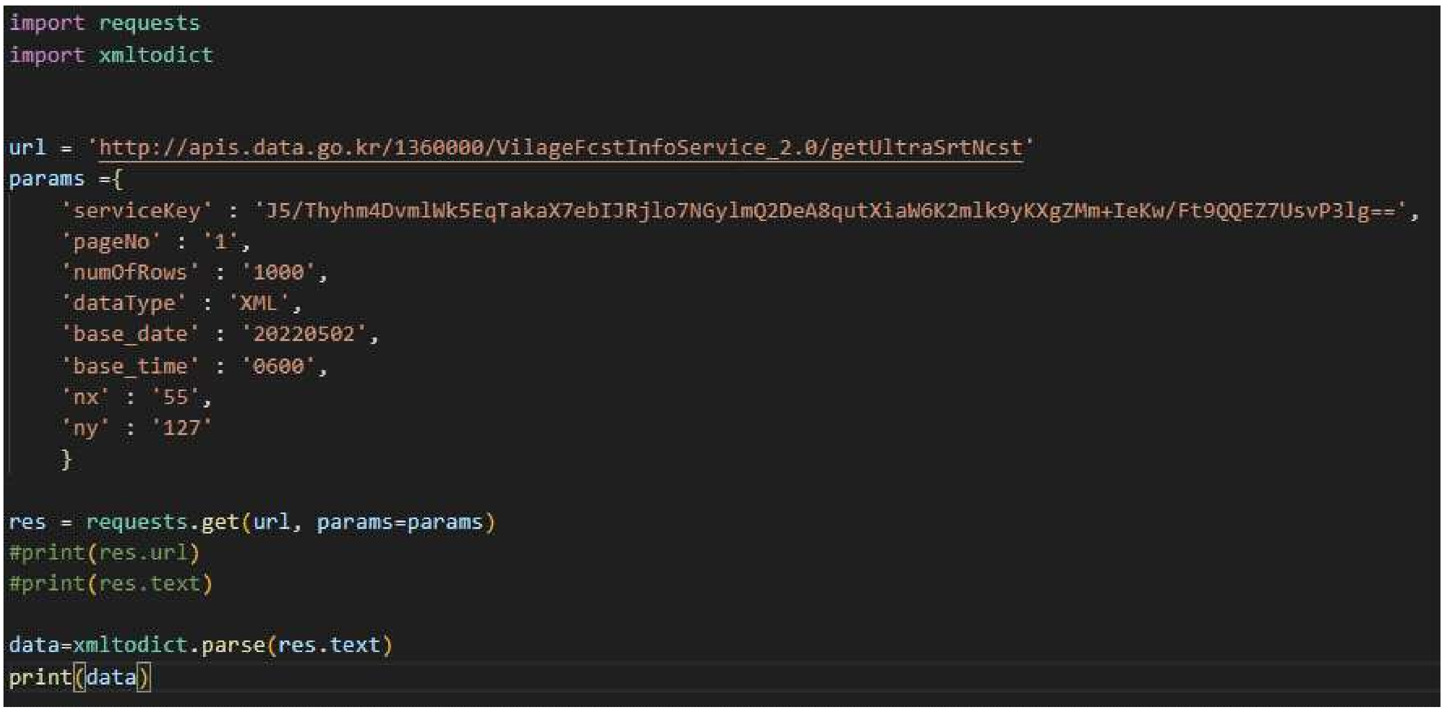
```python
import requests


url = 'http://apis.data.go.kr/1360000/VilageFcstInfoService_2.0/getUltraSrtNcst'
params ={

'serviceKey':'J5/Thyhm4DvmlWk5EqTakaX7ebIJRjlo7NGylmQ2DeA8qutXiaW6K2mlk9yKXgZMm+IeKw/Ft9QQEZ7UsvP3lg==',
        'pageNo' : '1',
        'numOfRows' : '1000',
        'dataType' : 'XML',
        'base_date' : base_date, ##'20220504'
        'base_time' : base_time, #'2300',##
        'nx' : '59',
        'ny' : '89'
        }
res = requests.get(url, params=params)
print(res.url)
print(res.text)
```

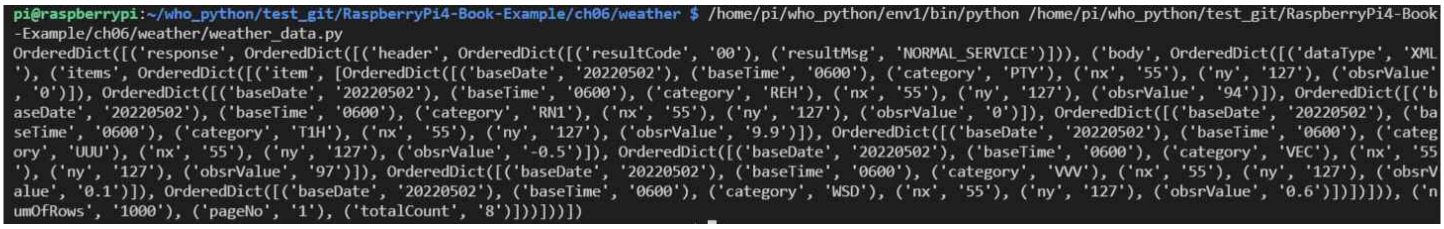
```
pi@raspberrypi:~/who_python $ pip install xmltodict
Defaulting to user installation because normal site-packages is not writeable
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting xmltodict
  Downloading https://www.piwheels.org/simple/xmltodict/xmltodict-0.12.0-py2.py3-none-any.whl (9.2 kB)
Installing collected packages: xmltodict
Successfully installed xmltodict-0.12.0
pi@raspberrypi:~/who_python $
```

임의의 디렉토리에서 작업하면 나중에 에러가 난다. 그러니 본인의 작업 폴더에서 인스톨한다.

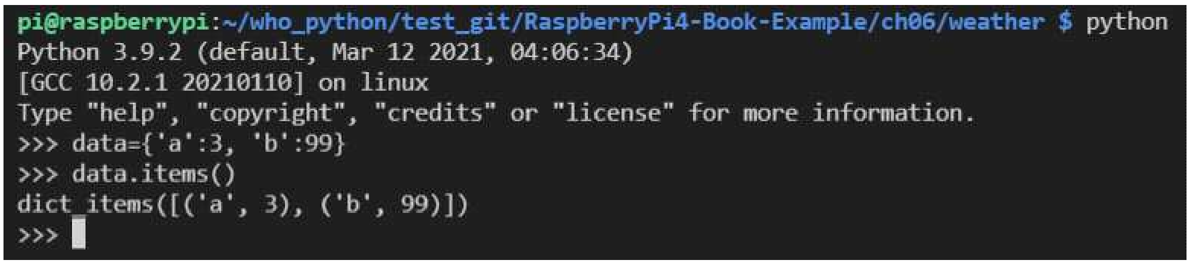
```
pi@raspberrypi:~/who_python/test_git/RaspberryPi4-Book-Example/ch06/weather $ pip install xmltodict
```



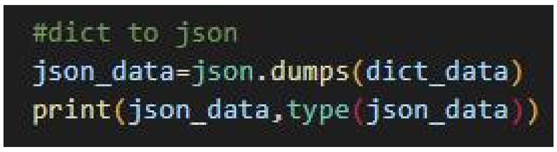
```python
import requests
import xmltodict


url = 'http://apis.data.go.kr/1360000/VilageFcstInfoService_2.0/getUltraSrtNcst'
params ={
    'serviceKey' : 'J5/Thyhm4DvmlWk5EqTakaX7ebIJRjlo7NGylmQ2DeA8qutXiaW6K2mlk9yKXgZMm+IeKw/Ft9QQEZ7UsvP3lg==',
    'pageNo' : '1',
    'numOfRows' : '1000',
    'dataType' : 'XML',
    'base_date' : '20220502',
    'base_time' : '0600',
    'nx' : '55',
    'ny' : '127'
    }

res = requests.get(url, params=params)
#print(res.url)
#print(res.text)

data=xmltodict.parse(res.text)
print(data)
```



