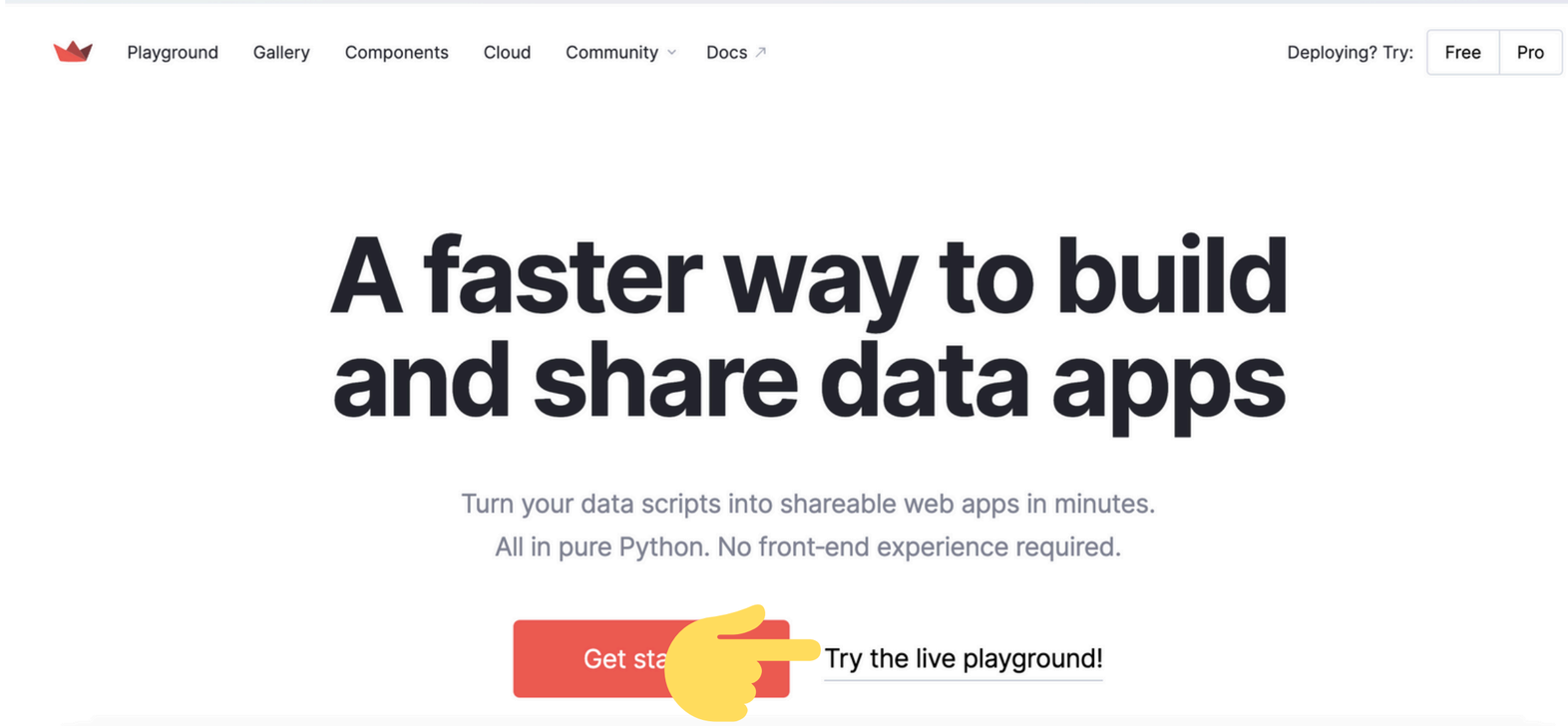
Nano Movie Recommender System

CLASS 3 - PART 3
HOW TO SET UP THE DEVELOPMENT ENVIRONMENT (STREAMLIT)

Yaoyao(Freax) Qian
h-freax.github.io

What is Streamlit?

https://streamlit.io



Playground

Try out a limited version of Streamlit right in your browser.
Just edit the code below and the app on the right updates automatically.
For the real thing, **install our Python library**.

EXAMPLES Blank Hello Charts Dataframes LLM chat Computer vision Geospatial

```python
import streamlit as st

st.title("Hello Streamlit-er👋")
st.markdown(
    """
    This is a playground for you to try Streamlit and have fun.

    **There's :rainbow[so much] you can build!**

    We prepared a few examples for you to get started. Just
    click on the buttons above and discover what you can do
    with Streamlit.
    """
)

if st.button("Send balloons!"):
    st.balloons()
```

Share

Hello Streamlit-er👋

This is a playground for you to try Streamlit and have fun.

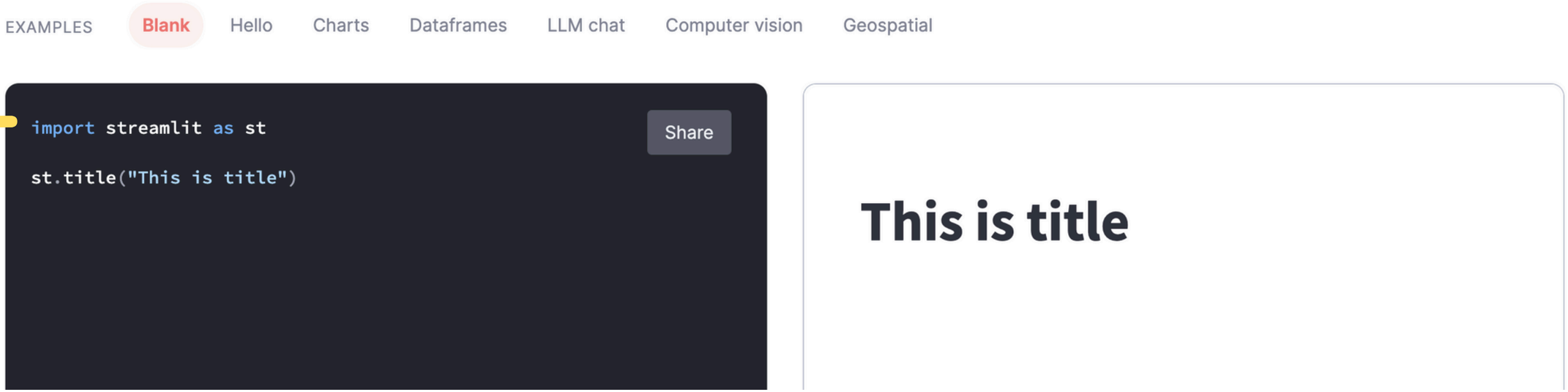
There's so much you can build!

We prepared a few examples for you to get started. Just click on the buttons above and discover what you can do with Streamlit.

Send balloons!

EDIT HERE

APP



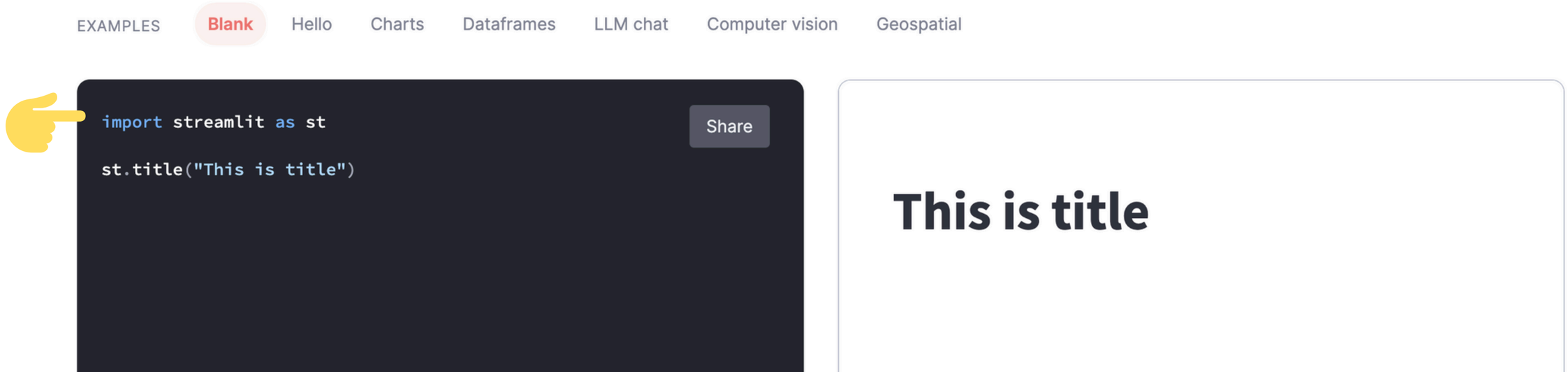
This line of code tells Python that you want to use the streamlit tool.

