

# 파이썬 웹 크롤링

---

강사장철원

## □ API를 활용한 날씨 데이터 크롤링하기

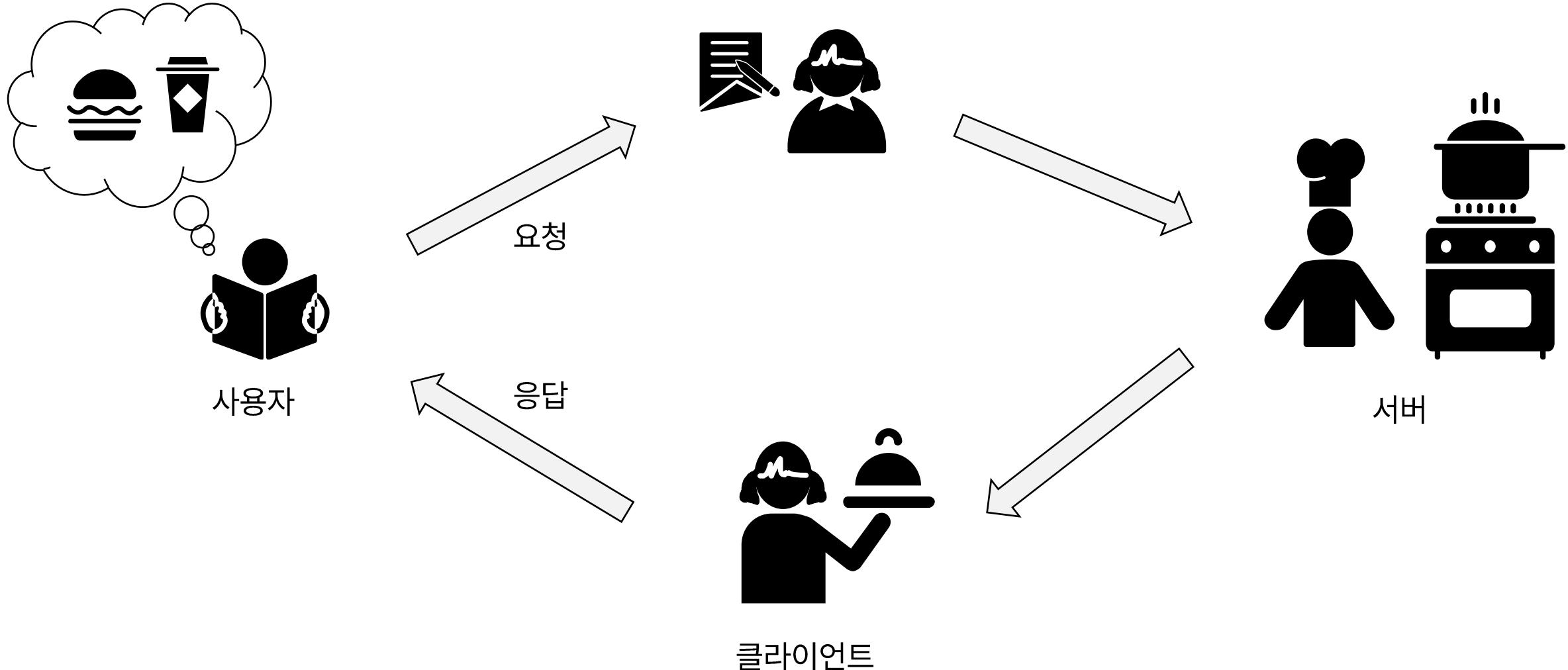
# 파이썬 웹 크롤링

Section 1. 날씨 데이터 크롤링

---

Section 1-1. 크롤링 웹사이트 확인

# API(Application Programming Interface)



## Section

### 크롤링 웹사이트 확인

# 크롤링 개요

<https://openweathermap.org/>

The screenshot shows the OpenWeatherMap homepage with a search bar at the top. The search bar has "seoul" typed into it and a "Search" button. Below the search bar, the current weather for Seoul, KR is displayed. The temperature is 17°C, feels like 16°C, with clear sky and light air. The hourly forecast shows temperatures from 18°C down to 10°C over a 24-hour period. The 8-day forecast for Seoul shows temperatures ranging from 17°C to 18°C with mostly clear sky. To the left, the current weather for London, GB is shown: 9°C, feels like 8°C, overcast clouds, light breeze. A map of London is also visible.

