snowflake®

Streamlit Overview

Diana Shaw & Jeff Carpenter
September 2022

Enormous opportunity to help put data to work

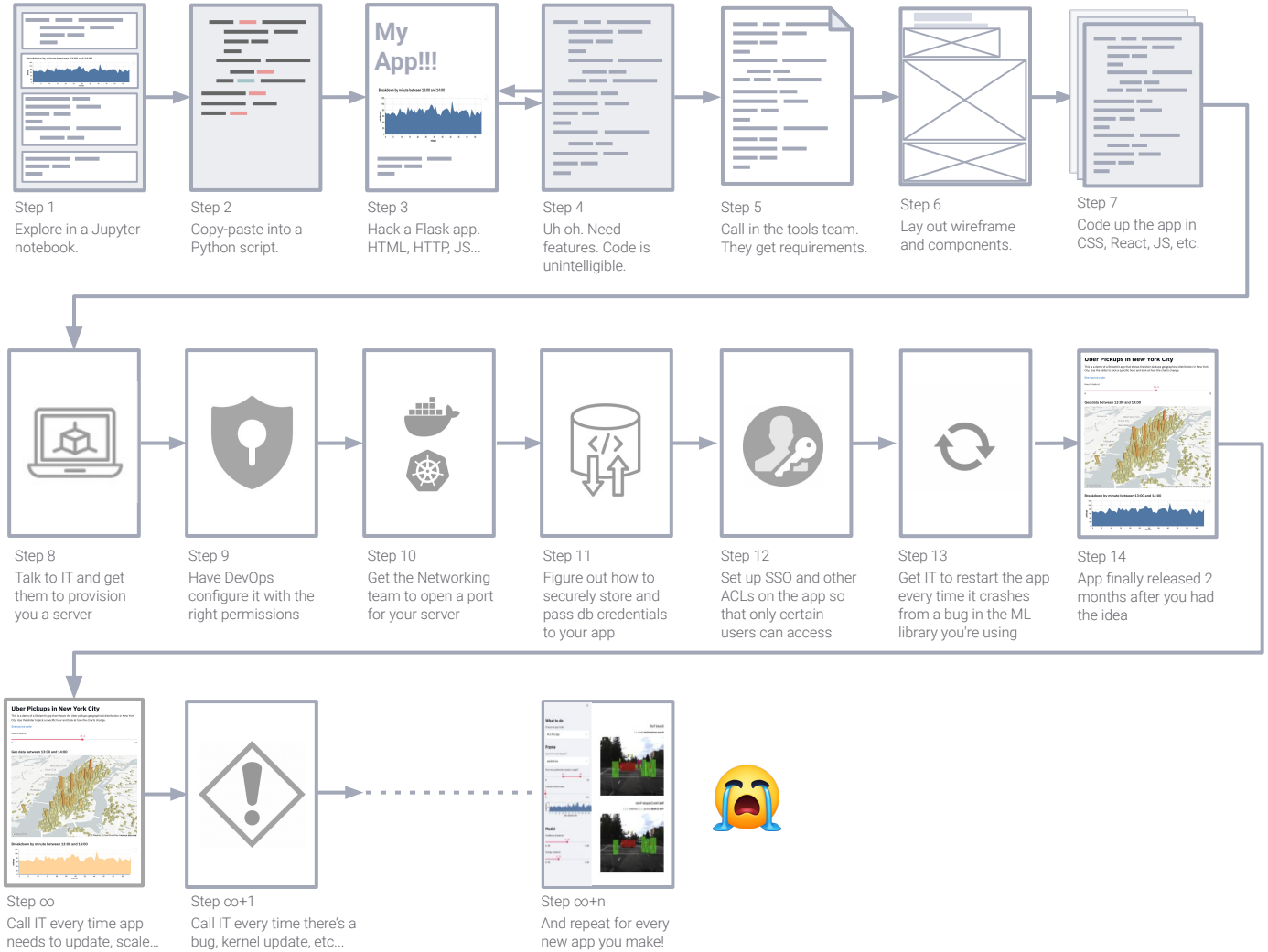
73%

of data is rendered effectively unused because it's just too hard to build the tools to use it

Source

Getting from data to action is hard

Only a fraction of data questions are ever answered since companies haven't built tools to capture or analyze the data. 73% of data is rendered effectively unused* because it's just too hard to build the tools to use it.



Step 1
Explore in a Jupyter notebook.

Step 2
Copy-paste into a Python script.

Step 3
Hack a Flask app. HTML, HTTP, JS...

Step 4
Uh oh. Need features. Code is unintelligible.

Step 5
Call in the tools team. They get requirements.

Step 6
Lay out wireframe and components.

Step 7
Code up the app in CSS, React, JS, etc.

Step 8
Talk to IT and get them to provision you a server

Step 9
Have DevOps configure it with the right permissions

Step 10
Get the Networking team to open a port for your server

Step 11
Figure out how to securely store and pass db credentials to your app

Step 12
Set up SSO and other ACLs on the app so that only certain users can access

Step 13
Get IT to restart the app every time it crashes from a bug in the ML library you're using

