

[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](#)
remove
 - [st.audio](#)
 - [st.image](#)
 - [st.logo](#)
 - [st.video](#)
 - [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/](#)

- [Media elements/](#)

- [st.video](#)

st.video



Streamlit Version 

Display a video player.

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

st.video(data, format="video/mp4", start_time=0, *, subtitles=None, end_time=None, loop=False, autoplay=False, muted=False)

Parameters

```
st.video(data, format="video/mp4", start_time=0, *, subtitles=None, end_time=None, loop=False, autoplay=False, muted=False)
```

The video to play. This can be one of the following:

data (str, Path, bytes, io.BytesIO, numpy.ndarray, or file)

- A URL (string) for a hosted video file, including YouTube URLs.
- A path to a local video file. The path can be a `str` or `Path` object. Paths can be absolute or relative to the working directory (where you execute `streamlit run`).
- Raw video data. Raw data formats must include all necessary file headers to match the file format specified via `format`.

format (str)

The MIME type for the video file. This defaults to `"video/mp4"`. For more information, see <https://tools.ietf.org/html/rfc4281>.

The time from which the element should start playing. This can be one of the following:

start_time (int, float, timedelta, str, or None)

- `None` (default): The element plays from the beginning.
- An `int` or `float` specifying the time in seconds. `float` values are rounded down to whole seconds.
- A string specifying the time in a format supported by [Pandas' Timedelta constructor](#), e.g. `"2 minute"`, `"20s"`, or `"1m14s"`.
- A `timedelta` object from [Python's built-in datetime library](#), e.g. `timedelta(seconds=70)`.

Optional subtitle data for the video, supporting several input types:

subtitles (str, bytes, Path, io.BytesIO, or dict)

- `None` (default): No subtitles.
- A string, bytes, or `Path`: File path to a subtitle file in `.vtt` or `.srt` formats, or the raw content of subtitles conforming to these formats. Paths can be absolute or relative to the working directory (where you execute `streamlit run`). If providing raw content, the string must adhere to the WebVTT or SRT format specifications.
- `io.BytesIO`: A `BytesIO` stream that contains valid `.vtt` or `.srt` formatted subtitle data.
- A dictionary: Pairs of labels and file paths or raw subtitle content in `.vtt` or `.srt` formats to enable multiple subtitle tracks. The label will be shown in the video player. Example: `{"English": "path/to/english.vtt", "French": "path/to/french.srt"}`

When provided, subtitles are displayed by default. For multiple tracks, the first one is displayed by default. If you don't want any subtitles displayed by default, use an empty string for the value in a dictionary's first pair: `{"None": "", "English": "path/to/english.vtt"}`

Not supported for YouTube videos.

The time at which the element should stop playing. This can be one of the following:

end_time (int, float, timedelta, str, or None)

- `None` (default): The element plays through to the end.
- An `int` or `float` specifying the time in seconds. `float` values are rounded down to whole seconds.
- A string specifying the time in a format supported by [Pandas' Timedelta constructor](#), e.g. `"2 minute"`, `"20s"`, or `"1m14s"`.
- A `timedelta` object from [Python's built-in datetime library](#), e.g. `timedelta(seconds=70)`.

```
st.video(data, format="video/mp4", start_time=0, *, subtitles=None, end_time=None, loop=False, autoplay=False, muted=False)
```

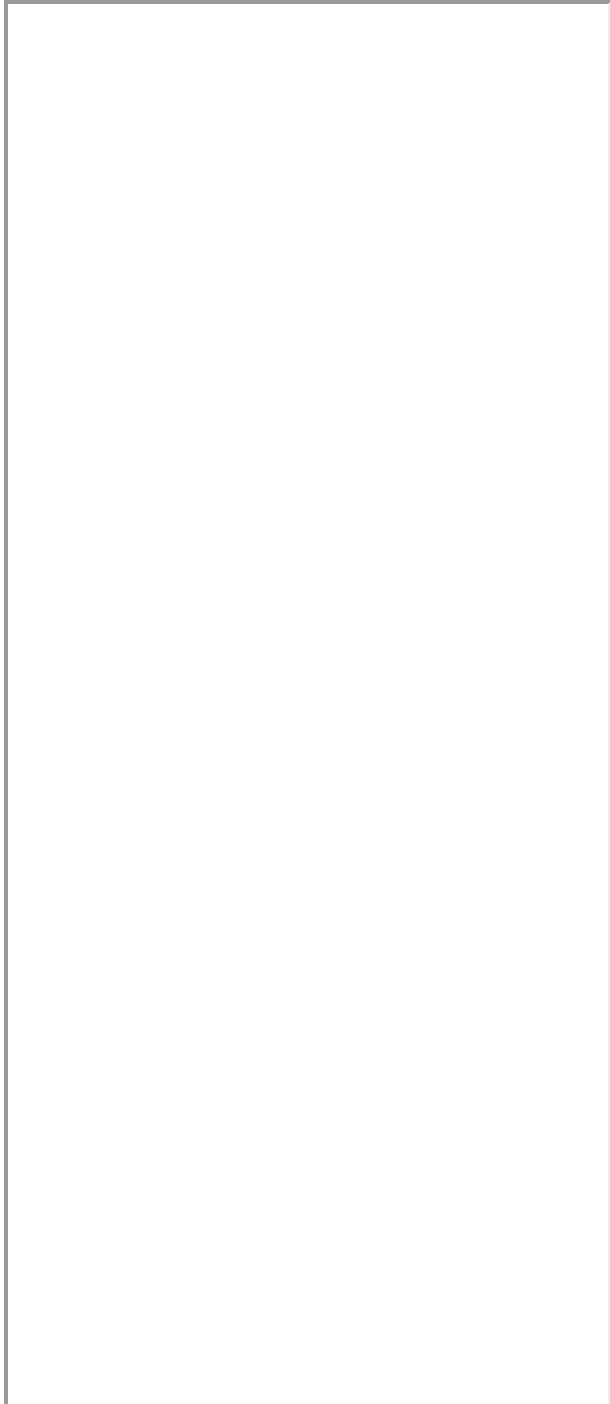
loop (bool)	Whether the video should loop playback.
autoplay (bool)	Whether the video should start playing automatically. This is <code>False</code> by default. Browsers will not autoplay unmuted videos if the user has not interacted with the page by clicking somewhere. To enable autoplay without user interaction, you must also set <code>muted=True</code> .
muted (bool)	Whether the video should play with the audio silenced. This is <code>False</code> by default. Use this in conjunction with <code>autoplay=True</code> to enable autoplay without user interaction.


