

Import 모듈
*설치 시 pip모듈 필요.
requests모듈 →pip3 install requests
telegram모듈 →pip3 install python-telegram-bot==13.15 *항상 13.15버전으로 해야 함. 그 이상 버전은 async사용 필요.

Openweather API
API_key: ca06c1e43db582bd454d1ba26cdb9d7f

Telegram-bot API
Access_token: 6355974422:AAFxsdBZNmDXU3ZW5LqY8TyXfEu5MFTGZls Chat_id: 6228512197

Python programming 변수(모두 소문자, word 가 자동으로 수정함)	
변수 이름	변수 용도
Now	현재의 시간 가져오기.
Current	현재의 시간을 저장하기.
Access_token	Telegram bot 에 액세스가 가능한 토큰.
Chat_id	Telegram bot 의 chatid.
Apikey	Openweather API 를 호출할 수 있는 API.
City	도시명.
Lang	국가명.
Lat	위도.
Lon	경도.
Full_Url	Openweather API(Current weather data)를 가져오는 Url.
Full1_Url	Openweather API(Air pollution data)를 가져오는 Url.
Response_org	Full_Url 의 응답을 받아옴.
Response1_org	Full1_Url 의 응답을 받아옴.
Temperature	Openweather API(current weather data)에서 가져온 현재 기온.
Temperature_min	Openweather API(current weather data)에서 가져온 오늘 최저 기온.
Temperature_max	Openweather API(current weather data)에서 가져온 오늘 최고 기온.
Temperature_feels_like	Openweather API(current weather data)에서 가져온 체감 온도.
Humidity	Openweather API(current weather data)에서 가져온 현재 습도.
Weather	Openweather API(current weather data)에서 가져온 현재 날씨.
Pressure	Openweather API(current weather data)에서 가져온 현재 기압.
Time	Openweather API(current weather data)에서 가져온 현재 시각.
Wind_speed	Openweather API(current weather data)에서 가져온 현재 풍속.
Wind_deg	Openweather API(current weather data)에서 가져온 현재 풍향.
Visibility	Openweather API(current weather data)에서 가져온 현재 가시거리.

Clouds	Openweather API(current weather data)에서 가져온 현재 구름의 비율.
Co	Openweather API(air pollution data)에서 가져온 co 농도.
No	Openweather API(air pollution data)에서 가져온 no 농도.
No2	Openweather API(air pollution data)에서 가져온 no2 농도.
O3	Openweather API(air pollution data)에서 가져온 o3 농도.
So2	Openweather API(air pollution data)에서 가져온 so2 농도.
Pm2_5	Openweather API(air pollution data)에서 가져온 pm2.5 농도.
Pm10	Openweather API(air pollution data)에서 가져온 pm10 농도.
Humidity_pleasant	Openweather API(current weather data)에서 가져온 기온, 습도 등을 이용한 불쾌지수.

<programming code>

코딩 class	코딩 code,결과, 설명
1.모듈 import	<p><코딩 code></p> <pre>import requests import telegram import datetime</pre> <p><결과></p> <p>-</p> <p><설명></p> <p>python 코딩에 필요한 코드를 불러온다.</p>
2.현재 시각 가져오기	<p><코딩 code></p> <pre>now = datetime.datetime.now() current = datetime.datetime(now.year, now.month, now.day, now.hour, now.minute, now.second)</pre> <p><결과></p> <p>-</p> <p><설명></p> <p>현재의 시각을 가져온다.</p>
3.엑세스 토큰,api 키, 위치 정보 등록	<p><코딩 code></p> <pre>access_token = '자신의 telegram bot 접근 토큰을 넣어 주세요' #telegram apikey = "자신의 openweatherAPI 키를 넣어주세요" #openweather city = "Seoul" #자신이 원하는 지역 코드를 넣어주세요(ex.seoul.daegu) lang = "kr" #자신이 원하는 국가 코드를 넣어주세요.(ex.kr) lat = "37.5642135" #자신이 원하는 지역의 위도를 입력하세요. lon = "127.0016985" #자신이 원하는 지역의 경도를 입력하세요.</pre> <p><결과></p> <p>-</p> <p><설명></p> <p>코딩에 필요한 엑세스 토큰,api 키를 등록하고 위치 정보를 등록한다.</p>
4.api 정보를 받아올 Url 과 그에 따른 결과를 출력한다.	<p><코딩 code></p> <pre>full_url = f"http://api.openweathermap.org/data/2.5/forecast?id=524901&q={city} &appid={apikey}&lang={lang}&units=metric"# 분당 60번만 가능, 하루 간격 16일치 날씨 정보 조회 full_url = f"http://api.openweathermap.org/data/2.5/air_pollution?lat={lat}&lo</pre>

