Cheat Sheet

This is a summary of the docs, as of Streamlit v1.27.0.

Install & Import

```
streamlit run first_app.py

# Import convention
>>> import streamlit as st
```

Command line

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

Pre-release features

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

Learn more about experimental features

Magic commands

```
# Magic commands implicitly
# call st.write().
'_This_ is some **Markdown***'
my_variable
'dataframe:', my_data_frame
```

Control flow

```
# Stop execution immediately:
st.stop()
# Rerun script immediately:
st.rerun()

# Group multiple widgets:
>>> with st.form(key='my_form')
>>>    username = st.text_input
>>>    password = st.text_input
>>>    st.form_submit_button('Lo
```

Connect to data sources

```
st.experimental_connection('pe
conn = st.experimental_connect
conn = st.experimental_connect

>>> class MyConnection(Experim
>>>    def _connect(self, **kwa
>>>        return myconn.connec
>>>    def query(self, query):
>>>        return self._instanc
```

Display text

```
st.text('Fixed width text')
st.markdown('_Markdown_') # se
st.latex(r''' e^{i\pi} + 1 = 0
st.write('Most objects') # df,
st.write(['st', 'is <', 3]) #
st.title('My title')
st.header('My header')
st.subheader('My sub')
st.code('for i in range(8): fo
* optional kwarg unsafe_allow_
```

Display interactive widgets

```
st.button("Click me")
st.download_button("Download f
st.link_button("Go to gallery"
st.data_editor("Edit data", da
st.checkbox("I agree")
st.toggle("Enable")
st.radio("Pick one", ["cats", "
st.selectbox("Pick one", ["cats
st.multiselect("Buy", ["milk",
st.slider("Pick a number", 0, 1
st.select_slider("Pick a size"
st.text_input("First name")
st.number_input("Pick a number"
st.text_area("Text to translate
st.date_input("Your birthday")
st.time_input("Meeting time")
st.file_uploader("Upload a CSV"
st.camera_input("Take a picture
st.color_picker("Pick a color"

# Use widgets' returned values
>>> for i in range(int(st.numbe
>>>    foo()
>>> if st.sidebar.selectbox('I
>>>    b()
>>> my_slider_val = st.slider('
>>> st.write(slider_val)

# Disable widgets to remove int
>>> st.slider('Pick a number',
```

Optimize performance

Cache data objects

```
# E.g. Dataframe computation, s
>>> @st.cache_data
... def foo(bar):
...    # Do something expensive
...    return data
# Executes foo
>>> d1 = foo(ref1)
# Does not execute foo
# Returns cached item by value,
>>> d2 = foo(ref1)
# Different arg, so function fo
>>> d3 = foo(ref2)
# Clear all cached entries for
>>> foo.clear()
# Clear values from *all* in-me
>>> st.cache_data.clear()
```

Cache global resources

```
# E.g. TensorFlow session, data
>>> @st.cache_resource
... def foo(bar):
...    # Create and return a nor
...    return session
# Executes foo
>>> s1 = foo(ref1)
# Does not execute foo
# Returns cached item by refere
>>> s2 = foo(ref1)
# Different arg, so function fo
>>> s3 = foo(ref2)
# Clear all cached entries for
>>> foo.clear()
# Clear all global resources fr
>>> st.cache_resource.clear()
```

Deprecated caching

Display data

```
st.dataframe(my_dataframe)
st.table(data.iloc[0:10])
st.json({'foo':'bar','fu':'ba'
st.metric('My metric', 42, 2)
```

Display media

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

Display charts

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

st.altair_chart(chart)
st.bokeh_chart(fig)
st.graphviz_chart(fig)
st.plotly_chart(fig)
st.pydeck_chart(chart)
st.pyplot(fig)
st.vega_lite_chart(df)
```

Add widgets to sidebar

```
# Just add it after st.sidebar.
>>> a = st.sidebar.radio('Selec
```

Build chat-based apps

```
# Insert a chat message contain
>>> with st.chat_message("user"
>>>    st.write("Hello 👋")
>>>    st.line_chart(np.random.
```

```
>>> a = st.sidebar.radio('Sele

# Or use "with" notation:
>>> with st.sidebar:
>>>     st.radio('Select one:', |
```

Columns

```
# Two equal columns:
>>> col1, col2 = st.columns(2)
>>> col1.write("This is column
>>> col2.write("This is column

# Three different columns:
>>> col1, col2, col3 = st.colur
# col1 is larger.

# You can also use "with" notat
>>> with col1:
>>>     st.radio('Select one:', |
```

Tabs

```
# Insert containers separated
>>> tab1, tab2 = st.tabs(["Tab
>>> tab1.write("this is tab 1")
>>> tab2.write("this is tab 2")

# You can also use "with" notat
>>> with tab1:
>>>     st.radio('Select one:', |
```

Personalize apps for users

```
# Show different content based
>>> if st.user.email == 'jane@
>>>     display_jane_content()
>>> elif st.user.email == 'adar
>>>     display_adam_content()
>>> else:
>>>     st.write("Please contac
```

```
# Display a chat input widget.
>>> st.chat_input("Say somethir
```

Learn how to build chat-based apps

Mutate data

```
# Add rows to a dataframe after
# showing it.
>>> element = st.dataframe(df1)
>>> element.add_rows(df2)

# Add rows to a chart after
# showing it.
>>> element = st.line_chart(df1
>>> element.add_rows(df2)
```

Display code

```
>>> with st.echo():
>>>     st.write('Code will be ex
```

Placeholders, help, and options

```
# Replace any single element.
>>> element = st.empty()
>>> element.line_chart(...)
>>> element.text_input(...)  #

# Insert out of order.
>>> elements = st.container()
>>> elements.line_chart(...)
>>> st.write("Hello")
>>> elements.text_input(...)  #

st.help(pandas.DataFrame)
st.get_option(key)
st.set_option(key, value)
st.set_page_config(layout='wide
st.experimental_get_query_param
st.experimental_set_query_param
```

```
>>> @st.cache
... def foo(bar):
...     # Do something expensive
...     return data
>>> # Executes foo
>>> d1 = foo(ref1)
>>> # Does not execute foo
>>> # Returns cached item by re
>>> d2 = foo(ref1)
>>> # Different arg, so functio
>>> d3 = foo(ref2)
```

Display progress and status

```
# Show a spinner during a proce
>>> with st.spinner(text='In pr
>>>     time.sleep(3)
>>>     st.success('Done')

# Show and update progress bar
>>> bar = st.progress(50)
>>> time.sleep(3)
>>> bar.progress(100)

>>> with st.status('Authenticat
>>>     time.sleep(2)
>>>     st.write('Some long respo
>>>     s.update(label='Response

st.balloons()
st.snow()
st.toast('Warming up...')
st.error('Error message')
st.warning('Warning message')
st.info('Info message')
st.success('Success message')
st.exception(e)
```

← **Previous:** Changelog

Next: Streamlit Community Cloud →

Still have questions?

Our forums are full of helpful information and Streamlit experts.