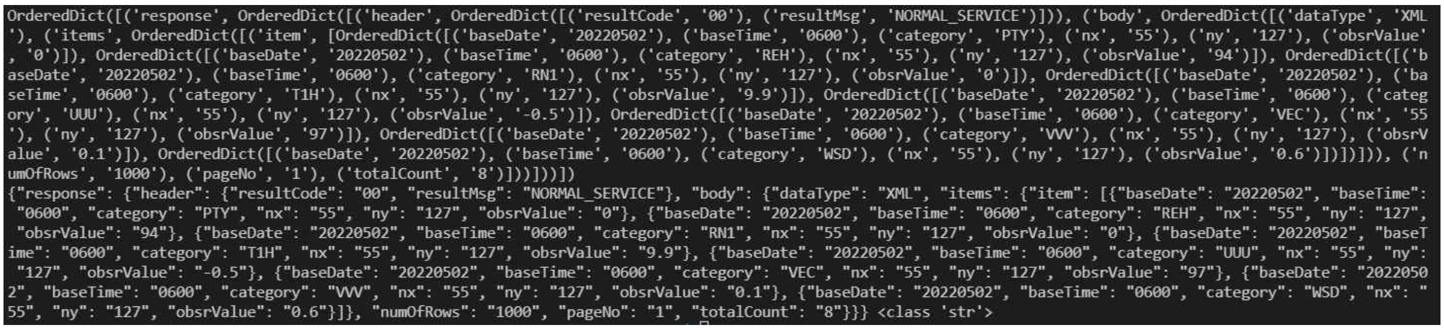
```
pi@raspberrypi:~/who_python/test_git/RaspberryPi4-Book-Example/ch06/weather $ /home/pi/who_python/env1/bin/python /home/pi/who_python/test_git/RaspberryPi4-Book
-Example/ch06/weather/weather_data.py
OrderedDict([('response', OrderedDict([('header', OrderedDict([('resultCode', '00'), ('resultMsg', 'NORMAL_SERVICE')])), ('body', OrderedDict([('dataType', 'XML
'), ('items', OrderedDict([('item', [OrderedDict([('baseDate', '20220502'), ('baseTime', '0600'), ('category', 'PTY'), ('nx', '55'), ('ny', '127'), ('obsrValue'
, '0')]), OrderedDict([('baseDate', '20220502'), ('baseTime', '0600'), ('category', 'REH'), ('nx', '55'), ('ny', '127'), ('obsrValue', '94')]), OrderedDict([('b
aseDate', '20220502'), ('baseTime', '0600'), ('category', 'RN1'), ('nx', '55'), ('ny', '127'), ('obsrValue', '0')]), OrderedDict([('baseDate', '20220502'), ('ba
seTime', '0600'), ('category', 'T1H'), ('nx', '55'), ('ny', '127'), ('obsrValue', '9.9')]), OrderedDict([('baseDate', '20220502'), ('baseTime', '0600'), ('categ
ory', 'UUU'), ('nx', '55'), ('ny', '127'), ('obsrValue', '-0.5')]), OrderedDict([('baseDate', '20220502'), ('baseTime', '0600'), ('category', 'VEC'), ('nx', '55
'), ('ny', '127'), ('obsrValue', '97')]), OrderedDict([('baseDate', '20220502'), ('baseTime', '0600'), ('category', 'VVV'), ('nx', '55'), ('ny', '127'), ('obsrV
alue', '0.1')]), OrderedDict([('baseDate', '20220502'), ('baseTime', '0600'), ('category', 'WSD'), ('nx', '55'), ('ny', '127'), ('obsrValue', '0.6')])])])), ('n
umOfRows', '1000'), ('pageNo', '1'), ('totalCount', '8')])])])])
```



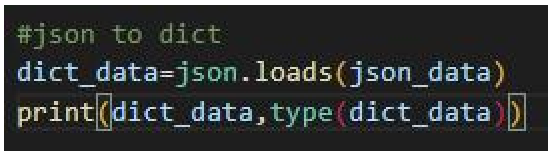
```
pi@raspberrypi:~/who_python/test_git/RaspberryPi4-Book-Example/ch06/weather $ python
Python 3.9.2 (default, Mar 12 2021, 04:06:34)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> data={'a':3, 'b':99}
>>> data.items()
dict_items([('a', 3), ('b', 99)])
>>> 
```



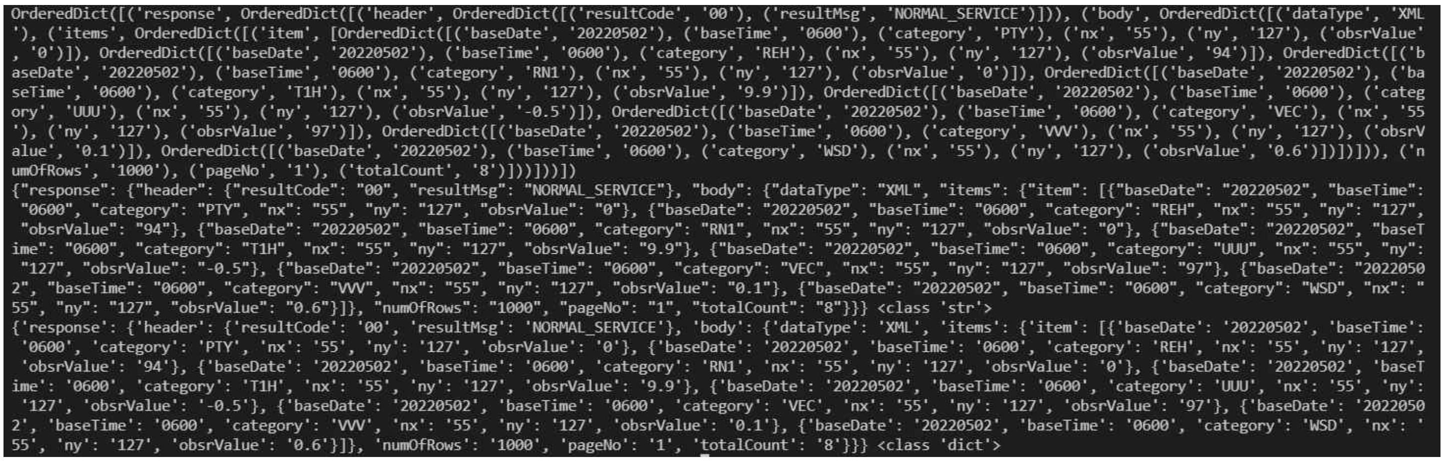
```python
#dict to json
json_data=json.dumps(dict_data)
print(json_data,type(json_data))
```



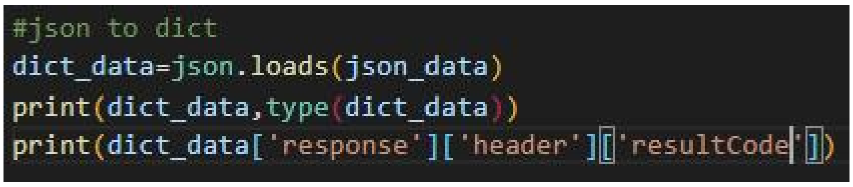
```
OrderedDict([('response', OrderedDict([('header', OrderedDict([('resultCode', '00'), ('resultMsg', 'NORMAL_SERVICE')])), ('body', OrderedDict([('dataType', 'XML
'), ('items', OrderedDict([('item', [OrderedDict([('baseDate', '20220502'), ('baseTime', '0600'), ('category', 'PTY'), ('nx', '55'), ('ny', '127'), ('obsrValue'
, '0')]), OrderedDict([('baseDate', '20220502'), ('baseTime', '0600'), ('category', 'REH'), ('nx', '55'), ('ny', '127'), ('obsrValue', '94')]), OrderedDict([('b
aseDate', '20220502'), ('baseTime', '0600'), ('category', 'RN1'), ('nx', '55'), ('ny', '127'), ('obsrValue', '0')]), OrderedDict([('baseDate', '20220502'), ('ba
seTime', '0600'), ('category', 'T1H'), ('nx', '55'), ('ny', '127'), ('obsrValue', '9.9')]), OrderedDict([('baseDate', '20220502'), ('baseTime', '0600'), ('categ
ory', 'UUU'), ('nx', '55'), ('ny', '127'), ('obsrValue', '-0.5')]), OrderedDict([('baseDate', '20220502'), ('baseTime', '0600'), ('category', 'VEC'), ('nx', '55
'), ('ny', '127'), ('obsrValue', '97')]), OrderedDict([('baseDate', '20220502'), ('baseTime', '0600'), ('category', 'VVV'), ('nx', '55'), ('ny', '127'), ('obsrV
alue', '0.1')]), OrderedDict([('baseDate', '20220502'), ('baseTime', '0600'), ('category', 'WSD'), ('nx', '55'), ('ny', '127'), ('obsrValue', '0.6')])])])), ('n
umOfRows', '1000'), ('pageNo', '1'), ('totalCount', '8')]))]))])
{"response": {"header": {"resultCode": "00", "resultMsg": "NORMAL_SERVICE"}, "body": {"dataType": "XML", "items": {"item": [{"baseDate": "20220502", "baseTime":
 "0600", "category": "PTY", "nx": "55", "ny": "127", "obsrValue": "0"}, {"baseDate": "20220502", "baseTime": "0600", "category": "REH", "nx": "55", "ny": "127",
 "obsrValue": "94"}, {"baseDate": "20220502", "baseTime": "0600", "category": "RN1", "nx": "55", "ny": "127", "obsrValue": "0"}, {"baseDate": "20220502", "baseT
ime": "0600", "category": "T1H", "nx": "55", "ny": "127", "obsrValue": "9.9"}, {"baseDate": "20220502", "baseTime": "0600", "category": "UUU", "nx": "55", "ny":
 "127", "obsrValue": "-0.5"}, {"baseDate": "20220502", "baseTime": "0600", "category": "VEC", "nx": "55", "ny": "127", "obsrValue": "97"}, {"baseDate": "2022050
2", "baseTime": "0600", "category": "VVV", "nx": "55", "ny": "127", "obsrValue": "0.1"}, {"baseDate": "20220502", "baseTime": "0600", "category": "WSD", "nx": "
55", "ny": "127", "obsrValue": "0.6"}]}, "numOfRows": "1000", "pageNo": "1", "totalCount": "8"}}} <class 'str'>
```



