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

Execution flow

Change execution

By default, Streamlit apps execute the script entirely, but we allow some functionality to handle control flow in your applications.

Modal dialog

Insert a modal dialog that can rerun independently from the rest of the script.

```
@st.dialog("Sign up") def email_form(): name = st.text_input("Name") email = st.text_input("Email")
```

Fragments

Define a fragment to rerun independently from the rest of the script.

```
@st.fragment(run_every="10s") def fragment(): df = get_data() st.line_chart(df)
```

Rerun script

Rerun the script immediately.

```
st.rerun()
```

Stop execution

Stops execution immediately.

```
st.stop()
```

Group multiple widgets

By default, Streamlit reruns your script everytime a user interacts with your app. However, sometimes it's a better user experience to wait until a group of related widgets is filled before actually rerunning the script. That's what `st.form` is for!

Forms

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

```
with st.form(key='my_form'): name = st.text_input("Name") email = st.text_input("Email")
st.form_submit_button("Sign up")
```

Form submit button

Display a form submit button.

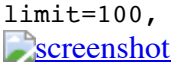
```
with st.form(key='my_form'): name = st.text_input("Name") email = st.text_input("Email")
st.form_submit_button("Sign up")
```

Third-party components

These are featured components created by our lovely community. For more examples and inspiration, check out our Components Gallery and Streamlit Extras!

Autorefresh

Force a refresh without tying up a script.
Created by @kmcgrady.

```
from streamlit_autorefresh import
st_autorefresh
st_autorefresh(interval=2000,
limit=100, key="fizzbuzzcounter")
```


Pydantic

Auto-generate Streamlit UI from Pydantic Models and Dataclasses. Created by @lukasmasuch.

```
import streamlit_pydantic as sp
sp.pydantic_form(key="my_form",
model=ExampleModel)
```


Streamlit Pages

An experimental version of Streamlit Multi-Page Apps. Created by @blackary.

```
from st_pages import Page,
show_pages, add_page_title
show_pages([
Page("streamlit_app.py", "Home",
"🏠"),
Page("other_pages/page2.py", "Page
2", ":books:"), ])
```
←Previous: Navigation and pagesNext: st.dialog→

forum

Still have questions?

Our forums are full of helpful information and Streamlit experts.
