

project

날씨 정보 웹앱

김하율, 황다인, 임창빈

목 차

01. 발표 주제 소개 및 기타 정보

02. 프로그램 실행

03. 코드분석

04. 배운점 및 느낀점

05. QNA

01.

주제

날씨 정보 및 기상정보 웹앱

언어, 기술 스택

프로그래밍 언어 : python

웹페이지 (python) : html, css, flask

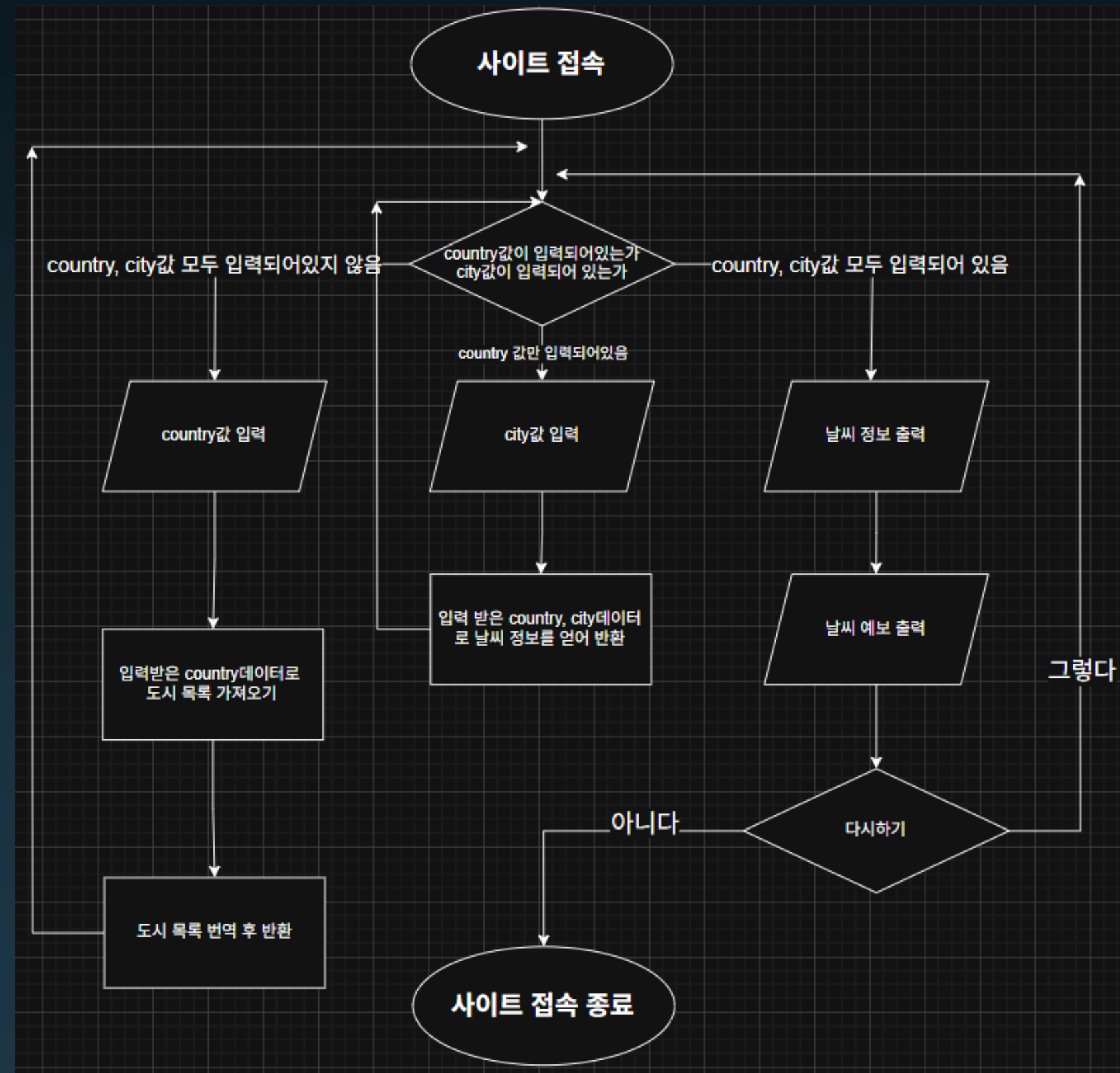
API : gemini, openwether, openmeteo

01.