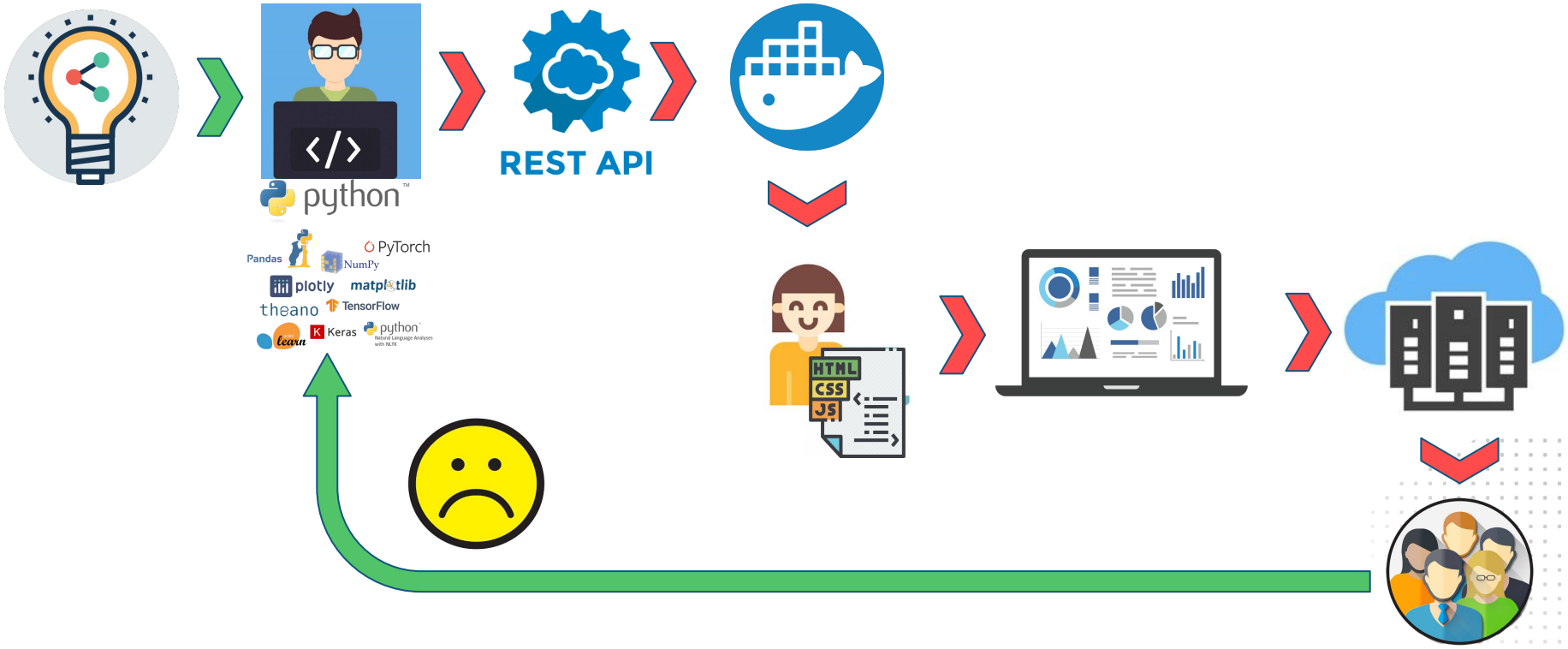
Step 14
App finally released 2 months after you had the idea

Step ∞
Call IT every time app needs to update, scale...

Step ∞+1
Call IT every time there's a bug, kernel update, etc...

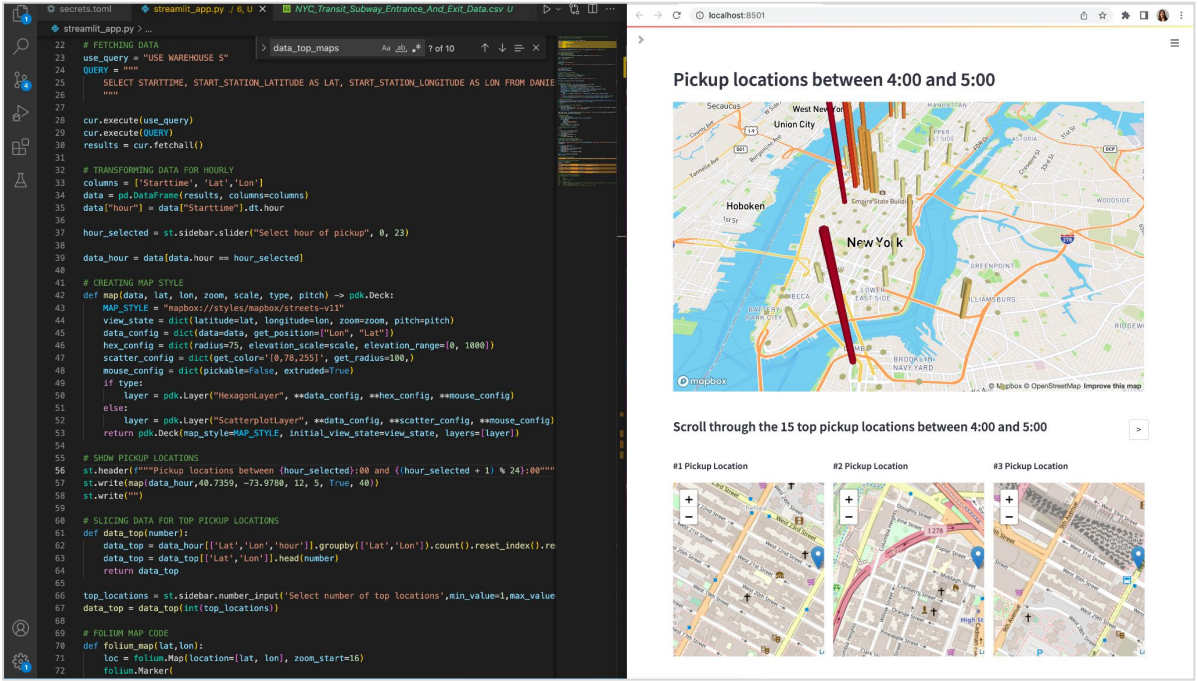
Step ∞+n
And repeat for every new app you make!

The before times…



Source

WHAT IF BUILDING TOOLS WERE AS EASY AS WRITING PYTHON SCRIPTS?



Using Streamlit

Source

Iterate quickly with Streamlit's fast prototyping flow

Get coding in minutes

Share apps with your team and iterate on them together

Publish apps to your company, get feedback, and quickly make updates to expand and improve them

Streamlit is built by and for data practitioners

Doing Data Science

Coding in Python

If they are in the Snowpark preview for Python that's a good signal! If they are not using Python already, then they are probably better off (at the moment) using their existing workflows.

Looking for speed!

There are many options for making data apps. Snowflake supports and will continue to support lots of options. You don't have to remake apps in Streamlit, but it's a good place to start if you're in Python and want something fast.

Common use cases for Streamlit

Rapid prototyping

Data science teams quickly build data apps in Python to test hypotheses, gather feedback, and iterate quickly to deliver insights to their business stakeholders.

