[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/
- st.chat_message

star

Tip

Read the Build a basic LLM chat app tutorial to learn how to use `st.chat_message` and `st.chat_input` to build chat-based apps.

st.chat_message

Streamlit Version Version 1.41.0

Insert a chat message container.

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

Function signature[source]

st.chat_message(name, *, avatar=None)

Parameters

name ("user", "assistant", "ai", "human", or str)

The name of the message author. Can be "human"/"user" or "ai"/"assistant" to enable preset styling and avatars.

Currently, the name is not shown in the UI but is only set as an accessibility label. For accessibility reasons, you should not use an empty string.

avatar (Anything supported by st.image (except list), str, or None)

The avatar shown next to the message.

If `avatar` is `None` (default), the icon will be determined from `name` as follows:

- If `name` is `"user"` or `"human"`, the message will have a default user icon.
- If `name` is `"ai"` or `"assistant"`, the message will have a default bot icon.
- For all other values of `name`, the message will show the first letter of the name.

In addition to the types supported by `st.image` (except list), the following strings are valid:

- A single-character emoji. For example, you can set `avatar="🧑‍💻"` or `avatar="🦖"`. Emoji short codes are not supported.

- An icon from the Material Symbols library (rounded style) in the format `":material/icon_name:"` where "icon_name" is the name of the icon in snake case.

  For example, `icon=":material/thumb_up:"` will display the Thumb Up icon. Find additional icons in the Material Symbols font library.

Returns

(Container)

A single container that can hold multiple elements.

Examples

You can use `with` notation to insert any element into an expander

```
import streamlit as st
import numpy as np

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

Built with Streamlit 🎈
Fullscreen *open_in_new*

Or you can just call methods directly in the returned objects:

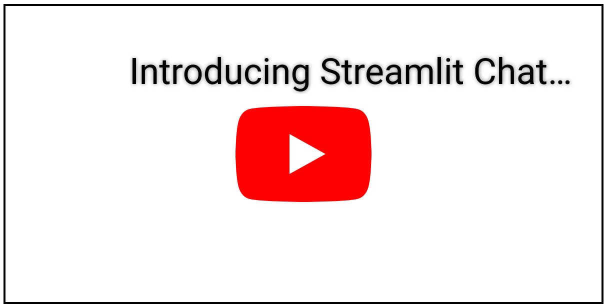
```
import streamlit as st
import numpy as np

message = st.chat_message("assistant")
message.write("Hello human")
message.bar_chart(np.random.randn(30, 3))
```

Built with Streamlit 🎈

Fullscreen *open_in_new*

For an overview of the `st.chat_message` and `st.chat_input` API, check out this video tutorial by Chanin Nantasenamat (@dataprofessor), a Senior Developer Advocate at Streamlit.



←Previous: st.chat_inputNext: st.status→

forum

Still have questions?

Our forums are full of helpful information and Streamlit experts.

Cookie policy

forum Ask AI