



Which number do you like best?

1

- 1
- 2
- 3
- 4

Get started

Streamlit là công cụ giúp chúng ta build nhanh một web app cho mục đích demo hoặc các dự án nhỏ. Mục tiêu là sử dụng Streamlit để tạo ứng dụng tương tác cho dữ liệu hoặc mô hình của bạn và trong quá trình sử dụng Streamlit để xem xét, gỡ lỗi, hoàn thiện và chia sẻ mã của bạn.

Tạo app streamlit đầu tiên của bạn

Đầu tiên chúng ta sẽ tạo một script Python và import Streamlit

1. Tạo **first_app.py**.
2. Import streamlit

```
import streamlit as st
import numpy as np
import pandas as pd
```

3. Chạy app của bạn với terminal dòng lệnh sau:

```
streamlit run first_app.py
```

4. Nhấn **CTRL+C** để hủy terminal

Thêm text và data

Thêm text

Streamlit có một số cách để thêm text vào ứng dụng của bạn. Check <https://docs.streamlit.io/en/stable/api.html> để có danh sách đầy đủ

Bây giờ chúng ta thêm văn bản vào bằng cú pháp sau:

```
st.title("My first app")
```

My first app

Sau đó app của bạn đã có tiêu đề. Bạn có thể sử dụng một số hàm text đặc biệt để thêm content vào app của bạn, hoặc bạn có thể sử dụng **st.write()** và thêm vào markdown

Tạo dataframe

```
st.write("Here's our first attempt at using data to create a table:")
st.write(pd.DataFrame({
    'first column': [1, 2, 3, 4],
    'second column': [10, 20, 30, 40]
}))
```

Here's our first attempt at using data to create a table:

	first column	second column
0	1	10
1	2	20
2	3	30
3	4	40

Ngoài ra thì bạn cũng có thể sử dụng các hàm như **st.dataframe()** và **st.table()** để hiển thị data.

Use magic

Bạn có thể dễ dàng viết app mà không cần gọi các phương thức của Streamlit. Streamlit hỗ trợ sử dụng magic command, có nghĩa là bạn không cần sử dụng **st.write()**. Thử code sau:

```
df = pd.DataFrame({
    'first_column': [1,2,3,4],
    'second_column': [10,20,30,40]
})
df
```

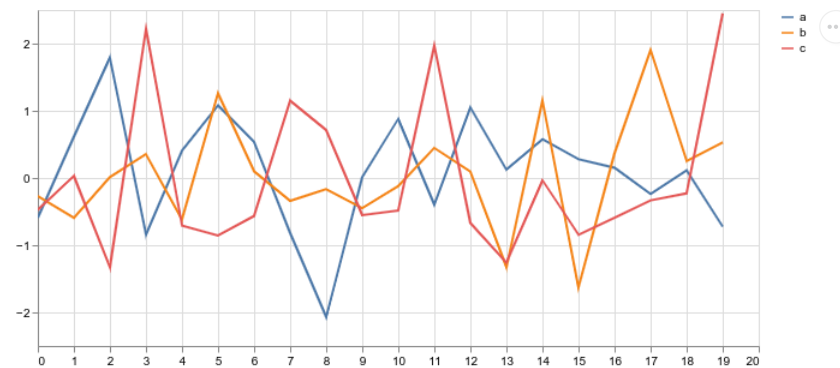
	first_column	second_column
0	1	10
1	2	20
2	3	30
3	4	40

Draw charts and maps

Streamlit hỗ trợ các thư viện để mô phỏng dữ liệu như **matplotlib**, **altair**, **deck.gl**, Trong phần này, chúng ta sẽ thêm một bar chart, line chart:

Draw a line chart

```
chart_data = pd.DataFrame(
    np.random.randn(20,3),
    columns=['a', 'b', 'c']
)
st.line_chart(chart_data)
```



