

[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.fragment](#)

st.fragment



Streamlit Version ▼

Decorator to turn a function into a fragment which can rerun independently of the full app.

When a user interacts with an input widget created inside a fragment, Streamlit only reruns the fragment instead of the full app. If `run_every` is set, Streamlit will also rerun the fragment at the specified interval while the session is active, even if the user is not interacting with your app.

To trigger an app rerun from inside a fragment, call `st.rerun()` directly. To trigger a fragment rerun from within itself, call `st.rerun(scope="fragment")`. Any values from the fragment that need to be accessed from the wider app should generally be stored in Session State.

When Streamlit element commands are called directly in a fragment, the elements are cleared and redrawn on each fragment rerun, just like all elements are redrawn on each app rerun. The rest of the app is persisted during a fragment rerun. When a fragment renders elements into externally created containers, the elements will not be cleared with each fragment rerun. Instead, elements will accumulate in those containers with each fragment rerun, until the next app rerun.

Calling `st.sidebar` in a fragment is not supported. To write elements to the sidebar with a fragment, call your fragment function inside a `with st.sidebar` context manager.

Fragment code can interact with Session State, imported modules, and other Streamlit elements created outside the fragment. Note that these interactions are additive across multiple fragment reruns. You are responsible for handling any side effects of that behavior.

Warning

- Fragments can only contain widgets in their main body. Fragments can't render widgets to externally created containers.

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

`st.fragment(func=None, *, run_every=None)`

Parameters

func (callable) The function to turn into a fragment.

The time interval between automatic fragment reruns. This can be one of the following:

- `None` (default).
- An `int` or `float` specifying the interval in seconds.
- A string specifying the time in a format supported by [Pandas' Timedelta constructor](#), e.g. "1d", "1.5 days", or "1h23s".
- A `timedelta` object from [Python's built-in datetime library](#), e.g. `timedelta(days=1)`.

If `run_every` is `None`, the fragment will only rerun from user-triggered events.

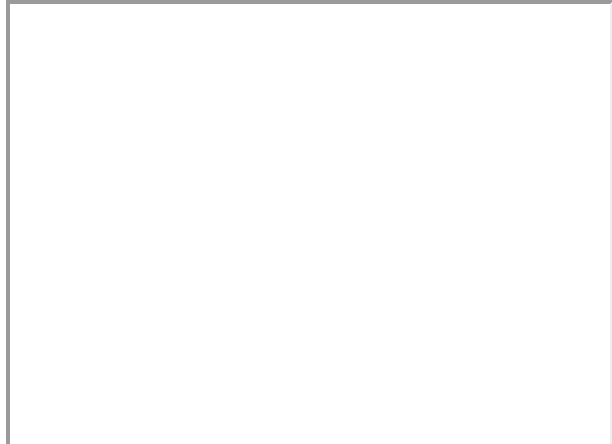
Examples

The following example demonstrates basic usage of `@st.fragment`. As an analogy, "inflating balloons" is a slow process that happens outside of the fragment. "Releasing balloons" is a quick process that happens inside of the fragment.

```
import streamlit as st
import time

@st.fragment
def release_the_balloons():
    st.button("Release the balloons", help="Fragment rerun")
    st.balloons()

with st.spinner("Inflating balloons..."):
    time.sleep(5)
release_the_balloons()
st.button("Inflate more balloons", help="Full rerun")
```



[Built with Streamlit !\[\]\(34c5d6a15de5cee4fef2fa4252527f03_img.jpg\)](#)
[Fullscreen open in new](#)

This next example demonstrates how elements both inside and outside of a fragment update with each app or fragment rerun. In this app, clicking "Rerun full app" will increment both counters and update all values displayed in the app. In contrast, clicking "Rerun fragment" will only increment the counter within the fragment. In this case, the `st.write` command inside the fragment will update the app's frontend, but the two `st.write` commands outside the fragment will not update the frontend.

```
import streamlit as st

if "app_runs" not in st.session_state:
    st.session_state.app_runs = 0
    st.session_state.fragment_runs = 0

@st.fragment
def my_fragment():
    st.session_state.fragment_runs += 1
    st.button("Rerun fragment")
    st.write(f"Fragment says it ran {st.session_state.fragment_runs} times.")

st.session_state.app_runs += 1
my_fragment()
st.button("Rerun full app")
st.write(f"Full app says it ran {st.session_state.app_runs} times.")
st.write(f"Full app sees that fragment ran {st.session_state.fragment_runs} times.")
```



[Built with Streamlit !\[\]\(8af806fb1314382d09bc5ec5b767526c_img.jpg\)](#)
[Fullscreen open in new](#)

You can also trigger an app rerun from inside a fragment by calling `st.rerun`.

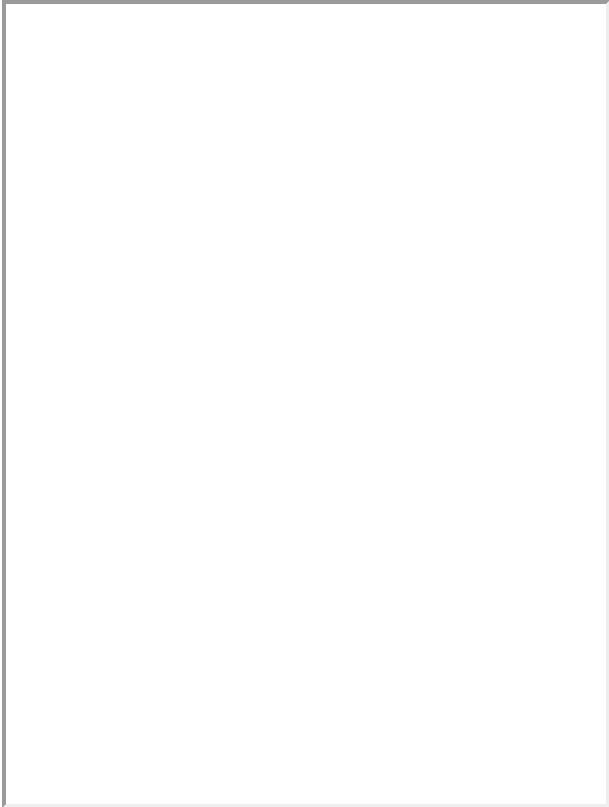
```
import streamlit as st

if "clicks" not in st.session_state:
    st.session_state.clicks = 0

@st.fragment
def count_to_five():
    if st.button("Plus one!"):
        st.session_state.clicks += 1
        if st.session_state.clicks % 5 == 0:
            st.rerun()
    return

count_to_five()
st.header(f"Multiples of five clicks: {st.session_state.clicks // 5}")

if st.button("Check click count"):
    st.toast(f"## Total clicks: {st.session_state.clicks}")
```



[Built with Streamlit](#) 

[Fullscreen open in new](#)

[←Previous: st.form submit button](#)[Next: st.rerun→](#)
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](#)