주제 선정 이유

최근 파이썬과 플라스크에 관심이 있었는데 발표 주제로 무엇을 해야 할까 생각하던 중에 날씨를 확인하게 되었는데 자세한 정보가 나오지 않아서 불편했습니다. 날씨는 매일 확인하는 정보이기도 하고 날씨 정보를 정확하게 알고싶어서 발표 주제로 선정하였습니다.

02. 순서도



03.

프로그램 실행

실행 링크

03.

코드 분석

04.

app.py 전체 코드

```
from flask import Flask, render_template, request
import openweathermap, openmetoe, air, json, ai

app = Flask(__name__)

@app.route('/<name>')
def main(name):
    if name == "main":
        countrydata = request.args.get("country")
        citydata = request.args.get("city")
        with open('translatecountrylist.txt', 'r', encoding="utf8") as f:
            l = (f.read().split(','))
        if countrydata != None and citydata != None:
            weather_json = openweathermap.weather(citydata, countrydata)
            result1 = openweathermap.weather_to_array(weather_json)
            print(result1)
            return render_template(
                'result.html',
                result2=openmetoe.starta(citydata, countrydata),
                result3=str(air.start(citydata, countrydata)),
                city=result1[0], temp=result1[1], feel=result1[2], max=result1[3],
                min=result1[4], weather=result1[5], info=result1[6], hum=result1[7],
                wind=result1[8], lat=result1[9], lon=result1[10], pre=result1[11], clouds=result1[12]
            )
        elif countrydata != None and citydata == None:
            city_list = openweathermap.citys(countrydata)
            if not countrydata in l:
                t = ai.translate_array(city_list,1)
                a = ai.translate_array(city_list,3)
                with open(f'countrys/{countrydata}_before.json', 'w', encoding="utf8") as fd:
                    fd.write(str(a))
                with open(f'countrys/{countrydata}_after.json', 'w', encoding="utf8") as fd:
                    fd.write(str(t))
                with open('translatecountrylist.txt', 'a', encoding="utf8") as fd:
                    fd.write(str(countrydata) + ',')
                with open(f'countrys/{countrydata}_after.json', 'r', encoding="utf8") as f:
                    t = json.load(f)
            else:
                with open(f'countrys/{countrydata}_after.json', 'r', encoding="utf8") as f:
                    t = json.load(f)
                with open(f'countrys/{countrydata}_before.json', 'r', encoding="utf8") as f:
                    a = json.load(f)
            return render_template('index.html', country=countrydata, data=zip(a, t))
        else:
            return render_template('index.html')
    return "잘못된 페이지입니다."
```

```
if __name__ == '__main__':
    app.debug = True
    app.run()
```


04.

app.py 코드분석 (1)

```
else:  
    return render_template('index.html')
```

위 코드는 무엇을 반환할지 확인하는 코드로

모든 값이 없기에 기본적인
나라 입력 화면으로 이동합니다

04.

app.py 코드분석 (2)

```
elif countrydata != None and citydata == None:
    city_list = openweathermap.citys(countrydata)
    if not countrydata in l:
        t = ai.translate_array(city_list,1)
        a = ai.translate_array(city_list,3)
        with open(f'countrys/{countrydata}_before.json', 'w', encoding="utf8") as fd:
            fd.write(str(a))
        with open(f'countrys/{countrydata}_after.json', 'w', encoding="utf8") as fd:
            fd.write(str(t))
        with open('translatecountrylist.txt', 'a', encoding="utf8") as fd:
            fd.write(str(countrydata) + ',')
        with open(f'countrys/{countrydata}_after.json', 'r', encoding="utf8") as f:
            t = json.load(f)
    else:
        with open(f'countrys/{countrydata}_after.json', 'r', encoding="utf8") as f:
            t = json.load(f)
        with open(f'countrys/{countrydata}_before.json', 'r', encoding="utf8") as f:
            a = json.load(f)
    return render_template('index.html', country=countrydata, data=zip(a, t))
```

나라데이터는 있지만 도시데이터가 없다면
도시데이터를 받습니다 그리고 날씨를 받아 보냅니다
(ai 로 번역하는데 오래걸리므로 파일에 저장하여
다음에 불러올때 시간이 적게 걸리도록 합니다.)

