

Streamlit

- Open-source Python library
- Let's us create **interactive web apps** for data visualization and ML
- Installed as: **pip install streamlit**

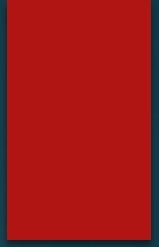
Features

- Pure Python: We write app using Python. HTML, CSS, JS not required
- Instant UI: Widgets like sliders, buttons, and dropdowns are created with a single line of code.
- Live Update
- Fast Deployment

Use Case

- Data Dashboard
- ML model Demos
- Interactive Reports
- Internal tools for teams

Text Elements



```
import streamlit as st

st.title("My Dashboard")
st.header("Overview")
st.subheader("Key Highlights")
st.text("Streamlit makes text rendering easy.")
st.markdown("**Bold Example** | Italic Example")

# streamlit run file_name.py
```

Interactivity

Text and Number Inputs

- `st.text_input("Enter text")`
- `st.text_area("Free Text", "Overwrite this")`
- `num = st.number_input("Enter a Quantity", 1, 10)`
- `dt = st.date_input("DOB", date(2000, 1, 20))`
- `tim = st.time_input("Tim", time(10, 3))`

Selection Inputs

- `st.button("Click Me")`
- `st.checkbox("Check Me")`
- `gender = st.radio("Gender", ["Male", "Female"], horizontal=True)`
- `st.selectbox("Choose option", ["A", "B", "C"])`
- `skills = st.multiselect("Select Skills", ["Python", "DS", "SQL", "Java"])`
- `st.slider("Pick a value", 0, 100)`
- `st.color_picker("Pick Color")`

Example

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

age = st.slider("Select age", 1, 100, 25)
st.write("You selected:", age)

name = st.text_input("Enter your name")
st.write("Hello,", name)
```

File Input

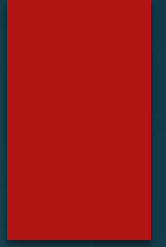
- `uploaded_file = st.file_uploader("Choose a file")`
- `img = st.camera_input()`

Code

```
uploaded_file = st.file_uploader("Upload a CSV")
if uploaded_file:
    df = pd.read_csv(uploaded_file)
    st.write(df.head())
```

More at: <https://docs.streamlit.io/develop/api-reference/widgets?>

Displaying Data



- `st.dataframe(df)` → interactive table
- `st.table(df)` → Static Table
- `st.line_chart(df)` → quick line chart
- `st.bar_chart(df)` → bar chart
- `st.pyplot(fig)` → Matplotlib / sns fig object

Code

```
df = pd.read_csv(file_path)
st.dataframe(df, hide_index=True)
st.bar_chart(data=df, x="Team", y="Height")
```


Layout

- For creating multi-column dashboards

Code

```
col1, col2, col3 = st.columns(3)
col1.write("First Column")
col2.write("Second Column")
col3.write("Third Column")
```


Testing ML Model

```
import streamlit as st
import numpy as np
import joblib
```

```
model = joblib.load("iris_knn.joblib")
st.title("🌸 Iris Flower Prediction App")
st.markdown("Fill in the measurements below to predict the Iris species:")
```

```
# Create a form
with st.form("iris_form"):
    sepal_length = st.slider("Sepal length (cm)", 4.0, 8.0, 5.0)
    sepal_width = st.slider("Sepal width (cm)", 2.0, 4.5, 3.0)
    petal_length = st.slider("Petal length (cm)", 1.0, 7.0, 4.0)
    petal_width = st.slider("Petal width (cm)", 0.1, 2.5, 1.0)
    submitted = st.form_submit_button("🔮 Predict")
```

```
if submitted:
    features = np.array([[sepal_length, sepal_width, petal_length, petal_width]])
    prediction = model.predict(features)
    species = ["Setosa", "Versicolor", "Virginica"][prediction[0]]
    st.success(f"✅ Predicted Species: **{species}**")
```

```
# Show probability distribution if available
if hasattr(model, "predict_proba"):
    probs = model.predict_proba(features)[0]
    st.write("📊 Prediction Probabilities:")
    st.bar_chart(probs)
```



Dashboard

Deploy App

- Making your app accessible to users via a web URL
- Share ML dashboards, visualizations, or interactive tools

Options

- Streamlit Community cloud
- Heroku
- AWS / Azure / GCP
- Docker Container

Deployment Steps

1. Push your app to GitHub
 - Containing **app.py** and **requirements.txt**
2. Sign in to Streamlit cloud and connect GitHub account
3. Click on **New App** → Select Repository, Branch, File Path & Deploy
4. Access URL