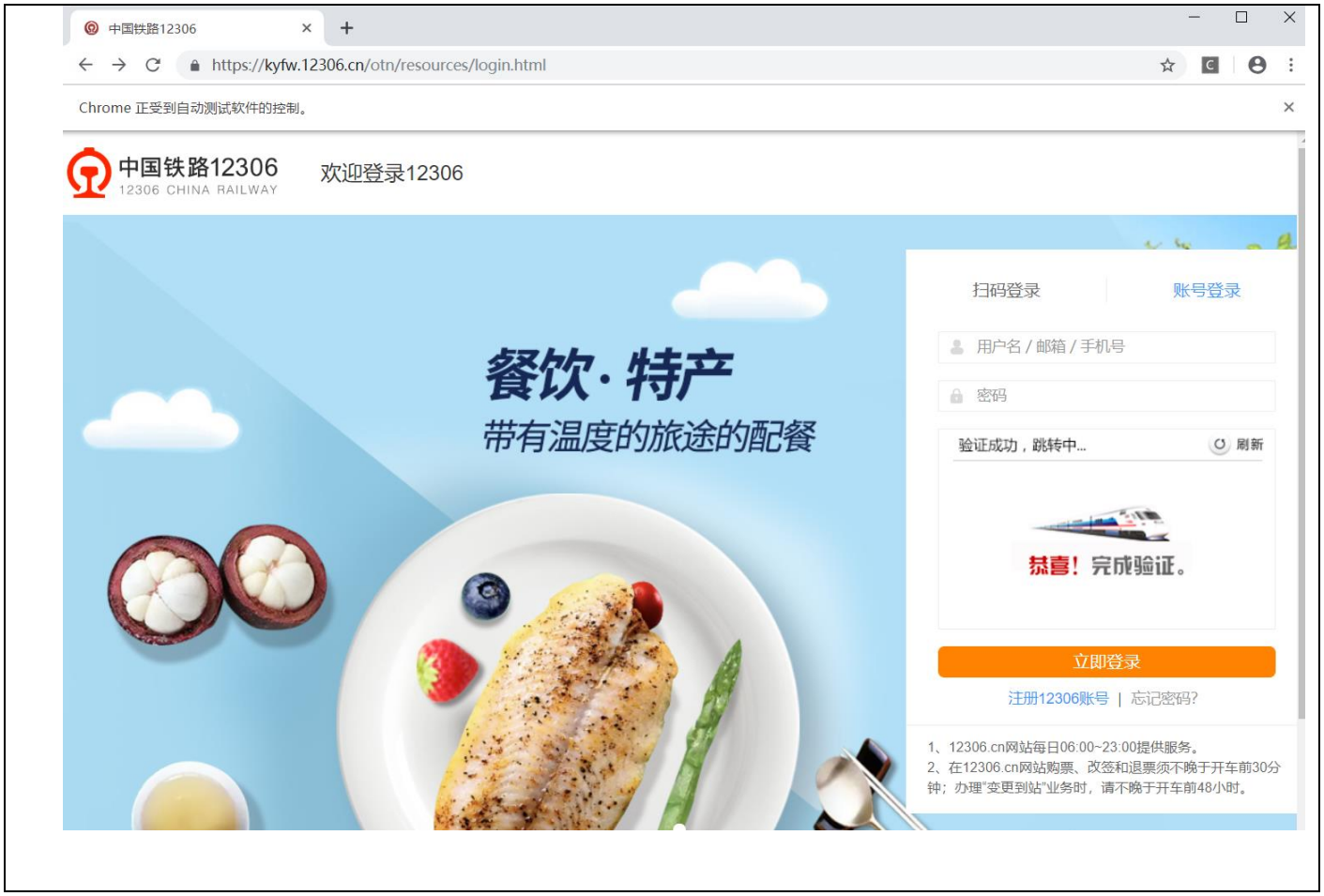
```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.action_chains import ActionChains
import requests
```

```
import base64
import re
import time

class Demo():
    def __init__(self,username,password):
        self.coordinate=[[-105,-20],[-35,-20],[40,-20],[110,-20],[-105,50],[-35,50],[40,50],[110,50]]
        self.username=username
        self.password=password
    def login(self):
        login_url="https://kyfw.12306.cn/otn/resources/login.html"
        driver = webdriver.Chrome()
        driver.set_window_size(1200, 900)
        driver.get(login_url)
        account=driver.find_element_by_class_name("login-hd-account")
        account.click()
        userName=driver.find_element_by_id("J-userName")
        userName.send_keys(self.username)
        password=driver.find_element_by_id("J-password")
        password.send_keys(self.password)
        self.driver=driver
    def getVerifyImage(self):
        try:

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

        except Exception as e:
            print(u"网络开小差,请稍后尝试")
        base64_str=img_element.get_attribute("src").split(",")[-1]
        imgdata=base64.b64decode(base64_str)
        with open('verify.jpg','wb') as file:
            file.write(imgdata)
        self.img_element=img_element
    def getVerifyResult(self):
        url="http://littlebigluo.qicp.net:47720/"
        response=requests.request("POST",url,data={"type":"1"},files={'pic_xxfile':open('verify.jpg','rb')})
        result=[]
        print(response.text)
        for i in re.findall("<B>(.*)</B>",response.text)[0].split(" "):
            result.append(int(i)-1)
        self.result=result
        print(result)
    def moveAndClick(self):
        try:
            Action=ActionChains(self.driver)
            for i in self.result:

Action.move_to_element(self.img_element).move_by_offset(self.coordinate[i][0],self.coordinate[i][1]).click()
```

```
            Action.perform()
        except Exception as e:
            print(e.message())
    def submit(self):
        self.driver.find_element_by_id("J-login").click()
    def __call__(self):
        self.login()
        time.sleep(3)
        self.getVerifyImage()
        time.sleep(1)
        self.getVerifyResult()
        time.sleep(1)
        self.moveAndClick()
        time.sleep(1)
        self.submit()
        time.sleep(10000)


Demo('11111','222222')()
```




```
<!DOCTYPE html>
<title>12306图片验证码破解</title>
<h1>请上传一张12306验证码图片</h1>
<form method=post enctype=multipart/form-data>
    <input type=file name=pic_xxfile>
    <input type=submit value=上传>
</form>
<br><img src=/upload/2019-06-18-22-28-34_6673_verify.jpg><p>经过仔细揣摩-图片貌似选:  <font color="red"><font size="+2"><B>1 2 6</B></font></font></p><p><font size="1">第一排图片从左到右编号依
```


中国铁路12306
https://kyfw.12306.cn/otn/resources/login.html
Chrome 正受到自动测试软件的控制。
中国铁路12306
12306 CHINA RAILWAY
欢迎登录12306
餐饮·特产
带有温度的旅途的配餐
扫码登录
账号登录
用户名 / 邮箱 / 手机号
密码
验证成功，跳转中...
刷新
恭喜！完成验证。
立即登录
注册12306账号 | 忘记密码?
1、12306.cn网站每日06:00~23:00提供服务。
2、在12306.cn网站购票、改签和退票须不晚于开车前30分钟；办理"变更到站"业务时，请不晚于开车前48小时。