MAKE THINGS

Editing Video

Cuts, sound, colour and export — taught on DaVinci Resolve, which costs nothing.

First edition, September 2026

Addaly

Series editor: Dr Mustafa Mun

Draft, open for academic review

MAKE THINGS

Editing Video

Cuts, sound, colour and export — taught on DaVinci Resolve, which costs nothing.

Series editor: Dr Mustafa Mun

Addaly

First edition, September 2026 · Draft, open for academic review

Editing Video

First edition, September 2026, build 62a26075.

Written by AI models to a written standard, with Dr Mustafa Mun as series editor. Exam keys checked blind by a second model: 3,665 of 3,666 agreed. Academic review invited.

The manuscript was drafted with the assistance of a large language model and was edited and published under his name. That is stated plainly here rather than left to be discovered, because a reader is entitled to know how a book they are trusting was made.

How to cite: *Mun, M. (2026). Editing Video. Addaly. addaly.com/learn/video-editing.*

This book is the printed form of the course of the same name at addaly.com/learn/video-editing. The website is the living version and is corrected first. Where the two differ, the website is right.

Free to read, free to download, free to print, and free to teach from. There is no paid edition and there never will be.

Every diagram in it is drawn from data by code, not generated as a picture, so the numbers on the page are the numbers in the argument. The answers to every check, test and examination question are at the back.

8 modules · 77 lessons · 28 figures · 8 case studies · 66,565 words · about 11 hours

Brief contents

Contents

1

MODULE 1

Getting the footage into a state you can work with

Set up a project whose frame rate, resolution, colour handling and media layout are chosen from the footage rather than from a default, offload and verify a card so you can prove the copy is byte-identical, and diagnose a clip that drifts out of sync by naming the property of the source file that causes it.

Lesson 1 · 10 min

Your laptop is probably not too slow

THE ONE THING TO KEEP

Camera files store frames as differences, not pictures, so edit a proxy and let the export go back to the original.

The belief that stops most people

Someone imports twenty minutes of 4K video off a phone, drops it on a timeline, presses play, and gets three frames per second and a fan noise. They conclude their computer cannot edit video and they need a better one. Usually they do not. They need to stop editing the delivery file.

Here is the mechanism, and it is worth understanding once because it explains half of what follows in this course.

The H.264 and H.265 files your phone and camera produce are **long-GOP** formats. They do not store every frame. They store an occasional complete frame — a keyframe — and then, for everything between, only the differences from neighbouring frames. That is why a ten-minute phone clip is 1 GB instead of 100 GB. It is superb for delivery and terrible for editing, because to show you the frame at 4 minutes 12 seconds the decoder has to find the last keyframe and reconstruct every frame since. Scrub backwards and it does that continuously. Your machine is not weak; you have asked it to solve a puzzle sixty times a second.

Proxies: the single biggest quality-of-life change

A proxy is a low-resolution, easy-to-decode copy of each clip. You edit the proxy. At export the editor silently switches back to the original file, so nothing about your final quality changes.

Figure 1.1

Why the proxy plays and the camera file does not

The long-GOP camera file

- One whole frame every one to two seconds
- Everything between is stored as differences
- Showing one frame rebuilds every frame since the last whole one
- Ten minutes of 4K fits in about a gigabyte
- Three frames a second and a hot fan

A quarter-size DNxHR proxy

- Every frame stored whole
- Scrubbing lands where you put it
- An hour of 4K costs 8 to 15 GB of disk
- About twenty minutes to build, once
- The export silently goes back to the original

Nothing about the finished quality changes. The proxy exists only between import and export, which is why the machine is almost never the problem.

Every serious editor does this, including the free ones.

- **DaVinci Resolve** (free): right-click clips in the Media Pool and generate optimised media or proxy media, then toggle proxy use in the Playback menu. Set proxy resolution to quarter or half.
- **Kdenlive** (free, open source): proxy clips are built in, with a checkbox in project settings to create them automatically on import.
- **Shotcut** (free, open source, light): a Proxy menu with a single toggle. Shotcut is the one to try first on an old or low-RAM machine — it is far less demanding than Resolve.
- **Premiere Pro** (subscription, and the name most job ads use): proxies via ingest settings or by attaching them after import.

- **CapCut** and phone editors handle this invisibly and are honestly fine for short-form.

If you want to make proxies by hand, FFmpeg is free and does it in one line:

```
ffmpeg -i source.mp4 -vf scale=-2:540 -c:v dnxhd -profile:v
dnxhr_lb \
       -c:a pcm_s16le proxy.mov
```

DNxHR LB is a good cross-platform intermediate: every frame stored whole, so scrubbing is instant, and it encodes on Windows and Linux without the licensing awkwardness ProRes has outside macOS.

The trade is disk space and a one-off wait. A quarter-resolution proxy of an hour of 4K might be 8–15 GB and take twenty minutes to build. You get that back in the first afternoon.

The timeline is a set of instructions, not a copy of your video

This is the second idea that changes how you work. When you drag a clip to the timeline, the editor does not copy the video. It stores a pointer to a file on disk plus a set of instructions: start at 00:04:12:08, run for 3 seconds, at 80% volume, with these colour adjustments.

Three consequences follow, and all three bite beginners.

- **Editing is non-destructive.** Nothing you do in the timeline changes the source file. You can trim, undo, and re-trim forever. Stop duplicating clips "to be safe".
- **Moving or renaming a source file breaks the link.** The pointer no longer resolves and you get a wall of red offline clips. Relinking is possible in every editor and it will still cost you an evening. Decide on file names and folders *before* you import.
- **The project file is tiny and the footage is not.** A Resolve or Kdenlive project is a few megabytes. That matters for backups, which the last lesson comes back to.

Bins, not a pile

The Media Pool or project bin is where an edit is won or lost on projects longer than five minutes. A flat list of 300 clips named C0034.MP4 is unusable by week two.

Make bins before you import: `interview`, `b-roll`, `audio`, `music`, `graphics`. Inside `b-roll`, subdivide by subject or by day, not by camera. Use markers and clip colours to flag the takes you know you want. In Resolve, Smart Bins can gather clips by keyword automatically, which is worth setting up once for anything with more than an hour of material.

The habit that pays most: watch everything once before you cut anything, marking in and out points on the good bits as you go. It feels like a waste of an afternoon. It is the afternoon that makes the edit take two days instead of five.

Where the free tools genuinely lose

Be honest with yourself about this. Resolve's free version does an enormous amount — the full colour page, the Fairlight audio page, Fusion compositing — and it costs nothing, permanently. What it holds back for Resolve Studio (a one-off purchase, around US$295 in 2026, no subscription) includes temporal and spatial noise reduction, optical-flow retiming for smooth slow motion, and several machine-learning features. Output is capped at 4K UHD. That list moves with every release, so read Blackmagic's own comparison rather than any list in a course.

Resolve is also the heaviest of the free editors and wants a real GPU and 16 GB of RAM to be comfortable. On an eight-year-old laptop with integrated graphics, Shotcut or Kdenlive will be a better afternoon than Resolve will.

Today

Take the longest clip you own, generate a proxy, and scrub through both. The difference is the whole lesson.

BEFORE YOU MOVE ON

An editor working on a laptop enables proxy media, and playback becomes smooth. A colleague warns that the final video will now be low resolution. What is the accurate response?

A. Proxies are only used during playback; at export the editor relinks to the original camera files, so output resolution and quality are unchanged.
B. The colleague is right, and the fix is to disable proxies before exporting so the timeline re-renders at full quality.
C. Quality drops slightly, because the colour and exposure adjustments were judged against a compressed proxy image.
D. It depends on the proxy resolution: half-resolution proxies export fine, quarter-resolution ones soften the picture.

The answer and why: p. 443

Lesson 2 · 10 min

What the camera actually wrote to the card

THE ONE THING TO KEEP

A camera file carries four fixed decisions — whole frames or differences, how much colour was kept, how many brightness levels, and how many bits per second — and each one sets a hard limit on what the edit can do later.

Four decisions the camera made without asking you

A video file is not a stack of photographs. It is the result of four separate compromises, made at the moment of recording, that you inherit for the rest of the project. You cannot undo any of them in the edit. Knowing what they were tells you, before you start, which shots will hold up and which will fall apart the moment you touch them.

One: which frames are whole

Intra-frame codecs store every frame complete. ProRes, DNxHR and most camera RAW work this way. Files are large and scrubbing is instant, because frame 4,312 is simply there.

Long-GOP codecs — the H.264 and H.265 that phones and consumer cameras write — store an occasional complete frame and describe everything in between as differences. GOP means group of pictures; a typical consumer camera uses a keyframe every 30 to 60 frames, so one to two seconds apart.

The consequence bites in three places. Playback is heavy, which the proxy lesson solved. Cutting is imprecise in some tools, because a cut in the middle of a GOP has to be reconstructed. And each re-encode compounds: every generation of H.264 throws away detail the previous one kept.

Two: how much colour was kept

Cameras record more brightness detail than colour detail, because the eye notices luminance far more than chroma. The notation is three numbers.

- **4:4:4** — full colour for every pixel. Rare outside cinema cameras and graphics.
- **4:2:2** — colour sampled at half horizontal resolution. Most professional cameras, and many mirrorless cameras when recording externally.
- **4:2:0** — colour sampled at half horizontal *and* half vertical resolution. Every phone, every consumer camera, every streaming delivery file.

In 4:2:0 one colour sample covers a 2×2 block of pixels. For watching, this is invisible. For working, it is not. A green-screen key builds its matte from colour, so at 4:2:0 the matte has a quarter of the resolution of the picture and every edge comes back blocky. Fine detail against a strongly coloured background — hair, a chain, text — smears. Heavy saturation changes in a grade expose the blocks.

This is the single most common reason a key that looked achievable on set is impossible in the edit.

Three: how many steps there are between black and white

8-bit gives 256 levels per channel. **10-bit** gives 1,024. That sounds like a small difference and it is not, because the levels are not spread evenly across what you see — they are spread across what the camera captured.

If the camera recorded a normal, contrasty picture, 8-bit is usually enough. If it recorded a **log** profile — flat and grey, holding a wide brightness range so you can redistribute it later — then you are stretching 256 levels across a much bigger range. Add the contrast back in the grade and the gaps between levels become visible as **banding**: stepped rings in a sky, a wall, a gradient behind a face.

The practical rule: log is a 10-bit feature. On an 8-bit phone or entry-level camera, a well-exposed standard profile beats log almost every time.

Four: how many bits per second

The capture bitrate is the budget the encoder had. A phone at 1080p might write 20 Mbps; the same phone at 4K writes 50 to 100; a mirrorless camera at 4:2:2 10-bit might write 400. Content matters more than resolution here — rain, foliage, confetti and fast camera moves exhaust a low bitrate long before a talking head does.

The tell for a starved bitrate is **macroblocking**: 16×16 squares appearing in the darkest and busiest parts of the frame. No amount of grading recovers it, and sharpening makes it worse.

Figure 1.2

Four decisions the camera made before you saw the file

Whole frames, or differences
Intra-frame scrubs instantly and is huge; long-GOP is small and expensive to decode

How much colour was kept
4:2:0 gives one colour sample per 2 by 2 block, which is what breaks a key

How many levels between black and white
256 per channel at 8-bit, 1,024 at 10-bit. Log stretches them thin

How many bits per second
About 20 Mbps on a phone at 1080p, 400 on a camera recording 4:2:2 10-bit

None of the four can be undone in the edit. One ffprobe command answers all four, once per camera rather than once per clip.

Read the file rather than guessing

FFmpeg is free, installs on every platform, and answers all four questions in one command:

```
ffprobe -v error -select_streams v:0 \
  -show_entries
stream=codec_name,pix_fmt,bit_rate,r_frame_rate,width,height \
  -of default=noprint_wrappers=1 clip.mp4
```

A typical phone answers `codec_name=hevc`, `pix_fmt=yuv420p`, `bit_rate=48000000`, `r_frame_rate=30/1`. The `p` in `yuv420p` means progressive; `yuv422p10le` would mean 4:2:2, 10-bit, little-endian. MediaInfo is a free graphical alternative if you would rather click.

Do this once per camera, not once per clip. The answer is a property of how the camera was set, and it will be the same for the whole shoot.

What this changes about your decisions

- Planning a key: check chroma subsampling **before** the shoot, not after.
- Planning a heavy grade: check bit depth, and turn log off if the camera is 8-bit.
- Planning archival: your master should be intra-frame, even though the camera original was not.
- Handed someone else's footage: run `ffprobe` first. It takes ten seconds and it tells you what the day is going to be like.

Today

Run the command above on the newest clip on your phone and the oldest clip you still have. Write down the four answers for each. The difference between them is usually the difference between two afternoons of work.

BEFORE YOU MOVE ON

An editor is handed 4K phone footage of a presenter in front of a green sheet. The lighting was even and the sheet is clean, but every attempt at a key produces blocky, stepped edges around the hair. What is the mechanism?

A. The bitrate is too low, so the encoder has discarded the fine detail around the hair; re-encoding the clip at a higher bitrate before keying will restore the edge.
B. Phone footage is 4:2:0, so the colour information the keyer builds its matte from exists at half the horizontal and half the vertical resolution of the picture.
C. The clip is long-GOP, so frames between keyframes are reconstructed approximations and the keyer sees a different edge on every frame.
D. The green sheet is the wrong shade for the keyer's default, and sampling a different point on it will fix the edges.

