# Streamlit for Geospatial

May 2022

Streamlit

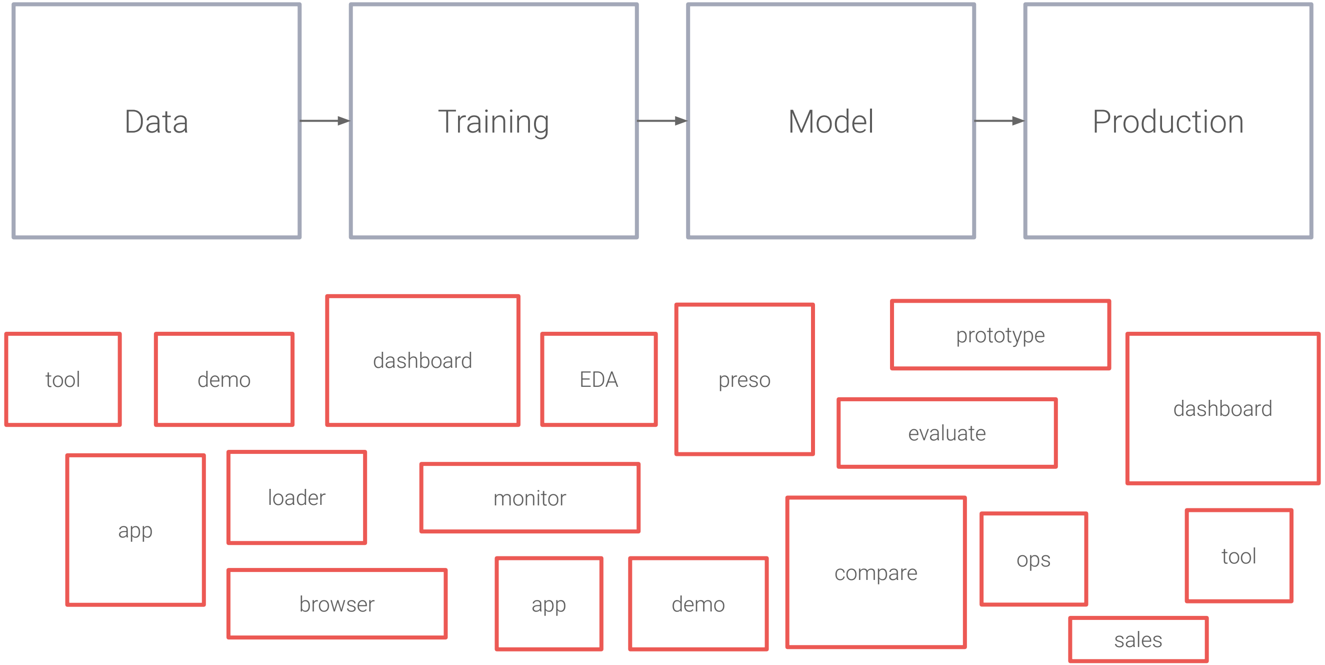
# Outline

- Introduction to Streamlit (20 mins)
- streamlit-folium demo (30 mins)
- Where to get more information and Q&A (10 mins)

# Introduction to Streamlit

# What apps do data scientists need?

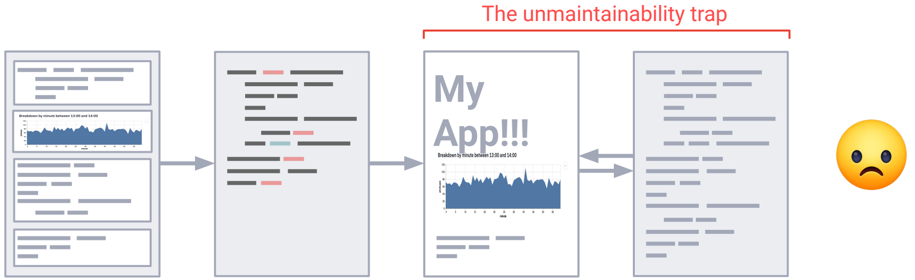
Lots! You're always making apps for your work or to communicate your results to others.



