# 4. Streamlit

**Understanding of Visualization** 





# Streamlit 소개





### Streamlit

#### ✓ Stremalit이란?

• 데이터 분석 및 시각화 애플리케이션을 빠르고 쉽게 구축할 수 있도록 도와주는 Python 라이브러리

#### ✓ 특징

- 간편성: 코드 작성이 간단하고 직관적이며, 데이터 분석 경험이 많지 않아도 손쉽게 사용할 수 있음
- 데이터 분석: 다양한 데이터 처리 및 분석 기능 제공
- 시각화: 다양한 차트와 그래프를 손쉽게 생성하여 데이터
   분석 결과를 효과적으로 표현할 수 있음
- 실시간 처리: 데이터 스트리밍을 지원하여 실시간 데이터 분석 가능
- 웹 애플리케이션: 데이터 분석 결과를 공유하기 위한 웹 애플리케이션 구축 가능







## Streamlit 장점

#### √ 간편성

- 코드 작성이 쉽고 직관적이며 손쉽게 사용 가능
- Pandas와 같은 다른 데이터 분석 라이브러리와도 쉽게 연동 가능
- ✓ 빠른 개발
  - 간단한 코드로 데이터 분석 및 시각화 애플리케이션을 빠르게 구축
  - 프로토타입 제작에 효과적

#### √ 인터랙티브

- 사용자와 상호 작용 가능한 웹 애플리케이션 제작 가능
- 슬라이더, 버튼, 텍스트 입력 등 다양한 위젯을 활용하여 사용자 참여를 유도할 수 있음

#### ✓ 공유

웹 애플리케이션 형태로 데이터 분석 결과를 쉽게 공유 팀원이나 클라이언트와 협업 용이







## Streamlit 활용 사례

#### ✓ 데이터 탐색 및 분석

- 데이터 분포 확인, 이상치 검출, 상관관계 분석 등을 통해 데이터를 탐색하고 분석
- 데이터 분석 결과를 시각화

#### ✓ 실시간 데이터 분석

데이터 변화 추이를 실시간으로 확인하고 모니터링할 수 있음

#### ✓ 웹 애플리케이션 및 대시보드 구축

- 데이터 분석 결과를 시각화하고 사용자와 공유하기 위한 웹 애플리케이션 및 대시보드를 쉽게 구축
- 사용자 맞춤형 대시보드를 제작하여 데이터 활용도를 높일 수 있음

#### √ 교육 및 연구

- 데이터 분석 개념을 쉽게 이해하고 실습하기 위한 교육 자료 및 연구 도구로도 활용
- 학습자들이 데이터 분석에 대한 이해도를 높이고 실제 문제 해결에 활용







# Streamlit 설치 및 주요기능

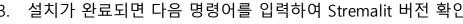




## Streamlit 설치

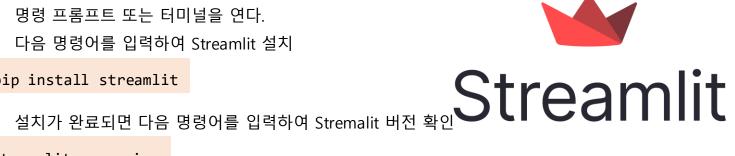
- ✓ 필수 조건: Python 3.6이상 설치
- ✓ 단계
  - 명령 프롬프트 또는 터미널을 연다.
  - 2. 다음 명령어를 입력하여 Streamlit 설치

pip install streamlit



streamlit --version

✓ 확인: Streamlit 버전이 출력되면 설치 완료







## Streamlit 설치

```
# hello_streamlit.py
import streamlit as st

# 앱 제목 설정
st.title("첫 번째 Steamlit 앱")

# 텍스트 출력
st.write("안녕하세요! Streamlit입니다.")

# 이미지 출력
st.image("image.jpg", with=200)
```



streamlit run hello\_streamlit.py





## 데이터 표시 방법

st.write() # 문자열 출력 st.markdown() # Markdown 문법을 사용하여 텍스트 서식 지정

st.image() # 이미지 파일 로드하여 출력

st.dataframe() # Pandas 데이터 프레임을 테이블 형식으로 출력

st.map() # 지도를 출력하고 데이터 포인트 표시





## 데이터 표시 방법

```
st.bar_chart() # 막대 그래프 생성
st.line_chart() # 선 그래프 생성
st.area_chart() # 면적 그래프 생성
st.altair_chart() # Altair 라이브러리를 사용하여 다양한 차트 생성
```





