

Streamlit?



Streamlit

- **python으로 데이터 분석을 위한 웹앱을 만들어주는 라이브러리**
 - 모델 배포, 그래프 시각화등 웹어플리케이션을 배포할수 있는 패키지
- **"데이터/ML 앱을 만들기 위한 가장 빠른 방법!"**
- **The fastest way to build and share data apps!**

Streamlit의 장점 및 한계

➤ 장점

- 백엔드 개발이나 HTTP 요청이 필요 없다.
- 다양한 입력 위젯 (컴포넌트) 제공.
- 간단히 배포 가능.
- 빠르고 쉽게 배울 수 있다.

➤ 한계

- Django, Flask, 등과 같은 웹 프레임워크를 대체하지는 못한다.
- 기본적으로 제공되는 기능을 초월하기가 어렵다.
- 실행이 효율적이지는 않다.

개발환경

➤ Anaconda 설치

- <https://www.anaconda.com>

➤ Visual Studio Code 설치

- <https://code.visualstudio.com>

➤ Streamlit 설치 및 참고

- <https://streamlit.io>
- `pip install streamlit`
 - Docs --> Get started --> Installation
 - Docs --> API reference

가상환경 만들기 및 삭제하기

- 가상환경을 만들면 프로젝트 별로 필요한 버전의 파이썬과 라이브러리를 설치할 수 있다.
- **base** 환경으로 가기
 - (STR) C:\Users\Streamlit> conda deactivate
 - (base) C:\Users\Streamlit>
- **가상환경 목록 보기**
 - conda env list
- **가상환경 STR 만들기**
 - (base) C:\Users\Streamlit> conda create -n STR python=3.9.13
 - (base) C:\Users\Streamlit> conda activate STR
 - (STR) C:\Users\Streamlit>
- **필요한 라이브러리들을 일괄 설치**
 - (STR) C:\Users\Streamlit> pip install -r ready.txt
- **가상환경 삭제**
 - (base) C:\Users\Streamlit> conda env remove -n STR

➤ 다시 가상환경 STR 만들기 (앞 장과 중복된 내용임)

- (base) C:\Users\₩Streamlit> conda create -n STR python=3.9.13
- (base) C:\Users\₩Streamlit> conda activate STR
- (STR) C:\Users\₩Streamlit>

➤ 가상환경 STR에 Streamlit 설치하기

- (STR) C:\Users\₩USER>pip install streamlit

➤ 앱 실행하기

- (STR) C:\Users\₩Streamlit> streamlit run myApp.py

Text elements

➤ 타이틀, 헤더, 서브헤더

- `st.title('Title *Markdown* 인식')`
- `st.header('Title *Markdown* 인식')`
- `st.subheader('Title *Markdown* 인식')`

➤ 텍스트, 마크다운

- `st.text('title *Markdown* 인식 못함.')`
- `st.markdown('*Markdown* 출력.')`

➤ 텍스트 또는 다양한 Python 변수/객체 출력.

- `st.text('This is some text.')`
- `x = 10`
- `y = 20`
- `st.write('x =', x, 'y =', y)`

Text elements

- `import pandas as pd`
- `df = pd.DataFrame({'col1': [1,2,3]})`
- `df`
- `st.write('데이터 프레임', df)`

➤ 그래프 출력

- `import matplotlib.pyplot as plt`
- `import numpy as np`
- `arr = np.random.normal(1, 1, size=100)`
- `fig, ax = plt.subplots()`
- `ax.hist(arr, bins=20)`
- `fig`

➤ 코드 출력.

- `code = '''def hello():`
- `print("Hello, Streamlit!")'''`
- `st.code(code, language='python')`

Text elements

➤ Markdown 텍스트 컬러 적용

- 'This is :blue[blue]'
- 'This is :red[red]'
- 'This is :green[green]'

➤ 캡션 출력.

- `st.caption('This')`
- `st.caption('A caption with _italics_ :blue[colors] and emojis :sunglasses:')`

➤ 이모티콘 삽입

- '여름엔 딱 좋아 :sunglasses:'
- ":100:점~ :smile:ㅎㅎ :thumbsup:최고!!"

➤ Streamlit에서 지원하는 이모티콘 단축 코드

- <https://streamlit-emoji-shortcodes-streamlit-app-gwckff.streamlit.app/>

앱 배포

- Streamlit cloud를 통해서 간편하게 앱을 배포할 수 있다.
- Github를 통해서 설치된다.
- 앱 배포 순서
 1. 가상환경 터미널에서 다음을 실행해서 설치가 필요한 라이브러리 목록을 만든다.
 - `pip freeze > ready.txt`
 - **주의:** 혹시 라이브러리 목록에 `certifi @ file:///private ~~~` 가 있으면 삭제한다.
 2. `www.github.com`으로 가서 계정을 열고 로그인 한다.
 - => 왼쪽 상단에서 [Create repository] 버튼을 클릭하고 새로운 repository를 만든다.

Create your first project

Ready to start building? Create a repository for a new idea or bring over an existing repository to keep contributing to it.

Create repository

Import repository

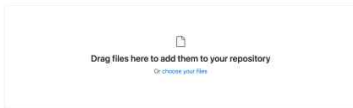
- 1) Repository name : **stapp**
- 2) Add a README file : **check**

앱 배포

3. 새롭게 만들어진 repository에서 [Add file]을 클릭하고 "Upload files"를 선택한다.



4. 앱 파일들을 drag & drop 하고 아래 [Commit changes]를 클릭한다.



앱 배포

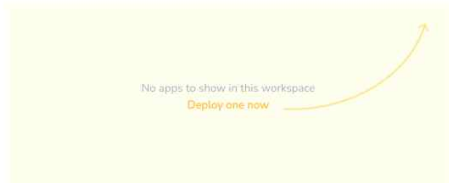
5. share.streamlit.io로 가서 [Continue with GitHub]를 선택하고 들어간다.
=> 처음이면 set up 절차가 있으니 따라서 한다.

Sign in



6. 다음 화면에서 [New app] 버튼을 누르고 GitHub repository와의 연결을 승인한다.

Your apps



앱 배포

7. 설정을 채워 넣고 [Deploy!] 버튼을 눌러서 완료한다.



The screenshot shows a deployment configuration form with three sections:

- Repository:** Includes a text input field containing "Si" followed by a path, and a button labeled "Paste GitHub URL".
- Branch:** Includes a text input field containing "main".
- Main file path:** Includes a text input field containing "iris.py".

8. 잠시의 기다림 이후 앱 배포가 완료된다. 앱의 URL을 기억해 둔다.
예). <https://encore-streamlitdata-app01-wqhf5x.streamlit.app/>

➤ 참고 사이트

- <https://streamlit.io>
- <https://streamlit.io/gallery>
- <https://blog.zarathu.com/posts/2023-02-01-streamlit/#streamlit-%EC%86%8C%EA%B0%9C>
- <https://towardsdatascience.com/streamlit-vs-dash-vs-shiny-vs-voila-vs-flask-vs-jupyter-24739ab5d569>