# Building data apps is slow and expensive

Building even simple data apps today requires weeks or months of investment, distracts from core work, and often yields an unmaintainable product. And because it's so costly **only a fraction of needed apps and tools are created.**

## Workflow 1: You build the app



The unmaintainability trap

Step 1
Explore in a Jupyter notebook.

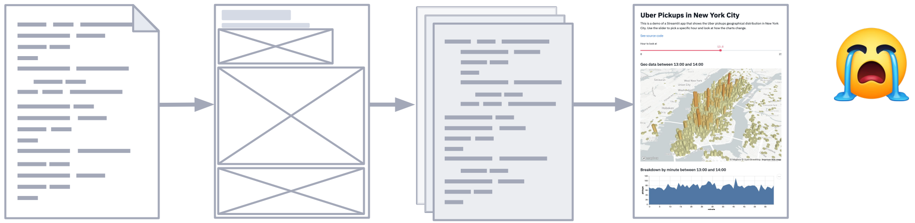
Step 2
Copy-paste into a Python script.

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

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

## Workflow 2: Your "tools team" builds the app

The frozen zone



Step 1
Collect requirements.

Step 2
Lay out wireframe and components.

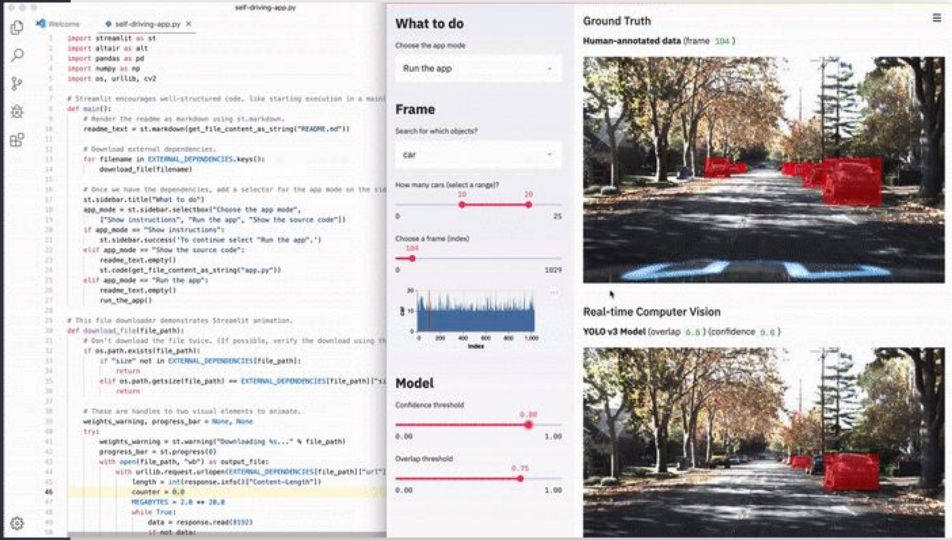
Step 3
Code up the app in CSS, React, JS,, etc.

Step 4
Released 1 month later + 3 months for updates

# Streamlit is the fastest and easiest way to create data apps

Quickly create your own elegant data apps for visualization, debugging, comparing models and presenting data - all in Python.

