In [1]:

```
import requests
import time
from bs4 import BeautifulSoup
import forecastio
```

Install

-$ pip3 install bs4

mac

-$ pip3 install python-forecastio

windows

-$ conda install -c auto python-forecastio

-$ pip install python-forecastio

Requests

- http://docs.python-requests.org/en/master/ (http://docs.python-requests.org/en/master/)

네이버 주식 데이터 가져오기

- api 사용 : json 파싱을 한다.
- 네이버 주식 페이지에서 주식 데이터를 가져와 데이터 프레임으로 만들기
- http://m.stock.naver.com (http://m.stock.naver.com)

In [2]:

```
def make_url(pageSize=10, page=1):
    return "http://m.stock.naver.com/api/json/sise/siseListJson.nhn?menu=market_sum&s

def get_data(url):
    response = requests.get(url)
    json_info = response.json()
    companys = json_info["result"]["itemList"]
    df = pd.DataFrame(columns=["종목", "시세", "전일비", "등락율", "시가총액", "거래량"])
    for company in companys:
        df.loc[len(df)] = {
            "종목":company["nm"],
            "시세":company["nv"],
            "전일비":company["cv"],
            "등락율":company["cr"],
            "시가총액":company["mks"],
            "거래량":company["aq"],
        }
    return df
```

In [3]:

```
url = make_url(1406,1)
df = get_data(url)
print(len(df))
df.tail()
```

```
1406
```

Out[3]:

	종목	시세	전일비	등락율	시가총액	거래량
1401	덕성우	4100	-325	-7.34	57	221828
1402	ARIRANG 국채선물10년	50945	120	0.24	57	1497
1403	KINDEX 필리핀MSCI(합성)	14175	-70	-0.49	57	296
1404	TIGER 삼성그룹펀더멘털	9440	-55	-0.58	57	26
1405	KODEX 미국러셀2000(H)	11230	-90	-0.80	56	8280

Dark Sky API

- 날씨 정보를 알려주는 api
- https://darksky.net/dev (https://darksky.net/dev)
- pip3 install python-forecastio

In [4]:

```
import forecastio
FORECAST_TOKEN = "c259d4aeb593bc5e83d0c3f7ed916f5d"
```

In [5]:

```
def forecast(lat, lng):
    url = "https://api.darksky.net/forecast/{}/{},{}".format(FORECAST_TOKEN, lat, ln
    response = requests.get(url)
    json_info = response.json()
    return json_info["hourly"]["summary"], json_info["timezone"]
```

In [6]:

```
lat = 37.5124413
lng = 126.9540519
forecast(lat,lng)
```

Out[6]:

```
('Overcast throughout the day.', 'Asia/Seoul')
```

In [7]:

```
def forecast(lat,lng):
    forecast = forecastio.load_forecast(FORECAST_TOKEN, lat, lng)
    byHour = forecast.hourly()
    return byHour.summary
```

In [8]:

```
lat = 37.5124413
lng = 126.9540519
forecast(lat,lng)
```

Out[8]:

```
'Overcast throughout the day.'
```

BeautifulSoup

- https://www.crummy.com/software/BeautifulSoup/bs4/doc/ (https://www.crummy.com/software/BeautifulSoup/bs4/doc/)
- pip3 install bs4

네이버 검색어 순위

- bs4 사용 : html element를 selector를 사용한다.
- 네이버 검색어 순위를 가져와 데이터 프레임으로 만들기
- http://naver.com (http://naver.com)

In [9]:

```
def naver_top20():
    df = pd.DataFrame(columns=["rank","keyword"])
    response = requests.get("http://naver.com")
    dom = BeautifulSoup(response.content, "html.parser")
    keywords = dom.select(".ah_roll .ah_l .ah_item")
    for keyword in keywords:
        df.loc[len(df)] = {
            "rank": keyword.select_one(".ah_r").text,
            "keyword": keyword.select_one(".ah_k").text,
        }
    return df
```

In [10]:

```
naver_df = naver_top20()
naver_df
```

Out[10]:

	rank	keyword
0	1	라이프 온 마스
1	2	로또810회당첨번호
2	3	그것이 알고싶다
3	4	시크릿 마더
4	5	김명수
5	6	정태옥
6	7	이별이 떠났다
7	8	이부망천
8	9	전지적 참견 시점
9	10	아는형님
10	11	sbs 온에어
11	12	임신중독증 유전
12	13	임신중독증
13	14	스케치
14	15	라이프 온 마스 뜻
15	16	로또
16	17	한혜진 그림
17	18	기안84
18	19	가을의전설
19	20	레반도프스키

다음 검색어 순위

In [11]:

```
def daum_top10():
    df = pd.DataFrame(columns=["rank","keyword"])
    response = requests.get("http://daum.net")
    dom = BeautifulSoup(response.content, "html.parser")
    keywords = dom.select("#mArticle ol.list_hotissue.issue_row.list_mini > li")
    for keyword in keywords:
        df.loc[len(df)] = {
            "rank":keyword.select_one(".ir_wa").text.replace("위",""),
            "keyword":keyword.select_one(".link_issue").text,
        }
    return df
```

In [12]:

```
daum_df = daum_top10()
daum_df
```

Out[12]:

	rank	keyword
0	1	라이프 온 마스
1	2	가을의 전설
2	3	고아라
3	4	810회 로또 당첨 번호
4	5	김명수
5	6	권세도
6	7	정태옥
7	8	스티븐스
8	9	인피니트
9	10	정경호

중복된 키워드 찾아서 출력하기

In [13]:

```
result = [keyword for keyword in daum_df["keyword"] if naver_df["keyword"].str.conta
result
```

Out[13]:

```
['라이프 온 마스', '김명수', '정태옥']
```

file download

In [14]:

```
def download(title, download_link):
    response = requests.get(download_link, stream=True)
    download_path = "./data/" + title
    size = 0
    with open(download_path, 'wb') as f:
        for chunk in response.iter_content(chunk_size=1024):
            if chunk:
                size += 1024
                f.write(chunk)
    return size
```

In [17]:

```
title = "iPhone X — Pay with a message.mp4"
download_link = "https://bit.ly/2sIazNq"
size = download(title, download_link)
print("download done : {} Mbyte".format(round(size/1024/1024,2)))
```

```
download done : 1.63 Mbyte
```