Crawling Day 3

삼성전자 주가 crawling

```
import requests
from bs4 import BeautifulSoup as bs

url = "https://finance.naver.com/item/main.nhn?code=005930"
html = bs(requests.get(url).text)

html.select('em.no_up>span.blind')[0].text
```

```
'61,300'
```

Stock API

pykrx

삼성전자

```
!pip install pykrx # 주가 조회 모듈
from pykrx import stock
stock.get_market_ohlcv('20230201', '20230228', '005930')
```

날짜	시가	고가	저가	종가	거래량	거래대금	등락률
2023-02-01	62600	62700	61000	61800	18570133	1145781815984	1.31
2023-02-02	63200	63900	62600	63500	23285983	1474629229812	2.75
2023-02-03	63900	64000	63000	63800	15194598	967336146677	0.47
2023-02-06	62800	63000	61600	61600	15529356	967890468300	-3.45
2023-02-07	61900	62500	61600	61900	14491039	898617000700	0.49
2023-02-08	62800	63300	62400	63100	12243125	770472521000	1.94
2023-02-09	63000	63300	62300	63000	12164865	764921355400	-0.16
2023-02-10	62600	63000	62400	62800	9382576	587897501100	-0.32
2023-02-13	62900	63000	62300	62900	10730362	672253190500	0.16
2023-02-14	63600	63900	63200	63200	9126664	579858006300	0.48
2023-02-15	63900	63900	62000	62200	13208103	826031235400	-1.58
2023-02-16	62500	63700	62400	63700	13798831	873759607962	2.41
2023-02-17	62900	63300	62400	62600	10791265	677809979471	-1.73
2023-02-20	62900	63000	61800	62700	12908073	806198780800	0.16
2023-02-21	62700	62800	62000	62100	7665046	476711896776	-0.96
2023-02-22	61500	61800	61000	61100	11959088	732238903100	-1.61
2023-02-23	61700	62500	61500	62000	13047099	809465376980	1.47
2023-02-24	62300	62600	61300	61300	10308143	638634052687	-1.13
2023-02-27	60800	60800	60200	60500	11155697	674292520900	-1.31
2023-02-28	60800	61400	60500	60600	13715861	834902635220	0.17

---

yfinance

Apple

```
!pip install yfinance
import yfinance as yf
yf.download('AAPL', start='2023-02-28')
```

Date	Open	High	Low	Close	Adj Close	Volume
2023-02-28	147.050003	149.080002	146.830002	147.410004	147.410004	50547000
2023-03-01	146.830002	147.228500	145.009995	145.309998	145.309998	51688609

tesla

```
yf.download('TSLA', start='2023-02-28')
```

Date	Open	High	Low	Close	Adj Close	Volume
2023-02-28	210.589996	211.229996	203.750000	205.710007	205.710007	153144900
2023-03-01	206.210007	207.198700	198.520401	202.770004	202.770004	147701973

COIN API

빗썸 비트코인 crawling

```
import requests
from bs4 import BeautifulSoup as bs
from selenium import webdriver

driver = webdriver.Chrome('chromedriver.exe')
url = "https://www.bithumb.com/trade/order/BTC_KRW"
driver.get(url)
txt = driver.page_source
html = bs(txt)

html.select('div.current_price>h3')[0].text
```

```
'31,122,000'
```

빗썸 Open API

```
!pip install pybithumb

import pybithumb as pb
import time

pb.get_tickers() # 종목명 출력

pb.get_current_price('BTC')
```

비트코인 가격 반복 조회

```
# 비트코인 가격 반복 조회
while True:
    print(pb.get_current_price('BTC')) # 비트코인 가격 조회
    time.sleep(3) # 3초 마다
```

```
31100000.0
31100000.0
31087000.0


---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
/var/folders/_j/xfcnlqrd2nn2csm0b_0f87hc0000gn/T/ipykernel_1134/3806061896.py in <module>
      3 while True:
      4     print(pb.get_current_price('BTC')) # 비트코인 가격 조회
----> 5     time.sleep(3)

KeyboardInterrupt:
```

