Python3 实现淘女郎照片爬虫

一、实验介绍

1.1 实验内容

本项目通过使用 Python 实现一个淘女郎图片收集爬虫，学习并实践 BeautifulSoup、Selenium Webdriver 及正则表达式等知识。在项目开发过程中采用瀑布流开发模型。

1.2 实验知识点

- Python3
- BeautifulSoup
- Selenium Webdriver
- 正则表达式

1.3 实验环境

- python3
- Xfce终端

1.4 适合人群

本课程难度为一般，属于初级级别课程，适合具有Python基础的用户，熟悉python基础知识加深巩固。

1.5 代码获取

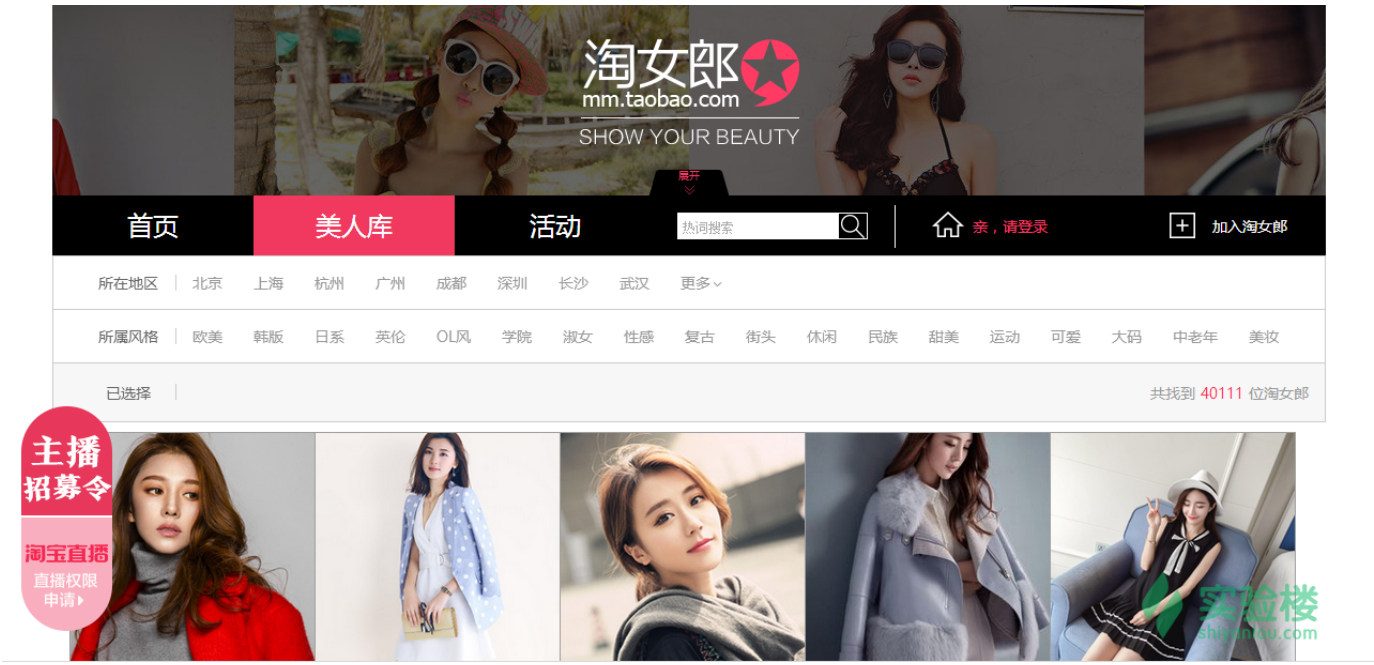
你可以通过下面命令将代码下载到实验楼环境中，作为参照对比进行学习。

```
$ wget http://labfile.oss.aliyuncs.com/courses/595/crawler.py
```

