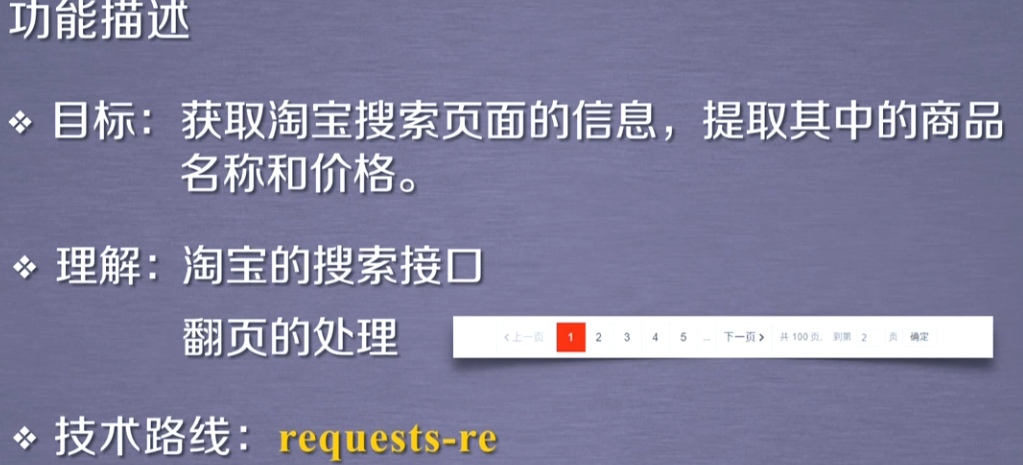
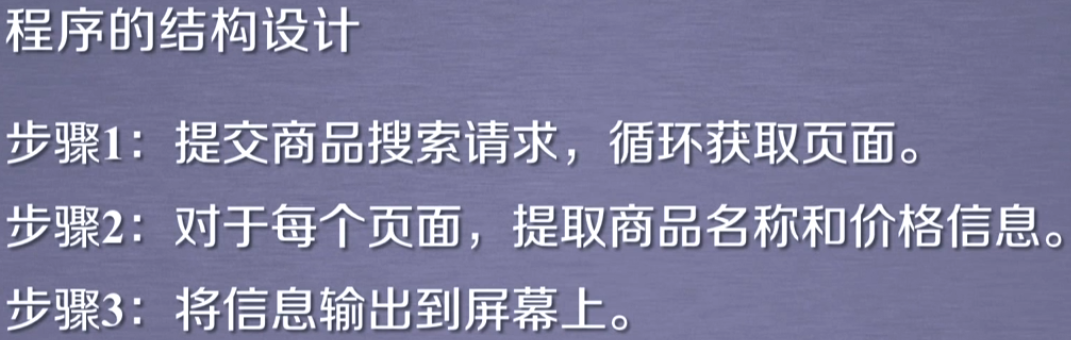
实战Projects

# 淘宝商品比价定向爬虫





查看robots协议：（robots.txt一般都存放在网站根目录下）



代码：

import requests

import re

**def getHTMLText(url):**

try:

html = requests.get(url,'html.parser');

html.raise\_for\_status();

html.encoding = html.apparent\_encoding;

return html.text;

except:

return "";

**def fillGoodsList(goodsList,html):**

**view\_price = r'\"view\_price\"\:\"[\d\.]\*\"';**

**view\_name = r'\"raw\_title\"\:\".\*?\"';**

pricesList = re.findall(view\_price,html);

namesList = re.findall(view\_name,html);

for i in range(len(pricesList)):

price = eval(pricesList[i].split(':')[1]);

name = eval(namesList[i].split(':')[1]);

goodsList.append([price,name]);

**def printGoodsList(goodsList):**

geshi = '{0:^15}\t{1:^10}\t{2:{3}^30}';

print(geshi.format('序号','价格','书包名称',chr(12288)))

count = 0;

for good in goodsList:

count = count+1;

print(geshi.format(count,good[0],good[1],chr(12288)))

**def main():**

goods = "书包";

start\_url = "https://s.taobao.com/search?q="+goods;

goodsList = [];

depth = 2;

for i in range(depth):

try:

url = start\_url+"&s="+str(44\*i);

html = getHTMLText(url);

fillGoodsList(goodsList,html);