✅ Analogy:

You download an app on your phone (streamlit is installed).

But before using it, you need to open the app.

import streamlit as st is like opening the Streamlit app in Python.



What is import?

🔹 import is a command in Python that allows you to use a tool (library).

✅ Analogy:

You have a rice cooker at home.

But if you don't plug it in, it won't work.

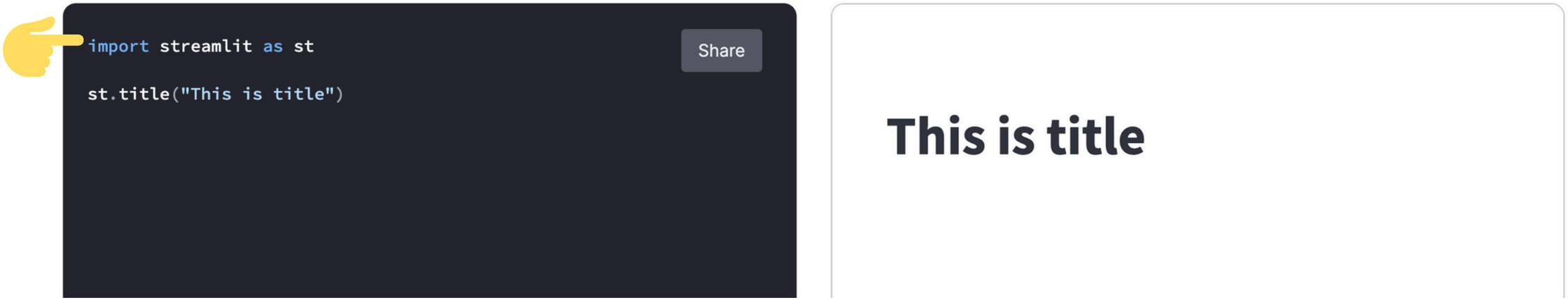
import is like "plugging in" the rice cooker, so Python knows you want to use streamlit.

📌 In Python, many useful tools (like streamlit) must be imported before you can use them.

EXAMPLES Blank Hello Charts Dataframes LLM chat Computer vision Geospatial

```
import streamlit as st

st.title("This is title")
```

Share



What does as st do?

- as st is a shortcut to rename streamlit as st, making it easier to use.

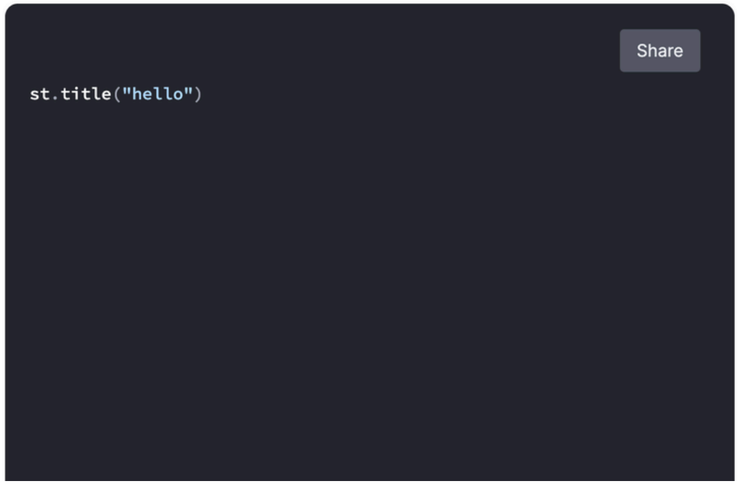
✅ Analogy:

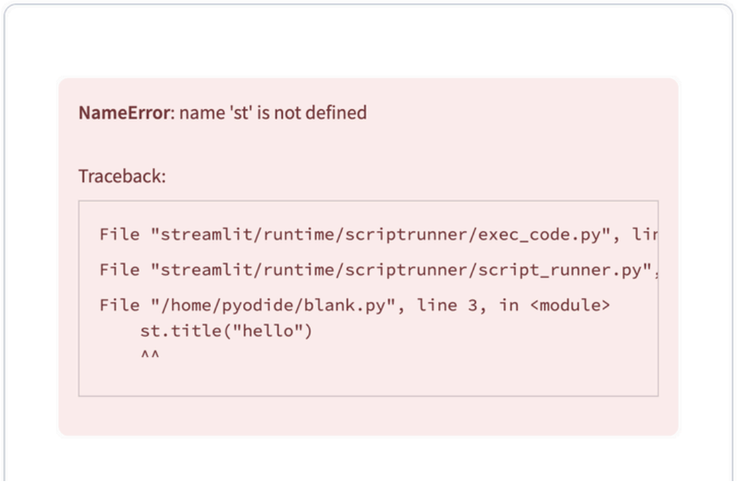
Imagine your friend's full name is "Alexander Smith".

You don't want to say the full name every time, so you call him "Alex".

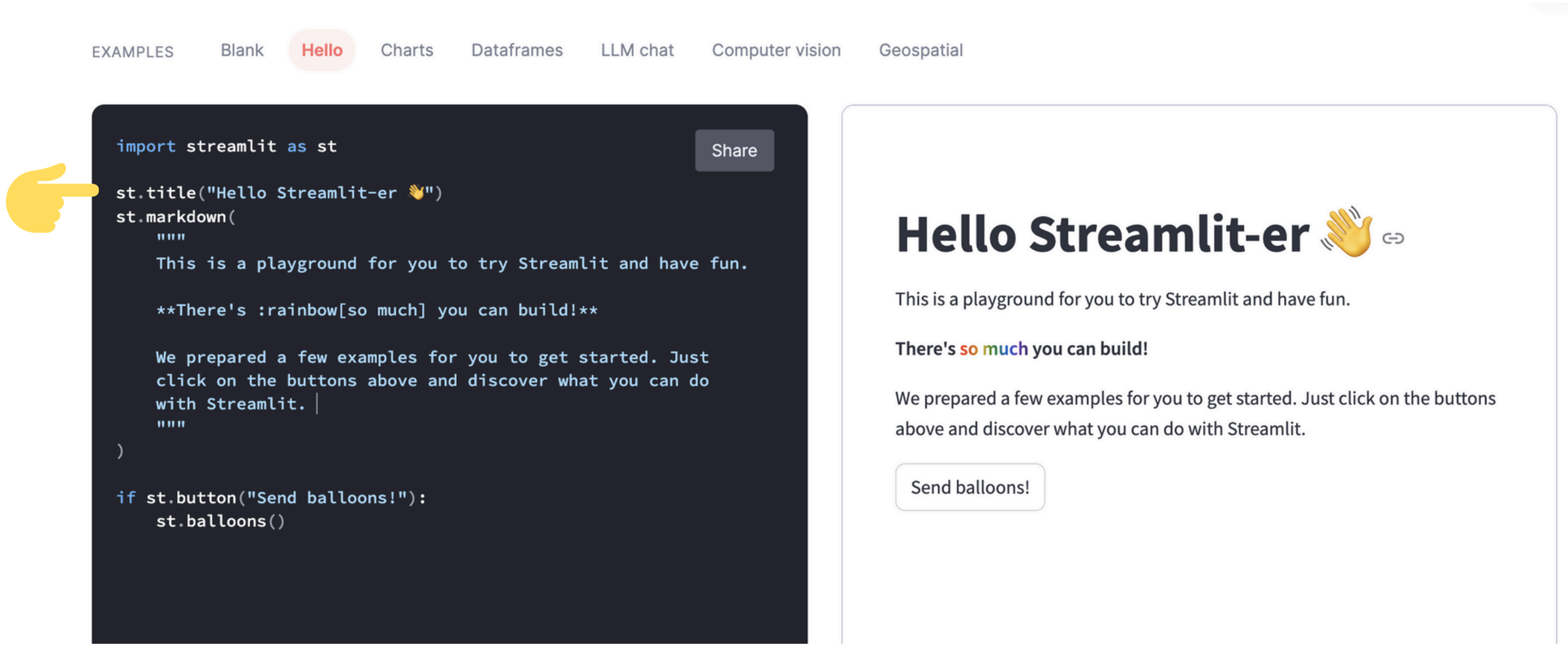
as st means you can now type st instead of streamlit, making your code shorter.