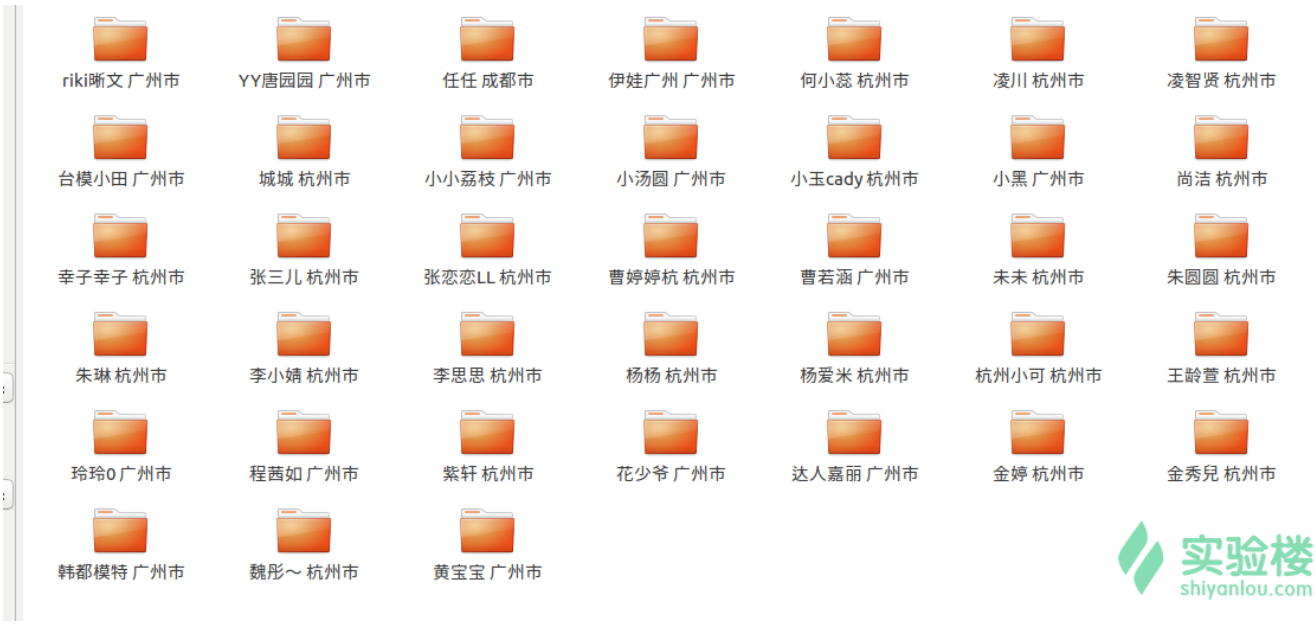
1.6 实验效果

这是我们要爬取的目标页面：

淘女郎：https://mm.taobao.com/search_tstar_model.htm (https://mm.taobao.com/search_tstar_model.htm?)

