STREAMLIT BEGINNER'S COMPLETE GUIDE

Streamlit is an open-source Python framework designed to help you build highly interactive web applications using pure Python. It is extremely popular among data analysts, data scientists, and machine learning engineers because it eliminates the need for frontend technologies like HTML, CSS, and JavaScript. Streamlit transforms simple Python scripts into powerful web apps instantly.

WHY STREAMLIT?

1. Easy to Use – Write Python code and instantly get a web app.
2. Fast Development – No frontend coding required.
3. Live Reload – App updates every time you save changes.
4. Supports Popular Libraries – Works with pandas, NumPy, Matplotlib, Plotly, Scikit-learn, etc.
5. Great for Data Apps – Dashboards, ML models, data visualizations, forms, etc.
6. Simple Deployment – Publish using Streamlit Cloud, HuggingFace, Render, etc.

INSTALLATION

Install Streamlit: pip install streamlit
Run example app: streamlit hello

YOUR FIRST STREAMLIT APP

Create a file `app.py` with the following code:
import streamlit as st st.title("Hello Streamlit")
st.write("This is my first Streamlit app!")
Run it: streamlit run app.py

TEXT & DISPLAY ELEMENTS

Use the following Streamlit functions:
• st.title("Title")
• st.header("Header")
•st.subheader("Subheader")
• st.text("Simple text")
• st.markdown("Markdown content")
•st.write("General-purpose writer")
You can also display formatted text and HTML-like content using markdown.

INPUT WIDGETS

Streamlit provides a wide range of input elements:

• st.text_input("Enter name")
•st.number_input("Enter value")
• st.slider("Pick a number", 1, 10)
• st.selectbox("Choose option", ["A","B", "C"])
• st.multiselect("Pick multiple", ["A", "B", "C"])
• st.checkbox("Accept terms")
• st.radio("Select one", ["Yes", "No"])
• st.button("Submit")

DISPLAYING DATA

Use:

• st.table(df) – Static table
• st.dataframe(df) – Interactive table
• st.json(data) – Display JSON
•st.metric(label, value, delta)

These make your data interactive and easy to read.

CHARTS & VISUALIZATIONS

Streamlit supports:

1. Matplotlib:

```
import matplotlib.pyplot as plt
fig, ax = plt.subplots()
ax.plot([1,2,3],[4,5,6])
st.pyplot(fig)
```

2. Plotly:

```
import plotly.express as px
fig = px.line(df, x='year', y='sales')
st.plotly_chart(fig)
```

3. Altair:

```
import altair as alt
st.altair_chart(chart)
```

FILE UPLOAD & DOWNLOAD

Upload

```
file = st.file_uploader("Upload CSV")
```

Download

```
st.download_button("Download", data="Hello", file_name="text.txt")
```

LAYOUTS

Columns

```
col1, col2 = st.columns(2)
col1.write("Left")
col2.write("Right")
```

Sidebar

```
st.sidebar.header("Filters")
st.sidebar.selectbox("Choose", ["A","B"])
```

Tabs

```
tab1, tab2 = st.tabs(["Home", "Analytics"])
```

EXAMPLE: DASHBOARD APP

```
import streamlit as st
import pandas as pd

df = pd.read_csv("data.csv")

st.title("Sales Dashboard")
product = st.sidebar.selectbox("Select Product", df['Product'].unique())

filtered = df[df['Product'] == product]
st.dataframe(filtered)

st.bar_chart(filtered['Sales'])
```

EXAMPLE: MACHINE LEARNING APP

```
import streamlit as st
from sklearn.datasets import load_iris
from sklearn.ensemble import RandomForestClassifier

iris = load_iris()
X = iris.data
y = iris.target

model = RandomForestClassifier().fit(X, y)

st.title("Iris Flower Predictor")

sl = st.slider("Sepal Length", 4.0, 8.0)
sw = st.slider("Sepal Width", 2.0, 4.4)
pl = st.slider("Petal Length", 1.0, 7.0)
pw = st.slider("Petal Width", 0.1, 2.5)

prediction = model.predict([[sl, sw, pl, pw]])
st.write("Prediction:", iris.target_names[prediction][0])
```

ADVANCED TOPICS

- Multipage apps
- Caching data: st.cache_data
- Caching models/resources: st.cache_resource
- Custom CSS/styling
- Themes and branding
- Streaming text outputs
- Embedding maps, videos, and interactive elements

LEARNING ROADMAP

1. Basics: Text, inputs, tables
2. Layout: Columns, tabs, sidebars
3. Data: Charts, metrics,visualizations

4. Files: Upload, download
5. State: Session management
6. Apps: Dashboards, MLapps
7. Deployment: Streamlit Cloud, HuggingFace Spaces, Render

Perfect Path to Master Streamlit (Beginner → Advanced)

Step 1: Basics

Title, headers, write
Input widgets
Dataframes
Charts

Step 2: Layout
Sidebar
Columns
Tabs
Expanders

Step 3: File Handling
File upload
File download

Step 4: State Management
st.session_state
st.cache_data
st.cache_resource

Step 5: Building Projects
Dashboard
ML predictor
Inventory system
Chatbot interface
Data form app

Step 6: Deployment
Streamlit Cloud
Docker + Render
GitHub + HuggingFace