**Streamlit's open-source app framework** is built specifically for data scientists to rapidly create beautiful, performant apps in only a few hours!



```python
import streamlit as st
import altair as alt
import pandas as pd
import numpy as np
import os, urllib, cv2

# Streamlit encourages well-structured code, like starting execution in a main(
def main():
    # Render the readme as markdown using st.markdown.
    readme_text = st.markdown(get_file_content_as_string("README.md"))

    # Download external dependencies.
    for filename in EXTERNAL_DEPENDENCIES.keys():
        download_file(filename)

    # Once we have the dependencies, add a selector for the app mode on the sid
    st.sidebar.title("What to do")
    app_mode = st.sidebar.selectbox("Choose the app mode",
        ["Show instructions", "Run the app", "Show the source code"])
    if app_mode == "Show instructions":
        st.sidebar.success('To continue select "Run the app".')
    elif app_mode == "Show the source code":
        readme_text.empty()
        st.code(get_file_content_as_string("app.py"))
    elif app_mode == "Run the app":
        readme_text.empty()
        run_the_app()

# This file downloader demonstrates Streamlit animation.
def download_file(file_path):
    # Don't download the file twice. (If possible, verify the download using th
    if os.path.exists(file_path):
        if "size" not in EXTERNAL_DEPENDENCIES[file_path]:
            return
        elif os.path.getsize(file_path) == EXTERNAL_DEPENDENCIES[file_path]["si
            return

    # These are handles to two visual elements to animate.
    weights_warning, progress_bar = None, None
    try:
        weights_warning = st.warning("Downloading %s..." % file_path)
        progress_bar = st.progress(0)
        with open(file_path, "wb") as output_file:
            with urllib.request.urlopen(EXTERNAL_DEPENDENCIES[file_path]["url"]
                length = int(response.info()["Content-Length"])
                counter = 0.0
                MEGABYTES = 2.0 ** 20.0
                while True:
                    data = response.read(8192)
                    if not data:
```

**What to do**

Choose the app mode: Run the app

**Frame**

Search for which objects?: car

How many cars (select a range)?: 10 – 20 (0 – 25)

Choose a frame (index): 104 (0 – 1029)

**Model**

Confidence threshold: 0.80 (0.00 – 1.00)

Overlap threshold: 0.75 (0.00 – 1.00)

**Ground Truth**

**Human-annotated data** (frame 104)

**Real-time Computer Vision**

**YOLO v3 Model** (overlap 0.5) (confidence 0.6)

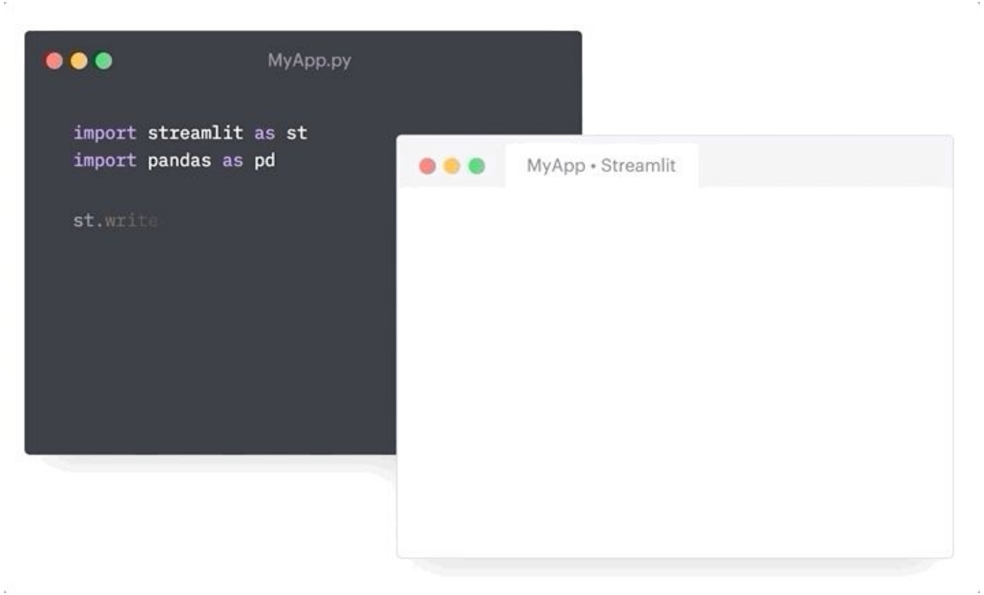
# Streamlit works on 3 simple principles

Embrace
Python scripting

Treat widgets
as variables.

Reuse data and
computation.