What happens if you don't write import streamlit as st?





📢 This is like trying to cook without plugging in the rice cooker—it won't work.



st.title("Hello Streamlit-er 👋")

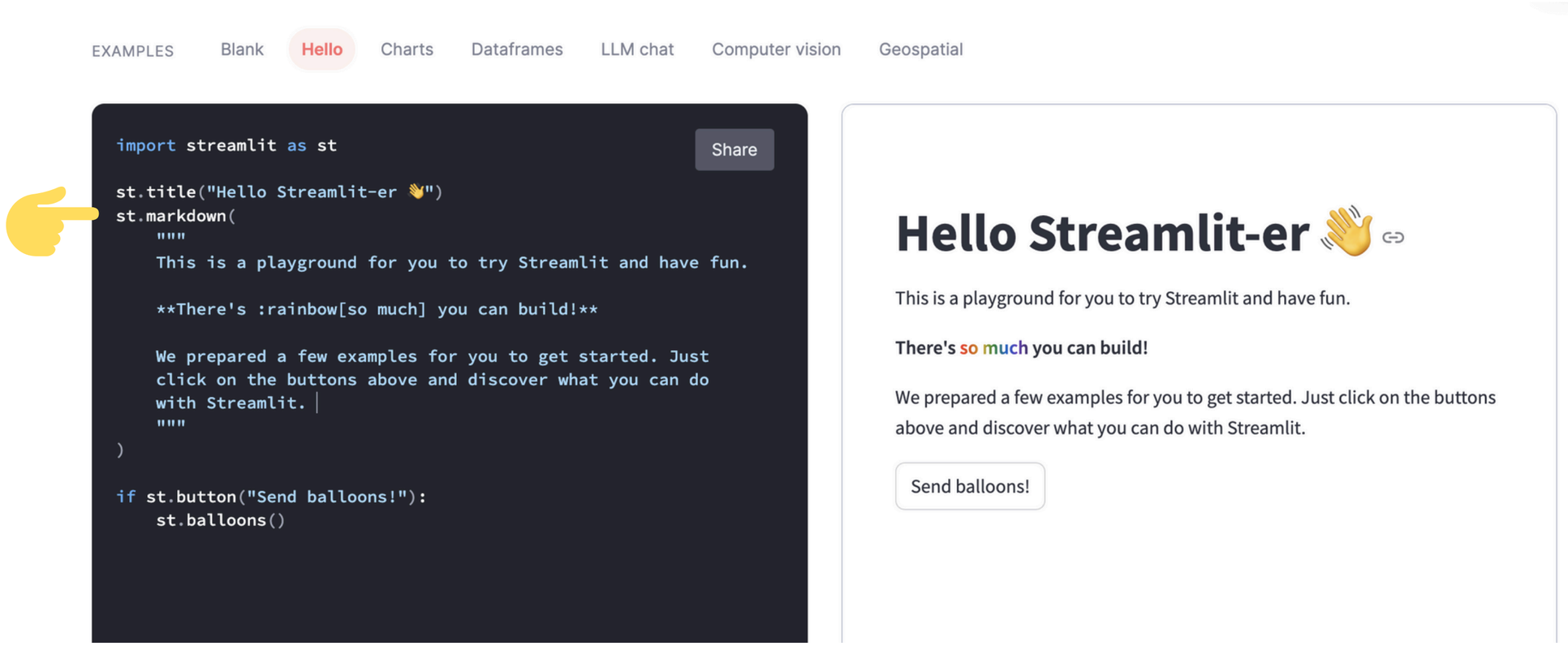
👉 This creates a big title at the top of the webpage.

✅ Analogy:

Think of a webpage like a book or article.

The title of a book (like "Harry Potter") is big and stands out.

st.title() makes a big, bold title at the top of your Streamlit app.



st.markdown(...)

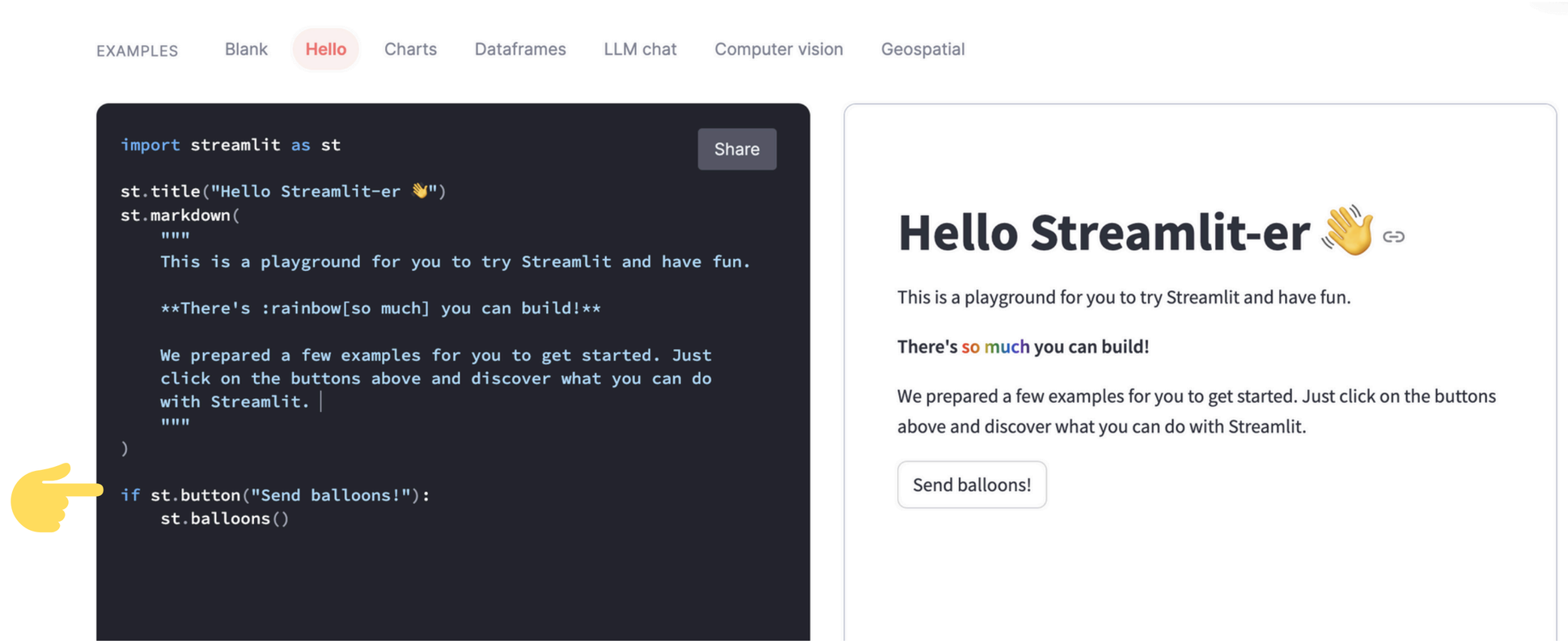
👉 This adds some text to the webpage, but with rich formatting (bold, colors, and more).

✅ Analogy:

Imagine you are writing a blog post.

Some words are bold, some are italic, and some are 🎨 colorful.

st.markdown() lets you write text with formatting, just like in a blog or document.