The answer and why: p. 444

Lesson 3 · 9 min

Frame rate, shutter, and the flicker in the light

THE ONE THING TO KEEP

Pick one frame rate from the mains frequency where you shoot, set every camera and the timeline to it, and keep shutter at roughly 180 degrees — because mismatched rates cost you frames at every conversion and mismatched shutter costs you flicker you cannot remove.

The number you are stuck with

Frame rate is the most consequential setting nobody checks. Get it wrong and every pan in the film judders slightly, forever, and you will not be able to say why.

The world runs on two families. Where mains electricity is 50 Hz — India, most of Europe, Africa, Australia — video is **25 and 50 fps**. Where it is 60 Hz — North America, Japan, parts of South America — it is **29.97 and 59.94**. Cinema is **24**, and its broadcast cousin is 23.976.

Those fractional numbers are not a mistake. When American television added colour in 1953, the colour subcarrier interfered with the audio at exactly 30 fps, so the whole system was slowed by one part in a thousand. Seventy years later every NTSC-region camera still records 29.97 frames per second, and every editor still has to think about it.

Why the mains frequency decides your frame rate

Most artificial light does not shine steadily. Fluorescent tubes, many LED bulbs and almost all cheap LED panels pulse with the mains supply — twice per cycle, so 100 times a second on a 50 Hz supply.

If your shutter interval does not divide evenly into that pulse, each frame catches a slightly different amount of light, and the footage **flickers** — a slow pulsing of brightness, or on a rolling-shutter sensor, horizontal bands crawling up the frame.

Shoot 25 or 50 fps on 50 Hz mains and the arithmetic works. Shoot 30 fps under 50 Hz lighting and it does not: you get a visible beat. This is why an Indian creator shooting at 30 fps indoors sometimes gets bands and cannot reproduce them outdoors. It is not the camera.

Figure 1.3

Frame rate against the mains frequency where you are shooting

	Shooting 25 or 50 fps	Shooting 30 or 60 fps
Mains supply is 50 Hz	**Clean** The shutter interval divides the 100 Hz pulse of the light	**Flicker** A beat indoors, and nothing at all outdoors
Mains supply is 60 Hz	**Flicker** A beat indoors. Set the shutter to 1/60 or 1/120 to cancel it	**Clean** The shutter interval divides the 120 Hz pulse of the light

Most cheap LED and fluorescent light pulses twice per mains cycle. This is why an Indian creator shooting 30 fps gets bands indoors and cannot reproduce them outside.

Where you cannot change the frame rate, change the shutter. A shutter speed that is a whole multiple of the flicker period — 1/50 or 1/100 on 50 Hz mains, 1/60 or 1/120 on 60 Hz — cancels it. Some cameras have a flicker-reduction or anti-flicker setting that does this for you.

Shutter angle, and why 180 degrees

Shutter speed controls how long each frame is exposed. Expressed as an **angle**, it is the fraction of the frame interval the shutter is open: 180° means open for half the interval.

At 25 fps, 180° is 1/50 s. At 50 fps, it is 1/100 s.

This is the default for one reason: it produces about the amount of motion blur people are used to from a century of cinema. Go much faster — 1/500 s on a 25 fps shoot, a 18° shutter — and each frame is sharp, so motion becomes a sequence of crisp stills and pans stutter. That is the *Saving Private Ryan* look, and it is a choice, not an accident. Go much slower and movement smears.

The editing consequence: sharp-shutter footage cuts badly against normal footage in the same scene, and it is the footage that resists slow motion the worst, because there is nothing to blend between frames.

Mixing frame rates on one timeline

Your timeline has one frame rate. Everything that is not already at it has to be converted, and every conversion is a compromise.

- **30 fps clip on a 25 fps timeline**: five frames a second have to go. The editor either drops them, producing a small hitch on every pan, or blends pairs, producing a soft ghost.
- **25 on 30**: five frames a second have to be invented, by duplication or blending.
- **50 or 60 on 25 or 30**: this one is clean and useful. Every second frame is used and the rest discarded, or — better — you interpret the clip at half speed and get genuine, real-frame slow motion.

The rule that avoids all of it: **decide the delivery frame rate before the shoot and set every camera to it.** Where you are handed mixed rates, set the timeline to whichever rate the majority of your picture is, and accept the conversion on the minority.

Choosing your project's number

- Uploading to a platform, anywhere: 25 or 30, whichever your region and your lights agree with. Nobody will notice which.
- Wanting slow motion: shoot 50 or 60 and cut on a 25 or 30 timeline.
- Delivering to a broadcaster or festival: they will specify, and they will reject a file that does not match.
- Screen recordings: capture at your timeline's rate. A 60 fps screen capture on a 30 fps timeline makes cursor movement stutter.

In Resolve, the timeline frame rate locks once the first clip lands in a project. To change it you create a new timeline, or change the project setting before importing anything. Kdenlive and Shotcut set it per project in the same way. This is the most common reason somebody's project is at the wrong rate: they never chose, and the first clip chose for them.

Today

Check the frame rate of the last three things you filmed. If they differ, that explains something you have already seen.

BEFORE YOU MOVE ON

A creator in Delhi shoots indoors at 30 fps under LED panels and gets horizontal bands crawling up the frame. The same camera outdoors in daylight is clean. What is happening, and what fixes it without a new camera?

A. The sensor is overheating under continuous indoor recording; shorter takes will stop the banding.
B. The LED panels are too dim, so the camera has raised gain and the bands are amplified noise; more light removes them.
C. The camera's rolling shutter is scanning the sensor far too slowly for the indoor movement in the frame, so each row of the image is recorded at a slightly different moment; stabilisation in post will straighten it out.
D. The LEDs pulse at 100 Hz on 50 Hz mains, which does not divide evenly into a 30 fps frame interval; shooting 25 fps or setting shutter to 1/50 or 1/100 cancels it.

The answer and why: p. 444

Lesson 4 · 8 min

Resolution, aspect ratio, and having room to move

THE ONE THING TO KEEP

Shoot larger than you deliver so the edit has room to punch in, stabilise and reframe, set one image-scaling behaviour at the start of the project, and keep everything that must be read inside 90 per cent of the frame.

Resolution is a budget, not a quality

People talk about 4K as though it were a grade of picture. It is a count of pixels: 3840×2160 for UHD, 4096×2160 for DCI. A badly exposed, softly focused 4K shot is worse than a good 1080p one, and the platform is going to compress both.

What 4K actually buys you, in an edit, is **room**. Deliver at 1080p from a 4K source and you can:

- Punch in to 200% and still be at full delivery resolution, which manufactures a second camera angle out of one.
- Reframe a badly composed shot after the fact.
- Stabilise, which costs you a crop of 5 to 15%.
- Crop a 16:9 shot to 9:16 vertical without going soft.

That is the honest case for shooting 4K and delivering 1080p, and it is a strong one. The cost is storage, heat, and a heavier timeline.

The frame shapes you will actually meet

- **16:9** — 1920×1080 or 3840×2160. Default for anything that lands on a television or a desktop browser.
- **9:16** — 1080×1920. Vertical short-form. Not a crop of 16:9 that happens to be tall; a different composition.

- **1:1** — 1080×1080. Feed posts that have to work either way up.
- **4:5** — 1080×1350. Taller than square, still supported in some feeds, and the largest area a feed post can occupy on a phone in several layouts.
- **2.39:1** — cinema widescreen, usually achieved by letterboxing a 16:9 frame.

A widescreen look on a 16:9 delivery is black bars *inside* the picture. That is a stylistic choice that costs a quarter of your pixels, and on a phone in a feed it makes an already small picture smaller. Use it when the piece is genuinely cinematic; do not use it as decoration.

Pixel aspect ratio, briefly, because it still catches people

Most formats you will meet have square pixels: PAR 1.0. Some older standard-definition and HDV material does not — DV PAL is 720×576 with non-square pixels displayed as 16:9 or 4:3. Drop that into a square-pixel timeline without telling the editor, and everybody is thin or fat.

If a clip looks stretched and the numbers are old, check the clip's pixel aspect ratio attribute. In Resolve it is in Clip Attributes; in Kdenlive, clip properties. Set it, do not scale it by eye.

Scaling: which way is cheap

Scaling **down** discards information, and a good resampler does it well. Scaling **up** invents information, and every method is a guess — bicubic guesses smoothly, machine-learning upscalers guess plausibly, and both will produce something that was not in the original.

The practical consequences:

- Putting a 1080p clip on a 4K timeline means it is being upscaled 200% for the whole film. If most of your material is 1080p, work at 1080p.
- Putting a 4K clip on a 1080p timeline is free quality. It is the normal way to work.
- Mixing both: set the timeline to your delivery size, and let the 4K clips carry the punch-ins.

In Resolve, how mismatched clips behave is set once in Project Settings under Image Scaling: **Scale to Fit** (whole frame visible, bars if the shape differs), **Scale to Fill** (fills the frame, crops the excess), **Centre Crop**. Choose it deliberately at the start, because changing it later re-frames every mismatched shot in the project.

Designing for the safe area

The outer band of your frame is not reliably visible. Platforms crop, televisions overscan, and vertical feeds put interface elements over the top and bottom of the picture.

Working margins:

- **Title safe**: keep all text within 90% of the frame, which is 5% in from every edge.
- **Vertical short-form**: the bottom 15 to 20% carries the caption, username and buttons, and the top 10% the status bar and platform chrome. Anything that must be read goes in the middle third.
- **Any 16:9 video that might be cropped to square**: keep the subject within the central square.

Every editor can show a guide overlay for these. Turn it on and leave it on.

One project, several shapes

When the same material has to go out horizontal and vertical, do not re-edit. Finish the 16:9 cut, then duplicate the timeline, change its resolution to 1080×1920, and reframe each shot — which is exactly the room a 4K source was bought for. Resolve, Premiere and CapCut all have automatic subject-tracking reframe tools; treat their output as a first pass and fix the shots where the subject is not what the tracker locked onto.

Today

Take a 16:9 shot you like, crop it to 9:16, and see how much of the composition survives. That gap is what you now shoot for.

BEFORE YOU MOVE ON

An editor has a project where 80% of the material is 1080p and 20% is 4K. They set the timeline to 4K so as not to waste the 4K clips. What have they actually done?

A. Made a reasonable trade: the 4K clips stay pristine, and the 1080p clips are no worse than they would have been on a 1080p timeline.
B. Committed the majority of the film to a 200% upscale, and taken on a heavier timeline, for the benefit of a fifth of the shots.
C. Ruined the 4K clips, because a timeline always resamples every clip to its working resolution regardless of match.
D. Made no difference to the picture, since the export will be 1080p in the end anyway.

The answer and why: p. 445

Lesson 5 · 8 min

The settings you cannot change later

THE ONE THING TO KEEP

Frame rate, image scaling and colour management are decided before the first import and are expensive to reverse, so configure a template project once and duplicate it rather than choosing settings under time pressure.

Ten minutes that save a week

Most project settings are harmless and adjustable. A handful are not: change them after you have started and you re-frame every shot, re-grade every scene, or lose sync across the whole timeline. This lesson is the short list, and it is worth doing as a ritual before importing anything.

The locked ones

Timeline frame rate. In DaVinci Resolve this is set in Project Settings → Master Settings, and it greys out as soon as media exists in the project. Not the timeline — the project. Import one clip and the decision is made. If you need to change it, the honest routes are: start a new project at the right rate and import the old timeline, or create a new timeline at the right rate inside the same project, which leaves you converting the old one anyway.

Timeline resolution. Changeable, but changing it re-frames every clip whose scaling was set relative to the old frame. If you have already punched in on forty shots, all forty move.

Image scaling behaviour. Covered in the previous lesson. Set it once.

Colour science and colour management. In Resolve, the choice between DaVinci YRGB (no management — what you see is what the clip is) and DaVinci YRGB Color Managed (everything converted into a working space and out to a timeline space) changes how every clip looks. Switching it mid-grade means every grade you have built was made on a different starting image.

For most people starting out, the honest recommendation is: leave colour management **off**, set your timeline colour space to Rec.709 Gamma 2.4, and handle log clips individually with an input transform. It is one more step per log clip and it is far easier to reason about than a managed pipeline you have not yet understood.

The ones that quietly cost you

Audio sample rate. 48 kHz is the video standard. If your project is at 48 kHz and a clip is 44.1, the editor resamples, usually correctly. If your project is at 44.1 because some music file set it, every piece of camera audio is being resampled instead. Set it to 48000 and leave it.

Optimised media and proxy resolution. Quarter resolution is comfortable on most machines; half if you need to judge focus. Setting it to full defeats the purpose.

Cache location. Resolve writes render cache, optimised media and proxies to a folder you choose. The default is often the system drive. Point it at your working drive with the footage, or you will fill your operating system partition and find out during an export.

Auto-save and backup interval. On, and set to ten minutes. In Resolve this is in Preferences → User → Project Save and Load, with Live Save and Project Backups as separate checkboxes. Turn both on, and note the backup folder path somewhere you will find it at 11pm.

Start from a template project

The reliable way to never get this wrong is to set it once and copy it. Make a project called `_TEMPLATE_1080p25`, configure everything above, create your standard bins — `01_footage`, `02_audio`, `03_music`, `04_graphics`, `05_exports` — add your title styles and caption styles, save your export presets, and then duplicate it at the start of every job.