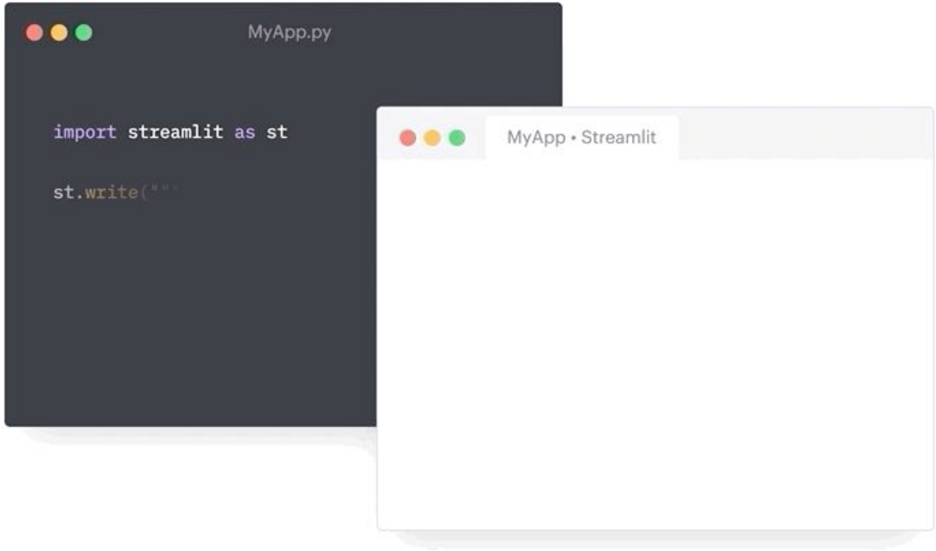
# Embrace Python scripting



```
import streamlit as st
import pandas as pd

st.write
```

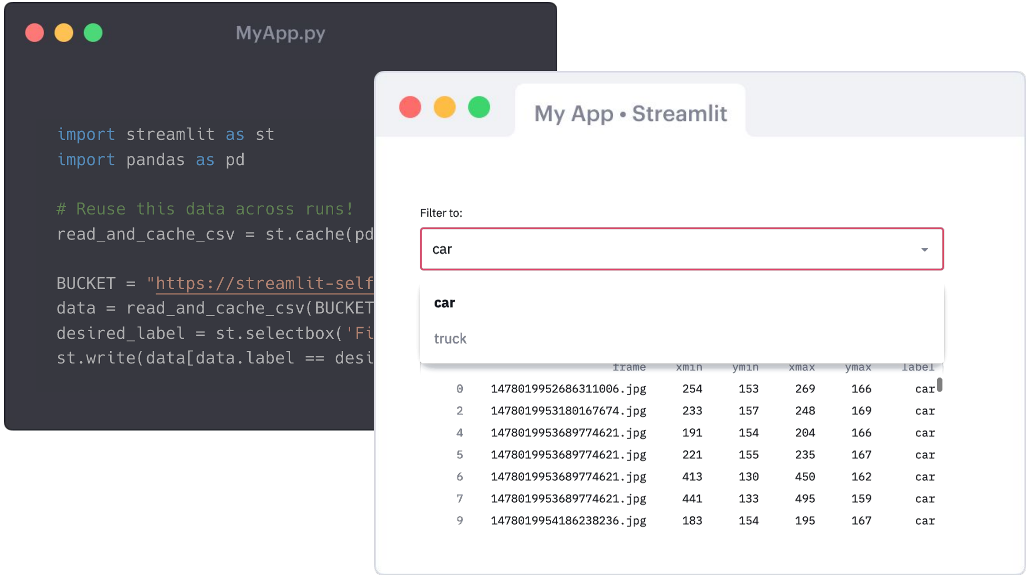
# Treat widgets as variables



```
import streamlit as st

st.write("""
```

# Reuse data and computation



```
import streamlit as st
import pandas as pd

# Reuse this data across runs!
read_and_cache_csv = st.cache(pd

BUCKET = "https://streamlit-self
data = read_and_cache_csv(BUCKET
desired_label = st.selectbox('Fi
st.write(data[data.label == desi
```

