



Basic Commands and Options

Basic Information

<code>ffmpeg -version</code>	Display FFmpeg version information.
<code>ffmpeg -formats</code>	List all available formats (containers, codecs, protocols).
<code>ffmpeg -codecs</code>	List all available codecs.
<code>ffmpeg -decoders</code>	List all available decoders.
<code>ffmpeg -encoders</code>	List all available encoders.
<code>ffmpeg -devices</code>	List available devices (e.g., cameras, microphones).

Basic Syntax

<code>ffmpeg [global_options] {[input_file_options] -i input_url} ... {[output_file_options] output_url} ...</code>
Example: <code>ffmpeg -i input.mp4 -c:v libx264 output.avi</code>

Global Options

<code>-y</code>	Overwrite output files without asking.
<code>-n</code>	Do not overwrite output files. Exit if specified output file exists.
<code>-i input_url</code>	Specify input file.
<code>-c[:stream_spec c:a] codec</code>	Specify codec. <code>-c:v</code> for video, <code>-c:a</code> for audio, <code>-c:s</code> for subtitles.
<code>-vn</code>	Disable video recording.
<code>-an</code>	Disable audio recording.

Video Processing

Basic Video Encoding

<code>ffmpeg -i input.mov -c:v libx264 output.mp4</code>	Encode video to H.264 (libx264).
<code>ffmpeg -i input.avi -c:v libvpx-vp9 -c:a libopus output.webm</code>	Encode video to VP9 (libvpx-vp9) with Opus audio.
<code>ffmpeg -i input.mov -c:v copy -c:a copy output.mp4</code>	Copy video and audio streams (re-mux).
<code>ffmpeg -i input.mov -c:v libx265 -preset slow -crf 28 output.mp4</code>	Encode video to H.265 (libx265) with CRF (Constant Rate Factor) for quality.

Video Resizing and Cropping

<code>ffmpeg -i input.mov -vf scale=640:480 output.mp4</code>	Resize video to 640x480.
<code>ffmpeg -i input.mov -vf scale=hd720 output.mp4</code>	Resize video to 720p (1280x720).
<code>ffmpeg -i input.mov -vf crop=w=320:h=240:x=100:y=50 output.mp4</code>	Crop video to 320x240 starting at coordinates (100, 50).
<code>ffmpeg -i input.mov -vf "scale=iw*0.5:ih*0.5" output.mp4</code>	Scale the video to half of the original width and height.

Frame Rate and Aspect Ratio

<code>ffmpeg -i input.mov -r 30 output.mp4</code>	Set frame rate to 30 fps.
<code>ffmpeg -i input.mov -aspect 16:9 output.mp4</code>	Set aspect ratio to 16:9.
<code>ffmpeg -i input.mov -r 24 -aspect 4:3 output.mp4</code>	Change both frame rate and aspect ratio.

Audio Processing

Basic Audio Encoding

<code>ffmpeg -i input.wav -c:a libmp3lame output.mp3</code>	Encode audio to MP3 (libmp3lame).
<code>ffmpeg -i input.wav -c:a aac -b:a 128k output.aac</code>	Encode audio to AAC with a bitrate of 128kbps.
<code>ffmpeg -i input.mp3 -c:a copy output.mp3</code>	Copy audio stream (re-mux).

Audio Conversion and Manipulation

<code>ffmpeg -i input.mp3 -ar 44100 output.wav</code>	Change audio sample rate to 44100 Hz.
<code>ffmpeg -i input.wav -ac 2 output.wav</code>	Change audio channels to stereo.
<code>ffmpeg -i input.mp3 -af volume=0.5 output.mp3</code>	Reduce audio volume to 50%.
<code>ffmpeg -i input.mp3 -ss 00:00:10 -to 00:00:20 output.mp3</code>	Extract audio snippet from 10 to 20 seconds.

Extracting Audio from Video

<code>ffmpeg -i input.mp4 -vn -acodec copy output.aac</code>	Extract audio as AAC.
<code>ffmpeg -i input.mov -vn -acodec libmp3lame output.mp3</code>	Extract audio as MP3.

Advanced Techniques

Concatenating Media Files

Create a text file `mylist.txt` with the following content:

```
file 'input1.mp4'  
file 'input2.mp4'  
file 'input3.mp4'
```

Then run:

```
ffmpeg -f concat -safe 0 -i mylist.txt -c copy  
output.mp4
```

Note: The files must have the same encoding and properties for seamless concatenation.

Adding Subtitles

```
ffmpeg -i input.mp4 -i subtitles.srt -c copy -  
c:s mov_text output.mp4
```

Add subtitles to a video file.

```
ffmpeg -i input.mp4 -vf subtitles=subtitles.srt  
output.mp4
```

Another way to add subtitles (may require specific codecs).

Creating GIFs

```
ffmpeg -i input.mp4 -t 3 -vf  
"fps=10, scale=320:-1:f1  
ags=lanczos, split[s0]  
[s1];[s0]palettegen[p];  
[s1][p]paletteuse"  
output.gif
```

Create a GIF from a video, limiting to the first 3 seconds, setting frame rate, scaling and using palette optimization.

```
ffmpeg -i input.mp4 -ss 00:01:00 -t 00:00:05  
-vf  
"fps=10, scale=320:-1:f1  
ags=lanczos, split[s0]  
[s1];[s0]palettegen[p];  
[s1][p]paletteuse"  
output.gif
```

Convert specific part of the video to gif.