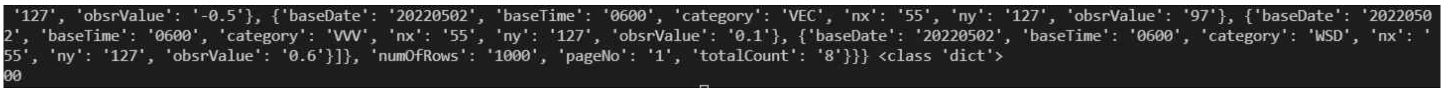
```python
#json to dict
dict_data=json.loads(json_data)
print(dict_data,type(dict_data))
```



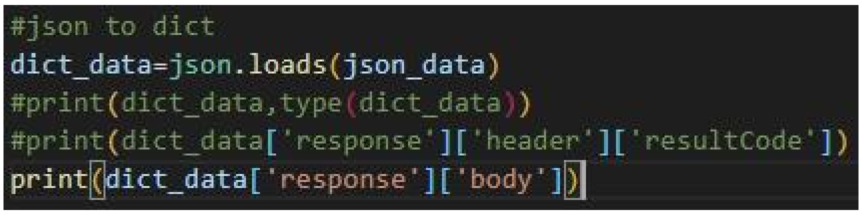
```
OrderedDict([('response', OrderedDict([('header', OrderedDict([('resultCode', '00'), ('resultMsg', 'NORMAL_SERVICE')])), ('body', OrderedDict([('dataType', 'XML
'), ('items', OrderedDict([('item', [OrderedDict([('baseDate', '20220502'), ('baseTime', '0600'), ('category', 'PTY'), ('nx', '55'), ('ny', '127'), ('obsrValue'
, '0')]), OrderedDict([('baseDate', '20220502'), ('baseTime', '0600'), ('category', 'REH'), ('nx', '55'), ('ny', '127'), ('obsrValue', '94')]), OrderedDict([('b
aseDate', '20220502'), ('baseTime', '0600'), ('category', 'RN1'), ('nx', '55'), ('ny', '127'), ('obsrValue', '0')]), OrderedDict([('baseDate', '20220502'), ('ba
seTime', '0600'), ('category', 'T1H'), ('nx', '55'), ('ny', '127'), ('obsrValue', '9.9')]), OrderedDict([('baseDate', '20220502'), ('baseTime', '0600'), ('categ
ory', 'UUU'), ('nx', '55'), ('ny', '127'), ('obsrValue', '-0.5')]), OrderedDict([('baseDate', '20220502'), ('baseTime', '0600'), ('category', 'VEC'), ('nx', '55
'), ('ny', '127'), ('obsrValue', '97')]), OrderedDict([('baseDate', '20220502'), ('baseTime', '0600'), ('category', 'VVV'), ('nx', '55'), ('ny', '127'), ('obsrV
alue', '0.1')]), OrderedDict([('baseDate', '20220502'), ('baseTime', '0600'), ('category', 'WSD'), ('nx', '55'), ('ny', '127'), ('obsrValue', '0.6')])])])), ('n
umOfRows', '1000'), ('pageNo', '1'), ('totalCount', '8')]))]))])
{"response": {"header": {"resultCode": "00", "resultMsg": "NORMAL_SERVICE"}, "body": {"dataType": "XML", "items": {"item": [{"baseDate": "20220502", "baseTime":
 "0600", "category": "PTY", "nx": "55", "ny": "127", "obsrValue": "0"}, {"baseDate": "20220502", "baseTime": "0600", "category": "REH", "nx": "55", "ny": "127",
 "obsrValue": "94"}, {"baseDate": "20220502", "baseTime": "0600", "category": "RN1", "nx": "55", "ny": "127", "obsrValue": "0"}, {"baseDate": "20220502", "baseT
ime": "0600", "category": "T1H", "nx": "55", "ny": "127", "obsrValue": "9.9"}, {"baseDate": "20220502", "baseTime": "0600", "category": "UUU", "nx": "55", "ny":
 "127", "obsrValue": "-0.5"}, {"baseDate": "20220502", "baseTime": "0600", "category": "VEC", "nx": "55", "ny": "127", "obsrValue": "97"}, {"baseDate": "2022050
2", "baseTime": "0600", "category": "VVV", "nx": "55", "ny": "127", "obsrValue": "0.1"}, {"baseDate": "20220502", "baseTime": "0600", "category": "WSD", "nx": "
55", "ny": "127", "obsrValue": "0.6"}]}, "numOfRows": "1000", "pageNo": "1", "totalCount": "8"}}} <class 'str'>
{'response': {'header': {'resultCode': '00', 'resultMsg': 'NORMAL_SERVICE'}, 'body': {'dataType': 'XML', 'items': {'item': [{'baseDate': '20220502', 'baseTime':
 '0600', 'category': 'PTY', 'nx': '55', 'ny': '127', 'obsrValue': '0'}, {'baseDate': '20220502', 'baseTime': '0600', 'category': 'REH', 'nx': '55', 'ny': '127',
 'obsrValue': '94'}, {'baseDate': '20220502', 'baseTime': '0600', 'category': 'RN1', 'nx': '55', 'ny': '127', 'obsrValue': '0'}, {'baseDate': '20220502', 'baseT
ime': '0600', 'category': 'T1H', 'nx': '55', 'ny': '127', 'obsrValue': '9.9'}, {'baseDate': '20220502', 'baseTime': '0600', 'category': 'UUU', 'nx': '55', 'ny':
 '127', 'obsrValue': '-0.5'}, {'baseDate': '20220502', 'baseTime': '0600', 'category': 'VEC', 'nx': '55', 'ny': '127', 'obsrValue': '97'}, {'baseDate': '2022050
2', 'baseTime': '0600', 'category': 'VVV', 'nx': '55', 'ny': '127', 'obsrValue': '0.1'}, {'baseDate': '20220502', 'baseTime': '0600', 'category': 'WSD', 'nx': '
55', 'ny': '127', 'obsrValue': '0.6'}]}, 'numOfRows': '1000', 'pageNo': '1', 'totalCount': '8'}}} <class 'dict'>
```



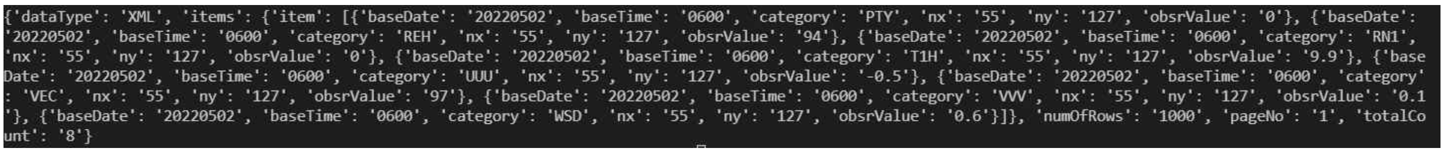
```python
#json to dict
dict_data=json.loads(json_data)
print(dict_data,type(dict_data))
print(dict_data['response']['header']['resultCode'])
```



