

# 삼성청년 SW·AI아카데미

관통 프로젝트 01

# 관통 프로젝트 소개

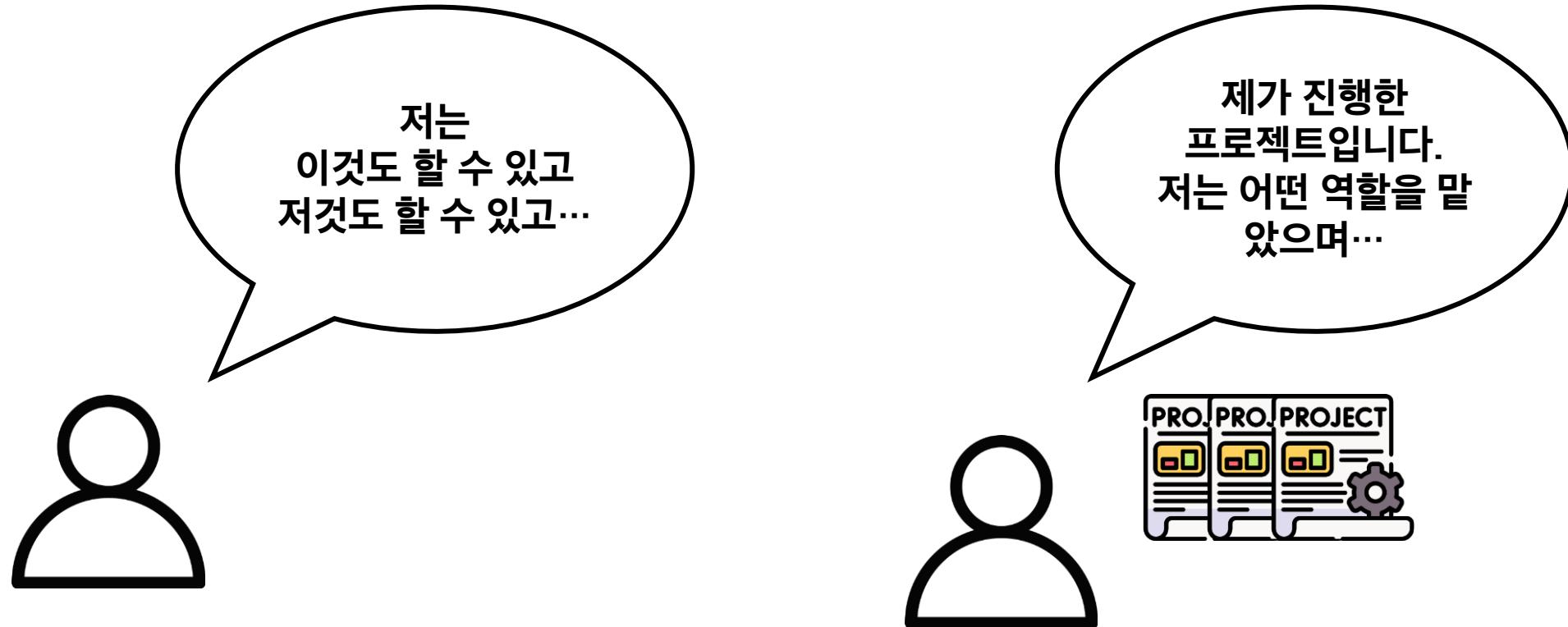
## 관통 프로젝트란?

- 강의 시간에 배운 내용을 모두 포함(관통)
- 추가적인 내용 학습
- 프로젝트 도전!



## 관통 프로젝트를 열심히 해야 하는 이유

- 어느 지원자가 더 설득력 있나요 ?



## | 진행 방식

1. 프로젝트 목표 소개
2. 이론 설명
3. 강사님들과 함께 실습 진행
4. 도전 과제를 스스로 진행

## | 도전 과제는 세 개의 버전이 존재

- 버전 1: 금융 상품 비교 앱
  - PJT01 --> PJT02 --> PJT03 --> PJT04 --> PJT05 --> … > PJT09 --> PJT10
- 버전 2: 영화 추천 서비스
  - PJT01 --> PJT02 --> PJT03 --> PJT04 --> PJT05 --> … > PJT09 --> PJT10
- 버전 3: 도서 정보 검색 서비스
  - PJT01 --> PJT02 --> PJT03 --> PJT04 --> PJT05 --> … > PJT09 --> PJT10
- \* 여러분들은 선택할 수 기회가 있는 행운의 기수 \*

금융상품비교

영화추천서비스

도서정보검색서비스

## 버전1(금융 상품 비교 앱) 최종 프로젝트 소개

- SSAFY 10기부터 적용된 PJT

The screenshot shows the MyBank app's product comparison feature. On the left, there are filters for 'Interest Type' (Term Deposit), 'Interest Rate' (15.4%), 'Term Type' (Term), and 'Term Length' (6 months). The main area displays a table of 5 products:

순위	6개월	12개월	세전이자	세후이자	24개월	36개월	금융기관	상품	UPDATE
1	-	<b>5.02</b>	502,000원	424,692원	-	-	MG 연희	MG더뱅킹정기예금	07.13
2	-	<b>5.00</b>	500,000원	423,000원	-	-	MG 안양만안	정기예탁금	07.10
3	-	<b>5.00</b>	500,000원	423,000원	-	-	MG 강북	MG더뱅킹정기예금	07.12
4	3.50	<b>4.90</b>	490,000원	414,540원	-	-	MG 복현	정기예탁금	07.10
5	-	<b>4.90</b>	490,000원	414,540원	-	-	MG 금호	MG더뱅킹정기예금	07.13

마이뱅크

The screenshot shows the Bankbook app's user profile page for 'test'. It includes a navigation bar with 'test' and '로그아웃' (Logout). The main content area has tabs for '예금비교' (Deposit Comparison), '환율계산기' (Currency Converter), '은행 지도' (Bank Map), and '게시판' (Board). Below this is a section titled 'test 님의 프로필 페이지' (test's profile page).

In the bottom right corner, there is a large callout box titled '상품 추천 받기' (Receive Product Recommendation) which lists 15 recommended products from various banks.

