Python

Streamlit

Alish Bista - 11/11/2025

Streamlit

Streamlit

Easy Data Web Apps

- Turns Python scripts to web apps.
- Pure Python. No HTML, CSS, or JavaScript needed
- Perfect for dashboards, data apps, and quick demos

```
pip install streamlit
```

Create Streamlit App

Creating & Running an App

- Create a file called app.py (or any other suitable name)
- Add the code
- Run it with *streamlit run app.py*
- Browser automatically opens at *http://localhost:8501*

```
import streamlit as st


st.title("My First App")
st.write("Hello World!")
```

Text

Displaying Text

- Provides multiple ways to display text with different hierarchy levels.
- Key Functions:
  - **title()** - Main page title (largest)
  - **header()** - Section headers
  - **subheader()** - Subsection headers
  - **write()** - Most flexible (auto-formats)
  - **markdown()** - For rich formatting

```
import streamlit as st

# Different text sizes
st.title("Main Title")
st.header("Section Header")
st.subheader("Subsection")
st.text("Plain text")
st.write("Flexible write function")

# Markdown formatting
st.markdown("**Bold** and *italic*")
st.markdown("- Item 1\n- Item 2")
```

Data

Displaying Data

- Can display tabular data and key metrics.
- Display Options:
  - **dataframe()** - Interactive (sortable, scrollable)
  - **table()** - Static HTML table
  - **metric()** - Display KPIs with deltas

```
# import required packages here

df = pd.DataFrame({
    'Name': ['Alice', 'Bob', 'Carol'],
    'Sales': [100, 150, 120],
    'Region': ['North', 'South', 'East']
})

st.dataframe(df)
st.table(df)
st.metric(
    label="Revenue",
    value="$1.2M",
    delta="+12%"
)
```

User Inputs

Adding Widgets

- Widgets let users interact with the app.
- When a user changes a widget value, Streamlit reruns the script with the new value.
- Common Widgets:
  - Buttons - Trigger actions
  - Sliders - Select numeric ranges
  - Selectbox - Dropdown menus
  - Text input - Collect text data
  - Checkboxes - Toggle options

```
if st.button("Click me"):
    st.write("Button clicked!")

age = st.slider("Age", 0, 100, 25)
st.write(f"Selected: {age}")

city = st.selectbox(
    "City",
    ["NYC", "LA", "Chicago"]
)

name = st.text_input("Your name")
st.write(f"Hello, {name}!")

if st.checkbox("Show data"):
    st.write("Data displayed!")
```

Embedding Viz

Adding Charts and Visualization

Two Ways to Visualize:

- Built-in Streamlit charts - Quick and simple for basic needs
- Matplotlib/Seaborn - Full control and professional-quality charts

- Create matplotlib figure. Pass to st.pyplot()

```
# import required packages here …

df = pd.DataFrame({
    'x': [1, 2, 3, 4],
    'y': [10, 20, 30, 40]
})

st.line_chart(df)
st.bar_chart(df)

fig, ax = plt.subplots()
sns.barplot(data=df, x='x', y='y', ax=ax)
ax.set_title('My Chart')
st.pyplot(fig)
```

Handling Files

Working With Files

- Can let users upload their own data files for analysis.
- Key Concepts:
  - File Uploader - Returns file object or None if nothing uploaded
  - Supported Formats: CSV, Excel, JSON
  - Always check if file exists before processing to avoid errors

```
#import here

uploaded = st.file_uploader(
    "Upload CSV",
    type="csv"
)

if uploaded is not None:
    df = pd.read_csv(uploaded)

    st.dataframe(df)
    st.write(f"Rows: {len(df)}")
    st.write(f"Columns: {len(df.columns)}")

else:
    st.info("Please upload a CSV file")
```

Deployment

Share the Data App

- Streamlit Community Cloud offers hosting for public apps
- Deployment Steps:
  - Push code to GitHub (include requirements.txt)
  - Visit share.streamlit.io
  - Connect your GitHub repo
  - Click Deploy
  - Get shareable URL
- Create a Requirements.txt that lists all packages your app needs.

```
import streamlit as st
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

st.title("My Data App")

# App code here...
```

Q&A