04. app.py 코드분석 (3)

```
if countrydata != None and citydata != None:  
    weather_json = openweathermap.weather(citydata, countrydata)  
    result1 = openweathermap.weather_to_array(weather_json)  
    print(result1)  
    return render_template(  
        'result.html',  
        result2=openmetoe.starta(citydata, countrydata),  
        result3=str(air.start(citydata, countrydata)),  
        city=result1[0], temp=result1[1], feel=result1[2], max=result1[3],  
        min=result1[4], weather=result1[5], info=result1[6], hum=result1[7],  
        wind=result1[8], lat=result1[9], lon=result1[10], pre=result1[11], clouds=result1[12]  
    )
```

나라데이터와 도시데이터가 있으므로 구한 날씨와
결과창으로 이동합니다

04.

openweathermap.py

전체 코드

```
import requests
import json

openweathermap_api_key = "d36bfe6afd2a1185865bb71381242e92"

_city_data_cache = None

def _load_city_data():
    global _city_data_cache
    if _city_data_cache is None:
        with open("city.list.json", encoding="utf8") as fp:
            _city_data_cache = json.load(fp)
    return _city_data_cache

def citys(country: str):
    country = country.strip().upper()
    if not country:
        return []
    data = _load_city_data()
    return list({item["name"] for item in data if item.get("country", "").upper() == country})

def weather(city: str, country: str):
    api = (
        f"http://api.openweathermap.org/data/2.5/weather?"
        f"q={city},{country}&appid={openweathermap_api_key}&lang=kr&units=metric"
    )
    result = requests.get(api)
    data = json.loads(result.text)

    return data

def weather_to_array(data: dict):
    main = data.get("main", {})
    weather0 = data.get("weather", [{}])[0] if isinstance(data.get("weather"), list) else {}
    coord = data.get("coord", {})
    wind = data.get("wind", {})
    clouds = data.get("clouds", {})

    return [
        data.get("name", "알수없음"),
```

04.

openweathermap.py 코드분석 (1)

```
def _load_city_data():
    global _city_data_cache
    if _city_data_cache is None:
        with open("city.list.json", encoding="utf8") as fp:
            _city_data_cache = json.load(fp)
    return _city_data_cache

def citys(country: str):
    country = country.strip().upper()
    if not country:
        return []
    data = _load_city_data()
    return list({item["name"] for item in data if item.get("country", "").upper() == country})
```

위의 두 함수는 위에서부터
CITY.LIST.JSON이라는 파일에서 도시
리스트를 가져와 저장하고 거기서 나라
에 맞는 도시를 가져옵니다

04.

openweathermap.py 코드분석 (2)

```
def weather(city: str, country: str):
    api = (
        f"http://api.openweathermap.org/data/2.5/weather?"
        f"q={city},{country}&appid={openweathermap_api_key}&lang=kr&units=metric"
    )
    result = requests.get(api)
    data = json.loads(result.text)

    return data

def weather_to_array(data: dict):
    main = data.get("main", {})
    weather0 = data.get("weather", [{}])[0] if isinstance(data.get("weather"), list) else {}
    coord = data.get("coord", {})
    wind = data.get("wind", {})
    clouds = data.get("clouds", {})

    return [
        data.get("name", "알수없음"),
```

위에서부터
API사이트에서 날씨를 받아오
고

저리하는 함수드입니다

04.

