

# 파이썬 프로그래밍

## 23. Python 활용 심화 (2)

## ❖ 수업 목표

- JSON 데이터 형식을 이해하고 파이썬에서 사용할 수 있다.
- API를 활용하여 날씨 데이터를 가져올 수 있다.
- 수집한 데이터를 활용하여 시각화 할 수 있다.
- 텍스트 데이터에서 이메일과 전화번호를 추출할 수 있다.
- 로그 파일에서 오류 메시지를 추출 할 수 있다.

## ❖ 세부 목표

- 23.1 JSON과 API 활용 기초
- 23.2 API 데이터 활용
- 23.3 정규 표현식 활용1
- 23.4 정규 표현식 활용2
- 23.5 데이터 통합 및 보고서 자동화

## ❖ JSON 개요

### ■ JSON(JavaScript Object Notation) 소개

- JSON은 데이터를 저장하고 전송하는 데 사용하는 경량 데이터 형식
- 사람과 기계가 모두 읽기 쉬운 텍스트 기반 구조
- 주요 데이터 구조: 객체(딕셔너리)와 배열(리스트)

### ■ JSON 예시

- sample.json
  - {
  - "name": "Alice",
  - "age": 25,
  - "city": "Seoul",
  - "skills": ["Python", "Django", "Machine Learning"]
  - }

## ❖ JSON 개요

### ■ JSON 문자열을 파이썬 객체로 변환

- 예)
  - `import json`
  - `json_string = '{"name": "Alice", "age": 25, "city": "Seoul"}'`
  - `# JSON 문자열 -> 딕셔너리 변환`
  - `data = json.loads(json_string)`
  - `print(data["name"])`                      `# Alice 출력`

### ■ 파이썬 객체를 JSON 문자열로 변환

- `python_dict = {"name": "Bob", "age": 30, "city": "Busan"}`
- `# JSON 포맷 변환`
- `json_output = json.dumps(python_dict, indent=4)`
- `print(json_output)`

## ❖ JSON 개요

### ■ JSON 파일 입출력

#### • JSON 파일 저장

- with open("data.json", "w") as file:
- json.dump(python\_dict, file, indent=4)

#### • JSON 파일 불러오기

- with open("data.json", "r") as file:
- loaded\_data = json.load(file)
- print(loaded\_data)

