StreamLit

Build interactive websites around your Python code

David Fekke, September 2025 PyJax

JaxNode needs you!

About me

David Fekke

- JaxNode user group
- Application Architect
- JS, TS, React, C#, Swift, Obj-C, Kotlin, Java and SQL
- fek.io/blog/
- youtube.com/c/polyglotengineer
- github.com/davidfekke
- @jaxnode @polyglotengine1

PRAGMATIC WORKS


HTML
JS
CSS
<h1>

StreamLit

From Data to the Web

- StreamLit is a simple web application framework for building web sites around your data
- Built in Python
- Has hot reloading
- Create web apps without knowing HTML, CSS or JS

StreamLit Features

- Open source web framework
- Built on top of the React framework
- Easily create UI for your Python programs
- Great for displaying and working with data
- Great for visualizing data

Types of Apps

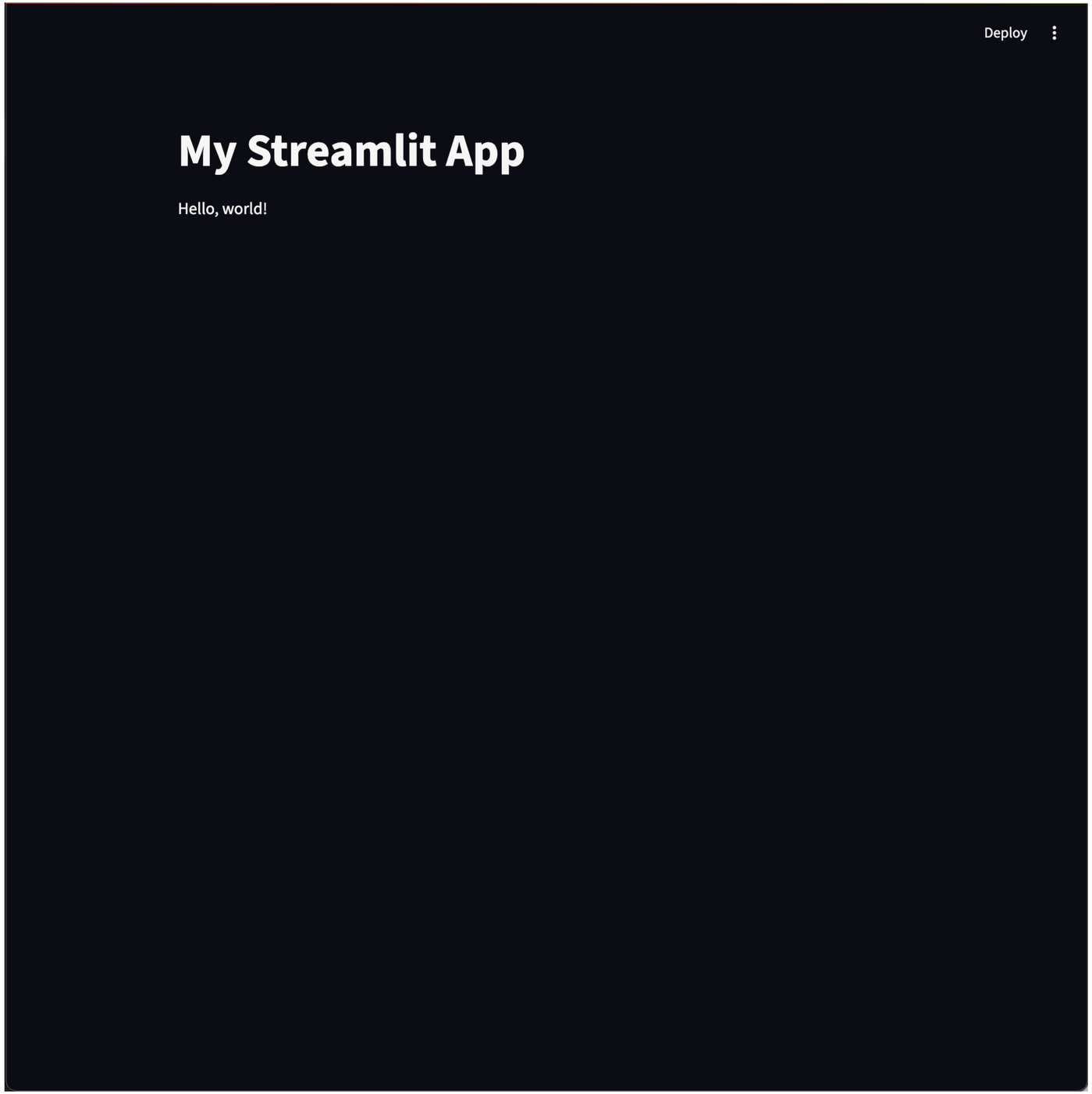
Kings of apps being built with Streamlit

- Simple interactive apps
- Data Science apps
- Dashboards
- Machine Learning apps
- AI and chatbot apps
- Visualization apps

```
import streamlit as st


st.title("My Streamlit App")

st.write("Hello, world!”)
```



Markdown

Streamlit supports markdown syntax

- Markdown is a simple way of attributing HTML tags without using HTML
- # This would be a an H1 tag
- ## This is an H2 tag
- * Would be a bullet using LI tag

Magic function

- Write is a magic function
- It will output almost any data type to your web page
- st.write(myData)
- You can import a table with Pandas and write it out to a table control