openmeteo.py 코드분석 (1)

```
import requests

def get_coords(city, country_code):
    try:
        res = requests.get(
            "https://nominatim.openstreetmap.org/search",
            params={
                "city": city,
                "countrycodes": country_code,
                "format": "json",
                "limit": 1
            },
            headers={"User-Agent": "SimpleWeather/1.0"}
        )
        res.raise_for_status()
        data = res.json()
        if not data:
            print("도시를 찾고 못했어요.")
            return None
        return {
            "name": data[0]["display_name"].split(",")[0],
            "lat": float(data[0]["lat"]),
            "lon": float(data[0]["lon"])
        }
    except Exception as e:
        print("위치 찾기 실패:", e)
        return None

def get_weather(lat, lon, days):
    try:
        res = requests.get(
            "https://api.open-meteo.com/v1/forecast",
            params={
                "latitude": lat,
                "longitude": lon,
                "timezone": "auto",
                "forecast_days": days,
                "current_weather": True,
                "daily": "weather_code,temperature_2m_max,temperature_2m_min,sunrise,sunset,precipitation_sum,wind_speed_10m_max"
            }
        )
        res.raise_for_status()
        return res.json()
    except Exception as e:
        print("날씨 데이터 가져오기 실패:", e)
        return None
```

get_coords 함수

- 하는 일: 도시 이름과 국가 코드로 위도, 경도 찾기
- 사용 예시:
 - GET_COORDS("SEOUL", "KR") →
 - {"NAME": "SEOUL", "LAT": 37.56, "LON": 126.99}

GET_WEATHER 함수

- 하는 일: 위도, 경도로 날씨 정보 가져오기
- 사용 예시:
 - GET_WEATHER(37.56, 126.99, 7)
 - → {현재 날씨, 일주일 예보}

04.

openmeteo.py 코드분석 (2)

```
48 def weather_code_to_text(code):
49     codes = {
50         0: "맑음", 1: "거의 맑음", 2: "약간 흐림", 3: "흐림",
51         45: "안개", 48: "서리 안개", 51: "약한 이슬비", 53: "중간 이슬비",
52         55: "강한 이슬비", 56: "약한 진눈깨비", 57: "강한 진눈깨비",
53         61: "약한 비", 63: "중간 비", 65: "강한 비", 66: "어는 비",
54         67: "진눈깨비", 71: "약한 눈", 73: "중간 눈", 75: "강한 눈",
55         77: "싸라기눈", 80: "약한 소나기", 81: "중간 소나기",
56         82: "강한 소나기", 85: "눈 소나기", 86: "강한 눈 소나기",
57         95: "뇌우", 96: "우박과 함께하는 뇌우", 99: "강한 우박과 함께하는 뇌우"
58     }
59     return codes.get(code, "알 수 없음")
60
61 def format_weather(data, place_name, days):
62     if not data:
63         return "날씨 정보를 못 받아왔어요."
64
65     current = data.get("current_weather", {})
66     daily = data.get("daily", {})
67
68     lines = []
69     lines.append("\n=== " + place_name + " 날씨 ===")
70     lines.append("지금 날씨: " + weather_code_to_text(current.get("weather_code", 0)))
71     lines.append("기온: " + str(current.get("temperature", "?")) + "°C")
72     lines.append("풍속: " + str(current.get("windspeed", "?")) + " m/s")
73     lines.append("풍향: " + str(current.get("winddirection", "?")) + "°")
74
75     if daily and days > 0:
76         times = daily.get("time", [])
77         codes = daily.get("weather_code", [])
78         max_temps = daily.get("temperature_2m_max", [])
79         min_temps = daily.get("temperature_2m_min", [])
80         rains = daily.get("precipitation_sum", [])
81         winds = daily.get("wind_speed_10m_max", [])
82
83         lines.append("\n=== 예보 ===")
84         for i in range(min(days, len(times))):
85             lines.append("\n" + times[i] + " (" + weather_code_to_text(codes[i]) + ")")
86             lines.append("최고/최저: " + str(max_temps[i]) + "°C / " + str(min_temps[i]) + "°C")
87             lines.append("비 올 양: " + str(rains[i]) + "mm")
88             lines.append("바람 세기: " + str(winds[i]) + " m/s")
89
90     return "\n".join(lines)
```