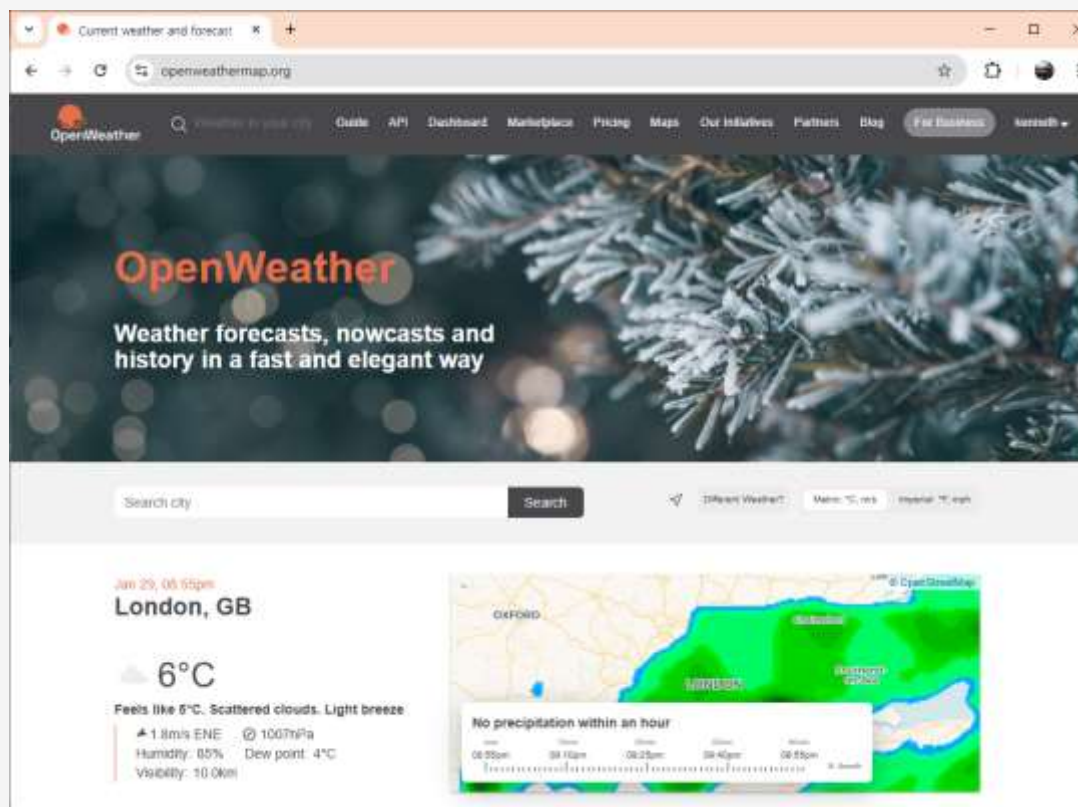
# 1. JSON과 API 활용 기초

## ❖ API 기초

- API(Application Programming Interface) 소개
  - 프로그램 간 데이터와 기능을 주고받을 수 있도록 하는 인터페이스
  - REST API: HTTP 프로토콜을 기반으로 JSON 형태의 데이터를 주고받는 방식

## ❖ API 활용

- 웹 API 사용하기
  - OpenWeatherMap 날씨 정보
    - 홈페이지
      - » <https://openweathermap.org/>



# 1. JSON과 API 활용 기초

## ❖ API 활용

### ■ 웹 API 사용하기

- 사용자 등록
  - openweathermap > Sign in > Create an Account
    - » [https://home.openweathermap.org/users/sign\\_up](https://home.openweathermap.org/users/sign_up)

Members

home.openw...

OpenWeather

### 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](#)

I consent to receive communications from OpenWeather Group of Companies and their partners:

☐ System news (API usage alert, system update, temporary system shutdown, etc)

☐ Product news (change to price, new product features, etc)

☐ Corporate news (our life, the launch of a new service, etc)

☐ I'm not a robot

reCAPTCHA

Windows 제품 인증

Create Account 이동을 위해 Windows를 인증합니다.