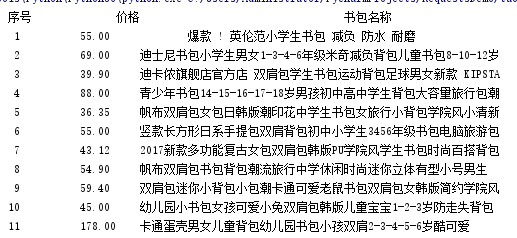
except:

continue;

printGoodsList(goodsList);

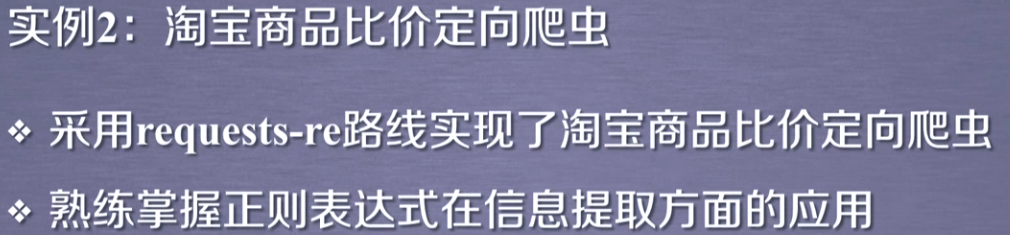
main();

结果：



**书写代码习惯：**

## 先书写大体流程，将功能需求进行模块化，定义成函数，先声明成函数，需要定义该函数的输入和输出参数。总体流程写好之后，再把具体的函数体所完成的功能补充完整即可。

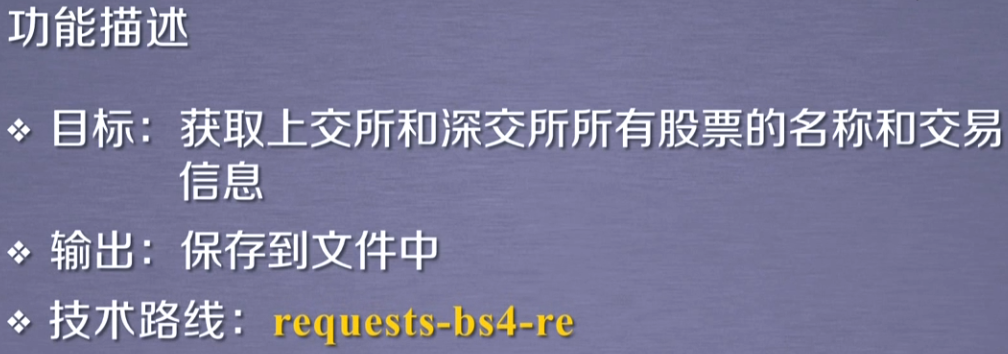


# 股票数据定向爬虫

Python那些事 公众号详细介绍：

<http://mp.weixin.qq.com/s?__biz=MzAwOTQ4MzY1Nw==&mid=2247484247&idx=1&sn=1742c76296f8db2d37f698db6b2d687e&chksm=9b5fa7b1ac282ea7f50936b60abe92b01ab58639f6d08b40a1f5fe3590f6267b24594bf5af17&mpshare=1&scene=1&srcid=0827Vw2k9EHM9sDvdWMH2LpJ#rd>