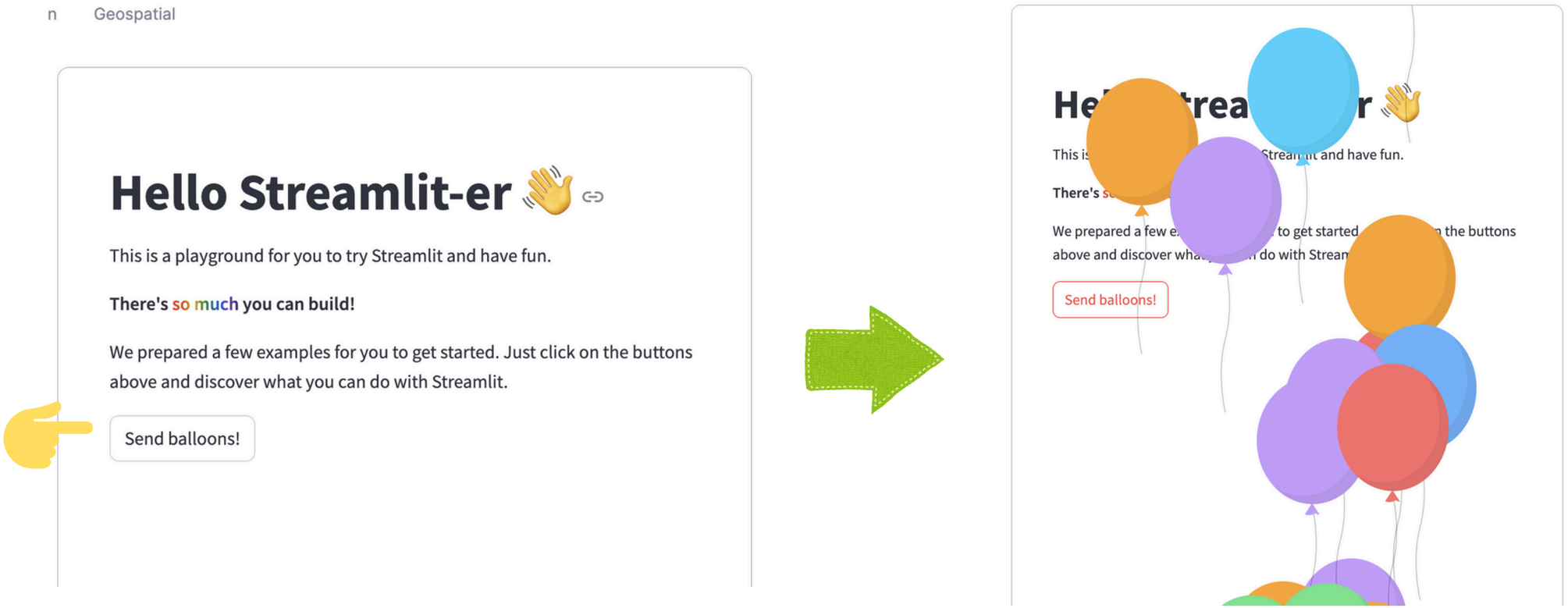


if st.button("Send balloons!"):

👉 This creates a button that a user can click.

✅ Analogy:

Imagine an elevator button. When you press it, something happens (the elevator moves).
This button works the same way: when clicked, it does something special.



st.balloons()

👉 This makes animated balloons fly across the screen! 🎈🎈🎈

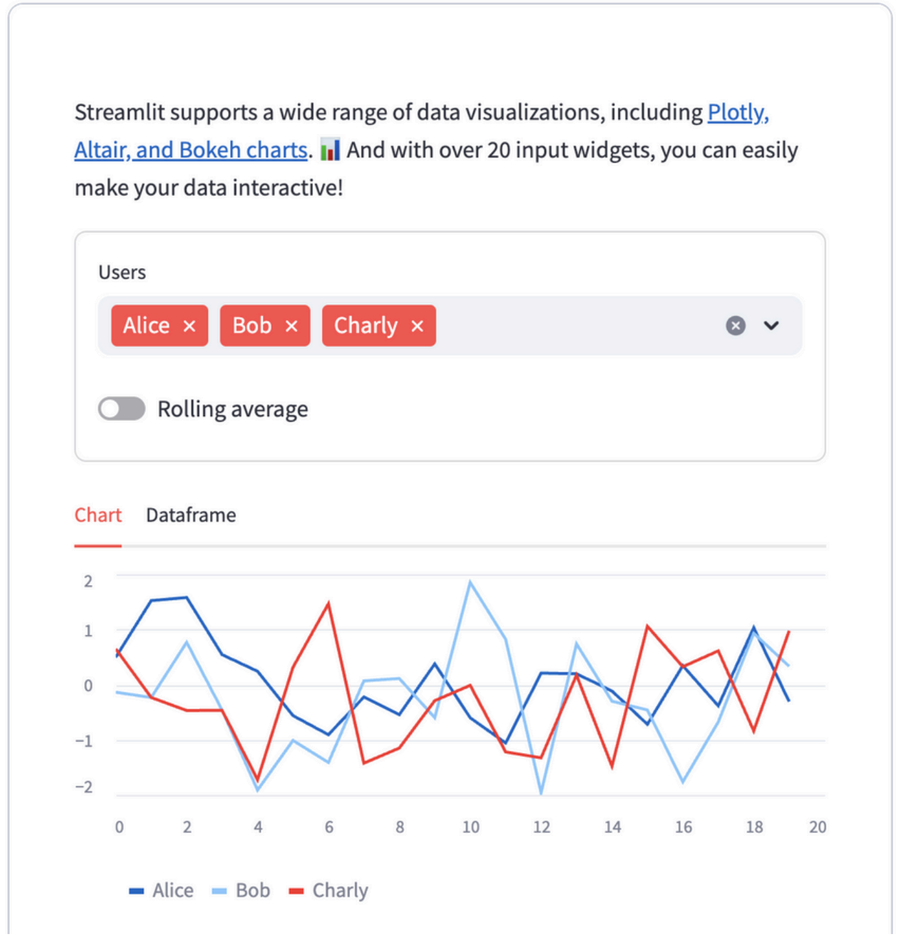
✅ Analogy:

Imagine clicking a button at a birthday party, and suddenly, balloons fill the room!

This command adds a fun animation when the button is pressed.

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

st.write("Streamlit supports a wide range of data visualizations, i

all_users = ["Alice", "Bob", "Charly"]
with st.container(border=True):
    users = st.multiselect("Users", all_users, default=all_users)
    rolling_average = st.toggle("Rolling average")

np.random.seed(42)
data = pd.DataFrame(np.random.randn(20, len(users)), columns=users)
if rolling_average:
    data = data.rolling(7).mean().dropna()

tab1, tab2 = st.tabs(["Chart", "Dataframe"])
tab1.line_chart(data, height=250)
tab2.dataframe(data, height=250, use_container_width=True)
```

Share



all_users = ["Alice", "Bob", "Charly"]

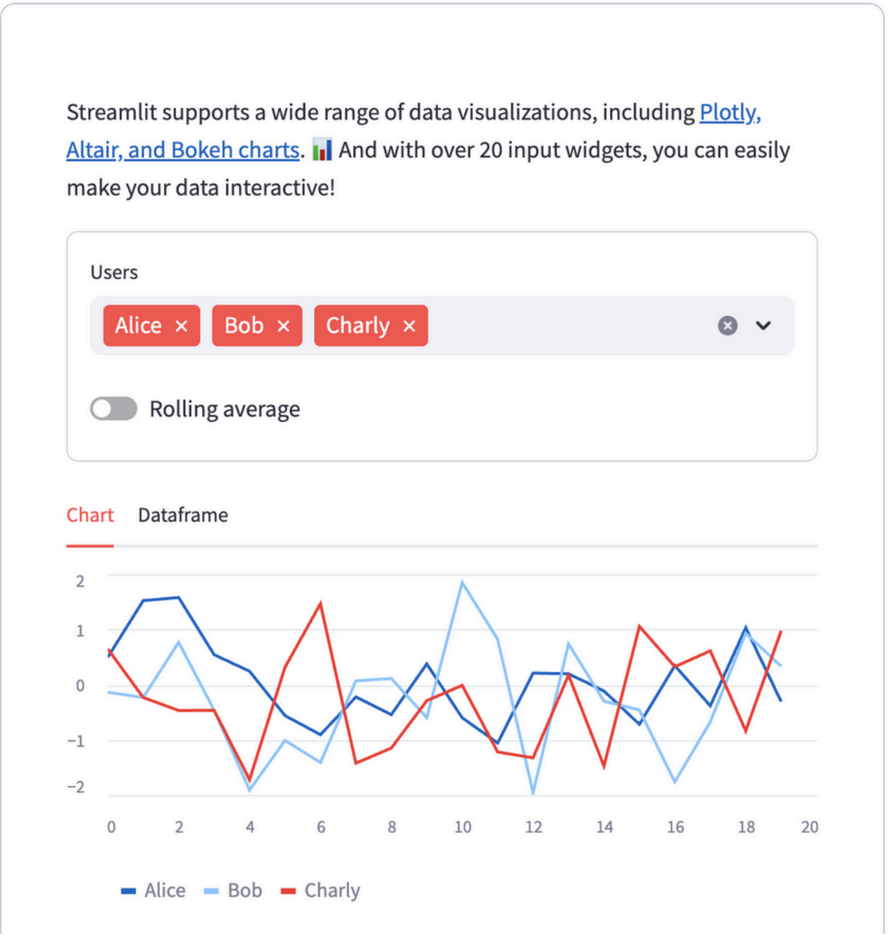
👉 Creates a list of users.

✅ Analogy:

Imagine you are listing your three best friends:

This list will be used later for selection.

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

st.write("Streamlit supports a wide range of data visualizations, i

all_users = ["Alice", "Bob", "Charly"]
with st.container(border=True):
    users = st.multiselect("Users", all_users, default=all_users)
    rolling_average = st.toggle("Rolling average")

np.random.seed(42)
data = pd.DataFrame(np.random.randn(20, len(users)), columns=users)
if rolling_average:
    data = data.rolling(7).mean().dropna()

tab1, tab2 = st.tabs(["Chart", "Dataframe"])
tab1.line_chart(data, height=250)
tab2.dataframe(data, height=250, use_container_width=True)
```



