

[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](#)  
*add*
  - [Chat elements](#)  
*add*
  - [Status elements](#)  
*add*
  - [Third-party components](#)*open in new*
  - APPLICATION LOGIC

---
  - [Navigation and pages](#)  
*add*
  - [Execution flow](#)  
*remove*
    - [st.dialog](#)
    - [st.form](#)
    - [st.form submit button](#)
    - [st.fragment](#)
    - [st.rerun](#)

- [st.stop](#)
- [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/](#)
- [Execution flow/](#)
- [st.form](#)

*star*

## Tip

This page only contains information on the `st.form` API. For a deeper dive into creating and using forms within Streamlit apps, read our guide on [Using forms](#).

## st.form



Create a form that batches elements together with a "Submit" button.

A form is a container that visually groups other elements and widgets together, and contains a Submit button. When the form's Submit button is pressed, all widget values inside the form will be sent to Streamlit in a batch.

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

Forms have a few constraints:

- Every form must contain a `st.form_submit_button`.
- `st.button` and `st.download_button` cannot be added to a form.
- Forms can appear anywhere in your app (sidebar, columns, etc), but they cannot be embedded inside other forms.
- Within a form, the only widget that can have a callback function is `st.form_submit_button`.

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

**`st.form(key, clear_on_submit=False, *, enter_to_submit=True, border=True)`**

Parameters

key (str)	A string that identifies the form. Each form must have its own key. (This key is not displayed to the user in the interface.)
clear_on_submit (bool)	If True, all widgets inside the form will be reset to their default values after the user presses the Submit button. Defaults to False. (Note that Custom Components are unaffected by this flag, and will not be reset to their defaults on form submission.)
enter_to_submit (bool)	<p>Whether to submit the form when a user presses Enter while interacting with a widget inside the form.</p> <p>If this is True (default), pressing Enter while interacting with a form widget is equivalent to clicking the first <code>st.form_submit_button</code> in the form.</p> <p>If this is False, the user must click an <code>st.form_submit_button</code> to submit the form.</p> <p>If the first <code>st.form_submit_button</code> in the form is disabled, the form will override submission behavior with <code>enter_to_submit=False</code>.</p>
border (bool)	<p>Whether to show a border around the form. Defaults to True.</p> <p>Note</p> <p>Not showing a border can be confusing to viewers since interacting with a widget in the form will do nothing. You should only remove the border if there's another border (e.g. because of an expander) or the form is small (e.g. just a text input and a submit button).</p>

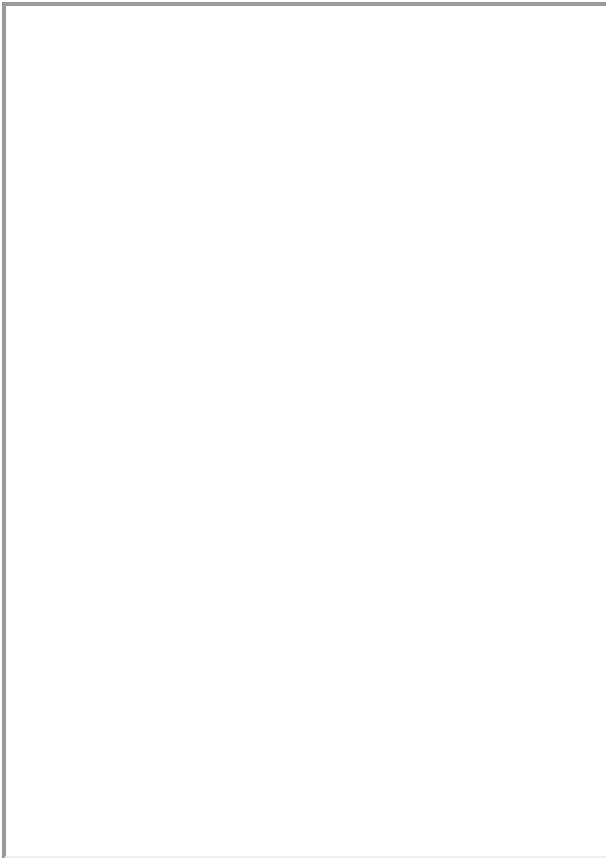
Examples

Inserting elements using `with` notation:

```
import streamlit as st

with st.form("my_form"):
    st.write("Inside the form")
    slider_val = st.slider("Form slider")
    checkbox_val = st.checkbox("Form checkbox")
```

```
# Every form must have a submit button.
submitted = st.form_submit_button("Submit")
if submitted:
    st.write("slider", slider_val, "checkbox", checkbox_val)
st.write("Outside the form")
```



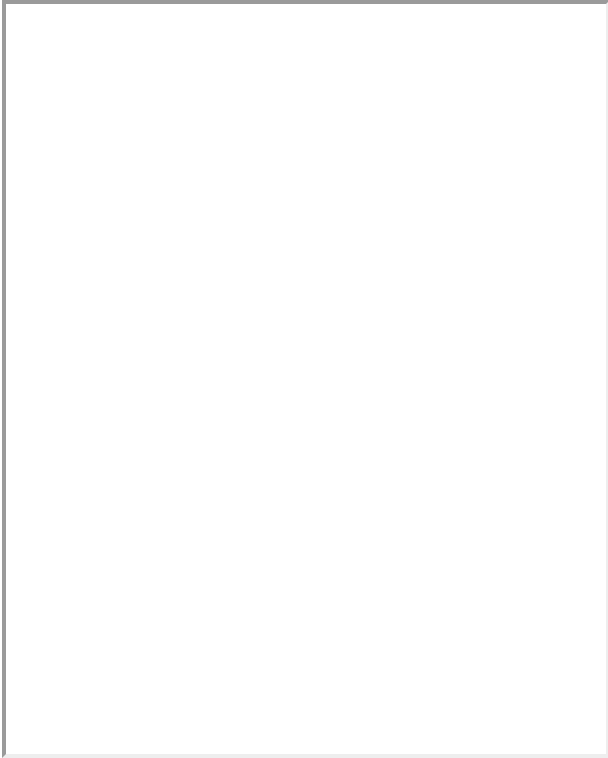
[Built with Streamlit !\[\]\(2bdfe261b986065ee0ac76460d6528c9\_img.jpg\)](#)  
[Fullscreen open in new](#)

Inserting elements out of order:

```
import streamlit as st

form = st.form("my_form")
form.slider("Inside the form")
st.slider("Outside the form")

# Now add a submit button to the form:
form.form_submit_button("Submit")
```



[Built with Streamlit](#) 

[Fullscreen open in new](#)

[←Previous: st.dialog](#)[Next: st.form\\_submit\\_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](#)