Demo-ing work

Creating tools for business users

Data teams struggle to share the outputs of their models in an easily consumable way for their stakeholders. Folks from sales, operations, marketing, and support can often benefit from interacting with models the data team has built. Streamlit creates a fast and easy way to create new tools from data that empower business users.

Streamlit is an open-source Python library built on 3 core principles

(1)

Embrace Python scripting

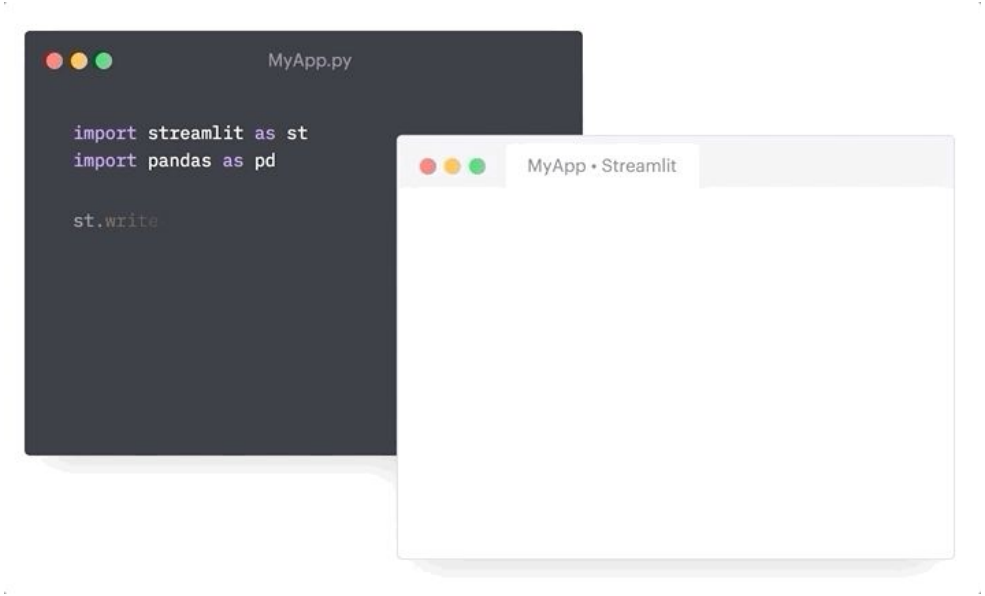
(2)

Treat widgets as variables

(3)

Reuse data and computation

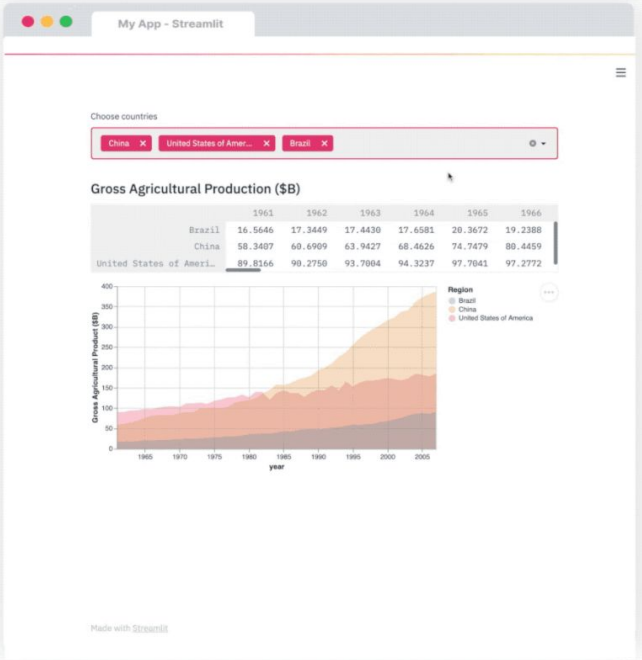
1. Embrace Python scripting



Streamlit lets you go straight from Python script to web app

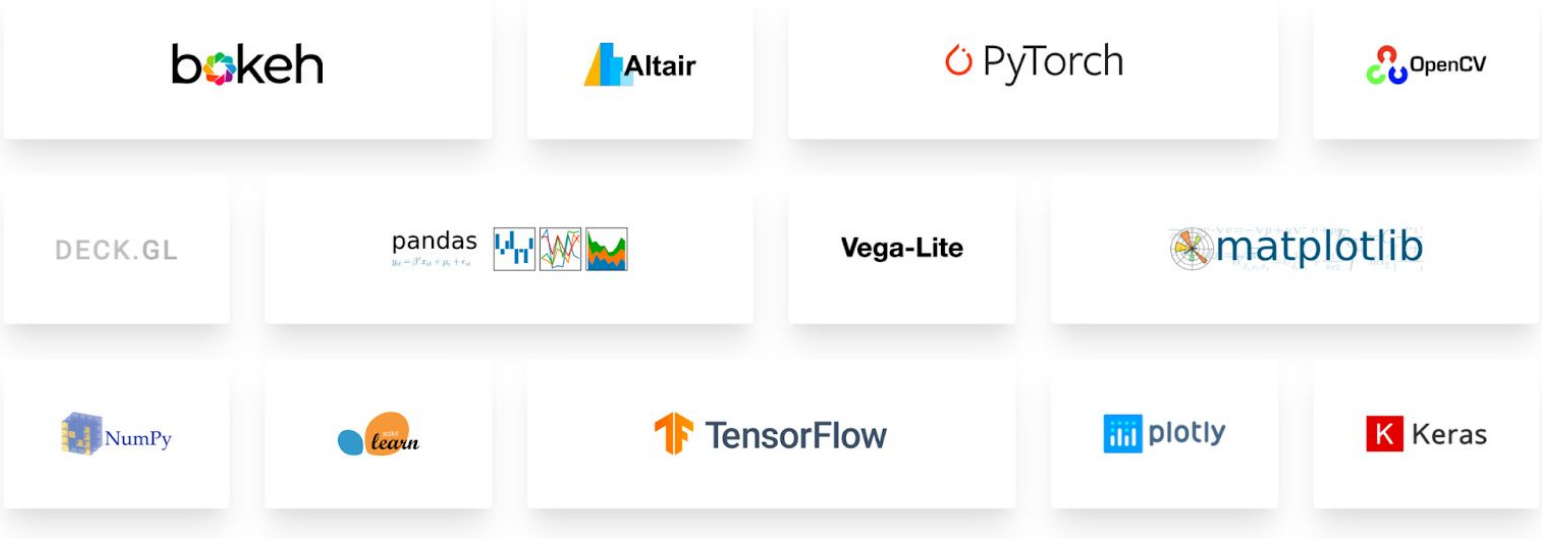
```python
import streamlit as st
import pandas as pd
import altair as alt

@st.cache
def get_UN_data():
    AWS_BUCKET_URL = "https://streamlit-demo-data.s3-us-west-2.amazonaws.com"
    df = pd.read_csv(AWS_BUCKET_URL + "/agri.csv.gz")
    return df.set_index("Region")

df = get_UN_data()

countries = st.multiselect(
    "Choose countries", list(df.index), ["China", "United States of America"]
)

data = df.loc[countries]
data /= 1000000.0
st.write("### Gross Agricultural Production ($B)", data.sort_index())

data = data.T.reset_index()
data = pd.melt(data, id_vars=["index"]).rename(
    columns={"index": "year", "value": "Gross Agricultural Product ($B)"}
)
chart = (
    alt.Chart(data)
    .mark_area(opacity=0.3)
    .encode(
        x="year:T",
        y=alt.Y("Gross Agricultural Product ($B):Q", stack=None),
        color="Region:N",
    )
)
st.altair_chart(chart, use_container_width=True
```

