

[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](#)

*remove*

- BUTTONS

---

- [st.button](#)

- [st.download button](#)

- [st.form submit button](#)[link](#)

- [st.link button](#)

- [st.page link](#)

- SELECTIONS

---

- [st.checkbox](#)

- [st.color picker](#)

- [st.feedback](#)

- [st.multiselect](#)

- [st.pills](#)

- [st.radio](#)

- [st.segmented control](#)

- [st.selectbox](#)

- [st.select slider](#)

- [st.toggle](#)

- NUMERIC

- [st.number\\_input](#)
  - [st.slider](#)
  - DATE & TIME
- 

- [st.date\\_input](#)
  - [st.time\\_input](#)
  - TEXT
- 

- [st.chat\\_input](#)*link*
  - [st.text\\_area](#)
  - [st.text\\_input](#)
  - MEDIA & FILES
- 

- [st.audio\\_input](#)
- [st.camera\\_input](#)
- [st.data\\_editor](#)*link*
- [st.file\\_uploader](#)

- [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](#)  
*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/](#)
- [Input widgets/](#)
- [st.download\\_button](#)

## st.download\_button



Streamlit Version  ▼

Display a download button widget.

This is useful when you would like to provide a way for your users to download a file directly from your app.

Note that the data to be downloaded is stored in-memory while the user is connected, so it's a good idea to keep file sizes under a couple hundred megabytes to conserve memory.

If you want to prevent your app from rerunning when a user clicks the download button, wrap the download button in a [fragment](#).

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

**st.download\_button(label, data, file\_name=None, mime=None, key=None, help=None, on\_click=None, args=None, kwargs=None, \*, type="secondary", icon=None, disabled=False, use\_container\_width=False)**

### Parameters

A short label explaining to the user what this button is for. The label can optionally contain GitHub-flavored Markdown of the following types: Bold, Italics, Strikethroughs, Inline Code, Links, and Images. Images display like icons, with a max height equal to the font height.

label (str)      Unsupported Markdown elements are unwrapped so only their children (text contents) render. Display unsupported elements as literal characters by backslash-escaping them. E.g., "1\ . Not an ordered list".

See the body parameter of [st.markdown](#) for additional, supported Markdown directives.

### Returns

(bool)      True if the button was clicked on the last run of the app, False otherwise.

**st.download\_button(label, data, file\_name=None, mime=None, key=None, help=None, on\_click=None, args=None, kwargs=None, \*, type="secondary", icon=None, disabled=False, use\_container\_width=False)**

data (str or bytes or file)	The contents of the file to be downloaded. See example below for caching techniques to avoid recomputing this data unnecessarily.
file_name (str)	An optional string to use as the name of the file to be downloaded, such as 'my_file.csv'. If not specified, the name will be automatically generated.
mime (str or None)	The MIME type of the data. If None, defaults to "text/plain" (if data is of type <i>str</i> or is a textual <i>file</i> ) or "application/octet-stream" (if data is of type <i>bytes</i> or is a binary <i>file</i> ).
key (str or int)	An optional string or integer to use as the unique key for the widget. If this is omitted, a key will be generated for the widget based on its content. No two widgets may have the same key.
help (str)	An optional tooltip that gets displayed when the button is hovered over.
on_click (callable)	An optional callback invoked when this button is clicked.
args (tuple)	An optional tuple of args to pass to the callback.
kwargs (dict)	An optional dict of kwargs to pass to the callback.
type ("primary", "secondary", or "tertiary")	<p>An optional string that specifies the button type. This can be one of the following:</p> <ul style="list-style-type: none"> <li>"primary": The button's background is the app's primary color for additional emphasis.</li> <li>"secondary" (default): The button's background coordinates with the app's background color for normal emphasis.</li> <li>"tertiary": The button is plain text without a border or background for subtly.</li> </ul>
icon (str or None)	<p>An optional emoji or icon to display next to the button label. If <code>icon</code> is None (default), no icon is displayed. If <code>icon</code> is a string, the following options are valid:</p> <ul style="list-style-type: none"> <li>A single-character emoji. For example, you can set <code>icon="🍌"</code> or <code>icon="🔥"</code>. Emoji short codes are not supported.</li> <li>An icon from the Material Symbols library (rounded style) in the format <code>"material/icon_name:"</code> where "icon_name" is the name of the icon in snake case.</li> </ul> <p>For example, <code>icon="material/thumb_up:"</code> will display the Thumb Up icon. Find additional icons in the <a href="#">Material Symbols</a> font library.</p>
Returns	
(bool)	True if the button was clicked on the last run of the app, False otherwise.