In Resolve, right-click the project in the Project Manager and choose Duplicate. In Kdenlive, save a project template file. In Premiere, save a project as a template and copy the file. Twenty minutes, once.

A worked order of operations

1. Decide delivery: resolution, frame rate, aspect. Write it down.
2. Duplicate the template project, or create one and set: frame rate, resolution, image scaling, audio 48 kHz, colour space Rec.709 Gamma 2.4.
3. Point cache and proxy paths at the working drive.
4. Confirm auto-save and backups are on and note the path.
5. Create bins.
6. *Then* import.

The order matters because step 2 is only fully available before step 6.

What a template cannot save you from

It cannot tell you what the client's delivery spec is. It cannot know that this particular shoot was at 50 fps for slow motion. And it cannot stop you from opening last month's project, editing in it because it was already open, and exporting a video at last month's resolution — which is common enough that the first line of your quality-control checklist should be reading the export's actual dimensions back off the file.

Today

Open your editor, build a template project, and duplicate it. The next project you start should take thirty seconds to set up.

BEFORE YOU MOVE ON

Halfway through grading a documentary in Resolve, an editor switches the project from DaVinci YRGB to DaVinci YRGB Color Managed because a tutorial recommended it. What is the cost?

A. Nothing but a re-render of the cache, since colour management is a display setting and the underlying grades are untouched.
B. Every existing grade was built on a differently transformed starting image, so the grades are still applied but are now landing on inputs they were not made for.
C. The grades are deleted, because switching colour science clears the node graph on every clip.
D. Only the log clips in the project are affected, because managed mode simply applies the correct input transform to log material and otherwise leaves ordinary Rec.709 clips exactly as they were.

The answer and why: p. 445

Lesson 6 · 8 min

Copying a card so you know the copy is good

THE ONE THING TO KEEP

A drag-and-drop copy reports success without reading anything back, so verify irreplaceable footage with a checksum manifest at offload and keep the material in two places on two devices before any card is formatted.

The failure that has no warning

A card copies. Every file appears. The folder sizes match. Three weeks later you open shot 47 and the last four seconds are green blocks, or the file will not open at all.

A plain drag-and-drop copy does not verify anything. The operating system writes bytes and reports success; it does not read them back. Most of the time this is fine. Occasionally a cable, a card reader, a failing card or a full disk produces a file that exists at the right size and contains the wrong bytes. You find out when you need it.

The fix is a **checksum**: compute a fingerprint of the source file, compute one of the copy, compare. If they match, the copy is byte-identical. This is what every broadcast and film production does at the card, and it takes one extra command.

Doing it free, on any machine

On macOS or Linux, `rsync` will copy and verify in one step:

```
rsync -aH --checksum --progress \
  /Volumes/CARD_01/ ~/Projects/2026-03-
12_ClinicFilm/01_footage/CARD_01/
```

`--checksum` makes rsync compare file contents rather than size and timestamp. It is slower, and slow is the point.

To produce a record you can check later, write a manifest:

```
cd ~/Projects/2026-03-12_ClinicFilm/01_footage/CARD_01
find . -type f -exec shasum -a 256 {} \; > ../CARD_01.sha256
```

And months later, verify the whole folder against it:

```
shasum -a 256 -c CARD_01.sha256
```

Every file prints `OK` or `FAILED`. On Windows, `certutil -hashfile <file> SHA256` does one file, and PowerShell's `Get-FileHash` handles folders in a loop. TeraCopy is free for personal use and verifies copies with a checkbox.

Dedicated offload tools exist — Hedge, Silverstack, ShotPut Pro — and are worth their money on a production with a runner doing this all day. For one person with a laptop, `rsync --checksum` does the same job for nothing.

Copy the whole card structure, not just the clips

Dig into a card and it is tempting to grab only the `.MP4` files. Do not.

Many cameras store metadata, audio, spanned-clip indexes and colour information in sibling folders. AVCHD splits long recordings across multiple files that only reassemble via a playlist file. Sony's XAVC, Canon's structure and Panasonic's all carry sidecar data. Copy the **entire card**, folder structure intact, and keep it as an untouched original. Then work from a copy of that.

Name the folder for the card and the day: `2026-03-12_CARD_01_A7III`. Never reuse a card name within a project, even across days, because two folders called `CARD_01` will end up merged by somebody eventually.

Two copies before the card is formatted

The rule everybody agrees on and half of people break under time pressure: **the card is not formatted until the material exists in two places on two different devices.**

A working drive plus a laptop's internal disk counts. A working drive plus the same drive's second partition does not — one drive failure takes both.

If you can afford the cards, keep them unformatted until the project delivers. It is the cheapest insurance in this course, and the only one that protects you from the mistake where somebody deletes the wrong folder.

A log for the shoot

A plain text file in `06_docs` that lists, per card: date, camera, what is on it, the checksum manifest filename, where the two copies live, and whether the card has been formatted. It takes two minutes at the end of a shoot day and it is the document that answers, in March, the question of whether the missing interview was ever copied at all.

Where this is genuinely overkill

If you filmed a two-minute vertical video on your phone and it is already in your phone's cloud backup, do not build a manifest for it. Checksums earn their cost when the material is irreplaceable and large: an event that happened once, an interview with somebody who has now moved country, a client shoot you cannot repeat.

The honest test is the one question: *if this folder were gone tomorrow, could it be re-shot?* If yes, copy it normally. If no, verify it.

Today

Take the most irreplaceable folder of footage you own, generate a SHA-256 manifest for it, and store the manifest somewhere separate from the footage. You have just made every future copy of that folder checkable.

BEFORE YOU MOVE ON

A shooter copies a card by dragging it to an external drive, confirms the folder sizes match exactly, then formats the card to keep shooting. What risk remains that the size check did not address?

A. Bytes may have been corrupted in transit while the file length stayed correct, and there is now only one copy, so nothing can be compared against.
B. None in practice — if every file is present at the correct size and the folder totals agree exactly, then the copy is complete and the card can safely be formatted and taken back out to the shoot.
C. The drive might not have finished flushing its write cache, so waiting a few minutes before formatting would have made the copy safe.
D. The camera's sidecar metadata folders were probably skipped, which is the only real risk in a size-matched copy.

The answer and why: p. 446

Lesson 7 · 9 min

Watching an hour so you can find any second of it

THE ONE THING TO KEEP

Watch everything once before cutting anything, marking moments with written notes and building a selects timeline, because editing is a search problem and you cannot search material you never formed a memory of.

The afternoon that decides the week

The instinct on day one is to start cutting. Resist it for one session. Watch everything, once, at normal speed, marking as you go. On a project with an hour of material this takes an afternoon, and it is the afternoon that turns a five-day edit into a two-day one.

The reason is not diligence. It is that an edit is a search problem. Every decision you make in the timeline is really the question *what should be here instead?* — and you cannot answer it about material you cannot recall. Editors who skip the watch spend their week scrubbing, and scrubbing is both slow and unreliable, because a shot's value is usually in something you cannot see at 8× speed.

Markers, and a code you will still understand later

Every editor supports markers on clips and on the timeline, with a colour and a note. Resolve is `M` to place, `M` again immediately to open the note field. Kdenlive, Shotcut and Premiere are all similar.

Agree a colour code with yourself and write it in your project notes. One that works:

- **Green** — a take you will definitely use.
- **Blue** — a usable alternative.
- **Yellow** — something said that matters to structure, even if this take is bad.
- **Red** — a problem: clipped audio, a focus miss, a wrong fact, a face that has not consented.
- **Purple** — b-roll worth remembering, with a one-word subject.

The note matters more than the colour. `best answer re cost` is findable. `good` is not.

Subclips and selects timelines

Two mechanisms turn markers into something you can edit from.

A **subclip** is a named fragment of a longer clip that behaves like its own clip in the bin. Mark in and out, make a subclip, name it for its content. An hour-long interview becomes twenty subclips with names like `origin story`, `the mistake`, `what she would tell a beginner`.

A **selects timeline** — also called a string-out — is a timeline containing only the good bits, in the order they were shot, with nothing removed but the dross. You build it once and then edit *from* it rather than from the bins. In practice this is how most documentary and interview work is actually done: the first cut is a re-ordering of the selects.

Keywords and smart bins

Where markers describe moments, keywords describe clips. Resolve's Metadata panel has a Keywords field; anything you type there is searchable, and a **Smart Bin** is a saved search that stays current.

A smart bin for `Keyword contains interview AND Flag is green` gathers every good interview clip automatically, including the ones you keyword tomorrow. On a project of 300 clips this is the difference between a bin structure that helps and one that is another thing to maintain.

The minimum useful metadata on a documentary-style project: scene or subject, person, location, and a good/alt/no flag. Four fields, typed as you watch.

The transcript is metadata too

For anything with speech, generate a transcript during the watch and keep it open beside the timeline. It makes the material text-searchable, which is a different and much faster kind of finding than scrubbing. The next lesson but one builds a whole editing method on this; for now it is enough that `Ctrl+F` on a transcript beats twenty minutes of scrubbing.

What to write down that is not in the footage

During the watch, keep a second document — a plain text file — and note:

- Every factual claim that will need checking.
- Every name and its spelling, for the supers later.
- Every place where two takes contradict each other.

- Every moment you had an emotional reaction, with its timecode. These are the moments that end up in the cut, and you will not remember them tomorrow.
- Every shot that is missing. This is the list that gets you a pickup shoot while the location is still available.

Where this is too much

A thirty-second vertical video from a single take does not need a selects timeline. The threshold is roughly: **more material than you can hold in your head**, which for most people is about fifteen minutes of footage or about eight clips. Below that, cut. Above it, log first.

The honest limitation of logging is that it front-loads the boredom. There is no way to make watching an hour of mediocre footage interesting, and plenty of editors skip it and get away with it on short projects. On anything longer than five minutes of output, they do not.

Today

Take the longest set of footage you have, watch fifteen minutes of it at normal speed, and place a marker with a written note at every moment you reacted. Count them. That number is roughly how many usable beats you have per fifteen minutes, and it is the number that tells you whether you have enough material.

BEFORE YOU MOVE ON

Two editors have the same three-hour interview. One builds a selects timeline of forty minutes before cutting; the other starts assembling directly from the bins. Why does the first usually finish sooner, even after losing a day?

A. Because the selects timeline is shorter, so the editor has less material to render and the machine responds faster.
B. Because a selects timeline removes the bad takes permanently, so there are fewer decisions to make later.
C. Because every timeline decision is really a question about what else exists, and the second editor has to answer it by scrubbing rather than by remembering.
D. Because assembling directly from the bins produces far more offline media errors, since each clip is referenced individually rather than through the single container a selects timeline provides.

The answer and why: p. 446

Lesson 8 · 9 min

Two cameras, one microphone, one sync point

THE ONE THING TO KEEP

Give every take one sharp sync point at the start, use waveform sync as a first pass and check it at a hard consonant, and convert variable-frame-rate phone footage to constant rate at offload because drift is usually the file, not the clocks.

Why sound and picture arrive separately

Camera microphones are close to the camera and far from the speaker, which is the wrong way round. So the good recording usually comes from somewhere else — a lavalier into a phone or recorder in a pocket, a shotgun on a boom into a field recorder, a second camera in another position.

That gives you two or more independent recordings of the same event, each started at a different moment, each running on its own quartz clock. Your job at the start of the edit is to make them agree.

The clap, and why it is still the best method

A single sharp sound, in frame, at the top of every take. A clapperboard if you have one, two hands if you do not.

It works because it produces one unmistakable vertical spike in every waveform and one identifiable frame in every picture. You line the spikes up and you are done. It takes fifteen seconds on set and saves half an hour per take in the edit.

Three details people get wrong:

- Clap **in shot, in focus, and audible on every device**. A clap heard by the recorder but not the camera syncs nothing.
- Clap at the **start** of the take. A clap at the end works, but you cannot use it until the take is over, and someone will stop recording early.
- Clap again if anything stops. Every restart is a new sync problem.

Waveform sync, and when it fails

Every modern editor can align clips by comparing their audio waveforms. Resolve does it from the timeline or the media pool; Premiere has Synchronize; PluralEyes does it as a paid specialist.

It works by cross-correlation: sliding one waveform against the other and finding the offset with the highest match. It is accurate to the sample when it works, and it fails in four situations that are all common.

- **No shared sound.** A camera 30 metres away in wind shares almost nothing with a lavalier.
- **Very quiet or very uniform audio.** Steady room hum correlates with itself at every offset, so the peak is ambiguous.

- **Non-overlapping recordings.** If the recorder ran continuously and the camera recorded four separate takes, the tool has to find four separate offsets in one long file — most will, some will not.
- **Different sample rates or variable frame rate.** Which is the next section.

The workflow that works: try automatic sync, then check every take by eye at a hard consonant. A `t` or `k` sound has a visible attack, and a one-frame error is visible on a close-up of a mouth.

Drift: when sync is right at the start and wrong at the end

You sync the clap perfectly. Ten minutes later the lips are half a second ahead. Nothing slipped; the two devices simply counted time at slightly different rates.

