

[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](#)
remove
 - BUTTONS

 - [st.button](#)
 - [st.download button](#)
 - [st.form submit button](#)[link](#)
 - [st.link button](#)
 - [st.page link](#)
 - SELECTIONS

 - [st.checkbox](#)
 - [st.color picker](#)
 - [st.feedback](#)
 - [st.multiselect](#)
 - [st.pills](#)
 - [st.radio](#)
 - [st.segmented control](#)
 - [st.selectbox](#)
 - [st.select slider](#)
 - [st.toggle](#)
 - NUMERIC

- [st.number_input](#)
 - [st.slider](#)
 - DATE & TIME
-

- [st.date_input](#)
 - [st.time_input](#)
 - TEXT
-

- [st.chat_input](#)*link*
 - [st.text_area](#)
 - [st.text_input](#)
 - MEDIA & FILES
-

- [st.audio_input](#)
- [st.camera_input](#)
- [st.data_editor](#)*link*
- [st.file_uploader](#)

- [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
 - [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/](#)
- [Input widgets/](#)
- [st.button](#)

st.button



Streamlit Version ▼

Display a button widget.

Function signature[\[source\]](#)

st.button(label, key=None, help=None, on_click=None, args=None, kwargs=None, *, type="secondary", icon=None, disabled=False, use_container_width=False)

Parameters

	A short label explaining to the user what this button is for. 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.
label (str)	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.
key (str or int)	An optional string or integer to use as the unique key for the widget. If this is omitted, a key will be generated for the widget based on its content. No two widgets may have the same key.
help (str)	An optional tooltip that gets displayed when the button is hovered over.
on_click (callable)	An optional callback invoked when this button is clicked.
Returns	
(bool)	True if the button was clicked on the last run of the app, False otherwise.

```
st.button(label, key=None, help=None, on_click=None, args=None, kwargs=None, *, type="secondary", icon=None, disabled=False, use_container_width=False)
```

args (tuple) An optional tuple of args to pass to the callback.

kwargs (dict) An optional dict of kwargs to pass to the callback.

An optional string that specifies the button type. This can be one of the following:

type ("primary", "secondary", or "tertiary")

- "primary": The button's background is the app's primary color for additional emphasis.
- "secondary" (default): The button's background coordinates with the app's background color for normal emphasis.
- "tertiary": The button is plain text without a border or background for subtly.

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

icon (str or None)

- 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.

disabled (bool) An optional boolean that disables the button if set to `True`. The default is `False`.

use_container_width (bool) Whether to expand the button's width to fill its parent container. If `use_container_width` is `False` (default), Streamlit sizes the button to fit its contents. If `use_container_width` is `True`, the width of the button matches its parent container.

In both cases, if the contents of the button are wider than the parent container, the contents will line wrap.

Returns

(bool) True if the button was clicked on the last run of the app, False otherwise.

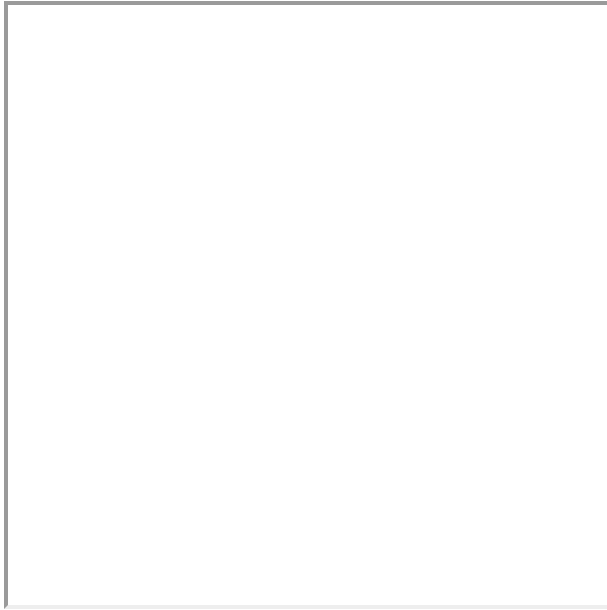
Examples


Example 1: Customize your button type

```
import streamlit as st

st.button("Reset", type="primary")
if st.button("Say hello"):
    st.write("Why hello there")
else:
    st.write("Goodbye")
```

```
if st.button("Aloha", type="tertiary"):
    st.write("Ciao")
```



[Built with Streamlit](#) 
[Fullscreen open in new](#)


Example 2: Add icons to your button

Although you can add icons to your buttons through Markdown, the `icon` parameter is a convenient and consistent alternative.

```
import streamlit as st

left, middle, right = st.columns(3)
if left.button("Plain button", use_container_width=True):
    left.markdown("You clicked the plain button.")
if middle.button("Emoji button", icon="😄", use_container_width=True):
    middle.markdown("You clicked the emoji button.")
if right.button("Material button", icon=":material/mood:", use_container_width=True):
    right.markdown("You clicked the Material button.")
```



[Built with Streamlit](#) 
[Fullscreen open in new](#)

Advanced functionality



Although a button is the simplest of input widgets, it's very common for buttons to be deeply tied to the use of `st.session_state`. Check out our advanced guide on [Button behavior and examples](#).

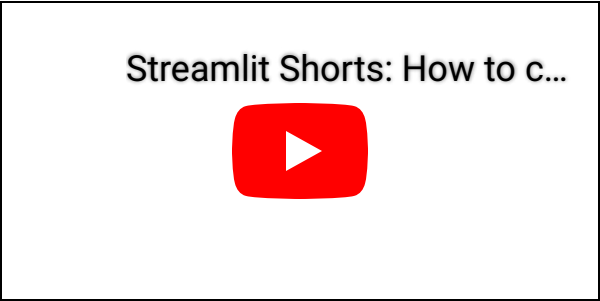
Featured videos



Check out our video on how to use one of Streamlit's core functions, the button!



In the video below, we'll take it a step further and learn how to combine a [button](#), [checkbox](#) and [radio button](#)!



[←Previous: Input widgets](#)[Next: st.download_button→](#)
forum

Still have questions?

Our [forums](#) are full of helpful information and Streamlit experts.

[Home](#)[Contact Us](#)[Community](#)



© 2025 Snowflake Inc. [Cookie policy](#)

[forum](#) [Ask AI](#)