FFmpeg cheat sheet - 20 commands every developer needs

Inspect, copy, transcode, package, watermark, accelerate, stream - on one page Fora Soft Learn

Companion to article 5.2 of Block 5. Print on A4. FFmpeg 7.1 / 8.1.

1. Inspect (run before every transcode)

```
ffprobe -hide_banner -show_streams -show_format input.mp4         # full stream and container dump
ffprobe -v error -select_streams v:0 -show_entries stream=codec_name,width,height,bit_rate -of csv=p=0 input.mp4
```

2. Stream copy (100x+ realtime, no quality loss)

```
ffmpeg -i input.mkv -c copy output.mp4                            # re-wrap container only
ffmpeg -ss 00:00:10 -to 00:01:10 -i input.mp4 -c copy output.mp4  # trim on keyframes (fast)
ffmpeg -i input.mp4 -c copy -movflags +faststart output.mp4       # move moov atom to the front
```

3. Universal web MP4 - H.264 + AAC + faststart (works everywhere)

```
ffmpeg -i input.mov -c:v libx264 -preset medium -crf 23 -profile:v high -level 4.0 \
       -pix_fmt yuv420p -c:a aac -b:a 192k -ac 2 -movflags +faststart output.mp4
```

4. Modern web AV1 - SVT-AV1 preset 6, 10-bit, film grain

```
ffmpeg -i input.mov -c:v libsvtav1 -preset 6 -crf 30 -pix_fmt yuv420p10le \
       -svtav1-params tune=0:film-grain=8 -c:a libopus -b:a 128k -movflags +faststart out.mp4
```

5. Apple HEVC - x265 with hvc1 tag (required for iPhone / Safari)

```
ffmpeg -i input.mov -c:v libx265 -preset medium -crf 26 \
       -x265-params vbv-maxrate=5000:vbv-bufsize=10000 -tag:v hvc1 -pix_fmt yuv420p \
       -c:a aac -b:a 192k -movflags +faststart out.mp4
```

6. Hardware acceleration (live and conferencing - 10x to 100x faster)

```
NVENC : ffmpeg -hwaccel cuda -i in -c:v h264_nvenc -preset p5 -tune hq -cq 23 out.mp4
QSV   : ffmpeg -hwaccel qsv  -i in -c:v hevc_qsv -preset medium -global_quality 23 out.mp4
VTBox : ffmpeg -i in -c:v hevc_videotoolbox -q:v 60 -tag:v hvc1 out.mp4   # Apple Silicon
```

7. Adaptive Bitrate ladder + fMP4 HLS (CMAF, works with DASH too)

```
ffmpeg -i in -filter_complex "[0:v]split=4[v1][v2][v3][v4];[v1]scale=1920:1080[a]; \
       [v2]scale=1280:720[b];[v3]scale=854:480[c];[v4]scale=640:360[d]" \
       -map [a] -c:v:0 libx264 -b:v:0 5000k -maxrate:v:0 5500k -bufsize:v:0 10000k \
       -map [b] -c:v:1 libx264 -b:v:1 2800k -map [c] -c:v:2 libx264 -b:v:2 1400k \
       -map [d] -c:v:3 libx264 -b:v:3 800k  -map a:0 -c:a aac -b:a 128k \
       -f hls -hls_segment_type fmp4 -hls_time 4 -master_pl_name master.m3u8 \
       -var_stream_map "v:0,a:0 v:1,a:0 v:2,a:0 v:3,a:0" stream_%v/playlist.m3u8
```

8. Filters - scale, crop, overlay, drawtext, deinterlace, denoise

```
Scale 720p :  -vf "scale=1280:-2"                                  # -2 keeps even height (H.264 needs it)
Crop 9:16  :  -vf "crop=ih*9/16:ih"                                # vertical from horizontal source
Watermark  :  -i logo.png -filter_complex "[0:v][1:v]overlay=W-w-20:H-h-20"
Timestamp  :  -vf "drawtext=text='%{pts\:hms}':x=20:y=20:fontsize=28:fontcolor=white:box=1:boxcolor=black@0.5"
Deinterlace:  -vf yadif=mode=1                                      # broadcast / set-top sources
Denoise    :  -vf hqdn3d=4:3:6:4.5                                  # low-light cleanup
```