Example

```
import streamlit as st

video_file = open("myvideo.mp4", "rb")
video_bytes = video_file.read()

st.video(video_bytes)
```



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

When you include subtitles, they will be turned on by default. A viewer can turn off the subtitles (or captions) from the browser's default video control menu, usually located in the lower-right corner of the video.

Here is a simple VTT file (`subtitles.vtt`):

```
WEBVTT
```

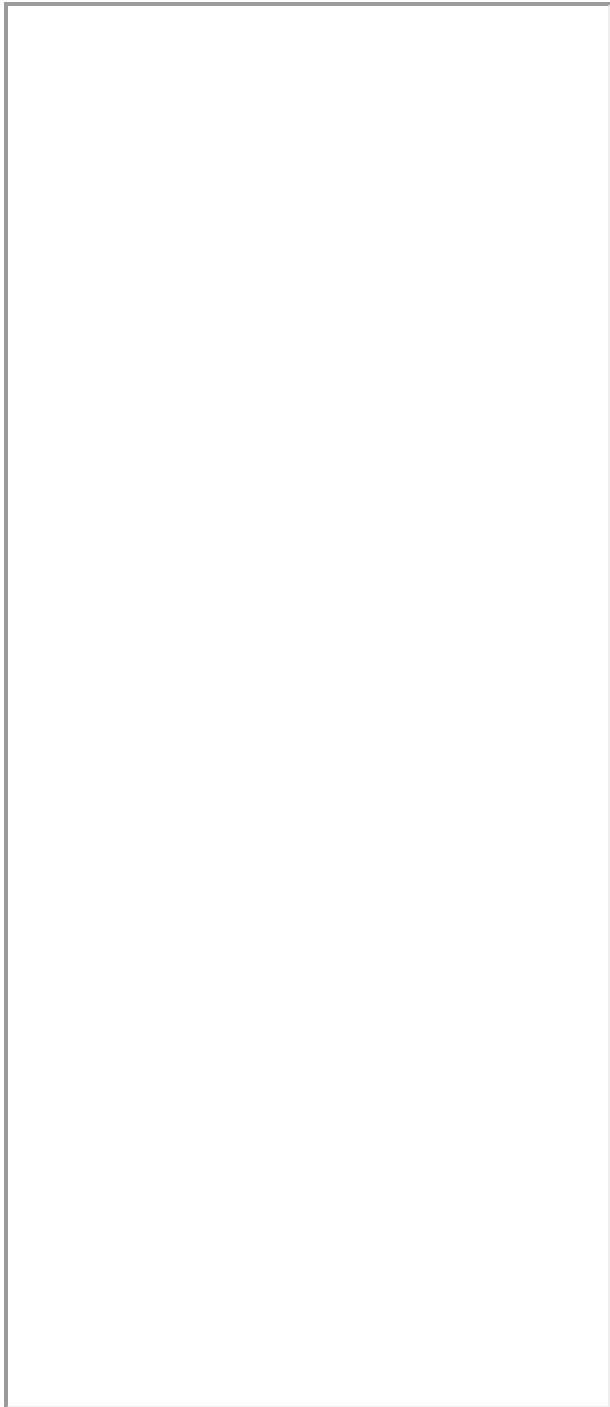
```
0:00:01.000 --> 0:00:02.000  
Look!
```


```
0:00:03.000 --> 0:00:05.000  
Look at the pretty stars!
```

If the above VTT file lives in the same directory as your app, you can add subtitles like so:

```
import streamlit as st
```

```
VIDEO_URL = "https://example.com/not-youtube.mp4"  
st.video(VIDEO_URL, subtitles="subtitles.vtt")
```



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

See additional examples of supported subtitle input types in our [video subtitles feature demo](#).

Note

Some videos may not display if they are encoded using MP4V (which is an export option in OpenCV), as this codec is not widely supported by browsers. Converting your video to H.264 will allow the video to be displayed in Streamlit. See this [StackOverflow post](#) or this [Streamlit forum post](#) for more information.

[←Previous: st.logo](#)[Next: Layouts and containers→](#)
forum

Still have questions?

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