Filter to: car

|   | frame | xmin | ymin | xmax | ymax | label |
|---|---|---|---|---|---|---|
| 0 | 1478019952686311006.jpg | 254 | 153 | 269 | 166 | car |
| 2 | 1478019953180167674.jpg | 233 | 157 | 248 | 169 | car |
| 4 | 1478019953689774621.jpg | 191 | 154 | 204 | 166 | car |
| 5 | 1478019953689774621.jpg | 221 | 155 | 235 | 167 | car |
| 6 | 1478019953689774621.jpg | 413 | 130 | 450 | 162 | car |
| 7 | 1478019953689774621.jpg | 441 | 133 | 495 | 159 | car |
| 9 | 1478019954186238236.jpg | 183 | 154 | 195 | 167 | car |

# Demo

# streamlit-folium

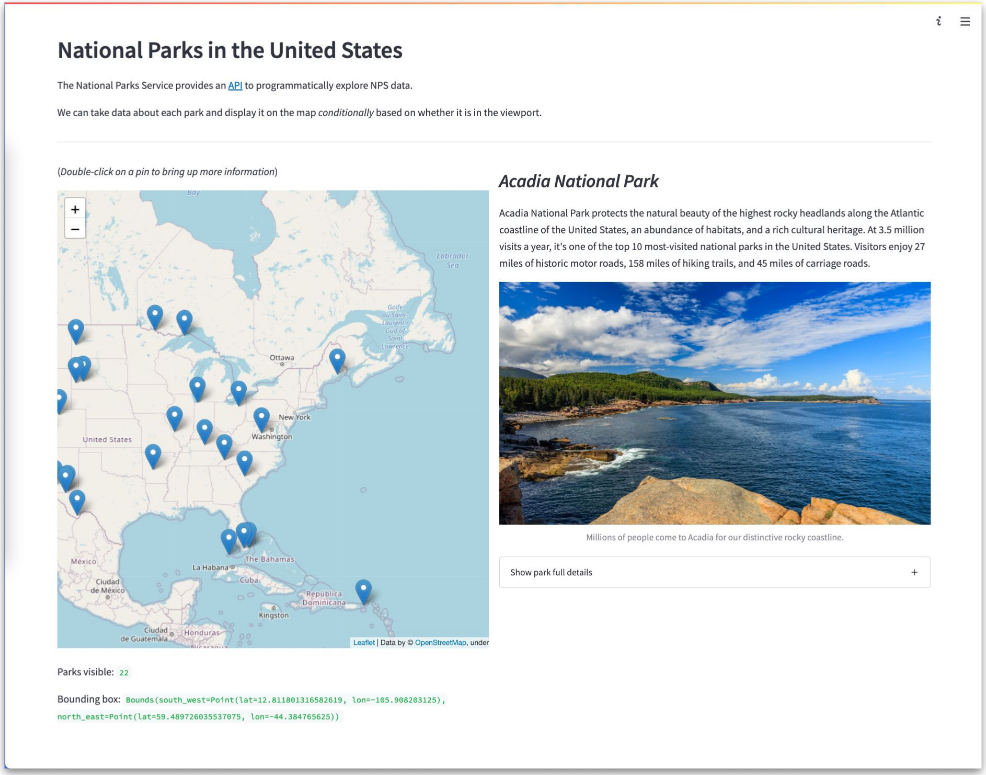
**folium** builds on the data wrangling strengths of the Python ecosystem and the mapping strengths of the Leaflet.js library

**streamlit-folium** adds bi-directional data transfer to folium, allowing for the creation of dynamic geospatial apps within Streamlit!

https://github.com/randyzwitch/streamlit-folium

Live demo apps:
https://share.streamlit.io/randyzwitch/streamlit-folium/examples/streamlit_app.py

https://share.streamlit.io/randyzwitch/streamlit-folium/examples/interactive_app.py



## National Parks in the United States

The National Parks Service provides an API to programmatically explore NPS data.

We can take data about each park and display it on the map *conditionally* based on whether it is in the viewport.