Callout Box Content:

- 1 : (정기예금)제주은행 - 제주Dream 정기예금 (개인/만기 지급식)
- 2 : (정기예금)광주은행 - 월운박스예금
- 3 : (정기예금)충소기업은행 - i-ONE놀이터예금
- 4 : (정기예금)부산은행 - 더(The) 특한 정기예금
- 5 : (정기예금)대구은행 - DGB복합파트너예금(일반형)
- 6 : (정기예금)수협은행 - SH평생주택우대예금 (만기일시지급식)
- 7 : (정기예금)수협은행 - J정기예금 (만기지급식)
- 8 : (정기예금)수협은행 - 해이(Hey)정기예금
- 9 : (정기예금)국민은행 - KB Star 정기예금
- 10 : (정기예금)충소기업은행 - IBK평생한가족동창(실세금리정기예금)
- 11 : (정기예금)수협은행 - Sh첫만남우대예금
- 12 : (정기예금)우리은행 - BNK플러스예금
- 13 : (정기예금)주식회사 카카오뱅크 - 카카오뱅크 정기예금
- 14 : (정기예금)경남은행 - BNK주거래우대정기예금
- 15 : (정기예금)경남은행 - BNK주거래우대정기예금

Bankbook

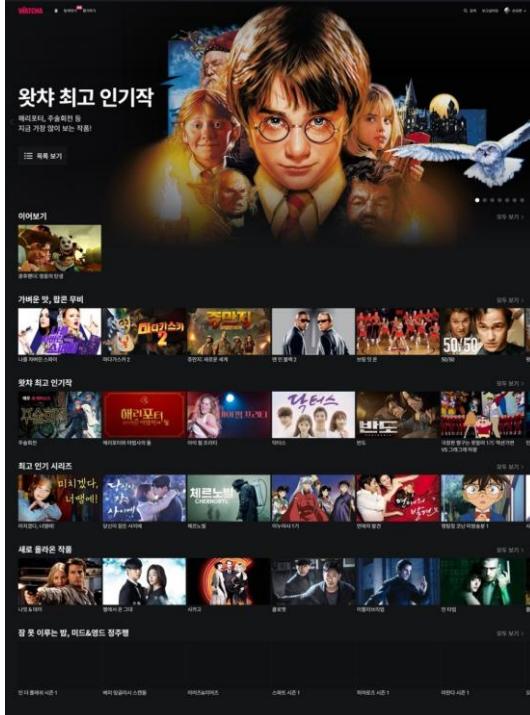
금융상품비교

영화추천서비스

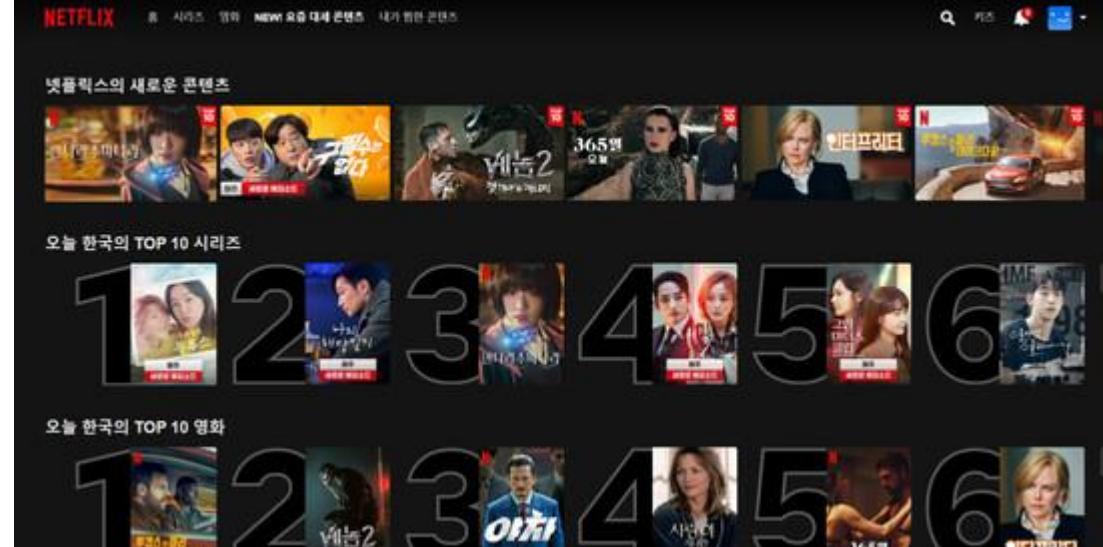
도서정보검색서비스

## 버전2(영화 추천 서비스) 최종 프로젝트 소개

- SSAFY 트랙 초기부터 진행하고 있는 유래 깊은 PJT



왓챠



넷플릭스

금융상품비교

영화추천서비스

도서정보검색서비스

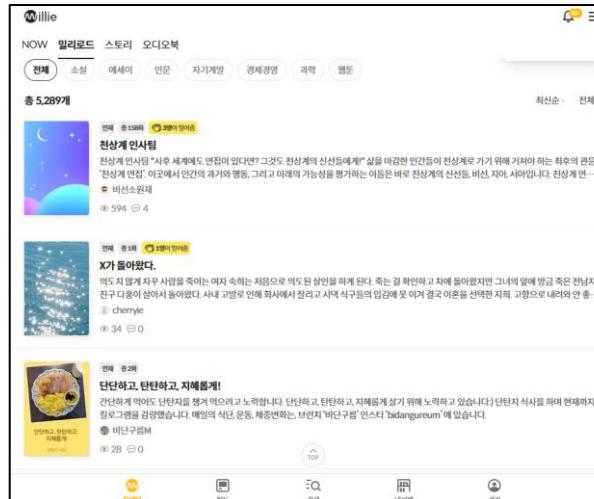
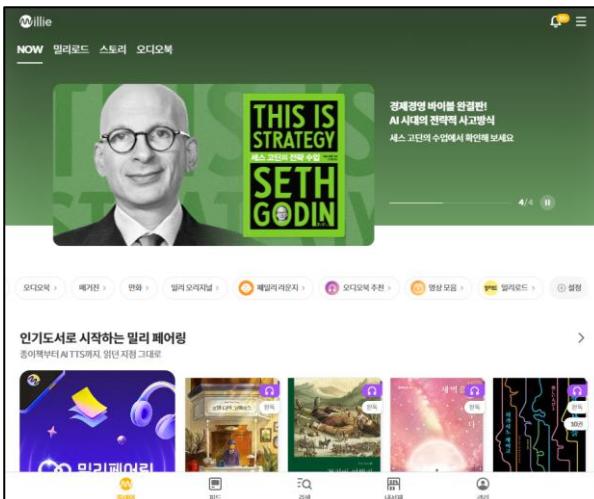
## 버전3(도서 정보 검색 서비스) 최종 프로젝트 소개

- SSAFY 13기부터 적용되는 PJT
- 도서 정보를 검색하고 요약하여 짧은 오디오 북으로 제작
- 마일리지 충전과 사용을 통한 도서 구매 및 대여 시스템 구현

금융상품비교

영화추천서비스

도서정보검색서비스



## 버전 선택

- 버전 1: 금융 상품 비교 앱
  - PJT01 --> PJT02 --> PJT03 --> PJT04 --> **PJT05 --> … > PJT09 --> PJT10**
- 버전 2: 영화 추천 서비스
  - PJT01 --> PJT02 --> PJT03 --> PJT04 --> **PJT05 --> … > PJT09 --> PJT10**
- 버전 3: 도서 정보 검색 서비스
  - PJT01 --> PJT02 --> PJT03 --> PJT04 --> **PJT05 --> … > PJT09 --> PJT10**

금융상품비교

영화추천서비스

도서정보검색서비스

(필수) 세 가지 과제 중 하나를 선택하여 진행해야 함

추후 프로젝트 버전 확정 전까지 자유롭게 변경 가능 (어느 PJT부터 확정인지는 나중에 공지)

그 후로는 최종 프로젝트를 진행하기 위한 내용이 포함 -> 변경 불가

## 교안 표기

- Youtube 에서는 세 가지 프로젝트를 모두 설명 드리기에,  
다음과 같이 교안에 구분하여 표기합니다.