~/Documents/python_files/gap-example.py 34:48 LF UTF-8 Python GitHub Git (0)



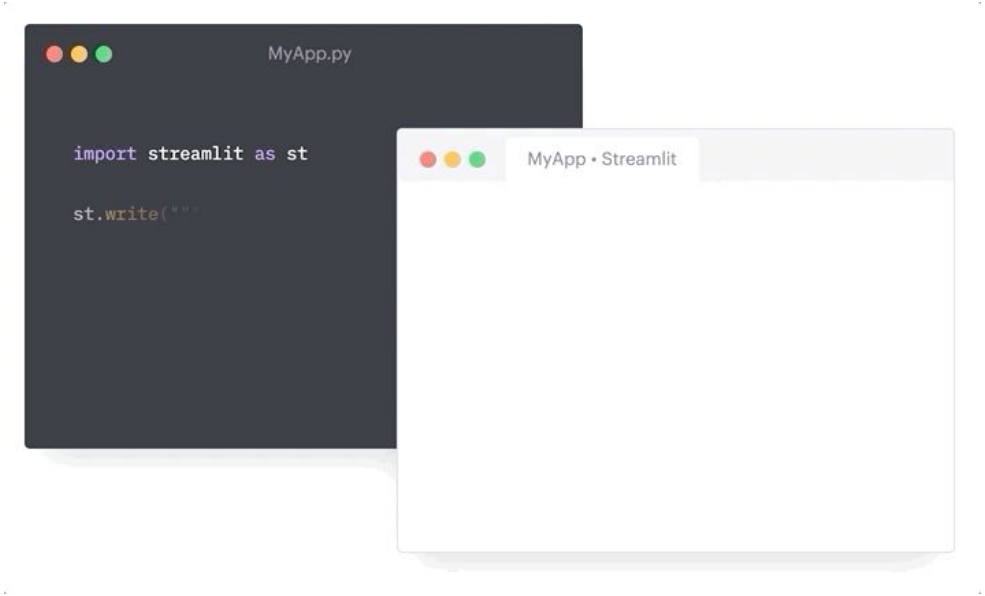
From code to app in minutes!

And it works with all your favorite Python libraries

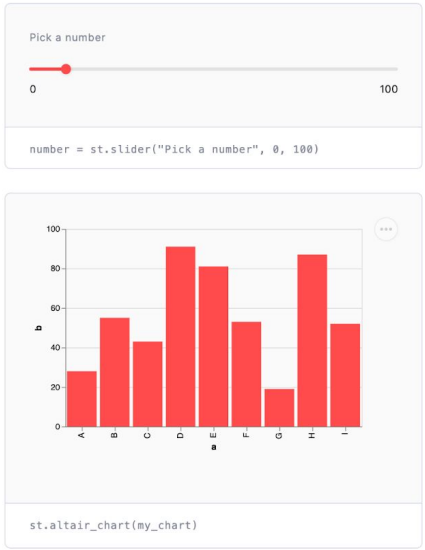


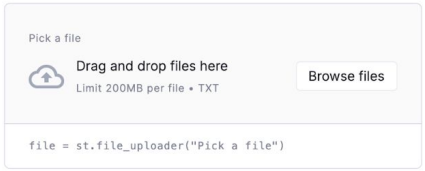
And many more!

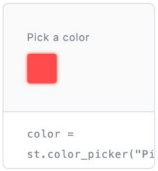
2. Treat widgets as variables

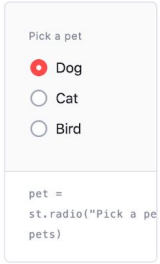


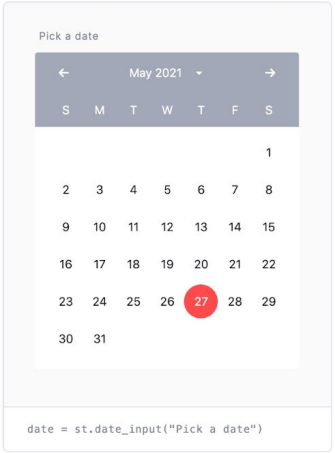
It’s easy to add in interaction, layout, and themes