Nov 8, 04:35pm  
**Seoul, KR**

17°C

Feels like 16°C. Clear sky. Light air

▲ 1.5m/s WNW ⚡ 1027hPa  
Humidity: 36% UV: 0  
Dew point: 2°C Visibility: 10.0km

Hourly forecast

Time	Temp (°C)	Precip (%)
4pm	18	0%
5pm	17	0%
6pm	16	0%
7pm	15	0%
8pm	14	0%
9pm	13	0%
10pm	12	0%
11pm	11	0%
Nov 09	11	0%

8-day forecast

Date	Temp (°C)	Condition
Fri, Nov 08	17 / 8°C	clear sky
Sat, Nov 09	17 / 10°C	clear sky
Sun, Nov 10	17 / 12°C	overcast clouds
Mon, Nov 11	18 / 12°C	clear sky
Tue, Nov 12	18 / 12°C	clear sky
Wed, Nov 13	18 / 12°C	overcast clouds
Thu, Nov 14	18 / 13°C	clear sky

## Section

### 크롤링 웹사이트 확인

# 크롤링 개요

<https://openweathermap.org/>

Guide API Dashboard Marketplace Pricing Maps Our Initiatives Partners Blog

Guide API Dashboard Marketplace Pricing Maps Our Initiatives Partners

## Sign In To Your Account

 Enter email

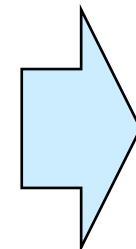
 Password

Remember me

**Submit**

Not registered? [Create an Account](#).

Lost your password? [Click here to recover](#).



## Create New Account

Username

Enter email

Password

Repeat Password

We will use information you provided for management and administration purposes, and for keeping you informed by mail, telephone, email and SMS of other products and services from us and our partners. You can proactively manage your preferences or opt-out of communications with us at any time using Privacy Centre. You have the right to access your data held by us or to request your data to be deleted. For full details please see the OpenWeather [Privacy Policy](#).

I am 16 years old and over

I agree with [Privacy Policy](#), [Terms and conditions of sale](#) and [Websites terms and conditions of use](#)

## Section

### 크롤링 웹사이트 확인

# 크롤링 개요

<https://openweathermap.org/>

#### ★ OpenWeatherMap Account confirmation ↗

^ 보낸사람 OpenWeather Team <robot4@openweathermap.org>

받는사람 Customer

2024년 11월 8일 (금) 오후 4:38

⟳ 영어 → 한국어 번역하기

Dear Customer!

Thank you for choosing [OpenWeatherMap!](#)

Please confirm your email address to help us ensure your account is always protected.

[Verify your email](#)

이메일 인증

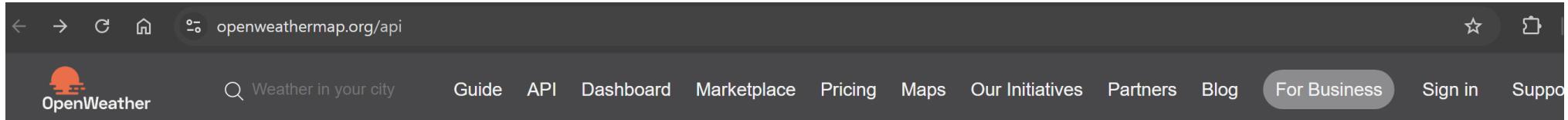
For further technical questions and support, please contact us at [info@openweathermap.org](mailto:info@openweathermap.org)

## Section

### 크롤링 웹사이트 확인

# 크롤링 개요

<https://openweathermap.org/api>



## Weather API

[Home](#) / [Weather API](#)

Please, [sign up](#) to use our fast and easy-to-work weather APIs. As a start to use OpenWeather products, we recommend our [One Call API 3.0](#). For more functionality, please consider our products, which are included in [professional collections](#).

### One Call API 3.0

[API doc](#) [Subscribe](#)

Make an API call to receive access to the various data:

- **Current weather and forecasts:**
  - minute forecast for 1 hour
  - hourly forecast for 48 hours
  - daily forecast for 8 days
- **Weather data for any timestamp** for 45+ years historical archive and 4 days ahead forecast
- **Daily aggregation** of weather data for 45+ years archive and 1.5 years ahead forecast
- **Weather overview** with a human-readable weather summary for today and tomorrow's forecast

Read more about this API and subscription plan in the [FAQ](#).

Pay as you call

**1,000 API calls per day for free**

**0.0012 GBP** per API call over the daily limit

[Subscribe](#)

This is a separate subscription plan, which includes only One Call API.

## Section

### 크롤링 웹사이트 확인

# 크롤링 개요

<https://openweathermap.org/>

The screenshot shows the top navigation bar of the OpenWeatherMap website. It includes a logo with a sun icon, a search bar, and links for Weather in your city, Guide, API, Dashboard, Marketplace, Pricing, Maps, Our Initiatives, Partners, Blog, and For Business.

- Included in the Professional and Enterprise subscription plans
- [Monthly subscription](#). Please [contact us](#) to get access.