금융상품비교  
영화추천서비스  
도서정보검색서비스

모든 서비스를 위한  
배경지식 페이지

금융상품비교  
영화추천서비스  
도서정보검색서비스

금융 상품 비교  
서비스 관련 페이지

금융상품비교  
영화추천서비스  
도서정보검색서비스

영화 추천 서비스  
관련 페이지

금융상품비교  
영화추천서비스  
도서정보검색서비스

도서 정보 검색 서비스  
관련 페이지

금융상품비교

영화추천서비스

도서정보검색서비스

# 프로젝트를 위한 배경지식

금융상품비교

영화추천서비스

도서정보검색서비스

## 챕터의 포인트

- 목표
- API 이해하기
- 날씨 데이터 수집

금융상품비교

영화추천서비스

도서정보검색서비스

## 실습 목표

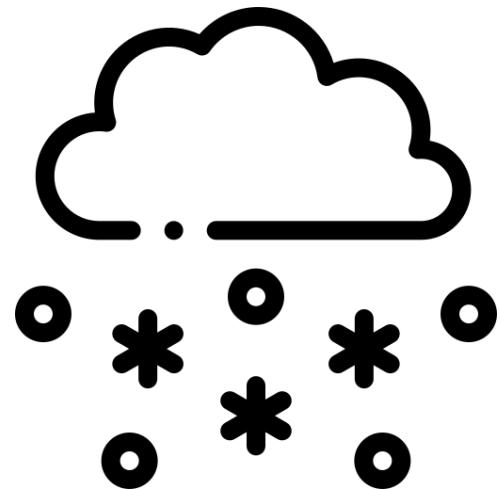
## 실습 목표

- 파이썬으로, 인터넷에 있는 날씨 정보를 가져와, 내가 원하는 정보만 출력

금융상품비교

영화추천서비스

도서정보검색서비스



## | 날씨 정보

- 실습 프로젝트를 진행하기 위해선 날씨 데이터가 있어야 합니다.
- 그러나... 직접 데이터를 모으기엔 너무 어렵습니다!
- 간단하게, 인터넷에 있는 데이터를 가져오면 됩니다.
  - 데이터를 가져오는 방법을 이해하기 위해서 반드시 알아야 할 전문용어들이 있습니다.

금융상품비교

영화추천서비스

도서정보검색서비스

## 전문용어 이해하기

- 먼저, 서버와 클라이언트입니다.
- 서버: 부탁을 받으면 처리해주거나, 부탁대로 원하는 값을 돌려주는 역할을 합니다.
- 클라이언트: 부탁하는 역할입니다.



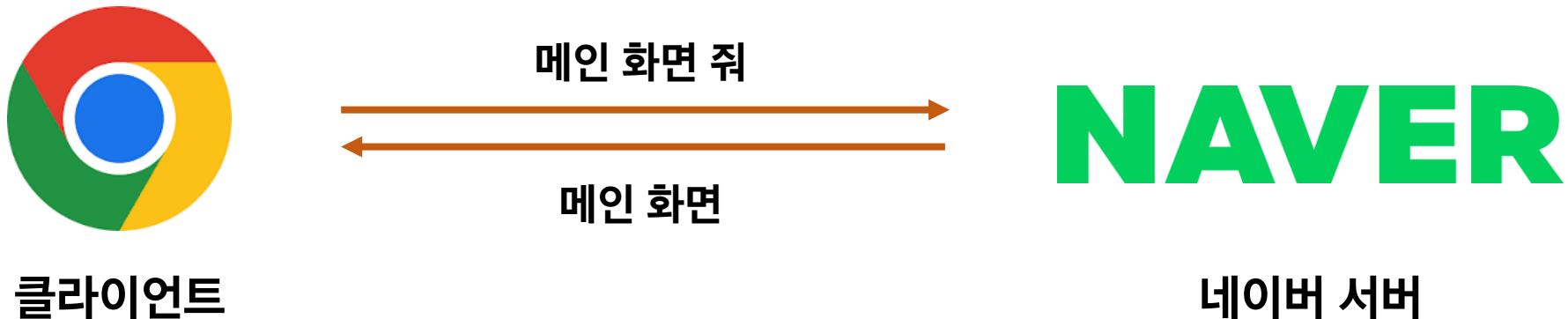
금융상품비교

영화추천서비스

도서정보검색서비스

## 전문용어 이해하기

- 우리가 네이버 홈페이지에 접속하는 건 다음과 같이 표현할 수 있습니다.
  - 네이버 주소를 입력하면 익히 알고 있는 네이버 메인 화면을 달라고 요청합니다.
  - 서버는 클라이언트가 요청한 네이버 메인 화면을 돌려줍니다.



금융상품비교

영화추천서비스

도서정보검색서비스

## 전문용어 이해하기

- 이번 프로젝트에서는 날씨 정보가 필요합니다.
  - 날씨 정보를 가지고 있는 서버가 있습니다.
  - 해당 서버에 날씨 정보를 달라고 요청하면 됩니다.



클라이언트



날씨 정보를  
가지고 있는 서버

금융상품비교

영화추천서비스

도서정보검색서비스

## | 전문용어 이해하기

- 정리하면 다음과 같습니다.
  - 클라이언트가 정보를 달라고 요청한다.
  - 서버는 클라이언트의 요청에 따라 원하는 정보를 돌려준다.
- 클라이언트는 어떻게 요청을 보낼 수 있을까요 ?

금융상품비교

영화추천서비스

도서정보검색서비스

## | 클라이언트가 서버에 요청하는 두 가지 방법

1. 웹 브라우저(크롬)을 켜서 주소창에 주소(URL)를 입력한다.
2. 서버에 정보를 요청하는 파이썬 코드를 작성한다.

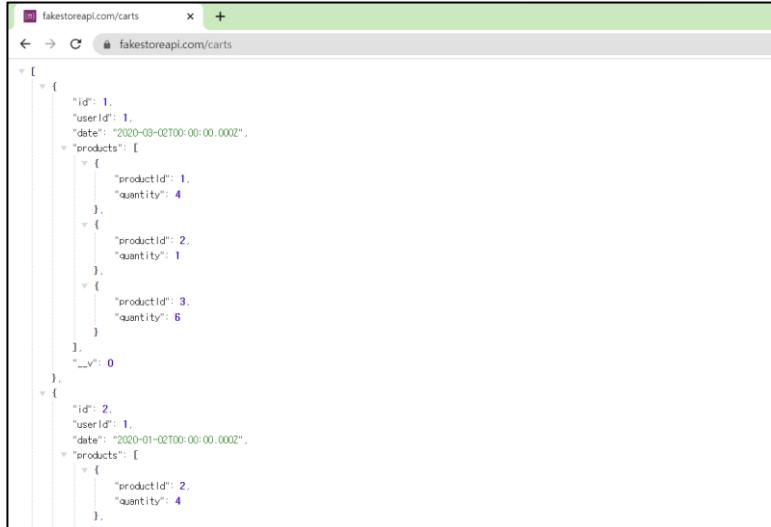
금융상품비교

영화추천서비스

도서정보검색서비스

## 1. 웹 브라우저(크롬)을 켜서 URL에 접속

- 크롬을 켜서 주소창에 아래 URL을 입력해보세요.
  - <https://fakestoreapi.com/carts>
- 예시



금융상품비교

영화추천서비스

도서정보검색서비스

## | 2. 서버에 정보를 달라는 파이썬 코드를 작성

- vscode를 켜고, 터미널 창을 열어주세요.
  - 터미널 창 여는 방법
    - 상단 Terminal 메뉴 + New Terminal 클릭
    - 단축키: Ctrl + ` (백틱)
  - 아래 명령어를 실행하여 필요한 도구를 설치합니다.
  - requests: 파이썬에서 서버에 요청을 보낼 수 있는 도구

```
$ pip install requests
```