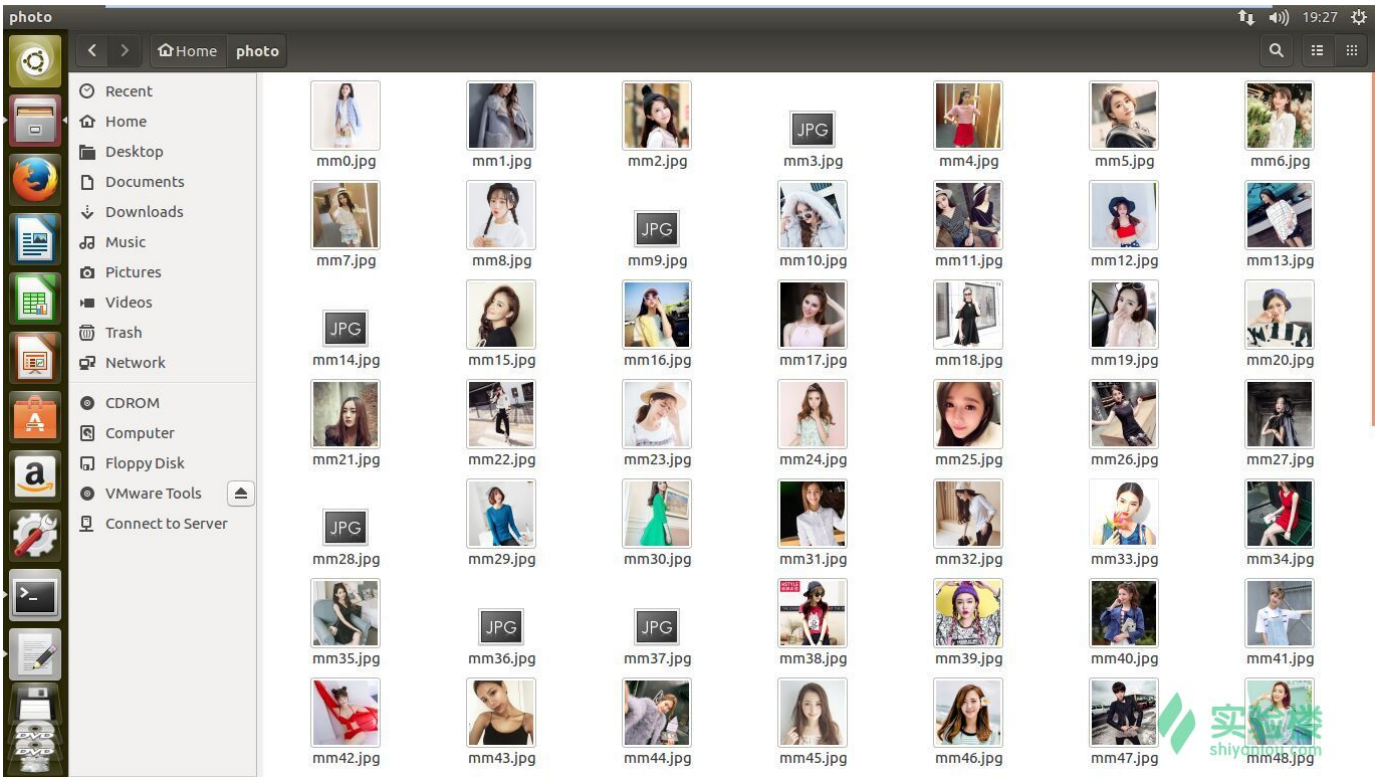


爬取后的目录结构如下：



每个目录中都有一系列的图片：

动手实践是学习 IT 技术最有效的方式！ 开始实验



二、开发准备

本节主要介绍和安装项目中将用到的几个基础工具。本实验使用实验楼的环境开发，中间部分步骤在不同版本的 Linux 环境下会有不同。

在开始之后的步骤之前，先更新源：

```
sudo apt-get update
```

2.1 安装 pip3

首先，由于使用的工具都需要通过 pip3 进行安装，实验楼的环境中没有安装 pip3（现在已经有了，可以跳过此步），所以需要先将pip3准备好。

打开桌面上的 Xfce 终端，输入下面的命令安装 pip3：

```
sudo apt-get install python3-pip
```

2.2 安装 BeatifulSoup

简介

BeautifulSoup 库的名字取自刘易斯·卡罗尔在《爱丽丝梦游仙境》里的同名歌词。就像故事中他在仙境中的说法一样， BeautifulSoup 试图化平淡为神奇。它通过定位 HTML 标签来去格式化和组织复杂的网络信息，用简单易用的 Python 对象为我们展现 XML 结构信息。

安装

由于这次实验是在 python3.X 版本以上的所以，将拓展库安装到特定的库中使用 pip3 ，从而安装到 python3 的系统目录中，仍然是在实验楼中的 Xfce 终端执行命令：

```
sudo pip3 install beautifulsoup4
```

BeautifulSoup4 是现今的最新版本，也是接下来重点使用的工具。

此外，项目中 beautifulsoup 还会用到 html5lib 这个模块，所以也需要安装 html5lib：

```
sudo apt-get install python3-html5lib
```