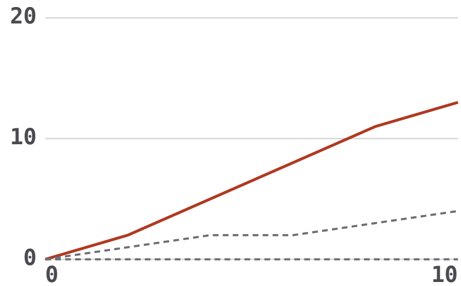
Two causes, with different fixes.

Clock drift. Two independent crystals are never exactly equal. Over ten minutes a cheap recorder can gain or lose a few frames. Fix: stretch one clip's speed by the tiny ratio needed — if the audio is 4 frames late after 10 minutes at 25 fps, that is 4/15000, so set the audio clip's speed to about 99.973%. Resolve lets you type a duration and conforms the clip to it.

Variable frame rate. Phones and screen recorders often write VFR: the nominal rate is 30 fps but the actual rate wanders with light and processing load. The editor assumes a constant rate, so picture and sound diverge steadily. This is by far the more common cause with phone footage. Fix it before importing, by transcoding to constant frame rate:

Figure 1.4

Two causes of drift, across one ten-minute take



Across: Minutes into the take

Up: Frames out of sync at 25 fps

—— Phone file at variable frame rate

– – Cheap recorder, clock drift

– – Both conformed at offload

The clap was right. Thirteen frames is half a second of lip sync. Transcode the phone file to constant rate at offload and the top line becomes the bottom one.

```
ffmpeg -i phone.mp4 -vsync cfr -r 30 -c:v libx264 -crf 18 \
       -c:a aac -b:a 320k phone_cfr.mp4
```

Do this to the whole shoot at offload and the problem disappears for the rest of the project.

Timecode, for when there are more than two devices

Professional sets solve all of this with timecode: every device stamps each frame with a time of day from a shared reference, and the editor aligns by number rather than by sound. Timecode generators — Tentacle Sync and similar — cost real money.

The free approximation: set every device's internal clock to the same second before the shoot, and use file creation time as a coarse alignment before waveform sync refines it. It gets you into the right minute, which is enough for the automatic tools to find the right second.

Multicam, practically

Once clips are in sync, group them into a multicam clip — Resolve calls it a Multicam Clip, Premiere a Multi-Camera Source Sequence, Kdenlive does it through its own mechanism. You then play the timeline and cut between angles live by pressing 1, 2, 3, and refine afterwards.

Two habits make the result watchable rather than frantic. Set the audio to come from one source — the good microphone — for the whole multicam clip, so switching angles never switches sound. And give each angle at least three seconds before switching away, unless the cut is on an action; a multicam cut every second reads as a person who could not decide.

Today

Record ten seconds on your phone and ten on any second device, clap once at the start, and sync them by hand. Then let the automatic tool do it and compare. You will trust it appropriately after that.

BEFORE YOU MOVE ON

An interview is shot on a phone with a separate audio recorder. The clap syncs perfectly, but by minute eight the lips lead the sound by about half a second. Which explanation should be checked first, and why?

A. The recorder's clock drifted; speed-stretching the audio by a fraction of a percent will realign it.
B. The phone recorded variable frame rate, so the editor's constant-rate assumption makes picture and sound diverge steadily.
C. The audio was recorded at 44.1 kHz in a 48 kHz project, so it plays back at the wrong speed.
D. The waveform sync tool locked onto the wrong correlation peak, and re-running it with a longer analysis window will fix it.

The answer and why: p. 447

Lesson 9 · 9 min

Editing the words before you touch a picture

THE ONE THING TO KEEP

For speech-led work, transcribe everything and build the structure in text with timecodes attached, because reading is four times faster than listening and rearranging a paragraph costs a keystroke where rearranging a timeline costs a minute.

The method that changed interview editing

For anything built on speech — an interview, a podcast with cameras, a vlog, a lecture, a tutorial — the fastest route to a structure is to stop looking at pictures. Transcribe everything, edit the transcript like a document, and only then go to the timeline.

This is not a shortcut invented by software. Documentary editors have done it with paper transcripts and scissors since the 1960s. What changed is that transcription is now free and instant, and several editors can cut the timeline directly from the text.

Why text is faster than video for this

Three reasons, and they compound.

- **Reading is roughly four times faster than listening.** A 40-minute interview is a 15-minute read. You can hold the whole thing in your head at once, which you cannot do at playback speed.
- **Text is searchable.** Finding the sentence about cost takes two seconds, not ten minutes of scrubbing.
- **Rearranging text is free.** Moving a paragraph costs a keystroke. Moving a section of timeline costs a minute and risks breaking the audio underneath.

The cost is that you are working blind to performance. The best-worded answer is sometimes not the best-delivered one. So the transcript decides the *structure*, and the timeline decides the *takes*.

Getting a transcript for nothing

Whisper, OpenAI's open-weights speech model, is the workhorse. `whisper.cpp` runs it on an ordinary laptop CPU with no GPU and nothing leaving the machine, which matters when the audio is confidential.

```
# convert to 16 kHz mono, then transcribe with word timings
ffmpeg -i interview.mov -ar 16000 -ac 1 interview.wav
./main -m models/ggml-medium.bin -f interview.wav \
       -l en --output-srt --max-len 42
```

The `medium` model is a reasonable accuracy-for-time trade on a laptop; `small` is about three times faster and noticeably worse on accented speech.

Other free routes: YouTube's automatic captions, by uploading unlisted and downloading the track; Resolve's built-in transcription on versions that have it; Kdenlive's speech-to-text, which uses Whisper or Vosk underneath.

Accuracy caveat, and it matters here more than for captions. Recognition degrades on accented English, code-switching mid-sentence, proper nouns and technical vocabulary — which is most of the speech this readership works with. For a caption track that error is a wrong word on screen. For a paper edit it is worse: you may cut out the sentence you needed because the transcript says something else. Always play the audio of any line you are about to cut on the strength of its text.

The paper edit itself

1. Get a transcript with timecodes. Paste it into any document editor.
2. Read it once, end to end, marking nothing.
3. Read it again, deleting everything you are certain you will not use. Do not agonise; you are removing the obvious 60%.

4. What is left, cut into blocks and group by subject, not by the order they were spoken.
5. Order the blocks into an argument. Say the argument out loud in one sentence. If you cannot, the structure is not finished.
6. Print or export the result with its timecodes. That is your assembly plan.

The discipline that makes this work: keep the timecodes attached to every line, all the way through. A paper edit without timecodes is a lovely document you cannot execute.

Text-based editing inside the editor

Several editors now do this without leaving the timeline. Resolve 19 and later has a Text Based Editing panel: transcribe a clip, and deleting a sentence in the text pane deletes it from the source, with the selected text appended to the timeline. Premiere has the same feature. Descript is built entirely around it.

The mechanic is genuinely fast for assembling a first cut of a talking head. Two honest limits. Word-level trim points land on word boundaries in the transcript, which are approximate — you will still be trimming by frame afterwards to remove breath and lip smack. And the picture is being cut wherever the words are cut, which produces a wall of jump cuts that the next block's techniques exist to handle.

The radio edit

The result of all this is what editors call a **radio edit**: the entire piece cut for sound only, with the pictures ignored or left as whatever happened to be under the words. It should be listenable end to end with the screen off, and it should make its argument.

Get the radio edit right and the picture edit becomes a covering job — choosing what the viewer looks at while they listen — which is a far easier problem than building both at once. Get it wrong and no amount of b-roll rescues it.

What this method is bad at

Anything where the picture carries the meaning. A dance piece, an action sequence, a silent montage, a music video, a process film where somebody's hands are the story. Transcript-first editing is a tool for speech-led work, and applying it to picture-led work produces something that sounds coherent and looks like nothing.

Today

Transcribe any five-minute recording you own and cut the transcript down to two minutes in a text editor, keeping the timecodes. Time yourself. Then estimate how long the same reduction would have taken in a timeline.

BEFORE YOU MOVE ON

An editor builds a paper edit from an automatic transcript of an accented interview and cuts the timeline straight from it. What is the specific risk that a caption workflow would not have?

A. The transcript's timecodes will be slightly off, so every cut lands a few frames late and the piece feels sluggish.
B. Recognition errors are confident and grammatical, so a misheard sentence can cause the editor to discard material that actually said what they needed.
C. Automatic transcripts omit filler words, so the cut will be too tight and the speaker will sound rushed.
D. The model will quietly normalise the speaker's dialect and idiom into standard written English, which changes the tone of the piece and misrepresents how the person actually speaks.

The answer and why: p. 447

Lesson 10 · 9 min

When the footage arrives broken

THE ONE THING TO KEEP

Most broken footage falls into four fixable classes — variable frame rate, an unsupported codec, a missing index, and a mis-mapped audio channel — and the ones that are genuinely unrecoverable are clipped highlights, clipped audio and missed focus, which are worth naming to a client on day one.

The day that starts badly

Sometimes the material is wrong before you have made a single decision. A file will not import. A clip plays but has no audio. Everything is in sync for a minute and then is not. A card reads as empty. This lesson is the triage list, in the order worth trying, and the honest note about which of these have no fix.

Variable frame rate

The most common problem in amateur footage, and the one that looks like something else. Phones, screen recorders, OBS captures and most game capture write **variable frame rate**: the file claims 30 fps but the actual interval between frames wanders.

The symptom is drift — sync is fine at the start and progressively wrong. Diagnose it:

```
ffprobe -v error -select_streams v:0 -show_entries \
  stream=r_frame_rate,avg_frame_rate -of csv clip.mp4
```

If `r_frame_rate` and `avg_frame_rate` differ — `30/1` against `29893/1000` — it is variable. Convert the whole shoot at offload:

```
ffmpeg -i clip.mp4 -vsync cfr -r 30 -c:v libx264 -crf 18 \
       -preset medium -c:a aac -b:a 320k clip_cfr.mp4
```

CRF 18 is visually near-lossless for H.264. HandBrake does the same job with a graphical interface and a Constant Framerate checkbox, and is free on every platform.

A file that will not import

In order:

1. **Play it in VLC.** Free, and it plays things editors refuse. If VLC plays it, the file is fine and your editor lacks the codec.
2. **Check the codec** with `ffprobe`. Editors commonly refuse 10-bit HEVC, some variants of AV1, and anything with an unusual pixel format.
3. **Transcode to an intermediate.** This is the universal answer:

```
ffmpeg -i awkward.mp4 -c:v dnxhd -profile:v dnxhr_hq \
       -pix_fmt yuv422p -c:a pcm_s16le converted.mov
```

DNxHR HQ is large and every editor opens it. ProRes 422 is the macOS equivalent.

4. **If VLC also refuses it**, the file is damaged. Go to the next section.

A damaged or truncated file

A recording interrupted by a flat battery or a card pull often has no index — the data is there but the header that says where everything is was never written. Some of these are recoverable.

- `untrunc` is a free open-source tool that rebuilds a broken MP4 or MOV using a *working* file from the same camera with the same settings as a reference. It works surprisingly often.
- Some cameras write a repair-on-next-power-up routine; putting the card back in and turning the camera on is worth trying before anything else.

- `ffmpeg -err_detect ignore_err -i broken.mp4 -c copy out.mp4` sometimes extracts what is readable.

If the card itself reads as empty or unformatted, **stop writing to it immediately** and run a photo-recovery tool — PhotoRec is free, open source and cross-platform — which scans for file signatures rather than trusting the file table. Recovered clips arrive without names, in one pile, which is unpleasant but is better than nothing.

No audio, or audio on one side only

A lavalier plugged into a single camera input lands on one channel. The clip sounds fine on speakers and is silent in one earbud.

The fix is in the clip's audio configuration, not in a pan control: set the clip to **mono** and choose the channel that has signal. In Resolve this is Clip Attributes → Audio, setting the track format to Mono and selecting Source Channel 1. Doing it with a pan knob instead leaves you at half level and having to compensate.

No audio at all usually means the clip has an audio stream your editor cannot decode — some cameras write PCM in unusual layouts — in which case extract it and reattach:

```
ffmpeg -i clip.mov -vn -c:a pcm_s16le clip_audio.wav
```

Rolling shutter and other unfixables

Some problems are in the pixels and no tool removes them cleanly.

- **Clipped highlights.** Once a channel hit maximum, there is no detail to recover. Grading can roll the shoulder off so it is less offensive; it cannot invent the cloud.
- **Clipped audio.** Declipping tools interpolate the flattened peaks and can improve mild cases. Heavy clipping is destroyed.
- **Missed focus.** Sharpening raises acutance at the edges and raises noise everywhere. It does not restore detail.

- **Rolling shutter skew.** Stabilisers and dedicated rolling-shutter tools reduce it. Extreme cases — a propeller, a fast whip pan — stay bent.

Saying this clearly to a client on day one is far better than discovering it on delivery day. "That shot cannot be saved, here is what we do instead" is a professional answer.

A triage habit

Before you cut anything, play every clip through once at speed and watch for: black frames, sudden silence, sync slip, corrupted blocks. Fifteen minutes now. The alternative is finding the corruption at minute 43 of a fine cut, when the fix is expensive and the reshoot window has closed.

Today

Run the `r_frame_rate` against `avg_frame_rate` check on a video from your phone. If they differ, you have just found the cause of a sync problem you have probably already had.

BEFORE YOU MOVE ON

A clip imports and plays, but at minute six the audio is clearly ahead of the picture, and the gap grew steadily from zero. `ffprobe` reports `r_frame_rate=30/1` and `avg_frame_rate=29897/1000`. What does that pair of numbers tell you?