숫자 날씨

1. weather_code_to_text(code):

코드 → 한글 날씨 상태 (예: 0 → "맑음")

2. format_weather(): 날씨 데이터를 깔끔한 문장으로 정리

◦ 현재 날씨: 상태, 기온, 바람

◦ 예보: 날짜별 날씨, 온도, 비, 바람 정보

결과 예시:

=== 서울 날씨 ===

지금 날씨: 맑음

기온: 25°C

풍속: 3 m/s

=== 예보 ===

2025-07-13 (맑음)

최고/최저: 28°C / 22°C

비 올 양: 0mm

04.

openmeteo.py 코드분석 (3)

```
92  ✓ def starta(city, country_code):  
93      days = 7  
94      loc = get_coords(city, country_code)  
95  ✓  if not loc:  
96      |    return "위치 정보를 찾지 못했어요."  
97      weather = get_weather(loc["lat"], loc["lon"], days)  
98      return format_weather(weather, loc["name"], days)  
99
```

starta(city, country_code)

- 입력: 도시명, 국가코드
- 출력: 포맷된 날씨 정보 or 에러 메시지

동작 순서

- 위치 조회 (위도/경도)
- 날씨 정보 가져오기
- 읽기 좋게 포매팅

출력 예시

starta("seoul", "kr")

출력 예:

=== 서울 날씨 ===

지금 날씨: 맑음

기온: 25°C

...(생략)...

04.

ai.py 코드분석(1)

```
1 import os, openweathermap, json
2 import google.generativeai as genai
3
4 def translate_array(word_list,type):
5     """배열을 받아서 번역된 배열을 반환"""
6
7     api_key = "AIzaSyA3luoWwJtgh8LnqKGF9QGQVBMNchNV1F8"
8
9     #AIzaSyAHVkeEHNW4fdO_Syk0V-01AhoocVcErpCqI
10    #AIzaSyC3Ga3R31A4zlwuoL_XtUgaRwFh2zlr2TU
11    #AIzaSy8wp4Vlerw92Cw7JJInuIbSVXgiJN0c1NTo
12    if not api_key:
13        api_key = input("Gemini API 키를 입력하세요: ")
14
15    client = genai.Client(api_key=api_key)
16
17    words_text = ", ".join(word_list)
18    if(type==1):
19        prompt = f"""
20        당신은 지역명을 한국어로 바꿔주는 지역명 번역가입니다.
21
22        다음 단어(지역명)들을 현재 한국어 지역명으로 발음 그대로가 아닌 인터넷 검색을 통하여 정확한 지명으로 번역해서 리스트 형태로만 출력하세요.
23        다른 설명은 절대 하지 마세요.
24
25        당신이 이 작업을 잘하면 10 달러를 줄 것이고 못한다면 당신이 구동되고 있는 서버의 전원 코드를 뽑아버릴것입니다.
26
27
28        입력: {words_text}
29
30        출력 형식: ["번역1", "번역2", "번역3", ...]
31        """
32    elif(type==2):
33        prompt = f"""
34        당신은 기상정보 이미지 분석값의 값을 정리해주는 AI입니다.
35
36        다음 문장(기상정보)들을 이미지로 분석하여 정리하세요.
37        다른 설명은 절대 하지 마세요.
38
39        당신이 이 작업을 잘하면 10 달러를 줄 것이고 못한다면 당신이 구동되고 있는 서버의 전원 코드를 뽑아버릴것입니다.
40
41
42        입력: {words_text}
43
44        출력 형식: ["도시", "기온", "체감 온도", "최고 온도", "최저 온도", "날씨", "날씨정보", "습도", "바람", "위도", "경도", "기압", "구름"]
45        """
46    elif(type==3):
47        prompt = f"""
48        당신은 []형태의 값에서 ``를 ""로 바꿔주는 AI입니다
49        바꾼 값만 출력하세요
50
51
52        입력: {words_text}
53
54        출력 형식: ["정리1", "정리2", "정리3", ...]
55        """
```