2.3 Selenium

简介

Selenium (http://www.seleniumhq.org/) 是一个强大的网络数据采集工具，最初是为网站自动化测试而开发的。近几年，他还被广泛用于获取精确的网站快照，因为他们可以直接运行在浏览器上。 Selenium 可以让浏览器自动加载页面，获取需要的数据，甚至页面截屏，或者判断网站上某些动作上是否发生。

Selenium 自己不带浏览器，它需要与第三方浏览器结合在一起使用。例如，可以在实验楼桌面上的 Firefox 浏览器上运行 Selenium ，可以直接看到一个 FireFox 窗口被打开，进入网站，然后执行你在代码中设置的动作。虽然使用 Firefox 浏览器看起来更清楚，但在本实验中我们采用 PhantomJS (http://phantomjs.org/download.html)来代替真实的浏览器结合使用。

安装

```
sudo pip3 install selenium
```

测试是否都安装成功：

```
root@fcf600f8d9e4:/home/shiyanlou# python3
Python 3.4.3 (default, Oct 14 2015, 20:28:29)
[GCC 4.8.4] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from bs4 import BeautifulSoup
>>> from selenium import webdriver
>>>
```


2.4 PhantomJS

简介

一个 无头 的浏览器， PhantomJS 会把网站加载到内存并执行页面上的 JavaScript ，但是不会向用户展示网页的图形化界面，可以用来处理 cookie 、 JavaScript 及 header 信息，以及任何你需要浏览器协助完成的事情。

安装

我们从阿里的镜像下载包，然后解压到你喜欢的目录，这里我们解压到 /opt/ 。

```
wget https://npm.taobao.org/mirrors/phantomjs/phantomjs-2.1.1-linux-x86_64.tar.bz2
sudo tar xjf phantomjs-2.1.1-linux-x86_64.tar.bz2 -C /opt/
```

2.5 Ajax 信息加载

现在有很多页面都是采用 Ajax 加载数据，我们实验的目标网址也是这样的：淘女郎 (https://mm.taobao.com/search_tstar_model.htm?)

如果我们用传统的方法采集这个页面，只能获取加载前的页面，而我们真正需要的信息（ Ajax 执行之后的页面）却抓不到，后续实验中可以看到效果的区别。

三、实验步骤

3.1 实验目标

本节实验中我们将分别按照如下步骤：

1. 抓取淘宝MM 的姓名，封面图片，年龄、所在城市等信息
2. 抓取每一个MM个人主页内的写真图片
3. 把每一个MM的写真图片按照文件夹保存到本地

3.2 可行性分析

淘女郎首页上的源码信息是公开的，本次实验仅仅是用来技术实践，并不带盈利性目的，也不将图片用于其他商业环境，并不会产生商业上的产权纠纷，所以这个项目是可行的。

3.3 程序结构

1. 遍历淘女郎主页上所有 MM
2. 抓取各个 MM 的姓名，封面图片，年龄、所在城市等信息
3. 遍历MM 个人主页内的所有写真图片
4. 把每 MM 的写真图片按照文件夹保存到本地

3.4 流程说明

1. 通过 Selenium Webdriver 获得目标页面源码，之后通过 BeautifulSoup 解析概源码，通过正则表达式提取出模特名字、所在城市、身高、体重，个人主页、封面图片地址等信息，根据模特名字和城市建立文件夹。
2. 再次通过 Selenium Webdriver 获得模特个人主页的页面源码，之后通过 BeautifulSoup 解析源码，通过正则获得页面艺术照的URL地址信息。
3. 最后通过 urllib 内置库，打开图片地址，通过二进制读写的方式获得模特艺术照，并将艺术照存在相应文件夹里面。

3.5 获取信息模块实现

获得页面源码

最简单的查看网页源码的方式就是在浏览器中右键选择审查元素，其他类型浏览器也是类似的：

而 Python 代码中的实现则是调用 Selenium Webdriver 和 PhantomJS 来模拟打开该页面源码，最后使用 BeautifulSoup 进行解析。

