Documentation

search

Search

- *rocket_launch*

  Get started

  - Installation
    add
  - Fundamentals
    add
  - First steps
    add
- *code*

  Develop

  - Concepts
    add
  - API reference
    remove
    - PAGE ELEMENTS

    - Write and magic
      remove
      - st.write
      - st.write_stream
      - magic
    - Text elements
      add
    - Data elements
      add
    - Chart elements
      add
    - Input widgets
      add
    - Media elements
      add
    - Layouts and containers
      add
    - Chat elements
      add
    - Status elements
      add
    - Third-party components*open_in_new*
    - APPLICATION LOGIC

    - Navigation and pages
      add
    - Execution flow
      add
    - Caching and state
      add

st.write

Streamlit Version Version 1.41.0

Write arguments to the app.

This is the Swiss Army knife of Streamlit commands: it does different things depending on what you throw at it. Unlike other Streamlit commands, write() has some unique properties:

1. You can pass in multiple arguments, all of which will be written.
2. Its behavior depends on the input types as follows.
3. It returns None, so its "slot" in the App cannot be reused.

Function signature[source]

st.write(*args, unsafe_allow_html=False, **kwargs)

Parameters

*args (any)

One or many objects to print to the App.

Arguments are handled as follows:

- write(string) : Prints the formatted Markdown string, with support for LaTeX expression, emoji shortcodes, and colored text. See docs for st.markdown for more.

- write(dataframe) : Displays any dataframe-like object in an interactive table.
- write(dict) : Displays dict-like in an interactive viewer.
- write(list) : Displays list-like in an interactive viewer.
- write(error) : Prints an exception specially.
- write(func) : Displays information about a function.
- write(module) : Displays information about a module.
- write(class) : Displays information about a class.
- write(DeltaGenerator) : Displays information about a DeltaGenerator.
- write(mpl_fig) : Displays a Matplotlib figure.
- write(generator) : Streams the output of a generator.
- write(openai.Stream) : Streams the output of an OpenAI stream.
- write(altair) : Displays an Altair chart.
- write(PIL.Image) : Displays an image.
- write(keras) : Displays a Keras model.
- write(graphviz) : Displays a Graphviz graph.
- write(plotly_fig) : Displays a Plotly figure.
- write(bokeh_fig) : Displays a Bokeh figure.
- write(sympy_expr) : Prints SymPy expression using LaTeX.
- write(htmlable) : Prints _repr_html_() for the object if available.
- write(db_cursor) : Displays DB API 2.0 cursor results in a table.
- write(obj) : Prints str(obj) if otherwise unknown.

unsafe_allow_html (bool)

Whether to render HTML within `*args`. This only applies to strings or objects falling back on `_repr_html_()`. If this is `False` (default), any HTML tags found in `body` will be escaped and therefore treated as raw text. If this is `True`, any HTML expressions within `body` will be rendered.

Adding custom HTML to your app impacts safety, styling, and maintainability.

Note

If you only want to insert HTML or CSS without Markdown text, we recommend using `st.html` instead.

**kwargs (any)

delete

`**kwargs` is deprecated and will be removed in a later version. Use other, more specific Streamlit commands to pass additional keyword arguments.

Keyword arguments. Not used.

Example

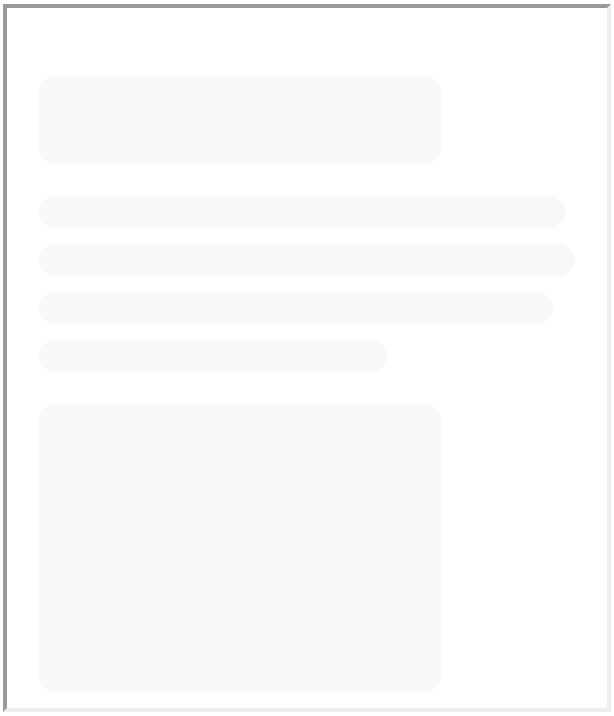
Its basic use case is to draw Markdown-formatted text, whenever the input is a string:

```
import streamlit as st

st.write("Hello, *World!* :sunglasses:")
```

Built with Streamlit 🎈
Fullscreen *open_in_new*

As mentioned earlier, `st.write()` also accepts other data formats, such as numbers, data frames, styled data frames, and assorted objects:

```
import streamlit as st
import pandas as pd

st.write(1234)
st.write(
    pd.DataFrame(
        {
            "first column": [1, 2, 3, 4],
            "second column": [10, 20, 30, 40],
        }
    )
)
```



Built with Streamlit 🎈
Fullscreen *open_in_new*

Finally, you can pass in multiple arguments to do things like:

```
import streamlit as st
```

```
st.write("1 + 1 = ", 2)
st.write("Below is a DataFrame:", data_frame, "Above is a dataframe.")
```

Built with Streamlit 🎈
Fullscreen *open_in_new*

Oh, one more thing: `st.write` accepts chart objects too! For example:

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

df = pd.DataFrame(np.random.randn(200, 3), columns=["a", "b", "c"])
c = (
    alt.Chart(df)
    .mark_circle()
    .encode(x="a", y="b", size="c", color="c", tooltip=["a", "b", "c"])
)

st.write(c)
```

Built with Streamlit 🎈
Fullscreen *open_in_new*

Featured video

🔗

Learn what the `st.write` and magic commands are and how to use them.

←Previous: Write and magicNext: st.write_stream→
forum

Still have questions?

Our forums are full of helpful information and Streamlit experts.

---

forum Ask AI