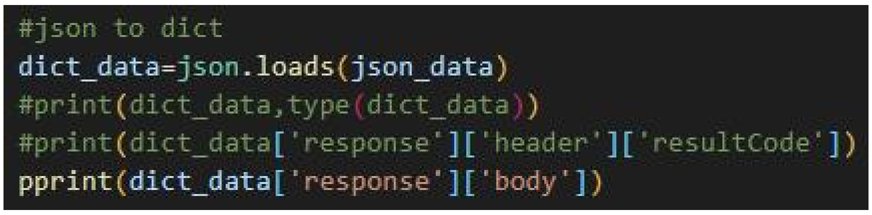
```
 '127', 'obsrValue': '-0.5'}, {'baseDate': '20220502', 'baseTime': '0600', 'category': 'VEC', 'nx': '55', 'ny': '127', 'obsrValue': '97'}, {'baseDate': '2022050
2', 'baseTime': '0600', 'category': 'VVV', 'nx': '55', 'ny': '127', 'obsrValue': '0.1'}, {'baseDate': '20220502', 'baseTime': '0600', 'category': 'WSD', 'nx': '
55', 'ny': '127', 'obsrValue': '0.6'}]}, 'numOfRows': '1000', 'pageNo': '1', 'totalCount': '8'}}} <class 'dict'>
00
```



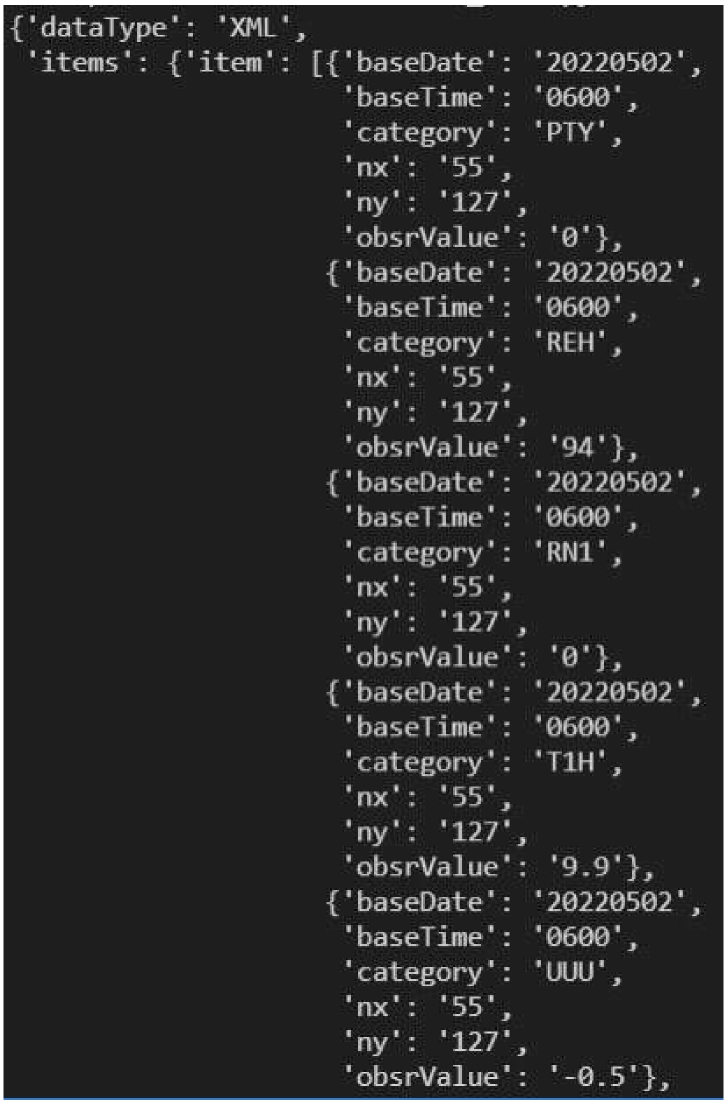
```python
#json to dict
dict_data=json.loads(json_data)
#print(dict_data,type(dict_data))
#print(dict_data['response']['header']['resultCode'])
print(dict_data['response']['body'])
```



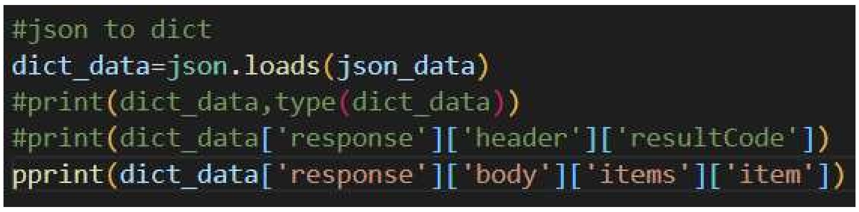
```
{'dataType': 'XML', 'items': {'item': [{'baseDate': '20220502', 'baseTime': '0600', 'category': 'PTY', 'nx': '55', 'ny': '127', 'obsrValue': '0'}, {'baseDate':
'20220502', 'baseTime': '0600', 'category': 'REH', 'nx': '55', 'ny': '127', 'obsrValue': '94'}, {'baseDate': '20220502', 'baseTime': '0600', 'category': 'RN1',
'nx': '55', 'ny': '127', 'obsrValue': '0'}, {'baseDate': '20220502', 'baseTime': '0600', 'category': 'T1H', 'nx': '55', 'ny': '127', 'obsrValue': '9.9'}, {'base
Date': '20220502', 'baseTime': '0600', 'category': 'UUU', 'nx': '55', 'ny': '127', 'obsrValue': '-0.5'}, {'baseDate': '20220502', 'baseTime': '0600', 'category'
: 'VEC', 'nx': '55', 'ny': '127', 'obsrValue': '97'}, {'baseDate': '20220502', 'baseTime': '0600', 'category': 'VVV', 'nx': '55', 'ny': '127', 'obsrValue': '0.1
'}, {'baseDate': '20220502', 'baseTime': '0600', 'category': 'WSD', 'nx': '55', 'ny': '127', 'obsrValue': '0.6'}]}, 'numOfRows': '1000', 'pageNo': '1', 'totalCo
unt': '8'}
```



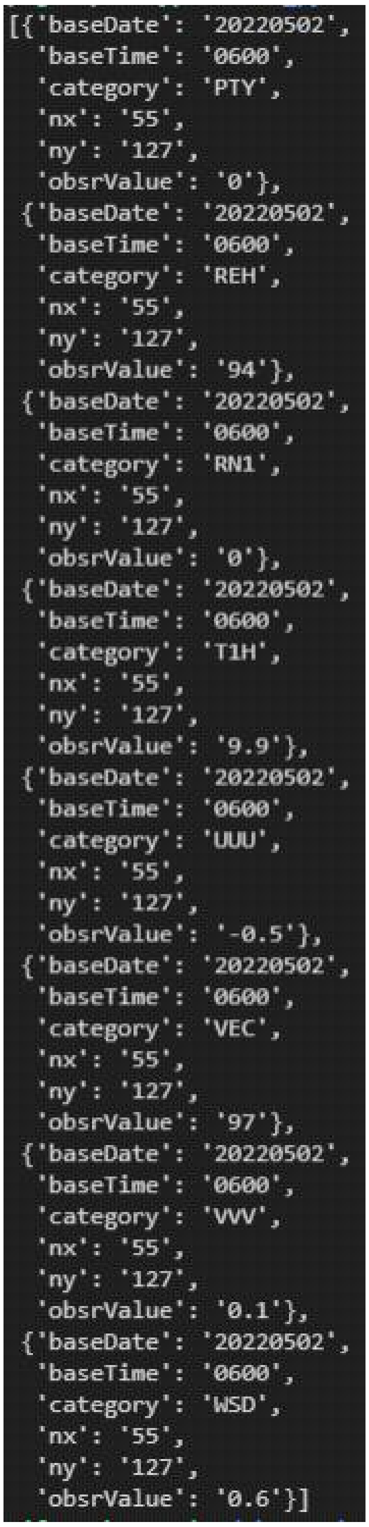
```python
#json to dict
dict_data=json.loads(json_data)
#print(dict_data,type(dict_data))
#print(dict_data['response']['header']['resultCode'])
pprint(dict_data['response']['body'])
```



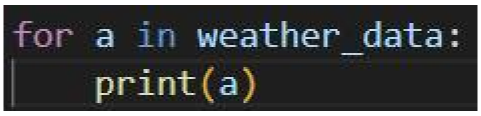
```
{'dataType': 'XML',
 'items': {'item': [{'baseDate': '20220502',
                     'baseTime': '0600',
                     'category': 'PTY',
                     'nx': '55',
                     'ny': '127',
                     'obsrValue': '0'},
                    {'baseDate': '20220502',
                     'baseTime': '0600',
                     'category': 'REH',
                     'nx': '55',
                     'ny': '127',
                     'obsrValue': '94'},
                    {'baseDate': '20220502',
                     'baseTime': '0600',
                     'category': 'RN1',
                     'nx': '55',
                     'ny': '127',
                     'obsrValue': '0'},
                    {'baseDate': '20220502',
                     'baseTime': '0600',
                     'category': 'T1H',
                     'nx': '55',
                     'ny': '127',
                     'obsrValue': '9.9'},
                    {'baseDate': '20220502',
                     'baseTime': '0600',
                     'category': 'UUU',
                     'nx': '55',
                     'ny': '127',
                     'obsrValue': '-0.5'},
```



