Documentation

search

Search

        - Execution flow
          add
        - Caching and state
          add
        - Connections and secrets
          add
        - Custom components
          add
        - Utilities
          add
        - Configuration
          add
        - TOOLS

        - App testing
          add
        - Command line
          add
    - Tutorials
      add
    - Quick reference
      add
- *web_asset*

  Deploy

    - Concepts
      add
    - Streamlit Community Cloud
      add
    - Snowflake
    - Other platforms
      add
- *school*

  Knowledge base

    - FAQ
    - Installing dependencies
    - Deployment issues

- Home/
- Develop/
- API reference/
- Data elements/
- st.table

star

Tip

Static tables with `st.table` are the most basic way to display dataframes. For the majority of cases, we recommend using `st.dataframe` to display interactive dataframes, and `st.data_editor` to let users edit dataframes.

st.table

Streamlit Version Version 1.41.0

Display a static table.

This differs from `st.dataframe` in that the table in this case is static: its entire contents are laid out directly on the page.

Function signature[source]

st.table(data=None)

Parameters

data (Anything supported by st.dataframe) The table data.

Example

```
import streamlit as st
import pandas as pd
import numpy as np

df = pd.DataFrame(
    np.random.randn(10, 5), columns=("col %d" % i for i in range(5))
)

st.table(df)
```

Built with Streamlit 🎈
Fullscreen *open_in_new*

element.add_rows

Streamlit Version Version 1.41.0

Concatenate a dataframe to the bottom of the current one.

Function signature[source]

element.add_rows(data=None, **kwargs)

Parameters	
data (pandas.DataFrame, pandas.Styler, pyarrow.Table, numpy.ndarray, pyspark.sql.DataFrame, snowflake.snowpark.dataframe.DataFrame, Iterable, dict, or None)	Table to concat. Optional.
**kwargs (pandas.DataFrame, numpy.ndarray, Iterable, dict, or None)	The named dataset to concat. Optional. You can only pass in 1 dataset (including the one in the data parameter).

Example

```
import streamlit as st
import pandas as pd
import numpy as np

df1 = pd.DataFrame(
    np.random.randn(50, 20), columns=("col %d" % i for i in range(20))
)

my_table = st.table(df1)

df2 = pd.DataFrame(
    np.random.randn(50, 20), columns=("col %d" % i for i in range(20))
)

my_table.add_rows(df2)
# Now the table shown in the Streamlit app contains the data for
# df1 followed by the data for df2.
```

You can do the same thing with plots. For example, if you want to add more data to a line chart:

```
# Assuming df1 and df2 from the example above still exist...
my_chart = st.line_chart(df1)
my_chart.add_rows(df2)
# Now the chart shown in the Streamlit app contains the data for
# df1 followed by the data for df2.
```

And for plots whose datasets are named, you can pass the data with a keyword argument where the key is the name:

```
my_chart = st.vega_lite_chart(
    {
        "mark": "line",
        "encoding": {"x": "a", "y": "b"},
        "datasets": {
            "some_fancy_name": df1,  # <-- named dataset
        },
        "data": {"name": "some_fancy_name"},
    }
)
my_chart.add_rows(some_fancy_name=df2)  # <-- name used as keyword
```

Previous: st.column_configNext: st.metric

forum

Still have questions?

Our forums are full of helpful information and Streamlit experts.

---

forum Ask AI