

**POLITECNICO**  
MILANO 1863

# Coding Low-Code Dashboards

A practical Streamlit tutorial

Francesco Invernici, Andrea Colombo  
@DEIB 05/07/2024

**DATA**  
SCIENCE GROUP

DIPARTIMENTO DI ELETTRONICA  
INFORMAZIONE E BIOINGEGNERIA

## What are we going to do?

01

### Part 1 – The Basics

- What is Streamlit? Why?
- Architecture and principles
- Fundamental components
- **From a jupyter notebook to a st dashboard**
- Pack, deploy, and expose with Docker



[streamlit.io](https://streamlit.io)

02

### Part 2 – Advanced Stuff

- Advanced concepts
- **Walkthrough of a real app**
- Tips and tricks

# The Basics

## What is Streamlit? Why?

An open-source,  
low-code, pure-Python  
framework to make web  
app quickly.



**A faster way to build  
and share data apps**

Streamlit turns data scripts into shareable web apps in minutes.  
All in pure Python. No front-end experience required.

[streamlit.io](https://streamlit.io)

... but why?

- Faster to build
- Easier to maintain
- You are not into front-ends
- Perfect for dashboard, small apps, and landing pages

# The Basics

## What is streamlit? Why?

An open  
code  
framework

A fast  
and simple

Streamlit turns  
data science  
code into  
beautiful  
dashboards

### rephrase



Generate professional sounding emails based on your direct comments - powered by Artificial Intelligence (OpenAI GPT-3) Implemented by [stefanrmmr](#) - view project source code on [GitHub](#)

#### What is your email all about?

##### SECTION - Email Input

Enter email contents down below! (currently 2x separate topics supported)

topic 1

topic 2 (optional)

Sender Name

[rephrase]

Recipient Name

[recipient]

Writing Style

formal



Generate Email

?

on  
front-ends  
board, small  
g pages

# The Basics

## What is streamlit? Why?

### Configuration

Choose your preferred API:

echarts

Choose an example

Line: Stacked Line Chart

ECharts demos are extracted from <https://echarts.apache.org/examples/en/index.html>, by copying/formatting the 'option' json object into st\_echarts. Definitely check the echarts example page, convert the JSON specs to Python Dicts and you should get a nice viz.