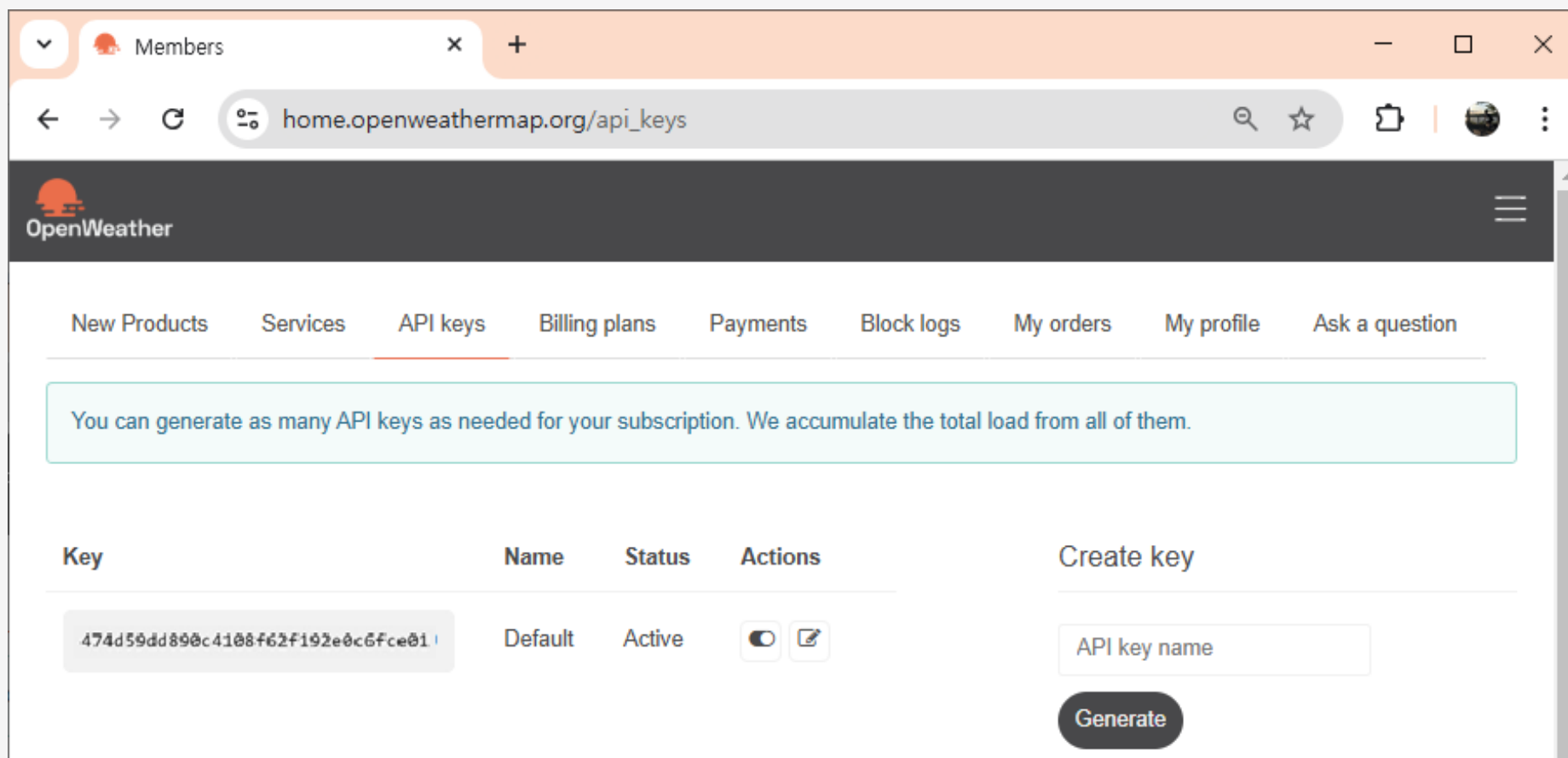
오전 6:02  
2025-01-30

# 1. JSON과 API 활용 기초

## ❖ API 활용

### ■ 웹 API 사용하기

- [API keys] 탭에서 API 키 확인
  - Key는 32자 길이의 알파벳과 숫자 조합





# 1. JSON과 API 활용 기초

## ❖ API 활용

### ■ 웹 API 사용하기

#### • 예) 서울 날씨 확인하기

- `import requests`
- `import json`
  
- `# 1. API 키를 지정합니다. 자신의 키로 변경해서 사용`
- `apikey = "3dbd030dd02ad7cbf24ef298fa0c6d00"`
  
- `# 2. 날씨를 확인할 도시 지정하기`
- `city = "Seoul,KR"`
  
- `# 3. API 지정`
- `api = "http://api.openweathermap.org/\`
- `data/2.5/weather?q={city}&APPID={key}"`
  
- `# 4. 켈빈 온도를 섭씨 온도로 변환하는 함수`
- `k2c = lambda k: k - 273.15`

## ❖ API 활용

### ■ 웹 API 사용하기

- 예) 서울 날씨 확인하기
  - # 5. 도시 정보 추출하기
  - # 6. API의 URL 구성하기
  - `url = api.format(city=city, key=apikey)`
  - # API에 요청을 보내 데이터 추출하기
  - `r = requests.get(url)`
  - # 7. 결과를 JSON 형식으로 변환하기
  - `data = json.loads(r.text)`

## ❖ API 활용

### ■ 웹 API 사용하기

#### • 예) 서울 날씨 확인하기

- # 8. 결과 출력하기
- `print("+ 도시 =", data["name"])`
- `print("| 날씨 =", data["weather"][0]["description"])`
- `print("| 최저 기온 =", k2c(data["main"]["temp_min"]))`
- `print("| 최고 기온 =", k2c(data["main"]["temp_max"]))`
- `print("| 습도 =", data["main"]["humidity"])`
- `print("| 기압 =", data["main"]["pressure"])`
- `print("| 풍향 =", data["wind"]["deg"])`
- `print("| 풍속 =", data["wind"]["speed"])`
- `print("")`

# 1. JSON과 API 활용 기초

## ❖ API 활용

### ■ 웹 API 사용하기

- 예) 서울 날씨 확인하기
  - 실행 결과
    - » + 도시 = Seoul
    - » | 날씨 = clear sky
    - » | 최저 기온 = -7.239999999999952
    - » | 최고 기온 = -6.21999999999997
    - » | 습도 = 68
    - » | 기압 = 1029
    - » | 풍향 = 310
    - » | 풍속 = 3.09