## 功能描述



## 候选股票信息网站的选择

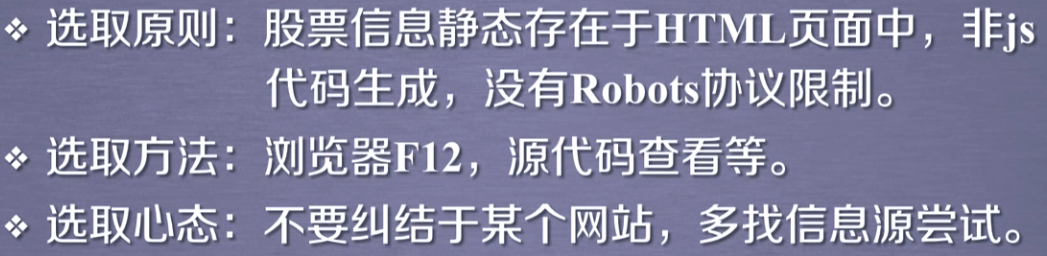
### 备选网站

百度股票网站 ： <https://gupiao.baidu.com/>

新浪股票网站： <http://finance.sina.com.cn/stock/>

<http://quote.eastmoney.com/stocklist.htm>

### 选取原则及方法



具体分析：

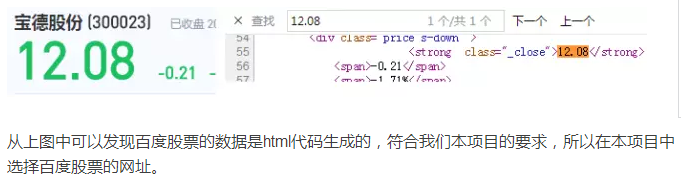
 网站选择原则： 股票信息静态存在于html页面中，非js代码生成，没有Robbts协议限制。

 选取方法： **打开网页，查看源代码，搜索网页的股票价格数据是否存在于源代码中。 发现新浪股票网址的网页上的数据信息是js提供，所以不采用新浪股票网址。**

<http://finance.sina.com.cn/realstock/company/sz000877/nc.shtml>

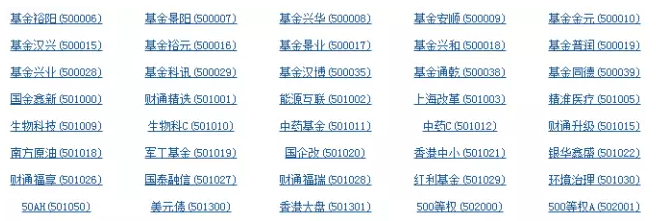


再打开百度股票的网址：链接描述(<https://gupiao.baidu.com/stock/sz300023.html> )，如下图所示：



### 东方财富网获取股票代码

**由于百度股票只有单个股票的信息，所以还需要当前股票市场中所有股票的列表，在这里我们选择东方财富网**，网址为：链接描述(http://quote.eastmoney.com/stocklist.html)，界面如下图所示：



## 原理分析

查看百度股票每只股票的网址：<https://gupiao.baidu.com/stock/sz300023.html> ，可以发现网址中有一个编号300023正好是这只股票的编号，**sz表示的深圳交易所**。因此我们构造的程序结构如下：

* 步骤1： 从东方财富网获取股票列表；
* 步骤2： 逐一获取股票代码，并增加到百度股票的链接中，最后对这些链接进行逐个的访问获得股票的信息；
* 步骤3： 将结果存储到文件。

程序设计思路：

### 从东方财富网上爬取各个公司的股票代码到一个stockList中，页面中有4000多个，只截取一部分。<http://quote.eastmoney.com/stocklist.html>利用函数getStockList(url,stockList)来实现。

### 对于每一个公司的股票在百度上都有一个html页面，即

<https://gupiao.baidu.com/stock/stockcode.html> **，这样循环迭代这些页面即可。**

### 从各自页面中寻找特点，找出各个股票的一些信息，并抓取下来，存放到一个字典中，然后写到文件中。

### 由于过程比较慢，可以写一个类似进度条的东西给用户看，用户体验会更好。

代码：

**import requests**

**import re**

**from bs4 import BeautifulSoup**

**import traceback**

**import bs4**

def getHTMLText(url):

try:

**html = requests.get(url);**

**html.raise\_for\_status();**

**html.encoding = html.apparent\_encoding;**

**return html.text;**

except:

return "";

**def getStockList(url,stockList):**

try:

html = getHTMLText(url);

**soup = BeautifulSoup(html,'html.parser');**

**lst = soup.select('div .quotebody')[0].select('li');**

**pattern = re.compile('s[hz][\d]{6}');**

num = 0;

for li in lst:

num = num +1;

if (num < 501) &(num>300):

str = pattern.findall(li.a['href'])[0];

stockList.append(str);

elif num > 502:

return "";

else:

continue;

except:

print('Fail')

return ""

