# ffmpeg Cheatsheet

By Dejan Panovski • Updated on Sep 13, 2026 •

Keep essential ffmpeg commands close at hand, from stream inspection and lossless remuxing to CRF encoding, filters, subtitles, and GPU acceleration.

ffmpeg reads and writes nearly every audio and video format from the command line. This cheatsheet covers inspecting files, converting containers, compressing video with CRF, extracting audio, scaling, trimming, concatenating, working with subtitles, and common filter recipes.

## Install ffmpeg

Get the ffmpeg binaries and confirm the build.

| | |
|---|---|
| `sudo apt install ffmpeg` | Install on Ubuntu, Debian, and derivatives |
| `sudo dnf install ffmpeg` | Install the full build from RPM Fusion on Fedora or RHEL |
| `sudo dnf install ffmpeg-free` | Install the codec-limited build from Fedora or EPEL |
| `ffmpeg -version` | Print the version and build configuration |
| `ffmpeg -encoders` | List every available encoder |
| `ffmpeg -decoders` | List every available decoder |
| `ffmpeg -formats` | List supported container formats |

## Inspect Media Files

Read stream details before deciding how to re-encode.

| | |
|---|---|
| `ffprobe -hide_banner in.mp4` | Show streams, codecs, and duration |
| `ffprobe -v error -show_format in.mp4` | Print container metadata only |
| `ffprobe -v error -show_streams in.mp4` | Print every stream property |
| `ffprobe -v error -print_format json -show_format -show_streams in.mp4` | Machine-readable output |
| `ffprobe -v error -select_streams v:0 -show_entries stream=width,height -of csv=p=0 in.mp4` | Print the video resolution |
| `ffprobe -v error -show_entries format=duration -of csv=p=0 in.mp4` | Print the duration in seconds |

## Common Options

Options that control inputs, outputs, and command behavior.

| | |
|---|---|
| `-i input` | Set an input file (repeat for multiple inputs) |
| `-y` | Overwrite the output file without asking |
| `-n` | Never overwrite an existing output file |
| `-hide_banner` | Suppress the build and library banner |
| `-loglevel error` | Print errors only |
| `-stats` | Keep the progress line while quiet |
| `-threads:v 4` | Request four threads for the video encoder when supported |
| `-f format` | Force an input or output format |

## Convert Between Formats

Change the container, with or without re-encoding.

| | |
|---|---|
| `ffmpeg -i in.mkv -c copy out.mp4` | Repackage compatible streams without re-encoding |
| `ffmpeg -i in.avi out.mp4` | Convert using the default encoders |
| `ffmpeg -i in.mov -c:v libx264 -c:a aac out.mp4` | Convert to H.264 and AAC |
| `ffmpeg -i in.mp4 -c:v libvpx-vp9 -c:a libopus out.webm` | Convert to WebM |
| `ffmpeg -i in.mp4 -c:v copy -c:a aac out.mp4` | Re-encode the audio, keep the video |
| `ffmpeg -i in.wav out.flac` | Convert between audio formats |

## Compress Video

Lower the bitrate with constant quality or a controlled bitrate.

| | |
|---|---|
| `ffmpeg -i in.mp4 -c:v libx264 -crf 23 -preset slow -c:a aac -b:a 128k out.mp4` | Compress with H.264 |
| `ffmpeg -i in.mp4 -c:v libx265 -crf 28 -c:a copy out.mp4` | Compress with H.265 at a similar quality |
| `ffmpeg -i in.mp4 -c:v libx264 -crf 18 out.mp4` | Keep near-source quality |
| `ffmpeg -i in.mp4 -c:v libx264 -b:v 2M -maxrate 2M -bufsize 4M -c:a copy out.mp4` | Target 2 Mb/s with a 2 Mb/s VBV ceiling |
| `ffmpeg -i in.mp4 -c:v h264_nvenc -cq 23 -c:a copy out.mp4` | Encode on an NVIDIA GPU |
| `ffmpeg -vaapi_device /dev/dri/renderD128 -i in.mp4 -vf 'format=nv12,hwupload' -c:v h264_vaapi -c:a copy out.mp4` | Encode with VA-API |

CRF runs from 0 (lossless) to 51 (worst). Lower values mean larger files, and 18 to 28 covers most work. Slower presets such as `slow` or `veryslow` shrink the file further at the same CRF.

## Audio

Extract, convert, and adjust audio tracks.

| | |
|---|---|
| `ffmpeg -i in.mp4 -vn -c:a libmp3lame -q:a 2 out.mp3` | Extract audio as MP3 |
| `ffmpeg -i in.mp4 -vn -c:a copy out.m4a` | Copy compatible audio without re-encoding |
| `ffmpeg -i in.mp4 -vn -c:a aac -b:a 192k out.aac` | Extract audio as AAC |
| `ffmpeg -i in.mp3 -ac 1 -ar 16000 out.wav` | Downmix to mono at 16 kHz |
| `ffmpeg -i in.mp4 -c:v copy -af 'volume=1.5' out.mp4` | Raise the volume by 50 percent |
| `ffmpeg -i in.mp4 -c:v copy -af 'loudnorm=I=-23' -ar 48000 out.mp4` | Apply one-pass EBU R128 normalization at -23 LUFS |
| `ffmpeg -i in.mp4 -an -c:v copy out.mp4` | Strip the audio track |

The `-q:a` scale for libmp3lame runs from 0 (best) to 9 (worst), and 2 is a good default for speech and music alike.