A. The file is damaged and needs its index rebuilt before it can be trusted.

B. The clip was recorded at 29.97 fps and is being interpreted by the editor as a flat 30, so it should be conformed to the correct rate rather than transcoded to a new file.

C. The nominal and actual rates disagree, so the file is variable frame rate and the editor's constant-rate assumption is accumulating error.

D. The audio stream is at the wrong sample rate, and the video figures are a red herring.

The answer and why: p. 448

CASE STUDY · MODULE 1

Two cards, three days, and the format button

An illustrative case, built from documented patterns.

A two-person video team filming a three-day teacher-training programme for a coaching centre in Kota, on a fixed fee of thirty-two thousand rupees.

Ritu shoots and her brother Ankit handles sound and the laptop. They own one mirrorless camera, one lavalier into a phone, and two 64 GB cards. The brief is a twelve-minute film plus six short clips for the centre's social feeds. Three days of sessions, four teachers interviewed between them, and one closing address by the centre's founder that happens once.

On the morning of day one the arithmetic is already tight. Sixty-four gigabytes at the camera's 4K bitrate is a little over an hour and a half of recording. A full day of sessions, interviews and cutaways is four to five hours. Both cards will be full by the afternoon.

The laptop is a five-year-old machine with a 512 GB internal drive and one USB port that works reliably. They have a 2 TB portable hard disk. A straight drag-and-drop copy of one full card to the hard disk takes about twelve minutes over that port. Ankit knows, because he read about it the week before, that a drag-and-drop copy is not verified: the operating system writes the bytes and reports success without ever reading them back. Copying with `rsync --checksum` reads everything twice and takes closer to forty minutes a card.

So the decision arrives at lunchtime on day one, and it is not a technical question. It is a scheduling one.

Option one: verify, and lose the afternoon's first session. Forty minutes a card, twice a day, on the only laptop, which is also the machine Ankit uses to keep an eye on the audio recordings. The sessions do not pause for them. Whatever happens in the hall between two and half past two is not filmed.

Option two: copy fast and format. Twelve minutes a card, no verification, and they are back in the hall before the session restarts. If a file is silently bad they will find out in the edit, three weeks later, when the card has been formatted eight more times and the session cannot be repeated.

Option three: stop formatting. Buy four more 64 GB cards. In Kota in March that is roughly eleven hundred rupees each, so forty-four hundred rupees against a thirty-two thousand rupee fee. The cards stay unformatted until the film is delivered, which means every card is a second copy of itself, and the copy on the hard disk is a third. Nothing needs verifying, because nothing is being destroyed.

Ritu's objection to option three is straightforward and correct: the fee already has no margin in it, the camera is theirs, the drive is theirs, and forty-four hundred rupees is most of what they will clear on the job.

Ankit's objection to option two is that the founder's closing address happens at four o'clock on day three, it lasts nine minutes, and it is the only thing in the brief that cannot be re-created. If that file is the one that comes back as green blocks, there is no film.

They are also aware, as they argue about it on the stairs, that both of them have done option two on every job they have ever taken, and nothing has ever gone wrong.

The question is what the material is worth, not what the process costs. The sessions can be filmed again next month, because the centre runs the programme every term. The founder's address cannot: he is a guest, he is not local, and he will not be in the building again.

That distinction is the one that resolves it, and it resolves it unevenly. The correct answer is not the same for every card on the shoot.

YOUR ANSWERS

1. Why did the same shoot end up with two different offload procedures rather than one?

2. A drag-and-drop copy said it succeeded and two files were still wrong. What does that tell you about what the operating system was reporting?

3. The team had a 2 TB hard disk and a 512 GB laptop. If they had put both copies on the hard disk, in two different folders, would the rule have been satisfied?

STOP HERE

Write your answers before you turn the page. How it played out starts on p. 56.

CASE STUDY · MODULE 1

How it played out

They split the material by whether it could be re-shot. Sessions and cutaways were copied fast and the cards formatted. The four interviews and the founder's address went onto two cards that were never formatted, copied to the hard disk and verified with a SHA-256 manifest that evening, in the hotel, while nothing else was happening. Total extra cost: nothing, because the verification ran during dinner rather than during a session. On day three the card reader failed part-way through an interview copy; the manifest caught two truncated files, the card was still intact, and the copy was simply run again. The team now keeps a one-page log in the project folder recording, per card, what is on it, where the two copies are, and whether it has been formatted.

The questions, discussed

1. Why did the same shoot end up with two different offload procedures rather than one?

Because the value of verification depends on whether the material can be re-created, not on how it was recorded. The session footage could be filmed again the following term at no cost, so a fast unverified copy risked a small, recoverable loss. The interviews and the address happened once, so the same risk was unbounded. Applying the slow procedure to everything would have cost sessions they needed; applying the fast procedure to everything would have staked the whole film on a copy nobody had checked.

2. A drag-and-drop copy said it succeeded and two files were still wrong. What does that tell you about what the operating system was reporting?

It was reporting that the write operation completed, not that the data on the destination matches the source. Nothing read the copy back. A checksum manifest computes a fingerprint of the source and of the copy and compares them, so a failing reader, cable or card shows up as a mismatch at the time rather than as green blocks three weeks later. That is the whole of what `rsync --checksum` and `shasum -c` add.

3. The team had a 2 TB hard disk and a 512 GB laptop. If they had put both copies on the hard disk, in two different folders, would the rule have been satisfied?

No. The rule is two copies on two devices, because the failure being guarded against is a device failure. Two folders on one disk both disappear when that disk dies, and they also both disappear when somebody deletes the wrong folder. The laptop's internal drive, the hard disk, and an unformatted card are three separate devices; two folders are one.

MODULE 1 TEST

Check what stuck

6 questions, one right answer each. This one is for you, not for a record. The answers, and the reason behind each, are in the key on p. 442. If two go wrong, re-read the module before the exam.

1 The same 4K phone clip plays at three frames a second from the original file and smoothly from a quarter-size proxy. What is the mechanism?

A. The camera file is larger, so the disk cannot deliver it fast enough
B. Proxies are held in memory while camera files are read from disk every time
C. The camera file stores frames as differences, so each one must be rebuilt
D. The proxy has fewer pixels, and the number of pixels is the only thing that limits smooth playback

2 A camera recorded 8-bit 4:2:0 in a log profile. The shot is corrected and contrast is added back. What appears?

A. Banding in skies and smooth gradients
B. Colour fringing on saturated red edges
C. A green cast that no amount of white balance correction will remove
D. Macroblocking in the darkest parts of the frame

3 Filming at 30 fps indoors on a 50 Hz supply produces bands crawling up the frame. The same camera outdoors shows nothing. Why?

A. Rolling shutter only affects indoor footage, because artificial light is read out line by line
B. The camera's anti-flicker setting engages automatically in low light
C. Daylight is brighter, so the sensor needs less gain
D. Daylight does not pulse, so every frame catches the same amount of light

4 A card was copied by dragging it to a drive. Every file appeared and the folder sizes matched. Three weeks later one clip is green blocks. What had the operating system actually reported?

A. That every byte had been read back and compared against the original source file
B. That the write completed, not that the copy matches the source
C. That the card contained no bad sectors at the time of the copy
D. That the file's checksum matched the one written by the camera

5 A phone clip is in sync at the clap and half a second late by minute ten. What is the usual cause, and what fixes it?

A. A variable frame rate file; transcode it to constant rate before importing
B. A sample rate mismatch; the editor cannot resample 44.1 kHz audio at all
C. The clap was placed at the end of the take rather than at the start
D. Waveform sync failed because the recording was too quiet to correlate against the camera track

6 In DaVinci Resolve, why is the timeline frame rate described as a setting you cannot change later?

A. It is stored in the render preset, which is created when the project is created
B. It is written into each clip's metadata at import and cannot be rewritten
C. It greys out as soon as any media exists in the project
D. Changing it re-frames every clip whose scaling was set relative to the old frame

2

MODULE 2

The cut itself

Place and trim a cut you can defend frame by frame — naming the moment it lands on, the trim operation you used and what that operation did to everything downstream — and diagnose a cut that feels wrong by identifying which of the six continuities the eye lost.

Lesson 11 · 9 min

Your first cut is two seconds late

THE ONE THING TO KEEP

Trim every shot to the frame where it stops giving new information, and cut to the reaction more often than you think.

The edit that feels slow, and why you cannot see it

You finish your first assembly. You watch it back. Something is wrong and you cannot name it. The shots are the ones you chose. The order makes sense. It just sits there, heavy.

Almost always the problem is length. Nearly every shot is held about two seconds past the moment it stopped saying anything. You cannot see it because you already know what is coming next. The viewer does not, and they are not being patient on your behalf.

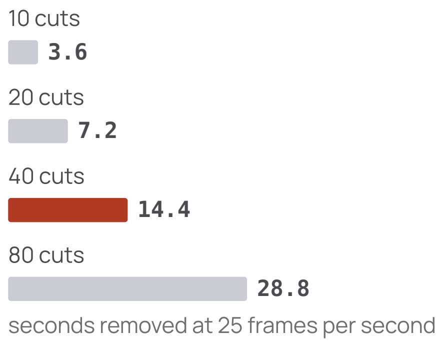
A cut is a decision about attention

Stop thinking of editing as joining clips. A cut is you deciding, for someone else, that they are finished with this and should now look at that. Two questions have to have answers: **why now**, and **why there**. If a cut has no answer to either, the viewer feels a small unexplained bump, and enough of those and they leave without ever forming the thought "the editing was bad".

The two-second fault has a mechanism. Recognition is much faster than completion. Someone reaches for a cup: you understand "reaching for a cup" within a handful of frames. The physical action takes a second and a half. If you hold to the end of the movement, you have held roughly a second and a quarter past the point where the shot delivered its information. Do that at forty cut points and you have added nearly a minute of nothing.

Figure 2.1

What trimming nine frames at every cut takes out



Nine frames is about a third of a second, which nobody notices at a single join. Forty joins is fourteen seconds of a film that was not saying anything.

The six-frame pass

This is a drill, not a rule, and it works on almost every first edit.

1. Get your assembly to the end. Do not polish anything.
2. Go to the first cut. Trim six to twelve frames off the end of the outgoing shot. At 25 fps that is a quarter to half a second.
3. Do it at every cut. Do not watch as you go.
4. Now watch the whole thing.

Most of it will be better and you will feel it immediately. A few cuts will be worse, and those are the valuable ones — they tell you which shots were carrying something you had not consciously noticed. Put those frames back, one at a time, until it breathes again.

The reverse drill is just as useful. Watch once with the sound off: you see length, because nothing is covering it. Watch once with the picture off: you hear whether the thing actually holds together as an argument.

Shorter is not the rule. "No shot outlives its reason" is

There are shots that should be long. A face taking in bad news needs time, because the subject of that shot is the changing, not the face. A joke needs the pause before the answer. A landscape you want the viewer to sink into needs longer than it takes to identify a landscape.

The test is not duration. It is whether the shot is still doing something at the frame you are looking at. Reaction shots are the usual place people cut too early: the interesting thing in a conversation is frequently the person listening, not the person talking.

You cannot cut coverage you did not shoot

This is the honest limitation and it is worth taking seriously before you next film anything. An edit can only rearrange what exists. If you shot one continuous angle of an interview, you have one angle, and every cut you make will be a visible jump. No software fixes this.

So shoot the cutaway. Thirty seconds of hands, the room, the object being discussed, someone walking in — filmed after the fact, from anywhere — is what buys you permission to remove three minutes of waffle later. Editors call this B-roll and beginners think it is decoration. It is not decoration. It is the licence to change time.

The order of work

Edits go in stages, and skipping stages is the most expensive mistake in this course.

- **Assembly.** Everything usable, in order, roughly. Ugly and long.
- **Rough cut.** Structure fixed, length roughly right, still messy at the joins.
- **Fine cut.** Every cut point trimmed to the frame.
- **Picture lock.** No more picture changes.
- Then, and only then, sound design, music, colour, titles and export.

Grade a shot before picture lock and you will throw the grade away when you shorten the shot. Cut music in before the structure is settled and you will start protecting the music instead of the film. Lock first. Everything after this lesson assumes you did.

Today

Open the last thing you edited. Do the six-frame pass on the first ten cuts, and watch those thirty seconds against the original. If you have never edited anything, film a person making tea in five separate shots, cut it, then cut it again half as long.

Tools do not matter yet. This works the same in DaVinci Resolve, Kdenlive, Shotcut or CapCut on a phone, and the next lesson picks one.

BEFORE YOU MOVE ON

An editor removes eight frames from the end of every shot in a rough cut. Most cuts improve, but three feel abrupt and wrong. What do those three cuts most likely indicate?

A. The frame rate is wrong on those clips, so eight frames removed a different amount of time than elsewhere.
B. Shortening always damages a cut, and the other cuts only seemed better because the edit got faster overall.
C. Those shots were still delivering something in their final frames — a reaction completing, a movement resolving — so the information was cut off rather than the dead air.
D. Those three cuts needed a dissolve instead of a hard cut, and the abruptness is a missing transition.

The answer and why: p. 450

Lesson 12 · 10 min

A bad cut is felt before it is seen

THE ONE THING TO KEEP

Cut on movement, keep the camera on one side of the line, and let sound cross the join before the picture does.