**def getStockToTxt(stockList,urlPre,path):**

proceed = 0;

for code in stockList:

urlCode = urlPre + code + '.html';

try:

html = getHTMLText(urlCode);

if html == "":

continue;

infoDict = {};

soup = BeautifulSoup(html,'html.parser');

**stockInfo = soup.find('div',attrs = {'class':'stock-bets'});**

**if isinstance(stockInfo,bs4.element.Tag) :**

**name = stockInfo.find\_all(attrs={'class':'bets-name'})[0];**

**infoDict.update({'股票名称':name.text.split()[0]});**

**keyList = stockInfo.find\_all('dt');**

**valueList = stockInfo.find\_all('dd');**

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

**key = keyList[i].text;**

**value = valueList[i].text;**

**infoDict[key] = value;**

with open(path ,'a',encoding='utf-8') as f:

f.write(str(infoDict)+'\n');

**proceed += 1;**

**print('\r当前速度：{:.2f}%'.format(proceed\*100/len(stockList),end=''));**

**except:**

print("Fail")

traceback.print\_exc();

**proceed += 1;**

**print('\r当前速度：{:.2f}%'.format(proceed \* 100 / len(stockList), end=''));**

continue;

**主函数：**

**def main():**

**url = 'http://quote.eastmoney.com/stocklist.html';**

**urlPre = 'https://gupiao.baidu.com/stock/';**

**stockList = [];**

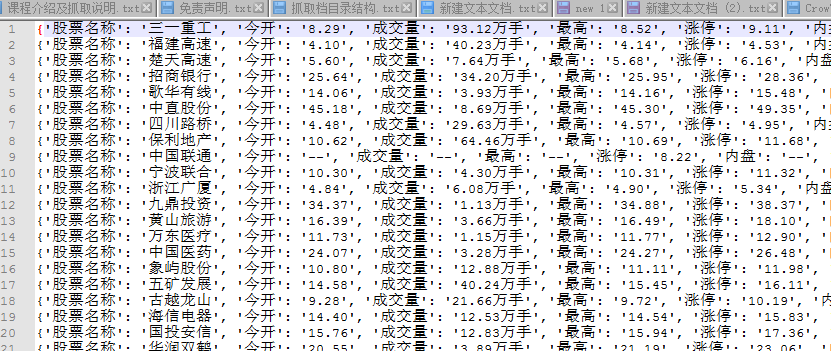
**path = 'E://stock3.txt'**

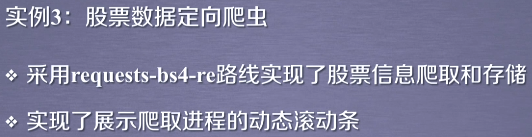
**getStockList(url,stockList);**

**getStockToTxt(stockList,urlPre,path);**

main()

