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
      add
    - Text elements
      add
    - Data elements
      add
    - Chart elements
      add
    - Input widgets
      add
    - Media elements
      add
    - Layouts and containers
      remove
      - st.columns
      - st.container
      - st.dialog*link*
      - st.empty
      - st.expander
      - st.form*link*
      - st.popover
      - st.sidebar
      - st.tabs
    - 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
        - 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/
- Layouts and containers/
- st.expander

st.expander

Streamlit Version Version 1.41.0 ▾

Insert a multi-element container that can be expanded/collapsed.

Inserts a container into your app that can be used to hold multiple elements and can be expanded or collapsed by the user. When collapsed, all that is visible is the provided label.

To add elements to the returned container, you can use the `with` notation (preferred) or just call methods directly on the returned object. See examples below.

Warning

Currently, you may not put expanders inside another expander.

Function signature[source]

st.expander(label, expanded=False, *, icon=None)

Parameters

label (str)

A string to use as the header for the expander. The label can optionally contain GitHub-flavored Markdown of the following types: Bold, Italics, Strikethroughs, Inline Code, Links, and Images. Images display like icons, with a max height equal to the font height.

Unsupported Markdown elements are unwrapped so only their children (text contents) render. Display unsupported elements as literal characters by backslash-escaping them. E.g., `"1\. Not an ordered list"`.

See the `body` parameter of `st.markdown` for additional, supported Markdown directives.

expanded (bool)

If True, initializes the expander in "expanded" state. Defaults to False (collapsed).

icon (str, None)

An optional emoji or icon to display next to the expander label. If `icon` is `None` (default), no icon is displayed. If `icon` is a string, the following options are valid:

- A single-character emoji. For example, you can set `icon="🚨"` or `icon="🔥"`. Emoji short codes are not supported.
- An icon from the Material Symbols library (rounded style) in the format `":material/icon_name:"` where "icon_name" is the name of the icon in snake case.

  For example, `icon=":material/thumb_up:"` will display the Thumb Up icon. Find additional icons in the Material Symbols font library.

Examples

You can use the `with` notation to insert any element into an expander

```
import streamlit as st

st.bar_chart({"data": [1, 5, 2, 6, 2, 1]})

with st.expander("See explanation"):
    st.write('''
        The chart above shows some numbers I picked for you.
        I rolled actual dice for these, so they're *guaranteed* to
        be random.
    ''')
    st.image("https://static.streamlit.io/examples/dice.jpg")
```

Built with Streamlit 🎈
Fullscreen *open_in_new*

Or you can just call methods directly on the returned objects:

```
import streamlit as st

st.bar_chart({"data": [1, 5, 2, 6, 2, 1]})

expander = st.expander("See explanation")
expander.write('''
    The chart above shows some numbers I picked for you.
    I rolled actual dice for these, so they're *guaranteed* to
    be random.
''')
expander.image("https://static.streamlit.io/examples/dice.jpg")
```

Built with Streamlit 🎈

Fullscreen *open in new*

←Previous: st.emptyNext: st.form→

forum

Still have questions?

Our forums are full of helpful information and Streamlit experts.

---

forum Ask AI