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
      remove
      - st.chat_input
      - st.chat_message
      - st.status*link*
      - st.write_stream*link*
    - 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/
- Chat elements

Chat elements

Streamlit provides a few commands to help you build conversational apps. These chat elements are designed to be used in conjunction with each other, but you can also use them separately.

`st.chat_message` lets you insert a chat message container into the app so you can display messages from the user or the app. Chat containers can contain other Streamlit elements, including charts, tables, text, and more. `st.chat_input` lets you display a chat input widget so the user can type in a message. Remember to check out `st.status` to display output from long-running processes and external API calls.

screenshot

Chat input

Display a chat input widget.

```
prompt = st.chat_input("Say something") if prompt: st.write(f"The user has sent: {prompt}")
```

screenshot

Chat message

Insert a chat message container.

```
import numpy as np with st.chat_message("user"): st.write("Hello 👋")
st.line_chart(np.random.randn(30, 3))
```

screenshot

Status container

Display output of long-running tasks in a container.

```
with st.status('Running'): do_something_slow()
```

st.write_stream

Write generators or streams to the app with a typewriter effect.

```
st.write_stream(my_generator) st.write_stream(my_llm_stream)
```

←Previous: Layouts and containersNext: st.chat_input→

forum

Still have questions?

Our forums are full of helpful information and Streamlit experts.

Cookie policy

forum Ask AI