### 股票数据定向爬虫

股票数据是量化分析的基础数据，但是海量的股票数据，“用手获取”数据未免太过繁琐。于是，利用网络爬虫获取数据成为一种更好的选择。但是，网络爬虫怎么使用？网络爬虫这么复杂我是否能编写？

本文将介绍一种利用python脚本编写的定向爬虫，可以对每支股票的相关数据进行爬取，并利用程序自动整理并输入到文件或数据库中。

（本文适用对象：对python代码有初步了解，有相关爬取需求的人）

本文采取requests-bs4的技术路线来实现这个简易爬虫，选用python 3.6，以下代码在IDLE环境下运行即可。

在编写一个爬虫之前，我们得先了解一下这个爬虫的运行流程。

这个定向爬虫爬取的流程是，先获取网页数据，再对网页内数据进行分析、筛选和匹配，最后将数据打包放入文件中。

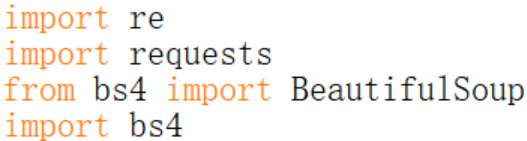
由于我们编写的是一个定向爬虫，功能上限制了我们只能爬取那种写在网页源代码中的数据，对于由JavaScript生成的数据，我们暂时还无法获取。经过比较和筛选，本文采用百度股票作为数据源进行爬取。

我们观察一下百度股票的网页链接可以发现，每一支股票都有一个专门的链接。所有股票的链接前面的部分都是相同的，除了最后一个斜杠处。观察可以发现，这个斜杠后面跟随的是这支股票的代码。

学习是一个渐进的过程，我们不妨先学习如何爬取一支股票的信息并将其输入到IDLE中，再通过对脚本的改进，增添批量爬取和将数据写入文件的功能。

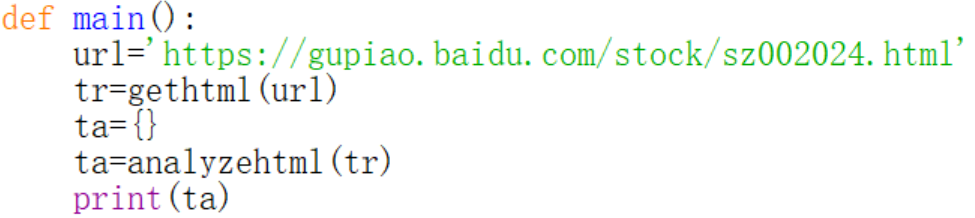
下面让我们一起一边分析，一边编写这个爬虫。

我们首先，引入我们最基本所需要的库：



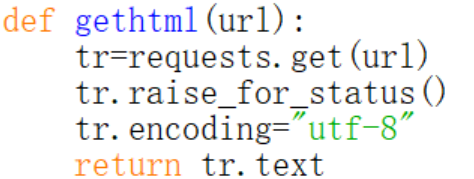
根据之前确定的爬取流程，我们需要两个函数，分别来实现获取网页信息（gethtml()）和分析网页信息（analyzehtml()），并且需要一个主函数来调用这些函数