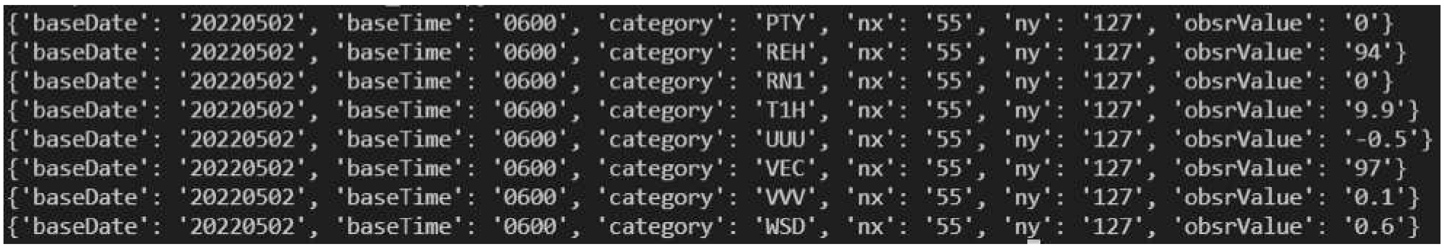
```python
#json to dict
dict_data=json.loads(json_data)
#print(dict_data,type(dict_data))
#print(dict_data['response']['header']['resultCode'])
pprint(dict_data['response']['body']['items']['item'])
```



```
[{'baseDate': '20220502',
  'baseTime': '0600',
  'category': 'PTY',
  'nx': '55',
  'ny': '127',
  'obsrValue': '0'},
 {'baseDate': '20220502',
  'baseTime': '0600',
  'category': 'REH',
  'nx': '55',
  'ny': '127',
  'obsrValue': '94'},
 {'baseDate': '20220502',
  'baseTime': '0600',
  'category': 'RN1',
  'nx': '55',
  'ny': '127',
  'obsrValue': '0'},
 {'baseDate': '20220502',
  'baseTime': '0600',
  'category': 'T1H',
  'nx': '55',
  'ny': '127',
  'obsrValue': '9.9'},
 {'baseDate': '20220502',
  'baseTime': '0600',
  'category': 'UUU',
  'nx': '55',
  'ny': '127',
  'obsrValue': '-0.5'},
 {'baseDate': '20220502',
  'baseTime': '0600',
  'category': 'VEC',
  'nx': '55',
  'ny': '127',
  'obsrValue': '97'},
 {'baseDate': '20220502',
  'baseTime': '0600',
  'category': 'VVV',
  'nx': '55',
  'ny': '127',
  'obsrValue': '0.1'},
 {'baseDate': '20220502',
  'baseTime': '0600',
  'category': 'WSD',
  'nx': '55',
  'ny': '127',
  'obsrValue': '0.6'}]
```

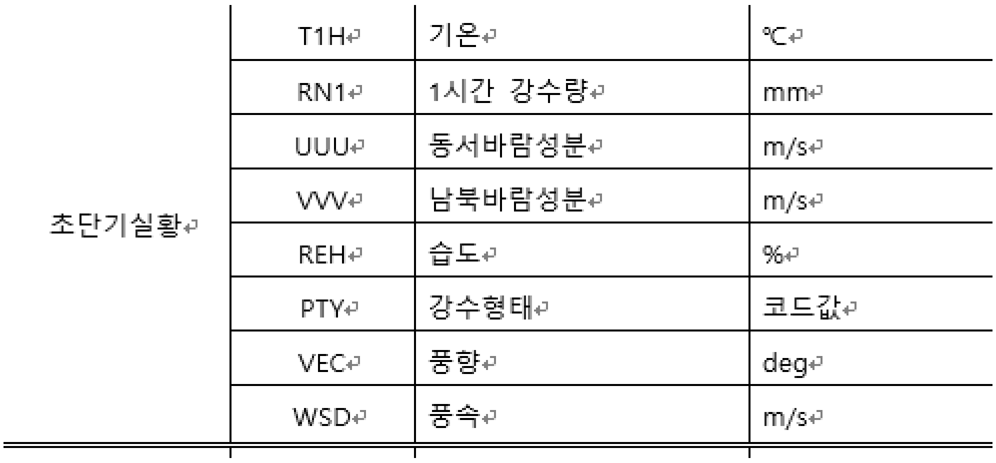


```
for a in weather_data:
    print(a)
```

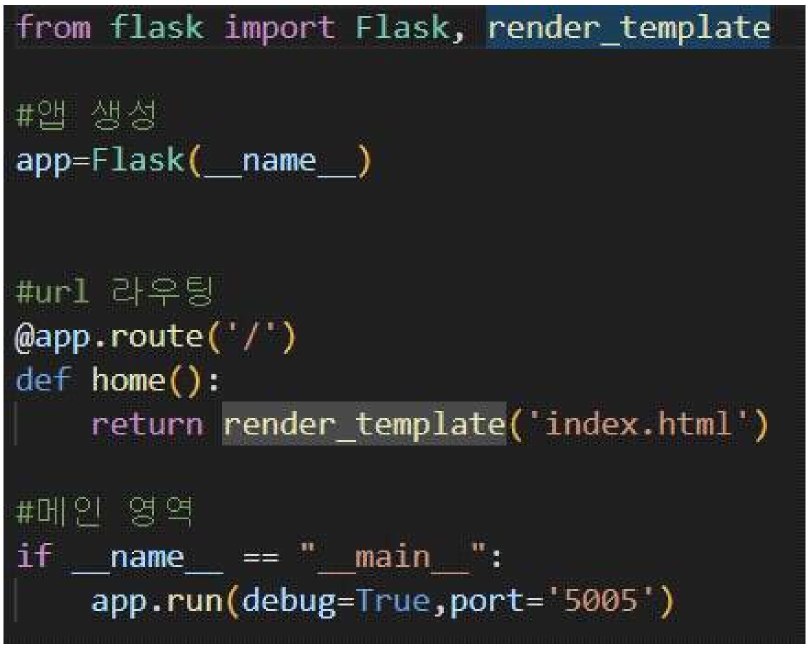


```
{'baseDate': '20220502', 'baseTime': '0600', 'category': 'PTY', 'nx': '55', 'ny': '127', 'obsrValue': '0'}
{'baseDate': '20220502', 'baseTime': '0600', 'category': 'REH', 'nx': '55', 'ny': '127', 'obsrValue': '94'}
{'baseDate': '20220502', 'baseTime': '0600', 'category': 'RN1', 'nx': '55', 'ny': '127', 'obsrValue': '0'}
{'baseDate': '20220502', 'baseTime': '0600', 'category': 'T1H', 'nx': '55', 'ny': '127', 'obsrValue': '9.9'}
{'baseDate': '20220502', 'baseTime': '0600', 'category': 'UUU', 'nx': '55', 'ny': '127', 'obsrValue': '-0.5'}
{'baseDate': '20220502', 'baseTime': '0600', 'category': 'VEC', 'nx': '55', 'ny': '127', 'obsrValue': '97'}
{'baseDate': '20220502', 'baseTime': '0600', 'category': 'VVV', 'nx': '55', 'ny': '127', 'obsrValue': '0.1'}
{'baseDate': '20220502', 'baseTime': '0600', 'category': 'WSD', 'nx': '55', 'ny': '127', 'obsrValue': '0.6'}
```