금융상품비교

영화추천서비스

도서정보검색서비스

## 2. 서버에 정보를 달라는 파이썬 코드를 작성

- test.py 파일을 만들고, 아래처럼 파이썬 코드를 작성하고 실행해 보세요.

```
import requests

url = 'https://fakestoreapi.com/carts'
data = requests.get(url).json()
print(data)
```

- 결과 예시

```
[{"id": 1, "userId": 1, "date": "2020-03-02T00:00:00.000Z", "products": [{"productId": 1, "quantity": 4}, {"productId": 2, "quantity": 1}, {"productId": 3, "quantity": 6}], "__v": 0}, {"id": 2, "userId": 1, "date": "2020-01-02T00:00:00.000Z", "products": [{"productId": 2, "quantity": 4}, {"productId": 1, "quantity": 10}, {"productId": 5, "quantity": 2}], "__v": 0}, {"id": 3, "userId": 2, "date": "2020-03-01T00:00:00.000Z", "products": [{"productId": 1, "quantity": 2}, {"productId": 9, "quantity": 1}], "__v": 0}, {"id": 4, "userId": 3, "date": "2020-01-01T00:00:00.000Z", "products": [{"productId": 1, "quantity": 4}], "__v": 0}, {"id": 5, "userId": 3, "date": "2020-03-01T00:00:00.000Z", "products": [{"productId": 7, "quantity": 1}, {"productId": 8, "quantity": 1}], "__v": 0}, {"id": 6, "userId": 4, "date": "2020-03-01T00:00:00.000Z", "products": [{"productId": 12, "quantity": 3}], "__v": 0}, {"id": 7, "userId": 8, "date": "2020-03-01T00:00:00.000Z", "products": [{"productId": 18, "quantity": 1}], "__v": 0}]
```

금융상품비교

영화추천서비스

도서정보검색서비스

## 파이썬 코드 이해하기

- url
  - 요청을 보내는 서버의 주소
- requests.get(url)
  - 해당 서버(url)에 데이터를 달라고 요청을 보내는 함수
- .json()
  - 내부의 데이터를 JSON(파이썬의 딕셔너리와 비슷함) 형태로 변환해주는 함수
  - JSON에 대한 자세한 설명은 뒷부분에서 다룰 예정입니다.

```
import requests
url = 'https://fakestoreapi.com/carts'
data = requests.get(url).json()
print(data)
```

금융상품비교

영화추천서비스

도서정보검색서비스

## | 서버는 어떻게 요청을 해석할까

- 웹 브라우저(크롬)와 파이썬을 통해 서버에 데이터를 요청하는 방법을 알았습니다.
- 그렇다면, 서버는 어떻게 요청을 이해하고 데이터를 반환할 수 있을까요 ?