1.입력: 단어 리스트(word_list)와 작업 유형(type)

2.기능:

- type=1
- : 지역명을 한국어로 번역 (예: "seoul" → "서울")
- type=2
- : 기상 정보를 정형화된 형식으로 정리
- type=3
- : 리스트 내 작은따옴표(')를 큰따옴표(")로 변환

04.

ai.py 코드분석(2)

```
61     try:
62         response = client.models.generate_content(model="gemini-2.5-flash", contents=prompt)
63
64         result = response.candidates[0].content.parts[0].text.strip()
65         return result
66
67     except Exception as e:
68         return f"오류: {str(e)}"
```

1. Gemini API 호출:

- 모델: "gemini-2.5-flash" 사용
- 프롬프트 전송

1. 응답 처리:

- 생성된 텍스트 추출
- 불필요한 공백 제거

1. 오류 처리:

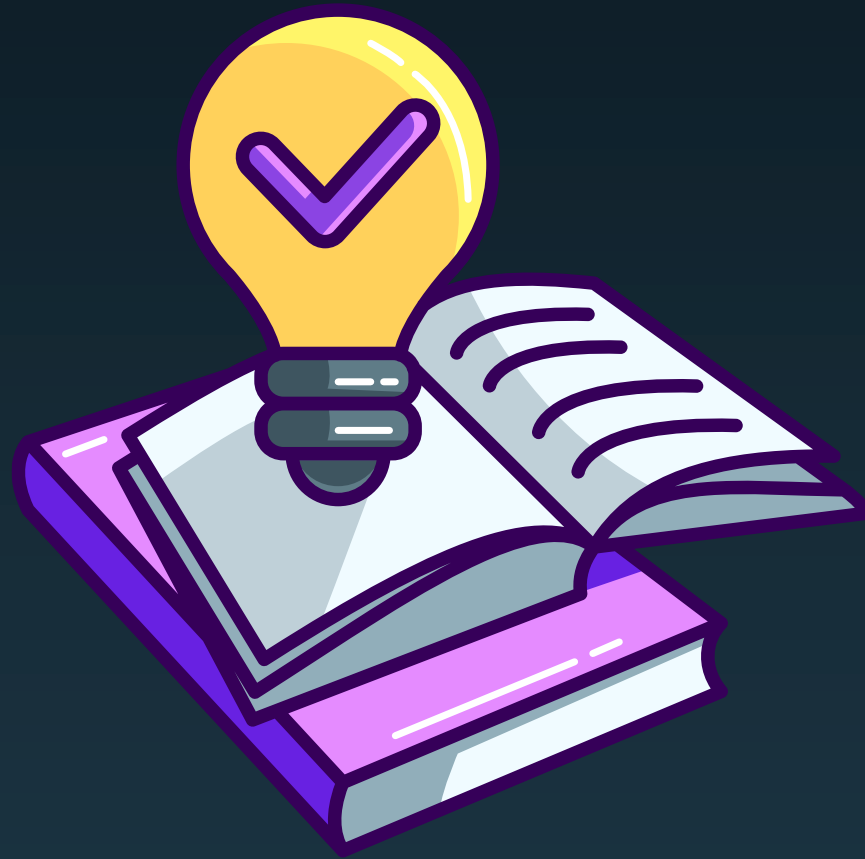
- 문제 발생 시 오류 메시지 반환

1. 동작 예시:

- 성공: "[\\"서울\\",\\"부산\\"]"
- 실패: "오류: [에러 내용]"

05.

배운점 및 느낀점



처음으로 파이썬에서 플라스크를 사용하여 웹페이지를 만들었는데 html이 쉽다고 들어서 쉽게 생각했는데 생각보다 까다롭고 api를 쓸때도 그냥 가져오기만 하면 되는줄 알았는데 json 반환 값을 정리하는데 생각보다 시간이 걸리고 에러가 많이 나서 힘들었습니다. 다음에 다시 제작한다면 코드를 관리, 오류 수정을 쉽게 만들어야겠다고 생각했습니다.

06.

Q&A

질의 응답

07.

끝

감사합니다