Streamlit cheat sheet

Summary of the docs, as of Streamlit v1.0.0.

How to install and import

```
$ pip install streamlit
```

Import convention

```
>>> import streamlit as st
```

Add widgets to sidebar

```
st.sidebar.<widget>
>>> a = st.sidebar.radio('R:',[1,2
```

Command line

```
$ streamlit --help
$ streamlit run your_script.py
$ streamlit hello
$ streamlit config show
$ streamlit cache clear
$ streamlit docs
$ streamlit --version
```

Pre-release features

Beta and experimental features

```
pip uninstall streamlit
pip install streamlit-nightly --up
```


Magic commands

```
# Magic commands implicitly `st.write()`
''' _This_ is some __Markdown__ '''
a=3
'dataframe:', data
```

Display text

```
st.text('Fixed width text')
st.markdown('_Markdown_') # see *
st.caption('Balloons. Hundreds of them...')
st.latex(r''' e^{i\pi} + 1 = 0 ''')
st.write('Most objects') # df, err, func, keras!
st.write(['st', 'is <', 3]) # see *
st.title('My title')
st.header('My header')
st.subheader('My sub')
st.code('for i in range(8): foo()')

* optional kwarg unsafe_allow_html = True
```

Display data

```
st.dataframe(my_dataframe)
st.table(data.iloc[0:10])
st.json({'foo':'bar','fu':'ba'})
st.metric(label="Temp", value="273 K", delta="1.2 K")
```

Display charts

```
st.line_chart(data)
st.area_chart(data)
st.bar_chart(data)
st.pyplot(fig)
st.altair_chart(data)
st.vega_lite_chart(data)
st.plotly_chart(data)
st.bokeh_chart(data)
st.pydeck_chart(data)
st.deck_gl_chart(data)
st.graphviz_chart(data)
st.map(data)
```

Display media

```
st.image('./header.png')
st.audio(data)
st.video(data)
```

Display interactive widgets

```
st.button('Hit me')
st.download_button('On the dl', data)
st.checkbox('Check me out')
st.radio('Radio', [1,2,3])
st.selectbox('Select', [1,2,3])
st.multiselect('Multiselect', [1,2,3])
st.slider('Slide me', min_value=0, max_value=10)
st.select_slider('Slide to select', options=[1,'2'])
st.text_input('Enter some text')
st.number_input('Enter a number')
st.text_area('Area for textual entry')
st.date_input('Date input')
st.time_input('Time entry')
st.file_uploader('File uploader')
st.color_picker('Pick a color')
```

Use widgets' returned values in variables:

```
>>> for i in range(int(st.number_input('Num:'))): foo()
>>> if st.sidebar.selectbox('I:',['f']) == 'f': b()
>>> my_slider_val = st.slider('Quinn Mallory', 1, 88)
>>> st.write(slider_val)
```

Control flow

```
st.stop()
```

Lay out your app

```
st.form('my_form_identifier')
st.form_submit_button('Submit to me')
st.container()
st.columns(spec)
>>> col1, col2 = st.columns(2)
>>> col1.subheader('Columnisation')
st.expander('Expander')
>>> with st.expander('Expand'):
>>>     st.write('Juicy deets')
```

Batch widgets together in a form:

```
>>> with st.form(key='my_form'):
>>>     text_input = st.text_input(label='Enter some text')
>>>     submit_button = st.form_submit_button(label='Submit')
```

Display code

```
st.echo()
>>> with st.echo():
>>>     st.write('Code will be executed and printed')
```

Display progress and status

```
st.progress(progress_variable_1_to_100)
st.spinner()
>>> with st.spinner(text='In progress'):
>>>     time.sleep(5)
>>>     st.success('Done')
st.balloons()
st.error('Error message')
st.warning('Warning message')
st.info('Info message')
st.success('Success message')
st.exception(e)
```

Placeholders, help, and options

```
st.empty()
>>> my_placeholder = st.empty()
>>> my_placeholder.text('Replaced!')
st.help(pandas.DataFrame)
st.get_option(key)
st.set_option(key, value)
st.set_page_config(layout='wide')
```

Mutate data

```
DeltaGenerator.add_rows(data)
>>> my_table = st.table(df1)
>>> my_table.add_rows(df2)
>>> my_chart = st.line_chart(df1)
>>> my_chart.add_rows(df2)
```

Optimize performance

```
@st.cache
>>> @st.cache
... def fetch_and_clean_data(url):
...     # Mutate bar
...     return data
>>> # Executes d1 as first time
>>> d1 = foo(ref1)
>>> # Does not execute d1; returns cached value, d1==d2
>>> d2 = foo(ref1)
>>> # Different arg, so function d1 executes
>>> d3 = foo(ref2)
```

Other key parts of the API

State API

Theme option reference

Components API reference

API cheat sheet