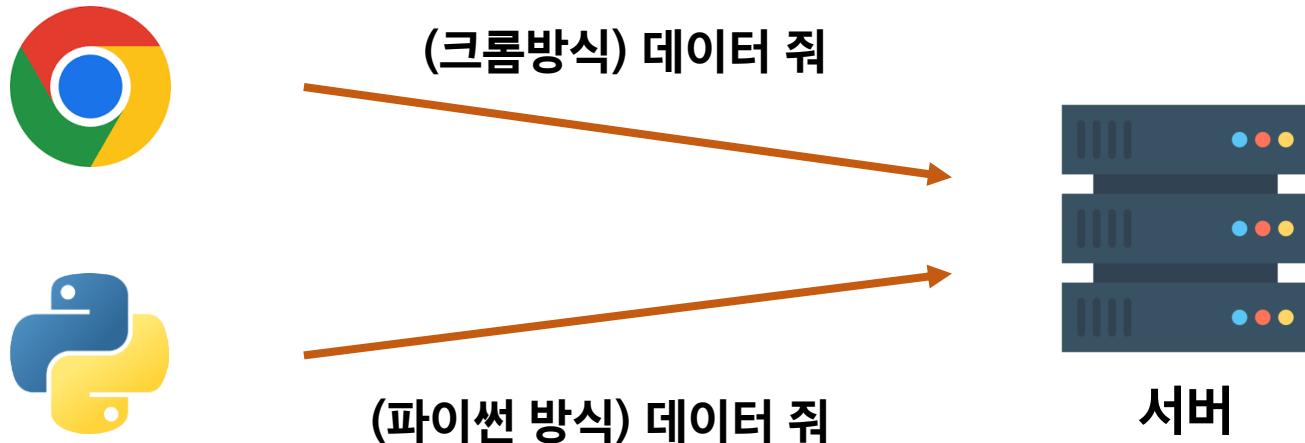
금융상품비교

영화추천서비스

도서정보검색서비스

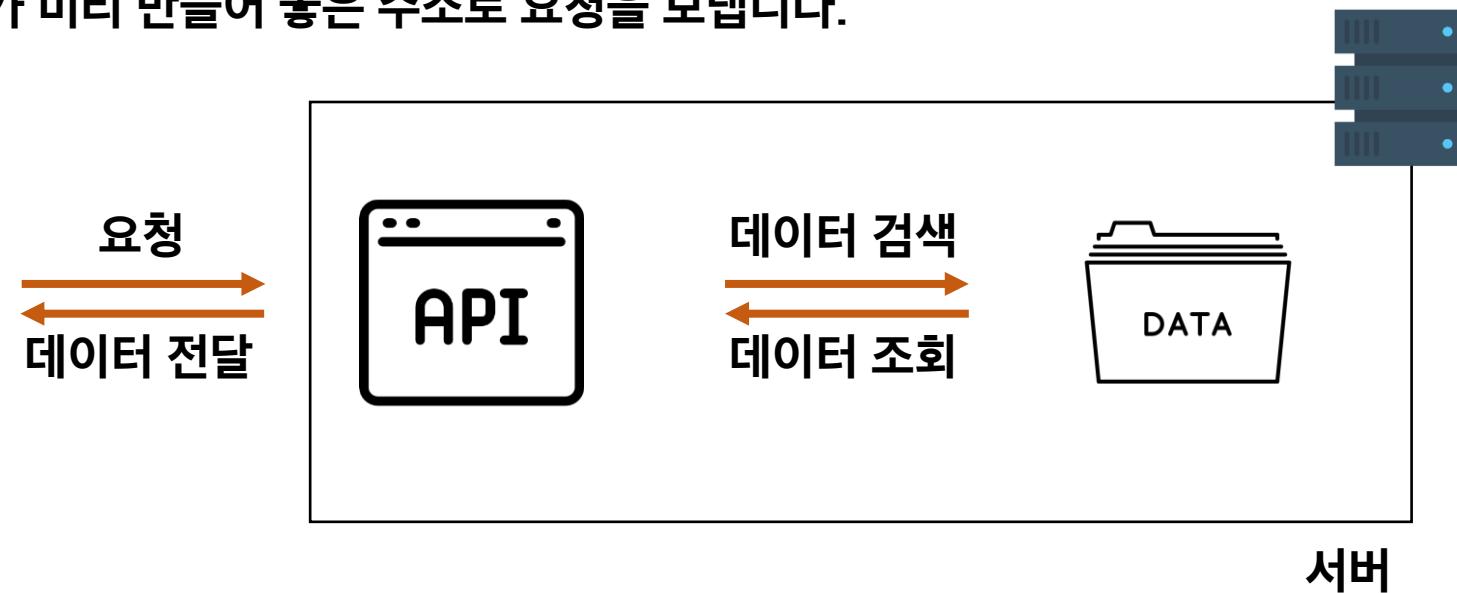
## 서버는 어떻게 요청을 해석할까

- 서버에 요청을 보내는 클라이언트는 매우 다양합니다.
  - 클라이언트들은 각자 다른 방식으로 요청을 보낼 것입니다.
- 서버가 어떻게 모두 해석할 수 있을까?



## API

- 클라이언트가 원하는 기능을 수행하기 위해서 서버 측에 만들어 놓은 프로그램
  - 기능 예시: 데이터 저장, 조회, 수정, 삭제 등등
- 서버 측에 특정 주소로 요청이 오면 정해진 기능을 수행하는 API를 미리 만들어 둡니다.
- 클라이언트는 서버가 미리 만들어 놓은 주소로 요청을 보냅니다.



금융상품비교

영화추천서비스

도서정보검색서비스

## | 날씨 정보를 제공해주는 API

- 날씨 정보를 수집하기 위해서는 두 가지를 찾아야 합니다.
  - 날씨 정보를 가지고 있는 서버
  - 해당 서버가 제공하는 API

금융상품비교

영화추천서비스

도서정보검색서비스

## 오픈 API

- 외부에서 사용할 수 있도록 무료로 개방(OPEN) 된 API
- 사용법은 공식 문서(Docs)에 명시되어 있습니다.
- 프로젝트에서 사용되는 API
  - [OpenWeatherMap API](#): 기상 데이터 및 날씨 정보를 제공하는 오픈 API
  - [금융상품통합비교공시 API](#): 금융감독원에서 제공하는 금융 상품 정보를 제공하는 오픈 API

금융상품비교

영화추천서비스

도서정보검색서비스

## | 오픈 API 특징 및 주의사항 (1/2)

- 악성 사용자가 100만 개의 계정을 생성해 API에 요청을 보내는 상황을 생각해 봅시다.
  - 너무 많은 계정에서 동시에 요청을 보내면 서버가 견디지 못합니다!
- 이러한 문제점을 해결하기 위해 오픈 API는 **API KEY**를 활용하여 사용자를 확인합니다.
  - 사용자 인증 혹은 회원가입을 하면 서버에서 API KEY를 발급해 줍니다.
  - 서버에 요청할 때마다 해당 API KEY를 함께 보내 정상적인 사용자인 것을 확인 받습니다.

금융상품비교

영화추천서비스

도서정보검색서비스

## | 오픈 API 특징 및 주의사항 (2/2)

- API KEY 를 가지고 있는 악성 사용자가 1초에 100만 개의 요청을 보내는 상황을 가정
  - 서버가 견디지 못하여 정상적인 서비스가 불가능합니다.
- 일부 오픈 API 는 사용량이 제한되어 있습니다.
  - 공식 문서의 일일 및 월간 사용량 제한을 반드시 확인하여야 합니다.
  - [주의] 사용량이 초과될 경우 요금이 청구될 수 있습니다.

금융상품비교

영화추천서비스

도서정보검색서비스

## | Quiz.

- 무료로 사용 가능한 API 를 오픈 API 라고 합니다.
- 인증된 사용자만 제공된 API 를 사용할 수 있습니다.
- 클라이언트는 원하는 데이터를 요청 할 때

금융상품비교

영화추천서비스

도서정보검색서비스

서버에게 무엇을 함께 보내어 본인이 인증된 사용자인 것을 확인 받을 수 있나요?

## 프로젝트 미션 정리

- 파이썬으로

날씨 정보를 제공해주는 서버의 URL 을 이용하여

날씨 정보를 가져옵니다.

금융상품비교

영화추천서비스

도서정보검색서비스

금융상품비교

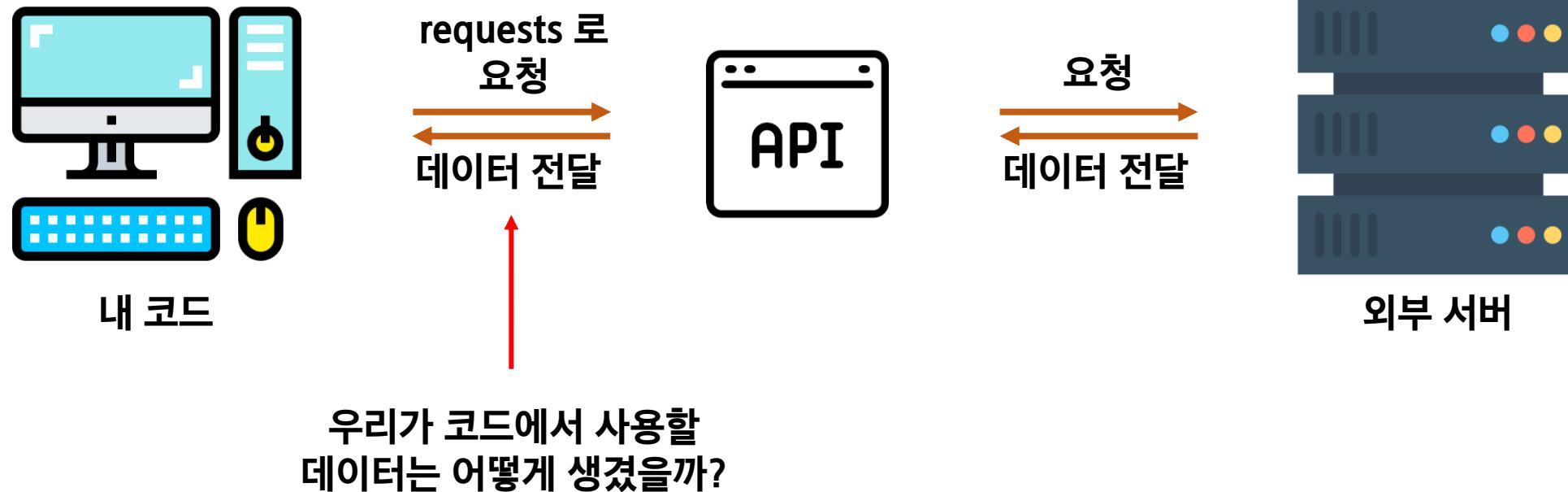
영화추천서비스

도서정보검색서비스

## 날씨 데이터 수집

## JSON

- API 가 반환하는 데이터는 어떻게 생겼을까?



