

【Day 8】 Streamlit 簡介

Streamlit 是一個為資料科學、視覺化打造的 Web 框架，它的機制強調直覺的代碼與快速部署原型。這篇文章會以介紹 Streamlit 的功能為主，不太深入代碼部分。範例都來自文檔。[\[文檔連結\]\(https://docs.streamlit.io\)](https://docs.streamlit.io)

1. Rerun 機制

1.1 session state

在 Streamlit 中，每當使用者與介面互動（例如點擊按鈕、輸入文字），整個 Python 文件就會從頭重新執行一次，這樣的好處是，我們不用考慮事件循環，寫代碼時，只要考慮頁面上有什麼即可。但因為 rerun 的關係，變量需要透過 `streamlit.session_state` 來保存。我們通常這樣加入一個 session state

```
import streamlit as st

if 'key' not in st.session_state:
    st.session_state['key'] = 'value'
```

1.2 cache

除此之外，我們還可以利用 `@st.cache_data` 和 `@st.cache_resource` 來存一些需要在 session 保持的資料。

`@st.cache_data`

- 適合快取可序列化的資料（如字串、數字、DataFrame、dict、list）。
- 每次呼叫函式都會建立新副本，避免修改跟 race conditions。
- 不知道用什麼，建議從這個開始。

`@st.cache_resource`

- 適合快取全域資源（如 ML 模型、資料庫連線）。
- 適用於不可序列化的物件，避免重複載入。
- 直接回傳快取物件本身，所有 session 共用。
- 若修改該物件，變動會在所有 session 保持。

2. 常用組件介紹

Streamlit 提供了大量互動元件，也有第三方的組件可以安裝。然而，如果想要的功能 API 沒有提供也比較難以客制化。以下整理最常見的：

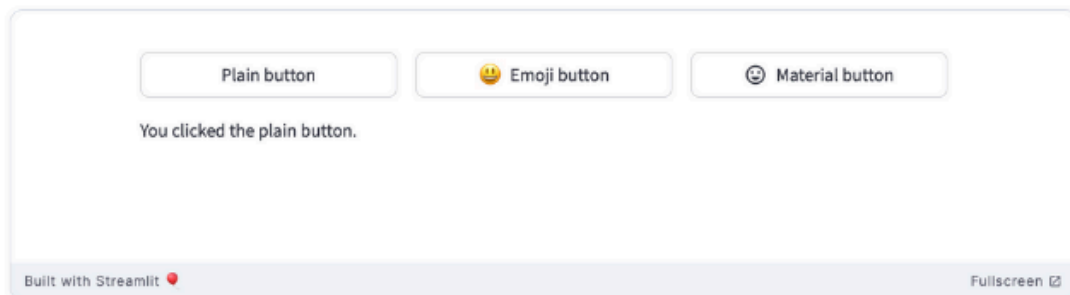
(a) 輸出元素

`st.write` , `st.markdown` , `st.stream_write` (用來串流輸出)