我们首先编写主函数。主函数代码如下：



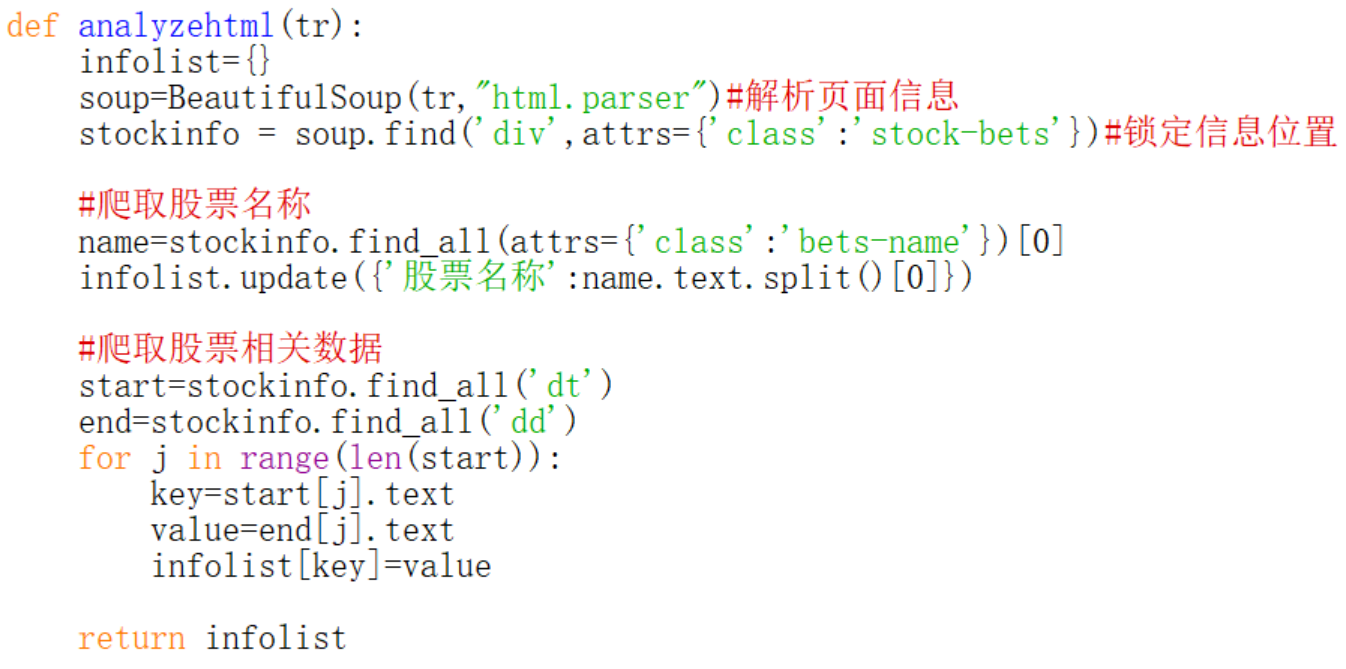
由于我们暂时只考虑爬取一支股票的信息，那我们就先以苏宁易购（股票代码sz002024）为例，主函数第一行给出的就是苏宁易购这只股票信息对应的链接地址。然后我们依次调用两个函数。

首先是gethtml()这个函数，其功能设定是获取指定HTML界面的信息，并将全部页面信息以文本形式返回给主函数，我们首先给出代码：



这个函数我们只设置了一个参数，即主函数传过来的url链接。这个函数使用到了BeautifulSoup库，第一步我们调用requests库的 get()方法，获取了整个网页的信息，并赋给了tr变量。然后我们调用tr.raise\_for\_status()方法，检测获取的状态；再调用r.encoding=utf-8，将网页编码修改为常用的utf-8格式。最后以文本形式(.text)将爬取的内容返回主函数

实际上，整个爬虫需联网“爬”的部分我们已经完成了，整个页面信息都包含在了我们返回的文本中。接下来我们的任务就是，从文本中筛选我们需要的信息，并形成一个列表返回给主函数变量tr。让我们一起分析一下analyzehtml()这个函数。analyzehtml()函数的代码如下：



由于接下来的爬取内容整理的需要，我们首先定义一个空字典infolist{}。然后我们开始使用BeautifulSoup库来解析这个网页。

解析这个网页之前，我们先打开浏览器，在网页源代码中找到我们要爬取的信息的位置。通过观察我们可以发现，我们需要找的所有信息，都包含在了一个大标签<div class="stock-info" data-spm="2">之下，并且除了股票名称外，我们要的信息都包含在了<dl></dl>这一对标签之中，并且每一支股票相关信息的名称和内容，都分别使用<dd></dd>和<dt></dt>这两对标签封装。因此，我们可以使用find\_all()方法很方便地将它们提取出来。

我们来分析一下这个函数。

我们首先通过BeautifulSoup()方法解析整个html界面并将结果赋值给soup变量，再使用find方法锁定这个<div class="stock-info" data-spm="2">大标签的位置，并赋值给stockinfo变量。

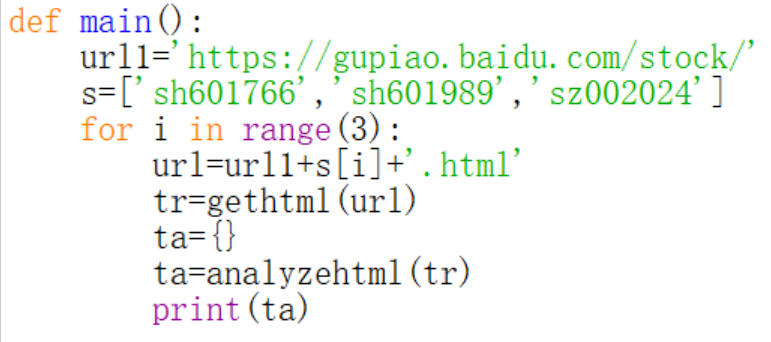
接下来我们先解析出股票名称。通过网页源代码我们可以看到，股票名称封装在<a class="bets-name" href="[/stock/sz002024.html](https://gupiao.baidu.com/stock/sz002024.html" \t "https://gupiao.baidu.com/stock/_blank)">这个标签之下，于是我们使用find\_all方法将名称提取出来，并赋值给name变量，再将其转化为键值对的形式存入字典infolist{}中