## API 가 사용하는 데이터 형식 - JSON

- JavaScript Object Notation 의 약자. 직역하면 ‘자바스크립트 객체 표기법’
- 데이터를 저장하거나 전송할 때 많이 사용되는 경량의 텍스트 기반의 데이터 형식
- 통신 방법이나 프로그래밍 문법이 아니라 단순히 데이터를 표현하는 표현 방법 중 하나
- 특징
  - 데이터는 중괄호({})로 둘러싸인 키-값 쌍의 집합으로 표현됨
  - 키 = 문자열 / 값 = 다양한 데이터 유형을 가질 수 있다
  - 값은 쉼표(,)로 구분됨
- 예시

```
{  
    "name": "김싸피",  
    "age": 28,  
    "city": "서울 캠퍼스",  
    "hobbies": [  
        "공부하기",  
        "복습하기"  
    ],  
    "isStudent": true  
}
```

금융상품비교

영화추천서비스

도서정보검색서비스

## JSON - python 예제

- 파이썬은 JSON 을 활용하는 기능을 가지고 있습니다.
- 참고
  - 파싱(Parsing)
    - 데이터를 의미 있는 구조로 분석하고 해석하는 과정
  - json.loads()
    - JSON 형식의 문자열을 파싱하여 python Dictionary 로 변환

```
import json

# JSON 데이터
json_data = '''
{
    "name": "김싸피",
    "age": 28,
    "city": "서울 캠퍼스",
    "hobbies": [
        "공부하기",
        "복습하기"
    ],
    "isStudent": true
}
'''

# JSON 데이터 파싱하기
data = json.loads(json_data)

# JSON 데이터에서 정보 읽기
name = data["name"]
age = data["age"]
city = data["city"]
hobbies = data["hobbies"]
is_student = data["isStudent"]

print(name) # 김싸피
print(age) # 28
print(city) # 서울 캠퍼스
print(hobbies) # ["공부하기", "복습하기"]
print(is_student) # True
```

금융상품비교

영화추천서비스

도서정보검색서비스

## | OpenWeatherMap API

- 기상 데이터 및 날씨 정보를 제공하는 오픈 API
- 전세계 날씨 데이터를 가져와 날씨, 일일 및 시간 별 예보 등 다양한 정보를 얻을 수 있습니다.
- API 사용량 제한
  - 60 calls/minute
  - 1,000,000 calls/month



금융상품비교

영화추천서비스

도서정보검색서비스

## | OpenWeatherMap API

- 생성형 AI 를 활용하여 OpenWeatherMap API 에 대해 검색해보자.

금융상품비교

영화추천서비스

도서정보검색서비스

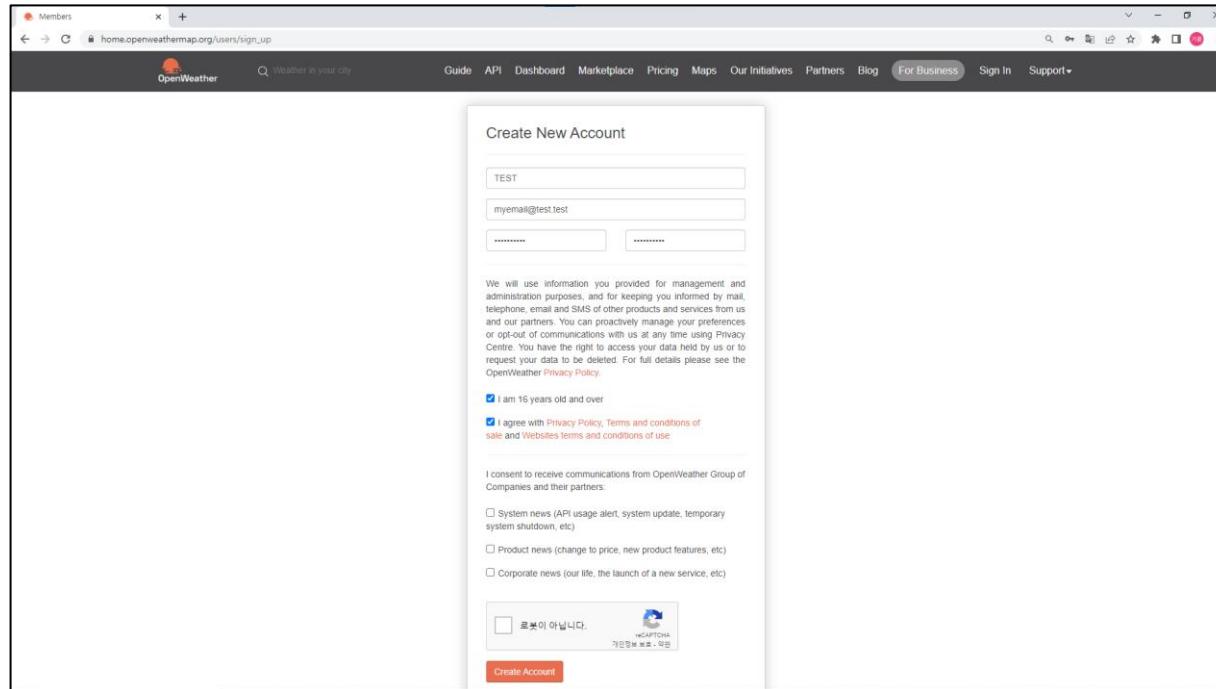
# OpenWeatherMap API - KEY 발급(1/3)

- 사이트 접속 및 회원가입 진행
  - 가입 후 반드시 이메일 인증을 완료해야 합니다.