자 그러면 weather_data.py를 함수로 만들어 보자

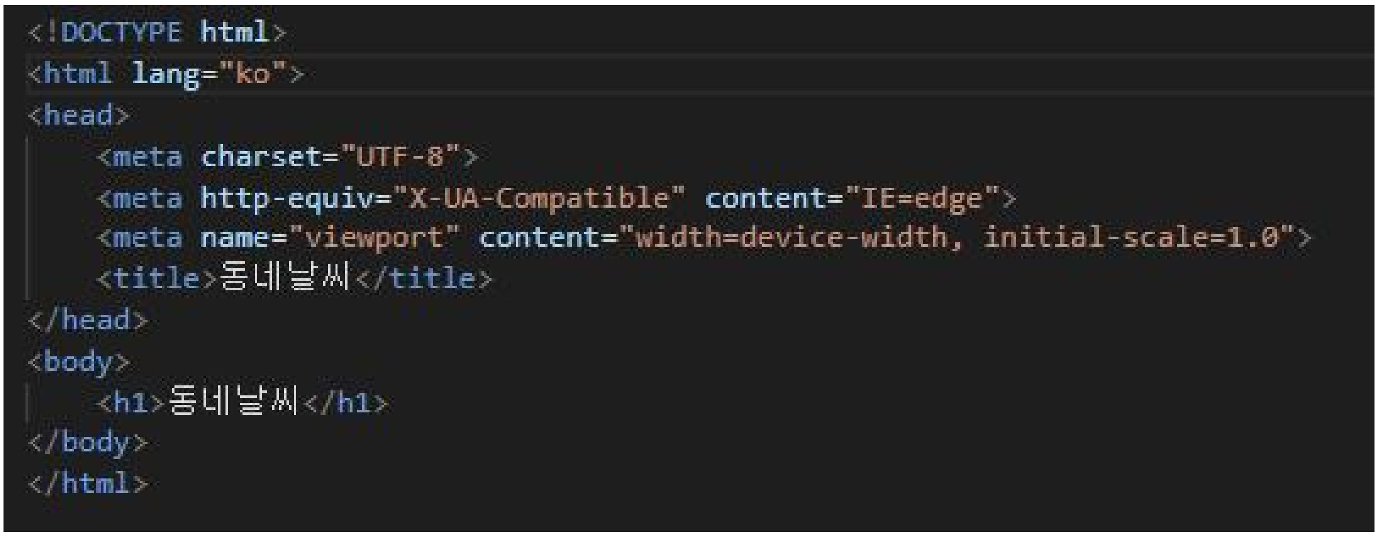


초단기실황	T1H	기온	℃
	RN1	1시간 강수량	mm
	UUU	동서바람성분	m/s
	VVV	남북바람성분	m/s
	REH	습도	%
	PTY	강수형태	코드값
	VEC	풍향	deg
	WSD	풍속	m/s



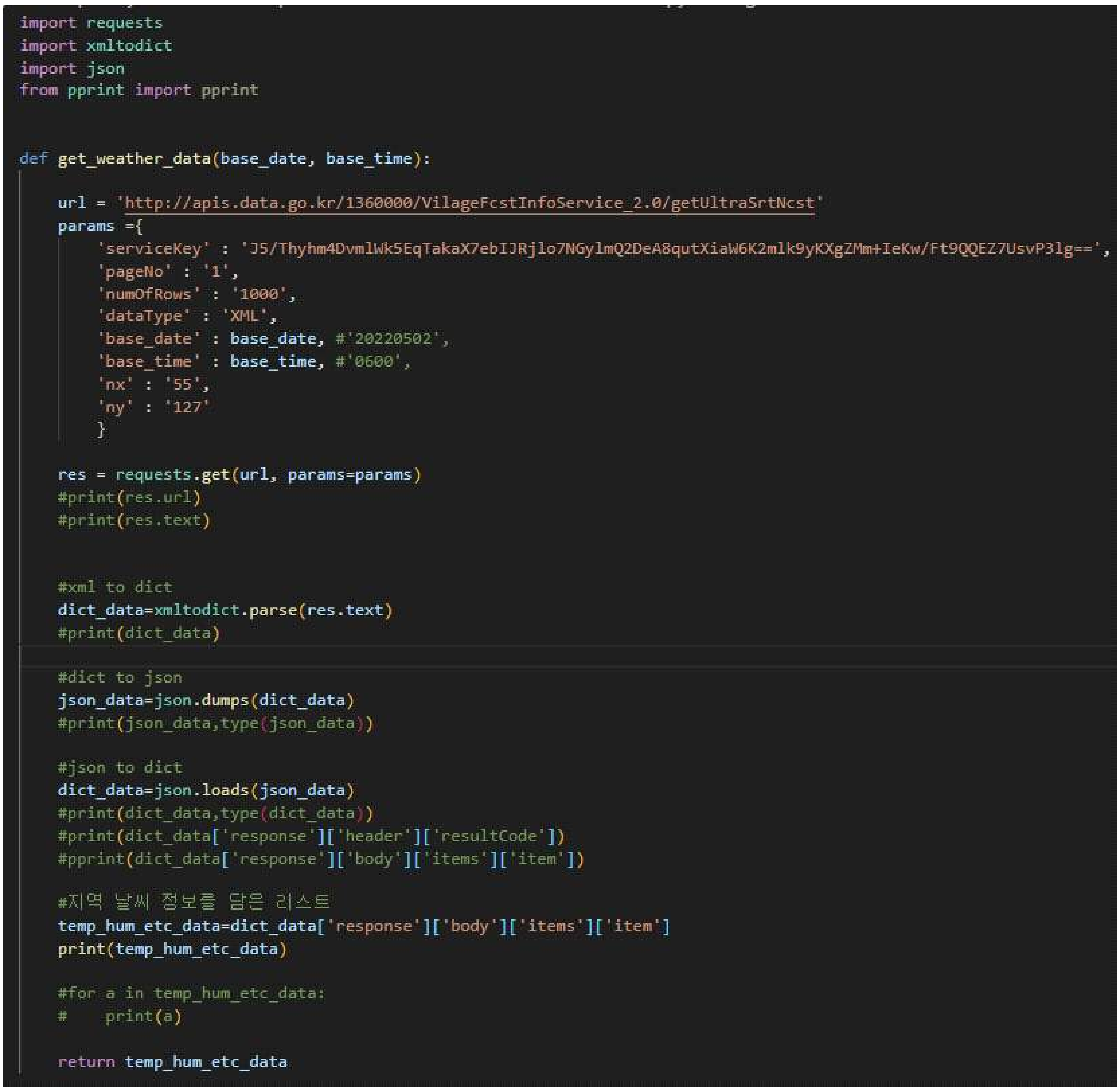
```python
from flask import Flask, render_template

#앱 생성
app=Flask(__name__)


#url 라우팅
@app.route('/')
def home():
    return render_template('index.html')

#메인 영역
if __name__ == "__main__":
    app.run(debug=True,port='5005')
```

그림 19 app.py



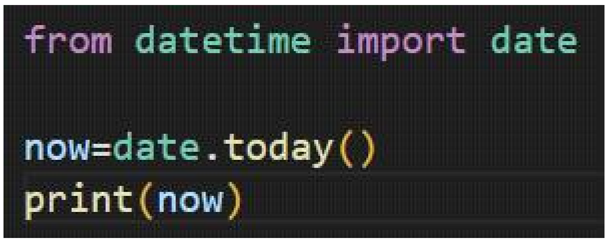
```html
<!DOCTYPE html>
<html lang="ko">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>동네날씨</title>
</head>
<body>
    <h1>동네날씨</h1>
</body>
</html>
```

그림 20 index.html(templates 폴더에 반드시 있어야만 함)

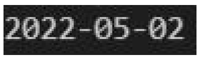


```python
import requests
import xmltodict
import json
from pprint import pprint


def get_weather_data(base_date, base_time):

    url = 'http://apis.data.go.kr/1360000/VilageFcstInfoService_2.0/getUltraSrtNcst'
    params ={
        'serviceKey' : 'J5/Thyhm4DvmlWk5EqTakaX7ebIJRjlo7NGylmQ2DeA8qutXiaW6K2mlk9yKXgZMm+IeKw/Ft9QQEZ7UsvP3lg==',
        'pageNo' : '1',
        'numOfRows' : '1000',
        'dataType' : 'XML',
        'base_date' : base_date, #'20220502',
        'base_time' : base_time, #'0600',
        'nx' : '55',
        'ny' : '127'
        }

    res = requests.get(url, params=params)
    #print(res.url)
    #print(res.text)


    #xml to dict
    dict_data=xmltodict.parse(res.text)
    #print(dict_data)

    #dict to json
    json_data=json.dumps(dict_data)
    #print(json_data,type(json_data))

    #json to dict
    dict_data=json.loads(json_data)
    #print(dict_data,type(dict_data))
    #print(dict_data['response']['header']['resultCode'])
    #pprint(dict_data['response']['body']['items']['item'])

    #지역 날씨 정보를 담은 리스트
    temp_hum_etc_data=dict_data['response']['body']['items']['item']
    print(temp_hum_etc_data)

    #for a in temp_hum_etc_data:
    #    print(a)

    return temp_hum_etc_data
```

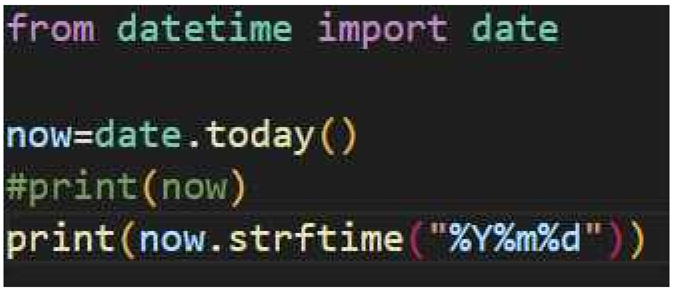
그림 21 weather_data.py



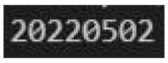
```python
from datetime import date

now=date.today()
print(now)
```