注意实验的时候，代码先不要直接写在脚本文件里，可以在交互模式的解释器里尝试一下代码，试着了解下运行原理。

我们先导入相关模块，然后设置一些变量（浏览器路径，起始页，输出目录，解析器名称）：

```
import os
import threading
import re
from bs4 import BeautifulSoup
from urllib.request import urlopen
from selenium import webdriver

browserPath = '/opt/phantomjs-2.1.1-linux-x86_64/bin/phantomjs'
homePage = 'https://mm.taobao.com/search_tstar_model.htm?'
outputDir = 'photo/'
parser = 'html5lib'
```

现在我们来看看怎样模拟浏览器查看源码：

```
driver = webdriver.PhantomJS(executable_path=browserPath)  #浏览器的地址
driver.get(homePage)  #访问目标网页地址
bsObj = BeautifulSoup(driver.page_source, parser)  #解析目标网页的 Html 源码
```

这个过程就相当于右键的点击审查的过程。

```
driver = webdriver.PhantomJS(executable_path=browserPath)
```

这里的意思是实例化一个 PhantomJS 浏览器对象，括号里面填的是浏览器的安装路径信息，填在单引号里面。 selenium 支持的浏览器类型有 chrome 、 Firefox 等，具体的自行查看 webdriver 的 API。

```
bsObj=BeautifulSoup(driver.page_source,parser)
```

这里的 driver.page_source 意思是网页的全部 HTML 源码，包含的具体内容，可以通过 print(driver.page_source) 打印查看。

获得MM个人信息

```
girlsList = driver.find_element_by_id('J_GirlsList').text.split(
        '\n')  #获得主页上所有妹子的姓名、所在城市、身高、体重等信息
```

上面的截图可以发现，整个图片层次是在 <ul class="girls-list clearfix" id="J_GirlsList"> 里面的通过 J_GirlsList 定位到这个层次，属性 text 包含网页中所有 HTML 标签的内容，类型为字符串，我们可以看看 text 里的东西是什么：

```
print(driver.find_element_by_id('J_GirlsList').text)
```

而后面的 split('\n') 则会将属性 text 中的字符串以换行符分割，得到一个包含所有分割后的字符串的列表。

获得 MM 个人主页地址


```
girlsUrl = bsObj.find_all("a",{"href": re.compile("\/\/.*\.htm\?(userId=)\d*")})  #解析出妹子的个人主页地址等信息
```

BeautifulSoup 的具体内容我这里不会讲深入的，想详细了解的，可以去他们的官网查阅API，用到的方法稍后会进行分析。

find_all 方法可以获得所有的你想通过定位获得的信息，可以使用 xml、xPath、正则表达式等语言来进行定位。

```
re.compile("\/\/.*\.htm\?(userId=)\d*")
```

这里双引号里面各种斜杆和反斜杆的符号就是正则表达式，专门用来的做信息配对的，Python 的正则匹配引擎，有很多东西可以研究，大家有兴趣的话可以学习实验楼的正则表达式 (https://www.shiyanlou.com/courses/90)课程。

获得 MM 封面图片地址

```
imagesUrl = re.findall('\/\/gtd\.alicdn\.com\/sns_logo.*\.jpg',
                       driver.page_source)  #获取所有妹子的封面图片
```

我们可以发现，妹子的封面图片的 **url** 都是形如 `//gtd.alicdn.com/sns_logo/xx/xxxxxx.jpg` 这样的字符串，这样我们就可以用正则表达式匹配它。

建立相应文件夹

本部分代码用来创建保存照片的目录结构：

```
def mkdir(path):
    # 判断路径是否存在
    isExists = os.path.exists(path)
    # 判断结果
    if not isExists:
        # 如果不存在则创建目录
        print("     [*]新建了文件夹", path)
        # 创建目录操作函数
        os.makedirs(path)
    else:
        # 如果目录存在则不创建，并提示目录已存在
        print('     [+]文件夹', path, '已创建')
```

这里是一些基本的文件操作，判断文件夹是否存在，存在则不创建，不存在就创建文件夹。

获得MM个人页面源码

这里前面的操作一样的原理：

```
driver = webdriver.PhantomJS(executable_path=browserPath)
driver.get(url)
bsObj = BeautifulSoup(driver.page_source, parser)
```

