《用 Python 玩转数据》爬虫小项目（3 项）

Dazhuang@NJU

1. “迷你爬虫编程小练习”进阶：抽取某本书的前 50 条短评内容并计算评分(star)的平均值。提示：有的评论中并不包含评分。

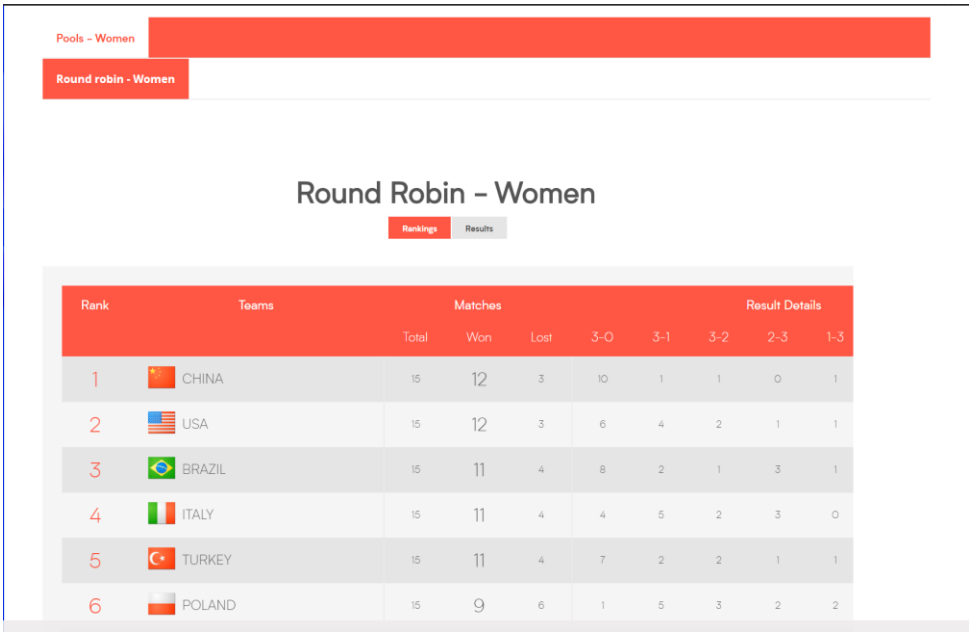
2. 在“https://money.cnn.com/data/markets/nasdaq/”上抓取纳斯达克成分股数据并将以下数据表抓取到一个列表中输出（你需要分析如下的列表）。

Companies in the NASDAQ NMS COMPOSITE INDEX

Company	Price	Change	% Change	P/E	Volume	YTD change
ZNGA	9.15	-0.03	-0.33%	NM	500.3K	+49.51%
ZYXI	14.07	-0.53	-3.63%	44.0	22.0K	+78.78%
ZYNE	3.84	+0.04	+1.05%	NM	33.7K	-36.42%
CNET	1.50	+0.03	+2.04%	NM	5.2K	+28.76%
ZUMZ	32.38	-0.29	-0.89%	13.5	9.0K	-6.25%
ZS	154.98	+0.01	+0.01%	NM	95.2K	+233.29%
ZVO	4.64	-0.13	-2.73%	NM	4.3K	+125.24%
ZSAN	0.66	+0.0068	+1.05%	NM	62.3K	-56.79%
ZI	42.63	-0.14	-0.33%	NM	21.2K	--
ZM	559.41	+23.01	+4.29%	708.1	651.2K	+722.18%
ZGNX	20.43	+0.81	+4.13%	NM	47.2K	-60.81%
ZKIN	1.40	-0.01	-0.71%	5.2	200.00	+8.53%
ZIXI	6.81	+0.09	+1.34%	NM	118.4K	+0.44%
ZIOP	2.72	+0.075	+2.84%	NM	36.0K	-42.48%

Data as of 7:40pm ET, 10/16/2020

3. 请爬取网页（https://www.volleyball.world/en/vnl/2019/women/resultsandranking/round1）上的数据（包括 TEAMS and TOTAL, WON, LOST of MATCHES）