Why the audience notices without knowing why

Show an ordinary viewer a badly cut scene and they will not say "the screen direction reversed". They will say it felt cheap, or they will simply disengage. The reason is physical rather than aesthetic. Your eye is tracking something — a face, a hand, a moving car. At a cut, if the thing you were tracking has jumped to a different part of the frame, or is now moving the other way, your eye has to find it again. That re-acquisition takes a fraction of a second, and during it the viewer is looking for the subject rather than watching the film.

Good cutting is mostly the business of not making anyone do that.

Cut on the action

The most useful technique in editing, and the easiest to apply tonight. When a movement happens across two shots — someone stands up, opens a door, turns their head — do not cut before or after the movement. Cut in the middle of it, near the point of greatest motion.

In practice you overlap. Let the outgoing wide shot begin the door opening. Start the incoming close shot two or three frames *earlier* than where the wide left off, so the action repeats slightly. The eye is busy following the motion and the join disappears. If you cut cleanly at the exact same frame it often reads as a tiny stutter, because a real body decelerates and the second camera never quite matches.

Match the position too. If the door is on the right of frame in the wide, keep it right in the close. Editors call this matching the frame, and it is the difference between a cut you can watch fifty times and one that itches.

The line, and eyelines

Two people talk. Draw an imaginary line between them. Keep every camera on one side of that line and person A always looks screen-right, person B always looks screen-left, and the audience knows without thinking that they are looking at each other. Cross the line and both of them look the same direction — at the same wall — and the scene quietly stops being a conversation.

This is the 180-degree rule and it is not a superstition; it is how screen direction encodes relationship. The same applies to travel: a character walking left to right who cuts to walking right to left reads as turning back.

If you were handed footage that crosses the line, you have three repairs: cut away to something neutral in between, use a shot taken *on* the line (straight down the middle, both people in profile or facing camera), or use a moving shot where the camera is visibly seen to cross.

J cuts and L cuts

Named for the shape they make in the timeline once you unlink audio from video.

- **L cut**: the audio of the outgoing shot continues over the incoming picture. You cut to the listener while the speaker is still talking.
- **J cut**: the audio of the incoming shot starts before its picture. You hear the next room, or the next line, a beat before you see it.

Both work for the same reason: the sound is continuous across the join, so the cut has nothing to announce itself with. A dialogue scene cut with hard, matched audio-and-picture cuts sounds like a slideshow. The same scene with two or three J and L cuts sounds like a film, and no viewer can tell you what changed.

Mechanically, find the link or linked-selection toggle in your editor's timeline, switch it off, and drag the audio edge past the video edge. Every editor mentioned in this course does it, including Kdenlive and Shotcut.

Jump cuts are a device, not an error

Classically a jump cut is a mistake: same subject, similar framing, some time removed, so the person appears to teleport within the frame. The eye tracks a face and the face has moved without cause.

In talking-head and short-form video the convention flipped, and it flipped for a reason. When the audience already knows the speech has been compressed — every breath, "um" and false start removed — the jump is not read as a break in continuity. It is read as honesty about the compression. It works.

It stops working in two situations, and both are common. If the speaker moves a lot between cuts, the teleport is large and the eye complains. And a hundred jump cuts destroy any sense of place, because the space is never allowed to exist.

The cheap fix, and it is genuinely what professionals use: on every second or third jump, scale the incoming clip to about 110–115% and reframe slightly. Now the framing has changed enough to read as a different shot rather than a discontinuity in the same one. Alternate the punch-in with a cutaway and the sequence stops feeling like a machine gun.

Transitions

A hard cut is right roughly nineteen times in twenty. A cross-dissolve means time has passed or we are moving to a different mental space, and it means only that. Everything else in the transitions bin — cubes, wipes, glitch presets, page peels — is a signal that a decision has not been made about what the cut is for. Use them when the piece is about that look, and not to smooth over a cut that does not work; a bad cut with a dissolve on it is a bad cut that now takes longer.

Today

Find any two-person conversation you have shot or downloaded. Cut it hard, watch it, then move two audio edges to build one J cut and one L cut. That single change is the fastest visible improvement in this course.

BEFORE YOU MOVE ON

A vlogger removes every pause from a piece to camera, producing forty jump cuts. Viewers say it feels frantic and claustrophobic. The vlogger adds a cross-dissolve to every cut. Why does this not fix it?

A. Cross-dissolves are only correct between different scenes, so applying them within one scene is a continuity error the viewer detects.
B. The dissolves reduce perceived sharpness, and the loss of detail is what reads as claustrophobic.
C. Dissolves increase total running time, which cancels out the tightening the jump cuts achieved.
D. The complaint is about the framing never changing and the space never being established, which a transition cannot supply; varying the shot size and adding cutaways is what the sequence lacks.

The answer and why: p. 451

Lesson 13 · 10 min

Ripple, roll, slip and slide

THE ONE THING TO KEEP

Four named operations move a cut — ripple changes duration and shifts everything downstream, roll moves the join only, slip changes the content inside a clip, slide changes its position — so decide which question you are asking before you touch the edge.

Four operations, and why dragging is none of them

Most self-taught editors have one move: grab the edge of a clip and drag. That single gesture is doing different things depending on where the mouse landed and whether a mode is active, which is why the result is sometimes a gap, sometimes everything shifting, and sometimes a cut that moved when you did not want it to.

There are four named operations. Learn them by name and the timeline stops surprising you.

Ripple

A **ripple trim** changes the length of one clip and moves everything after it by the same amount. Remove twelve frames from the end of shot 3 and shots 4, 5 and 6 all arrive twelve frames earlier. The overall duration changes.

This is the everyday trim. It is what you want when you are tightening — the six-frame pass from the first lesson is a ripple trim at every cut.

The cost: anything already timed to a position downstream moves out of place. Music cut to picture, a title that lands on a word, a graphic timed to a gesture — all shift. This is the main reason to lock structure before doing timed work.

Roll

A **roll trim** moves the cut point between two adjacent clips without changing the total duration. Shot 3 gets twelve frames longer and shot 4 gets twelve frames shorter. Nothing downstream moves.

This is the trim for fine-tuning a specific join once the structure is fixed. It is how you find the exact frame a cut on action works on, without disturbing the music you already placed.

Roll requires handles — unused material beyond the current in and out points on both clips. If shot 4 starts at the very first frame of its source file, you cannot roll the cut earlier, because there is nothing there. This is why shooting a few seconds before and after every take matters more than it looks.

Slip

A **slip** changes *which part* of a clip is shown without changing where it sits or how long it is. The clip stays in the same place on the timeline, the same number of frames long, but the content inside slides earlier or later.

Use it when the timing of the sequence is right but the moment inside the shot is wrong — the reaction arrives two frames late, the hand enters frame just after your cut. Slip six frames and the content lands where you need it, and nothing else in the timeline moves at all.

Slip is the most under-used trim among beginners and one of the most useful.

Slide

A **slide** moves a clip earlier or later along the timeline while the clips either side absorb the change — the one before gets longer, the one after gets shorter, or vice versa. The clip's own content and duration are untouched; its position changes.

Use it to reposition a cutaway. You know the insert shot is right and two seconds long; you just want it half a second later. Slide, and the shots around it adjust.

The table you should build in your head

Each operation answers a different question:

Figure 2.2

Four trims, split by what happens downstream

Changes the programme's length

- Ripple: the clip gets shorter or longer
- Everything after it moves by the same amount
- Music, titles and timed graphics all shift
- The everyday trim, and the six-frame pass

Leaves the length alone

- Roll: moves the join, the neighbour absorbs it
- Slip: changes the moment inside the clip
- Slide: moves the clip, neighbours absorb it
- Safe once music and titles are placed

Roll and slip both need handles, meaning unused frames beyond the current in and out points. A clip trimmed to the bounds of its source has none, and the editor marks the edge rather than saying so.

- **The shot is too long** → ripple.
- **The join is on the wrong frame, but the rest of the timeline is right** → roll.
- **The right moment is inside the clip but not where I put it** → slip.
- **The shot is right, its place is wrong** → slide.

Get into the habit of asking which of the four you need before you touch anything. The gesture then becomes obvious.

Where they live

In DaVinci Resolve's edit page, the trim tool is `T` and the selection tool is `A`. With the trim tool active, the pointer changes near an edit point: on the outgoing side you get a ripple, in the middle you get a roll, dragging the clip's body gives a slide, and holding the modifier gives a slip. Resolve also has a dedicated **Trim Edit Mode** where the arrow keys nudge by a frame and shift-arrow by five.

Premiere is `B` for ripple, `N` for rolling, `Y` for slip, `U` for slide. Kdenlive and Shotcut expose ripple and roll clearly and slip through a modifier drag. The names are consistent across all of them, which is why learning the names beats learning one program's mouse behaviour.

Trim by keyboard, not by pixel

The single habit that changes the quality of your cuts: stop dragging and start nudging. Select an edit point, then use the keys to move it one frame at a time while the timeline loops around the cut.

Resolve's dynamic trim — `W` to toggle — plays a loop around the selected edit point and lets `,` and `.` move it a frame each way *while it plays*. You feel the cut change rather than looking at numbers. Ten minutes of this on one difficult join teaches more about cutting than an hour of reading.

Two traps

Linked selection. When audio and video are linked, a ripple on the picture ripples the sound with it, which is usually right. When you are building J and L cuts you want them independent. Know where the toggle is — Resolve puts it in the timeline toolbar, and it is the cause of most "my audio went out of sync" reports.

Trimming with no handles. If a clip has been trimmed to the exact bounds of its source, roll and slip do nothing in one direction, and the editor will usually tell you with a small marker on the edge rather than an error message. Look for it before concluding the tool is broken.

Today

Take any three-shot sequence and perform all four operations on it in turn, watching what moves each time. Ten minutes, and the timeline stops being unpredictable.

BEFORE YOU MOVE ON

An editor has cut a montage to music, with every shot landing on a beat. One shot in the middle shows the subject's hand entering frame six frames after the cut, instead of on it. Which operation fixes it without disturbing the music sync?

A. Ripple trim six frames off the head of the shot, which removes the dead time and brings the hand to the cut point.
B. Slip the clip six frames, which changes which part of the source is shown while its position and duration stay exactly as they are.
C. Roll the incoming edit six frames earlier, so the shot starts sooner and the hand arrives on the beat as intended by the original design.
D. Slide the clip six frames earlier, so the content arrives sooner while the neighbouring shots absorb the change.

The answer and why: p. 451

Lesson 14 · 9 min

Six things a cut has to satisfy, in order

THE ONE THING TO KEEP

A cut can satisfy six things — emotion, story, rhythm, eye-trace, screen plane and physical space — and when they conflict, the audience forgives a broken geography far more readily than a wrong emotional beat.

A ranking, not a checklist

Walter Murch, who edited *Apocalypse Now* and *The Conversation*, wrote a short book called *In the Blink of an Eye* in which he set out six criteria a cut can satisfy, and — more usefully — ranked them by how much they matter. The ranking is the valuable part, because cuts that satisfy all six are rare and the question in practice is always which ones to give up.

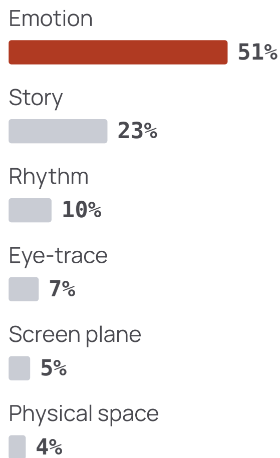
His weighting, roughly:

1. **Emotion** — does the cut serve what the audience should feel? About half the value.
2. **Story** — does it advance the narrative? A quarter.
3. **Rhythm** — does it come at an interesting and right moment? A fifth or so.
4. **Eye-trace** — does it respect where the audience is looking? A few per cent.
5. **Two-dimensional plane of screen** — does it respect screen direction and the line? A few per cent.
6. **Three-dimensional space** — is the physical geography continuous? Barely one per cent.

The numbers are Murch's own rough weighting and are not measurements. What they encode is a claim you can test against your own reactions: audiences forgive a spatial discontinuity readily and forgive an emotional wrong note almost never.

Figure 2.3

Murch's weighting of what a cut can satisfy



Murch's own rough weighting, not a measurement. What it encodes is testable against your own reactions: audiences forgive broken geography and almost never forgive a wrong emotional beat.

Why the bottom of the list is at the bottom

This is the part that surprises people who have been taught continuity as a set of rules.

If a cut is emotionally right but the actor's glass has moved between shots, essentially nobody notices, because the audience is busy feeling something. If a cut is technically flawless — perfect eyeline, perfect action match, perfect geography — but lands on the wrong beat emotionally, the scene dies and nobody can say why.

This is not permission to be sloppy. Continuity errors cost nothing individually and accumulate: enough of them and the piece reads as careless, which is itself an emotional signal. But when you have to choose — and in unscripted work you choose constantly — the ranking tells you which way.

Using it as a diagnostic

The practical value is in reverse. When a cut feels wrong and you cannot name why, walk the list from the bottom up, because the lower ones are the ones you can *see*:

- **Space**: has the geography flipped? Is somebody suddenly on the wrong side of the room?
- **Screen plane**: did the subject jump across the frame? Did screen direction reverse?
- **Eye-trace**: where was my eye at the last frame of the outgoing shot, and is there something there in the first frame of the incoming one?
- **Rhythm**: is this cut arriving at the same interval as the last four? Is it landing on the speech or between phrases?
- **Story**: does the incoming shot tell me something new?
- **Emotion**: am I being asked to feel the right thing at this moment, or am I being moved on before I have finished?

