

오픈소스 **SW** 기여

개발자 가이드

32182775 위성준
32183698 이현기

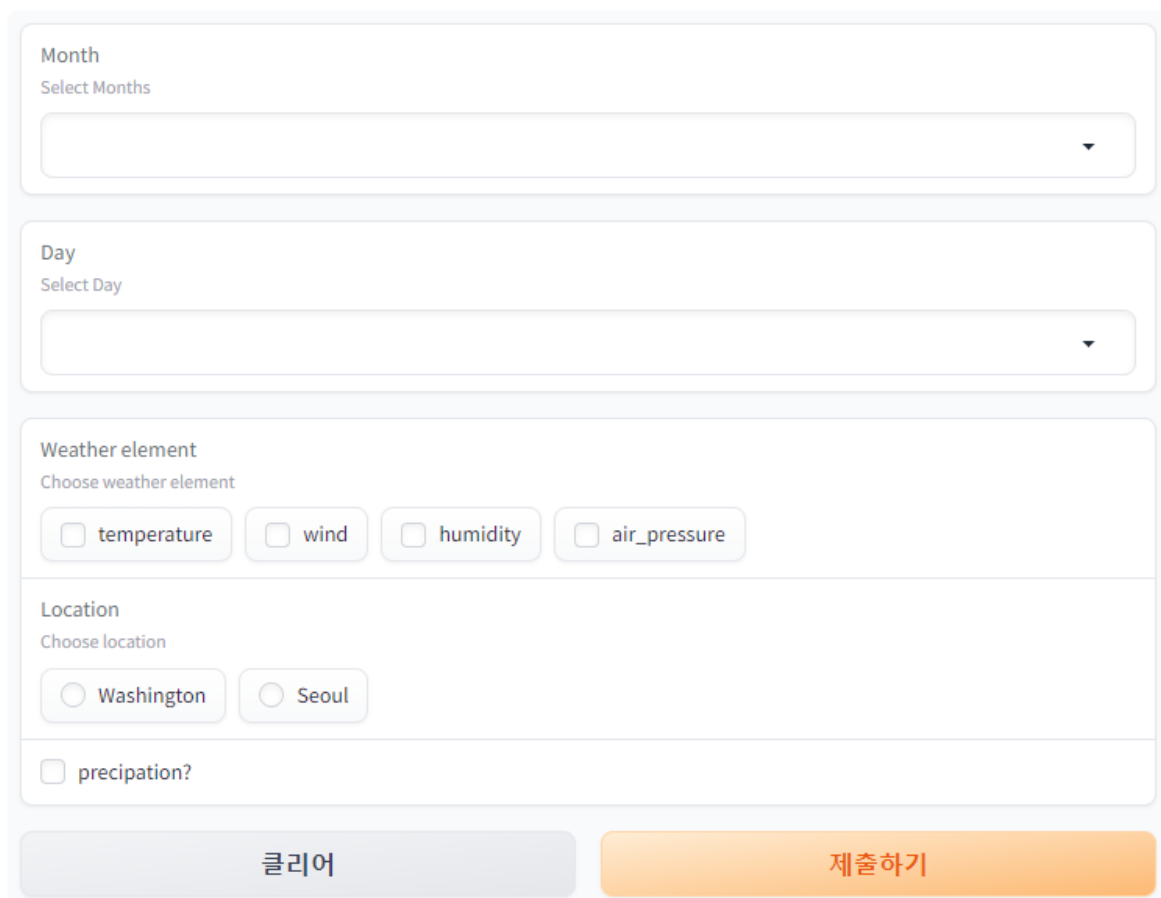
1) Gradio demo 요구사항

데모를 실행하기 위해서는 python 개발환경 내에 gradio, pandas, matplotlib 라이브러리를 설치한 상태여야 한다.

```
pip install gradio pandas matplotlib
```

명령 프롬프트에 해당 명령어를 입력하여 위 라이브러리들을 다운받을 수 있다.