Choose from light or dark theme or set your own custom colors

3. Reuse data and computation

Instantiate DB/model connectors only once

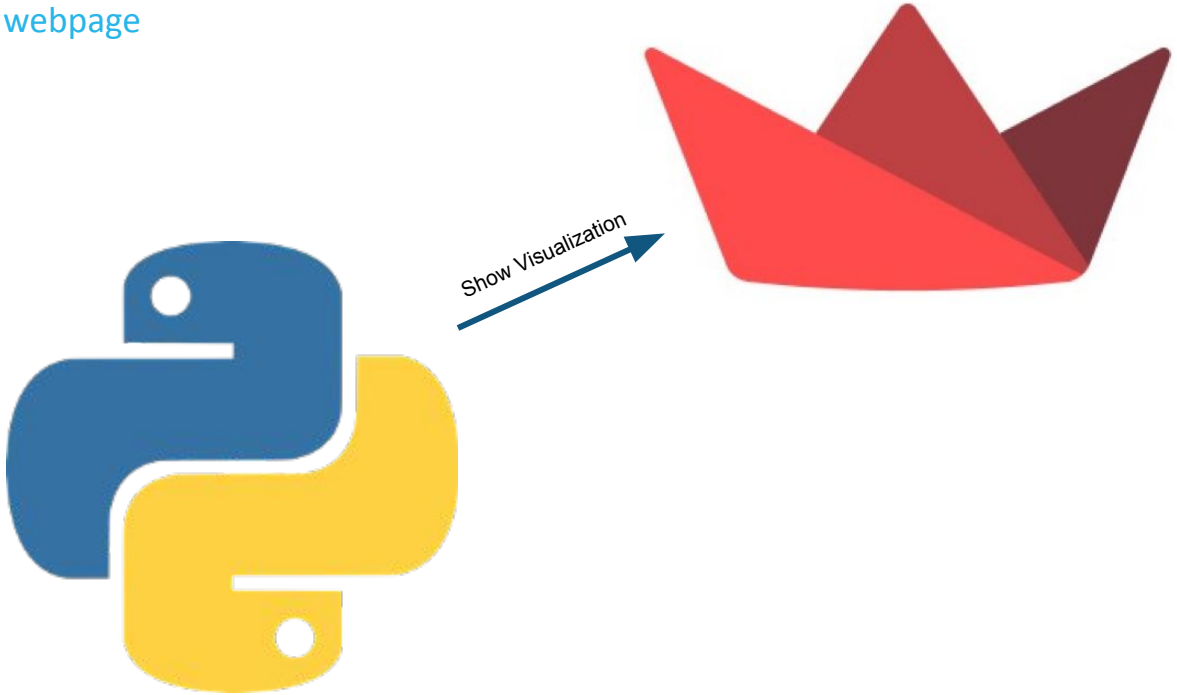
```python
@st.experimental_singleton
def sfconn(creds):
    return snowflake.connector.connect(**creds)

@st.experimental_memo
def query_database(sql, _connector):
  return pd.read_sql(sql, _connector)

conn = sfconn(**st.secrets['sfdevrel'])
df = query_database("SELECT * FROM users", conn)
```

Store query results in cache

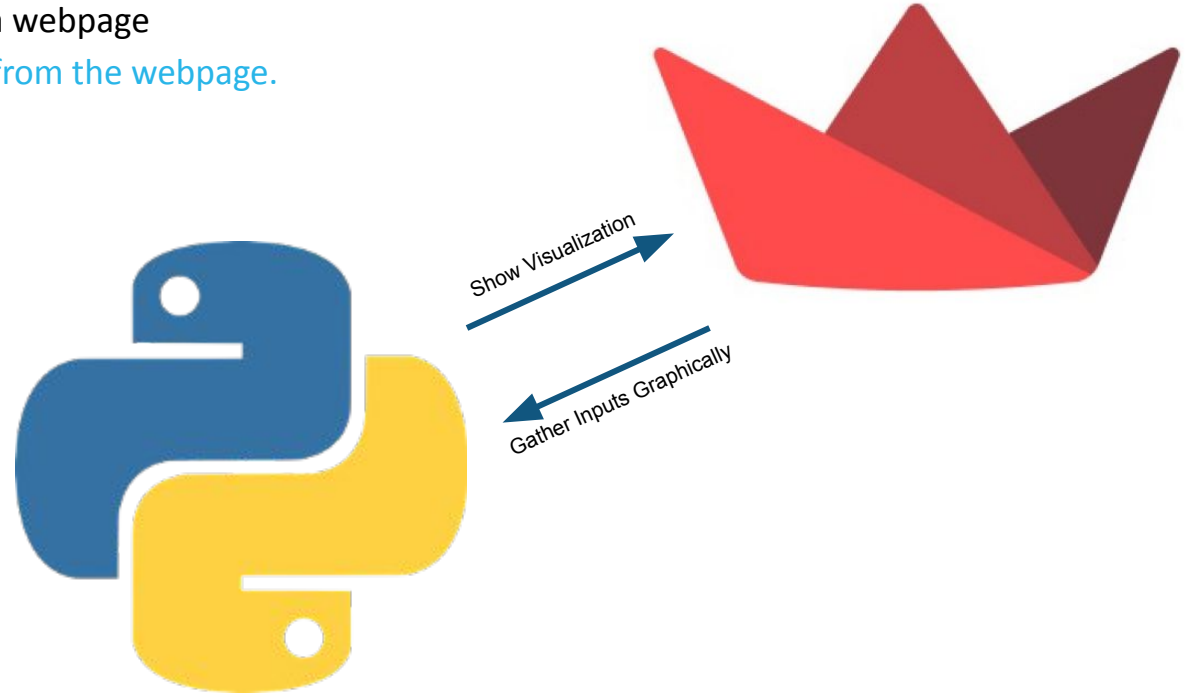
A Streamlit app is just a Python program