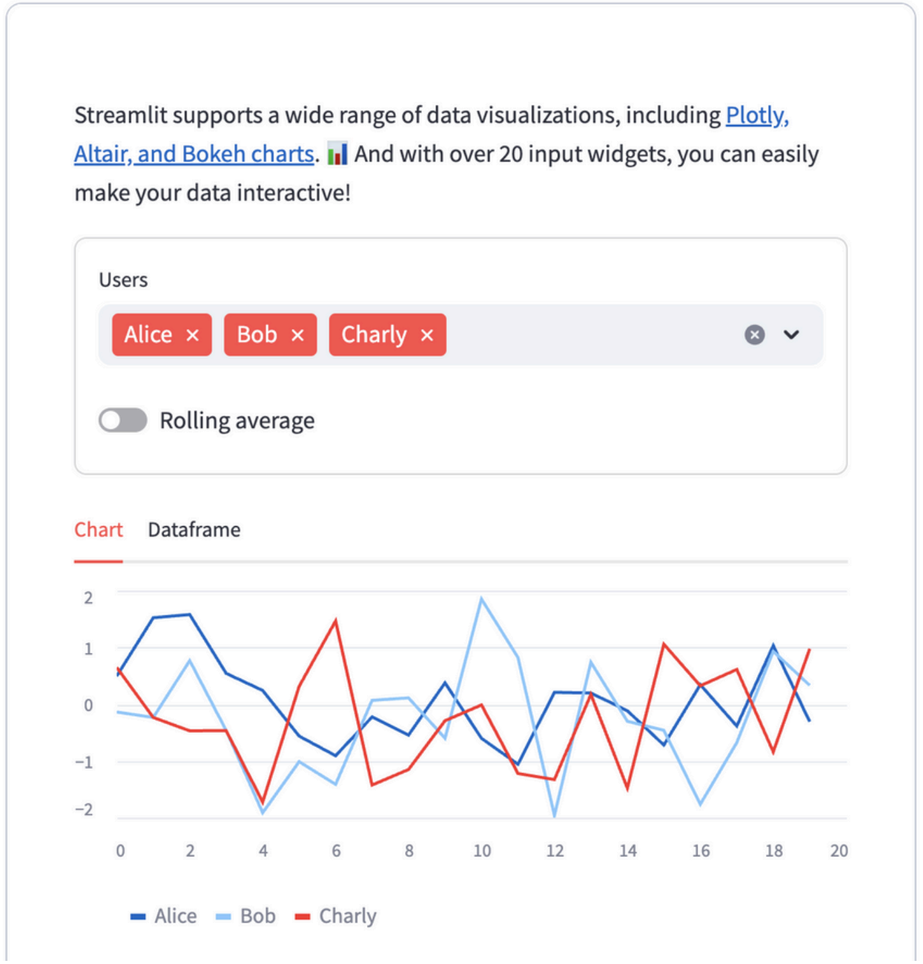
with st.container(border=True):

👉 Creates a section with a border to keep UI elements grouped.

✅ Analogy:

This is like a box on a webpage that holds related content.

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

st.write("Streamlit supports a wide range of data visualizations, i

all_users = ["Alice", "Bob", "Charly"]
with st.container(border=True):
    users = st.multiselect("Users", all_users, default=all_users)
    rolling_average = st.toggle("Rolling average")

np.random.seed(42)
data = pd.DataFrame(np.random.randn(20, len(users)), columns=users)
if rolling_average:
    data = data.rolling(7).mean().dropna()

tab1, tab2 = st.tabs(["Chart", "Dataframe"])
tab1.line_chart(data, height=250)
tab2.dataframe(data, height=250, use_container_width=True)
```



users = st.multiselect("Users", all_users, default=all_users)

👉 Creates a multi-selection dropdown menu.

🔹 What this does:

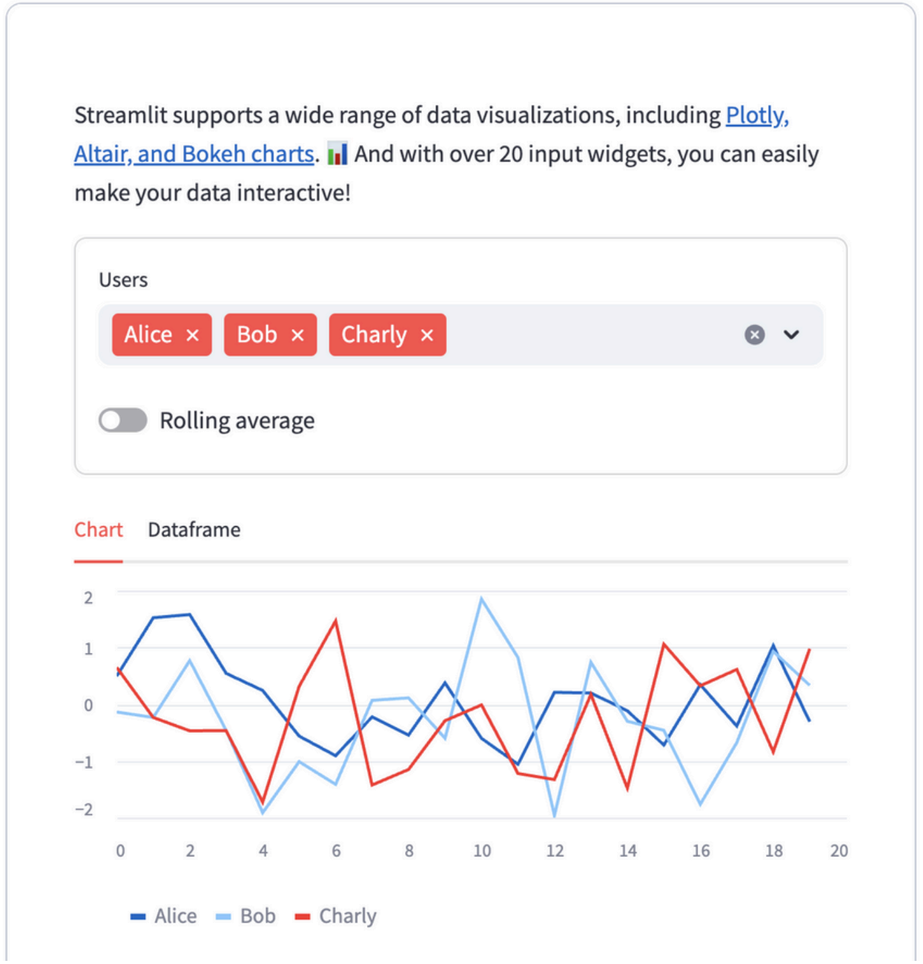
Shows a list: Alice, Bob, Charly.

Users can select one, two, or all names.

By default, all three names are selected.

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

st.write("Streamlit supports a wide range of data visualizations, i

all_users = ["Alice", "Bob", "Charly"]
with st.container(border=True):
    users = st.multiselect("Users", all_users, default=all_users)
    rolling_average = st.toggle("Rolling average")

np.random.seed(42)
data = pd.DataFrame(np.random.randn(20, len(users)), columns=users)
if rolling_average:
    data = data.rolling(7).mean().dropna()

tab1, tab2 = st.tabs(["Chart", "Dataframe"])
tab1.line_chart(data, height=250)
tab2.dataframe(data, height=250, use_container_width=True)
```

