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.rerun

Streamlit Version Version 1.41.0

Rerun the script immediately.

When `st.rerun()` is called, Streamlit halts the current script run and executes no further statements. Streamlit immediately queues the script to rerun.

When using `st.rerun` in a fragment, you can scope the rerun to the fragment. However, if a fragment is running as part of a full-app rerun, a fragment-scoped rerun is not allowed.

Function signature[source]

st.rerun(*, scope="app")

Parameters	
scope ("app" or "fragment")	Specifies what part of the app should rerun. If `scope` is `"app"` (default), the full app reruns. If `scope` is `"fragment"`, Streamlit only reruns the fragment from which this command is called. Setting `scope="fragment"` is only valid inside a fragment during a fragment rerun. If `st.rerun(scope="fragment")` is called during a full-app rerun or outside of a fragment, Streamlit will raise a `StreamlitAPIException`.

Caveats for `st.rerun`

`st.rerun` is one of the tools to control the logic of your app. While it is great for prototyping, there can be adverse side effects:

- Additional script runs may be inefficient and slower.
- Excessive reruns may complicate your app's logic and be harder to follow.
- If misused, infinite looping may crash your app.

In many cases where `st.rerun` works, callbacks may be a cleaner alternative. Containers may also be helpful.

A simple example in three variations

Using `st.rerun` to update an earlier header

```
import streamlit as st if "value" not in st.session_state: st.session_state.value = "Title" ##### Option using st.rerun ##### st.header(st.session_state.value) if st.button("Foo"): st.session_state.value = "Foo" st.rerun()
```

Using a callback to update an earlier header

```
##### Option using a callback ##### st.header(st.session_state.value) def update_value(): st.session_state.value = "Bar" st.button("Bar", on_click=update_value)
```

Using containers to update an earlier header

```
##### Option using a container ##### container = st.container() if st.button("Baz"): st.session_state.value = "Baz" container.header(st.session_state.value)
```

←Previous: st.fragmentNext: st.stop→

forum

Still have questions?

Our forums are full of helpful information and Streamlit experts.

---

forum Ask AI