WorkBooks
1

2020. 07. 01.
Fri.
Prepared by DaeKyeong Kim
Ph.D.

Contents

Section 1

Collection

1. Crawling 개관
2. Workbooks

학습목표

- 이 워크샵에서는 분석할 데이터를 웹에서 수집하는 크롤링 방법을 알 수 있습니다.
- 개발자를 위해 제공하는 API로 웹 데이터 크롤링을 할 수 있습니다.
- API를 제공하지 않는 웹 페이지를 크롤링할 수 있습니다.
- BeautifulSoup 라이브러리로 정적 웹 페이지를 크롤링할 수 있습니다.
- Selenium 라이브러리로 동적 웹 페이지를 크롤링할 수 있습니다.

Subsection 1

Crawling 개관

크롤링이란

크롤링

웹에서 데이터를 수집하는 작업
크롤링(Crawling)이란 사전적 의미로 기어다니다를 뜻하고, Web에서는 돌아다니면서 원하는 정보를 수집하는 행위를 의미한다.
크롤러 또는 스파이더라는 프로그램으로 웹 사이트에서 데이터를 추출
크롤링의 대상은 위에서 언급한 대로 웹 상에 존재하는 정보들이며, 해당 정보는 다양한 형태로 존재할 수 있다.(이미지, 텍스트, API 등)

웹 API

웹 API는 일반적으로 HTTP 통신을 사용하는데 사용
지도, 검색, 주가, 환율 등 다양한 정보를 가지고 있는 웹 사이트의 기능을 외부에서 쉽게 사용할 수 있도록 사용 절차와
규약을 정의한 것

크롤링이란

웹 API

종류	주소
네이버 개발자 센터	https://developers.naver.com
카카오 앱 개발 플랫폼 서비스	https://developers.kakao.com
페이스북 개발자 센터	https://developers.facebook.com
트위터 개발자 센터	https://developer.twitter.com
공공데이터포털	https://www.data.go.kr
세계 날씨	http://openweathermap.org
유료/무료 API 스토어	http://mashup.or.kr http://www.apistore.co.kr/api/apiList.do

스크린 스크래핑(Screen Scraping)

- 데이터가 파일로 또는 API를 통해 분석 프로그램이 판독할 수 있는 형식으로 제공될 때가 있습니다. 이 프로세스를 스크린 스크래핑, 웹 스크래핑, 데이터 스크래핑 등으로 부릅니다.
- 스크린 스크래핑은 본래 컴퓨터 터미널 화면상의 텍스트 데이터를 판독하는 것이 목적이었지만, 요즈음은 HTML 웹 페이지에 표시되는 데이터에 훨씬 더 보편적입니다.

- Martin Heller

크롤링이란?

크롤링 종류

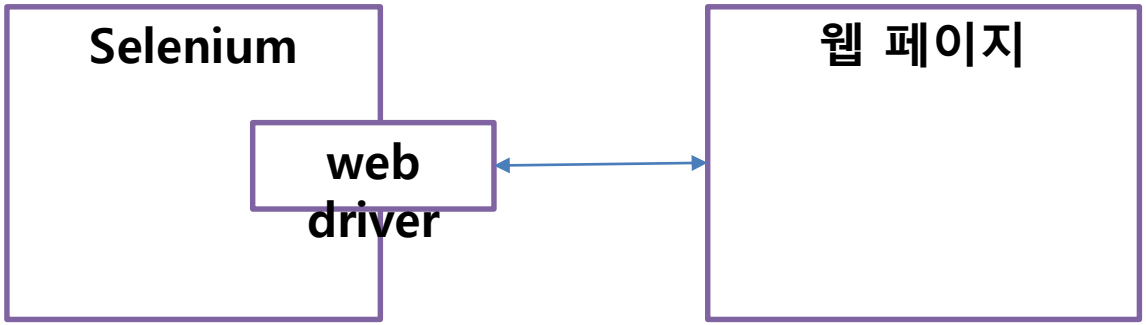
❖ 크롤링은 크게 두 가지로 나누어 질 수 있다. (정적 크롤링 VS 동적 크롤링)

❖ 정적 크롤링

- 특별한 절차 없이 특정 URL을 통해 데이터 수집 가능
- 새로고침하지 않으면 페이지 안의 데이터는 변하지 않는다.
- 속도가 빠르다.
- 수집 대상에 한계 존재한다.
- 사용 가능 라이브러리 : requests

❖ 동적 크롤링

1. 특별한 절차 없이 특정 URL을 통해 데이터 수집 불가능(네이버 메일의 경우)
2. 속도가 느리다.
3. 수집 대상에 한계가 거의 존재하지 않는다.
   - 사용 가능 라이브러리 : selenium



크롤링이란?

라이브러리 설치와 소개

```
!pip install requests
!pip install bs4
!pip install selenium
```

크롤링이란?

HTML 태그를 분석하는 모듈 BS4

```
from bs4 import BeautifulSoup

html='<div class="service_area"><a id="NM_set_home_btn" href="https://help.naver.com/support/welcomePage/guide.help" class="link_set
" data-clk="top.mkhome">네이버를 시작페이지로</a><i class="sa_bar"></i><a href="https://jr.naver.com" class="link_jrnaver" data-
clk="top.jrnaver"><i class="ico_jrnaver"></i><span class="blind">쥬니어네이버
</span></a><a href="https://happybean.naver.com" class="link_happybin" data-
clk="top.happybean"><i class="ico_happybin"></i><span class="blind">해피빈</span></a></div>'

soup=BeautifulSoup(html, 'html.parser')

print(soup.prettify())


soup.a

soup.find_all("a")
```

크롤링이란?

HTML 태그를 분석하는 모듈 BS4

```
#-------------------------------------------------------------------
#
#(1) 원하는 정보가 있는 태그 분석 (태그명, 속성값중 - 클래스값 또는 아이디값)
#보통 태그를 구분하는 기준으로 클래스명 혹은 아이디값으로 사용

#find_all() : 일치하는 모든 태그를 찾아 리스트로 반환
#find_all(태그명, id=아이디값) #태그의 아이디가 있는 경우
#find_all(태그명, class_= 클래스값) #태그의 클래스 값이 있는 경우
#find_all(태그명, attrs= {속성명:속성값}) #태그의 클래스나 아이디가 없는 경우
#                          특정 속성값을 통해 찾기

#find() : 일치하는 태그 딱하나 찾는 함수
#find(태그명, id=아이디값) #태그의 아이디가 있는 경우
#find(태그명, class_= 클래스값) #태그의 클래스 값이 있는 경우
#find(태그명, attrs= {속성명:속성값}) #태그의 클래스나 아이디가 없는 경우
#                          특정 속성값을 통해 찾기

#select() : css 설렉터 방식으로 태그 경로를 통해 원하는 모든 태그를 찾아 리스트로 반환
#select('태그명.클래스명 > 태그명#아이디값 > 태그명')

#태그에서 속성값 가져올때
#tag['속성명']

#태그의 값을 가져오는 경우
#tag.text
```

크롤링이란?

HTML 태그를 분석하는 모듈 BS4

```
from bs4 import BeautifulSoup

url = 'https://www.naver.com'
url_data = requests.get(url)
html=url_data.text
soup = BeautifulSoup(html, 'html.parser') #bs4 객체 생성

soup.prettify()

with open('네이버.html', 'w', encoding='utf-8') as f :
   f.write(soup.prettify())

#뉴스 스탠드 언론사명 가져오기
#find_all() : 해당 하는 태그를 모두 찾아서 리스트로 반환
newsTags = soup.find_all('div',
                  class_='thumb_box _NM_NEWSSTAND_THUMB _NM_NEWSSTAND_THUMB_press_valid')

for tag in newsTags :
   #find() : 원하는 태그를 딱 하나 찾아준다.
   imgTag = tag.find('img', class_='news_logo')
   #print(imgTag)
   print(imgTag['alt']) #이미지 태그의 속성값 중 alt 값을 가져옴

#-----------------------------------------------------
#select() : css 셀럭터 방식 태그의 경로를 통해서 원하는 태그 다 찾는다.
imgTags = soup.select('div.thumb_box._NM_NEWSSTAND_THUMB._NM_NEWSSTAND_THUMB_press_
valid'
               +' > a.thumb > img.news_logo')
for tag in imgTags :
   print(tag['alt'])
```

크롤링이란?

