Merge videos using FFmpeg concat

One of the core concepts of video editing involves stitching or merging together different clips to create a narrative or story. This process is easily done within a video editing application by dragging and dropping different video clips and placing them along a timeline. But how would you go about this if you wanted to do this with code?

We'll see how to do exactly that using FFmpeg, a command-line utility, that can be used to create, edit and process different types of media. We'll look at two different ways to use FFmpeg and we'll end by looking at how you can achieve the same results using Shotstack, a cloud-based video editing API.

Use FFmpeg concat to merge videos

When using FFmpeg to merge a series of video clips, the process is called concatenation, or concat for short. There are several ways that videos can be concatenated, depending on whether the individual videos share the same codec or not. We are going to look at how to concat videos that share the same codec, for the latter, please refer to the [FFmpeg concatenate](#) documentation.

For videos that have the same codec, you can concatenate them using **concat demuxer** or **concat protocol**.

Merge a list of videos with FFmpeg using concat demuxer

To merge videos using the concat demuxer, first create a text file with a list of videos you want to join. Below is an example file `videos.txt` . You can add comments to the file, they will be ignored by the demuxer.

```
# videos.txt
file 'input1.mp4'
file 'input2.mp4'
file 'input3.mp4'
```

With that done, if you run the following command, the videos listed in that file will be merged, according to the order they are listed.

```
$ ffmpeg -f concat -i videos.txt -c copy output8.mp4
```

Merge all videos in a directory with FFmpeg using concat demuxer

If you have a large number of videos in a directory, it might be easier to write a command that will create a file with the list of videos for you before concatenating them. To determine the order that the videos will be listed, you can name them alphabetically or numerically. They will be listed according to [LC_COLLATE](#).

BY JOYCE ECHESSA

28th June, 2021

```
$ for f in *.mp4; do echo "file '$f'" >> videos.txt; done
```

The above will create a file `videos.txt` with a list of files with the extension `.mp4`. After the file is generated, you can then stitch the listed video with the command we used earlier:

```
$ ffmpeg -f concat -i videos.txt -c copy output9.mp4
```

Merge videos with the concat protocol

You can also join videos with the concat protocol. The following code stitches four videos without re-encoding them.

```
ffmpeg -i "concat:input1.mp4|input2.mp4|input3.mp4|input4.mp4" -c copy output10.mp4
```

Merging videos with the Shotstack API

FFmpeg is a versatile and powerful tool for video manipulation but it can be quite cumbersome to use. The syntax can be confusing and if you want to edit video at scale it requires a sophisticated and expensive hosting platform.

The Shotstack video editing API has been designed to make many common FFmpeg use cases easier by providing a simple to understand JSON editing specification and a scalable rendering platform that handles the editing and encoding process.

To see how easy it is to stitch videos with Shotstack have a look at our how to merge videos using an API guide.
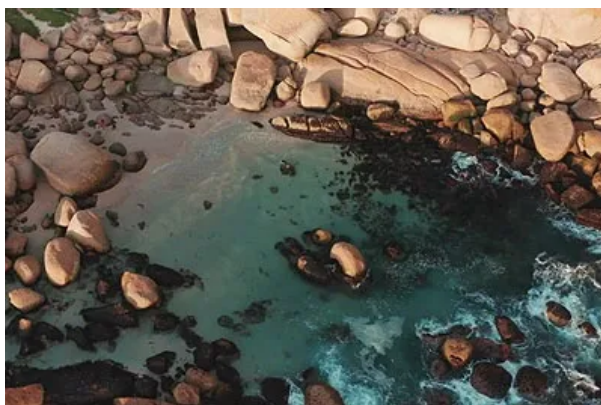

Related articles
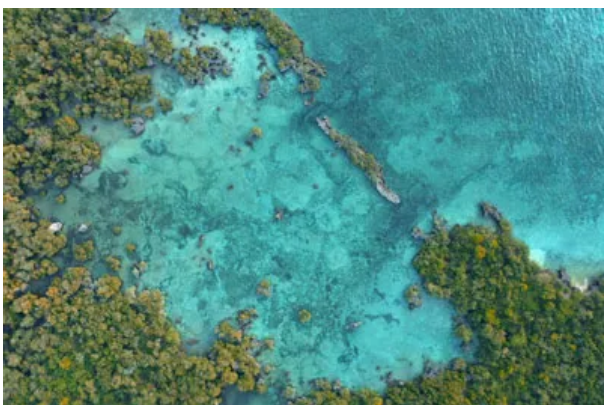
Merge videos using the Edit API

How to use the Shotstack video editing API to merge video clips together to edit a simple video.

Jeff Shillitto
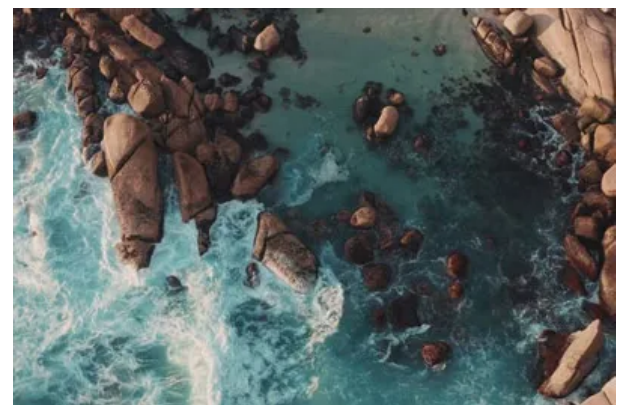24th September, 2021

How to trim a video using FFmpeg

Use FFmpeg to trim the start, end or middle of a video to create shorter segments.

Joyce Echessa
27th June, 2021

Trim a video using the Edit API

Learn how to trim the start, end or middle of a video using the Shotstack API.

Jeff Shillitto
20th January, 2022

SHOTSTACK

PRODUCT	SOLUTIONS	RESOURCES	ABOUT
Video Editing API	Startups	Learn	Our Story