

공공데이터 API를 통한 실시간 날씨 관측

Content



Introduction

서론_배경 및 목적



Main Body

본론_코드 구현



Conclusion

결론_결과 분석



01

서론

I Introduction

배경 및 목적

일상생활에서 날씨는 우리와 직접적으로 연관되어 있습니다. 날씨에 영향을 받는 직업군은 특히 날씨를 늘 주시하고 대비해야 합니다. 농업, 어업, 건설, 운송 등 날씨에 직접적으로 영향을 받는 산업 분야에서는 정확한 날씨 정보가 중요한 역할을 합니다. 예를 들어, 농작물 재배 일정을 결정할 때는 강우량과 기온 등 날씨 조건을 고려해야 합니다. 또한, 항공, 해상 운송에서는 안전한 운항을 위해 기상 예보를 신속하게 파악해야 합니다. 일반적인 직장인이나 학생들도 날씨에 따라 옷차림이나 우산 사용 등을 결정하게 됩니다. 더운 여름에는 가벼운 옷을 입고, 추운 겨울에는 따뜻한 옷을 선택합니다. 비가 오거나 눈이 내릴 것으로 예상될 때는 우산을 챙기고, 맑은 날씨라면 우산을 가져다 놓지 않습니다.

따라서, 실시간 날씨 정보를 제공하여 사람들이 적절한 조치를 취하고 편안하게 일상생활을 영위할 수 있도록 하는 것이 중요합니다. 이를 위해 이번 프로젝트는 공공데이터를 활용하여 API를 통해 실시간으로 날씨 정보를 사용자에게 제공하는 목적을 가지고 있습니다. 이를 통해 사람들은 안전하고 편리한 환경에서 생활할 수 있으며, 산업 분야에서도 정확한 날씨 정보를 활용하여 효율성을 높일 수 있습니다. 또한, 기상 예측과 연구에 활용되어 기후 변화에 대한 인식을 높이고 지속 가능한 미래를 위한 행동을 촉진할 수 있습니다.