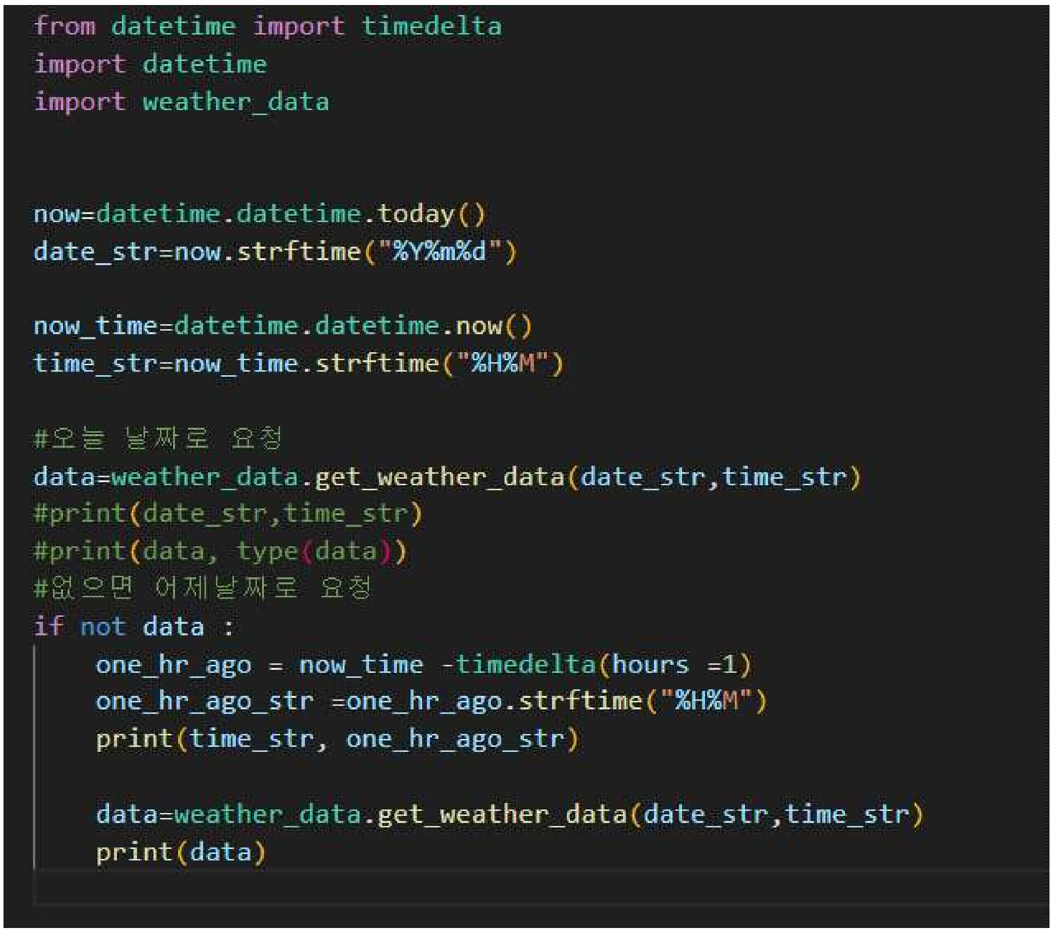
```
2022-05-02
```



```python
from datetime import date

now=date.today()
#print(now)
print(now.strftime("%Y%m%d"))
```

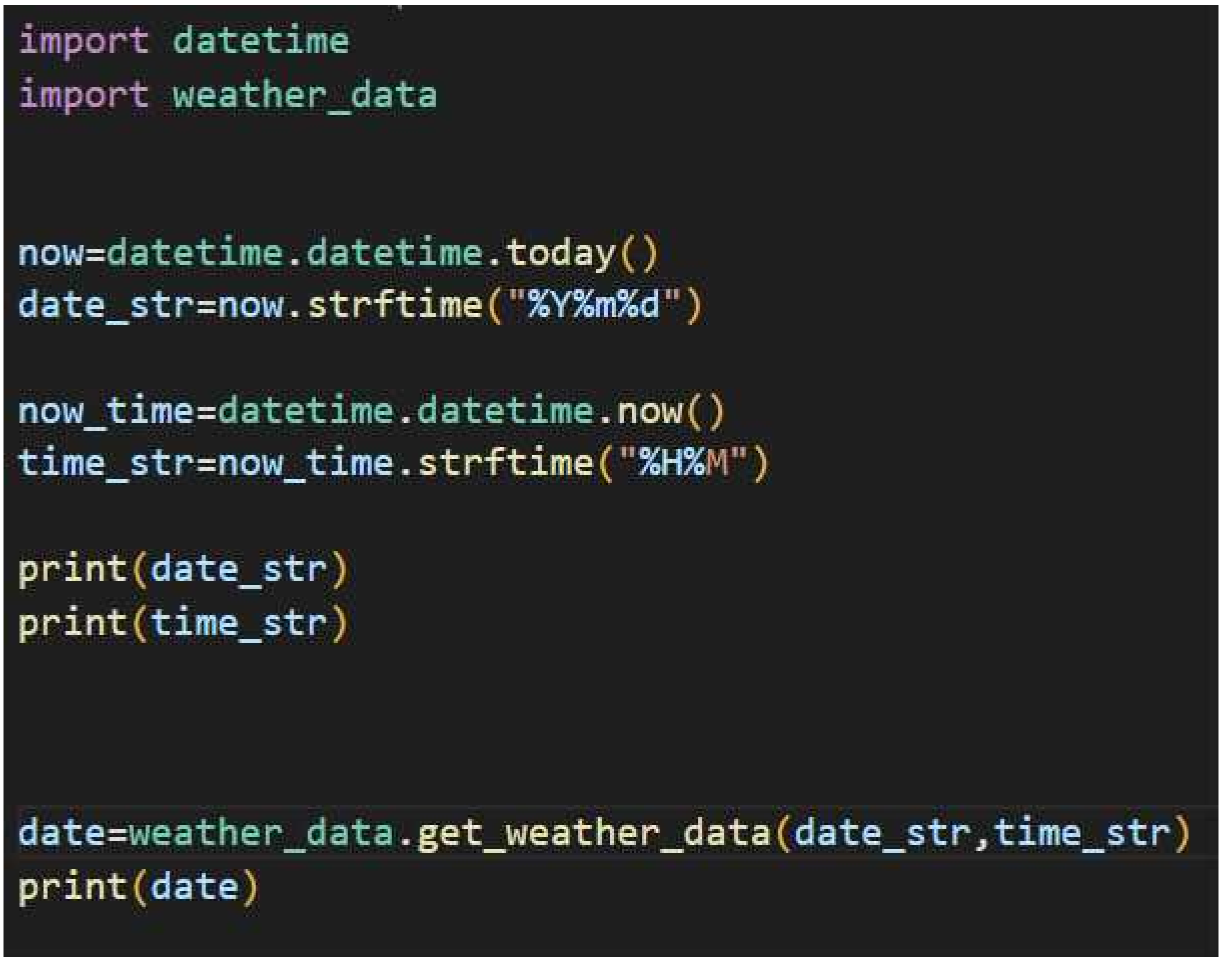


```
20220502
```



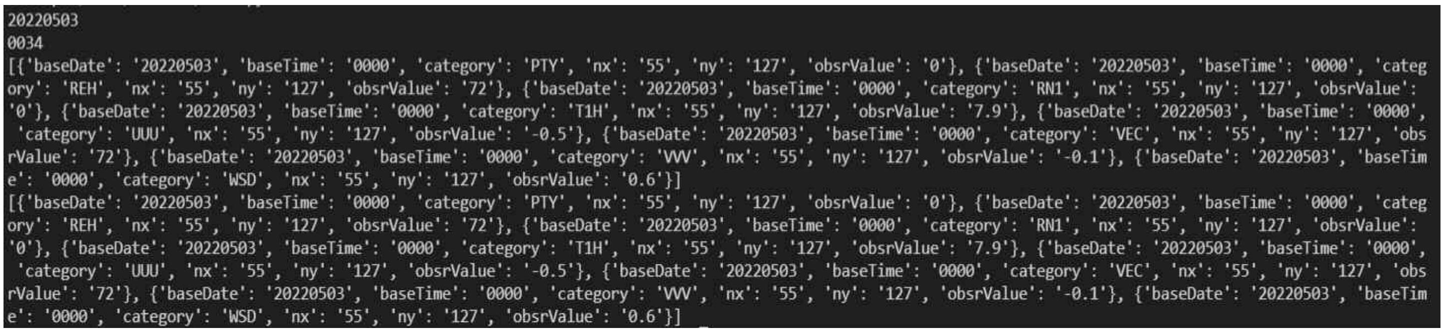
```python
from datetime import timedelta
import datetime
import weather_data


now=datetime.datetime.today()
date_str=now.strftime("%Y%m%d")

now_time=datetime.datetime.now()
time_str=now_time.strftime("%H%M")

#오늘 날짜로 요청
data=weather_data.get_weather_data(date_str,time_str)
#print(date_str,time_str)
#print(data, type(data))
#없으면 어제날짜로 요청
if not data :
    one_hr_ago = now_time -timedelta(hours =1)
    one_hr_ago_str =one_hr_ago.strftime("%H%M")
    print(time_str, one_hr_ago_str)

    data=weather_data.get_weather_data(date_str,time_str)
    print(data)
```