再往下我们解析股票信息。由于我们知道信息名称和信息内容分别封装在<dd></dd>和<dt></dt>两对标签中，我们便可以两次调用find\_all()方法，寻找这两个标签下的内容，这个方法会以列表形式将结果分别返回给指定变量start和end。接下来，我们需要整理刚解析出来的数据，我们知道start和end两个列表中的数据是正好一一对应的，所以我们可以使用一个循环，将两个列表对应的数据依次取出，作为一个键值对存放入字典infolist{}。最后用return将字典返回主函数，并输出。

至此，我们股票爬虫的基本框架就完成了，我们可以根据这个基本模式，对其进行优化。

优化1：批量爬取的实现

要实现批量爬取，我们只需要修改主函数即可。以下给出一种简单的修改方式，例如，我们要爬取中国重工（sh601989），苏宁易购（sz002024）和中国中车（sh601766）这三支股票，我们需要进行如下改动：

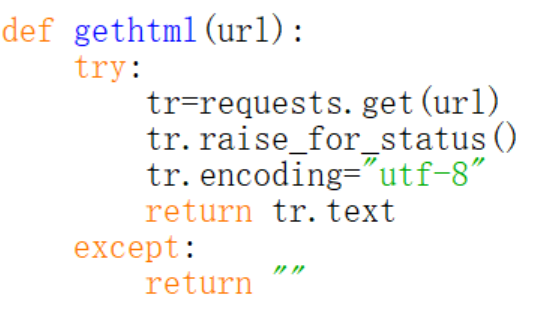


我们也可以再编写其他函数，对列表s进行自动扩充，以爬取更多的数据，读者可以另外去了解，此处不再赘述。

优化2：容错性优化

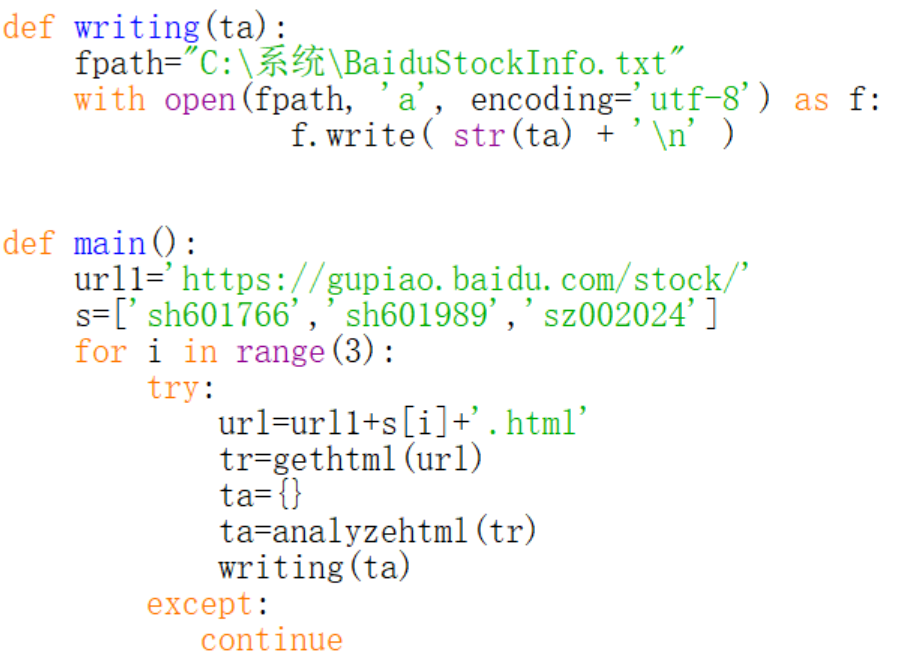
在脚本执行过程中，可能发生各种意外的错误，例如股票代码错误、网络异常等。因此我们需要增强代码的容错性。我们可以通过添加try...except....结构来优化。这样即使运行中发生异常，程序也不会意外中止。

eg：



优化3：将数据写入文件

我们以写入文本文件为例。我们只需添加一个函数writing(),并在主函数中调用它



除此之外，我们还可以将数据打包存入Excel或者数据库中。在此推荐使用pandas库来完成，本文也不再赘述。