

기말 프로젝트 보고서

[네이버 날씨에 따른 수치와 멘트 출력]

1. 크롤링을 통한 날씨 정보 가져오기

- BeautifulSoup을 사용하여 네이버에 검색한 결과를 가져오는 코드를 생성합니다. 저는 '(검색어) + 날씨', '(검색어) + 미세먼지' 이렇게 검색하여 해당 검색 페이지에 뜨는 정보를 가져올 수 있도록 하였습니다. Ex> 마포구 날씨, 부산 날씨 .etc
- 가져온 정보들은 각각 변수에 저장해 주었습니다.

현재 온도의 경우 nowTemp, 오늘 최저온도는 tdyL, 오늘 최고온도는 tdyH 주간 최저 온도는 weekL, 주간 최고 온도는 weekH, 오늘 오전 미세먼지는 tdfn1, 오늘 오후 미세먼지는 tdfn2입니다.

- 크롤링을 처음 해보다 보니 기존 proposal에서 수정하는 부분이 생겼습니다. 크롤링을 통해 온도, 미세먼지의 정보는 가져오기 수월하나 강수량의 정보는 가져오기 다소 힘든 부분이 있어 강수확률에 따른 멘트 출력 부분을 제외하고 프로젝트를 진행하게 되었습니다.

2. class Temp

- 현재 온도와 오늘 최고, 최저 온도 정보를 처리하기 위한 클래스입니다.

정규식을 이용하여 크롤링해온 정보들 중에서 온도를 의미하는 숫자부분만 찾아 돌려주도록 하였습니다.

3. class TMent

- 온도에 따른 멘트를 출력하는 클래스입니다.
- Temp 클래스를 상속받고 low, high 변수를 추가합니다.
- tdyment 함수로 현재 온도가 25도 이상이면 '오늘은 더워요!', 15도 이하이면 '오늘은 추워요!', 그 사이의 기온에서는 '오늘은 날이 선선해요!'를 출력합니다.
- temp_gap 함수로 오늘 하루 최고 기온과 최저 기온의 차이를 계산했을 때 10도 이상 높아진다면 '일교차가 큰 날이에요! 겉옷을 챙겨주세요!'를 출력하고 그 외의 경우에는 '오늘은 일교차가 크지 않아요!'를 출력합니다.

>> 최종적으로 현재온도, 오늘하루 최저기온, 오늘 하루 최고기온을 출력해주고 함수도 실행시킵니다.

4. class Finedust

- 오늘 오전과 오후 미세먼지 상태를 처리하는 클래스입니다.

정규식을 이용하여 크롤링해온 정보 중 문자에 해당하는 부분만 찾아 돌려주도록 합니다.

5. class FMent

- 미세먼지 상태에 따른 멘트를 출력하는 클래스입니다.
- Finedust 클래스를 상속받고 am, pm 변수를 추가해줍니다.
- tdfine_ment 함수를 통하여 오전과 오후 중 매우나쁨이 하나라도 있다면 '외출을 자제해주세요!', 나쁨이 하나라도 있다면 '마스크를 꼭 착용해주세요!', 한때 나쁨이 하나라도 있다면 '마스크를 지참해주세요!', 보통이 하나라도 있다면 '공기가 쾌적한 날이에요!', 그 외에는 '신선한 공기를 맘껏 즐기세요!'를 출력합니다.

-

>> 최종적으로 오늘 오전, 오후 미세먼지 상태를 출력하고 함수를 실행합니다.

6. 주간 온도 그래프

- 맨 처음 가져온 주간 최고, 최저 온도 정보에서 정규식을 통하여 수치만 가져와줍니다. 이를 각각 리스트에 저장합니다.
- Datetime을 통해 오늘날짜로부터 10일 날짜를 생성합니다.
- weeklytemp함수를 통해 날짜를 x값, 온도를 y값으로 하여 최고 최저 온도 그래프를 생성합니다. 최고온도는 빨강, 최저온도는 파랑으로 표시하였습니다.
- 추가적으로 조금 귀엽게 그래프를 표현하고 싶어 xkcd형식을 사용하였습니다.

