Streamlit Cheat Sheet: Beginner to Advanced

1. Getting Started

```
import streamlit as st

# Run app with: streamlit run app.py
```

- **Install Streamlit:**

```
pip install streamlit
```

2. Basic App Structure & Layout

```
st.title("My Streamlit App")
st.header("Header Text")
st.subheader("Subheader Text")
st.text("Simple text")
st.write("Display objects:", 123)
```

- **Markdown:**

```
st.markdown("**Bold**, *italic*, $\\LaTeX$")
```

- **Sidebar Usage:**

```
st.sidebar.title("Options")
var = st.sidebar.slider("Pick a number:", 1, 100, 50)
```

3. Input Widgets

Widget	Example	Returns
Button	`st.button("Click me")`	True if clicked
Checkbox	`st.checkbox("Accept terms")`	True/False
Radio	`st.radio("Choose one", ["A", "B"])`	Selected value
Selectbox	`st.selectbox("Pick", )`	Selected value

Multiselect	`st.multiselect("Pick", ["a", "b", "c"])`	List of selected
Slider	`st.slider("Age", 0, 130, 25)`	Value
Number input	`st.number_input("Number", min_value=0, max_value=100)`	Number
Date/Time input	`st.date_input("Birthday")`, `st.time_input("Time")`	Date/Time
File uploader	`st.file_uploader("Upload CSV")`	Uploaded file object
Text input, area	`st.text_input("Name")`, `st.text_area("Comments")`	String

4. Data Display

- **Tables:**

```
st.dataframe(df)        # Interactive
st.table(df)            # Static
```

- **JSON/Dict:**

```
st.json(my_dict)
```

- **Metrics:**

```
st.metric(label="Temperature", value="23°C", delta="+1°C")
```

5. Plotting and Charts

- **matplotlib/seaborn/plotly/altair:** Directly display most Python charts:

```
import matplotlib.pyplot as plt
fig, ax = plt.subplots()
ax.plot([1,2,3],[1,4,9])
st.pyplot(fig)


import plotly.express as px
fig = px.scatter(df, x="col1", y="col2")
st.plotly_chart(fig)
```

- **Built-in Charts:**

```
st.line_chart(df)
st.bar_chart(df)
st.area_chart(df)
```

6. Layout & Containers

- **Columns:**

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

- **Expander/Accordion:**

```
with st.expander("More details"):
    st.write("Hidden content...")
```

- **Tabs:**

```
tab1, tab2 = st.tabs(["Tab 1", "Tab 2"])
with tab1: st.write("Tab 1 Content")
```

7. Media & Files

Task	Example
Image	`st.image("file.png", caption="An image")`
Audio	`st.audio(open("audio.mp3", "rb").read())`
Video	`st.video(open("video.mp4", "rb").read())`
Download	`st.download_button("Download", data, file_name="x.csv")`

8. State Management

```
if "counter" not in st.session_state:
    st.session_state.counter = 0
if st.button("Increment"):
    st.session_state.counter += 1
```

```
st.write(st.session_state.counter)
```

9. Caching & Performance

```
@st.cache_data
def get_data(path):
    return pd.read_csv(path)
```

- Use `@st.cache_resource` to cache loaded models or large resources.

10. Advanced Interactivity

- **Form (multiple inputs, single submit):**

```
with st.form("my_form"):
    name = st.text_input("Name")
    submitted = st.form_submit_button("Submit")
if submitted:
    st.write("Hello", name)
```

- **Progress bar and status:**

```
import time
bar = st.progress(0)
for i in range(100):
    time.sleep(0.01)
    bar.progress(i+1)
st.success("Done!")
```

11. Deployment

- **Local:**

```
streamlit run app.py
```

- **Cloud:**
  - Streamlit Community Cloud
  - Others: Heroku, AWS, GCP, Azure (use requirements.txt for dependencies)

12. Extensions & Integrations

Feature	Tool/Example
Authentication	streamlit-authenticator
Database	SQLAlchemy, streamlit connections
Map visualizations	st.map(), pydeck, Folium
Custom themes	.streamlit/config.toml
Component ecosystem	streamlit-components

13. Pro Tips & Best Practices

- **Layout**: Use sidebar, columns, and containers for intuitive UI.
- **Responsiveness**: Minimize slow computations—use caching.
- **Versioning**: Keep dependencies in `requirements.txt`.
- **Debugging**: Use `st.write(vars)` to log/check app variables.
- **Docs**: Official docs at https://docs.streamlit.io

14. Example Minimal App

```
import streamlit as st
import pandas as pd

st.title("Demo Streamlit App")
df = pd.DataFrame({"x": range(10), "y": [i**2 for i in range(10)]})
if st.checkbox("Show DataFrame"):
    st.dataframe(df)
if st.button("Plot"):
    st.line_chart(df.set_index("x"))
```