A Streamlit app is just a Python program

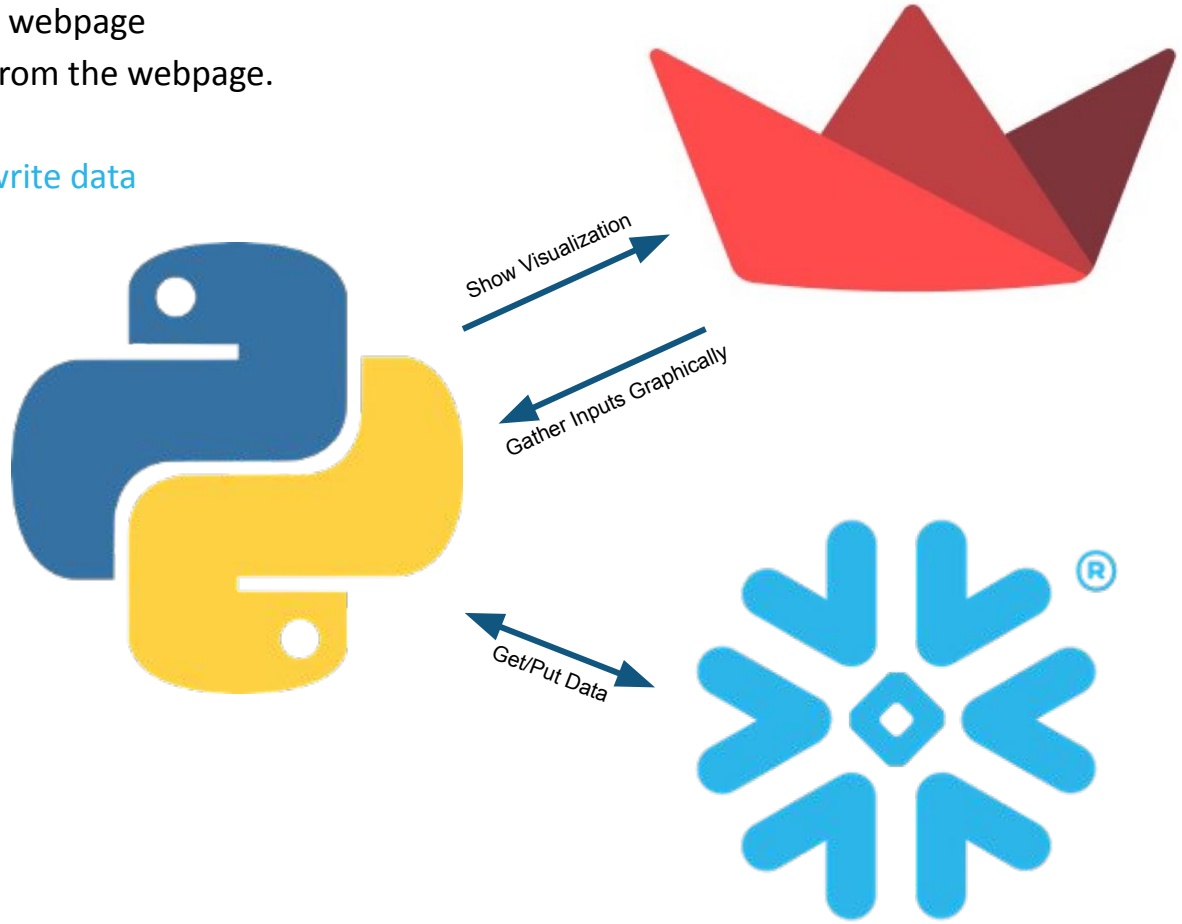
that uses the Streamlit library to display in a webpage



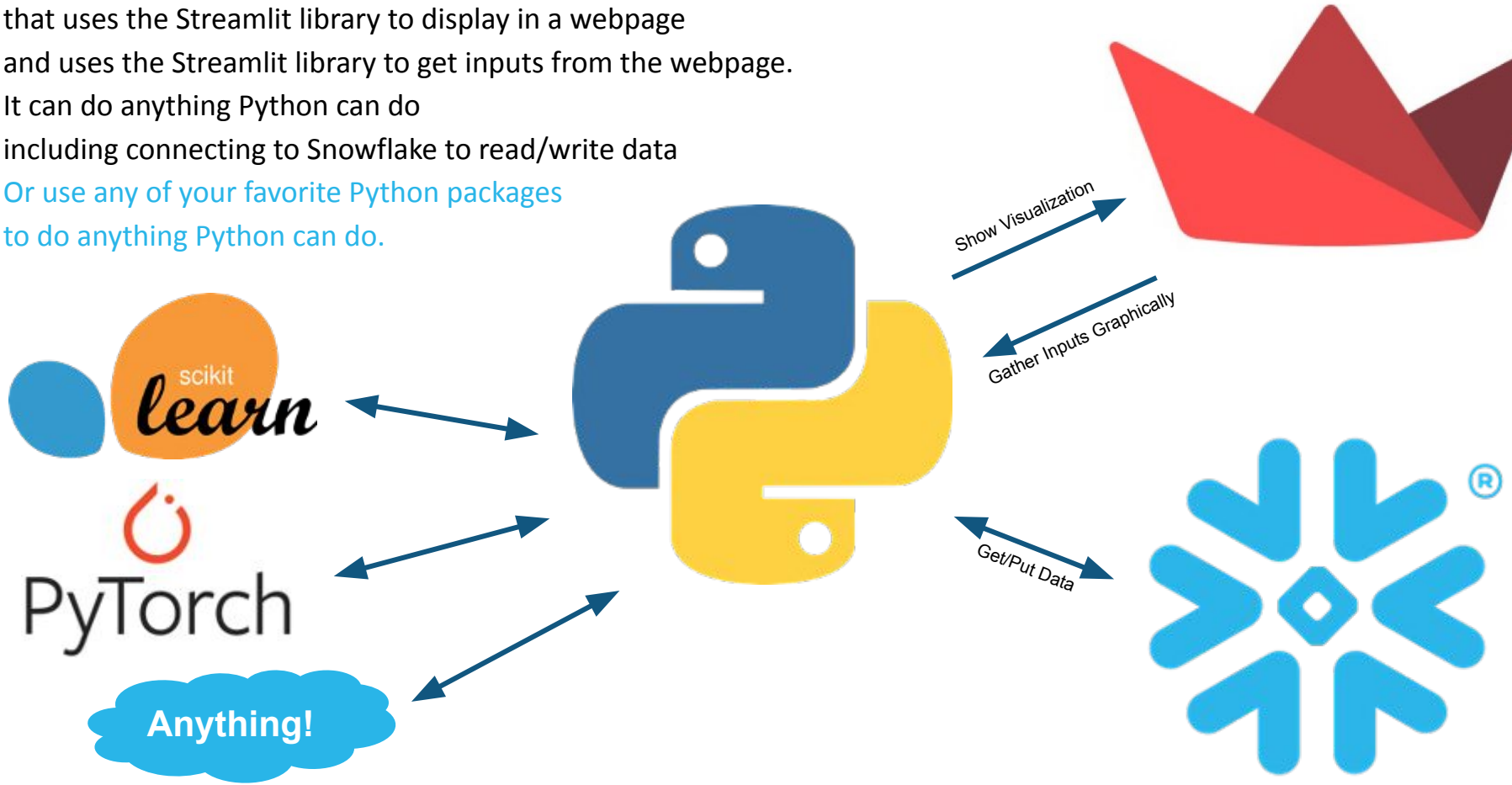
A Streamlit app is just a Python program
that uses the Streamlit library to display in a webpage
and uses the Streamlit library to get inputs from the webpage.



