1. 实验环境

操作系统：Windows8.1专业版

程序设计语言：Python3.4.3(32位)

浏览器测试：Chrome42

所用的第三方包：openpyxl(写excel),urllib(读取网站数据),time(计时功能),bs4(html被拆解为对象处理，提取标签中的内容),simplejson(解析json数据)

1. 预习与准备
2. 爬虫整体思路：对于给定的动态网址，首先要按F12键分析网页真实的响应网址，获得数据网页地址后，根据网页内容，如静态网页，json网页等按需爬取。
3. 语言选择：Python，拥有丰富的第三方库，并且程序小巧，网页模块比较完善，代码形式接近伪代码，实现起来比较容易。
4. Python相关库：
5. urllib:获取网页的源代码，使用urllib.request.urlopen获得网页，对于有中文的网页，一般还需要decode
6. beautifulsoup：获取标签(table,tr,td,th)中的内容，方便提取。
7. openpyxl:使用WorkBook模块存储爬取到的表格内容。
8. time：计时模块，将网页中的标准时间转换为本地时间，并且加上抓取日期，以及定时抓取。
9. simplejson：获取jsondata：simplejson.loads(list)，将jsondata转为python能够直接处理的字典数据类型。
10. 实验内容

给定网址(6): <http://app.finance.ifeng.com/list/gold.php>,将该网址中的表格提取到excel中

1. 实验过程
2. 获取网站响应地址：通过查询网站(6)的源代码可知，该网页属于动态网址，即网页数据通过Javascript代码产生的，因此需要分析出网页的真实地址。而一般的浏览器都有开发者工具选项，可以查看网站的依次响应情况。在谷歌浏览器中，按F12键选择Network选项，可以分析出。经过分析，我发现该网页的数据存储在两个网页里面，一个是接近json数据格式的网址(A)：<http://hq.finance.ifeng.com/q.php?l=g_Ag_T_D_,g_Au_T_D_,g_Au100g,g_Au99_95,g_Au99_99,g_Pt99_95,g_Au_T_N1_,g_Au_T_N2_,g_kingXianhuo,g_gangXianhuo,g_baiyingxianhuo,g_boXianhuo,g_baXianhuo,g_mbj,g_mj,g_djy,g_mhjl,g_mbyl,g_djb,g_djj,g_dj&f=json&e=show_hq();&random=0.40822120290249586>, 不过这里面的数据是无序的，因此必须手动指定对应项；另一个是一个静态网址(B)：<http://www.fyme.com.cn/pub/index_hq_ifeng.php> 并没有完全显示所有数据。
3. BeautifulSoup4获取动态网页表格(对应表格有：上海金交所，国际现货，国际近月)：先查找所有<table>标签，在每个标签中，先寻找<tr>标签，在每个<tr>标签中查找所有<th><td>，cols = tr.find\_all(["td", "th"]),如果cols长度为1，则为表的开始，使用标记Action=[True,False,False]，第一个参数为表头开始，第二个参数为写时间表头，第三个参数为写抓取的时间。然后对于cols中的每个td，获得标签中的数据：data\_set = str(td.string).strip()，然后对照原始网页中的数据和json数据项，结果发现data\_set中的标号一般为g\_Ag\_T\_D\_\_last等，而json中的为：g\_Ag\_T\_D\_，并且不是按顺序的，因此需要把data\_set中的标记截取字符串开头到最后一个下划线之间的子串作为字典中的key,按随机访问的方式将字典中的值对应到表格中的数据，在本网页中，经过比较得出表格中的数据分别对应到json数据中的 0,2,5,6,1,3,4,7项(下标从零开始)。
4. BeautifulSoup4获取静态网页(对应表：泛亚有色金属，以及原始网页，从原始网页中获得表头数据)：首先注意到泛亚有色金属静态网页写的并不规范，表头<tr><td>嵌入出现多次，导致BeautifulSoup4不能正确解析，直接流出一个空行，因此，对于这一个部分采用了手动写入表头数据，因为是静态数据， 直接使用 str(td.string).strip()获取数据就可以了。
5. 写数据到Excel中：首先创建一个工作簿Workbook(),因为要每十分钟运行一次程序，故每个sheet用当前运行的次数来命名。没写一行数据，只需使用sheet.append(colsArr)功能即可添加，最后使用 wb.save(Filename)可以保存到Excel中。在本程序中，设置了将Excel保存到真实的可执行程序的父目录中，因为使用了WinRAR制作了自解压压缩包文件，运行后会在本目录下产生一个解压后的文件夹，这样Excel结果就在当前目录了。
6. 每十分钟抓取一次：主程序是一个循环，每抓取一次，程序休眠十分钟，之后再运行。
7. 打包：使用python中的cx\_freeze模块将python代码打包成EXE文件，主要是将依赖库，python环境等以动态链接库的形式组合到一起。由于打包后文件分散，故我又采用了winrar自解压缩技术将其打包成一个EXE文件，方便使用。
8. 实验结果

经过比较，发现完全爬取了指定网站上的表格数据。

1. 实验总结与体会

通过此次网络爬虫实验，我对网络中Http的响应有了一定的了解。一开始以为数据就是直接存储在网页里面的，结果看网页源码却找不到，但通过审查元素却找到了数据，于是上网查了下，问了一些同学，发现这是一种动态网页，通过Javascript产生的，必须通过分析HttpResponse才能获得真实的网址。获得地址后，发现这个网页中的数据存储在两个地方，一个是类似于json格式的数据，另一个则是静态网页，之后通过比较对号入座就可以了。