HTML 태그를 분석하는 모듈 BS4

```
import requests
from lxml import html

page = requests.get('https://movie.naver.com/movie/bi/mi/basic.naver?code=208431#story')
tree = html.fromstring(page.text)

text = tree.xpath('//*[@id="content"]/div[1]/div[4]/div[1]/div/div/p/text()')

print(text)
```

크롤링이란?

크롤링 허용 여부 확인하기

- 크롤링 허용 여부 확인하기
- 웹 페이지를 크롤링하기 전에 크롤링 허용 여부를 확인하기 위해 주소 창에 '크롤링할 주소/ robots.txt'를 입력
- 만약 robots.txt 파일이 없다면 수집에 대한 정책이 없으니 크롤링을 해도 된다는 의미

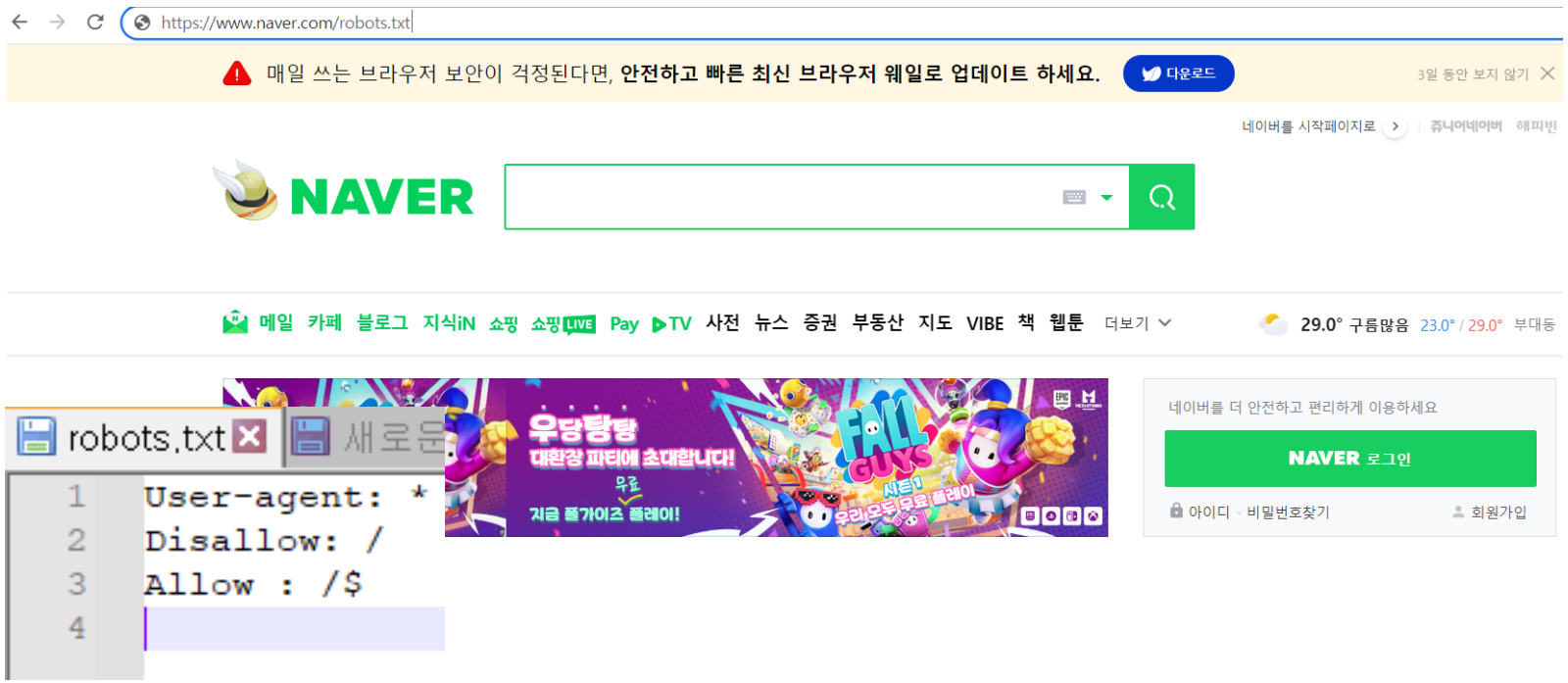
표시	허용 여부
User-agent: * Allow: / 또는 User-agent: * Disallow:	모든 접근 허용
User-agent: * Disallow: /	모든 접근 금지
User-agent: * Disallow: /user/	특정 디렉토리만 접근 금지

Unit 2

크롤링이란?

크롤링 허용 여부 확인하기

- 크롤링 허용 여부 확인하기
- 웹 페이지를 크롤링하기 전에 크롤링 허용 여부를 확인하기 위해 주소 창에 '크롤링할 주소/ robots.txt'를 입력
- 만약 robots.txt 파일이 없다면 수집에 대한 정책이 없으니 크롤링을 해도 된다는 의미

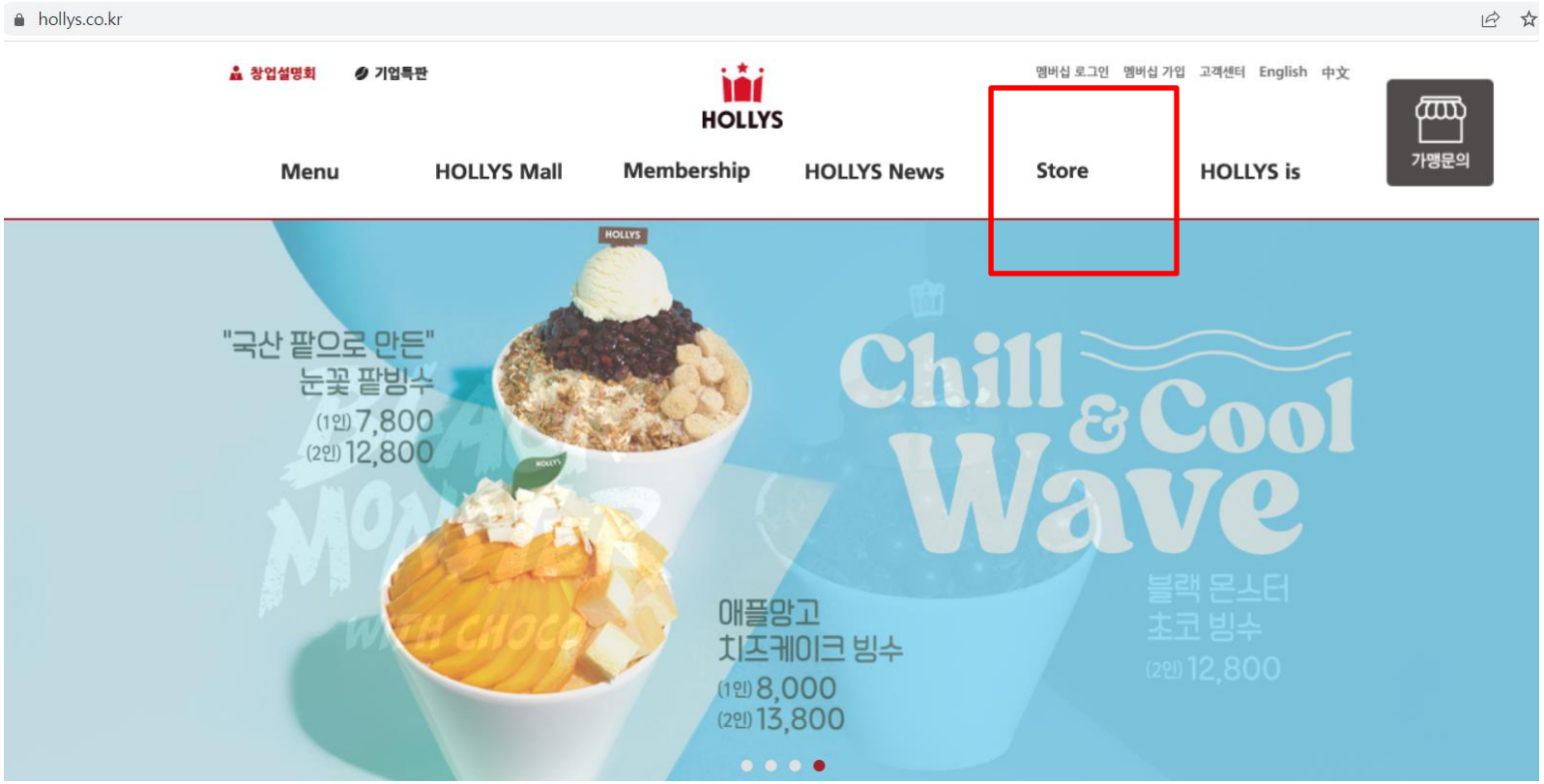


Unit 2

크롤링이란?