Share



`rolling_average = st.toggle("Rolling average")`

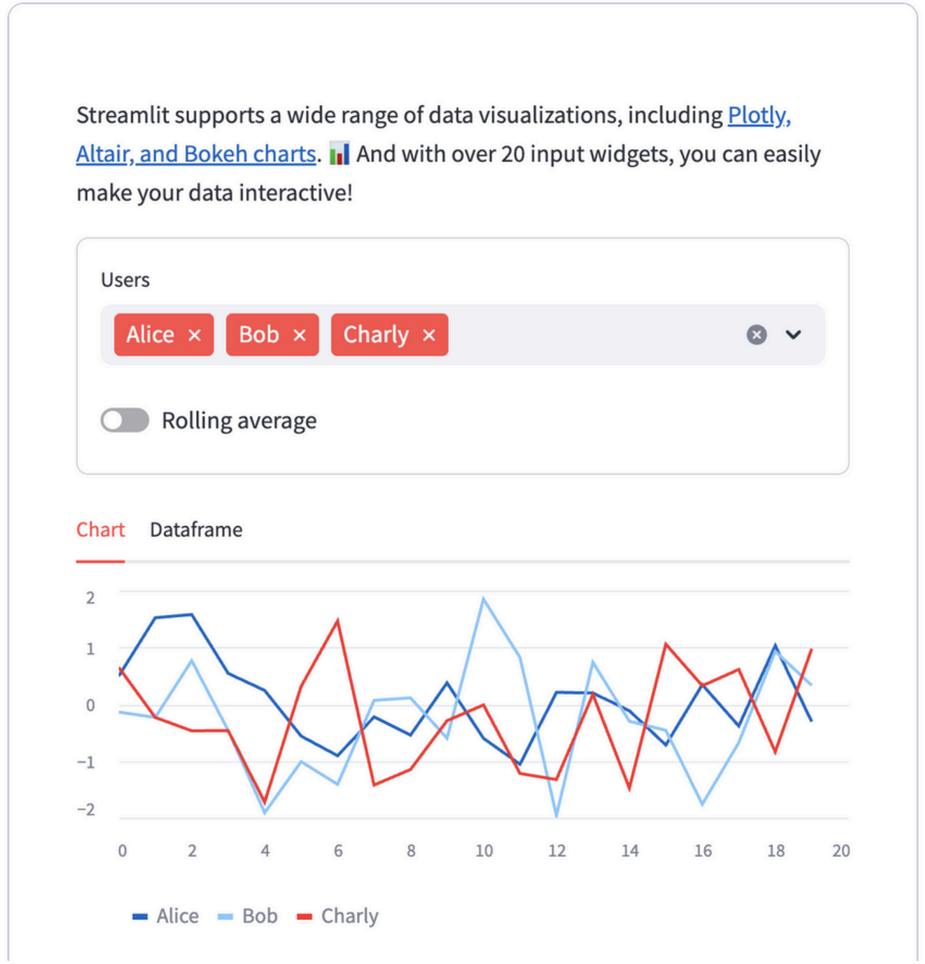
👉 Creates an ON/OFF switch.

✅ Analogy:

Like a light switch 🏠 → Turn it ON or OFF.

Here, it controls whether we apply a rolling average to the data.

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

st.write("Streamlit supports a wide range of data visualizations, i

all_users = ["Alice", "Bob", "Charly"]
with st.container(border=True):
    users = st.multiselect("Users", all_users, default=all_users)
    rolling_average = st.toggle("Rolling average")

np.random.seed(42)
data = pd.DataFrame(np.random.randn(20, len(users)), columns=users)
if rolling_average:
    data = data.rolling(7).mean().dropna()

tab1, tab2 = st.tabs(["Chart", "Dataframe"])
tab1.line_chart(data, height=250)
tab2.dataframe(data, height=250, use_container_width=True)
```



np.random.seed(42)

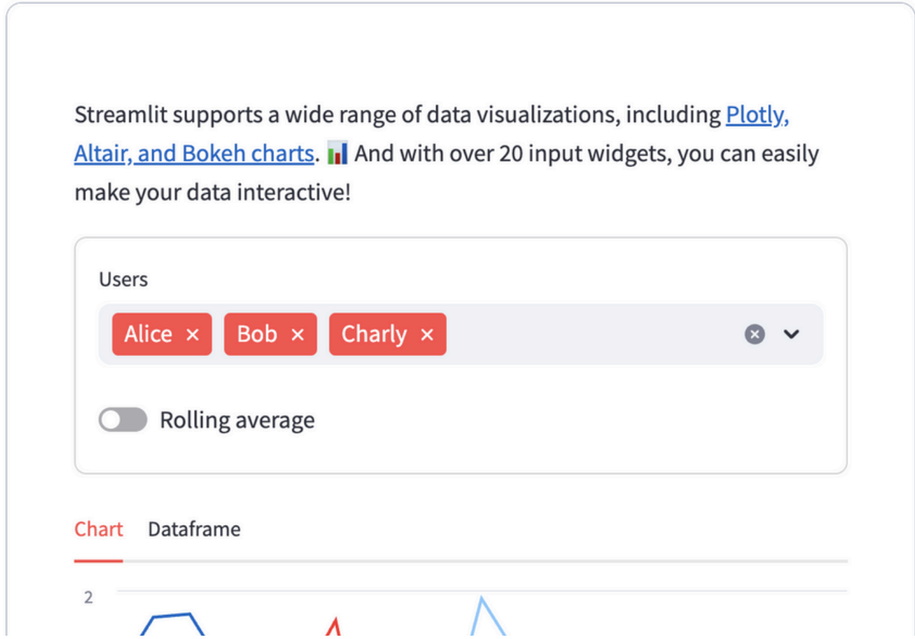
👉 Makes sure the random numbers are always the same.

✅ Analogy:

Like fixing the shuffle order of a playlist 🎵 so every time you play it, the order stays the same.

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

st.write("Streamlit supports a wide range of data visualizations, i

all_users = ["Alice", "Bob", "Charly"]
with st.container(border=True):
    users = st.multiselect("Users", all_users, default=all_users)
    rolling_average = st.toggle("Rolling average")

np.random.seed(42)
data = pd.DataFrame(np.random.randn(20, len(users)), columns=users)
if rolling_average:
    data = data.rolling(7).mean().dropna()

tab1, tab2 = st.tabs(["Chart", "Dataframe"])
tab1.line_chart(data, height=250)
tab2.dataframe(data, height=250, use_container_width=True)
```

Share



data = pd.DataFrame(np.random.randn(20, len(users)), columns=users)

👉 Creates a table with random numbers.