7. checkTime

- 함수 실행시간을 구해주는 checkTime 함수를 정의해주고 데코레이터를 사용하여 그래프를 생성하는 시간을 구해주고 출력합니다.

8. 예외처리

- Input을 지역명이 아닌 다른 검색어를 넣었을 경우 발생하는 AttributeError의 경우 '지역명을 검색해 주세요'를 출력하도록 합니다.
- 이외 예외상황이 나타난다면 '에러가 발생하였습니다. 다시 시도해주세요.'를 출력하도록 합니다.

이에 따른 출력화면은 다음과 같습니다.

In [1]: %%writefile weathercheck.py

```
import requests
from bs4 import BeautifulSoup
import re
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from datetime import datetime

def checkTime(func):
    def Func(*args, **kwargs):
        start = datetime.now()
        func(*args, **kwargs)
        end = datetime.now()
        print('=====')
        print("실행시간:", (end-start).total_seconds(), 'sec')
    return Func

try:
    location = input("지역을 입력하세요>>> ")
    findweather = location + '날씨'
    findfine = location + '미세먼지'

    url = 'https://search.naver.com/search.naver?sm=tab_hy_top&where=nexearch&query=' + findweather
    url2 = 'https://search.naver.com/search.naver?sm=tab_hy_top&where=nexearch&query=' + findfine
    req = requests.get(url)
    req2 = requests.get(url2)
    html = req.text
    html2 = req2.text
    soup = BeautifulSoup(html, 'html.parser')
    soup2 = BeautifulSoup(html2, 'html.parser')
```

```
#현재 온도
nowTemp = soup.find('div', {'class': 'temperature_text'}).text

# 오늘 최저, 최고 온도
tdyL = soup.find('span', {'class': 'lowest'}).text
tdyH = soup.find('span', {'class': 'highest'}).text

# 주간 온도 -> 그래프(matplotlib)
weekL = soup.find_all('span', {'class': 'lowest'})
weekH = soup.find_all('span', {'class': 'highest'})

# 미세먼지 (오전/오후, 오늘)
finet = soup2.find_all('div')
finet = []
for t in finet:
    finet.append(t.text)
tdfn1 = finet[0]
tdfn2 = finet[1]

# 현재 온도 출력(그냥 출력은 정규식 사용해도 될듯?)
# 오늘 최저, 최고 온도 출력

# 그에 따른 멘트 출력 (< 이면 무조건 00P)

# 일교차가 크다 최고 - 최저 >= 10
# 더운 날씨이다 25도 이상
# 추운 날씨이다 15도 이하
# 선선한 날씨이다 15-24

## 신선한 공기/쾌적한 날/마스크 지참/마스크 착용 필요/ 외출자제
```

```

class Temp:
    def __init__(self):
        self.temp = temp
    def now_Temp(self):
        t1 = re.compile('#d+.#d+')
        nowtemp = t1.findall(self)
        return nowtemp
    def tdy_Low(self):
        t2 = re.compile('#d+')
        tdlow = t2.findall(self)
        return tdlow
    def tdy_High(self):
        t2 = re.compile('#d+')
        tdhigh = t2.findall(self)
        return tdhigh

class TMent(Temp):
    def __init__(self, low, high):
        Temp.__init__(self)
        self.low = low
        self.high = high
    def tdy_ment(self):
        if pd.Series(list(map(float, Temp.now_Temp(self))))[0] >= 25:
            print('오늘은 더워요!')
        elif pd.Series(list(map(float, Temp.now_Temp(self))))[0] <= 15:
            print('오늘은 추워요!')
        else:
            print('오늘은 날이 선선해요!')
    def temp_gap(low, high):
        int_L = pd.Series(list(map(int, Temp.tdy_Low(low))))
        int_H = pd.Series(list(map(int, Temp.tdy_High(high))))
        result = int_H.sub(int_L)
        if result[0] >= 10:
            print('일교차가 큰 날이에요! 겉옷을 챙겨주세요!')
        else:
            print('오늘은 일교차가 크지 않아요!')