금융상품비교

영화추천서비스

도서정보검색서비스



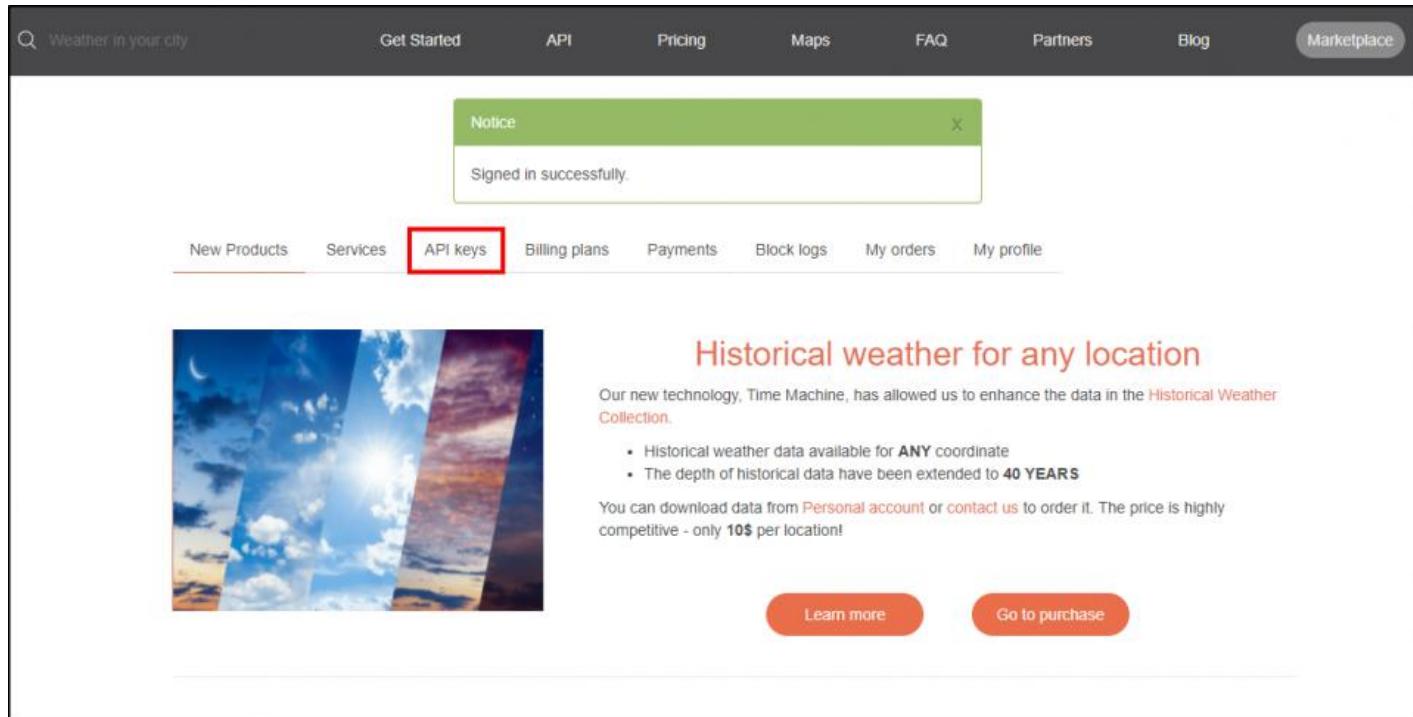
# | OpenWeatherMap API - KEY 발급(2/3)

- API Keys 탭으로 이동

금융상품비교

영화추천서비스

도서정보검색서비스



# OpenWeatherMap API - KEY 발급(3/3)

- API Key 복사
  - API Key 를 복사하여 코드에서 활용합니다.

The screenshot shows the OpenWeatherMap API keys management interface. At the top, there's a navigation bar with links like Weather in your city, Guide, API, Dashboard, Marketplace, Pricing, Maps, Our Initiatives, Partners, Blog, For Business, and Support. Below the navigation, there are several tabs: New Products, Services, API keys (which is underlined), Billing plans, Payments, Block logs, My orders, My profile, and Ask a question. A message box states: "You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them." Below this, there's a table with columns: Key, Name, Status, and Actions. One row is visible, showing a redacted key, the name "Default", and the status "Active". To the right of the table is a "Create key" section with a "Create key" button and a "Generate" button. The text "해당 부분의 Key 를 복사합니다." is overlaid at the bottom left of the screenshot.

금융상품비교

영화추천서비스

도서정보검색서비스

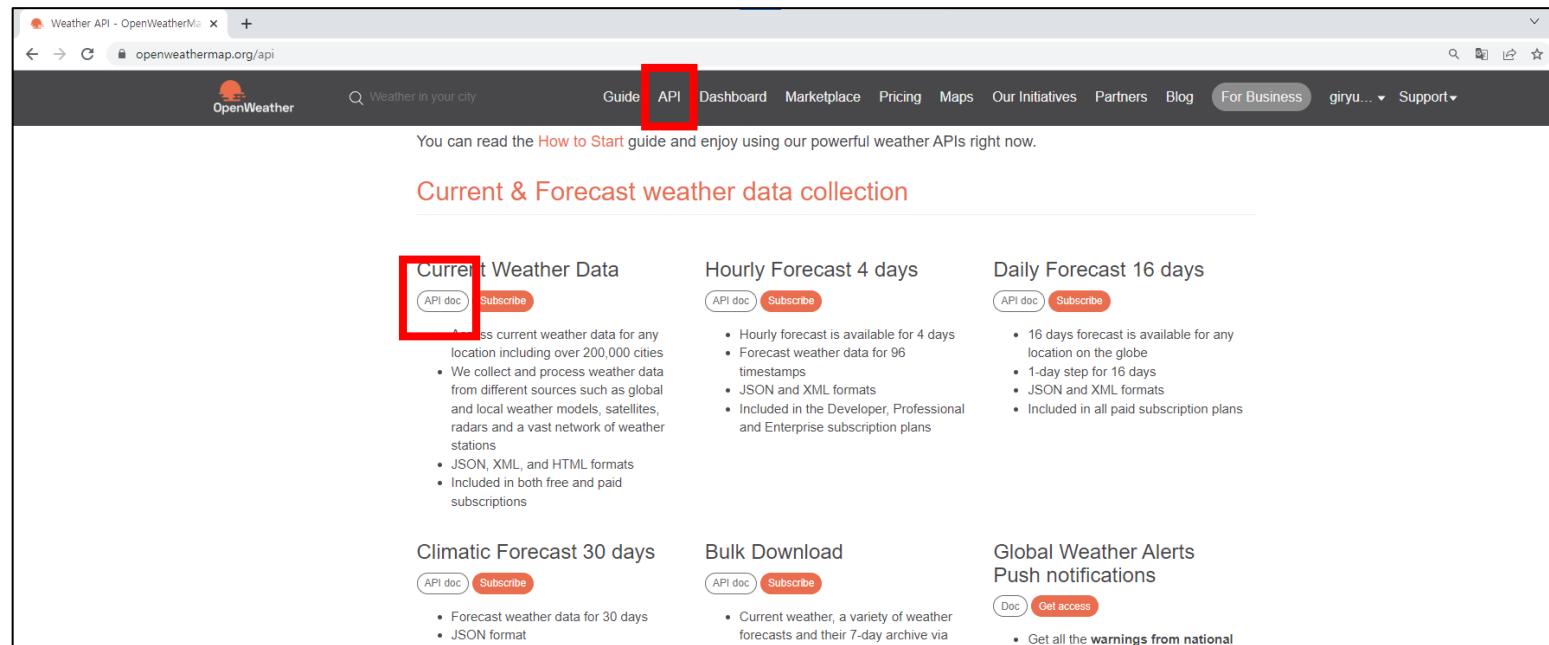
# OpenWeatherMap API - 사용법 확인

- 상단 API 탭을 클릭
  - 현재 날씨에 대한 데이터를 사용하기 위해 Current Weather Data 의 API doc 클릭

금융상품비교

영화추천서비스

도서정보검색서비스



금융상품비교

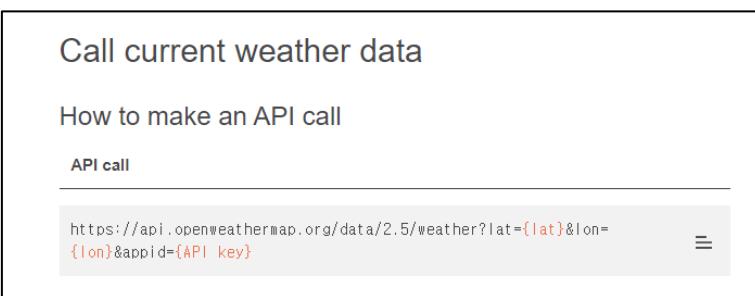
영화추천서비스

도서정보검색서비스

## OpenWeatherMap API - 실습(1/4)

- 특정 지역의 현재 날씨에 대한 모든 정보 출력하기

- 공식 문서 참고



- [참고] 서울의 위도: 37.56 / 경도: 126.97

```
# 서울의 위도
lat = 37.56
# 서울의 경도
lon = 126.97

# API 요청 URL
url = f'https://api.openweathermap.org/data/2.5/weather?lat={lat}&lon={lon}&appid={api_key}'
```