A Streamlit app is just a Python program
that uses the Streamlit library to display in a webpage
and uses the Streamlit library to get inputs from the webpage.
It can do anything Python can do
including connecting to Snowflake to read/write data



A Streamlit app is just a Python program
that uses the Streamlit library to display in a webpage
and uses the Streamlit library to get inputs from the webpage.
It can do anything Python can do
including connecting to Snowflake to read/write data
Or use any of your favorite Python packages
to do anything Python can do.



It only takes minutes to get started!

```
$ pip install streamlit
$ streamlit hello
```

See our docs for more getting started info
with the Streamlit open-source library

Internal resources here.

Streamlit Drives Rapid Insights

Monitoring apps

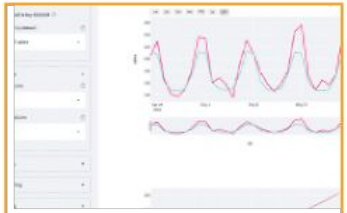
Tools that display trends and real-time insights.

Analysis tools

Apps that use machine learning to analyze new data types.

Detection apps

Tools that use computer vision to detect and track objects.

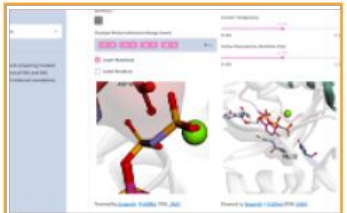


Prediction tools

Apps that predict prices, stock shortages, quality issues, etc.

Explanatory apps

Apps that analyze large datasets and present easy-to-understand insights.



Interactive tools

Tools that allow you to interact with the data to gain new insights.