	<pre> n={lon}&appid={apikey}" response_org = requests.get(fullUrl).json() response1_org = requests.get(fullUrl).json() print(response_org['list']) print("##### ###WnWn") print(response_org['list'][0]) print("##### ###WnWn") <결과> {'dt': 1695740400, 'main': {'temp': 19.61, 'feels_like': 20.19, 'temp_min': 18.92, 'temp_max': 19.61, 'pressure': 1012, 'sea_level': 1012, 'grnd_level': 1006, 'humidity': 98, 'temp_kf': 0.69}, 'weather': [{'id': 804, 'main': 'Clouds', 'description': '온흐림', 'icon': '04n'}], 'clouds': {'all': 100}, 'wind': {'speed': 1.56, 'deg': 70, 'gust': 2.73}, 'visibility': 10000, 'pop': 0.08, 'sys': {'pod': 'n'}, 'dt_txt': '2023-09-26 15:00:00'} <설명> 코딩에 필요한 액세스 토큰,api 키를 등록하고 위치 정보를 등록한다. </pre>
<p>5.날씨 데이터를 저장할 변수를 지정한다.</p>	<pre> <코딩 code> temperature = response_org['list'][0]['main']['temp'] temperature_min = response_org['list'][0]['main']['temp_min'] temperature_max = response_org['list'][0]['main']['temp_max'] temperature_feels_like = response_org['list'][0]['main']['feels_like'] humidity = response_org['list'][0]['main']['humidity'] weather = response_org['list'][0]['weather'][0]['description'] pressure = response_org['list'][0]['main']['pressure'] time = response_org['list'][0]['dt_txt'] wind_speed = response_org['list'][0]['wind']['speed'] wind_deg = response_org['list'][0]['wind']['deg'] visibility = response_org['list'][0]['visibility'] clouds = response_org['list'][0]['clouds']['all'] co = response1_org['list'][0]['components']['co'] no2 = response1_org['list'][0]['components']['no2'] o3 = response1_org['list'][0]['components']['o3'] so2 = response1_org['list'][0]['components']['so2'] pm2_5 = response1_org['list'][0]['components']['pm2_5'] pm10 = response1_org['list'][0]['components']['pm10'] <결과> - <설명> 날씨 정보를 등록한다. </pre>
<p>6.대기질 계산 조건문</p>	<pre> <코딩 code> if 0<=co<4400: result_co = "매우 좋음" elif 4400<=co<9400: result_co = "좋음" elif 9400<=co<12400: result_co = "보통" elif 12400<=co<15400: result_co = "나쁨" elif co>=15400: result_co = "매우 나쁨" if 0<=no2<40: </pre>

	<pre>result_no2 = "매우 좋음" elif 40<=no2<70: result_no2 = "좋음" elif 70<=no2<150: result_no2 = "보통" elif 150<=no2<200: result_no2 = "나쁨" elif no2>=200: result_no2 = "매우 나쁨" if 0<=o3<60: result_o3 = "매우 좋음" elif 60<=o3<100: result_o3 = "좋음" elif 100<=o3<140: result_o3 = "보통" elif 140<=o3<180: result_o3 = "나쁨" elif o3>=180: result_o3 = "매우 나쁨" if 0<=so2<20: result_so2 = "매우 좋음" elif 20<=so2<80: result_so2 = "좋음" elif 80<=so2<250: result_so2 = "보통" elif 250<=so2<350: result_so2 = "나쁨" elif so2>=350: result_so2 = "매우 나쁨" if 0<=pm2_5<10: result_pm2_5 = "매우 좋음" elif 10<=pm2_5<25: result_pm2_5 = "좋음" elif 25<=pm2_5<50: result_pm2_5 = "보통" elif 50<=pm2_5<75: result_pm2_5 = "나쁨" elif 75<=pm2_5: result_pm2_5 = "매우 나쁨" if 0<=pm10<20: result_pm10 = "매우 좋음" elif 20<=pm10<50: result_pm10 = "좋음" elif 50<=pm10<100: result_pm10 = "보통" elif 100<=pm10<200: result_pm10 = "나쁨" elif pm10>=200: result_pm10 = "매우 나쁨" <결과> - <설명> 저장한 대기질 data 를 바탕으로 조건에 따라서 출력할 문구를 정한다.</pre>
7.불쾌지수 계산 조건문	<코딩 code>