### ❖ API 활용

#### ■ 날씨 데이터 그래프로 출력하기

- 여러 도시들의 최저 온도를 산점도로 표시하기
- 소스 코드 수정(추가)
  - `import matplotlib.pyplot as plt`
  - `plt.rcParams['font.family'] = 'Malgun Gothic'` # 한글 표시
  - `plt.rcParams['axes.unicode_minus'] = False` # 음수('-') 표시
- `cities = ["Seoul", "New York", "Tokyo", "Sydney"]`
- # 데이터를 저장할 리스트
- `temp_min_list = []`
- `temp_max_list = []`
- `temp_min_list.append(k2c(data["main"]["temp_min"]))`
- `temp_max_list.append(k2c(data["main"]["temp_max"]))`

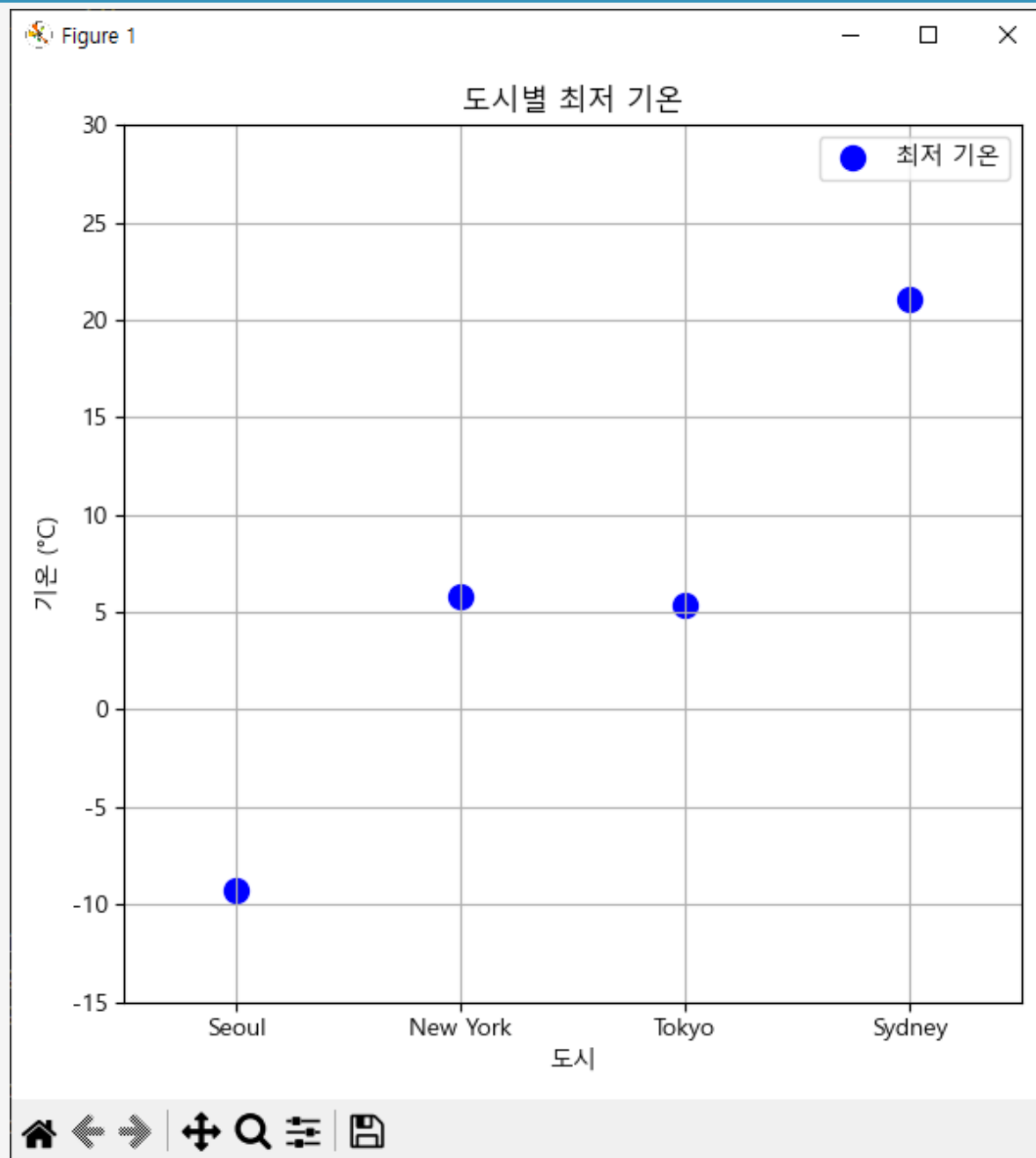
### ❖ API 활용

- 소스 코드 수정(추가)
  - # 최저 기온 산점도
  - `plt.scatter(cities, temp_min_list, color='blue',`
  - `label='최저 기온', s=100, marker='o')`
  - `plt.title('도시별 최저 기온')`
  - `plt.xlabel('도시')`
  - `plt.ylabel('기온 (° C)')`
  - `plt.ylim(-15, 30)` # y축 범위 설정
  - `plt.xlim(-0.5, len(cities) - 0.5)` # x축 여백 추가
  - `plt.grid(True)`
  - `plt.legend()`
- # 그래프 표시
- `plt.tight_layout()`
- `plt.show()`