*(Double-click on a pin to bring up more information)*

### *Acadia National Park*

Acadia National Park protects the natural beauty of the highest rocky headlands along the Atlantic coastline of the United States, an abundance of habitats, and a rich cultural heritage. At 3.5 million visits a year, it's one of the top 10 most-visited national parks in the United States. Visitors enjoy 27 miles of historic motor roads, 158 miles of hiking trails, and 45 miles of carriage roads.

Millions of people come to Acadia for our distinctive rocky coastline.

Parks visible: `22`

Bounding box: `Bounds(south_west=Point(lat=12.811801316582619, lon=-105.908203125), north_east=Point(lat=59.489726035537075, lon=-44.384765625))`

# Volunteers wanted!

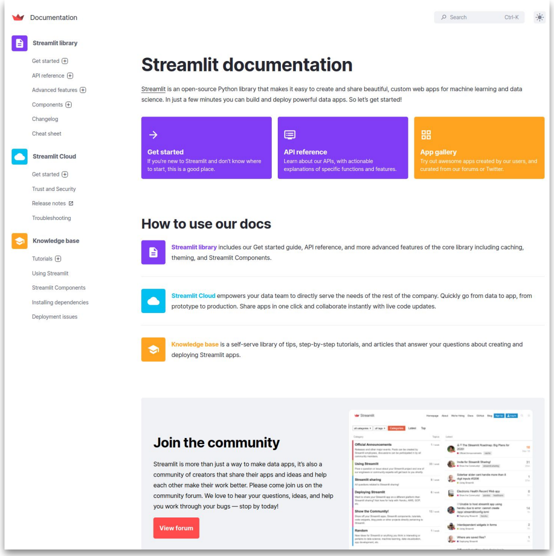
While the maintainers are Streamlit employees, streamlit-folium is an MIT-licensed *community* open-source project. Help is always appreciated!

- Improve the examples / test Folium docs examples
- Build something awesome, let us know the rough edges!
- Know JavaScript? Help us add Folium plugins!

  OR

- Just tell other people that streamlit-folium exists

# Where to get more information



## Streamlit documentation

Streamlit is an open-source Python library that makes it easy to create and share beautiful, custom web apps for machine learning and data science. In just a few minutes you can build and deploy powerful data apps. So let's get started!

**Get started** If you're new to Streamlit and don't know where to start, this is a good place.

**API reference** Learn about our APIs, with actionable explanations of specific functions and features.

**App gallery** Try out awesome apps created by our users, and curated from our forums or Twitter.

## How to use our docs

Streamlit library includes our Get started guide, API reference, and more advanced features of the core library including caching, theming, and Streamlit Components.

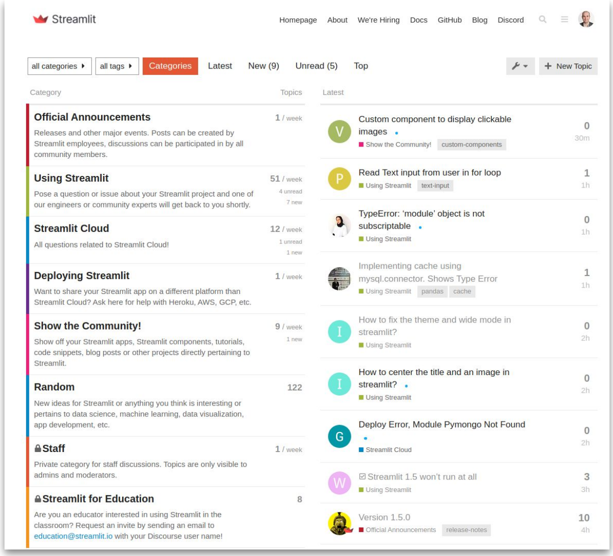
Streamlit Cloud empowers your data team to directly serve the needs of the rest of the company. Quickly go from data to app, from prototype to production. Share apps in one click and collaborate instantly with live code updates.

Knowledge base is a self-serve library of tips, step-by-step tutorials, and articles that answer your questions about creating and deploying Streamlit apps.