✅ Breaking it down:

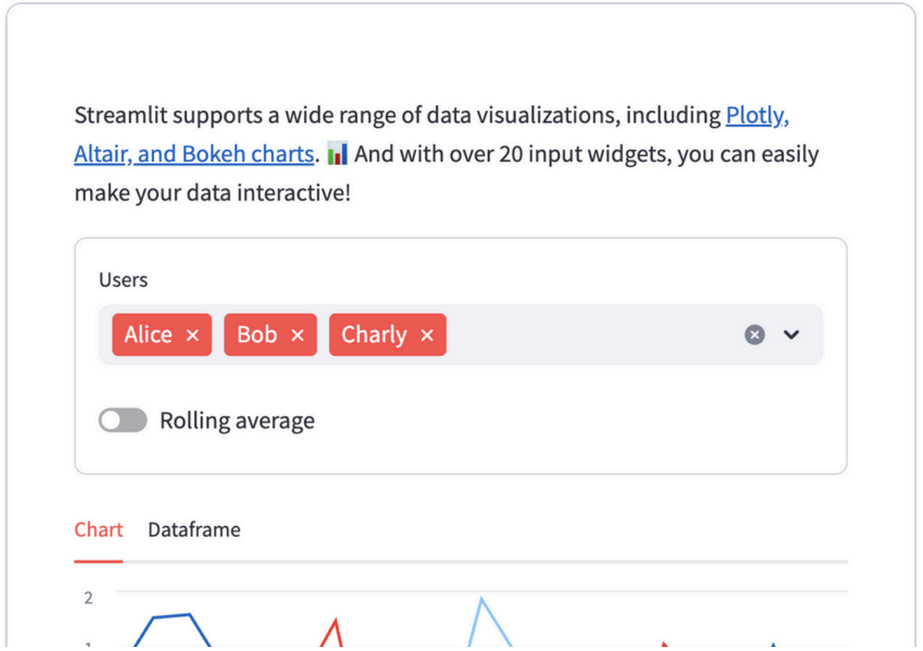
np.random.randn(20, len(users)) → Generates 20 rows of random numbers.

columns=users → Uses the selected users as column names.

🛌 Example Output (random numbers):

	Alice	Bob	Charly
0	0.49	-0.13	1.58
1	-0.86	0.76	0.12
2	0.46	0.34	-1.44
...	...	...	...

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

st.write("Streamlit supports a wide range of data visualizations, i

all_users = ["Alice", "Bob", "Charly"]
with st.container(border=True):
    users = st.multiselect("Users", all_users, default=all_users)
    rolling_average = st.toggle("Rolling average")

np.random.seed(42)
data = pd.DataFrame(np.random.randn(20, len(users)), columns=users)
if rolling_average:
    data = data.rolling(7).mean().dropna()

tab1, tab2 = st.tabs(["Chart", "Dataframe"])
tab1.line_chart(data, height=250)
tab2.dataframe(data, height=250, use_container_width=True)
```

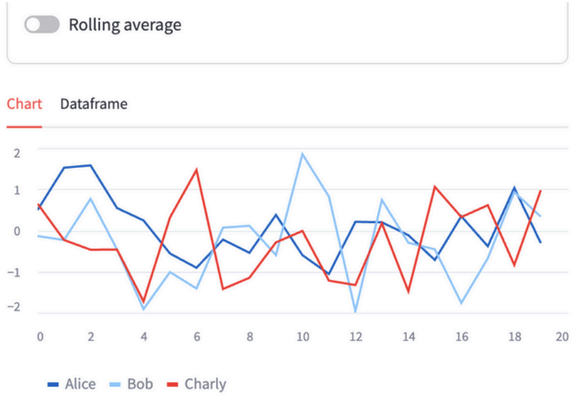


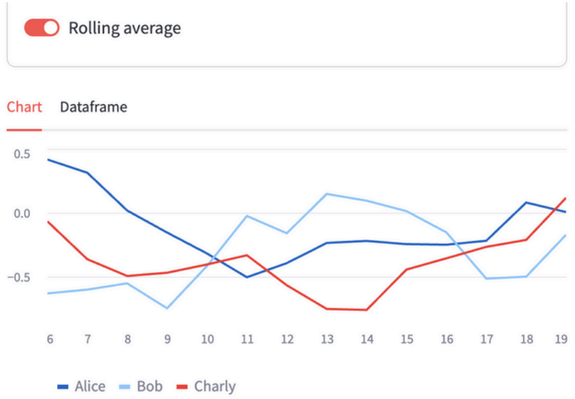
if rolling_average:

👉 If the user turned ON the rolling average, apply it.

✅ What this does:

data.rolling(7).mean() → Averages the last 7 rows.
.dropna() → Removes empty values.





📢 Effect:

If the switch is ON, the data gets smoothed.
If the switch is OFF, the data stays random.

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

st.write("Streamlit supports a wide range of data visualizations, i

all_users = ["Alice", "Bob", "Charly"]
with st.container(border=True):
    users = st.multiselect("Users", all_users, default=all_users)
    rolling_average = st.toggle("Rolling average")

np.random.seed(42)
data = pd.DataFrame(np.random.randn(20, len(users)), columns=users)
if rolling_average:
    data = data.rolling(7).mean().dropna()

tab1, tab2 = st.tabs(["Chart", "Dataframe"])
tab1.line_chart(data, height=250)
tab2.dataframe(data, height=250, use_container_width=True)
```

Share

Streamlit supports a wide range of data visualizations, including Plotly, Altair, and Bokeh charts. 📊 And with over 20 input widgets, you can easily make your data interactive!

Users

Alice × Bob × Charly ×

Rolling average

Chart Dataframe

tab1, tab2 = st.tabs(["Chart", "Dataframe"])

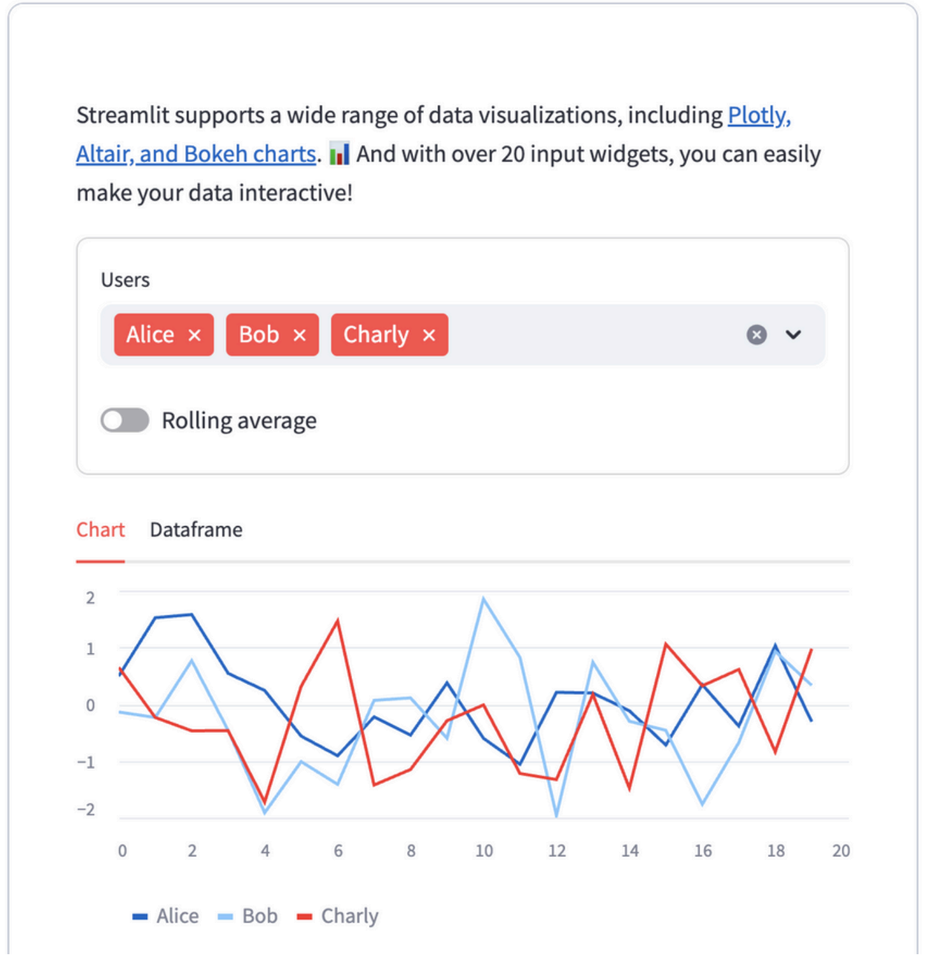
👉 Creates two tabs:

1. "Chart" → Shows a graph.
2. "Dataframe" → Shows the raw table.