https://streamlit.io/gallery

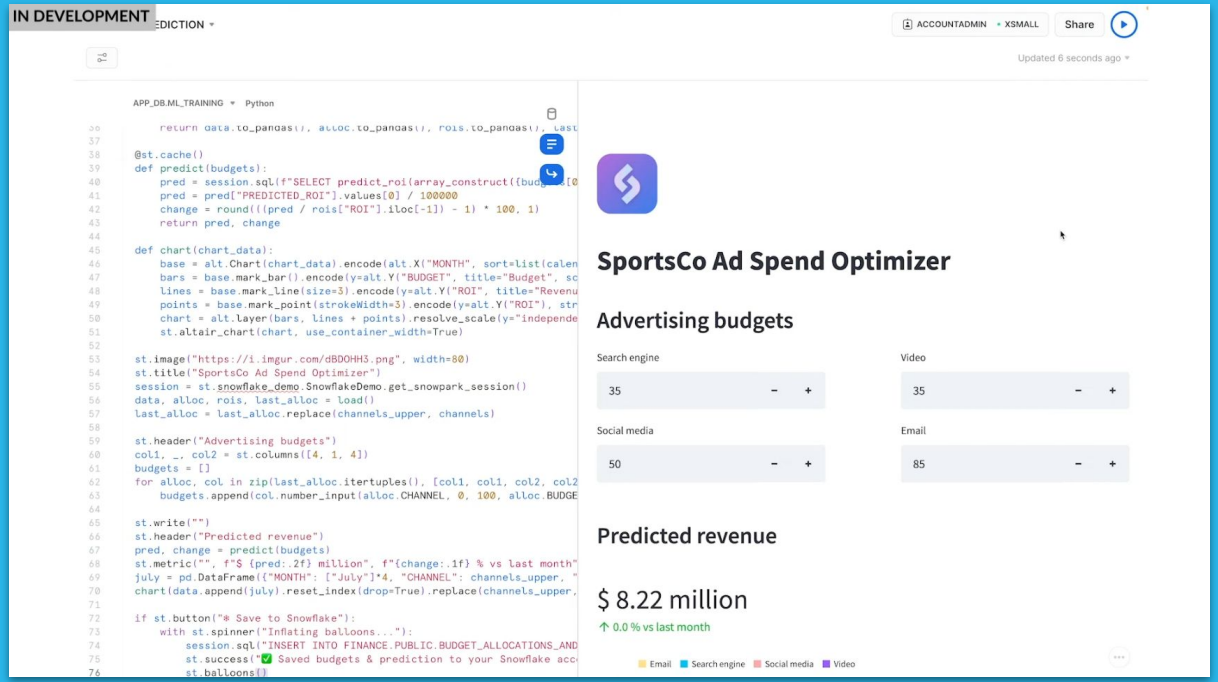
Streamlit

COMING SOON: Streamlit + Snowflake = 🤩

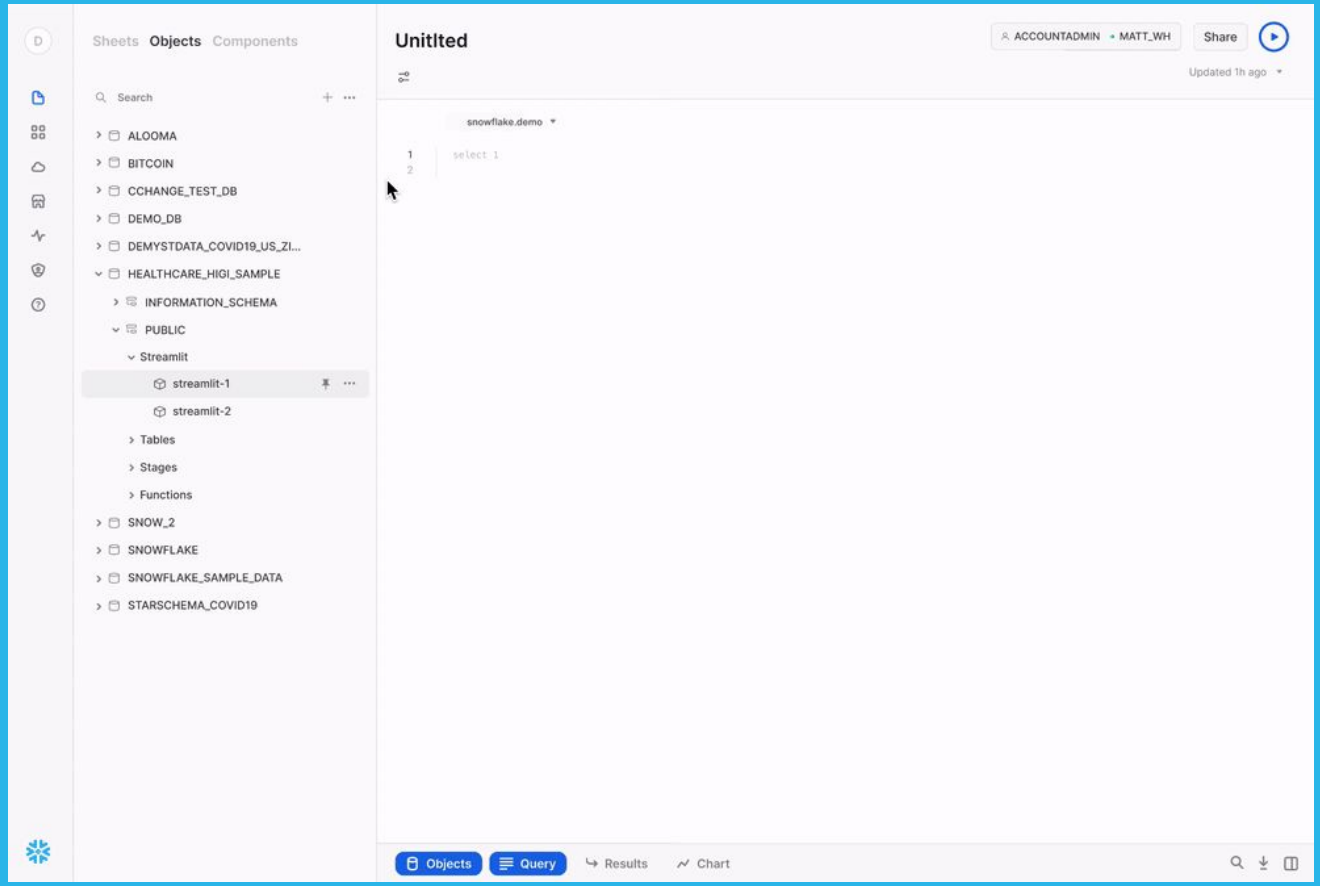
The near-term focus for Streamlit + Snowflake is about hosting Streamlit apps inside the Snowflake UI (i.e., Snowsight)

- Editing
- Sharing
- Permissions

- Make it *easy!*



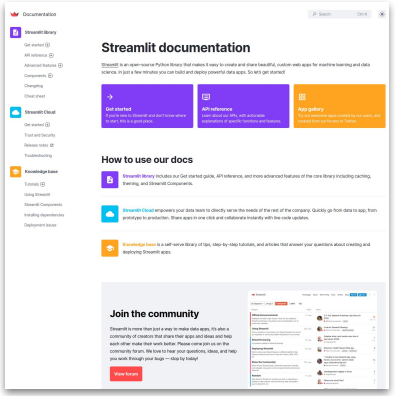
COMING SOON: Streamlit + Snowflake = 🤩



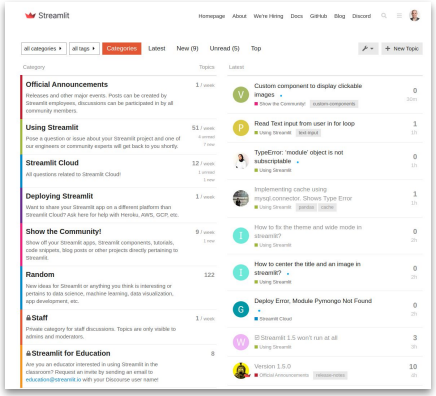
KEY TAKEAWAYS: Snowflake + Streamlit = 🪄

- Streamlit brings your data to life – build interactive apps with Python
- Today Snowflake customers can get started with the Streamlit Open Source Library
- Coming soon Snowlake integration to build, deploy, and share Streamlit apps in Snowflake

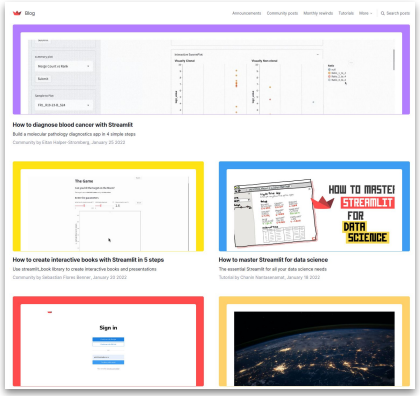
Join our community and see tons of examples in code!



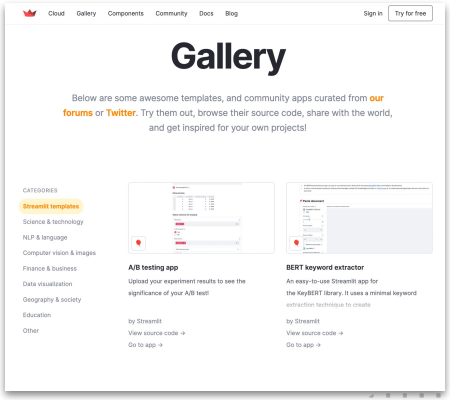
https://docs.streamlit.io



https://discuss.streamlit.io



https://blog.streamlit.io



https://streamlit.io/gallery