```

```

print('=====오늘 날씨=====')
print(f'현재온도: {"".join(Temp.now_Temp(nowTemp))}℃')
print(f'오늘 하루 최저기온: {"".join(Temp.tdy_Low(tdyL))}℃')
print(f'오늘 하루 최고기온: {"".join(Temp.tdy_High(tdyH))}℃')
TMent.temp_gap(tdyL, tdyH)
TMent.tdy_ment(nowTemp)

class Finedust:
    def __init__(self):
        self.fine = fine
    def tdy_am(self):
        t3 = re.compile('#w+')
        tdam = t3.findall(self)
        return tdam
    def tdy_pm(self):
        t3 = re.compile('#w+')
        tdpn = t3.findall(self)
        return tdpn

class FMent(Finedust):
    def __init__(self, am, pm):
        Finedust.__init__(self)
        self.am = am
        self.pm = pm
    def tdfine_ment(am, pm):
        if Finedust.tdy_am(am)[1] == '매우나쁨' or Finedust.tdy_am(pm)[1] == '매우나쁨':
            print('외출을 자제해주세요!')
        elif Finedust.tdy_am(am)[1] == '나쁨' or Finedust.tdy_am(pm)[1] == '나쁨':
            print('마스크를 꼭 착용해주세요!')
        elif Finedust.tdy_am(am)[1] == '한때나쁨' or Finedust.tdy_am(pm)[1] == '한때나쁨':
            print('마스크를 지참해주세요!')
        elif Finedust.tdy_am(am)[1] == '보통' or Finedust.tdy_am(pm)[1] == '보통':
            print('공기가 쾌적한 날이에요!')
        else:
            print('신선한 공기를 맘껏 즐기세요!')

```

```
print('=====오늘 미세먼지=====')
print(f'오늘 {Finedust.tdy_am(tdfn1)[0]} 미세먼지 수치:{Finedust.tdy_am(tdfn1)[1]}')
print(f'오늘 {Finedust.tdy_pm(tdfn2)[0]} 미세먼지 수치:{Finedust.tdy_pm(tdfn2)[1]}')
Fment.tdfine_ment(tdfn1, tdfn2)
```

주간 온도 그래프

```
@checkTime
def weeklytemp():
    tl = []
    for t in weekL:
        tl.append(t.text)
    th = []
    for t in weekH:
        th.append(t.text)

    import re
    tempL = []
    for i in tl:
        t = re.compile('#d+')
        m = t.findall(i)
        tempL.extend(m)
    result1 = list(map(int, tempL))

    tempH = []
    for i in th:
        t = re.compile('#d+')
        m = t.findall(i)
        tempH.extend(m)
    result2 = list(map(int, tempH))

    import matplotlib.pyplot as plt
    from datetime import datetime
    import numpy as np

    td = datetime.today().day
```

```
with plt.xkcd():
    plt.plot(np.arange(td, td+10), result1, 'bx-', label = 'low')
    plt.plot(np.arange(td, td+10), result2, 'rx-', label = 'high')
    plt.xlabel('Date')
    plt.ylabel('Temperature')
    plt.title('Weekly temperature')
    plt.legend()
plt.show
weeklytemp()

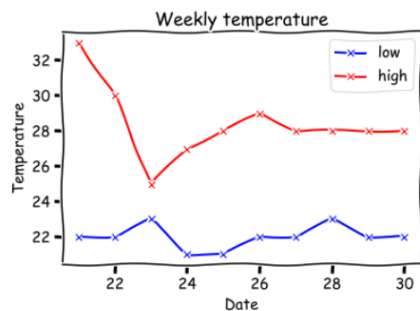
except AttributeError:
    print('지역명을 검색해 주세요.')
except:
    print('에러가 발생하였습니다. 다시 시도해주세요.')
```