### 차트 예제

```
# hello_streamlit.py
import streamlit as st
# 앱 제목 설정
st.title("첫 번째 Streamlit 앱")
# 텍스트 출력
st.write("안녕하세요! Streamlit입니다.")
# 이미지 추력
st.image("image.jpg", with=200)
# 데이터 프레임 생성
data = {'name':['Alice', 'Bob', 'Charlie'], 'age': [30, 25, 22]}
df = pd.DataFrame(data)
# 데이터 프레임 출력
st.dataframe(df)
# 데이터 프레임 출력
st.bar_char(df['age'])
```

streamlit run hello\_streamlit.py





# 데이터 처리 및 시각화





## 처리가능 데이터 소스

#### ✓ 데이터 소스

- 1. CSV파일
- 2. JSON 파일
- 3. Pandas 데이터 프레임
- 4. NumPy 배열
- 5. SQL 데이터베이스
- 6. 웹 API







### **DataFrame**

```
import pandas as pd
import streamlit as st

# 데이터 불러오기
data = pd.read_csv('data.csv')

# 데이터 프레임 변환
df = pd.DataFrame(data)

# 데이터 프레임 출력
st.dataframe(df)
```







### 데이터 처리

```
# 데이터 프레임 정렬
sorted_df = df.sort_values(by='age')

# 데이터 프레임 집계
average_age = sorted_df['age'].mean()

# 정렬된 데이터 프레임 출력
st.dataframe(sorted_df)

# 평균 연력 출력
st.write('평균 연령: ', average_age)
```







# 데이터 시각화

```
import matplotlib.pyplot as plt
import seaborn as sns
import streamlit as st
# 막대 그래프 생성
plt.figure(figsize=(8, 6))
sns.barplot(data=df, x='gender', y='age', ci=None)
plt.xlabel('name')
plt.ylabel('age')
plt.title('성별별 연령 분포')
# streamlit에 그래프 표시
st.pyplot(plt.gcf())
```







# 사용자 입력





## 사용자 입력 위젯

- ✓ st.slider(): 슬라이더를 사용하여 값을 선택할 수 있도록 합니다.
- ✓ st.selectbox(): 드롭다운 메뉴를 사용하여 옵션을 선택할 수 있도록 합니다.
- ✓ st.radio\_buttons(): 라디오 버튼을 사용하여 옵션 중 하나를 선택할 수 있도록 합니다.
- ✓ st.checkbox(): 체크박스를 사용하여 값을 선택하거나 취소할 수 있도록 합니다.
- ✓ st.text\_input(): 텍스트를 입력할 수 있도록 합니다.
- ✓ st.date\_input(): 날짜를 선택할 수 있도록 합니다.





## 슬라이더

✓ 슬라이더를 사용하여 사용자가 'age' 값을 선택

```
import streamlit as st

# 슬라이더 생성

age_range = (18, 65)

selected_age = st.slider('나이 선택: ', age_range([0], age_range[1])

# 선택된 값 출력

st.write('선택된 나이: ', selected_age)
```





## 데이터 제어

✓ 앞의 selected\_age가 df['age']에 있는 경우에만 데이터 프레임을 출력

```
import pandas as pd
import streamlit as st
# 데이터 불러오기
data = pd.read_csv('data.csv')
df = pd.DataFrame(data)
# 조건부 실행
if selected_age in df['age'].values:
      st.dataframe(df)
else:
      st.write('해당 데이터는 존재하지 않습니다.')
```





# 파일 업로드

✓ csv 파일을 업로드 해서 출력

```
import streamlit as st
import pandas as pd
import time
# 파일 업로드 버튼 (업로드 가능)
file = st.file_uploader('파일 선택(csv or excel)', type = ['csv', 'xls', 'xlsx'])
time.sleep(3)
# Excel or CSV 확장자를 구분하여 출력하는 경우
if file is not None:
               ext = file.name.split('.')[-1]
               if ext == 'csv':
                              df = pd.read_csv(file) # 파일 읽기
                              st.dataframe(df) # 출력
               elif 'xls' in ext:
                              df = pd.read_excel(file, engine='openpyxl') # 엑셀 로드
                              st.dataframe(df) # 출력
```





# 사이드바 & 멀티페이지





## 사이드바

#### ✓ 사이드바 생성







## 멀티페이지

✓ 멀티페이지 생성

```
import streamlit as st

st.set_page_config(page_title='홈페이지', page_icon=':sunglasses:')

st.title('홈페이지')

st.write('안녕하세요! 이곳은 홈페이지입니다.')
```



```
✓ pages♣ page1.py♣ page2.py♣ page3.py♣ Hello.py
```