## 2. API 데이터 활용

### ❖ API 활용

- 실행 결과



### 3. 정규 표현식 활용1

#### ❖ 정규 표현식 활용

- 텍스트 데이터에서 이메일 주소나 전화번호 추출 프로그램
  - 이메일과 전화번호의 패턴 정의
    - 이메일: 일반적으로 `username@domain.com` 형식
    - 전화번호: `010-1234-5678` 또는 `02-123-4567` 등의 형식



### 3. 정규 표현식 활용1

#### ❖ 실습 문제

- 텍스트 데이터에서 이메일 주소나 전화번호 추출 프로그램
  - 주어진 다음 텍스트에서 re 모듈을 사용하여 이메일과 전화번호를 추출하여 출력하는 프로그램을 작성하시오.

## 4. 정규 표현식 활용2

### ❖ 정규 표현식 프로젝트

#### ■ 로그 파일에서 오류 메시지만 추출하는 프로그램

- 로그 파일이란?

- 프로그램 실행 중 발생하는 이벤트, 오류, 정보 등을 기록한 파일
- 일반적으로 .log 확장자를 가짐

- 오류 메시지 패턴 정의

- 로그 파일에서 오류 메시지는 "ERROR" 또는 "Exception" 등의 키워드를 포함

## 4. 정규 표현식 활용2

### ❖ 실습 문제

- 로그 파일에서 오류 메시지만 추출하는 프로그램
  - 다음과 같은 텍스트가 포함된 파일(sample.log)에서 re 모듈을 사용하여 오류 메시지만 추출하여 출력하는 프로그램을 작성하시오.
- sample.log
  - [2024-01-27 10:15:32] INFO - 시스템이 정상적으로 시작되었습니다.
  - [2024-01-27 10:16:05] WARNING - 메모리 사용량이 80%를 초과했습니다.
  - [2024-01-27 10:17:12] ERROR - 데이터베이스 연결 실패. 재시도 중...
  - [2024-01-27 10:18:45] INFO - 사용자 로그인: user123
  - [2024-01-27 10:20:08] ERROR - 파일을 찾을 수 없습니다: config.yaml
  - [2024-01-27 10:21:30] DEBUG - 네트워크 요청: GET /api/data
  - [2024-01-27 10:23:14] EXCEPTION - NullPointerException 발생 (line 42)
  - [2024-01-27 10:24:55] INFO - 백업 작업이 완료되었습니다.
  - [2024-01-27 10:26:30] ERROR - 서버 응답 시간 초과.

## 5. 데이터 통합 및 보고서 자동화

### ❖ 데이터 통합 및 보고서 자동화

#### ■ 그래프를 PDF로 저장하기 위한 준비

- pdf 저장 라이브러리(모듈, 클래스) 활용
  - # PDF로 그래프를 저장하는 기능 제공
  - `from matplotlib.backends.backend_pdf import PdfPages`
  - `with PdfPages(file) as pdf:`
  - # 그래프를 PDF로 저장
  - `pdf.savefig()`     # 현재까지 그려진 그래프를 저장
  - `plt.show()`        # 이미지를 표시하고 그래프 창을 닫아서 메모리 해제

## 5. 데이터 통합 및 보고서 자동화

### ❖ 실습 문제

- 정리된 데이터를 시각화 후 PDF로 저장하는 프로그램
  - 앞서 작성한 날짜 데이터를 시각화한 그래프를 pdf로 저장하는 프로그램을 작성하시오.

## ❖ 과제

- 1. json 문자열을 파이썬 객체로 변환하는 코드 작성하기
- 2. API를 활용하여 데이터를 가져오는 코드 작성하기
- 3. 정규표현식을 활용하여 원하는 패턴의 문자열을 추출하기
- 4. 데이터를 활용하여 시각화 코드 작성하기

## ❖ 다음 수업 내용

- DevOps 및 DBMS
  - DevOps
  - DBMS(SQL)