## 5 Day / 3 Hour Forecast

구독

[API doc](#) [Subscribe](#)

- 5 day forecast for any location on the globe
- 5 day forecast with a 3-hour step
- JSON and XML formats
- Included in both free and paid subscriptions

## Road Risk API

[Doc](#) [Get a quote](#)

- Specify your route and get weather data and national alerts for the point of destination and along the route
- Current, forecast for 5 days and historical weather data for 1 year for your route
- Weather data are available for any point on the globe
- To receive information on price and get a quote, please [contact us](#)

[Solar Irradiance & Energy Prediction service](#)

## Section

### 크롤링 웹사이트 확인

# 크롤링 개요

<https://openweathermap.org/>

Current weather and forecasts collection

Free	Startup	Developer	Professional	Enterprise
	<b>30 GBP/ month</b> <a href="#">Subscribe</a>	<b>140 GBP/ month</b> <a href="#">Subscribe</a>	<b>370 GBP/ month</b> <a href="#">Subscribe</a>	<b>1500 GBP/ month</b> <a href="#">Subscribe</a>
<b>60 calls/minute 1,000,000 calls/month</b>	<b>600 calls/minute 10,000,000 calls/month</b>	<b>3,000 calls/minute 100,000,000 calls/month</b>	<b>30,000 calls/minute 1,000,000,000 calls/month</b>	<b>200,000 calls/minute 5,000,000,000 calls/month</b>
<b>Current Weather</b>	Current Weather	Current Weather	Current Weather	Current Weather
<b>3-hour Forecast 5 days</b>	3-hour Forecast 5 days	3-hour Forecast 5 days	3-hour Forecast 5 days	3-hour Forecast 5 days
Hourly Forecast 4 days	Hourly Forecast 4 days	<b>Hourly Forecast 4 days</b>	Hourly Forecast 4 days	Hourly Forecast 4 days
Daily Forecast 16 days	<b>Daily Forecast 16 days</b>	Daily Forecast 16 days	Daily Forecast 16 days	Daily Forecast 16 days
Climatic Forecast 30 days	Climatic Forecast 30 days	<b>Climatic Forecast 30 days</b>	Climatic Forecast 30 days	Climatic Forecast 30 days
Bulk Download	Bulk Download	Bulk Download	<b>Bulk Download (global cities)</b>	Bulk Download (global cities + ZIPs of US, EU, UK)

## Section

### 크롤링 웹사이트 확인

# 크롤링 개요

<https://openweathermap.org/>

The screenshot shows the top navigation bar of the OpenWeatherMap website. It includes a logo with a sun icon, a search bar, and links for Weather in your city, Guide, API, Dashboard, Marketplace, Pricing, Maps, Our Initiatives, Partners, Blog, and For Business.

- Included in the Professional and Enterprise subscription plans
- [Monthly subscription](#). Please [contact us](#) to get access.

## 5 Day / 3 Hour Forecast

API 문서  
확인하기

[API doc](#)

[Subscribe](#)

- 5 day forecast for any location on the globe
- 5 day forecast with a 3-hour step
- JSON and XML formats
- Included in both free and paid subscriptions

## Road Risk API

[Doc](#)

[Get a quote](#)

- Specify your route and get weather data and national alerts for the point of destination and along the route
- Current, forecast for 5 days and historical weather data for 1 year for your route
- Weather data are available for any point on the globe
- To receive information on price and get a quote, please [contact us](#)

[Solar Irradiance & Energy Prediction service](#)

## Section

### 크롤링 웹사이트 확인

# 크롤링 개요 <https://openweathermap.org/>

답장 전체답장 전달 | 삭제 스팸차단 안읽음 | 이동▼ 더보기▼

받는 사람 Customer

2024년 11월 8일 (금) 오후 4:41

영어 → 한국어 번역하기

Dear Customer!

Thank you for subscribing to Free [OpenWeatherMap!](#)

## API 키 보관 잘하기

API key:

- Within the next couple of hours, your API key [REDACTED] will be activated and ready to use
- You can later create more API keys on your [account page](#)
- Please, always use your API key in each API call

Endpoint:

- Please, use the endpoint [api.openweathermap.org](https://api.openweathermap.org) for your API calls
- Example of API call:

[api.openweathermap.org/data/2.5/weather?q=London,uk&APPID=b100918669f8c8c3a9f3f109eb9ab47a](https://api.openweathermap.org/data/2.5/weather?q=London,uk&APPID=b100918669f8c8c3a9f3f109eb9ab47a)

Useful links:

- API documentation <https://openweathermap.org/api>
- Details of your plan <https://openweathermap.org/price>
- Please, note that [16-days daily forecast](#) and [History API](#) are not available for Free subscribers

## Section

### 크롤링 웹사이트 확인

# 크롤링 개요

<https://openweathermap.org/>

## Coordinates by location name

### How to make an API call

#### API call

```
http://api.openweathermap.org/geo/1.0/direct?q={city name},{state  
code},{country code}&limit={limit}&appid={API key}
```

#### Parameters

`q` required City name, state code (only for the US) and country code divided by comma. Please use ISO 3166 country codes.

`appid` required Your unique API key (you can always find it on your account page under the "["API key" tab](#))

`limit` optional Number of the locations in the API response (up to 5 results can be returned in the API response)

#### Example of API call

```
http://api.openweathermap.org/geo/1.0/direct?  
q=London&limit=5&appid={API key}
```

## Call 5 day / 3 hour forecast data

### How to make an API call

You can search weather forecast for 5 days with data every 3 hours by geographic coordinates. All weather data can be obtained in JSON and XML formats.