Overwriting weathercheck.py

In [2]: %run weathercheck.py

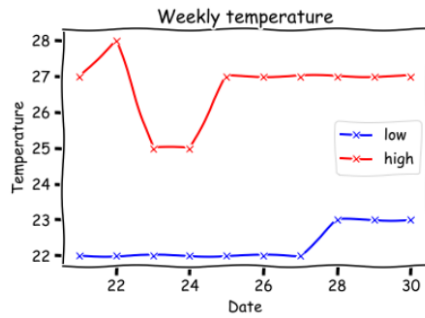
```
지역을 입력하세요
>>> 마포구
=====오늘 날씨=====
현재온도: 29.1℃
오늘 하루 최저기온: 22℃
오늘 하루 최고기온: 33℃
일교차가 큰 날이에요! 겉옷을 챙겨주세요!
오늘은 더워요!
=====오늘 미세먼지=====
오늘 오전 미세먼지 수치: 보통
오늘 오후 미세먼지 수치: 보통
공기가 쾌적한 날이에요!
=====
실행시간: 0.102067 sec
```

실행시간: 0.102067 sec



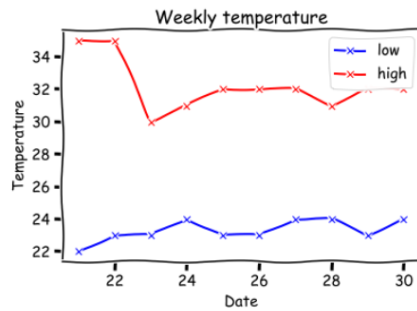
In [3]: %run weathercheck.py

```
지역을 입력하세요
>>> 부산
=====오늘 날씨=====
현재온도: 26.4℃
오늘 하루 최저기온: 22℃
오늘 하루 최고기온: 27℃
오늘은 일교차가 크지 않아요!
오늘은 더워요!
=====오늘 미세먼지=====
오늘 오전 미세먼지 수치: 보통
오늘 오후 미세먼지 수치: 보통
공기가 쾌적한 날이에요!
=====
실행시간: 0.084658 sec
```



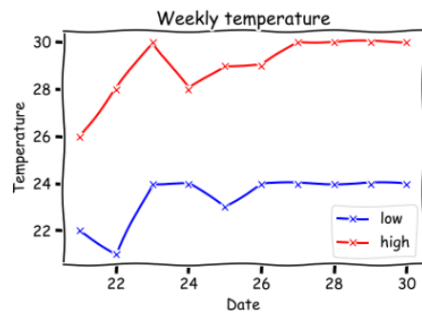
In [4]: %run weathercheck.py

```
지역을 입력하세요
>>> 대구
=====오늘 날씨=====
현재온도: 32.7℃
오늘 하루 최저기온: 22℃
오늘 하루 최고기온: 35℃
일교차가 큰 날이에요! 겉옷을 챙겨주세요!
오늘은 더워요!
=====오늘 미세먼지=====
오늘 오전 미세먼지 수치: 보통
오늘 오후 미세먼지 수치: 보통
공기가 쾌적한 날이에요!
=====
실행시간: 0.072932 sec
```



In [5]: %run weathercheck.py

```
지역을 입력하세요
>>> 제주도
=====오늘 날씨=====
현재 온도: 24.6℃
오늘 하루 최저기온: 22℃
오늘 하루 최고기온: 26℃
오늘은 일교차가 크지 않아요!
오늘은 날이 선선해요!
=====오늘 미세먼지=====
오늘 오전 미세먼지 수치: 보통
오늘 오후 미세먼지 수치: 보통
공기가 쾌적한 날이에요!
=====
실행시간: 0.051828 sec
```



한계점: 네이버에 검색한 결과를 가져오는 것이기 때문에 서울, 부산, 제주도부터 마포구, 송파구 등의 지역명까지 검색이 가능하지만 도곡동, 대치동 등을 검색하였을 때는 검색화면에 차이가 생겨 미세먼지 상태를 잘못 가져오는 현상이 일어난다는 한계점이 있습니다.

감사합니다.