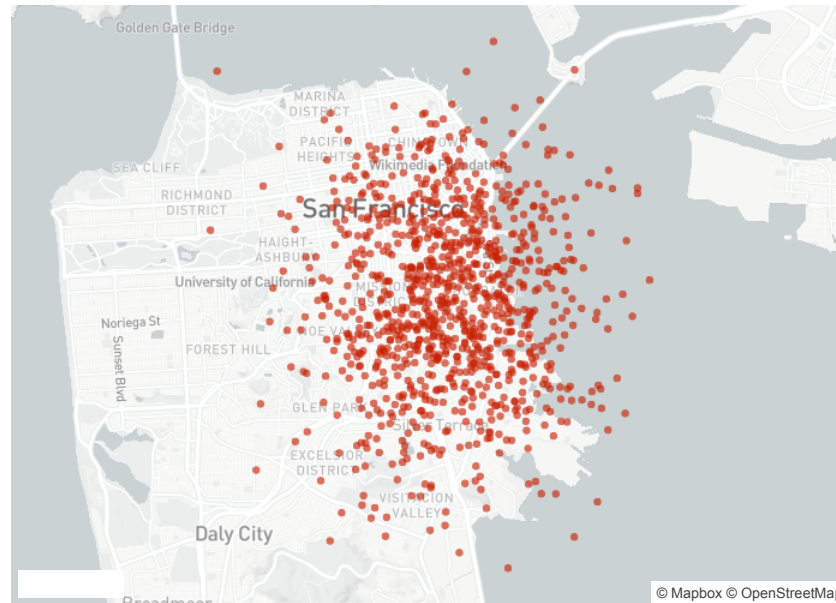
Plot a map

```
map_data = pd.DataFrame(
```

```

np.random.randn(1000,2)/[50, 50] + [37.76, -122.4],
columns=['lat', 'lon']
)
st.map(map_data)

```



Thêm tương tác với các widget

Với widgets, Streamlit cho phép bạn thêm các tương tác cho app của bạn với checkboxes, buttons, sliders, ...

Sử dụng checkboxes cho hiện/ẩn data

```

if st.checkbox("Show dataframe"):
    char_data = pd.DataFrame(
        np.random.randn(20, 3),
        columns=['a', 'b', 'c']
    )
    char_data

```

☐ Show dataframe

Sử dụng selectbox cho các option

Đoạn code dưới đây sẽ tạo thêm các lựa chọn ở bên phải trái trang của bạn

```

option = st.sidebar.selectbox(
    'Which number do you like best?',
    df['first_column']
)
'Your selected:', option

```

Your selected: 1

Nhiều phần tử bạn có thể thêm vào app của bạn vào trong sidebar sử dụng cú pháp: **st.sidebar.[element_name]()**. Dưới đây là một số ví dụ sử dụng nó như thế nào: **st.sidebar.markdown()**, **st.sidebar.slider()**, **st.sidebar.line_char()**



Bạn cũng có thể sử dụng **st.beta_columns** để sắp xếp các widget cạnh nhau hoặc **st.beta_expander** để tiết kiệm dung lượng bằng cách ẩn đi nội dung lớn.

```
left_column, right_column = st.beta_columns(2)
pressed = left_column.button("Press me?")
if pressed:
    right_column.write("Woohoo!")
expander = st.beta_expander("FAQ")
expander.write("Here toy could put in some really")
```

Press me?

FAQ

+

Show process

Khi mà chạy tính toán trên app, bạn có thể sử dụng **st.process()** để hiển thị tiến trình chạy real time

```
import time
'Starting a long computation...'
# Add a placeholder
lastes_iteration = st.empty()
bar = st.progress(0)

for i in range(100):
    # update the process bar with each iteration
    lastes_iteration.text(f'Iteration {i+1}')
    bar.progress(i+1)
    time.sleep(0.2)

'... and now we\'re done!'
```

Starting a long computation...

Iteration 100

... and now we're done!