#### API call

```
api.openweathermap.org/data/2.5/forecast?lat={lat}&lon={lon}&appid=  
{API key}
```

#### Parameters

`lat` required Latitude. If you need the geocoder to automatically convert city names and zip-codes to geo coordinates and vice versa, please use our [Geocoding API](#)

`lon` required Longitude. If you need the geocoder to automatically convert city names and zip-codes to geo coordinates and vice versa, please use our [Geocoding API](#)

`appid` required Your unique API key (you can always find it on your account page under the "["API key" tab](#))

`units` optional Units of measurement. `standard`, `metric` and `imperial` units are available. If you do not use the `units` parameter, `standard` units will be applied by default. [Learn more](#)

`mode` optional Response format. JSON format is used by default. To get data in

## Section

### 크롤링 웹사이트 확인

# 날씨 데이터 크롤링

<https://openweathermap.org/>

```
[1]: from bs4 import BeautifulSoup  
from urllib.request import urlopen
```

```
[12]: cityname = "Seoul"  
limit = 3  
apikey = "00000000000000000000000000000000"
```

```
[3]: url_location = "http://api.openweathermap.org/geo/1.0/direct?q={}&limit={}&appid={}".format(cityname, limit, apikey)
```

```
[4]: print(url_location)
```

```
http://api.openweathermap.org/geo/1.0/direct?q=Seoul&limit=3&appid=00000000000000000000000000000000
```

## Section

### 크롤링 웹사이트 확인

# 날씨 데이터 크롤링

<https://openweathermap.org/>

```
[5]: html = urlopen(url_location)
bs_obj = BeautifulSoup(html, "html.parser")
print(bs_obj)

[{"name": "Seoul", "local_names": {"ml": "\uc0ac\uc0ac\ud0d0", "ba": "Сеул", "an": "Se\u00f1l", "mk": "Сеул", "my": "\u209c\u209e\u2091\u209e\u2091", "bs": "Seul", "ky": "Сеул", "de": "Seoul", "vo": "S\u00f6ul", "sk": "Soul", "hr": "Seul", "fa": "\u209e\u2091\u2090\u2091", "ur": "\u209e\u2091\u2090\u2091", "nl": "Seoel", "kn": "\u209e\u2091\u2090\u2091", "az": "Seul", "mr": "\u209e\u2091\u2090\u2091", "pl": "Seul", "ku": "Se\u00f1l", "th": "\u209e\u2091\u2090\u2091", "eo": "Seulo", "eu": "Seu1", "cv": "Сеул", "uk": "Сеул", "qu": "Siul", "sr": "Сеул", "it": "Seul", "ro": "Seul", "hy": "\u209e\u2091\u2090\u2091", "os": "Сеул", "el": "\u209e\u2091\u2090\u2091", "ar": "\u209e\u2091\u2090\u2091", "tg": "Сеул", "ta": "\u209e\u2091\u2090\u2091", "bg": "Сеул", "ko": "\u209e\u2091\u2090\u2091", "ka": "\u209e\u2091\u2090\u2091", "tk": "Seul", "kk": "Сеул", "hu": "Sz\u00f6ul", "lt": "Seulas", "vi": "Seoul", "ca": "Se\u00f1l", "be": "Сеул", "uz": "Seu1", "bn": "\u209e\u2091\u2090\u2091", "mn": "Сеул", "tr": "Seul", "en": "Seoul", "oc": "Seol", "gl": "Se\u00f1l", "he": "\u209e\u2091\u2090\u2091", "es": "Se\u00f1l", "hi": "Сиёл", "fi": "Soul", "is": "Se\u00f1l", "lv": "Seul", "km": "\u209e\u2091\u2090\u2091", "ru": "Сеул", "bo": "\u209e\u2091\u2090\u2091", "af": "Seoel", "bh": "Сиёл", "et": "S\u00f6ul", "sl": "Seul", "yi": "\u209e\u2091\u2090\u2091", "ja": "\u209e\u2091\u2090\u2091", "zh": "\u209e\u2091\u2090\u2091 / \u209e\u2091\u2090\u2091", "fr": "S\u00e9oul", "la": "Seulum", "cs": "Soul", "am": "\u209e\u2091\u2090\u2091", "sv": "Seoul", "pt": "Seul"}, "lat": 37.5666791, "lon": 126.9782914, "country": "KR"}]

[44]: bs_obj.text

[44]: '[{"name": "Seoul", "local_names": {"ml": "\uc0ac\uc0ac\ud0d0", "ba": "Сеул", "an": "Se\u00f1l", "mk": "Сеул", "my": "\u209c\u209e\u2091\u209e\u2091", "bs": "Seul", "ky": "Сеул", "de": "Seoul", "vo": "S\u00f6ul", "sk": "Soul", "hr": "Seul", "fa": "\u209e\u2091\u2090\u2091", "ur": "\u209e\u2091\u2090\u2091", "nl": "Seoel", "kn": "\u209e\u2091\u2090\u2091", "az": "Seul", "mr": "\u209e\u2091\u2090\u2091", "pl": "Seul", "ku": "Se\u00f1l", "th": "\u209e\u2091\u2090\u2091", "eo": "Seulo", "eu": "Seu1", "cv": "Сеул", "uk": "Сеул", "qu": "Siul", "sr": "Сеул", "it": "Seul", "ro": "Seul", "hy": "\u209e\u2091\u2090\u2091", "os": "Сеул", "el": "\u209e\u2091\u2090\u2091", "ar": "\u209e\u2091\u2090\u2091", "tg": "Сеул", "ta": "\u209e\u2091\u2090\u2091", "bg": "Сеул", "ko": "\u209e\u2091\u2090\u2091", "ka": "\u209e\u2091\u2090\u2091", "tk": "Seul", "kk": "Сеул", "hu": "Sz\u00f6ul", "lt": "Seulas", "vi": "Seoul", "ca": "Se\u00f1l", "be": "Сеул", "uz": "Seu1", "bn": "\u209e\u2091\u2090\u2091", "mn": "Сеул", "tr": "Seul", "en": "Seoul", "oc": "Seol", "gl": "Se\u00f1l", "he": "\u209e\u2091\u2090\u2091", "es": "Se\u00f1l", "hi": "Сиёл", "fi": "Soul", "is": "Se\u00f1l", "lv": "Seul", "km": "\u209e\u2091\u2090\u2091", "ru": "Сеул", "bo": "\u209e\u2091\u2090\u2091", "af": "Seoel", "bh": "Сиёл", "et": "S\u00f6ul", "sl": "Seul", "yi": "\u209e\u2091\u2090\u2091", "ja": "\u209e\u2091\u2090\u2091", "zh": "\u209e\u2091\u2090\u2091 / \u209e\u2091\u2090\u2091", "fr": "S\u00e9oul", "la": "Seulum", "cs": "Soul", "am": "\u209e\u2091\u2090\u2091", "sv": "Seoul", "pt": "Seul"}, "lat": 37.5666791, "lon": 126.9782914, "country": "KR"}]'

[45]: len(bs_obj.text)

[45]: 988
```