	<pre> humidity_pleasant = (0.81*temperature)+(0.01*humidity)*(0.99*temperature-14.3)+46.3 if humidity_pleasant<=68: result = "전원이 쾌적함을 느낍니다." elif humidity_pleasant<=70: result = "전원 중 소수가 불쾌감을 느낍니다." elif humidity_pleasant<=75: result = "전원 중 10%가 불쾌감을 느낍니다." elif humidity_pleasant<=80: result = "전원 중 50%가 불쾌감을 느낍니다." elif humidity_pleasant<=83: result = "전원이 불쾌감을 느낍니다." else: result = "전원이 매우 불쾌감을 느낍니다" <결과> - <설명> 가져온 날씨 data 를 사용하여 불쾌지수를 계산한다. 또한, 그에 따른 문구를 조건문을 사용하여 제어한다. </pre>
<p>8.telegram 세팅 및 보내기[(cmd)창에도 띄우기])</p>	<pre> print("-----기본 기상 정보-----") print(response_org['list'][0]) print("온도: ", temperature) print("최저 온도: ", temperature_min) print("최고 온도: ", temperature_max) print("체감 온도: ", temperature_feels_like) print("습도", humidity) print("날씨: ", response_org['list'][0]['weather'][0]['description']) print("기압: ", pressure) print("시간: ", time) print("풍속: ", wind_speed) print("풍향: ", wind_deg) print("가시거리: ", visibility) print("구름: ", clouds) print("-----대기질 오염 정보-----") print("co(일산화탄소) :",co,"ppm", "으로 현재 co농도는",str(result_co),"입니다.") print("no2(이산화질소) :",no2,"ppm", "으로 현재 no2농도는",str(result_no2),"입니다.") print("o3(오존) :",o3,"ppm", "으로 현재 o3농도는",str(result_o3),"입니다.") print("so2(이산화 황) :",so2,"ppm", "으로 현재 so2농도는",str(result_so2),"입니다.") print("pm2.5(초미세먼지) : ",pm2_5,"μg/m³ ", "으로 현재 pm2.5농도는",str(result_pm2_5),"입니다.") print("pm10(미세먼지) :",pm10,"μg/m³ ", "으로 현재 pm10농도는",str(result_pm10),"입니다.") print("-----불쾌지수 정보-----") print("불쾌지수",humidity_pleasant) print("불쾌를 느끼는 사람의 비율:"+str(result)) print("-----") -WnWn") bot = telegram.Bot(token=access_token) chat_id = 자신의 chatid를 넣어주세요. bot.sendMessage(chat_id=chat_id, text="현재("+str(current)+"),오늘 날씨는 다음과 같습니다.Wn W </pre>

```

Wn-----기본 기상 정보-----
----"

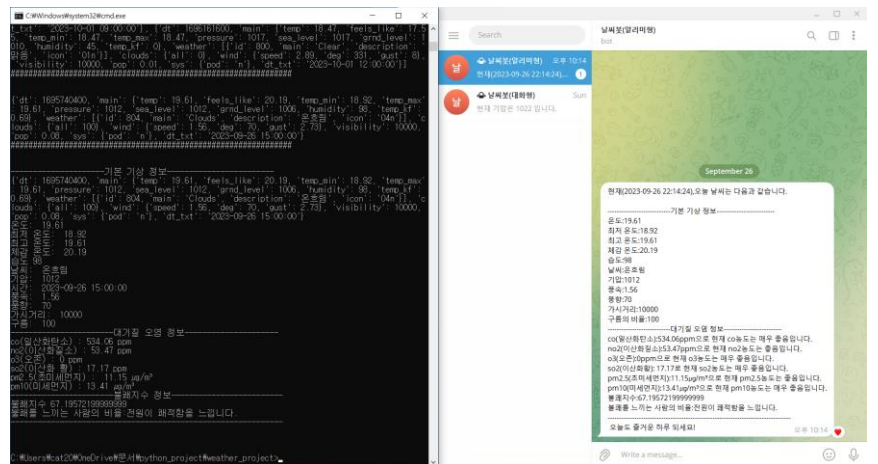
"Wn온도:" +str(temperature)+ W
"Wn최저 온도:" +str(temperature_min)+ W
"Wn최고 온도:" +str(temperature_max)+ W
"Wn체감 온도:" +str(temperature_feels_like)+ W
"Wn습도:" +str(humidity)+ W
"Wn날씨:" +str(weather)+ W
"Wn기압:" +str(pressure)+ W
"Wn풍속:" +str(wind_speed)+ W
"Wn풍향:" +str(wind_deg)+ W
"Wn가시거리:" +str(visibility)+ W
"Wn구름의 비율:" +str(clouds)+ W
"Wn-----대기질 오염 정보-----
-----"

"Wnco(일산화탄소):" +str(co)+"ppm으로 현재 co농도는
"+str(result_co)+"입니다." + W
"Wnno2(이산화질소):" +str(no2)+"ppm으로 현재 no2농도는
"+str(result_no2)+"입니다." W
"Wno3(오존):" +str(o3)+"ppm으로 현재 o3농도는
"+str(result_o3)+"입니다." W
"Wnso2(이산화황):" +str(so2)+"로 현재 so2농도는
"+str(result_so2)+"입니다." W
"Wnpm2.5(초미세먼지):" +str(pm2_5)+" $\mu\text{g}/\text{m}^3$  으로 현재 pm2.5농도는
"+str(result_pm2_5)+"입니다." W
"Wnpm10(미세먼지):" +str(pm10)+" $\mu\text{g}/\text{m}^3$  으로 현재 pm10농도는
"+str(result_pm10)+"입니다." W
"Wn불쾌지수:" +str(humidity_pleasant)+ W
"Wn불쾌를 느끼는 사람의 비율:" +str(result)+ W
"Wn-----
-----"

"Wn 오늘도 즐거운 하루 되세요!")

```

<결과>



<설명>

지금까지의 조건문에 따른 문장 및 날씨 데이터를 사용하고 json 등의 데이터를 가공하여 우리에게 날씨 정보를 보낸다.(최종)