모든 코인 가격 조회

```
# 모든 코인 가격 조회
tickers = pb.get_tickers()
for ticker in tickers:
    print(ticker, pb.get_current_price(ticker))
    time.sleep(1) # 시간 지연
```

```
BTC 31084000.0
ETH 2181000.0
ETC 27870.0
XRP 501.4


---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
/var/folders/_j/xfcnlqrd2nn2csm0b_0f87hc0000gn/T/ipykernel_1134/101441563.py in <module>
      4 for ticker in tickers:
      5     print(ticker, pb.get_current_price(ticker))
----> 6     time.sleep(1) # 시간 지연

KeyboardInterrupt:
```

자세한 코인 가격 조회

```
# 자세한 코인 가격 조회
print(pb.get_market_detail('BTC'))

# 시가/고가/저가/종가(현재가)/거래량
# 튜플로 리턴
```

```
(31180000.0, 31272000.0, 30937000.0, 31093000.0, 238.92958343)
```

모든 코인 가격 자세히 조회

```
# 모든 코인 가격 자세히 조회
tickers = pb.get_tickers()

for ticker in tickers:
    print(ticker, pb.get_market_detail(ticker))
    time.sleep(1)
```

```
BTC (31180000.0, 31272000.0, 30937000.0, 31046000.0, 231.31268343)
ETH (2180000.0, 2203000.0, 2169000.0, 2177000.0, 1891.82528367)
ETC (27990.0, 28100.0, 27790.0, 27830.0, 19683.15415325)


---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
/var/folders/_j/xfcnlqrd2nn2csm0b_0f87hc0000gn/T/ipykernel_1134/1621397851.py in <module>
      4 for ticker in tickers:
      5     print(ticker, pb.get_market_detail(ticker))
----> 6     time.sleep(1)

KeyboardInterrupt:
```

예외처리

```
# 네트워크 이용시 예외 발생 빈도가 높다
# 1. time.sleep 이용한 시간 지연
# 2. try except 이용한 예외 처리

tickers = pb.get_tickers()

try :
    for ticker in tickers:
        print(ticker, pb.get_market_detail(ticker))
        time.sleep(1) # 1
except :
    print('예외가 발생 되었습니다.') # 2
```

```
BTC (31180000.0, 31272000.0, 30937000.0, 31096000.0, 235.90558343)
ETH (2180000.0, 2203000.0, 2169000.0, 2181000.0, 1917.26478367)
ETC (27990.0, 28100.0, 27790.0, 27880.0, 20065.47505325)
예외가 발생 되었습니다.
```

상승장 하락장 알림 프로그램
이동 평균성 사용 : 이동 평균 값보다 현재가가 높다면 상승장
이동 평균 값보다 현재가가 낮으면 하락장

03/01 1000
03/02 1100
03/03 1200
03/04 1100
03/05 1000

03/06 1400 > 상승장

03/06 1000 > 하락장

```
pb.get_ohlcv('BTC')
```

time	open	high	low	close	volume
2013-12-27 00:00:00	737000.0	755000.0	737000.0	755000.0	3.780000
2013-12-28 00:00:00	750000.0	750000.0	750000.0	750000.0	12.000000
2013-12-29 00:00:00	750000.0	750000.0	728000.0	739000.0	19.058000
2013-12-30 00:00:00	740000.0	772000.0	740000.0	768000.0	9.488973
2013-12-31 00:00:00	768000.0	800000.0	763000.0	768000.0	18.650350
...	...	...	...	...	...
2023-02-26 00:00:00	30498000.0	31047000.0	30471000.0	30890000.0	838.625344
2023-02-27 00:00:00	30897000.0	31580000.0	30833000.0	31544000.0	1205.188234
2023-02-28 00:00:00	31539000.0	31632000.0	30838000.0	31081000.0	1126.423476
2023-03-01 00:00:00	31092000.0	31580000.0	30765000.0	31193000.0	1158.565146
2023-03-02 11:00:00	31180000.0	31272000.0	30937000.0	31097000.0	250.009834