## Join the community

Streamlit is more than just a way to make data apps, it's also a community of creators that share their apps and ideas and help each other make their work better. Please come join us on the community forum. We love to hear your questions, ideas, and help you work through your bugs — stop by today!

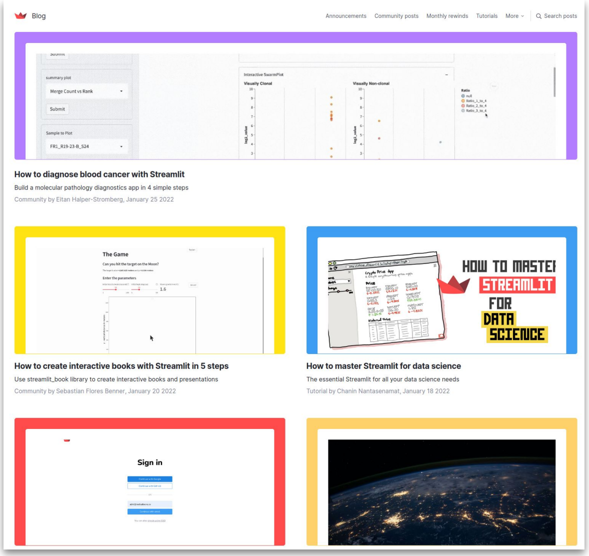
https://docs.streamlit.io



| Category | Topics |
|---|---|
| **Official Announcements**<br>Releases and other major events. Posts can be created by Streamlit employees, discussions can be participated in by all community members. | 1 / week |
| **Using Streamlit**<br>Pose a question or issue about your Streamlit project and one of our engineers or community experts will get back to you shortly. | 51 / week<br>4 unread<br>7 new |
| **Streamlit Cloud**<br>All questions related to Streamlit Cloud! | 12 / week<br>1 unread<br>1 new |
| **Deploying Streamlit**<br>Want to share your Streamlit app on a different platform than Streamlit Cloud? Ask here for help with Heroku, AWS, GCP, etc. | 1 / week |
| **Show the Community!**<br>Show off your Streamlit apps, Streamlit components, tutorials, code snippets, blog posts or other projects directly pertaining to Streamlit. | 9 / week<br>1 new |
| **Random**<br>New ideas for Streamlit or anything you think is interesting or pertains to data science, machine learning, data visualization, app development, etc. | 122 |
| **Staff**<br>Private category for staff discussions. Topics are only visible to admins and moderators. | 1 / week |
| **Streamlit for Education**<br>Are you an educator interested in using Streamlit in the classroom? Request an invite by sending an email to education@streamlit.io with your Discourse user name! | 8 |

| Latest | | |
|---|---|---|
| Custom component to display clickable images<br>Show the Community! custom-components | 0 | 30m |
| Read Text input from user in for loop<br>Using Streamlit text-input | 1 | 1h |
| TypeError: 'module' object is not subscriptable<br>Using Streamlit | 0 | 1h |
| Implementing cache using mysql.connector. Shows Type Error<br>Using Streamlit pandas cache | 1 | 1h |
| How to fix the theme and wide mode in streamlit?<br>Using Streamlit | 0 | 2h |
| How to center the title and an image in streamlit?<br>Using Streamlit | 0 | 2h |
| Deploy Error, Module Pymongo Not Found<br>Streamlit Cloud | 0 | 2h |
| ☑ Streamlit 1.5 won't run at all<br>Using Streamlit | 3 | 3h |
| Version 1.5.0<br>Official Announcements release-notes | 10 | 4h |

https://discuss.streamlit.io



**How to diagnose blood cancer with Streamlit**
Build a molecular pathology diagnostics app in 4 simple steps
Community by Eitan Halper-Stromberg, January 25 2022

**How to create interactive books with Streamlit in 5 steps**
Use streamlit_book library to create interactive books and presentations
Community by Sebastian Flores Benner, January 20 2022

**How to master Streamlit for data science**
The essential Streamlit for all your data science needs
Tutorial by Chanin Nantasenamat, January 18 2022

https://blog.streamlit.io