매장 정보 찾기

❖ https://www.hollys.co.kr/에서 [Store]클릭

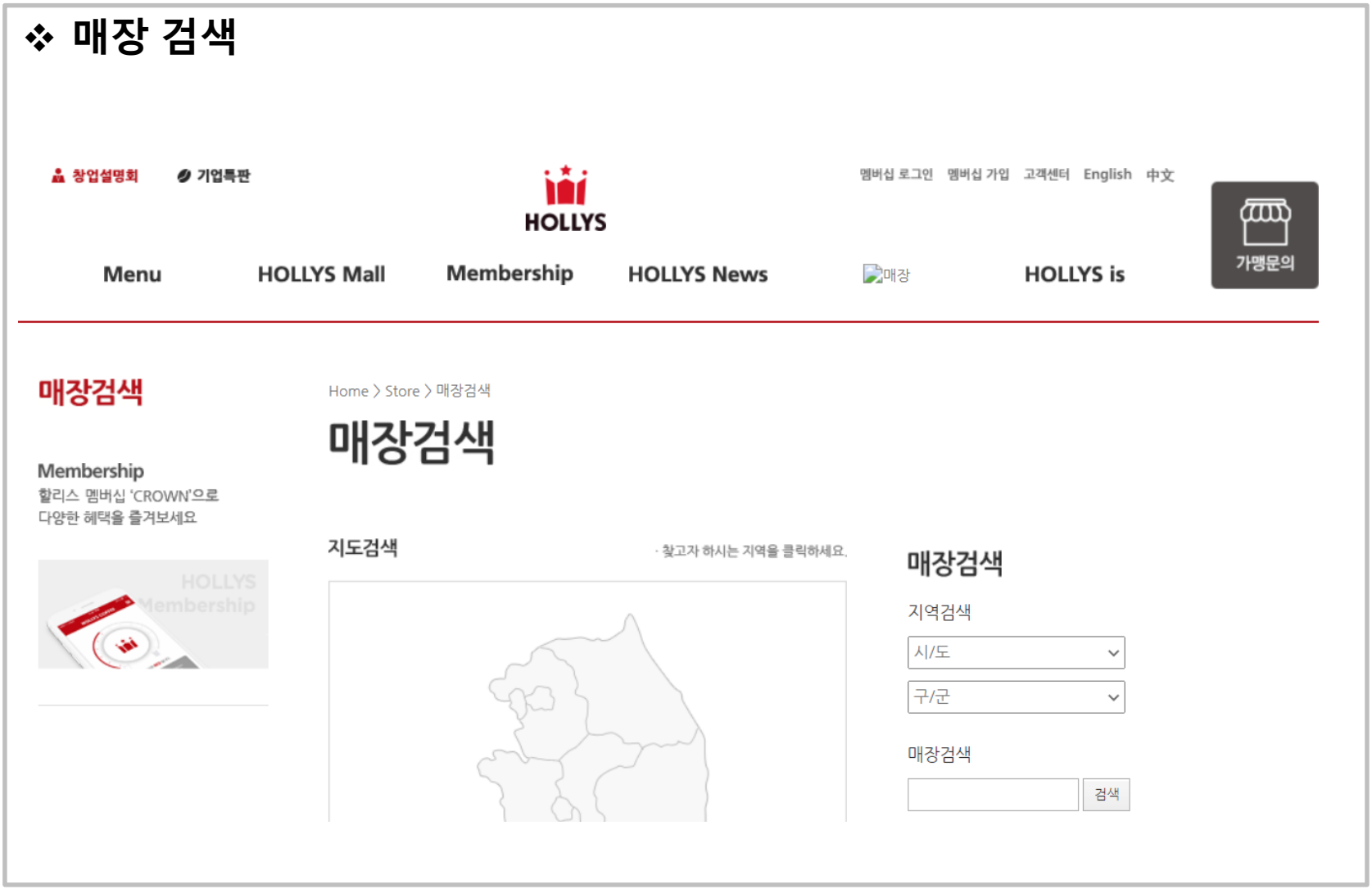


Unit 2

크롤링이란?

매장 정보 찾기

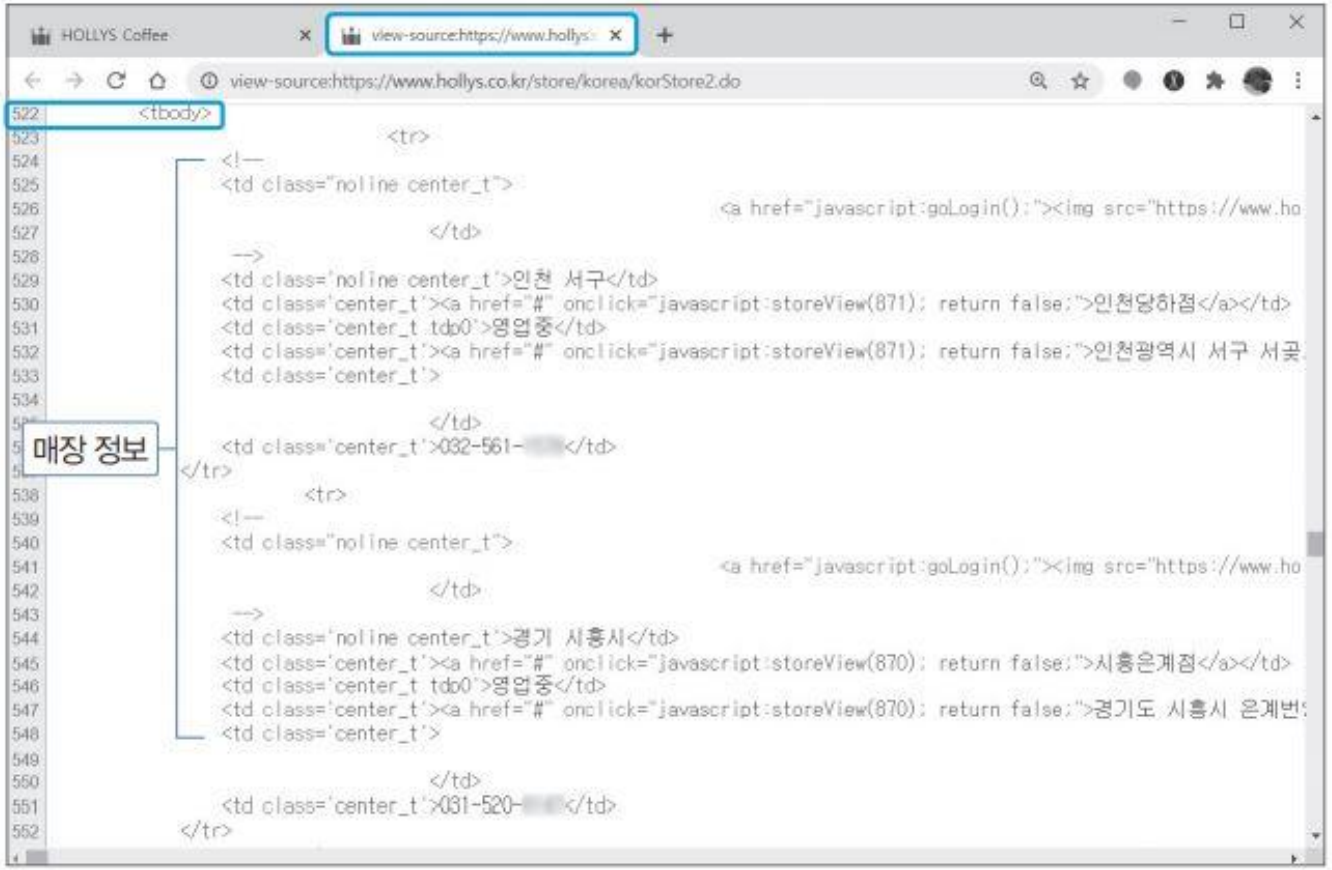
❖ 매장 검색



크롤링이란?