## Section

### 크롤링 웹사이트 확인

# 날씨 데이터 크롤링

<https://openweathermap.org/>

```
[46]: import json
```

```
[52]: loc_list = json.loads(bs_obj.text)
```

```
[53]: print(loc_list)
```

```
[{'name': 'Seoul', 'local_names': {'ml': '서울', 'ba': 'Сеул', 'an': 'Seúl', 'mk': 'Сеул', 'my': 'ဆိုးလှို့', 'bs': 'Seul', 'ky': 'Сеул', 'de': 'Seoul', 'vo': 'Söul', 'sk': 'Soul', 'hr': 'Seul', 'fa': 'سئول', 'ur': 'سُؤل', 'nl': 'Seoel', 'kn': 'ሰዕል', 'az': 'Seul', 'mr': 'सौल', 'pl': 'Seul', 'ku': 'Sêûl', 'th': 'โซล', 'eo': 'Seulo', 'eu': 'Seul', 'cv': 'Сеул', 'uk': 'Сеул', 'qu': 'Siul', 'sr': 'Сеул', 'it': 'Seul', 'ro': 'Seul', 'hy': 'Սեուլ', 'os': 'Сеул', 'el': 'Σεούλ', 'ar': 'سول', 'tg': 'Сеул', 'ta': 'சிவேப்பல்', 'bg': 'Сеул', 'ko': '서울', 'ka': 'სეული', 'tk': 'Seul', 'kk': 'Сеул', 'hu': 'Szöul', 'lt': 'Seulas', 'vi': 'Seoul', 'ca': 'Seül', 'be': 'Сеул', 'uz': 'Seul', 'bn': 'সিউল', 'mn': 'Сөүл', 'tr': 'Seul', 'en': 'Seoul', 'oc': 'Seol', 'gl': 'Seúl', 'he': 'סּוּל', 'es': 'Seúl', 'hi': 'सियोल', 'fi': 'Soul', 'is': 'Seúl', 'lv': 'Seula', 'km': 'ស៊ូល', 'ru': 'Сеул', 'bo': 'សោមុខ', 'af': 'Seoeul', 'bh': 'సియోల', 'et': 'Söul', 'sl': 'Seul', 'yi': 'שְׁוּל', 'ja': 'ソウル', 'zh': '首尔市 / 首爾', 'fr': 'Séoul', 'la': 'Seulum', 'cs': 'Soul', 'am': 'ሰል', 'sv': 'Seoul', 'pt': 'Seul'}, 'lat': 37.5666791, 'lon': 126.9782914, 'country': 'KR'}]
```

```
[54]: type(loc_list)
```

```
[54]: list
```

```
[55]: len(loc_list)
```

```
[55]: 1
```