(b) 輸入元件

`st.button` , `st.select_slider` , `st.segmented_control`

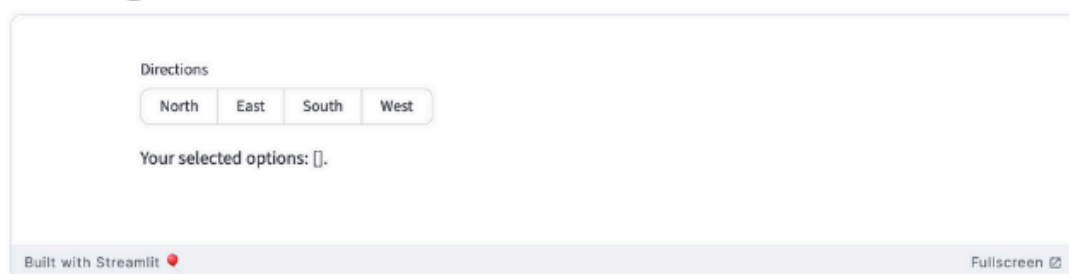
st.button



st.select_slider

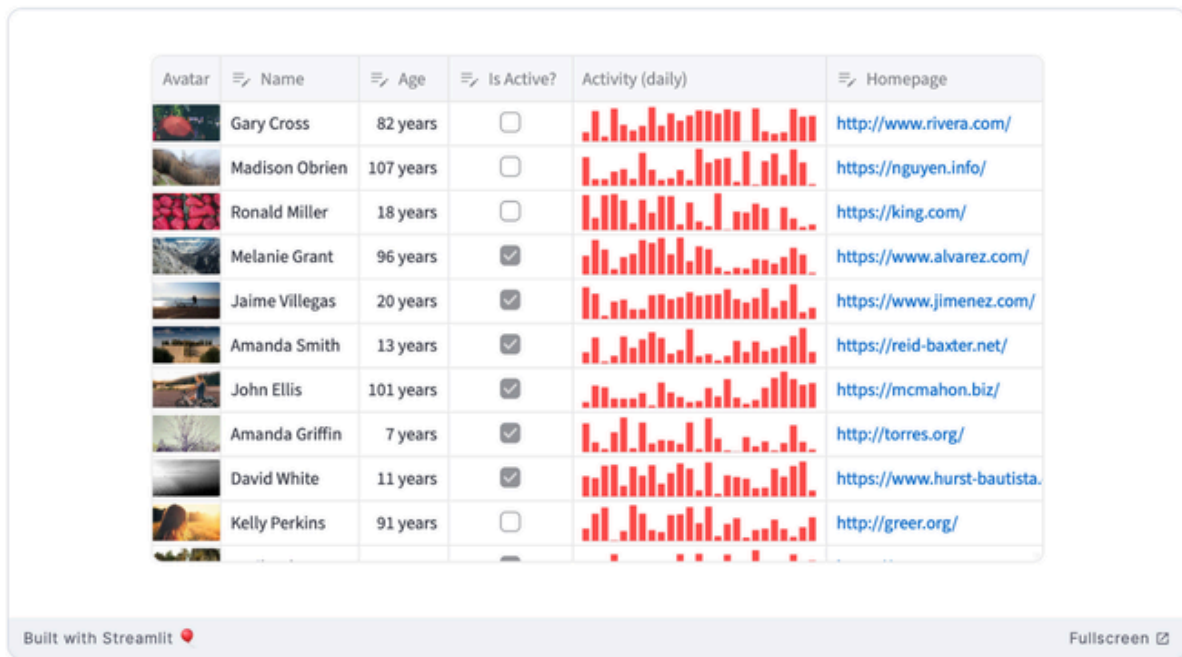


st.segmented_control



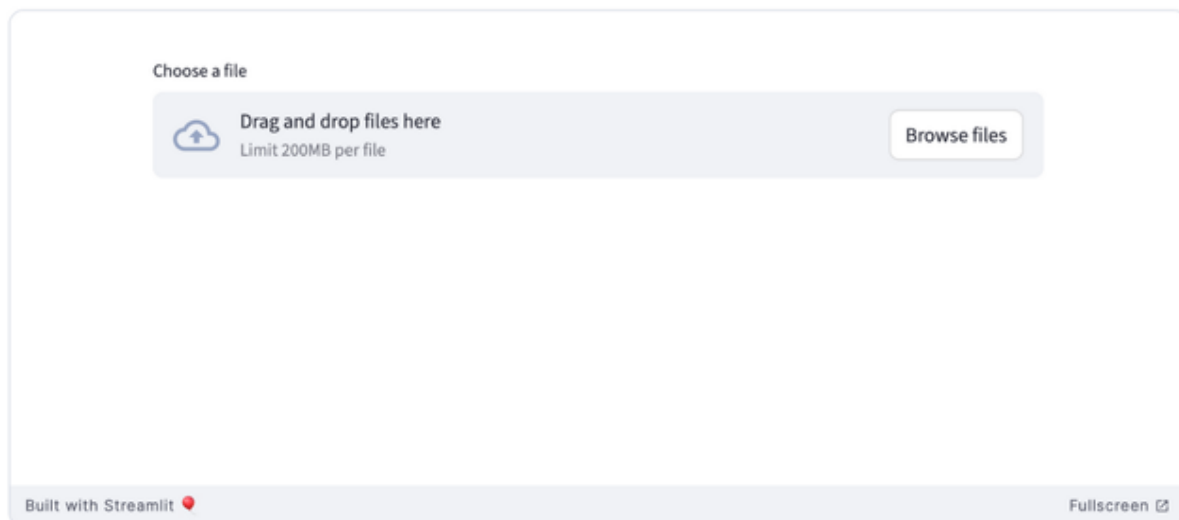
(c) 視覺化

`st.dataframe` 可以直接顯示 `pandas.DataFrame`，透過 column configuration 還能在 dataframe 中顯示圖片、圖表等等。之外，streamlit 也能畫 **line chart**, **pie chart** 等基本圖片，甚至直接顯示 `matplotlib.pyplot` 的圖片。



(d) 檔案處理

`st.file_uploader`



(e) Chat 元件 (`st.chat` 系列)

Streamlit 在 1.25 版本後，加入了原生的聊天介面，非常適合做 LLM 對話應用。

- `st.chat_message("role")` → 建立訊息氣泡，`role` 可為 `"user"` 或 `"assistant"`
- `st.chat_input("提示字")` → 建立輸入框

範例：

```

import streamlit as st

def response_generator():
    ... # 略
    yield

st.title("Simple chat")

# 創建對話歷史
if "messages" not in st.session_state:
    st.session_state.messages = []

# 展示對話歷史
for message in st.session_state.messages:
    with st.chat_message(message["role"]):
        st.markdown(message["content"])

# 接收 Human Message
if prompt := st.chat_input("What is up?"):

    # 寫 user 訊息
    with st.chat_message("user"):
        st.markdown(prompt)

    # 寫 assistant 訊息
    with st.chat_message("assistant"):
        response = st.write_stream(response_generator())

    # 把訊息加入 session_state
    st.session_state.messages.append({"role": "user", "content": prompt})

```

Simple chat




hello



Hello there! How can I assist you today?

What is up?



Built with Streamlit 

Fullscreen 