매장 정보 찾기

❖ HTML 코드 확인하기

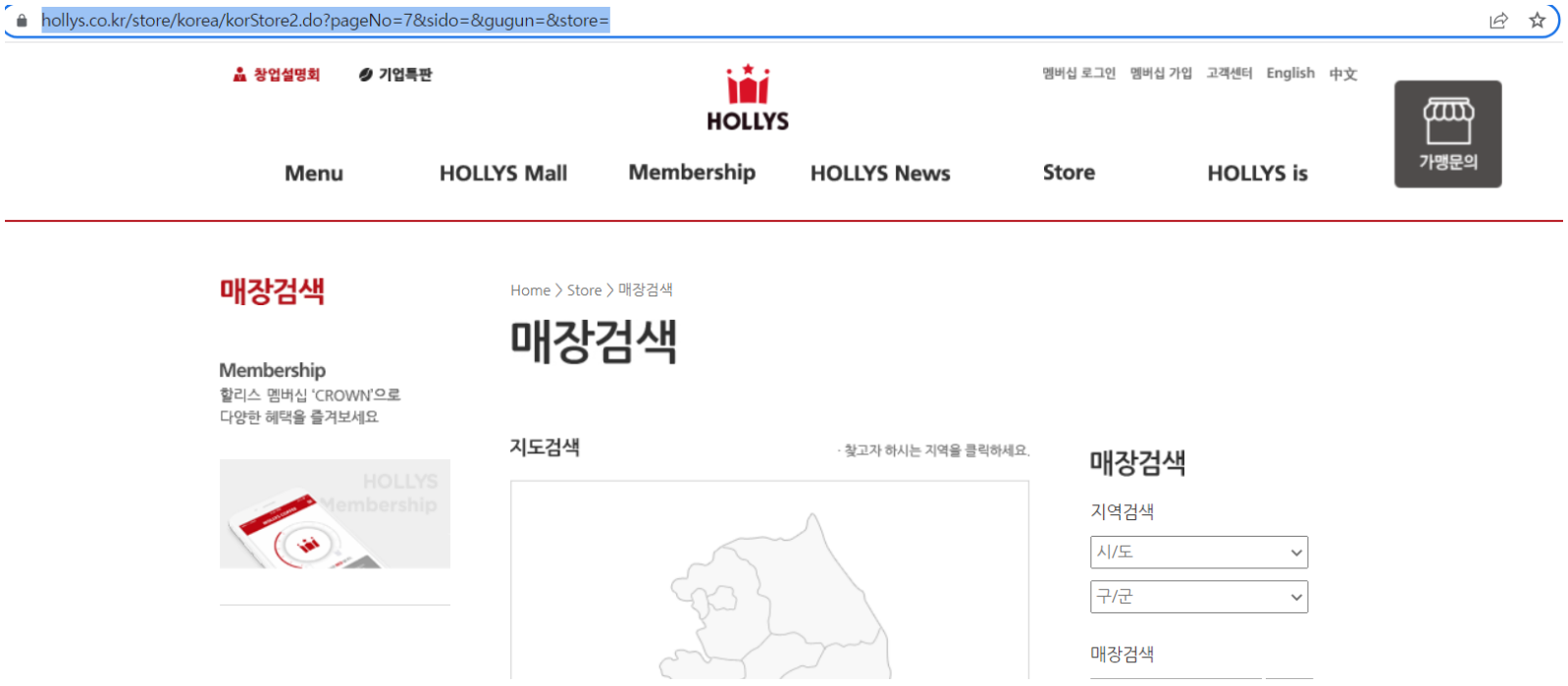


Unit 2

크롤링이란?

매장 정보 찾기

❖ **https://www.hollys.co.kr/store/korea/korStore2.do?pageNo=7&sido=&gugun=&store=**



크롤링이란?

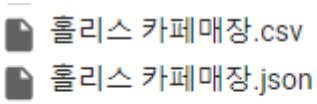
매장 정보 찾기

```
from bs4 import BeautifulSoup

itemList = []
for page in range(1, 53) :
    url =
f'https://www.hollys.co.kr/store/korea/korStore2.do?pageNo={page}&sido=&gugun=&store='
    url_data = requests.get(url)
    html=url_data.text
    soup = BeautifulSoup(html, 'html.parser') #bs4 객체 생성
    trs = soup.select('tbody')

    for tr in trs :
        tds = tr.find_all('td')
        # print(tds)
        itemDic = {}
        itemDic['지역'] = tds[0].text
        itemDic['매장명'] = tds[1].text
        itemDic['매장현황'] = tds[2].text #영업중, 오픈예정
        itemDic['주소'] = tds[3].text
        itemDic['매장 서비스'] = ''
        imgs = tds[4].find_all('img')
        if imgs != None :
            for img in imgs :
                itemDic['매장 서비스'] += img['alt'] + " "
        itemDic['매장 서비스'] = itemDic['매장 서비스'].strip()
        itemDic['전화번호'] = tds[5].text
        itemList.append(itemDic)

print('총 매장 개수 : ', len(itemList))
```

크롤링이란?

매장 정보 찾기

```
itemList

#-----------------------------------------------------
#데이터 저장
#pip install pandas #대표적인 데이터 분석 관련 라이브러리
import pandas as pd
df = pd.DataFrame(itemList) #데이터프레임은 테이블 구조
df.head() #가장 앞에 있는 5개 데이터 조회

#csv 파일 포맷으로 저장
df.to_csv('./홀리스 카페매장.csv', encoding='utf-8', index=False)
#index : 데이터프레임의 인덱스 저장 유무

#json 파일 포맷으로 저장
df.to_json('./홀리스 카페매장.json',
   orient='records', force_ascii=False, indent=4)
#force_ascii : 한글 저장하기 위해 False
#indent : 데이터 4칸 띄어서 저장
```



Unit 2

크롤링이란?