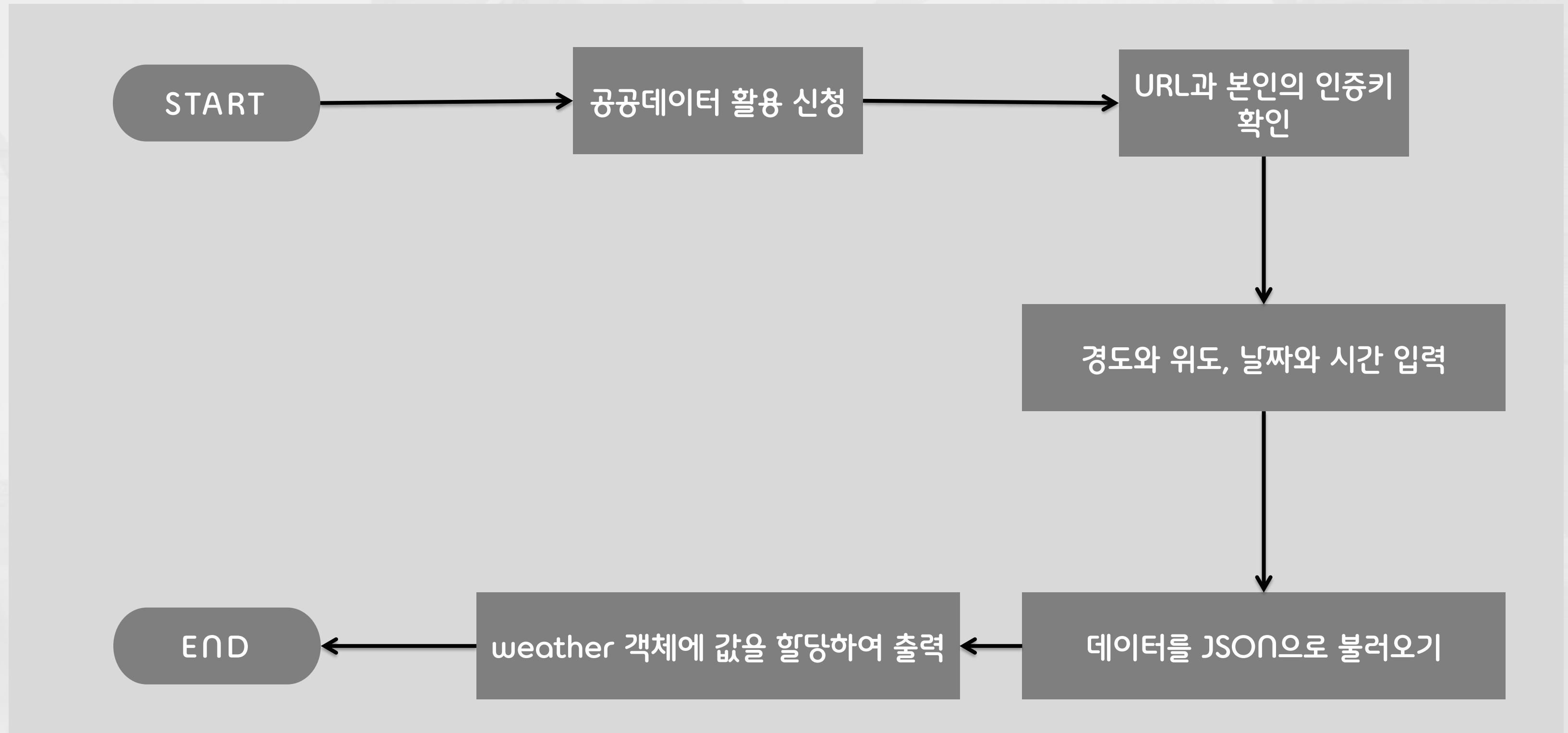


main Body

02

부 부

실행 과정



코드 구현

```
1 import requests
2 import json
3
4 class Weather:
5     def __init__(self):
6         self.locationCode = 0
7         self.date = ""
8         self.time = ""
9         self.PTY = 0.0
10        self.REH = 0.0
11        self.RN1 = 0.0
12        self.T1H = 0.0
13        self.UUU = 0.0
14        self.VVV = 0.0
15        self.VEC = 0.0
16        self.WSD = 0.0
17
18    def __str__(self):
19        return (
20            "locationCode = "
21            + str(self.locationCode)
22            + "\n" + "date = "
23            + self.date
24            + "\n" + "time = "
25            + self.time
26            + "\n" + "PTY = "
27            + str(self.PTY)
28            + "\n" + "REH = "
29            + str(self.REH)
30            + "\n" + "RN1 = "
31            + str(self.RN1)
32            + "\n" + "T1H = "
33            + str(self.T1H)
34            + "\n" + "UUU = "
35            + str(self.UUU)
36            + "\n" + "VEC = "
37            + str(self.VEC)
38            + "\n" + "VVV = "
39            + str(self.VVV)
40            + "\n" + "WSD = "
41            + str(self.WSD)
42        )
```

requests 모듈은 지정된 URL로 HTTP 요청을 보내고 응답을 받는 기능을 제공합니다.

json 모듈은 JSON 데이터를 처리하기 위한 함수를 제공합니다.

Weather 클래스는 날씨 정보를 담기 위한 클래스로, 다양한 날씨 요소들을 멤버 변수로 가지고 있습니다. __init__ 메서드는 객체가 생성될 때 호출되며, 각 변수들을 초기화합니다.

__str__ 메서드는 객체를 문자열로 표현하기 위한 특수한 메서드입니다. 이 메서드는 print() 함수나 str() 함수 등에서 객체를 문자열로 변환할 때 호출됩니다. __str__ 메서드에서는 객체의 각 변수들을 문자열로 변환하여 한 줄로 이어 붙인 결과를 반환합니다.

초단기실황	T1H	기온	°C
	RN1	1시간 강수량	mm
	UUU	동서바람성분	m/s
	VVV	남북바람성분	m/s
	REH	습도	%
	PTY	강수형태	코드값
	VEC	풍향	deg
	WSD	풍속	m/s

초단기실황의 정보는 좌측과 같으며, 사용하고자 하는 정보는 기온과 1시간 강수량, 습도입니다.

정보 필요의 유무에 따라 수정하기 위하여 코드에서는 모든 정보를 불러왔습니다.

