

[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.text_input](#)

st.text_input



Streamlit Version ▼

Display a single-line text input widget.

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

```
st.text_input(label, value="", max_chars=None, key=None, type="default", help=None, autocomplete=None, on_change=None, args=None, kwargs=None, *, placeholder=None, disabled=False, label_visibility="visible")
```

Parameters

A short label explaining to the user what this input 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.

For accessibility reasons, you should never set an empty label, but you can hide it with `label_visibility` if needed. In the future, we may disallow empty labels by raising an exception.

value (object or None)

The text value of this widget when it first renders. This will be cast to str internally. If `None`, will initialize empty and return `None` until the user provides input. Defaults to empty string.

Returns

(str or None)

The current value of the text input widget or `None` if no value has been provided by the user.

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

```
st.text_input(label, value="", max_chars=None, key=None, type="default", help=None, autocomplete=None, on_change=None, args=None, kwargs=None, *, placeholder=None, disabled=False, label_visibility="visible")
```


max_chars (int or None)	Max number of characters allowed in text input.
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.
type ("default" or "password")	The type of the text input. This can be either "default" (for a regular text input), or "password" (for a text input that masks the user's typed value). Defaults to "default".
help (str)	An optional tooltip that gets displayed next to the widget label. Streamlit only displays the tooltip when <code>label_visibility="visible"</code> .
autocomplete (str)	An optional value that will be passed to the <code><input></code> element's <code>autocomplete</code> property. If unspecified, this value will be set to "new-password" for "password" inputs, and the empty string for "default" inputs. For more details, see https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/autocomplete
on_change (callable)	An optional callback invoked when this text input's value changes.
args (tuple)	An optional tuple of args to pass to the callback.
kwargs (dict)	An optional dict of kwargs to pass to the callback.
placeholder (str or None)	An optional string displayed when the text input is empty. If None, no text is displayed.
disabled (bool)	An optional boolean that disables the text input if set to <code>True</code> . The default is <code>False</code> .
label_visibility ("visible", "hidden", or "collapsed")	The visibility of the label. The default is "visible". If this is "hidden", Streamlit displays an empty spacer instead of the label, which can help keep the widget aligned with other widgets. If this is "collapsed", Streamlit displays no label or spacer.
Returns	
(str or None)	The current value of the text input widget or <code>None</code> if no value has been provided by the user.

Example

```
import streamlit as st
```

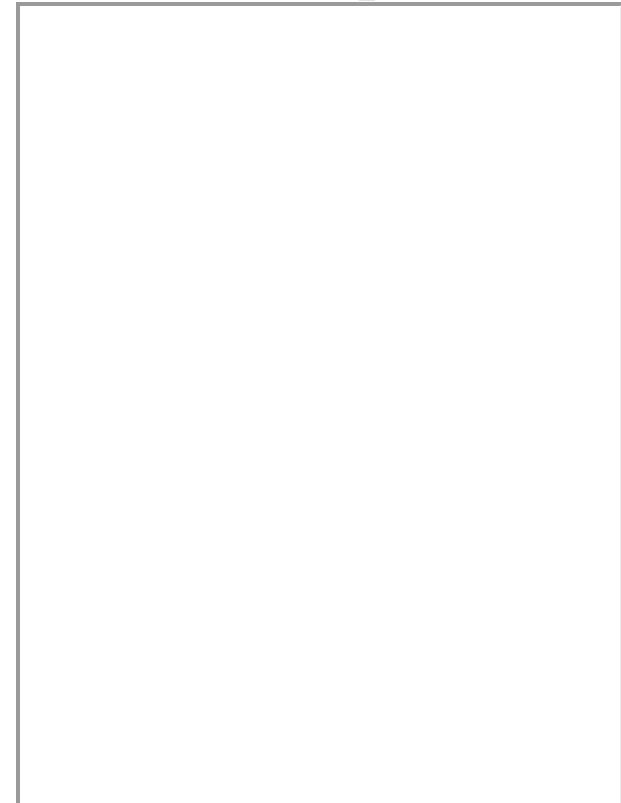
```
title = st.text_input("Movie title", "Life of Brian")
st.write("The current movie title is", title)
```



[Built with Streamlit](#) 
[Fullscreen open in new](#)

Text input widgets can customize how to hide their labels with the `label_visibility` parameter. If "hidden", the label doesn't show but there is still empty space for it above the widget (equivalent to `label=""`). If "collapsed", both the label and the space are removed. Default is "visible". Text input widgets can also be disabled with the `disabled` parameter, and can display an optional placeholder text when the text input is empty using the `placeholder` parameter:

```
import streamlit as st # Store the initial value of widgets in session state if "visibility" not in
st.session_state: st.session_state.visibility = "visible" st.session_state.disabled = False col1, col2
= st.columns(2) with col1: st.checkbox("Disable text input widget", key="disabled") st.radio( "Set
text input label visibility 🗑️", key="visibility", options=["visible", "hidden", "collapsed"], )
st.text_input( "Placeholder for the other text input widget", "This is a placeholder",
key="placeholder", ) with col2: text_input = st.text_input( "Enter some text 🗑️",
label_visibility=st.session_state.visibility, disabled=st.session_state.disabled,
placeholder=st.session_state.placeholder, ) if text_input: st.write("You entered: ", text_input)
```



Still have questions?

Our [forums](#) are full of helpful information and Streamlit experts.