3.6 图片存储模块实现

存储封面图片

下面的代码用于存储主页的封面图片（`girlCover` 是封面图片 **url** 地址）：

```
data = urlopen(girlCover).read()
with open(outputDir + girlNL + '/cover.jpg', 'wb') as f:
    f.write(data)
```

`urlopen()` 打开图片的**URL**地址，然后使用 `read()` 方法读取图片的二进制数据。

存储个人艺术照

下面的代码用来存储个人艺术照片：

```
imgs = bsObj.find_all("img", {"src": re.compile(".*\.jpg")})
for i, img in enumerate(imgs[1:]):
    html = urlopen('https:' + img['src'])
    data = html.read()
    fileName = "{}/{}.jpg".format(path, i + 1)
    print("    [+]Loading...", fileName)
    with open(fileName, 'wb') as f:
        f.write(data)
```

`html = urlopen('https:' + img['src'])` 这么做的原因是：我们获得的网址信息是不完整的，我们需要手动补充完整。

3.7 组装及调试

前面已经所有模块拆分并解释得很详细，现在应该是从全局的角度审视这个项目，然后增加一些异常处理、存储操作，以增加软件运行的健壮性。

整合数据

```
# 所有妹子的名字地点
girlsNL = girlsList[::3]
# 所有妹子的身高体重
girlsHW = girlsList[1::3]
# 所有妹子的个人主页地址
girlsHURL = [('http:' + i['href']) for i in girlsUrl]
# 所有妹子的封面图片地址
girlsPhotoURL = [('https:' + i) for i in imagesUrl]

girlsInfo = zip(girlsNL, girlsHW, girlsHURL, girlsPhotoURL)
```

3.8 组装各个模块

```
for girlNL, girlHW, girlHURL, girlCover in girlsInfo:
    print("[*]Girl :", girlNL, girlHW)
    # 为妹子建立文件夹
    mkdir(outputDir + girlNL)
    print("    [*]saving...")
    # 获取妹子封面图片
    data = urlopen(girlCover).read()
    with open(outputDir + girlNL + '/cover.jpg', 'wb') as f:
        f.write(data)
    print("    [+]Loading Cover... ")
    # 获取妹子个人主页中的图片
    getImgs(girlHURL, outputDir + girlNL)
```