- 출력 결과

- 날짜에 따라 출력 결과는 다를 수 있습니다.

```
{'base': 'stations',
'clouds': {'all': 0},
'cod': 200,
'coord': {'lat': 37.56, 'lon': 126.97},
'dt': 1684293603,
'id': 1835848,
'main': {'feels_like': 300.17,
'humidity': 39,
'pressure': 1013,
'temp': 300.43,
'temp_max': 301.93,
'temp_min': 297.96},
'name': 'Seoul',
'sys': {'country': 'KR',
'id': 8105,
'sunrise': 1684268515,
'sunset': 1684319716,
'type': 1},
'timezone': 32400,
'vesibility': 10000,
'weather': [{"description": "clear sky",
'icon': '01d',
'id': 800,
'main': 'Clear'}],
'wind': {'deg': 200, 'speed': 6.69}}
```

금융상품비교

영화추천서비스

도서정보검색서비스

## | OpenWeatherMap API - 실습(2/4)

- 특정 도시의 현재 날씨를 도시 이름으로 요청하여 모든 정보 출력하기

- 공식 문서 참고

**Built-in API request by city name**

You can call by city name or city name, state code and country code. Please note that searching by states available only for the USA locations.

**API call**

```
https://api.openweathermap.org/data/2.5/weather?q={city name}&appid={API key}
```

**Output Options**

- JSON
- XML
- List of condition codes
- Min/max temperature in current weather API and forecast API

**Other features**

- Geocoding API
- Built-in geocoding
- Built-in API request by city name**
- Built-in API request by city ID
- Built-in API request by ZIP code

**Format**

- [참고] 도시 이름으로 요청 URL 만들기

```
# 검색 조건
city = "Seoul,KR"

# 다른 지역
# city = "Tokyo,JP"
# city = "New York,US"

# API 요청 URL
url = f'http://api.openweathermap.org/data/2.5/weather?q={city}&appid={api_key}'
```

- 출력 결과

- 날짜에 따라 출력 결과는 다를 수 있습니다.

```
{"base": "stations",
"clouds": {"all": 0},
"cod": 200,
"coord": {"lat": 37.56, "lon": 126.97},
"dt": 1684293603,
"id": 1835848,
"main": {"feels_like": 300.17,
          "humidity": 39,
          "pressure": 1013,
          "temp": 300.43,
          "temp_max": 301.93,
          "temp_min": 297.96},
"name": "Seoul",
"sys": {"country": "KR",
          "id": 8105,
          "sunrise": 1684268515,
          "sunset": 1684319716,
          "type": 1},
"timezone": 32400,
"visibility": 10000,
"weather": [{"description": "clear sky",
            "icon": "01d",
            "id": 800,
            "main": "Clear"}],
"wind": {"deg": 200, "speed": 6.69}}
```

## | OpenWeatherMap API - 실습(3/4)

- 서울의 현재 날씨 중 온도만 출력하기
  - 기본적으로 캘빈(K) 온도를 반환합니다.
  - 섭씨 온도 = (캘빈 온도 - 273.15)로 계산할 수 있습니다.
  - 출력 결과

캘빈 온도: 300.71K

섭씨 온도: 27.56°C

금융상품비교

영화추천서비스

도서정보검색서비스

## | OpenWeatherMap API - 실습(4/4)

- 서울의 현재 날씨에 대한 설명(description) 데이터만 출력하기

- 출력 결과

```
'날씨 설명: clear sky'
```

금융상품비교

영화추천서비스

도서정보검색서비스

- Json 형태의 데이터를 분석하여 원하는 부분만 가져오도록 구성합니다.

```
json_response = requests.get(url).json()
description = json_response['weather'][0]['description']
return f'날씨 설명: {description}'
```

# 도전 과제 소개

금융상품비교

영화추천서비스

도서정보검색서비스

# 금융 상품 비교 앱 PJT 01

## 버전1 - PJT01 도전 과제

- **프로젝트명:** 파이썬을 활용한 API 데이터 수집
- **목표**
  - 파이썬으로 날씨 데이터 수집 및 미션 수행
- **특징**
  - 오픈 API를 활용한 데이터 수집
  - 요구사항에 맞게 JSON 형태 데이터 가공
  - [심화] 정기 예금 데이터 수집 및 미션 수행 (선택 사항)

금융상품비교

영화추천서비스

도서정보검색서비스

금융상품비교

영화추천서비스

도서정보검색서비스

# 영화 추천 서비스 PJT 01

## 버전2 - PJT01 도전 과제

- **프로젝트명:** 파이썬을 활용한 데이터 수집1
- **목표**
  - 파이썬으로 영화 및 아티스트 데이터 가공 및 미션 수행
- **특징**
  - 샘플 데이터 제공
  - 요구사항에 맞게 JSON 형태 데이터 가공

금융상품비교

영화추천서비스

도서정보검색서비스

## | 도전과제

- 버전2의 도전 과제는 아래 데이터들이 제공됩니다. (직접 수집 X)
  - data 폴더 (artist.json, artists.json, genres.json)

금융상품비교

영화추천서비스

도서정보검색서비스

금융상품비교

영화추천서비스

도서정보검색서비스

# 도서 정보 검색 서비스 PJT 01

## 버전3 - PJT01 도전 과제

- **프로젝트명:** 파이썬을 사용한 수집 데이터 활용
- **목표**
  - 파이썬으로 도서 데이터를 직접 수집하고 이를 활용하여 주어진 데이터를 분류하고 응용
- **특징**
  - 오픈 API를 활용한 데이터 수집
  - 요구사항에 맞게 JSON 형태 데이터 가공
  - gTTS 라이브러리를 활용한 음성 파일 생성

금융상품비교

영화추천서비스

도서정보검색서비스

## | 도전과제에서 사용하는 API

- **Aladin API**
  - 알라딘 온라인 서점
  - 책, 음반, 도서 등의 상품 정보 데이터를 제공하는 API
  - 공식문서가 블로그 형태로 제공됨
    - 블로그를 잘 보면서 API KEY 발급 및 사용 방법을 익혀야 함

금융상품비교

영화추천서비스

도서정보검색서비스