## Resize, Crop, and Rotate

Reshape the picture with video filters.

| | |
|---|---|
| `ffmpeg -i in.mp4 -vf scale=1280:-2 -c:a copy out.mp4` | Scale to 1280 wide, keep the aspect ratio |
| `ffmpeg -i in.mp4 -vf scale=-2:720 -c:a copy out.mp4` | Scale to 720 high |
| `ffmpeg -i in.mp4 -vf scale=1920:1080 -c:a copy out.mp4` | Force an exact resolution |
| `ffmpeg -i in.mp4 -vf 'crop=640:480:100:50' -c:a copy out.mp4` | Crop 640x480 starting at x=100, y=50 |
| `ffmpeg -i in.mp4 -vf 'transpose=1' -c:a copy out.mp4` | Rotate 90 degrees clockwise |
| `ffmpeg -i in.mp4 -vf 'hflip' -c:a copy out.mp4` | Mirror horizontally |
| `ffmpeg -i in.mp4 -vf 'pad=1920:1080:(ow-iw)/2:(oh-ih)/2' -c:a copy out.mp4` | Pad to 1080p with centered bars |

Use `-2` rather than `-1` for the free dimension so the calculated size stays even, as required by common 4:2:0 output formats.

## Trim and Concatenate

Cut clips and join files back together.

| | |
|---|---|
| `ffmpeg -ss 00:01:00 -to 00:02:30 -i in.mp4 -c copy clip.mp4` | Fast cut at the nearest keyframe |
| `ffmpeg -ss 00:01:00 -to 00:02:30 -i in.mp4 -c:v libx264 -c:a aac clip.mp4` | Frame-accurate cut by re-encoding |
| `ffmpeg -ss 00:00:30 -t 15 -i in.mp4 -c copy clip.mp4` | Cut 15 seconds starting at 30 seconds |
| `ffmpeg -i in.mp4 -t 60 -c copy first-minute.mp4` | Keep the first minute |
| `ffmpeg -f concat -safe 0 -i list.txt -c copy out.mp4` | Join files listed in list.txt |
| `ffmpeg -i in.mp4 -c copy -f segment -segment_time 600 part%03d.mp4` | Split near ten-minute boundaries at keyframes |

Each line of `list.txt` takes the form `file '/path/to/clip.mp4'`. The concat demuxer requires matching stream layouts and parameters, including codecs and time bases.

## Images, Thumbnails, and GIFs

Move between video and still images.

| | |
|---|---|
| `ffmpeg -ss 00:00:10 -i in.mp4 -frames:v 1 thumb.jpg` | Grab a single frame at ten seconds |
| `ffmpeg -i in.mp4 -vf fps=1 frame%04d.png` | Export one frame per second |
| `ffmpeg -i in.mp4 -vf 'fps=1/10' frame%04d.jpg` | Export one frame every ten seconds |
| `ffmpeg -framerate 30 -i frame%04d.png -c:v libx264 -pix_fmt yuv420p out.mp4` | Build a video from an image sequence |
| `ffmpeg -i in.mp4 -vf 'fps=10,scale=480:-1:flags=lanczos' out.gif` | Create a GIF |
| `ffmpeg -i in.mp4 -vf 'fps=10,scale=480:-1:flags=lanczos,palettegen' palette.png` | Build a GIF color palette |
| `ffmpeg -i in.mp4 -i palette.png -lavfi 'fps=10,scale=480:-1:flags=lanczos [x]; [x][1:v] paletteuse' out.gif` | Create a GIF with that palette |

## Subtitles

Attach, burn in, or pull out subtitle tracks.

| | |
|---|---|
| `ffmpeg -i in.mp4 -i subs.srt -map 0 -map 1:0 -c copy -c:s mov_text out.mp4` | Add soft subtitles to MP4 |
| `ffmpeg -i in.mkv -i subs.srt -map 0 -map 1:0 -c copy out.mkv` | Add soft subtitles to MKV |
| `ffmpeg -i in.mp4 -vf subtitles=subs.srt -c:a copy out.mp4` | Burn subtitles into the picture |
| `ffmpeg -i in.mkv -vf 'subtitles=in.mkv:si=0' -c:a copy out.mp4` | Burn in an embedded subtitle stream |
| `ffmpeg -i in.mkv -map 0:s:0 subs.srt` | Convert the first subtitle track to SRT when it is text based |
| `ffmpeg -i in.mkv -map 0 -sn -c copy out.mkv` | Remove every subtitle track and keep other streams |

## Stream Mapping and Metadata

Choose which streams reach the output.

| | |
|---|---|
| `ffmpeg -i in.mkv -map 0:v:0 -map 0:a:1 -c copy out.mkv` | Keep the first video and second audio stream |
| `ffmpeg -i in.mkv -map 0 -c copy out.mkv` | Keep every stream from the input |
| `ffmpeg -i video.mp4 -i audio.mp3 -map 0:v -map 1:a -c:v copy -shortest out.mp4` | Combine separate video and audio files |
| `ffmpeg -i in.mp4 -map_metadata -1 -c copy out.mp4` | Strip global metadata |
| `ffmpeg -i in.mp3 -metadata title='Track name' -c copy out.mp3` | Set a metadata tag |
| `ffmpeg -i in.mp4 -c copy -movflags +faststart out.mp4` | Move the MP4 index to the front for streaming |
| `for f in *.mkv; do ffmpeg -i "$f" -c:v libx264 -crf 23 -c:a aac "${f%.mkv}.mp4"; done` | Batch convert a directory |

## Related Guides

Use these guides for the longer explanations behind these commands.

| | |
|---|---|
| ffmpeg Command in Linux | Full ffmpeg tutorial with worked examples |
| Convert MP4 to MP3 with ffmpeg | Audio extraction and quality settings in detail |