将个人网页上的图片存储到相应的个人文件夹中

即之前调用过的 `getImgs()` 函数

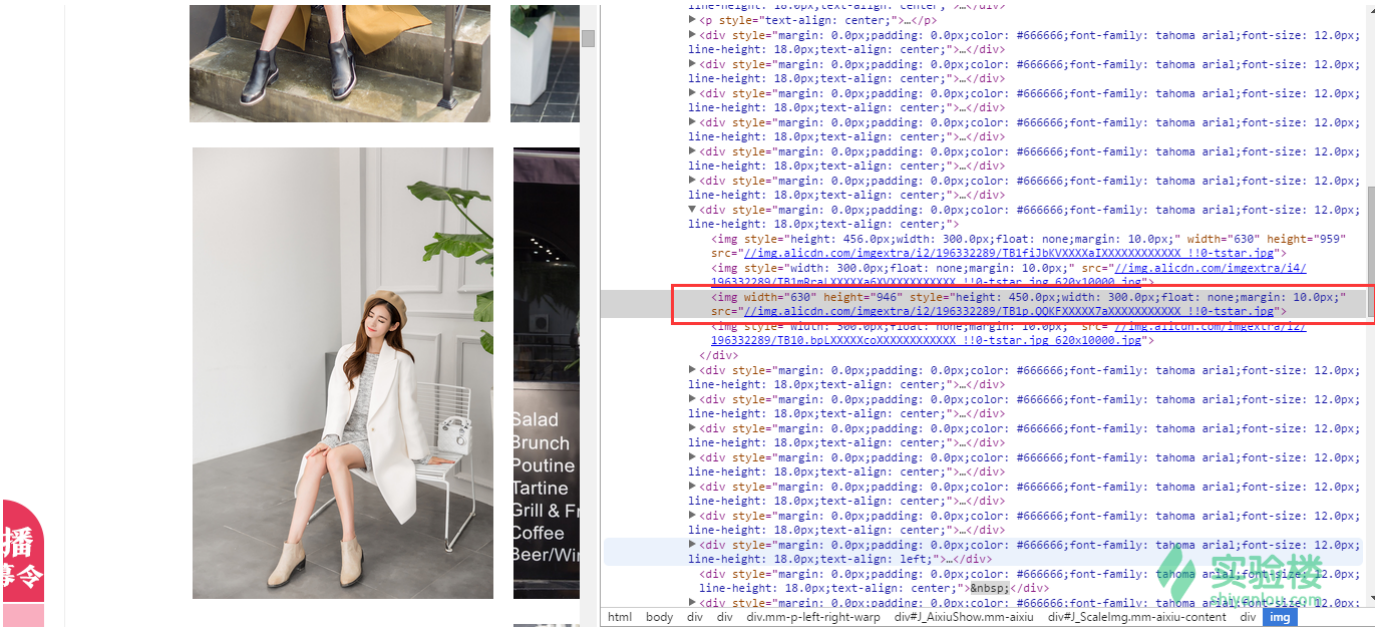
```
def getImgs(url, path):
    driver = webdriver.PhantomJS(executable_path=browserPath)
    driver.get(url)
    print("    [*]Opening...")
    bsObj = BeautifulSoup(driver.page_source, parser)
    #获得模特个人页面上的艺术照地址
    imgs = bsObj.find_all("img", {"src": re.compile(".*\.jpg")})
    for i, img in enumerate(imgs[1:]):  #不包含与封面图片一样的头像
        try:
            html = urlopen('https:' + img['src'])
            data = html.read()
            fileName = "{}/{}.jpg".format(path, i + 1)
            print("    [+]Loading...", fileName)
            with open(fileName, 'wb') as f:
                f.write(data)
        except Exception:
            print("    [!]Address Error!")
    driver.close()
```

`for i, img in enumerate(imgs[1:])`: 这一行为什么使用 `imgs[1:]` 而不是直接使用 `imgs`，因为每个 MM 的个人页面的第一个图片必定是头像，而头像却是与封面图片一样，分辨率还小了许多，就没必要抓取了：

另外注意两个地方：

1.正则表达式部分

```
imgs = bsObj.find_all("img", {"src": re.compile(".*\.jpg")})
```



用 `".*\.jpg"` 匹配任意以 '.jpg' 结尾的字符串。

2.异常处理

有时候解析出来的图片 url 会有问题，或者解析出错之类的，但我们程序没有必要因为个别错误抛出的异常而被终止。

所以我们应该做一下异常处理，使用 `try ... except ...` 异常处理语句。

`try` 里面是执行语句块，当抛出异常的时候，就会捕获异常，并跳转到 `except` 子句，这里的写法会捕获任意异常，这是因为这里会不只出现一种异常。不过在生产环境中不建议这样做