Made in  by @andfanilo

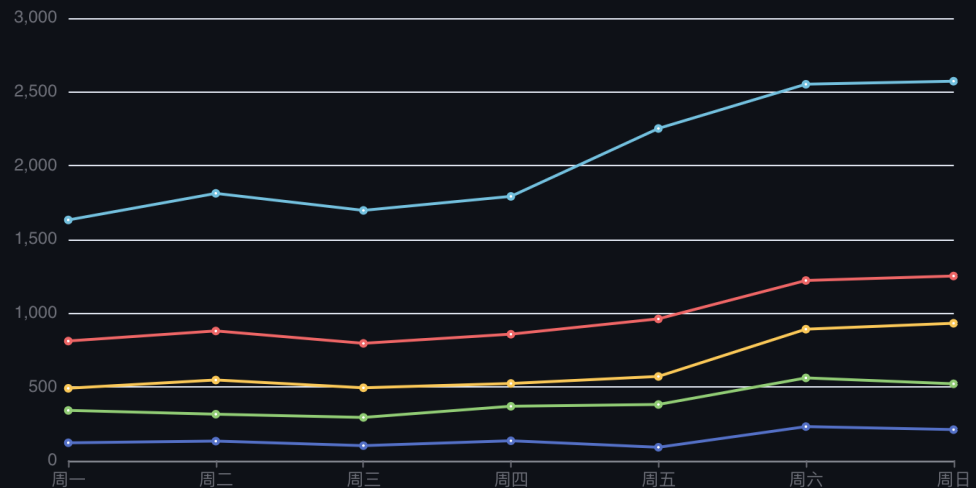


Buy me a coffee

## Streamlit ECharts Demo

折线图堆叠

邮件营销 联盟广告 视频广告 直接访问 搜索引擎



Source Code

Credit: <https://echarts.apache.org/examples/en/editor.html?c=line-stack>



# The Basics

## What is streamlit? Why?

The screenshot displays a Streamlit web application interface. On the left, a sidebar contains a 'Configuration' section with options to 'Choose your preferred /' (set to 'echarts') and 'Choose an example' (set to 'Line: Stacked Line'). Below this, there is a link to 'ECharts demos are extra' and a 'Buy me a coffee' button. The main area features the title 'Prettymapp' and four map thumbnails for Macau, Barcelona, Wunzburger, and Heerhugowaard. A configuration panel below the maps includes a 'Location address' field (Praça Ferreira do Amaral, Macau), a 'Radius' slider (100 to 1500, set to 1100), a 'Color theme' dropdown (Peach), a 'Customize map style' button, and a 'Submit' button. At the bottom, a status indicator shows 'Creating map... (may take up to a minute)'. On the right, a partial view of a line chart is visible, showing data points for '周六' (Saturday) and '周日' (Sunday).

# The Basics

## Architecture

**A Streamlit app is self contained  
in one (or more) .py script**



### **Fat server and thin client**

- ... but no hard separation!
- **Both server and client are in the same script**
- Everything in the .py script is executed on the server, just the I/O is rendered on the client(browser)
- All requests are automatically managed via WebSocket

# The Basics

## Principles

Run with *streamlit run my\_app.py*

The *my\_app.py* script is read top\_down

▶ The UI elements are placed following this order

### Development Flow

change app.py script → save it → "changes detected!" → rerun  
REPEAT

### Data Flow

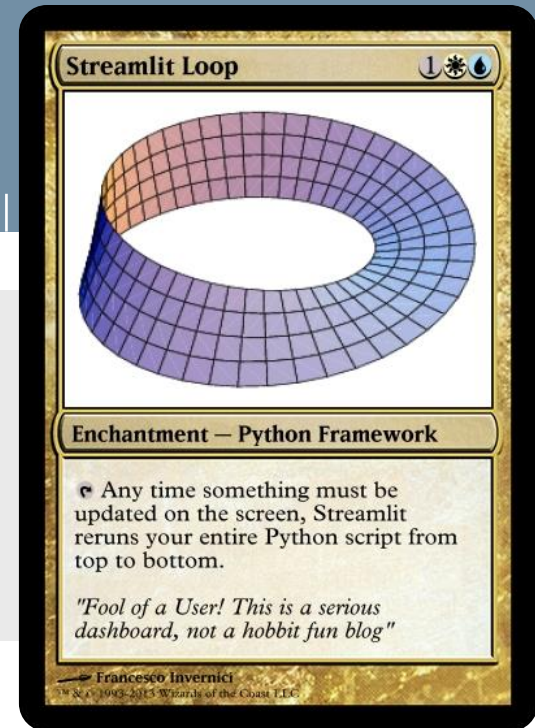
*"any time something must be updated on the screen, Streamlit reruns your entire Python script from top to bottom."*



# The Basics Principles

any time something must be updated on the screen...

- you changed the script
- **the user touched something**



## Mitigations:

- callbacks

e.g.

```
st.slider( ..., on_change=my_method)
```

- forms

- session state

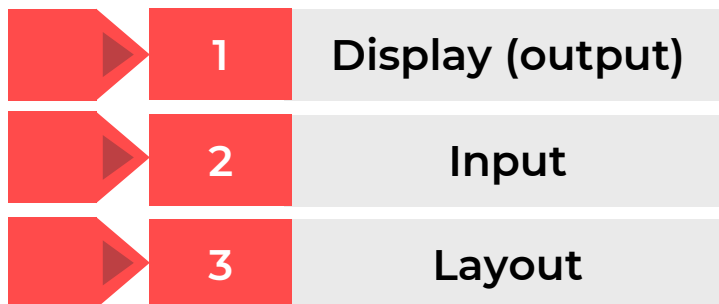
e.g.

```
st.session_state['my_string'] = 'hello'  
... time passes ...  
st.session_state.my_string >> still 'hello'
```

# The Basics

## Fundamental components

Three main categories of widget:



**Browsing the docs is fundamental!**

- <https://docs.streamlit.io/develop/api-reference/>
  - <https://docs.streamlit.io/develop/quick-reference/cheat-sheet>

# The Basics

## An Ordinary Dashboard

We will make a dashboard from a simple jupyter notebook  
*basics/an\_ordinary\_analysis.ipynb*

Tutorial  
time!

[https://github.com/FrInve/polimi\\_streamlit\\_tutorial/](https://github.com/FrInve/polimi_streamlit_tutorial/)