## Section

### 크롤링 웹사이트 확인

# 날씨 데이터 크롤링 <https://openweathermap.org/>

```
[57]: print(loc_list[0])
```

```
{'name': 'Seoul', 'local_names': {'ml': '서울', 'ba': 'Ceyl', 'an': 'Seúl', 'mk': 'Сеул', 'vo': 'Söul', 'sk': 'Soul', 'hr': 'Seul', 'fa': 'سئول', 'ur': 'سُؤل', 'nl': 'Seoel', 'kn': 'ලු', 'th': 'โซล', 'eo': 'Seulo', 'eu': 'Seul', 'cv': 'Сеул', 'uk': 'Сеул', 'qu': 'Siul', 's': 'Сеул', 'el': 'Σεούλ', 'ar': 'سول', 'tg': 'Сеул', 'ta': 'செஷபல்', 'bg': 'Сеул', 'ko': '서울', 'lt': 'Seulas', 'vi': 'Seoul', 'ca': 'Seül', 'be': 'Сеул', 'uz': 'Seul', 'bn': 'সিউল', 'l': 'Seúl', 'he': 'סְוָל', 'es': 'Seúl', 'hi': 'सियोल', 'fi': 'Soul', 'is': 'Seúl', 'lv': 'l', 'bh': 'সিয়োল', 'et': 'Söul', 'sl': 'Seul', 'yi': 'זְאוּל', 'ja': 'ソウル', 'zh': '首尔市', 'a': 'sv': 'Seoul', 'pt': 'Seul'}, 'lat': 37.5666791, 'lon': 126.9782914, 'country': 'KR'}
```

```
[58]: print(loc_list[1])
```

```
-----  
IndexError                                     Traceback (most recent call last)  
Cell In[58], line 1  
----> 1 print(loc_list[1])  
  
IndexError: list index out of range
```

```
[59]: type(loc_list[0])
```

```
[59]: dict
```

```
[60]: loc_list[0]['lat']
```

```
[60]: 37.5666791
```

```
[62]: loc_list[0]['lon']
```

```
[62]: 126.9782914
```

## Section

### 크롤링 웹사이트 확인

# 날씨 데이터 크롤링

<https://openweathermap.org/>

```
[63]: lat = loc_list[0]['lat']
lon = loc_list[0]['lon']

[64]: url_history = "http://api.openweathermap.org/data/2.5/forecast?lat={}&lon={}&appid={}".format(lat, lon, apikey)

[65]: html2 = urlopen(url_history)
bs_obj2 = BeautifulSoup(html2, "html.parser")
print(bs_obj2)

{"cod": "200", "message": 0, "cnt": 40, "list": [{"dt": 1731218400, "main": {"temp": 292.38, "feels_like": 291.66, "temp_min": 291.66, "temp_max": 292.38, "pressure": 1022, "humidity": 50, "sea_level": 1022, "grnd_level": 1015}, "weather": [{"id": 801, "main": "Clouds", "description": "few clouds", "icon": "02d"}, {"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 24}, "wind": {"speed": 1.75, "deg": 307, "gust": 2.18}, "visibility": 10000, "pop": 0, "sys": {"pod": "d"}, "dt_txt": "2024-11-10 09:00:00"}, {"dt": 1731240000, "main": {"temp": 288.51, "feels_like": 287.51, "temp_min": 287.51, "temp_max": 288.51, "pressure": 1021, "humidity": 50, "sea_level": 1021, "grnd_level": 1015}, "weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 2}, "wind": {"speed": 1.15, "deg": 319, "gust": 1.45}, "visibility": 10000, "pop": 0, "sys": {"pod": "n"}, "dt_txt": "2024-11-10 12:00:00"}, {"dt": 1731250800, "main": {"temp": 287.44, "feels_like": 286.2, "temp_min": 287.44, "temp_max": 287.44, "pressure": 1020, "humidity": 49, "sea_level": 1020, "grnd_level": 1015}, "weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 1}, "wind": {"speed": 1.25}, "visibility": 10000, "pop": 0, "sys": {"pod": "n"}, "dt_txt": "2024-11-10 15:00:00"}, {"dt": 1731261600, "main": {"temp": 286.51, "feels_like": 285.51, "temp_min": 286.51, "temp_max": 286.51, "pressure": 1022, "humidity": 49, "sea_level": 1022, "grnd_level": 1015}, "weather": [{"id": 800, "main": "Clear", "description": "clear sky", "icon": "01n"}], "clouds": {"all": 1}, "wind": {"speed": 1.15, "deg": 319, "gust": 1.45}, "visibility": 10000, "pop": 0, "sys": {"pod": "n"}, "dt_txt": "2024-11-10 18:00:00"}]}
```

## Section

### 크롤링 웹사이트 확인

# 날씨 데이터 크롤링

<https://openweathermap.org/>



json 뷰어



전체

이미지

동영상

쇼핑

뉴스

도서

웹

: 더보기

도구



Chrome Web Store

<https://chromewebstore.google.com/detail/json-viewer> :

## JSON Viewer - Chrome 웹 스토어

세부정보 · 버전. 0.18.1 · 업데이트됨. 2020년 12월 23일 · 우려사항 신고 · 제공. tulios · 크기. 310KiB

· 언어. English (United States) · 개발자. Tulio Ornelas



Online JSON Viewer

<https://jsonviewer.stack.hu> :

## Online JSON Viewer and Formatter

JSON, short for JavaScript Object Notation, is a lightweight computer data interchange format.

