Streamlit Library

Installation

pip install streamlit

Run Python Script

streamlit run [filename] [-- optional script args]

streamlit hello (in-built demo script)

Some Basic Streamlit Functions

st.text – writes raw text to app

st.write – standard function for writing anything to screen (“Swiss Army Knife” of Streamlit)

[Streamlit magic: type a variable on its own line and st.write will be applied to it]

st.line_chart – line chart

@st.cache decorator – caches pieces of data

st.title – give your app a title

st.dataframe, st.table – display data (table is static, dataframe is more dynamic)

st.empty – placeholder, “save” a slot in your app for later

st.balloons – celebratory balloons animation

Widgets

- st.slider
- st.button
- st.selectbox

Example

x = st.slider(‘x’)

st.write(x, ‘squared is’, x*x)

Sidebar

Organize your widgets in a left panel sidebar.

- st.sidebar

Use st.sidebar.slider instead of st.slider.

Same with other widgets.

Exceptions:

Use st.sidebar.markdown instead of st.write

Can't use st.echo, st.spinner in sidebar.

Caching

Use @st.cache in the line before a function. It checks 4 things:

1. Function parameters
2. External variables used in function
3. Body of function
4. Body of function used in cached function

- First time seeing these 4 components with these exact values -> run function & cache
- Any subsequent time the function is called with these 4 things the same, function isn't run. Returns output from cache.
- Keeps track of changes via hashing
- Cache is a key-value store in memory
- Key = Hash of the 4 things above. Value = Tuple of cached output & hash of cached output
- st.cache supports arguments. See here: https://docs.streamlit.io/api.html#streamlit.cache

Charts and Maps

- Support for several charting libraries, e.g. Matplotlib, Altair, Deck.Gl, Plotly etc.
- Line chart: st.line_chart
- Map: st.map

Display Progress

- st.progress function