2) 전체적인 시스템 구조



The image shows a Gradio demo interface with the following components:

- Month:** A dropdown menu with the label "Month" and the placeholder text "Select Months".
- Day:** A dropdown menu with the label "Day" and the placeholder text "Select Day".
- Weather element:** A section with the label "Weather element" and the placeholder text "Choose weather element". It contains four checkboxes: "temperature", "wind", "humidity", and "air_pressure".
- Location:** A section with the label "Location" and the placeholder text "Choose location". It contains two radio buttons: "Washington" and "Seoul".
- Precipitation:** A checkbox labeled "precipitation?".
- Buttons:** At the bottom, there are two buttons: "클리어" (Clear) and "제출하기" (Submit).

- Gradio components

Month, Day -> **Dropdown**

Weather elements (temperature, wind, humidity, air pressure)

-> **CheckboxGroup**

Location (Washington, Seoul) -> **Radio**

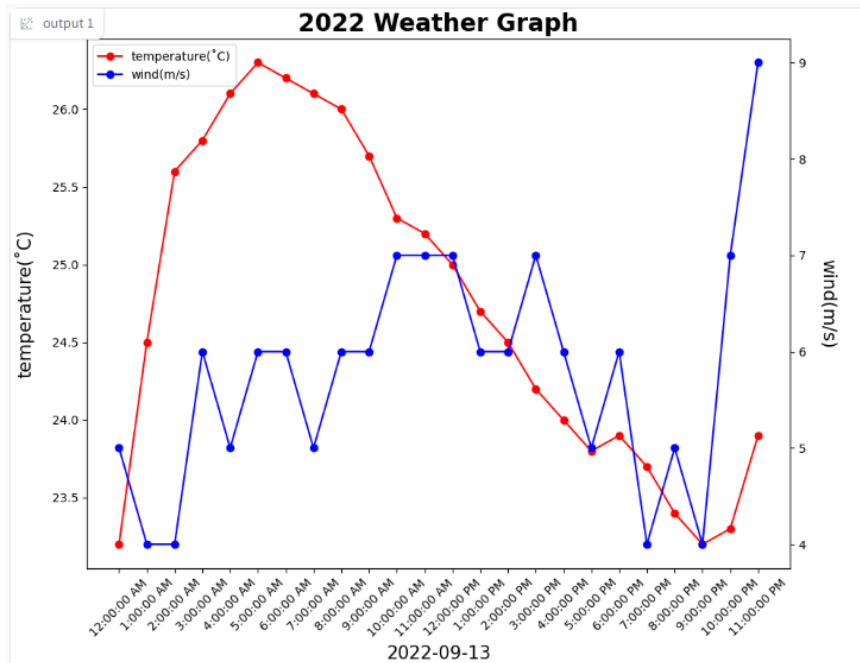
Precipitation -> **Checkbox**

- Matplotlib components

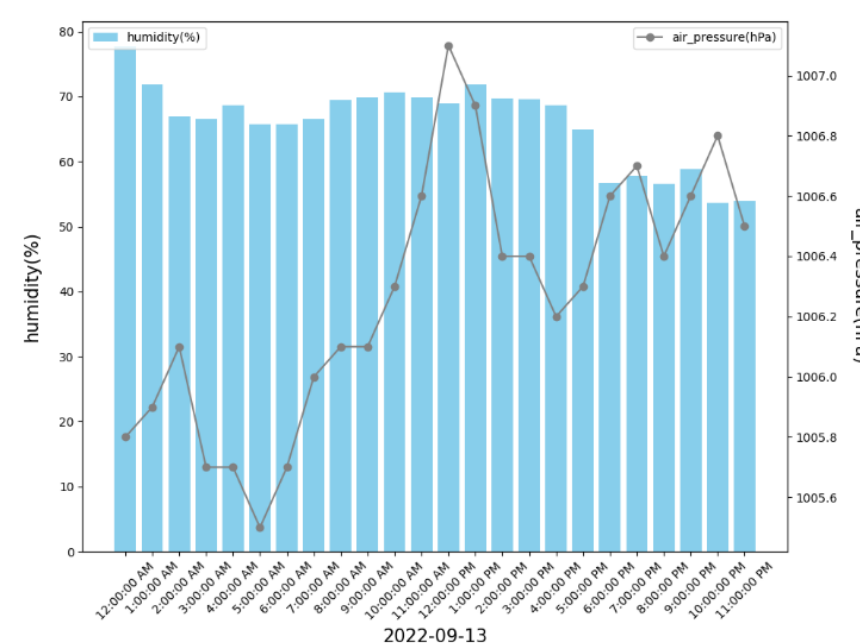
선 그래프 (기온, 풍속, 기압) -> matplotlib.pyplot의 **plot method** 이용.
막대 그래프 (습도) -> matplotlib.pyplot의 **bar method** 이용.