Four out of five times the answer is in the top two and you were looking at the bottom two.

Eye-trace, in practice

This is the one most worth learning to see, because it is invisible until somebody names it and then obvious forever.

At the last frame before a cut, your eye is somewhere specific — on a face, a hand, a moving object. At the first frame after, it has to find the new subject. If the new subject is roughly where the old one was, the transition is free. If it is on the other side of the frame, you spend a fraction of a second searching.

So: note where the attention is at the out point, and choose an in point that puts something worth looking at near the same place. This is what editors mean by "matching the frame", and it is why a cut between two wide shots of the same room often works better than a cut between a close-up on the left and a close-up on the right.

You can also use it deliberately. Deliberately putting the new subject somewhere else produces a small jolt, which is exactly what you want at a scene change or a reveal.

Rhythm is the one you can measure

Emotion and story are judgements. Rhythm has a number attached, and the next lesson is about it.

The honest limits of the framework

Murch was describing narrative fiction cut for a cinema audience watching in the dark for two hours. Three adjustments for the work most people are actually doing:

- **Short-form compresses everything.** A fifteen-second vertical video has no room for emotional build, so story and rhythm move up the list and emotion is carried almost entirely by the performance rather than by the cutting.
- **Documentary and interview work often cannot satisfy the bottom four at all.** One camera, one angle, no coverage. The whole reason jump cuts became acceptable is that the bottom of the list had to be abandoned wholesale.
- **Tutorials and explainers invert it.** Comprehension is the emotion. A cut that arrives before the viewer has understood the previous point is the equivalent of a wrong emotional beat.

The ranking is a tool for making a choice, not a law. What is worth taking from it permanently is the habit of asking what the cut is *for* before asking whether it is clean.

Today

Take a cut in your own work that you know is not right. Walk the six, bottom to top, and write down where it fails. The naming is most of the fix.

BEFORE YOU MOVE ON

In a cut scene, a subject's glass is half full in the wide and full in the close-up, and the cut lands exactly on the emotional turn of the scene. An assistant wants to move the cut two seconds later to fix the glass. What should the editor say?

A. Move it, because continuity errors accumulate and an audience that spots one starts looking for others, which breaks the scene more thoroughly than a slightly late cut would.
B. Move it and fix the emotion later with music, since scoring can restore a beat that the picture edit has shifted.
C. Cut away to an insert of something else for two seconds, which solves both problems at once and is always the correct answer to a continuity conflict.
D. Keep the cut, because the emotional beat outranks the physical continuity and moving it trades the most valuable property of the cut for the least.

The answer and why: p. 452

Lesson 15 · 9 min

Pace is a pattern, and you can count it

THE ONE THING TO KEEP

Average shot length gives you a number for pace, but rhythm lives in the variation around it — put long shots where the viewer should settle and short ones where they should be pushed, and cut speech at the breath rather than inside the phrase.

The thing you feel and can also measure

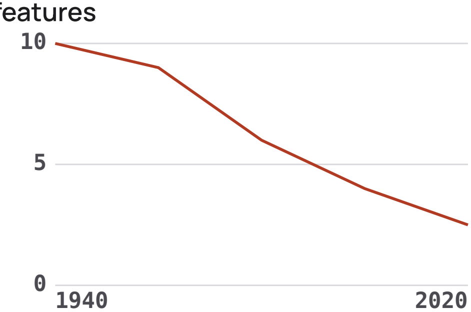
Pace is usually discussed as taste. It has at least one honest number behind it: **average shot length**, the total running time divided by the number of shots.

Film scholars have tracked this for a century. Hollywood features in the 1940s ran an average shot length of around 10 seconds. By the 2000s, action films were routinely under 3, and some are under 2. A modern talking-head YouTube video with jump cuts can be under 2 seconds. A slow-cinema feature can be over 60.

The number by itself proves nothing — a piece with an ASL of 3 seconds can be either urgent or exhausting. What it gives you is a way to check whether what you built matches what you intended, and to compare a section against the rest of the piece.

Figure 2.4

Average shot length in Hollywood features



Across: Decade

Up: Seconds per shot

—— Average shot length

Rounded from published shot counts, so read the shape rather than the points. The number proves nothing on its own; it tells you whether what you built matches what you intended.

Compute it, once, on your own work

Most editors will tell you the clip count of a timeline. Divide the duration by it. Then do it per section rather than for the whole piece, because the interesting information is in how the number changes.

A well-shaped ten-minute piece is almost never uniform. It typically runs faster at the opening, settles into a slower middle where information is delivered, accelerates through a sequence, and lengthens again at the end. If your per-section numbers are all the same, that is worth knowing: the piece has one gear.

Rhythm is the distribution, not the average

Two sequences can have the same average and feel nothing alike. One cuts every three seconds, metronomically. The other cuts at 1, 1, 1, 8, 1, 1, 1, 8. The second has a shape; the first has a pulse.

The practical version of this: **vary shot length deliberately, and put your long shots where you want the viewer to settle.** A run of similar-length shots numbs. A long shot after a run of short ones reads as arrival. A short shot after a run of long ones reads as a shock.

This is the mechanism behind the standard accelerating sequence — shots getting progressively shorter toward a climax — and it works because the audience is entrained to an interval and then denied it.

Cutting to breath, not to beat

For speech-led work, the rhythm you are cutting to is the speaker's, and it is written into the audio.

People speak in phrases separated by breaths. A cut placed *inside* a phrase feels like an interruption, even when the words are continuous. A cut placed at a breath is invisible. Look at the waveform: the gaps are visible, and they are the places your cuts belong.

Two refinements that sound professional immediately:

- **Cut on the inhale, not after it.** Leaving the breath at the head of the incoming clip sounds natural; leaving it at the tail of the outgoing clip sounds like a person was cut off.
- **Leave a small amount of silence after a point lands.** Roughly a quarter of a second. It is the audible equivalent of a paragraph break, and its absence is why dense edits feel airless.

The three-second problem in short-form

Short-form convention pushes cuts very fast — every one to two seconds, sometimes faster. The reason usually given is attention span. The real mechanism is that a change in the frame resets the viewer's decision to keep watching, so frequent change buys retention.

It works, and it has a cost that is worth naming: at that rate nothing can be *understood*, only registered. If your piece carries an idea rather than a vibe, cutting at short-form pace destroys comprehension while improving the retention graph. Tutorials cut this way test well and teach nothing.

The honest compromise for explanatory work: keep the *visual* change frequent — punch-ins, cutaways, a graphic arriving — while keeping the *conceptual* units long enough to land. The frame changes every two seconds; the idea changes every twenty.

Rhythm in the assembly, not just the fine cut

A mistake worth avoiding: trying to fix pace at the end by trimming. Pace is mostly structural. If the piece feels slow at minute four, the cause is usually that minute four contains a section that should not exist, not that its shots are half a second long each.

The diagnostic: watch the piece and mark every moment your attention went. Then look at what is immediately *before* each mark. That is where the problem is, not at the mark.

Music, and the trap in it

Cutting to a music track imposes the track's rhythm on the piece, which is superb for montage and corrosive for anything with an argument. The structure ends up serving the phrase lengths of a song somebody else wrote.

If you are cutting to music, cut picture first and place music second, except where the piece genuinely is the music — a montage, a title sequence, a dance piece.

Today

Take a five-minute piece you made, count the cuts, and compute the average shot length. Then compute it separately for the first minute and the middle minute. If the two numbers are the same, you have found something to change.

BEFORE YOU MOVE ON

A tutorial is recut at short-form pace — a visual change every one to two seconds — and its retention graph improves markedly, but viewers in the comments say they did not follow it. What is the mechanism?

A. The faster cuts introduced continuity errors that the slower version hid, and the errors are distracting from the content.
B. Retention measures whether the frame kept changing, while comprehension needs a conceptual unit to stay on screen long enough to be processed, and the recut shortened both together.
C. Retention graphs are unreliable on short videos and the improvement is probably noise rather than a real effect.
D. The audio has been cut to match the picture throughout, so the speaker now sounds rushed and breathless, and listeners have disengaged from the delivery rather than from the ideas being presented.

The answer and why: p. 452

Lesson 16 · 9 min

Building a sequence out of an action

THE ONE THING TO KEEP

A sequence is built out of fragments the viewer stitches into a continuous action, so shoot every action whole from three distances and join the shots with 'but' and 'therefore' rather than 'and then'.

Shots do not add, they multiply

In the 1920s Lev Kuleshov cut the same neutral shot of an actor's face against three different images — reportedly a bowl of soup, a child in a coffin, a woman on a divan — and audiences described the actor as hungry, grieving and desirous. The face was identical every time.

The experiment's details are disputed and the original footage is lost, so treat the anecdote as a demonstration rather than an evidence base. The principle behind it is not disputed and you can test it yourself in twenty minutes: **the meaning of a shot is produced by the shot next to it.** Editing is not selection; it is juxtaposition.

This is the whole craft in one sentence. It also means you can build things you never filmed.

The sequence, as a unit

A sequence is a group of shots that together perform one action or make one point. A person makes tea. Someone arrives at a building and goes inside. A process is explained. A sequence has a beginning, a middle and an end even if the film around it does not.

The classic construction, and it works for almost any physical action:

1. **Wide** — establishes where we are and what is about to happen.
2. **Medium** — the person doing the thing.

3. **Close on the hands** — the actual detail of the action.
4. **Insert** — the object itself, very close.
5. **Reaction** — the face, or somebody else watching.
6. **Wide again, or a new wide** — the state of the world after.

Six shots, and the viewer believes they watched a continuous event. They did not: they watched six fragments, probably filmed in a different order, possibly over an hour.

The shooting list this implies

If you also shoot, this is the most valuable paragraph in the block. For any action you intend to include, film it at least three times: once wide, once medium, once close on the hands. Repeat the whole action each time rather than filming fragments — repeated whole actions cut together; fragments do not, because the match points do not exist.

The thirty seconds this costs on set buys you the ability to compress the action to any length you like, hide any mistake, and cut away at any moment.

Cause and effect, and the word "therefore"

A sequence of shots joined by "and then" is a list. A sequence joined by "but" or "therefore" is a story.

This is a test you can apply to your own assembly. Read your shot order aloud, inserting the joining word between each pair. "She picks up the kettle, **and then** she fills it, **and then** she puts it down." Nothing is happening. "She picks up the kettle, **but** it is empty, **therefore** she goes to the tap." Now there is a small engine running.

In unscripted material you find these relationships rather than write them, but they are almost always there. Somebody tried something, it did not work, so they did something else. The edit's job is to make that visible.

The insert that does the work

The single most useful shot in any sequence is the close insert of the object. It is useful for three separate reasons at once:

- It tells the viewer what matters in the scene.
- It is neutral enough to be placed almost anywhere, so it hides a cut you could not otherwise make.
- It compresses time invisibly. Cut to hands for two seconds and cut back, and the viewer accepts that any amount of time has passed.

If you shoot nothing else extra, shoot inserts.

Compressing time without saying so

Real actions are boring at their real length. A person walking from a door to a desk takes eight seconds; a sequence can do it in two.

The mechanism is that the viewer reconstructs continuity from partial information. Shot of the door opening; cut to feet on the floor; cut to the person sitting down. Nobody asks what happened to the intervening six seconds, because the shots establish direction, and direction implies the journey.

The failure mode: compress too far and the viewer loses the geography. The test is whether they could draw the room. If not, add one wide.

Montage and its two meanings

The word means two different things and both are useful.

Montage as compression — the passage-of-time sequence, usually over music. Days of work in thirty seconds. Works because each shot is an *instance* of a repeated activity rather than a step in a chain.

Montage as argument — cutting two unrelated things together to produce a third idea. A politician's promise, cut to a closed factory. This is powerful and it is also the technique most easily used to say something untrue, because it asserts a relationship without stating it. If the juxtaposition would be a lie in words, it is a lie in pictures, and the fact that you never said it out loud is not a defence.

Today

Film somebody making tea, three times, from three distances, performing the whole action each time. Cut it into a six-shot sequence lasting fifteen seconds. It will take twenty minutes and it teaches the whole block.

BEFORE YOU MOVE ON

An editor needs to compress a two-minute process into twenty seconds but only has one continuous wide shot of it. Why does adding a close insert of the object, filmed separately afterwards, solve the problem that speeding up the wide does not?

A. Because the insert is higher resolution than the wide, so the compressed sequence retains more detail where it matters most to the viewer.
B. Because speeding a shot up introduces frame blending artefacts, while cutting to an insert leaves both shots at their native frame rate.
C. Because an insert re-establishes the geography of the scene and reminds the viewer where everything is, which is exactly what they lose when a long shot is compressed in time.
D. Because cutting to the insert removes the viewer from the continuous action, so any amount of elapsed time can pass behind it without a visible discontinuity.

The answer and why: p. 453

Lesson 17 · 10 min

Cutting a conversation, step by step

THE ONE THING TO KEEP

Cut a conversation as audio first, move every join to a breath, lay room tone underneath, and only then decide what the viewer looks at — which is the listener at least as often as the speaker.