매장 정보 찾기

문 1 | 새로운 2 | 홀리스 카페매장.csv

```
지역,매장명,매장현황,주소,매장 서비스,전화번호
부산 사하구,부산아트몰링영풍문고점,영업중,"부산광역시 사하구 낙동남로 1413 (하단동, 아트몰링) 13층",주차,051-201-0621
울산 남구,울산삼산로점,영업중,"울산광역시 남구 삼산로 283 (삼산동, 소망빌딩) 삼산동1525-8,1525-9",,052-275-9004
서울 강서구,목동삼성쉐르빌점,영업중,"서울특별시 양천구 목동동로 189 (신정동, 삼성쉐르빌1) 1층 101,102,103호",테라스 주차,
경기 고양시 덕양구,고양신원점,영업중,경기도 고양시 덕양구 권율대로 896-19 .,주차,02-381-7666
대전 서구,대전건양대병원점,영업중,"대전광역시 서구 관저동로 158 (관저동, 건양대학교병원) 신관 지하1층",주차,042-543-3138
충북 청주시 서원구,청주성화점,영업중,충청북도 청주시 서원구 신성화로 39 (성화동) 1~2층,,043-233-6296
경기 남양주시,남양주진접점,영업중,경기도 남양주시 진접읍 해밀예당1로 8-12 가동 1~2층,테라스 주차,031-523-3237
대구 동구,대구봉무공원점,영업중,대구광역시 동구 단산길 4 (봉무동) 1~2층,주차,053-984-3001
서울 강서구,개화산점,영업중,"서울특별시 강서구 양천로 28 (방화동, 벽산에어트리움) 1층",테라스 주차,02-2064-1455
경기 이천시,이천갈산점,영업중,경기도 이천시 향교로 161 (갈산동) 1층 .,주차,031-636-1791
경기 평택시,(하)평택휴게소1호점,영업중,경기도 평택시 청북읍 평택제천고속도로 6 (평택복합휴게시설) .,24시간 주차,031-683-480
경기 오산시,오산시청점,영업중,"경기도 오산시 성호대로 124 (원동, 신협빌딩) 101,102호",흡연시설,031-378-8778
서울 서초구,교대역점,영업중,"서울특별시 서초구 반포대로30길 81 (서초동, 웅진타워) 1층 .",주차,02-521-8944
경기 수원시 권선구,수원SKV1모터스점,영업중,경기도 수원시 권선구 평동로79번길 45 (평동) 1층 135호,주차,031-293-8810
서울 마포구,홍대역2번출구점,영업중,서울특별시 마포구 동교로 200 1~3층 .,24시간 주차,02-332-8079
경기 안산시 단원구,안산선부점,영업중,경기도 안산시 단원구 선부광장1로 56 (선부동) 할리스커피,테라스 흡연시설 주차,031-411-97
서울 서대문구,서대문농협생명빌딩점,영업중,"서울시 서대문구 통일로 87 미근동 257, NH농협생명빌딩 (주말 휴점)",주차,02-364-
서울 강서구,마곡나인스퀘어점,영업중,"서울특별시 강서구 공항대로 206 (마곡동, 나인스퀘어) 마곡동 800, 101,102,103호",주차,
```

홀리스 카페매장.csv

홀리스 카페매장.json

Subsection 2

Workbooks

NC SOFT 주식 데이터 예측하기

야후 fianace이용 :https://finance.yahoo.com/

pip install pandas-datareader

```
pip install pandas-datareader
```

```
Collecting pandas-datareader
  Using cached pandas_datareader-0.8.1-py2.py3-none-any.whl (107 kB)
Requirement already satisfied: pandas>=0.21 in c:\users\dec2819\anaconda3\envs\py37tf114\lib\site-packages (from pandas-datareade
r) (1.0.1)
Collecting requests>=2.3.0
  Using cached requests-2.23.0-py2.py3-none-any.whl (58 kB)
Requirement already satisfied: lxml in c:\users\dec2819\anaconda3\envs\py37tf114\lib\site-packages (from pandas-datareader) (4.5.
0)
Requirement already satisfied: pytz>=2017.2 in c:\users\dec2819\anaconda3\envs\py37tf114\lib\site-packages (from pandas>=0.21->pa
ndas-datareader) (2019.3)
Requirement already satisfied: python-dateutil>=2.6.1 in c:\users\dec2819\anaconda3\envs\py37tf114\lib\site-packages (from pandas
>=0.21->pandas-datareader) (2.8.1)
Requirement already satisfied: numpy>=1.13.3 in c:\users\dec2819\anaconda3\envs\py37tf114\lib\site-packages (from pandas>=0.21->p
andas-datareader) (1.18.1)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\dec2819\anaconda3\envs\py37tf114\lib\site-packages (from requests>=
2.3.0->pandas-datareader) (2019.11.28)
Collecting chardet<4,>=3.0.2
```

NC SOFT 주식 데이터 예측하기

야후 fianace이용 :https://finance.yahoo.com/

pip install yfinance --upgrade --no-cache-dir

```
pip install yfinance --upgrade --no-cache-dir
```

```
Collecting yfinance
  Downloading yfinance-0.1.54.tar.gz (19 kB)
Requirement already satisfied, skipping upgrade: pandas>=0.24 in c:\users\dec2819\anaconda3\envs\py37tf114\lib\site-packages (fr
m yfinance) (1.0.1)
Requirement already satisfied, skipping upgrade: numpy>=1.15 in c:\users\dec2819\anaconda3\envs\py37tf114\lib\site-packages (fro
yfinance) (1.18.1)
Requirement already satisfied, skipping upgrade: requests>=2.20 in c:\users\dec2819\anaconda3\envs\py37tf114\lib\site-packages (
rom yfinance) (2.23.0)
Collecting multitasking>=0.0.7
  Downloading multitasking-0.0.9.tar.gz (8.1 kB)
Requirement already satisfied, skipping upgrade: pytz>=2017.2 in c:\users\dec2819\anaconda3\envs\py37tf114\lib\site-packages (fr
m pandas>=0.24->yfinance) (2019.3)
Requirement already satisfied, skipping upgrade: python-dateutil>=2.6.1 in c:\users\dec2819\anaconda3\envs\py37tf114\lib\site-pa
kages (from pandas>=0.24->yfinance) (2.8.1)
Requirement already satisfied, skipping upgrade: chardet<4,>=3.0.2 in c:\users\dec2819\anaconda3\envs\py37tf114\lib\site-package
(from requests>=2.20->yfinance) (3.0.4)
Requirement already satisfied, skipping upgrade: idna<3,>=2.5 in c:\users\dec2819\anaconda3\envs\py37tf114\lib\site-packages (fr
```

NC SOFT 주식 데이터 예측하기

야후 fianace이용 :https://finance.yahoo.com/

```
from pandas_datareader import data
import datetime

import yfinance as yf
yf.pdr_override()

start_date = '2008-01-01'
name = '036570.KS'
nc = data.get_data_yahoo(name, start_date)
```

```
from pandas_datareader import data
import datetime
```

```
import yfinance as yf
yf.pdr_override()
```

```
start_date = '2008-01-01'
name = '036570.KS'
nc = data.get_data_yahoo(name, start_date)
```

```
[**********************100%**********************]  1 of 1 completed
```

NC SOFT 주식 데이터 예측하기

ns soft 주식

nc.head(3)

```
nc.head(3)
```

Date	Open	High	Low	Close	Adj Close	Volume
2008-01-02	49000.0	50100.0	48600.0	48700.0	43573.867188	83601
2008-01-03	48650.0	49500.0	48500.0	49150.0	43976.496094	49680
2008-01-04	48700.0	49200.0	48000.0	48750.0	43618.597656	88832

NC SOFT 주식 데이터 예측하기

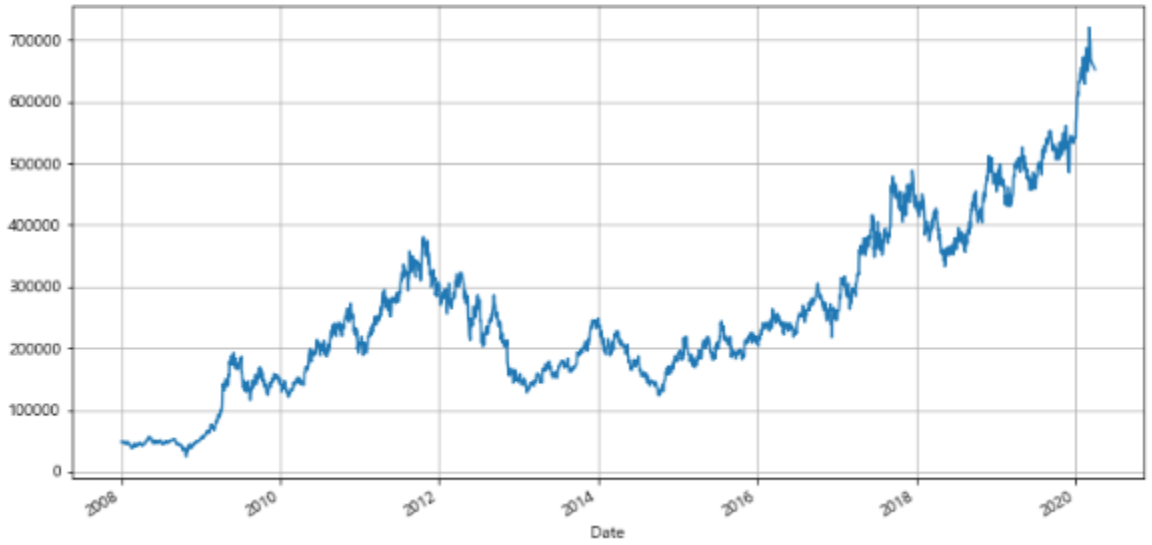
ns soft 주식

```
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline

nc['Close'].plot(figsize=(12, 6), grid=True)
```

```
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