JSON is a text-based, human-readable format for representing ...

## Section

### 크롤링 웹사이트 확인

# 날씨 데이터 크롤링

<https://jsonviewer.stack.hu/>

The screenshot shows a browser window displaying a JSON response from the URL https://jsonviewer.stack.hu/. The interface has two main sections: a tree view on the left and a table view on the right.

**Tree View (Viewer tab):**

- message : 0
- cnt : 40
- list
  - 0
    - dt : 1731218400
    - main
    - weather
      - 0
        - id : 801
        - main : "Clouds"
        - description : "few clouds"
        - icon : "02d"
    - clouds
      - all : 24
    - wind
      - speed : 1.75
      - deg : 307
      - gust : 2.18
    - visibility : 10000
    - pop : 0
  - sys
    - dt\_txt : "2024-11-10 06:00:00"
  - 1
  - 2
  - 3

**Table View:**

Name	Value
city	...
cnt	40
cod	"200"
list	...
message	0

## Section

### 크롤링 웹사이트 확인

# 날씨 데이터 크롤링

<https://openweathermap.org/>

```
[66]: weather = json.loads(bs_obj2.text)
```

```
[70]: type(weather)
```

```
[70]: dict
```

```
[73]: weather['list'][0]
```

```
[73]: {'dt': 1731218400,
       'main': {'temp': 292.38,
                 'feels_like': 291.66,
                 'temp_min': 291.37,
                 'temp_max': 292.38,
                 'pressure': 1022,
                 'sea_level': 1022,
                 'grnd_level': 1015,
                 'humidity': 50,
                 'temp_kf': 1.01},
       'weather': [{"id": 801,
                    'main': 'Clouds',
                    'description': 'few clouds',
                    'icon': '02d'}],
       'clouds': {'all': 24},
       'wind': {'speed': 1.75, 'deg': 307, 'gust': 2.18},
       'visibility': 10000,
       'pop': 0,
       'sys': {'pod': 'd'},
       'dt_txt': '2024-11-10 06:00:00'}
```

```
[74]: weather['list'][0]['main']
```

```
[74]: {'temp': 292.38,
       'feels_like': 291.66,
       'temp_min': 291.37,
       'temp_max': 292.38,
       'pressure': 1022,
       'sea_level': 1022,
       'grnd_level': 1015,
       'humidity': 50,
       'temp_kf': 1.01}
```

```
[75]: weather['list'][0]['main']['temp']
```

```
[75]: 292.38
```

```
[76]: len(weather['list'])
```

```
[76]: 40
```

## Section

### 크롤링 웹사이트 확인

# 날씨 데이터 크롤링

<https://openweathermap.org/>

```
[27]: temp_list = []
n = len(weather['list'])
for i in range(0, n):
    temp_value = weather['list'][i]['main']['temp']
    temp_list.append(temp_value)
```

```
[28]: print(temp_list)
```

```
[304.32, 302.38, 299.38, 297.83, 298.13, 301.66, 304.6, 305.19, 303.07, 300.9, 299.79, 298.86, 298.54, 302.09, 306.47, 307.79, 305.98, 302.84, 301.31, 300.18, 299.33, 303.4, 308.48, 310.33, 308.42, 304.83, 303.36, 302.03, 300.87, 304.58, 309.09, 311.14, 307.5, 304.92, 303.35, 302.52, 300.82, 304.31, 309.61, 309.7]
```

## Section

### 크롤링 웹사이트 확인

# 날씨 데이터 크롤링

<https://openweathermap.org/>

```
[29]: weather['list'][0]['dt_txt']
```

```
[29]: '2025-07-21 09:00:00'
```

```
[30]: dt_list = []
for i in range(0, n):
    dt_value = weather['list'][i]['dt_txt']
    dt_list.append(dt_value)
```

```
[31]: print(dt_list)
```

```
['2025-07-21 09:00:00', '2025-07-21 12:00:00', '2025-07-21 15:00:00', '2025-07-21 18:00:00', '2025-07-21 21:00:00', '2025-07-22 00:00:00', '2025-07-22 03:00:00', '2025-07-22 06:00:00', '2025-07-22 09:00:00', '2025-07-22 12:00:00', '2025-07-22 15:00:00', '2025-07-22 18:00:00', '2025-07-22 21:00:00', '2025-07-23 00:00:00', '2025-07-23 03:00:00', '2025-07-23 06:00:00', '2025-07-23 09:00:00', '2025-07-23 12:00:00', '2025-07-23 15:00:00', '2025-07-23 18:00:00', '2025-07-23 21:00:00', '2025-07-24 00:00:00', '2025-07-24 03:00:00', '2025-07-24 06:00:00', '2025-07-24 09:00:00', '2025-07-24 12:00:00', '2025-07-24 15:00:00', '2025-07-24 18:00:00', '2025-07-24 21:00:00', '2025-07-25 00:00:00', '2025-07-25 03:00:00', '2025-07-25 06:00:00', '2025-07-25 09:00:00', '2025-07-25 12:00:00', '2025-07-25 15:00:00', '2025-07-25 18:00:00', '2025-07-25 21:00:00', '2025-07-26 00:00:00']
```