Plot 구성 -> 기상 요소 개수에 따라 그래프 추가 (최대 plot 2개)

예시) 1번 plot X축 : 시간, Y축 : 기온, 풍속



예시) 2번 plot X축 : 시간, Y축 : 습도, 기압



※ 강수량은 일일 데이터의 양이 적기 때문에 그래프에 표시 X.

title = 2022년 기상 그래프

xlabel = 해당 date, ylabel = 해당 weather element

x축 rotation = 45

- Pandas components

기상데이터 표 -> **DataFrame**

DataFrame 구성

index : hide, column : Location, date, time, Weather elements

데이터 전처리 후 Gradio를 이용하여 표 출력

| location ▲ | date ▲ | time ▲ | temperature(°C) ▲ | wind(m/s) ▲ | air_pressure(|
|------------|------------|-------------|-------------------|-------------|---------------|
| Seoul | 2022-09-13 | 12:00:00 AM | 23.2 | 5 | 1005.8 |
| Seoul | 2022-09-13 | 1:00:00 AM | 24.5 | 4 | 1005.9 |
| Seoul | 2022-09-13 | 2:00:00 AM | 25.6 | 4 | 1006.1 |
| Seoul | 2022-09-13 | 3:00:00 AM | 25.8 | 6 | 1005.7 |
| Seoul | 2022-09-13 | 4:00:00 AM | 26.1 | 5 | 1005.7 |
| Seoul | 2022-09-13 | 5:00:00 AM | 26.3 | 6 | 1005.5 |
| Seoul | 2022-09-13 | 6:00:00 AM | 26.2 | 6 | 1005.7 |
| Seoul | 2022-09-13 | 7:00:00 AM | 26.1 | 5 | 1006 |

- 사용자 정의 함수

1. 기상 데이터를 찾아주는 함수

dataSearch(Month, Day, Weather_elements, Location, Precipitation)

pandas로 기상데이터.csv 파일 오픈

date에 맞는 형식으로 문자열 생성

매개변수와 매칭되는 data들만 DataFrame 형태로 추출

return DataFrame

2. 그래프와 표를 출력하는 함수

showOutput(Month, Day, Weather_elements, Location, Precipitation)

영문 month를 숫자로 매핑

weatherTable = dataSearch(매핑된 지역변수들)

weatherTable 기반으로 matplotlib.pyplot 사용해서 weatherPlot 생성

return [DataFrame, figure]

위 컴포넌트들을 구성하여 `gradio.interface(showOutput, Gradio components, Matplotlib·Pandas components, examples)`로 gradio demo를 구성했다. examples는 입력 예시를 보여주는 테이블을 넣어 사용에 도움을 주는 역할을 한다. Demo를 다 구성했으면 `gradio.interface.launch()`로 실행한다.

3) 확장 방향

- **Weather element** 추가

지금 사용하고 있는 기상요소 외에 추가하고 싶은 기상요소에 대한 데이터를 기상자료개방포털 등에서 수집하여 워싱턴, 서울의 **csv** 파일에 추가 후, 코드 수정을 통해 확장 가능하다.

- **Location** 추가

현재 **Location**으로 사용한 곳은 워싱턴, 서울 2곳 뿐이다. 지역 확장을 원한다면 기상자료개방포털에서 원하는 지역의 **2022**년 기상자료를 **csv**파일로 다운받아 워싱턴, 서울의 **csv**파일과 같은 형식으로 전처리하고, 코드 수정을 통해 확장 가능하다.

- 연도 추가

2022년도를 제외한 다른 연도에 기상 데이터를 수집하여 워싱턴, 서울의 **csv**파일과 같은 형식으로 전처리하고, 코드를 수정하여 확장 가능하다.