

네이버 주식 정보 가져와 보기

정보 가져오기에 앞서 페이지 확인해 보기

`https://finance.naver.com/robots.txt`

```
User-agent: *
Disallow: /
User-agent: yeti
Disallow: /
Allow: /sise/
Allow: /research/
Allow: /marketindex/
Allow: /fund/
Disallow: /fund/news/
Disallow: /marketindex/news/
```

- url : https://finance.naver.com/sise/sise_index.nhn?code=KOSPI
(https://finance.naver.com/sise/sise_index.nhn?code=KOSPI)
- 코스피의 지수 정보를 가지고 오기.
- 거래량(천주)은 정보를 가지고 와 보자.
- 장중 최고는 얼마인지 가지고 오자.
- 거래대금(백만)의 정보를 가지고 오자.
- 장중최저, 52주 최저의 정보를 가지고 오자.

In [1]:

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
```

In [2]:

```
url = "https://finance.naver.com/sise/sise_index.nhn?code=KOSPI"
page = urlopen(url)
page
soup = BeautifulSoup(page, "html.parser")
```

id 정보를 활용하여 거래량(천주), 장중 최고, 장중 최저 가져오기

In [3]:



```
quant = soup.find("td", id = "quant").text      # 거래량(천주)
high_value = soup.find("td", id = "high_value").text # 장중 최고
low_value = soup.find("td", id = "low_value").text  # 장중 최저
deal_money = soup.find("td", id="amount").text     # 거래 대금
print("거래량 : ", quant)
print("거래대금 : ", deal_money)
print("장중 최고 : ", high_value)
print("장중 최저 : ", low_value)
```

```
거래량 : 961,726
거래대금 : 12,530,442
장중 최고 : 2,409.01
장중 최저 : 2,383.82
```

class 정보를 활용한 정보를 가져와 보기

In [4]:

```
info = soup.find_all('div', class_="subtop_sise_detail")
info
```

Out[4]:

```
[<div class="subtop_sise_detail">
<!-- [D] 상승장일때 up, 하락장일때 dn 클래스 추가 (보합시 추가 없음) -->
<div class="quotient" id="quotient">
<em id="now_value">2,391.96</em>
<span class="fluc" id="change_value_and_rate"><span>0.00</span> -0.00%<span class
="blind">상승</span></span>
</div>
<table cellpadding="0" cellspacing="0" class="table_kos_index" summary="코스피 주요
시세 정보에 관한 표입니다.">
<caption>코스피 주요시세</caption>
<tr>
<th class="th"><span class="blind">거래량 (천주)</span></th>
<td class="td" id="quant">961,726</td>
<th class="th2"><span class="blind">거래대금 (백만)</span></th>
<td class="td2" id="amount">12,530,442</td>
</tr>
<tr>
<th><span class="blind">장중최고</span></th>
<td class="td" id="high_value">2,409.01</td>
<th><span class="blind">장중최저</span></th>
<td class="td2" id="low_value">2,383.82</td>
</tr>
<tr>
<th><span class="blind">52주최고</span></th>
<td class="td">2,458.17</td>
<th><span class="blind">52주최저</span></th>
<td class="td2">1,439.43</td>
</tr>
<tr>
<th class="th"><span class="blind">등락/종목</span></th>
<td class="td3" colspan="3">
<ul>
<li class="lst"><span class="blind">상한종목수</span><a href="/sise/sise_upper.nh
n"><span>0</span></a></li>
<li class="lst2"><span class="blind">상승종목수</span><a href="/sise/sise_rise.nh
n"><span>0</span></a></li>
<li class="lst3"><span class="blind">보합종목수</span><a href="/sise/sise_steady.nh
n"><span>0</span></a></li>
<li class="lst4"><span class="blind">하락종목수</span><a href="/sise/sise_fall.nh
n"><span>0</span></a></li>
<li class="lst5"><span class="blind">하한종목수</span><a href="/sise/sise_lower.nh
n"><span>0</span></a></li>
</ul>
</td>
</tr>
</table>
</div>]
```

In [5]:

```
max_52week = info[0].find_all("td", class_='td')[2].text # 52주 최고
min_52week = info[0].find_all("td", class_='td2')[2].text # 52주 최저
```

In [6]:

```
print("52주 최고 %s" % max_52week)
print("52주 최저 %s" % min_52week)
```

52주 최고 2,458.17

52주 최저 1,439.43

관심 종목의 주식 정보를 for문을 이용하여 가져와 보기

In [7]:

```
## 삼성전자 : 005930
## 카카오 : 035720
## 유한양행 : 000100
```

In [8]:

```
my_stock = ['삼성전자', '카카오', '유한양행', 'NAVER']
my_stock_code = ['005930', '035720', '000100', '035420']
```

In [9]:

```
chk_url = 'https://finance.naver.com/item/main.nhn?code=000100'
page = urlopen(chk_url)
soup = BeautifulSoup(page, 'lxml')
```

In [10]:

```
soup.title
```

Out[10]:

```
<title>유한양행 : 네이버 금융</title>
```

In [11]:

```
soup.find('p', class_='no_today').find('span', class_='blind').text
```

Out[11]:

```
'65,700'
```

In [12]:

```
base_url = 'https://finance.naver.com/item/main.nhn?code='
```

In [13]:

```

com = []    # 회사명
price_c = [] # 가격

for code in my_stock_code:
    cnt = 0    # 회사명 선택을 위한 count
    all_url = base_url + code
    page = urlopen(all_url)
    soup = BeautifulSoup(page, 'lxml')

    ## 현재가
    tmp = soup.find('p', class_='no_today')
    price = tmp.find('span', class_='blind').text

    com.append(my_stock[cnt])
    price_c.append(price)

    cnt += 1

print(com)
print(price_c)

```

```

['삼성전자', '삼성전자', '삼성전자', '삼성전자']
['59,700', '379,000', '65,700', '301,000']

```

In [14]:

```
import pandas as pd
```

In [15]:

```

dic_dat = {'회사명':com, '주가':price_c}
dat = pd.DataFrame( dic_dat )
dat.to_csv("date_0118.xlsx", index=False)

```

In [17]:

```

# 파일 생성 확인
!dir *date*

```

C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: C822-B81C

C:\Users\User\Documents\GitHub\AI_enovation\part02_library 디렉터리

```

2020-10-12 오전 08:39          112 date_0118.xlsx
                1개 파일              112 바이트
                0개 디렉터리 412,907,900,928 바이트 남음

```

In []:

