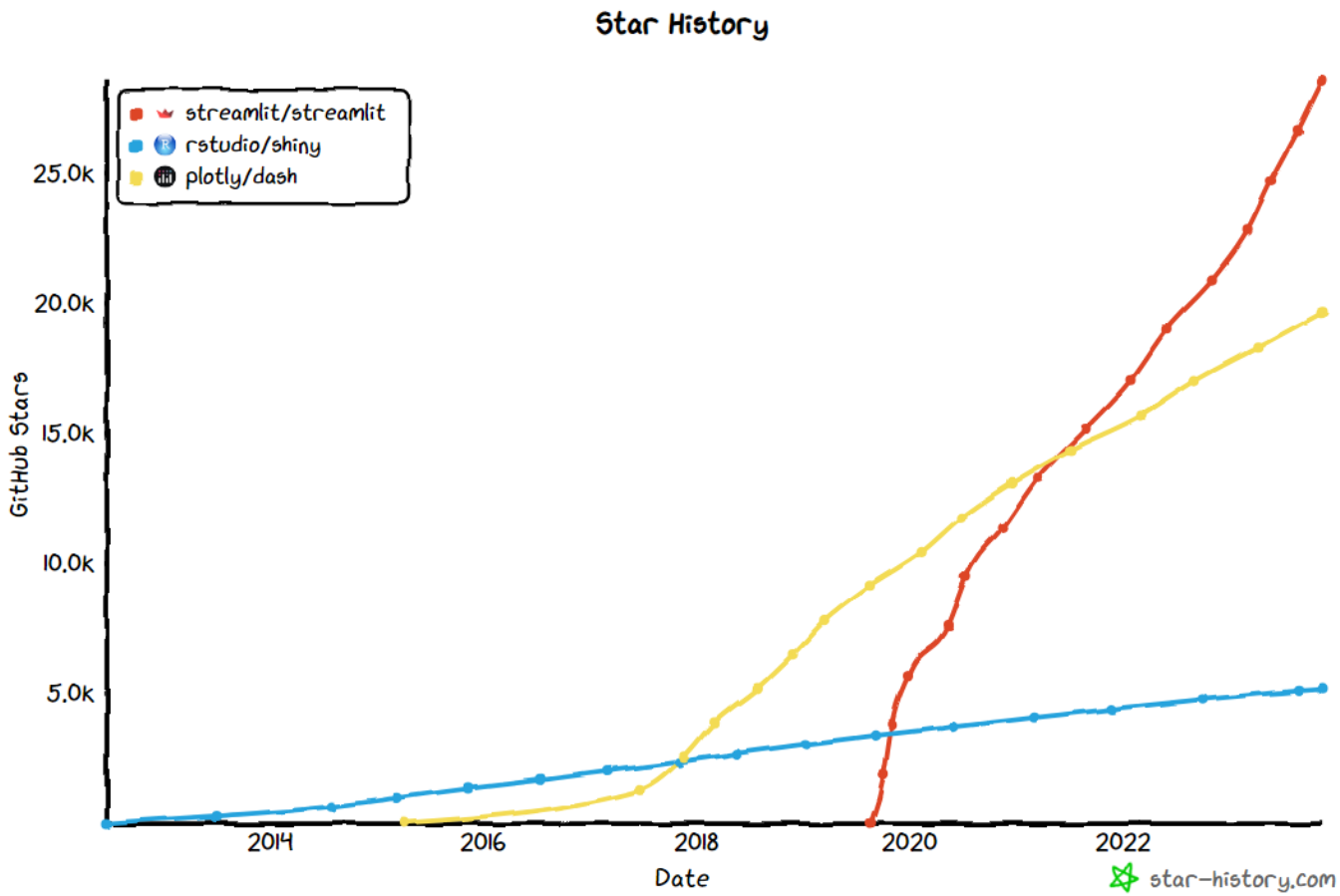
Streamlit

with  python™

01. Popularity

◆ Release

✓ 2019년 10월, 가장 빠르게 성장하는 Dashboard 프레임워크



02. Streamlit 탄생 스토리

◆ Streamlit 처음 설계한 창업자

- **CEO of Streamlit**
- **2022년 3월, Snowflake와 합병**
- **Computer Science Prof. at Carnegie Mellon**
- **Google X Project, VP at Zoox**

03. Streamlit 탄생 스토리

◆ Streamlit 처음 설계한 창업자

What if we could make building tools as easy as writing Python scripts?

"Adrien Treuille"

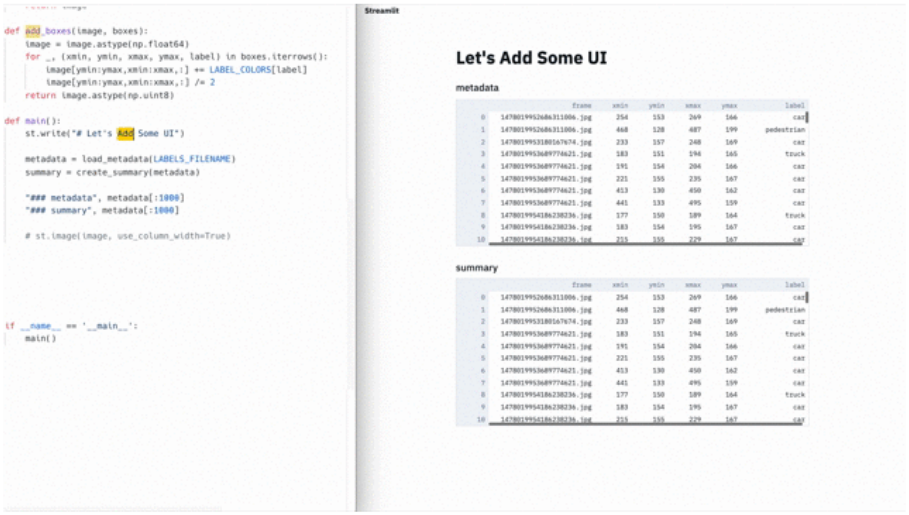
03. Streamlit 탄생 스토리

Adrien Treuille

Oct 1, 2019 · 7 min read · Member-only · Listen

Turn Python Scripts into Beautiful ML Tools

Introducing Streamlit, an app framework built for ML engineers



Coding a semantic search engine with real-time neural-net inference in 300 lines of Python.

In my experience, every nontrivial machine learning project is eventually stitched together with bug-ridden and unmaintainable internal tools. These tools — often a patchwork of Jupyter Notebooks and Flask apps — are difficult to deploy, require reasoning about client-server architecture, and don't integrate well with machine learning constructs like Tensorflow GPU sessions.

Adrien Treuille

2.1K Followers

Adrien is co-founder of Streamlit, the ML tooling framework. Adrien was a computer science prof at Carnegie Mellon, lead a Google X project, and was VP at Zoox.

Follow

More from Medium

Dennis Nig... in Python in Plain Engli...

Creating an Awesome Web App With Python and Streamlit

Frank Andra... in Towards Data Scie...

Predicting The FIFA World Cup 2022 With a Simple Model using Python

Moez Ali

Top AutoML Python libraries in 2022

Yang Zhou in TechToFreedom

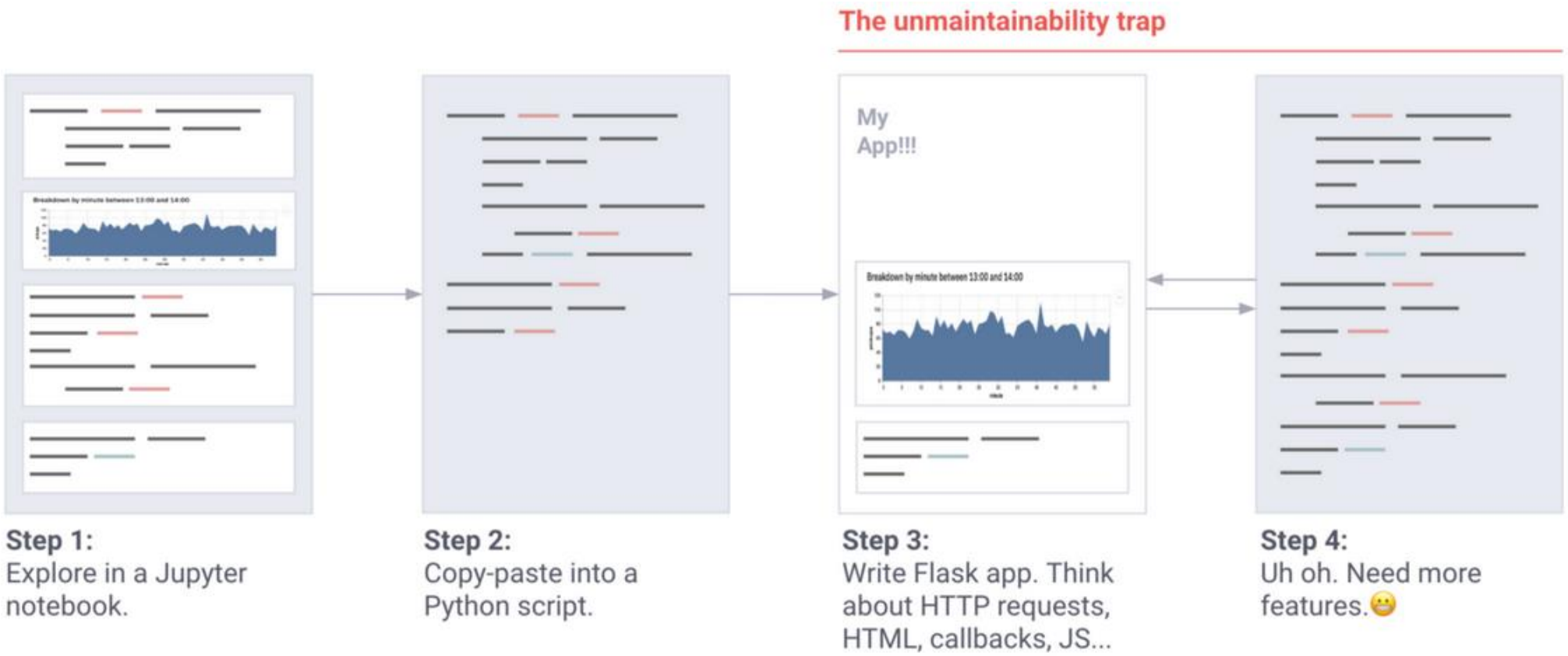
9 Fabulous Python Tricks That Make Your Code More Elegant

Release

- 2019년 10월 1일

03. Streamlit 탄생 스토리

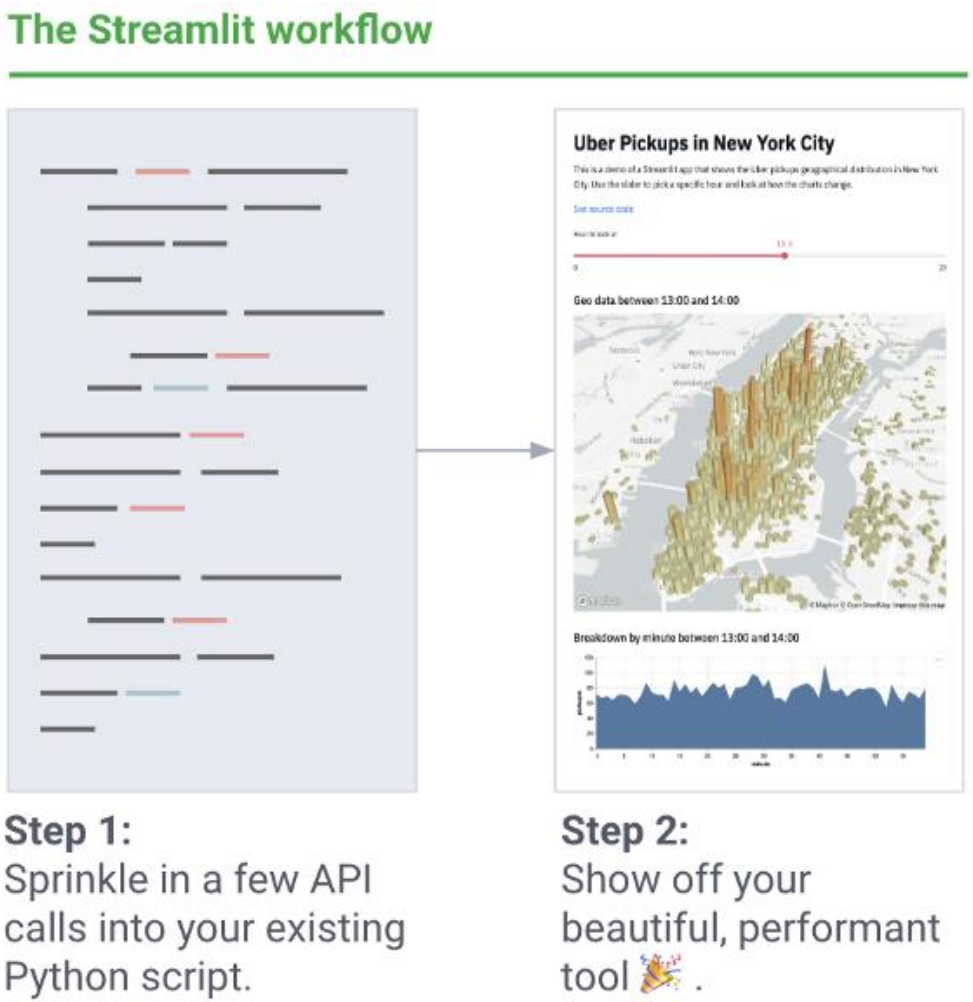
◆ 머신러닝 엔지니어의 가장 큰 문제점 (2019년 이전)

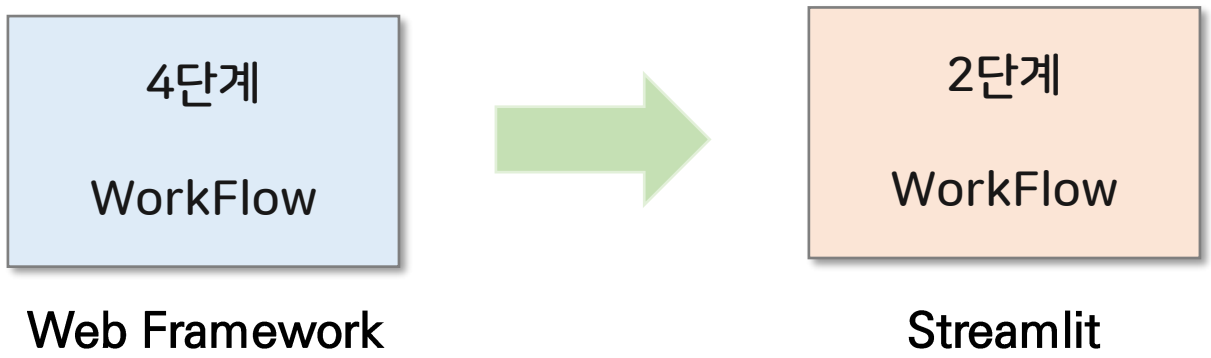


< The machine learning engineers' ad-hoc app building flow >

03. Streamlit 탄생 스토리

◆ Streamlit WorkFlow





04. Rules

◆ Scripting Workflow

✓ Coding just like on Google Colab or Jupyter Notebook For Data Analyst

기본 원칙	코드 예시
Embrace Python Scripting	>>> import streamlit as st >>> st.write('Hello, World!')

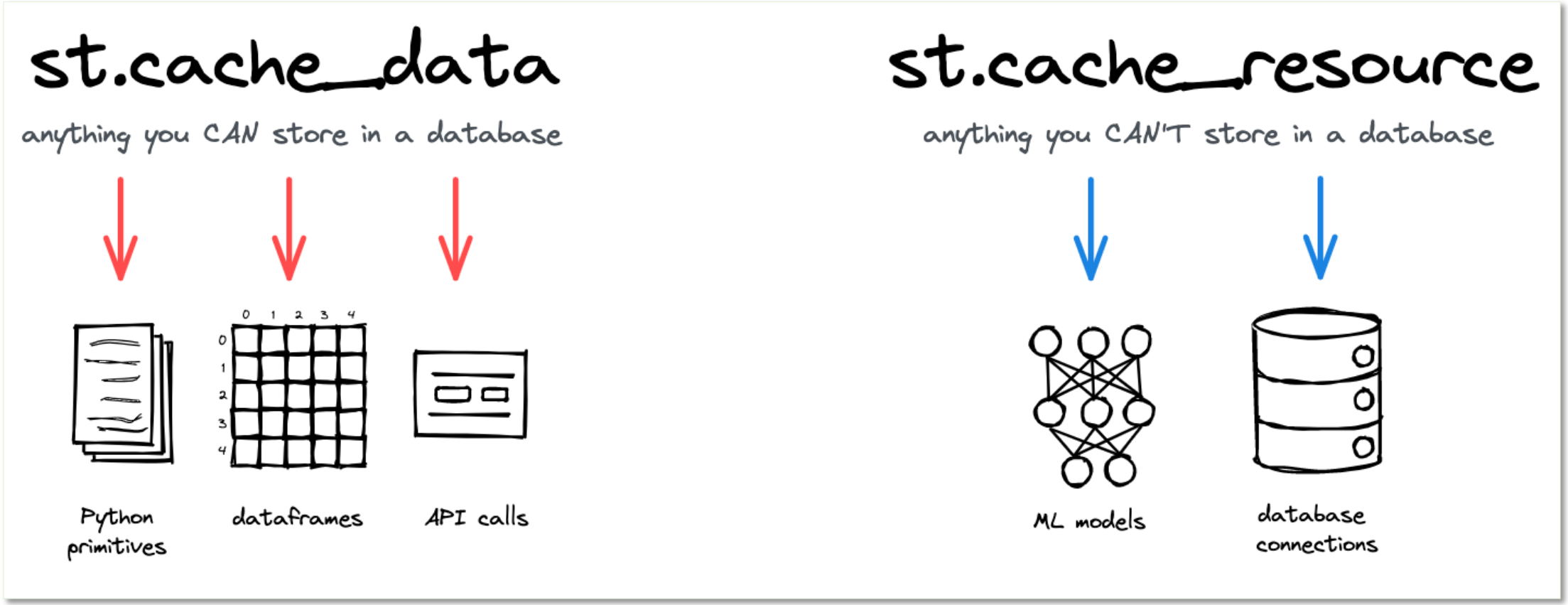
Treat widgets as variables	>>> import streamlit as st
	>>> x = st.slider('x')
	>>> st.write(x, 'squared is', x * x)

Reuse Data and Computation
Key : Cache (Persistent, Immutable by default)
	>>> import streamlit as st
	>>> import pandas as pd
	>>> data = pd.read_csv('iris.csv')
	>>> st.dataframe(data)

04. Rules

◆ Advanced Features Cache

✓ Just call long-running functions once and save it into session



04. Rules

◆ Advanced Features Cache

✓ 동일한 연산을 여러 번 수행하지 않아도 사용자 상호 작용이 빨라지고 웹 성능이 향상됨

✓ st.cache_data vs st.cache_resource

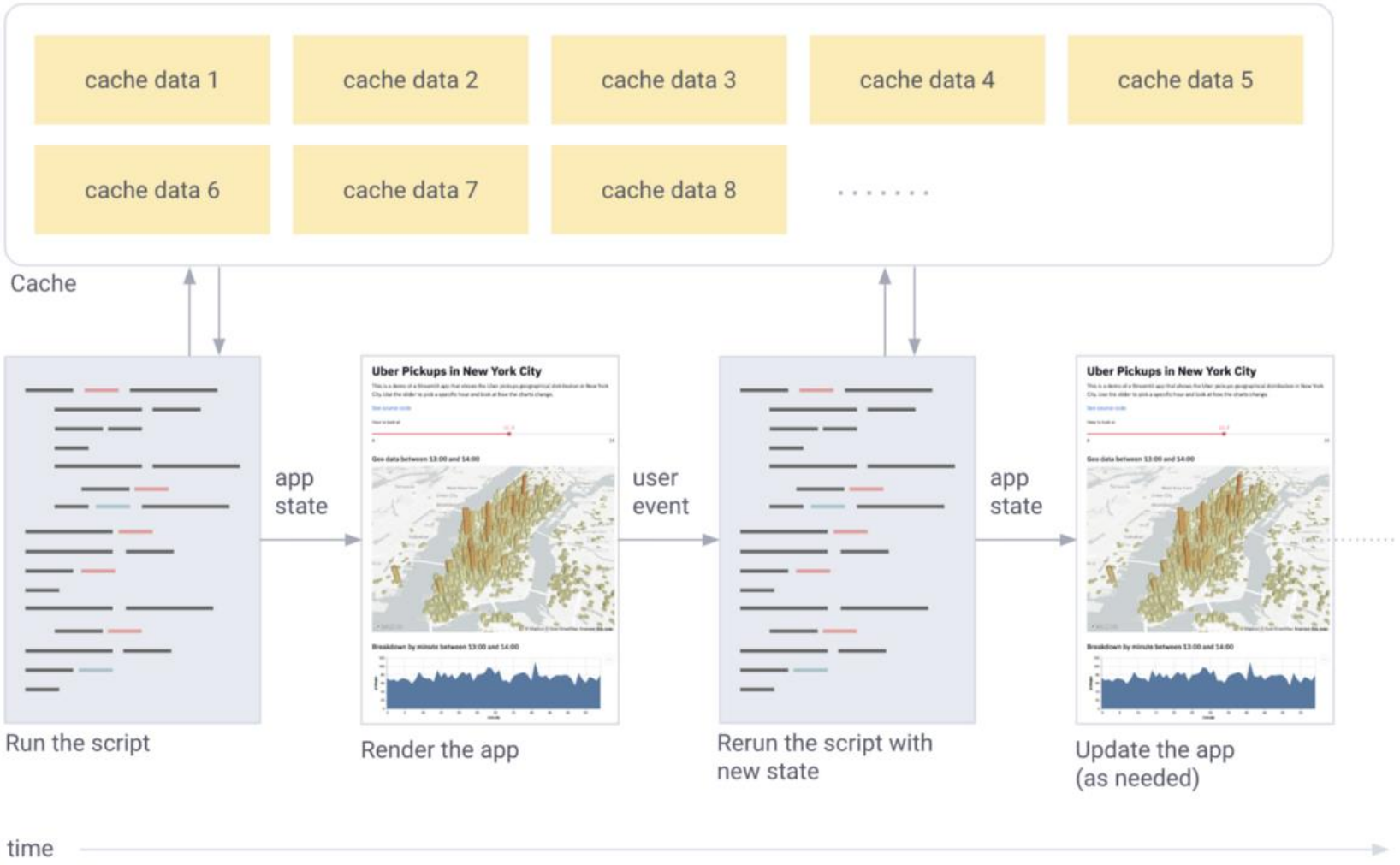
@st.cache_data	@st.cache_resource
✓ CSV 파일 불러오기 ✓ API 호출 ✓ NumPy 배열 변환 ✓ str, DataFrame, List 등이 함수 반환값	✓ ML Models or Database Connections

a serializable data object

unserializable objects

05. Cache Data

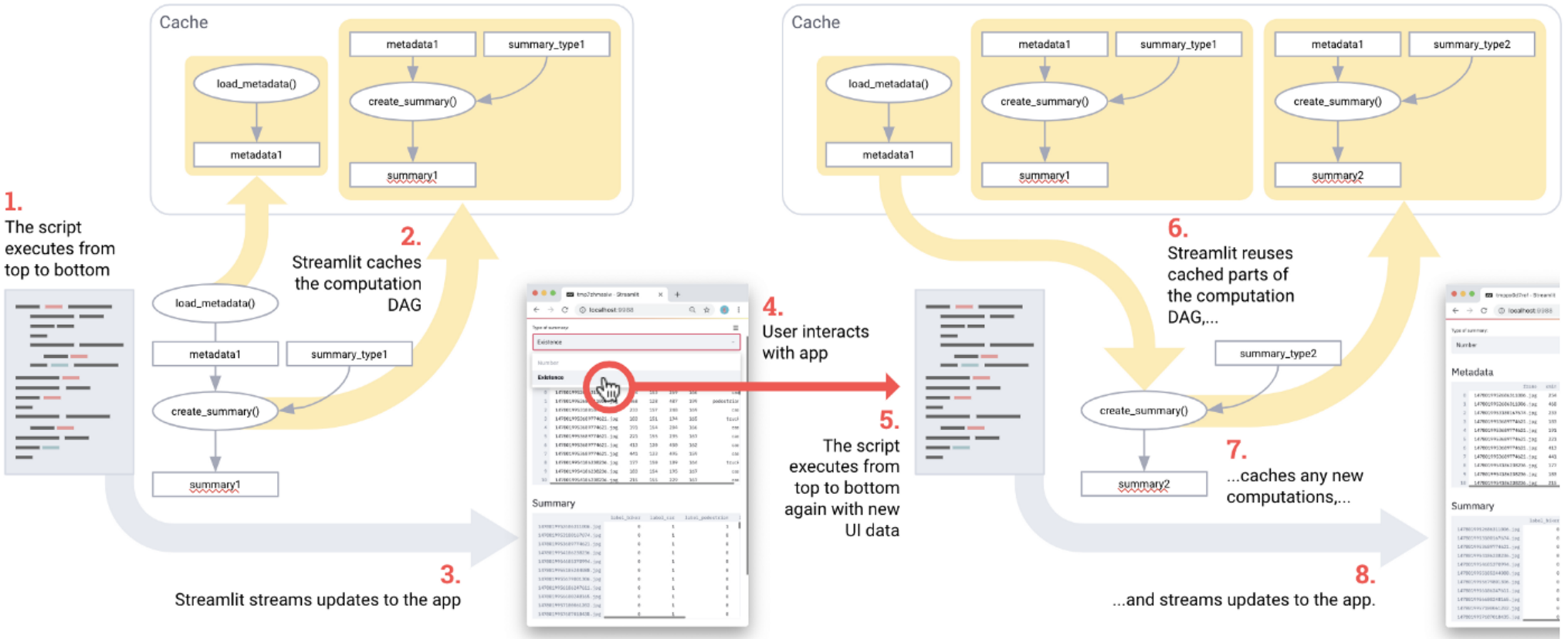
◆ Persisting across Running App



05. Cache Data

◆ Persisting across Running App

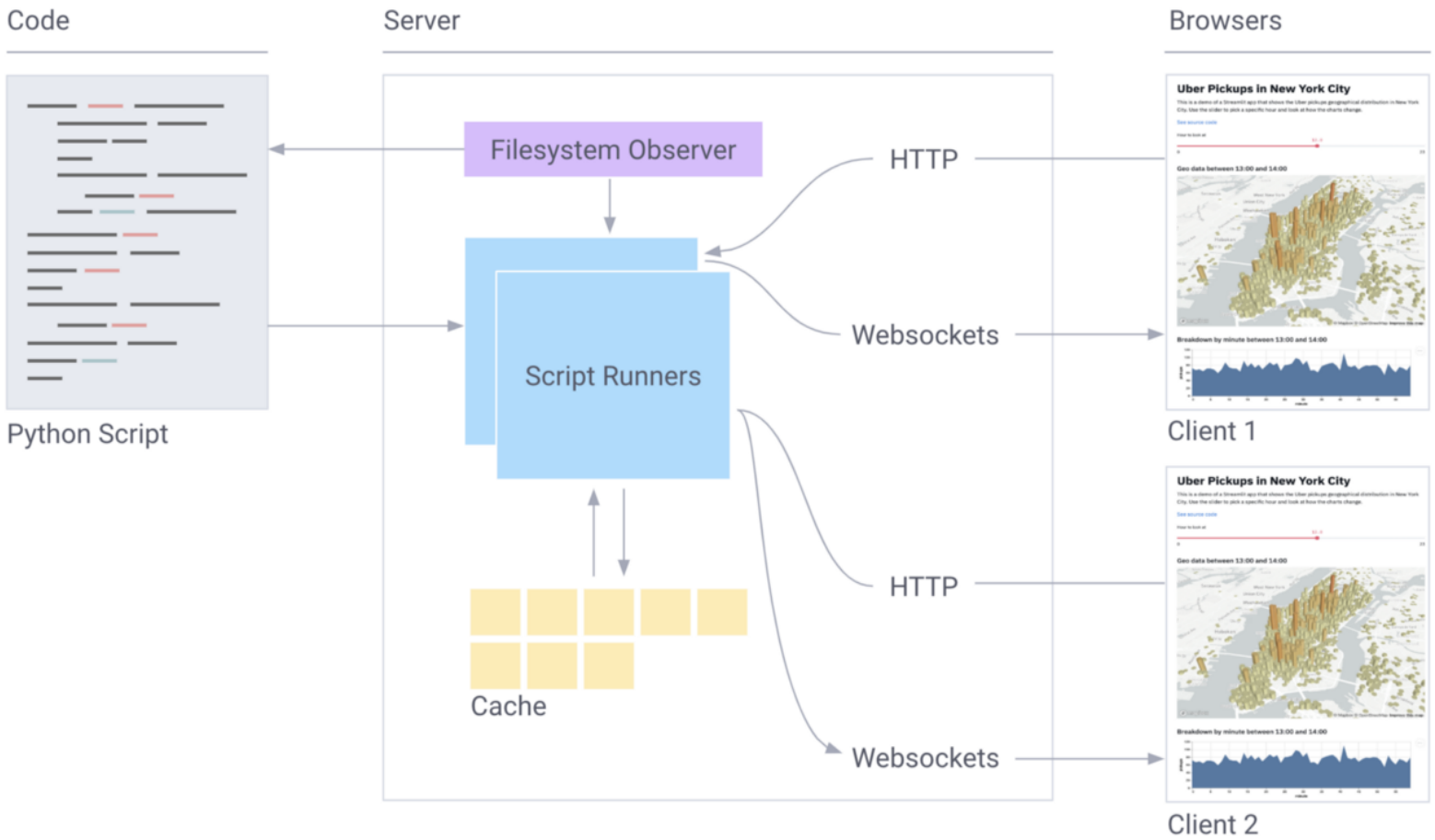
✓ UI가 변경될 때만, recomputing이 발생함



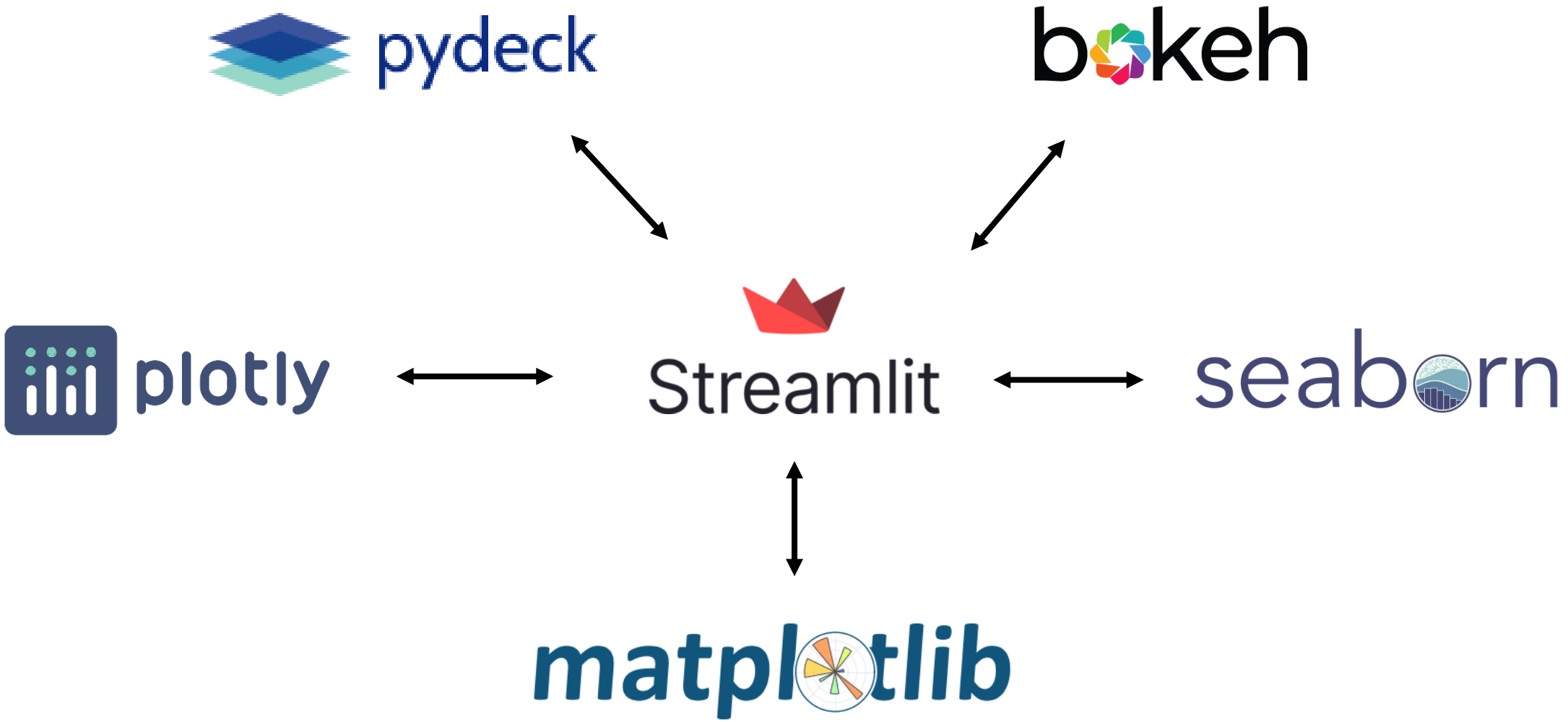
06. Block Diagram of Streamlit's Components

◆ Code → Server → Browsers

✓ Python Script → Cache → Client



07. Connect to Visualizations



08. Configuration

◆ 설정 시스템을 사용하면 Streamlit Web 동작을 사용자 지정할 수 있음

✓ Streamlit Configuration File은 '~/.strealmit/config.toml' 경로에 지정(Global Settings)

✓ Environment Variables

- `STREAMLIT_`으로 지정한다.

✓ 주요 sections and options

sections	설명
[server]	서버의 포트를 구성하고, Web 폴더의 기본 경로, CookieSecret 설정, CORS(Cross-Origin Resource Sharing) 설정
[theme]	웹의 기본 색상, 배경색 등의 옵션을 사용하여 Web의 형태 사용자 지정 가능
[browser]	실행할 기본 브라우저와 실행 동작 설정, IP address와 DNS name 설정 가능
[logger]	Logging 수준과, 로그 메시지의 대상을 정의
[runner]	경고 비활성화 또는 최대 메시지 크기 설정 등, Streamlit 스크립트가 실행하는 방식 조정

09. Interactive Widgets

◆ 기본 튜토리얼 확인 : https://docs.streamlit.io/library/cheatsheet

Home / Streamlit library / **Cheat sheet**

Cheat Sheet

This is a summary of the docs, as of Streamlit v1.28.0.

Install & Import

```
streamlit run first_app.py

# Import convention
>>> import streamlit as st
```

Command line

```
streamlit --help
streamlit run your_script.py
streamlit hello
streamlit config show
streamlit cache clear
streamlit docs
streamlit --version
```

Pre-release features

```
pip uninstall streamlit
pip install streamlit-nightly -
```

Learn more about experimental features

Magic commands

```
# Magic commands implicitly
# call st.write().
'_This_ is some **Markdown***'
my_variable
'dataframe:', my_data_frame
```

Control flow

```
# Stop execution immediately:
st.stop()
# Rerun script immediately:
st.rerun()

# Group multiple widgets:
>>> with st.form(key='my_form'
>>>     username = st.text_input
>>>     password = st.text_input
>>>     st.form_submit_button('L
```

Connect to data sources

```
st.connection("pets_db", type='
conn = st.connection("sql")
conn = st.connection("snowflak

>>> class MyConnection(BaseCon
>>>     def _connect(self, **kw
>>>         return myconn.conne
>>>     def query(self, query):
>>>         return self._instanc
```

Display text

```
st.text('Fixed width text')
st.markdown('_Markdown_') # see
```

Streamlit

강의 실습 영상 참고