코드 구현

```
1 def main():
2     print("날씨의 실시간 데이터 관측이다 보니 날짜는 최근 1일전까지만 가능합니다.")
3     weather = Weather()
4     apiUrl = "http://apis.data.go.kr/1880000/VillageFcstInfoService_2.0/getUltra8rtNcst"
5     authKey = "1cAPXQPIUUuDvgQ8WdJYPPQY29W1Y1Q80H7JasOzLahOPUI81tQsE9gYn4sL%2F%28FWPMWkeIqkALoDh21BnYBLuw%3D%3D"
6     nx = Input("경도를 입력해주세요 : ")
7     ny = Input("위도를 입력해주세요 : ")
8     baseDate = Input("날짜를 입력해주세요(ex.20230708) : ")
9     baseTime = Input("시간을 입력해주세요(ex.0800) : ")
10    dataType = "JSON"
11    urlBuilder = apiUrl + "?" + "serviceKey=" + authKey
12    urlBuilder += "&" + "numOfRows=10" # 숫자 10
13    urlBuilder += "&" + "pageNo=1" # 페이지 1
14    urlBuilder += "&" + "dataType=" + dataType # 받으려는 타입
15    urlBuilder += "&" + "base_date=" + baseDate # 조회하고싶은 날짜
16    urlBuilder += "&" + "base_time=" + baseTime # 조회하고싶은 시간 AM 02시부터 8시간 범위
17    urlBuilder += "&" + "nx=" + nx # 경도
18    urlBuilder += "&" + "ny=" + ny # 위도
19    response = requests.get(urlBuilder)
20    result = response.text
21    jsonData = json.loads(result)
22    parse_response = jsonData["response"]
23    parse_body = parse_response["body"]
24    parse_items = parse_body["items"]
25    parse_item = parse_items["item"]
26    print(" ")
27    print("-----")
28
29    for item in parse_item:
30        category = item["category"]
31        value = float(item["obsrValue"])
32
33        if category == "PTY":
34            weather.PTY = value
35        elif category == "REH":
36            weather.REH = value
37        elif category == "RN1":
38            weather.RN1 = value
39        elif category == "T1H":
40            weather.T1H = value
41        elif category == "UUU":
42            weather.UUU = value
43        elif category == "VEC":
44            weather.VEC = value
45        elif category == "VWV":
46            weather.VWV = value
47        elif category == "WSD":
48            weather.WSD = value
49
50    weather.date = baseDate
51    weather.time = baseTime
52
53    print(weather.date, weather.time, "의 날씨는 기온", weather.T1H, "도 이며,")
54    print("시간 강수량은", weather.RN1, "mm 이고,")
55    print("습도는", weather.REH, "% 입니다.")
56    print("-----")
57
58 if __name__ == "__main__":
59     main()
```

사용자로부터 경도(nx), 위도(ny), 날짜(baseDate), 시간(baseTime)을 입력 받습니다. API URL과 인증키, 입력 받은 값들을 기반으로 요청 URL을 생성합니다.

requests 모듈을 사용하여 생성한 URL에 GET 요청을 보내고, 응답을 받아옵니다. 받아온 응답은 JSON 형식으로 제공되므로, json 모듈을 사용하여 JSON 데이터를 파싱합니다. 파싱한 데이터에서 필요한 날씨 정보를 추출하여 Weather 객체의 변수에 할당합니다. Weather 객체의 date, time, T1H, RN1, REH 변수를 출력하여 날씨 정보를 보여줍니다.

이렇게 작성된 코드는 사용자로부터 입력 받은 위치와 시간에 해당하는 실시간 날씨 정보를 API를 통해 가져와 출력합니다. 사용자는 경도, 위도, 날짜, 시간을 입력하여 원하는 지역과 시점에 대한 날씨 정보를 확인할 수 있습니다.



Conclusion

03

결론

결론

날씨의 실시간 데이터 관측이다 보니 날짜는 최근 1일전까지만 가능합니다.
경도를 입력해주세요 : 55
위도를 입력해주세요 : 127
날짜를 입력해주세요(ex.20230708) : 20230709
시간을 입력해주세요(ex.0800) : 0300

20230709 0300 의 날씨는 기온 24.6 도이며,
1시간 강수량은 0.0 mm 이고,
습도는 90.0 % 입니다.

좌측과 같이 입력을 하면, 해당 날짜와 시간의
기온과 강수량, 습도가 나옵니다.

공공데이터에서 제공하는 데이터시트에서
확인하고자 하는 위치의 경도와 위도를 직접
찾아서 입력해야 한다는 불편함이 있습니다.
그러나 실시간으로 사이트를 들어가지 않고도
코드를 통해 확인할 수 있다는 점이 편리합니다.

THANK YOU