**st.download\_button(label, data, file\_name=None, mime=None, key=None, help=None, on\_click=None, args=None, kwargs=None, \*, type="secondary", icon=None, disabled=False, use\_container\_width=False)**

**disabled** (bool) An optional boolean that disables the download button if set to `True`. The default is `False`.

**use\_container\_width** (bool) Whether to expand the button's width to fill its parent container. If `use_container_width` is `False` (default), Streamlit sizes the button to fit its contents. If `use_container_width` is `True`, the width of the button matches its parent container.

In both cases, if the contents of the button are wider than the parent container, the contents will line wrap.

**Returns**

(bool) `True` if the button was clicked on the last run of the app, `False` otherwise.

## Examples

Download a large DataFrame as a CSV:

```
import streamlit as st

@st.cache_data
def convert_df(df):
    # IMPORTANT: Cache the conversion to prevent computation on every rerun
    return df.to_csv().encode("utf-8")

csv = convert_df(my_large_df)

st.download_button(
    label="Download data as CSV",
    data=csv,
    file_name="large_df.csv",
    mime="text/csv",
)
```

Download a string as a file:

```
import streamlit as st

text_contents = '''This is some text'''
st.download_button("Download some text", text_contents)
```

Download a binary file:

```
import streamlit as st

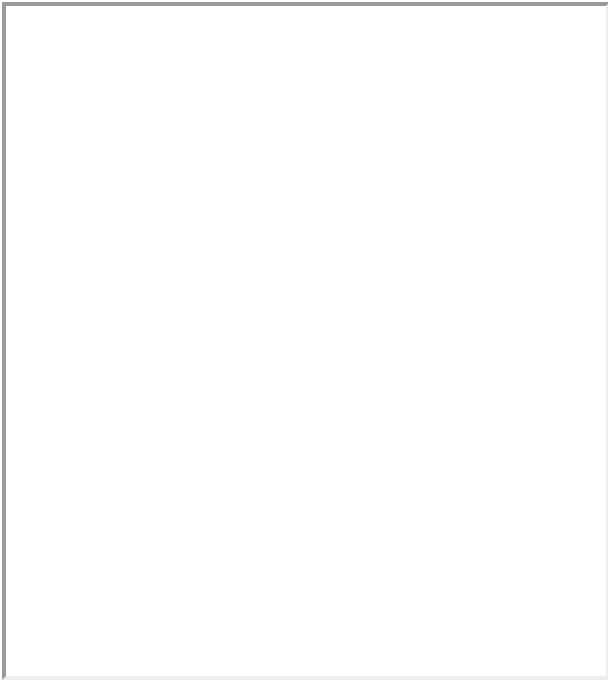
binary_contents = b"example content"
# Defaults to "application/octet-stream"
st.download_button("Download binary file", binary_contents)
```

Download an image:

```
import streamlit as st

with open("flower.png", "rb") as file:
    btn = st.download_button(
        label="Download image",
```

```
data=file,
file_name="flower.png",
mime="image/png",
)
```



[Built with Streamlit](#) 

[Fullscreen open in new](#)

[←Previous: st.buttonNext: st.form submit button→](#)  
*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