```
nc['Close'].plot(figsize=(12, 6), grid=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x19f103c2bc8>
```



NC SOFT 주식 데이터 예측하기

ns soft 주식

```
nc_trunc = nc['2016-12-31']
nc_trunc.head(3)

df = pd.DataFrame({'ds':nc_trunc.index, 'y':nc_trunc['Close']})
df.reset_index(inplace=True)
del df['Date']
df.head(3)
```

```
nc_trunc = nc['2016-12-31']
nc_trunc.head(3)
```

Date	Open	High	Low	Close	Adj Close	Volume
2008-01-02	49000.0	50100.0	48600.0	48700.0	43573.867188	83601
2008-01-03	48650.0	49500.0	48500.0	49150.0	43976.496094	49680
2008-01-04	48700.0	49200.0	48000.0	48750.0	43618.597656	88832

```
df = pd.DataFrame({'ds':nc_trunc.index, 'y':nc_trunc['Close']})
df.reset_index(inplace=True)
del df['Date']
df.head(3)
```

	ds	y
0	2008-01-02	48700.0
1	2008-01-03	49150.0
2	2008-01-04	48750.0

fbprophet 임포트

```
from fbprophet import Prophet

m = Prophet()
m.fit(df)

future = m.make_future_dataframe(periods=365*2)
future.tail(3)

forecast = m.predict(future)
forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail(3)
```

fbprophet 임포트

```
from fbprophet import Prophet

m = Prophet()
m.fit(df)
```

```
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
```

```
<fbprophet.forecaster.Prophet at 0x1e56eb66048>
```

```
future = m.make_future_dataframe(periods=365*2)
future.tail(3)
```

	ds
2962	2018-12-27
2963	2018-12-28
2964	2018-12-29

```
forecast = m.predict(future)
forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail(3)
```

	ds	yhat	yhat_lower	yhat_upper
2962	2018-12-27	356362.934590	100499.069013	564381.075375
2963	2018-12-28	356189.700767	90744.858021	573384.868191
2964	2018-12-29	354510.422844	96204.286205	572199.032422

NC SOFT 주식 데이터 예측하기

플롯

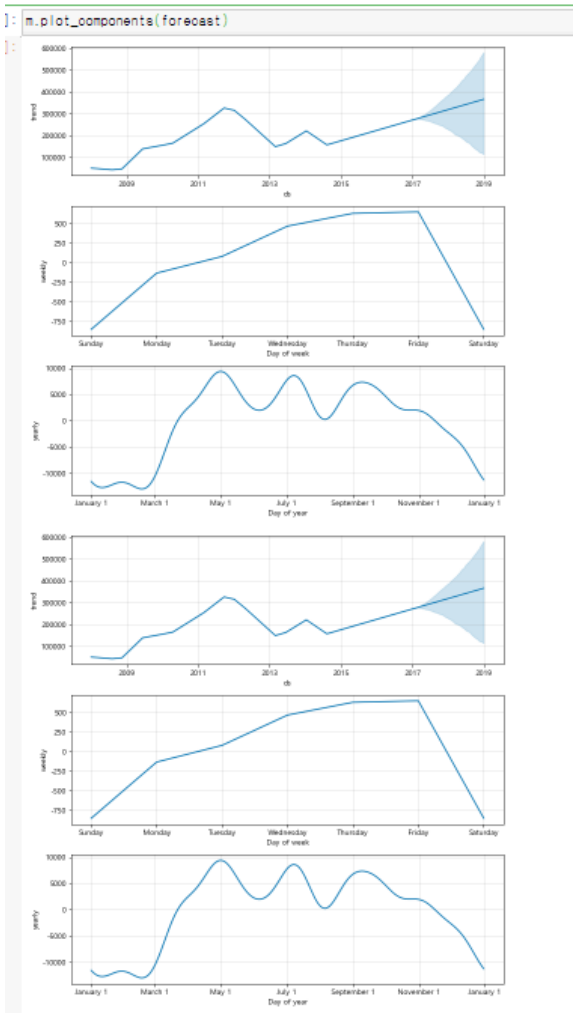
m.plot(forecast)

```
m.plot(forecast)
```

NC SOFT 주식 데이터 예측하기

플롯

m.plot_components(forecast)



NC SOFT 주식 데이터 예측하기

2017년도 기록을 가지고 실제데이터와 예측데이터를 비교

```
plt.figure(figsize=(12, 6))
plt.plot(nc.index, nc['Close'], label='real')
plt.plot(forecast['ds'], forecast['yhat'], label='forecast')
plt.grid()
plt.legend()
plt.show()
```

도시별 현재날씨 확인하기

Gathering Data

❖ **https://pythondojang.bitbucket.io/weather/observation/currentweather.html**

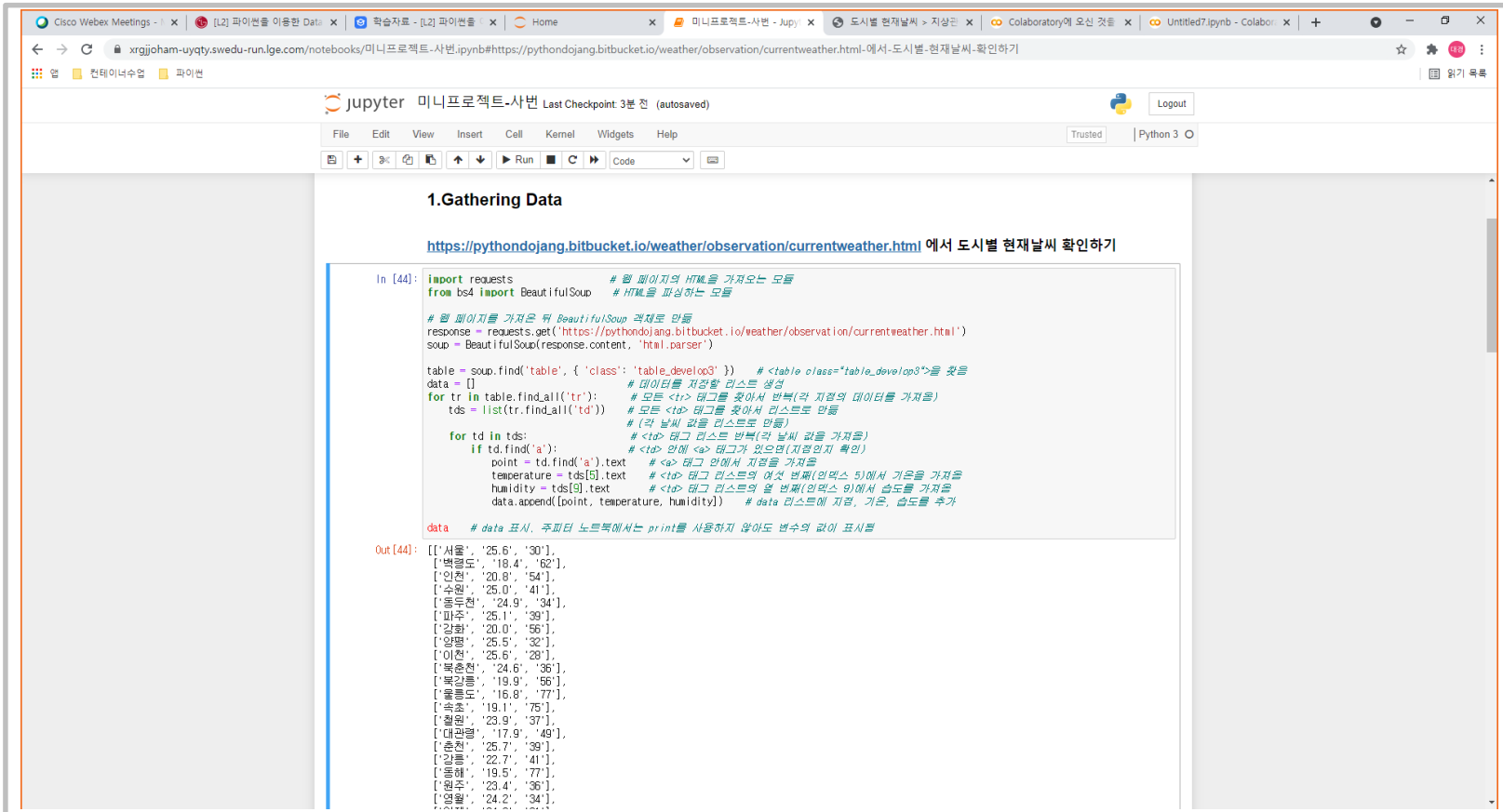
지점	날씨				기온(℃)			강수		바람		기압(hPa)
	현재일기	시정 km	운량 1/10	중하운량	현재 기온	이슬점 온도	불쾌 지수	일강수 mm	습도 %	풍향	풍속 m/s	해면 기압
서울	맑음	18.9	1	1	25.6	6.7	70		30	서남서	2.1	1010.1
백령도	구름조금	19.8	3	0	18.4	10.9	64		62	남서	5.2	1011.2
인천	맑음	20 이상	0	0	20.8	11.1	67		54	서남서	2.9	1011.0
수원	구름조금	12.6	3	3	25.0	10.8	71		41	북서	2.4	1010.6

도시별 현재날씨 확인하기