## Section

### 크롤링 웹사이트 확인

# 날씨 데이터 크롤링

<https://openweathermap.org/>

```
[83]: import matplotlib.pyplot as plt
```

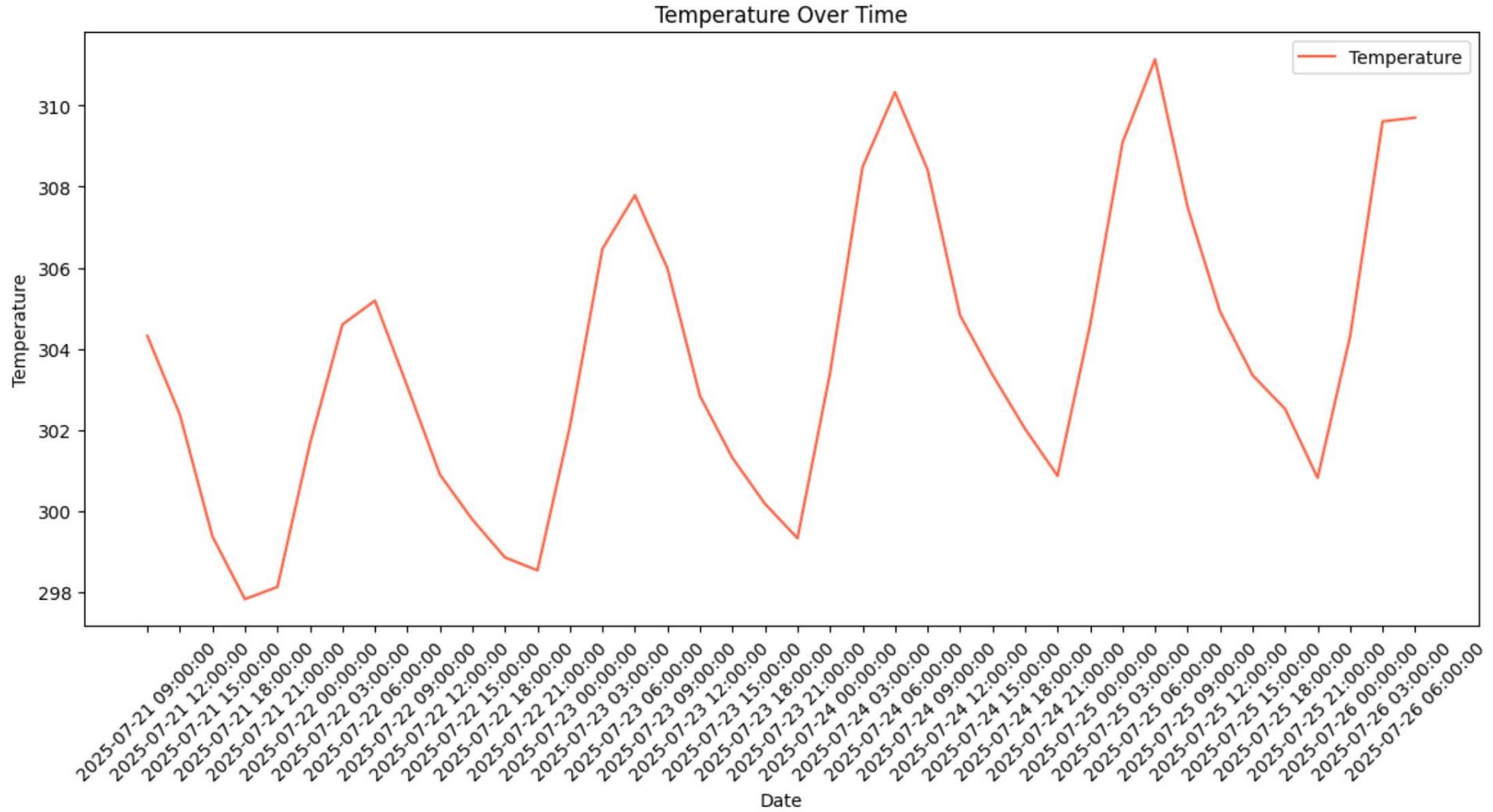
```
[89]: plt.figure(figsize=(14, 6))
plt.plot(dt_list, temp_list, color='tomato', label='Temperature')
plt.xlabel('Date')
plt.ylabel('Temperature')
plt.title('Temperature Over Time')
plt.xticks(rotation=45)
plt.legend()
plt.show()
```

## Section

### 크롤링 웹사이트 확인

# 날씨 데이터 크롤링

<https://openweathermap.org/>



감사합니다.

# Q & A