Pandas

Working with the Pandas library

- StreamLit works great with Pandas
- You can also take pandas data, and output it to graphs
- Support for Bar charts and Line charts
- The charts are also interactive

Multipage

What if you need more than one page

- There is a file structure you can follow where your streamlit apps are in a `pages` directory
- This works for simple structures only
- A menu on the side of the page will show your pages
- You can’t have nested pages

Layouts

- Streamlit by default displays each element sequentially along the vertical axis of a page
- You can create a columns using the st.columns function
- Columns output a table with all of you column objects
- You can then add elements to a column

Input

Builtin controls for input

- Button, Download Button
- Link Button
- Checkbox, Radio button, Color picker, Select, Multi-select
- Filters, Sliders, Toggle

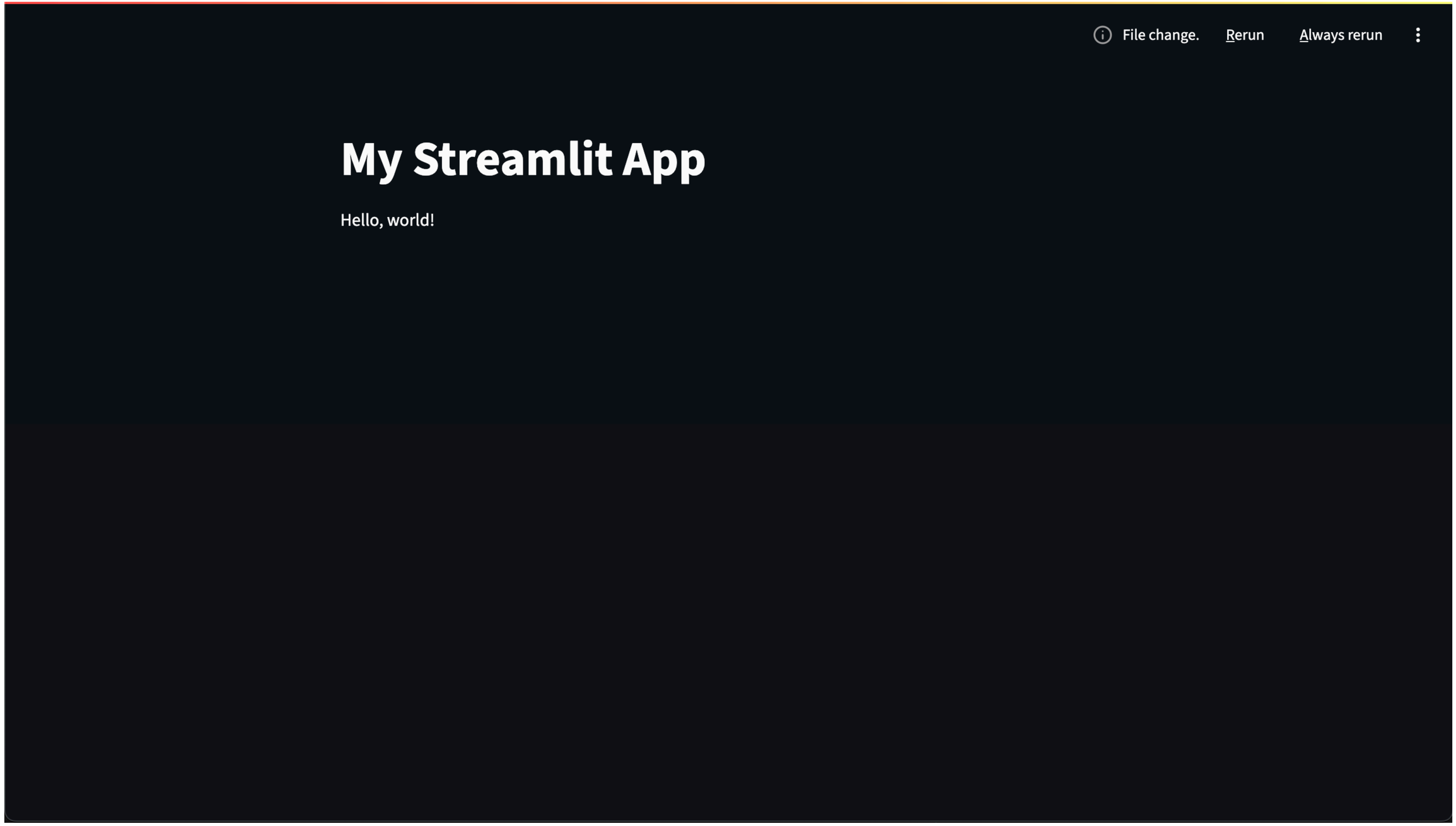
Running

How to run and test locally

- Once you have written your Streamlit app using the following command in your terminal
- $ streamlit run main.py

```
You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.4.50:8501
```


File change.
Rerun
Always rerun
My Streamlit App
Hello, world!

Other commands

Streamlit CLI

- $ streamlit init # Will init a new app
- $ streamlit docs # opens the documentation
- $ streamlit config show # Show your configurations
- $ streamlit cache clear # Clears the cache

Publishing your streamlit app

Hosting

- You can self host
- Streamlit has a hosting service you can use
- Push your app using a Docker container

Demo

Summary

- Use Streamlit for simple interactive apps
- If you need more sophisticated routing, use a regular framework like flask or Next.js
- If you have have pixel specific control over UI, use a framework with broad CSS control
- If you need to build something fast, Streamlit is a great option

Questions