Chart Dataframe

	Alice	Bob	Charly
0	0.4967	-0.1383	0.647
1	1.523	-0.2342	-0.234
2	1.5792	0.7674	-0.469
3	0.5426	-0.4634	-0.465
4	0.242	-1.9133	-1.724
5	-0.5623	-1.0128	0.314

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

st.write("Streamlit supports a wide range of data visualizations, i

all_users = ["Alice", "Bob", "Charly"]
with st.container(border=True):
    users = st.multiselect("Users", all_users, default=all_users)
    rolling_average = st.toggle("Rolling average")

np.random.seed(42)
data = pd.DataFrame(np.random.randn(20, len(users)), columns=users)
if rolling_average:
    data = data.rolling(7).mean().dropna()

tab1, tab2 = st.tabs(["Chart", "Dataframe"])
tab1.line_chart(data, height=250)
tab2.dataframe(data, height=250, use_container_width=True)
```

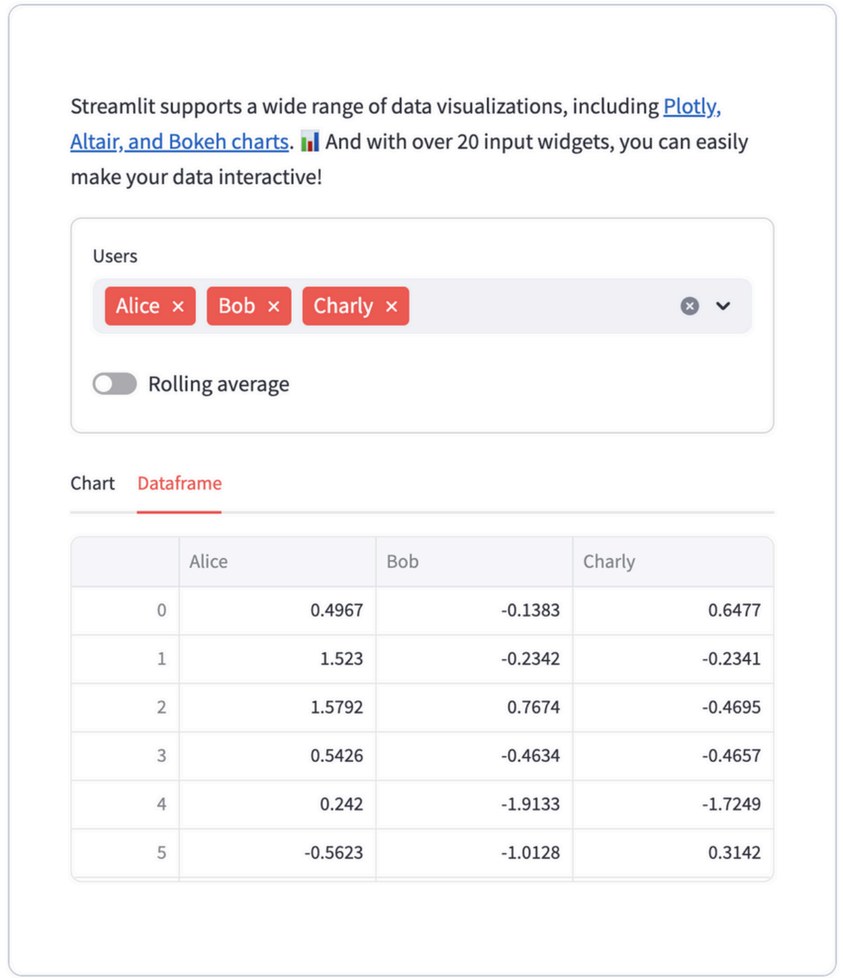


tabl.line_chart(data, height=250)

👉 Displays a line chart of the data.

✅ Analogy:

Like an Excel chart 📈, but in Python.



tab2.dataframe(data, height=250, use_container_width=True)

👉 Displays the data table.

✅ Analogy:

Like an Excel spreadsheet but interactive.

Home / Get started / **Installation**

Install Streamlit

There are multiple ways to set up your development environment and install Streamlit. Read below to understand these options. Developing locally with Python installed on your own computer is the most common scenario.

Summary for experts

1. Set up your Python development environment.
2. Run:

```
pip install streamlit
```

3. Validate the installation by running our Hello app:

```
streamlit hello
```

4. Jump to our Basic concepts.

Step 1: Open Terminal

Windows:
Press Win + S, search for "Anaconda Prompt" or "Command Prompt" (cmd).
Click Open.

Mac:
Press Cmd + Space, search for "Terminal".
Click Open.

Step 2: Create a New Project Folder

Next, we will create a folder to store our Streamlit project.

📌 Run these commands in the terminal:

```
cd ~  # Go to your home directory
mkdir nanomovie  # Create a new folder
cd nanomovie  # Enter the folder
```

✅ Now, you are inside your project folder!

Step 3:
Create & Activate a Conda Virtual Environment

It is recommended to use a Conda virtual environment to manage dependencies.

```
03:11:04 with freax in ~/nanomovie via ⬢ v20.18.1 via Ⓒ base
[→ conda create -n nanomovie python=3.10 -y
```

conda create -n nanomovie → Creates a new environment called "nanomovie".
python=3.10 → Installs Python 3.10 in this environment.
-y → Automatically confirms the installation.

```
03:13:13 with freax in ~/nanomovie via ⬢ v20.18.1 via Ⓒ base took 40.5s
[→ conda activate nanomovie

03:13:17 with freax in ~/nanomovie via ⬢ v20.18.1 via Ⓒ nanomovie …
→
```

Activate the environment

Step 4: Install Streamlit

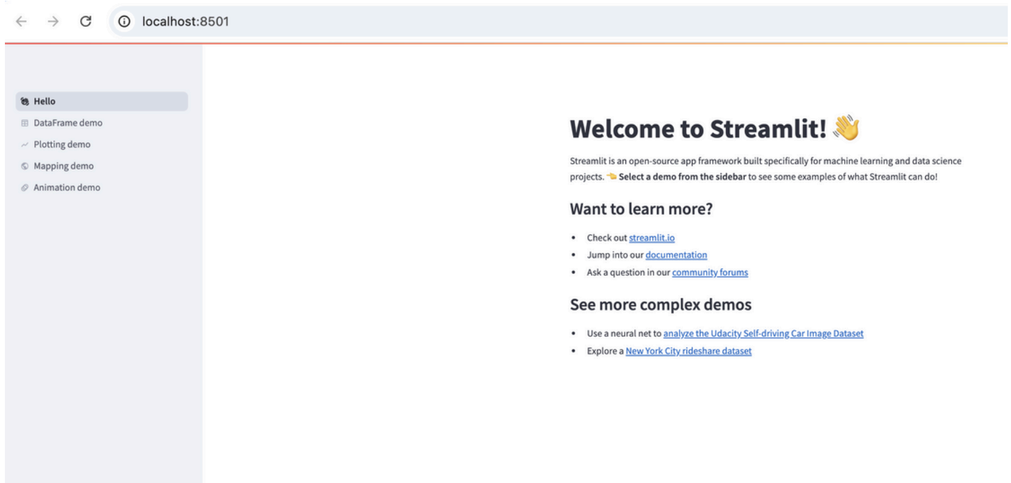
Now, let's install Streamlit inside this environment.

```
03:13:17 with freax in ~/nanomovie via ⬢ v20.18.1 via Ⓒ nanomovie
➜ pip install streamlit
```

Step 5:

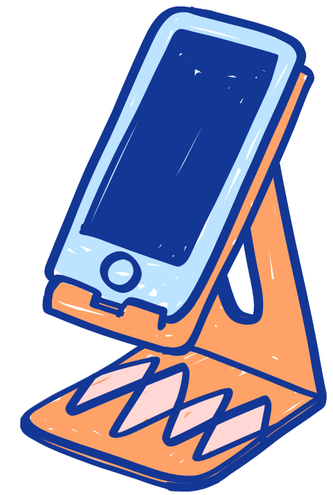
Validate the installation by running Hello app

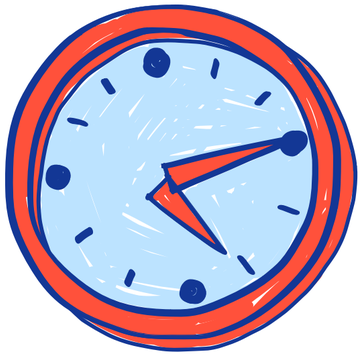
```
03:14:34 with freax in ~/nanomovie via ⬢ v20.18.1 via Ⓒ nanomovie took 22.7s
➜ streamlit hello
```



Success!

THANK YOU





See you next time!