3263 rows × 5 columns

```
btc = pb.get_ohlcv('BTC')
close = btc['close']
print(close)
```

```
time
2013-12-27 00:00:00      755000.0
2013-12-28 00:00:00      750000.0
2013-12-29 00:00:00      739000.0
2013-12-30 00:00:00      768000.0
2013-12-31 00:00:00      768000.0
                          ...
2023-02-26 00:00:00    30890000.0
2023-02-27 00:00:00    31544000.0
2023-02-28 00:00:00    31081000.0
2023-03-01 00:00:00    31193000.0
2023-03-02 11:00:00    31096000.0
Name: close, Length: 3263, dtype: float64
```

수동으로 이동평균 계산

```
# 수동으로 이동평균 계산
btc = pb.get_ohlcv('BTC')
close = btc['close']

print((close[0]+close[1]+close[2]+close[3]+close[4])/5)
print((close[1]+close[2]+close[3]+close[4]+close[5])/5)
print((close[2]+close[3]+close[4]+close[5]+close[6])/5)
```

```
756000.0
760400.0
766000.0
```

자동으로 이동평균 계산

```
# 자동으로 이동평균 계산
btc = pb.get_ohlcv('BTC')
close = btc['close']

roll5 = close.rolling(5)
mean5 = roll5.mean()
print(mean5)
```

```
time
2013-12-27 00:00:00          NaN
2013-12-28 00:00:00          NaN
2013-12-29 00:00:00          NaN
2013-12-30 00:00:00          NaN
2013-12-31 00:00:00     756000.0
                           ...
2023-02-26 00:00:00    31049000.0
2023-02-27 00:00:00    31101400.0
2023-02-28 00:00:00    31052600.0
2023-03-01 00:00:00    31041200.0
2023-03-02 11:00:00    31158000.0
Name: close, Length: 3263, dtype: float64
```

상승 하락 판단

```
# 상승 하락 판단
btc = pb.get_ohlcv('BTC')
close = btc['close']

roll5 = close.rolling(5)
mean5 = roll5.mean()
last_mean5 = mean5[-2]

# 비트코인 현재가
price = pb.get_current_price('BTC')

# 비교
if price>last_mean5:
    print('상승장')
else :
    print('하락장')
```

```
상승장
```

Starbucks

파이썬으로 브라우저 버튼 클릭

```
area_btn = '#container > div > form > fieldset > div > section > article.find_store_cont > article > header.loca_search > h3 > a'
driver.find_element(By.CSS_SELECTOR, area_btn).click()
```

```
import requests
from bs4 import BeautifulSoup as bs
import pandas as pd
from selenium.webdriver.common.by import By
import time

driver = webdriver.Chrome('chromedriver.exe')
url = "https://www.starbucks.co.kr/store/store_map.do"
driver.get(url)

# 파이썬으로 브라우저에서 버튼 클릭하기
area_btn = '#container > div > form > fieldset > div > section > article.find_store_cont > article > header.loca_search > h3 > a'
driver.find_element(By.CSS_SELECTOR, area_btn).click()
time.sleep(2)
seoul_btn = '#container > div > form > fieldset > div > section > article.find_store_cont > article > article:nth-child(4) > div.loca_
driver.find_element(By.CSS_SELECTOR, seoul_btn).click()
time.sleep(2)
all_btn = '#mCSB_2_container > ul > li:nth-child(1) > a'
driver.find_element(By.CSS_SELECTOR, all_btn).click()
time.sleep(2)

txt = driver.page_source
html = bs(txt)
len(html.select('ul.quickSearchResultBoxSidoGugun>li'))
```

```
600
```