그림 26 에러 발생 시 회피 방법



```python
import datetime
import weather_data


now=datetime.datetime.today()
date_str=now.strftime("%Y%m%d")

now_time=datetime.datetime.now()
time_str=now_time.strftime("%H%M")

print(date_str)
print(time_str)



date=weather_data.get_weather_data(date_str,time_str)
print(date)
```

그림 27 현재 날짜와 시간을 주어진 포맷을 변환



```
20220503
0034
[{'baseDate': '20220503', 'baseTime': '0000', 'category': 'PTY', 'nx': '55', 'ny': '127', 'obsrValue': '0'}, {'baseDate': '20220503', 'baseTime': '0000', 'categ
ory': 'REH', 'nx': '55', 'ny': '127', 'obsrValue': '72'}, {'baseDate': '20220503', 'baseTime': '0000', 'category': 'RN1', 'nx': '55', 'ny': '127', 'obsrValue':
'0'}, {'baseDate': '20220503', 'baseTime': '0000', 'category': 'T1H', 'nx': '55', 'ny': '127', 'obsrValue': '7.9'}, {'baseDate': '20220503', 'baseTime': '0000',
 'category': 'UUU', 'nx': '55', 'ny': '127', 'obsrValue': '-0.5'}, {'baseDate': '20220503', 'baseTime': '0000', 'category': 'VEC', 'nx': '55', 'ny': '127', 'obs
rValue': '72'}, {'baseDate': '20220503', 'baseTime': '0000', 'category': 'VVV', 'nx': '55', 'ny': '127', 'obsrValue': '-0.1'}, {'baseDate': '20220503', 'baseTim
e': '0000', 'category': 'WSD', 'nx': '55', 'ny': '127', 'obsrValue': '0.6'}]
[{'baseDate': '20220503', 'baseTime': '0000', 'category': 'PTY', 'nx': '55', 'ny': '127', 'obsrValue': '0'}, {'baseDate': '20220503', 'baseTime': '0000', 'categ
ory': 'REH', 'nx': '55', 'ny': '127', 'obsrValue': '72'}, {'baseDate': '20220503', 'baseTime': '0000', 'category': 'RN1', 'nx': '55', 'ny': '127', 'obsrValue':
'0'}, {'baseDate': '20220503', 'baseTime': '0000', 'category': 'T1H', 'nx': '55', 'ny': '127', 'obsrValue': '7.9'}, {'baseDate': '20220503', 'baseTime': '0000',
 'category': 'UUU', 'nx': '55', 'ny': '127', 'obsrValue': '-0.5'}, {'baseDate': '20220503', 'baseTime': '0000', 'category': 'VEC', 'nx': '55', 'ny': '127', 'obs
rValue': '72'}, {'baseDate': '20220503', 'baseTime': '0000', 'category': 'VVV', 'nx': '55', 'ny': '127', 'obsrValue': '-0.1'}, {'baseDate': '20220503', 'baseTim
e': '0000', 'category': 'WSD', 'nx': '55', 'ny': '127', 'obsrValue': '0.6'}]
```

#app.py

```python
from flask import Flask, render_template
from datetime import datetime,timedelta
import weather_data


#앱 생성
app=Flask(__name__)


#url 라우팅
@app.route('/')
def home():
   # now=datetime.datetime.today()
    now=datetime.today()
    date_str=now.strftime("%Y%m%d")
    #now_time=datetime.datetime.now()
    now_time=datetime.now()

    time_str=now_time.strftime("%H%M")
    #오늘 날짜로 요청
    data=weather_data.get_weather_data(date_str,time_str)
    print(date_str,time_str)
    print(data, type(data))
    #없으면 어제날짜로 요청
    if not data :
        one_hr_ago = now_time -timedelta(hours =1)
        one_hr_ago_str =one_hr_ago.strftime("%H%M")
        print(time_str, one_hr_ago_str)

        data=weather_data.get_weather_data(date_str,one_hr_ago_str)
        print(data,type(data))

    r_response = data.get("response")
    r_body = r_response.get("body")
    r_items = r_body.get("items")
    r_item = r_items.get("item")


    result1={}
    result2={}
    for item in r_item:
        if(item.get("category")=='T1H'):
               result1 =item

        if(item.get("category")=='REH'):
               result2 =item

    print(result1,type(result1),result2,type(result2))



    return render_template('index.html', data1=result1,data2=result2)

#메인 영역
if __name__ == "__main__":
    app.run(debug=True,port='5005')
```

# weather_data.py

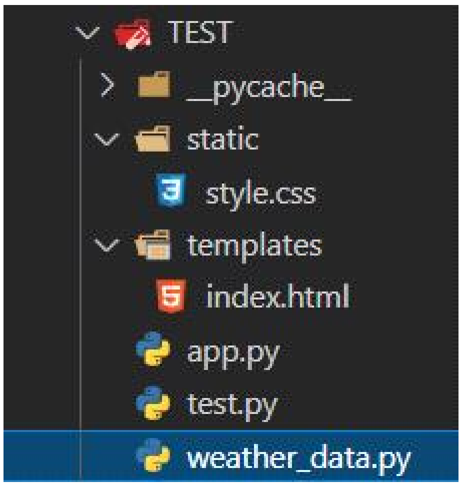
```python
import requests
import xmltodict
import json
from pprint import pprint


def get_weather_data(base_date, base_time):
    url = 'http://apis.data.go.kr/1360000/VilageFcstInfoService_2.0/getUltraSrtNcst'
    params ={
        'serviceKey':'J5/Thyhm4DvmlWk5EqTakaX7ebIJRjlo7NGylmQ2DeA8qutXiaW6K2mlk9yKXgZMm+IeKw/Ft9QQEZ7UsvP3lg==',
        'pageNo' : '1',
        'numOfRows' : '1000',
        'dataType' : 'XML',
        'base_date' : base_date, ##'20220504'
        'base_time' : base_time, #'2300',##
        'nx' : '59',
        'ny' : '89'
        }
    res = requests.get(url, params=params)
    #print(res.url)
    #print(res.text)


    #xml to dict
    dict_data=xmltodict.parse(res.text)
    #print(dict_data)
    #dict to json
    json_data=json.dumps(dict_data)
    #print(json_data,type(json_data))
    #json to dict
    dict_data=json.loads(json_data)
    print(dict_data,type(dict_data))
    #print(dict_data['response']['header']['resultCode'])
    #pprint(dict_data['response']['body']['items']['item'])


    #에러가 날 경우메시지 표시
    resultCode= dict_data['response']['header']['resultCode']
    if resultCode == "01":
        return False
    elif resultCode == "10":
        return '최근 1일 간의 자료만 제공합니다.'

    else:
        return dict_data
```

<! index.html>

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible"content="IE=edge">
    <meta name="viewport"content="width=device-width, initial-scale=1.0">
    <title>동네날씨</title>
    <link rel="stylesheet"href="{{ url_for('static', filename='style.css') }}">
</head>
<body>
    <h1>동  네  날  씨</h1>
        {{data1}}<hr>
        {{data2}}<hr>

{#
        {% for d in data %}
            {{d.category}},
        {% endfor %}    <hr>
        {% for d in data %}
            {{d.obsrValue}},
        {% endfor %}     <hr>
        {% for d in data %}
            {{d.baseTime}},
        {% endfor %}
#}

<table border=1>
    {% for key, value in data1.items() %}
    <tr>
        <th> {{key}} </th>
        <td> {{value}} </td>
    </tr>
       {% endfor %}

</table>
<table border=1>
    {% for key, value in data2.items() %}
    <tr>
        <th> {{key}} </th>
        <td> {{value}} </td>
    </tr>
        {% endfor %}

</table>

</body>
</html>
```

style.css

```
table{display : inline-block;}
```



```
TEST
  > __pycache__
  v static
      style.css
  v templates
      index.html
    app.py
    test.py
    weather_data.py
```