9. Audio, thumbnails, GIF previews

```
Extract MP3 :  ffmpeg -i in.mp4 -vn -c:a libmp3lame -b:a 320k out.mp3
Loudnorm    :  ffmpeg -i in.mp4 -af "loudnorm=I=-16:LRA=11:TP=-1.5" -c:v copy out.mp4
Thumbnail   :  ffmpeg -ss 5 -i in.mp4 -vframes 1 -q:v 2 thumb.jpg
Sprite 8x5  :  ffmpeg -i in.mp4 -vf "fps=1/60,scale=160:90,tile=8x5" sprite.jpg
GIF preview :  ffmpeg -ss 30 -t 5 -i in.mp4 -vf "fps=12,scale=480:-1,split[a][b];[a]palettegen[p];[b][p]paletteuse" out.gif
```

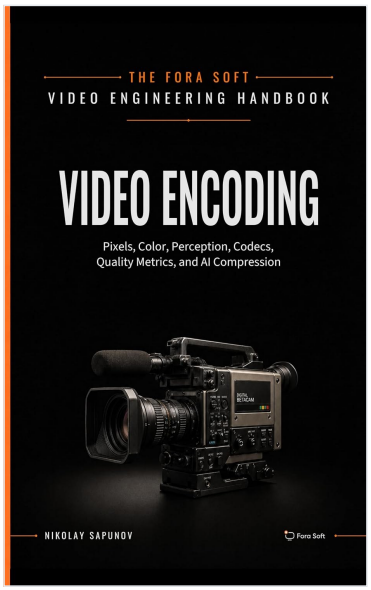
10. Live ingest - RTMP, SRT, WHIP (WebRTC)

```
RTMP  : ffmpeg -re -i in.mp4 -c:v libx264 -preset veryfast -tune zerolatency -b:v 4500k -maxrate 4500k \
        -bufsize 4500k -g 60 -keyint_min 60 -sc_threshold 0 -c:a aac -b:a 160k -f flv rtmp://...
SRT   : same as RTMP but -f mpegts srt://ingest.example.com:9000?streamid=publish/event42
WHIP  : ffmpeg -re -i in.mp4 -c:v libx264 -profile:v baseline -tune zerolatency -b:v 2500k \
        -c:a libopus -b:a 96k -ar 48000 -ac 2 -f whip https://whip.example.com/publish/event42
```

11. Common mistakes (and the fix)

Mistake	Fix
-ss after -i for trim	put -ss BEFORE -i for fast keyframe seek
forgot -pix_fmt yuv420p on Apple source	always set yuv420p for web; pro sources are 422/10-bit
-crf and -b:v together	pick one: -crf only OR -b:v only OR -crf + -maxrate + -bufsize
-c copy + a filter in same step	filter requires re-encode; drop -c copy
HEVC silent on iPhone	add -tag:v hvc1 (Apple needs it, default is hev1)
29.97 mixed with 30 fps	lock the framerate at source: -r 30000/1001 or -r 30, never both

(c) 2026 Fora Soft. v1.0



GO DEEPER · VOLUME 1 OF 7

Video Encoding: A Complete Guide to Video Compression

From Pixels and Color to H.264, HEVC, AV1, Quality Metrics, and AI

By Nikolay Sapunov, CEO at Fora Soft · 1,522 pages · Kindle & paperback on Amazon

Read it on Amazon → amazon.com/dp/B0HF52B65R

This cheat sheet gives you twenty working FFmpeg commands for inspecting, transcoding, packaging and streaming. Volume 1 explains what each flag changes: H.264, HEVC and AV1 encoder settings, rate control with CRF and VBV, pixel formats and bit depth, and how to check the result with quality metrics.

WHAT YOU WILL FIND INSIDE

- H.264, HEVC and AV1 encoder settings
- CRF, VBV and rate control
- Pixel formats, bit depth and color
- Quality metrics: PSNR, SSIM, VMAF

THE FULL HANDBOOK · 7 VOLUMES · 9,000+ PAGES · ALL ON AMAZON

Vol. 1 · Video Encoding — pixels and color to H.264, HEVC, AV1, quality metrics and AI
Vol. 2 · Video Streaming I — network transport, contribution and ingest, HLS, MPEG-DASH, CMAF
Vol. 3 · Video Streaming II — ABR, CDNs, players, WebRTC, DRM and the economics of delivery
Vol. 4 · Audio for Video I — PCM, codecs, loudness, accessibility, real-time voice processing
Vol. 5 · Audio for Video II — jitter buffers, lip sync, Atmos, quality metrics, neural codecs
Vol. 6 · AI for Video I — models and inference, computer vision, speech, search, generative video
Vol. 7 · AI for Video II — real-time pipelines, agents, serving, cost, compliance, verticals

Building a transcoding pipeline on FFmpeg?

We will review your commands and encoder settings for your content and devices and send back tuned presets with expected bitrate savings. 250+ video products shipped since 2005.

Book a 30-min call → **WhatsApp →** **Email us →**