爬取结果：





可以实现简单的类似进度条，展示进度：

**proceed = 0；**

**proceed += 1;**

**print('\r当前速度：{:.2f}%'.format(proceed\*100/len(stockList),end=''));**

由于特定网站的编码都是固定的，所以为了提高爬虫效率，可以人为获知编码方式，直接传给html.coding即可。

利用**html.apparent\_encoding**会比较耗费时间。

# 中国大学排名爬取

首先，先查看html页面的代码，分析标签，查找规律，获取需要的信息。

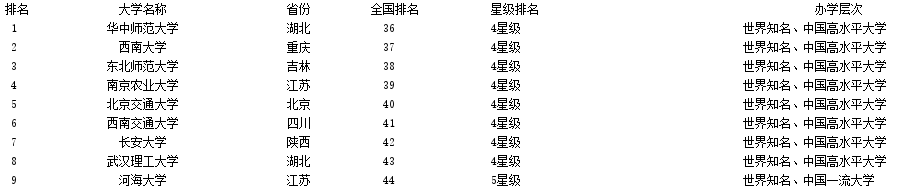
其次，定义方法和接口，实现整体框架，主要是完成main函数和功能函数定义声明。

最后，完成功能函数的内容。

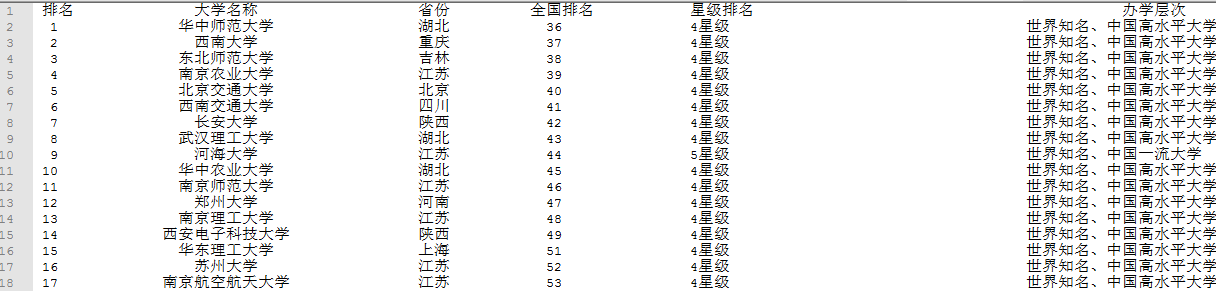
代码：

**import** requests;  
**import** re;  
**from** bs4 **import** BeautifulSoup;  
  
**def** getHTMLText(url):  
 **try**:  
 html = requests.get(url);  
 html.raise\_for\_status();  
 html.encoding = html.apparent\_encoding;  
 **return**(html.text);  
 **except**:  
 print(**"获取HTML失败"**);  
 **return ""**;  
  
**def** getUnivsList(univsList,htmlText):  
 soup = BeautifulSoup(htmlText,**'html.parser'**);  
 filePath = **'E://table.txt'**;  
 *# saveHtmlToFile(filePath,str(soup.table))* table = soup.table;  
 trList = table.find\_all(**'tr'**);  
 *# print(trList);* print(len(trList));  
  
 **for** tr **in** trList:  
 tdList = tr.find\_all(**'td'**);  
 univList = [];  
 *# univList.append("");* **if** len(tdList)==6:  
 **for** td **in** tdList:  
 univList.append(td.text.strip());  
 univsList.append(univList);  
 **else**:  
 **continue**;  
  
  
**def** printUnivs(univsList):  
 geshi = **'{0:^5}\t{1:{6}^16}\t{2:{6}^6}\t{3:{6}^7}\t{4:{6}^10}\t{5:{6}^40}\t'** str = geshi.format(**'排名'**,**'大学名称'**,**'省份'**,**'全国排名'**,**'星级排名'**,**'办学层次'**,chr(12288));  
 print(str);  
 saveHtmlToFile(**'E://univList.txt'**, str + **"\n"**);  
 **for** univLst **in** univsList:  
 str = geshi.format(univLst[0],univLst[1],univLst[2],univLst[3],univLst[4],univLst[5],chr(12288))  
 saveHtmlToFile(**'E://univList.txt'**,str+**"\n"**);  
 print(str);  
  
**def** saveHtmlToFile(filePath,htmlText):  
 **with** open(filePath,**'a'**,encoding=**'utf-8'**) **as** f:  
 f.write(htmlText);  
  
  
**def** main():  
 url = **"http://gaokao.xdf.cn/201702/10611319.html"**;  
 htmlText = getHTMLText(url);  
 **if** htmlText == **""**:  
 print(**"获取HTML文本失败！！！"**)  
 **return ""**;  
 univsList = [];  
 getUnivsList(univsList,htmlText);  
 printUnivs(univsList);  
  
main();

结果：



文件中：



# 体会

## 当html页面中，如果寻找的内容有规律，可以通过正则表达式获取的，可以利用正则表达式直接获取，比较简单。当html页面比较复杂时，利用BeautifulSoup定位到具体某一部分，然后再利用正则表达式也是比较方便的。

## BeautifulSoup可以精确对整个Html文档进行定位，且可以通过标签、属性、文本等进行查找标签。利用好BeautifulSoup是这部分的重点和难点。可以利用find或find\_all函数查找、（注意简化形式如soup.tag）、也可以利用CSS选择器即select函数查找。