```
import requests                 # 웹 페이지의 HTML을 가져오는 모듈
from bs4 import BeautifulSoup   # HTML을 파싱하는 모듈

# 웹 페이지를 가져온 뒤 BeautifulSoup 객체로 만듦
response =
requests.get('https://pythondojang.bitbucket.io/weather/observation/currentweather.html')
soup = BeautifulSoup(response.content, 'html.parser')

table = soup.find('table', { 'class': 'table_develop3' })   # <table class="table_develop3">을 찾음
data = []                       # 데이터를 저장할 리스트 생성
for tr in table.find_all('tr'):     # 모든 <tr> 태그를 찾아서 반복(각 지점의 데이터를 가져옴)
   tds = list(tr.find_all('td'))   # 모든 <td> 태그를 찾아서 리스트로 만듦
                                   # (각 날씨 값을 리스트로 만듦)
   for td in tds:                  # <td> 태그 리스트 반복(각 날씨 값을 가져옴)
      if td.find('a'):             # <td> 안에 <a> 태그가 있으면(지점인지 확인)
         point = td.find('a').text    # <a> 태그 안에서 지점을 가져옴
         temperature = tds[5].text    # <td> 태그 리스트의 여섯 번째(인덱스 5)에서 기온을 가져옴
         humidity = tds[9].text       # <td> 태그 리스트의 열 번째(인덱스 9)에서 습도를 가져옴
         data.append([point, temperature, humidity])   # data 리스트에 지점, 기온, 습도를 추가

data    # data 표시. 주피터 노트북에서는 print를 사용하지 않아도 변수의 값이 표시됨
```

도시별 현재날씨 확인하기

도시별 현재날씨 확인하기



jupyter 미니프로젝트-사번 Last Checkpoint: 3분 전 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Run Code

1.Gathering Data

https://pythondojang.bitbucket.io/weather/observation/currentweather.html 에서 도시별 현재날씨 확인하기

In [44]:

```
import requests                 # 웹 페이지의 HTML을 가져오는 모듈
from bs4 import BeautifulSoup   # HTML을 파싱하는 모듈

# 웹 페이지를 가져온 뒤 BeautifulSoup 객체로 만듦
response = requests.get('https://pythondojang.bitbucket.io/weather/observation/currentweather.html')
soup = BeautifulSoup(response.content, 'html.parser')

table = soup.find('table', { 'class': 'table_develop3' })    # <table class="table_develop3">를 찾음
data = []                                                    # 데이터를 저장할 리스트 생성
for tr in table.find_all('tr'):                              # 모든 <tr> 태그를 찾아서 반복(각 지점의 데이터를 가져옴)
    tds = list(tr.find_all('td'))                            # 모든 <td> 태그를 찾아서 리스트로 만듦
                                                             # (각 날씨 값을 리스트로 만듦)
    for td in tds:                                           # <td> 태그 리스트 반복(각 날씨 값을 가져옴)
        if td.find('a'):                                     # <td> 안에 <a> 태그가 있으면(지점인지 확인)
            point = td.find('a').text                        # <a> 태그 안에서 지점을 가져옴
            temperature = tds[5].text                        # <td> 태그 리스트의 여섯 번째(인덱스 5)에서 기온을 가져옴
            humidity = tds[9].text                           # <td> 태그 리스트의 열 번째(인덱스 9)에서 습도를 가져옴
            data.append([point, temperature, humidity])      # data 리스트에 지점, 기온, 습도를 추가

data    # data 표시. 주피터 노트북에서는 print를 사용하지 않아도 변수의 값이 표시됨
```

Out[44]:

```
[['서울', '25.6', '30'],
 ['백령도', '18.4', '62'],
 ['인천', '20.8', '54'],
 ['수원', '25.0', '41'],
 ['동두천', '24.9', '34'],
 ['파주', '25.1', '39'],
 ['강화', '20.0', '56'],
 ['양평', '25.5', '32'],
 ['이천', '25.6', '28'],
 ['북춘천', '24.6', '36'],
 ['북강릉', '19.9', '56'],
 ['울릉도', '16.8', '77'],
 ['속초', '19.1', '75'],
 ['철원', '23.9', '37'],
 ['대관령', '17.9', '49'],
 ['춘천', '25.7', '39'],
 ['강릉', '22.7', '41'],
 ['동해', '19.5', '77'],
 ['원주', '23.4', '36'],
 ['영월', '24.2', '34'],
```

도시별 현재날씨 확인하기

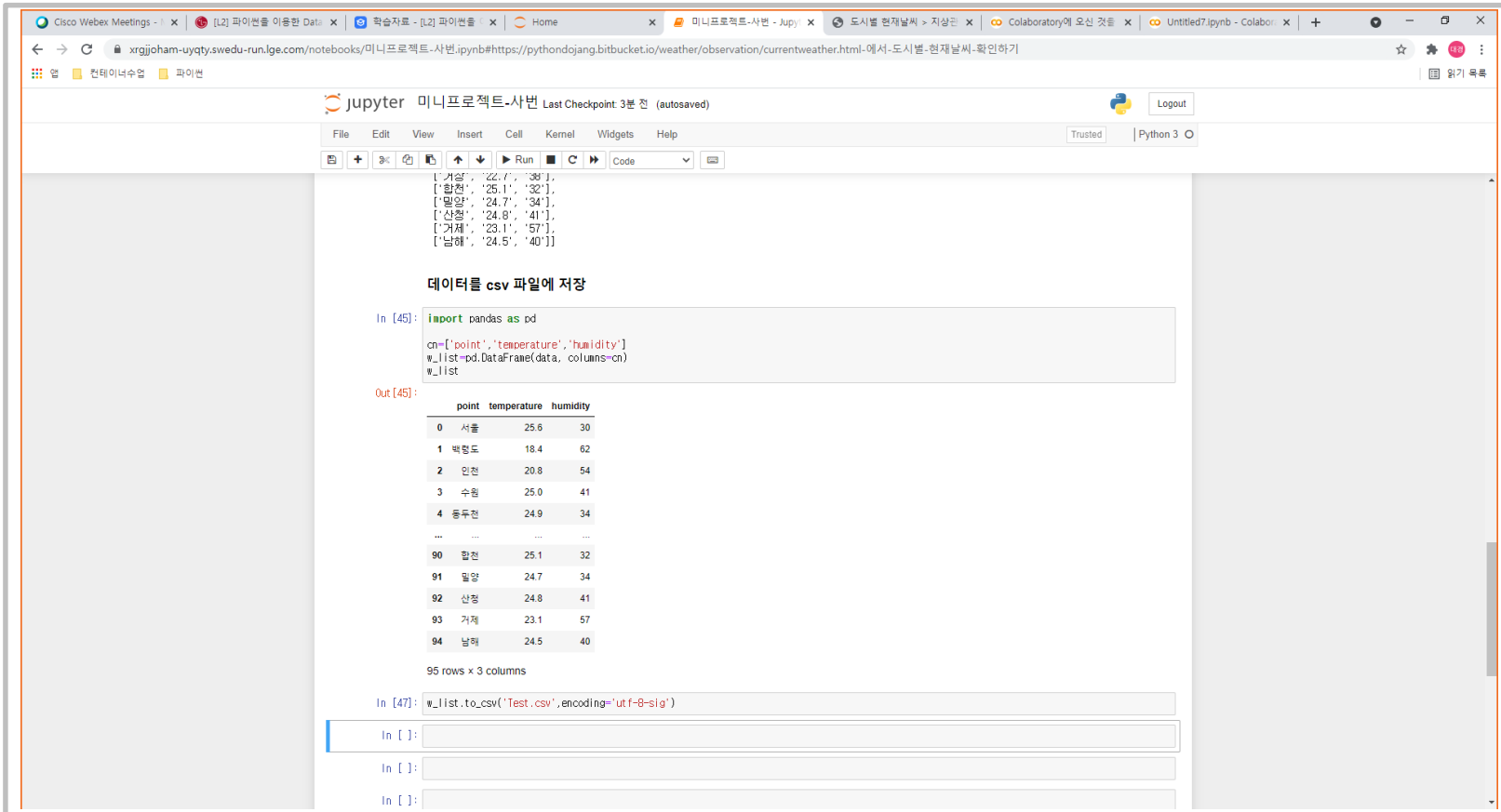
데이터를 csv 파일에 저장

```
import pandas as pd

cn=['point','temperature','humidity']
w_list=pd.DataFrame(data, columns=cn)
w_list




w_list.to_csv('Test.csv',encoding='utf-8-sig')
```

Unit 2

도시별 현재날씨 확인하기

데이터를 csv 파일에 저장



참고자료

참고자료

- http://www.ncs.go.kr
- NELLDALE/JOHN LEWIS지음, 조영석/김대경/박찬영/송창근 역, 단계별로배우는컴퓨터과학, 홍릉과학출판사,2018
- 혼자 공부하는 머신러닝+딥러닝 박해선 지음 | 한빛미디어 | 2020년 12월
- 머신러닝 실무 프로젝트 ,아리가 미치아키, 나카야마 신타, 니시바야시 다카시 지음 | 심효섭 옮김 | 한빛미디어 | 2018년 06월
- 파이썬을 활용한 머신러닝 쿡북 크리스 알본 지음 | 박해선 옮김 | 한빛미디어 | 2019년 09월
- 처음 배우는 머신러닝 김의중 지음 | 위키북스 | 2016년 07월
- 파이썬으로 배우는 머신러닝의 교과서 : 이토 마코토 지음 | 박광수(아크몬드) 옮김 | 한빛미디어 | 2018년 11월
- https://www.assemblyai.com/blog/pytorch-vs-tensorflow-in-2022/
- 기타 서적 및 웹 사이트 자료 다수 참조

참고자료

나눔고딕 인스톨

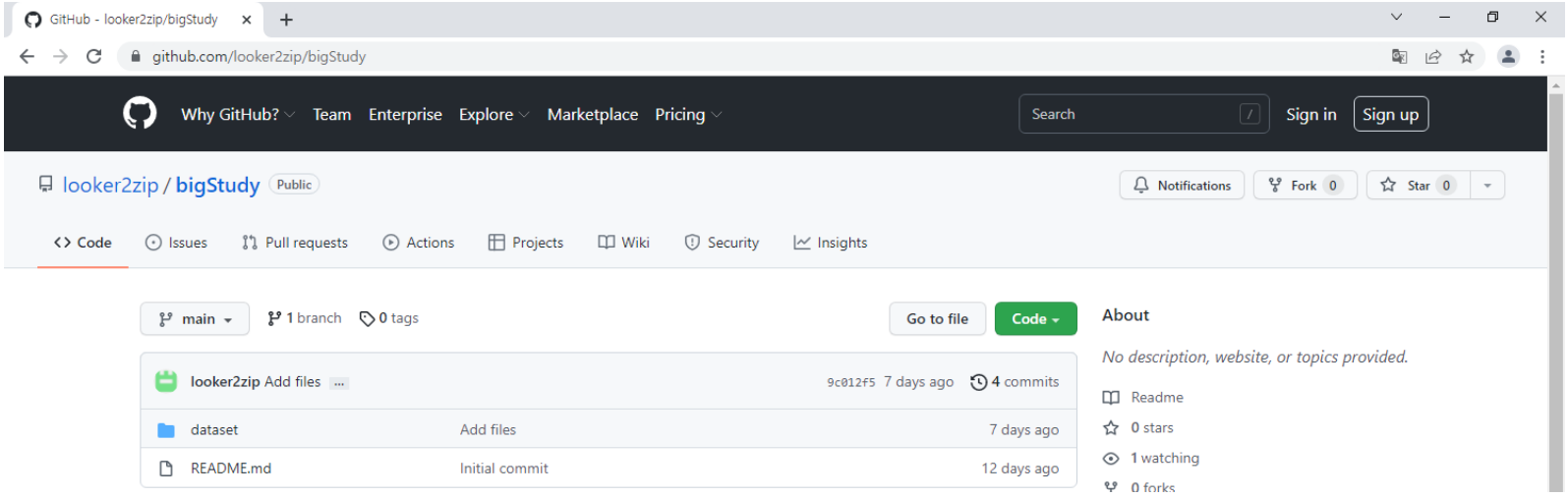
❖ 리눅스 다운로드

!rm -rf bigStudy

!git clone 'https://github.com/looker2zip/bigStudy.git'

❖ 윈도 다운로드

https://github.com/looker2zip/bigStudy



참고자료

1. 404에러 발생 시 방문기록 삭제 후 재접속 한다. 재접 속시 암호 물어보면... 암호는 goorm
2. !git clone https://github.com/onlookertozip/bigStudy.git
3. apt-get update
4. python3 -m pip install --upgrade pip
5. pip install selenium
6. apt-get update
7. apt install chromium-chromedriver
8. pip install beautifulsoup4
9. pip install fancyimpute
10. pip install mglearn

```
apt-get install -y fonts-nanum
fc-cache -fv
rm ~/.cache/matplotlib -rf

import matplotlib.pyplot as plt
plt.rc('font', family='NanumBarunGothic')
```

참고자료

1. vpip install lxml
2. pip3 install konlpy
3. pip install wordcloud
4. pip install nltk
5. apt-get update
6. apt-get install g++ openjdk-8-jdk
7. pip3 install konlpy JPype1-py3
8. bash <(curl -s https://raw.githubusercontent.com/konlpy/konlpy/master/scripts/mecab.sh)

Unit C

Colab에서 한 폴더 안 모든 파일 가져오기

```
import os

path = './xlsxdataset/'
file_list = os.listdir(path)
file_list_py = [file for file in file_list if file.endswith('.xlsx')]

for i in file_list_py:
  workbook = open_workbook(path + i)
```

- **Mobile: 010-9591-1401**
- **E-mail: onlooker2zip@naver.com**