提示：在处理时可以用已学的方法将每一项需要的内容（如 USA 和 15）单独解析出来，但这种做法将有联系的数据打散了，较好的做法是将每个 TEAM 的相关数据按组解析出来。但是由于包含这 4 项信息的源代码（请自行观察）分在多行并且行首有多个空格，因此在处理时在构造正则表达式时要把换行时的空白字符表示出来（用\s+可表示多个空白字符，包括换行符和空格）。

【参考程序见下一页】

【参考代码：将 url 中的 **bookid** 换成自己想查看的书的 id，例如 *1084336*】

```
# -*- coding: utf-8 -*-

"""

Comments    parsing

@author: Dazhuang

"""

import requests, re, time

from bs4 import BeautifulSoup



count = 0

i = 0

s, count_s, count_del = 0, 0, 0

lst_stars = []

headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.108 Safari/537.36'}

while count < 50:

    try:

        r = requests.get('https://book.douban.com/subject/bookid/comments/hot?p=' +
          str(i+1), headers = headers)

    except Exception as err:

        print(err)

        break

    soup = BeautifulSoup(r.text, 'lxml')

    comments = soup.find_all('span', 'short')

    pattern = re.compile('<span class="user-stars allstar(.*?) rating"')
```

```
        # Other way: we can use a whole regular expression to pattern comments and rangking stars
        p = re.findall(pattern, r.text)
        for item in comments:
            count += 1
            if count > 50:
                # count the number of comments more than 50 of the page
                count_del += 1
            else:
                print(count, item.string)
        for star in p:
            lst_stars.append(int(star))
        time.sleep(5)           # delay request from douban's robots.txt
        i += 1
        for star in lst_stars[:-count_del]:       # calculate the rating star of 50 comments
            s += int(star)
    if count >= 50:
        print(s // (len(lst_stars)-count_del))
```

【参考代码】

```
# -*- coding: utf-8 -*-
"""
Get dji stock data
@author: Dazhuang
"""

import requests
import re


def retrieve_dji_list():
    r = requests.get('https://money.cnn.com/data/markets/nasdaq/')
    # 先获取表头信息
    head = re.findall('<thead>(.*?)</thead>', r.text)
    assert len(head)==1
    table_head = ['Company'] + re.findall('<th>(.+?)<', head[0])
    # 先获取表的总体
    tbody_pat = re.compile('tbody>(.*?)</tbody')
    tbody = re.findall(tbody_pat, r.text)
    assert len(tbody) == 1
    # 再获取总体表中每一条记录
    tr_pat = re.compile('<tr>(.*?)</tr>')
    tr_list = re.findall(tr_pat, tbody[0])
    # 最后获取每一条记录中的各个字段
    table_pat = re.compile('>([^<^&]+?)<')
    stock_list = [table_head]
    for i in tr_list:
        s = re.findall(table_pat, i)
        stock_list.append(s)
    return stock_list

dji_list = retrieve_dji_list()
print(dji_list)
```

【参考代码】

```python
# -*- coding: utf-8 -*-
"""
Crawler
@author: Dazhuang
"""
import re
import requests

year = 2019
def crawler(url):
    try:
        r = requests.get(url)
    except requests.exceptions.RequestException as err:
        return err
    r.encoding = r.apparent_encoding
    # 一定要把下面这 3 行写在同一行上
    pattern = re.compile('href="/en/vnl/%s/women/teams/.*?">(.*?)</a></figcaption>\s+</figure>\s+</td>\s+<td></td>\s+<td class=".*?">(.*?)</td>\s+<td class=".*?">(.*?)</td>\s+<td class=".*?">(.*?)</td>' % year)
    p = re.findall(pattern, r.text)
    return p

# if __name__ == "__main__":
url = 'http://www.volleyball.world/en/vnl/%s/women/resultsandranking/round1' % year
result = crawler(url)
print(result)
```