至此，整个项目就完成啦。

四、整个项目源码

项目使用的完整代码供参考：

```
#!/usr/bin/env python3

import os
import threading
import re
from bs4 import BeautifulSoup
from urllib.request import urlopen
from selenium import webdriver

browserPath = '/opt/phantomjs-2.1.1-linux-x86_64/bin/phantomjs'
homePage = 'https://mm.taobao.com/search_tstar_model.htm?'
outputDir = 'photo/'
parser = 'html5lib'


def main():
    driver = webdriver.PhantomJS(executable_path=browserPath)  #浏览器的地址
    driver.get(homePage)  #访问目标网页地址
    bsObj = BeautifulSoup(driver.page_source, parser)  #解析目标网页的 Html 源码
    print("[*]OK GET Page")
    girlsList = driver.find_element_by_id('J_GirlsList').text.split(
        '\n')  #获得主页上所有妹子的姓名、所在城市、身高、体重等信息
    imagesUrl = re.findall('\/\/gtd\.alicdn\.com\/sns_logo.*\.jpg',
                           driver.page_source)  #获取所有妹子的封面图片
    girlsUrl = bsObj.find_all(
        "a",
        {"href": re.compile("\/\/.*\.htm\?(userId=)\d*")})  #解析出妹子的个人主页地址等信息
    # 所有妹子的名字地点
    girlsNL = girlsList[::3]
    # 所有妹子的身高体重
    girlsHW = girlsList[1::3]
    # 所有妹子的个人主页地址
    girlsHURL = [('http:' + i['href']) for i in girlsUrl]
    # 所有妹子的封面图片地址
    girlsPhotoURL = [('https:' + i) for i in imagesUrl]

    girlsInfo = zip(girlsNL, girlsHW, girlsHURL, girlsPhotoURL)

    # 姓名地址       girlNL，   身高体重 girlHW
    # 个人主页地址  girlHRUL, 封面图片 URL
    for girlNL, girlHW, girlHURL, girlCover in girlsInfo:
        print("[*]Girl :", girlNL, girlHW)
        # 为妹子建立文件夹
        mkdir(outputDir + girlNL)
        print("    [*]saving...")
        # 获取妹子封面图片
        data = urlopen(girlCover).read()
        with open(outputDir + girlNL + '/cover.jpg', 'wb') as f:
            f.write(data)
        print("    [+]Loading Cover... ")
        # 获取妹子个人主页中的图片
        getImgs(girlHURL, outputDir + girlNL)
    driver.close()


def mkdir(path):
    # 判断路径是否存在
    isExists = os.path.exists(path)
    # 判断结果
    if not isExists:
        # 如果不存在则创建目录
        print("    [*]新建了文件夹", path)
        # 创建目录操作函数
        os.makedirs(path)
    else:
        # 如果目录存在则不创建，并提示目录已存在
        print('    [+]文件夹', path, '已创建')


def getImgs(url, path):
    driver = webdriver.PhantomJS(executable_path=browserPath)
    driver.get(url)
    print("    [*]Opening...")
    bsObj = BeautifulSoup(driver.page_source, parser)
    #获得模特个人页面上的艺术照地址
    imgs = bsObj.find_all("img", {"src": re.compile('.jpg')})
    for i, img in enumerate(imgs[1:]):  #不包含与封面图片一样的头像
```

```
        try:
            html = urlopen('https:' + img['src'])
            data = html.read()
            fileName = "{}/{}.jpg".format(path, i + 1)
            print("    [+]Loading...", fileName)
            with open(fileName, 'wb') as f:
                f.write(data)
        except Exception:
            print("    [!]Address Error!")
    driver.close()


if __name__ == '__main__':
    if not os.path.exists(outputDir):
        os.makedirs(outputDir)
    main()
```

最后在调试的时候，可能会获取不到淘宝页面的 HTML 源码。很大原因是实验楼的用户人数过多，出口带宽不足，所以多试几次，或者拿到自己电脑本地运行一下。

五、实验总结

这个小项目通过爬取淘女郎的照片来熟悉 BeautifulSoup 、正则表达式、 Selenium Webdriver 、 Phantomjs 、文件流操作的基础知识。

六、课后习题

如果有兴趣可以对该程序进行扩展，一些扩展思路供参考：

1. 增强异常处理，使程序爬取的成功率更高，程序更加稳健。
2. 通过机器学习挑选长得好看 MM 照片
3. 增加多线程操作，以增加图片收集效率，但是从应用角度讲，这样会过度消耗服务器资源，这又是一种DDOS攻击
4. 继续衍生下去，爬取主页中详细的个人简历。

课程教师

阿treee
共发布过1门课程

查看老师的所有课程 > (/teacher/30174)

前置课程

Linux 基础入门（新版） (/courses/1)

Python3 简明教程 (/courses/596)

进阶课程

Python网络爬虫实战--Scrapy框架学习 (/courses/763)