The most common thing you will be asked to cut

Two people talking. An interview. A podcast with cameras. A panel. This is the bulk of real editing work, and there is a working method that is more reliable than instinct.

Step one: the radio edit

Ignore the picture entirely. Build the conversation as audio.

Cut out the false starts, the repeated question, the interruption, the long pause where somebody thinks, the "sorry, can I say that again". Cut for meaning: the shortest version of each answer that still says the thing.

The target is something you could publish as a podcast. Listen to it with the screen off. If it does not hold as sound, no picture will fix it.

This takes longer than you expect and it is where the piece is actually made.

Step two: find the breaths

Now go through the radio edit and move every cut to a breath. Zoom in on the waveform until you can see individual words; the gaps between phrases are visible as troughs.

A cut inside a phrase produces a small click of wrongness even when the words are grammatical. A cut at a breath is inaudible. Put the breath at the *head* of the incoming clip, not the tail of the outgoing one — a person finishing a sentence and then inhaling sounds natural; a person being cut off mid-inhale sounds edited.

Step three: fill the holes with room tone

Every cut you made joined two different moments of the same room, and the background noise level is slightly different at each. Lay a continuous bed of room tone under the whole scene at a level that matches. Now the joins have nothing to announce themselves with.

This single step is the largest difference between an edit that sounds amateur and one that does not, and it costs one track and five minutes.

Step four: decide what the viewer looks at

Only now does picture enter. For each moment, ask: what is the most interesting thing to be looking at right now?

The default answer is wrong. Beginners cut to whoever is speaking. Experienced editors cut to whoever is *reacting*, at least half the time. A conversation is about what is happening between two people, and half of that is on the face of the person not talking.

The practical rule: cut to the listener before a significant line, so we see them receive it. Cut to the speaker on a line where their face is doing something the voice is not.

Step five: J and L every join

Having placed the picture cuts, offset them from the audio cuts. Unlink audio and video and drag the picture edges a beat earlier or later than the sound edges.

As a starting point: bring picture in **before** the new speaker starts — a J cut, usually 6 to 20 frames — so we see the listener begin to respond before we hear them. And let the previous speaker's audio run over the new picture on the way out — an L cut.

Do this at most joins. A dialogue scene where every picture cut lands exactly on an audio cut sounds like a slideshow, and the fix is mechanical.

Step six: the single-camera problem

With one camera and one angle, every cut is a jump cut. Four repairs, in order of preference:

1. **A cutaway.** B-roll, the interviewer listening, the object being discussed, hands, the room. Two seconds of anything else and the cut is invisible.
2. **A punch-in.** Scale the incoming clip to about 110–120% and reframe slightly. 4K source at a 1080p timeline makes this free. Alternate between the wide framing and the punch-in rather than punching in twice in a row.
3. **A speed ramp or a dissolve** across the join, which reads as a deliberate compression rather than a mistake.
4. **Leave the jump cut.** In talking-head work this is an accepted convention and viewers read it as honesty about compression rather than as a break in continuity. It stops working if the subject moves a lot between the two frames.

Step seven: remove the tics, carefully

Um, er, lip smack, the sharp inhale before a sentence. Each is a small ripple trim.

The warning: over-cleaned speech sounds inhuman. Every pause removed, every breath gone, and the result is a rate of information delivery that nobody can follow and a person who apparently does not breathe. Remove the ones that obstruct; leave the ones that are thinking.

And there is a line beyond tidying. Reordering somebody's sentences so they appear to answer a question they were not asked, or splicing clauses to produce a statement they never made, is a fabrication regardless of how clean the join is. The next block deals with where that line sits.

Step eight: watch it once as a stranger

Full screen, from the top, without touching anything. Mark where your attention went. Fix those places and stop.

Today

Take a two-minute interview clip and do steps one to three only: radio edit, breaths, room tone. Listen with the screen off. That is the scene.

BEFORE YOU MOVE ON

An editor cuts an interview, places every picture cut exactly on its corresponding audio cut, and the result feels mechanical despite each individual cut being well chosen. What is the fix and why does it work?

A. Add a short cross-dissolve at every join, which softens the simultaneity and removes the mechanical quality.
B. Lengthen every shot slightly so the cuts arrive less frequently, since the mechanical feeling comes from cutting too often.
C. Offset the picture edges from the audio edges, so sound is continuous across each join and the cut has nothing to announce itself with.
D. Cut to the listener rather than to the speaker at each join, which breaks up the mechanical pattern of alternating between the two faces on every line of dialogue.

The answer and why: p. 453

Lesson 18 · 8 min

The shot that buys you everything

THE ONE THING TO KEEP

Cutaways and inserts are not decoration — they are the licence to remove time, hide a mistake and change rhythm, and an interview needs roughly one usable one for every thirty seconds of finished film.

Three words that get confused

Cutaway — a shot of something other than the main action. The interviewer nodding, the street outside, the audience.

Insert — a close shot of something *within* the main action. The hands on the keyboard, the letter being read, the switch being flipped.

B-roll — a catch-all production term for supplementary footage, mostly used to mean both of the above plus general scene-setting material.

The distinction matters because they do different jobs. An insert belongs to the scene and keeps the viewer inside it. A cutaway steps outside. Using a cutaway where you needed an insert is why some edits feel like they keep wandering off.

What they actually buy

Four things, and it is worth being explicit because beginners treat them as decoration.

Permission to remove time. This is the big one. Cut to hands for two seconds, cut back, and three minutes of the interview are gone with no visible join. Without a cutaway, every removal is a jump cut.

Repair of a bad moment. A cough, a phone ringing, a fluffed sentence, a person looking straight down the lens by accident. Cover it.

Rhythm. A long answer that is entirely one face is monotonous regardless of how good the answer is. Cutaways change the frame without changing the subject.

Information. When somebody describes a machine, show the machine. The viewer's picture of what is being described is wrong until you show it.

How many you need

A usable rule for interview work: **one usable cutaway or insert for every thirty seconds of finished piece.** A six-minute film needs about twelve. They can repeat, but not adjacently.

Most people come back from a shoot with two. The gap between two and twelve is the gap between an edit that works and an edit that fights you.

Shooting them, when you are the one filming

The ten-minute pass, after the interview and before you pack up:

- The subject's hands, doing anything. Turning a page, holding a cup, gesturing.
- The subject listening — ask them to sit still and look at where you were standing. This is the "noddy" and it has been standard practice in television for decades.
- The room. Wide, from the door.
- Whatever objects were mentioned. Each one, close.
- The way in — the door, the street sign, the corridor.
- Something moving. A fan, traffic outside a window, a screen.

Each is five to ten seconds. Hold longer than feels necessary; a two-second cutaway needs at least six seconds of stable footage to choose from.

The ethical line on the noddy

A reaction shot filmed after the fact, cut in as though it were a response to what was said, is standard practice in broadcast and is also the technique behind a well-known category of complaint. The distinction that broadcasters use, and it is a reasonable one to adopt: a neutral listening shot that stands in for the fact that the person was listening is acceptable. A shot of the interviewer laughing, frowning or looking sceptical, cut in against a specific statement to imply a reaction that never happened, is a fabrication.

If the cutaway asserts something about what somebody thought, it needs to be true.

Using them well

Cut to the cutaway on a reason. A cutaway that arrives because the editor got bored feels arbitrary. One that arrives when the speaker mentions the thing, or when their hand moves, feels motivated.

Come back before the viewer wants to. The default error is leaving the cutaway up too long. Two seconds is usually enough; four is often too many. The exception is where the cutaway carries new information the viewer needs to read.

Do not let the sound change. The interview audio runs continuously under the cutaway. If the cutaway's own audio comes in — the fan, the traffic — mix it well underneath, or mute it entirely and let room tone carry.

Vary the distance. Two cutaways of the same size in a row read as a mistake. Alternate close and wide.

When you have none

You will sometimes inherit footage with no cutaways at all. The options, all compromises:

- **Stills.** A photograph, a document, a screenshot, held with a slow push in. Free, fast, and better than nothing.

- **The punch-in.** Scale the same shot; a second framing is a weak cutaway but a real one.
- **Graphics.** A word, a number, a map. Works well for anything factual.
- **Archive or licensed stock.** Costs money or careful licence-checking, covered later in this course.
- **Go back and shoot them.** The room is often still there, and the object usually is. Nobody needs to be in the shot.

That last one is the answer more often than people assume, and it is why the shot list you write during the logging pass matters.

Today

Take something you have edited and count its cutaways per minute. If it is under two, that is the thing to change about how you shoot.

BEFORE YOU MOVE ON

An editor needs to remove ninety seconds from the middle of a single-camera interview. They have a two-second insert of the subject's hands. Why does placing the insert across the removal work, while a cross-dissolve at the same point does not?

A. The insert removes the subject from frame, so the viewer has no reference to notice the discontinuity against, whereas a dissolve keeps the same face visibly changing position.

B. The insert restores continuity of motion across the join in a way that a cross-dissolve never can, because a dissolve simply blends two static frames into one another over a fixed duration.

C. A dissolve reduces image quality at the join because both clips are re-encoded, while an insert is a straight cut with no re-encoding at all.

D. The insert lasts two seconds, which is long enough for the viewer to forget the previous framing, while a dissolve is too short to have that effect.

The answer and why: p. 454

Lesson 19 · 9 min

Repairs for coverage you never shot

THE ONE THING TO KEEP

One camera yields several usable shots through punch-ins, reframes, ramps, freezes, stills and graphics — but each repair has a cost, and a missing reverse angle of a real person is not among the things any of them can supply.

The situation

Somebody hands you an hour of footage from one locked-off camera and asks for a six-minute film. Or a wedding filmed from the back of the hall. Or a conference talk where the only other angle is a phone someone held for ninety seconds.

You cannot make shots that do not exist. You can make the ones that do exist go further, and there are about seven techniques for it. Each has a cost, and knowing the cost is what stops you using the wrong one.

The punch-in

Scale the image and reframe. On a 4K source cut at 1080p this is free — you are still delivering full resolution at up to 200%. Below that it costs sharpness.

What makes it read as a second angle rather than a zoom: change the *framing*, not just the size. Move the subject to a different part of the frame. A punch-in that keeps the subject in the same place looks like a zoom; one that reframes from centred to a third looks like a different camera.

The cost: no change in perspective. A punch-in has the same lens compression as the wide, so a viewer who is paying attention can tell. In practice almost nobody is.

The reframe as a move

Animate the punch-in slowly across the shot — a gentle push in over fifteen seconds, or a slow drift. Done at a rate the viewer does not consciously see, this gives a static shot some life.

The cost: overuse. If every shot drifts, the piece develops a seasick quality. Keep it to shots that are on screen for more than eight seconds.

The speed ramp

Compress a dull stretch by ramping speed up and back down over it. A hundred and fifty per cent for three seconds, easing in and out, reads as a stylistic choice rather than as missing footage.

The cost: audio. You cannot ramp sound and keep it intelligible, so this only works over music or a voice-over, not over the speech you are ramping.

The freeze frame

Hold a frame for a beat. Useful at an introduction, a name super, a point being made.

The cost: it stops the film. It works once or twice in a piece and becomes a tic on the third use.

Stills and documents

A photograph, a screenshot, a scanned document, a map. Add a very slow push or drift so it is not a dead rectangle.

The cost: rights, which most people forget. A photograph you found is somebody's photograph. A screenshot of a website shows their copyrighted design. This is dealt with properly in the structure block, and the short version is that "I found it online" is not a licence.

Graphics and text

A number, a quote, a simple animated diagram. For anything factual, often better than a cutaway would have been, because it is the information rather than a decoration around it.

The cost: time to make, and a graphic that looks amateur damages a piece more than a plain cut would.

The flip, and why to be careful

Mirroring a shot horizontally produces a shot facing the other way, which can rescue a screen-direction problem or manufacture a reverse angle.

It fails visibly whenever the frame contains anything asymmetric: text of any kind, a clock, a watch on the correct wrist, a shirt buttoning the standard way, a hair parting, a wedding ring, a steering wheel, a door handle. In practice usable only on plain-background close-ups and abstract b-roll. Check the frame carefully before you commit.

Shooting the pickups

The repair everybody forgets: go back and film the missing shots. Inserts, objects, the room, the street, hands — almost none of these require the original subject or the original day.

For an interview this is often twenty minutes of work that solves the whole problem. Keep the list from your logging pass, because that list is exactly the pickup shot list.

What genuinely cannot be repaired

Say these out loud to whoever is asking, early:

- **A missing reverse angle in a two-person conversation.** There is no way to invent a shot of the other person's face. You can cut to a listening shot of the same person, to an insert, or to a wide — but if the other participant was never filmed, they were never filmed.

- **An action that happened once and was missed.** The cake being cut, the goal, the moment of arrival.
- **Anything out of focus, clipped, or in the wrong frame.** Covered in the earlier lesson, and worth repeating because clients ask.

Generative tools are increasingly offered as the answer to some of this. Where the result is an invented shot of a real event, it is a fabrication regardless of quality, and in documentary, journalism or evidence it is not available to you at any level of polish. Where it is background or texture in something openly fictional, that is a different conversation, and Video with AI is where it happens.

Today

Take a single-camera clip and make three usable angles from it: the original, a reframed punch-in, and a slow drift across a different part of the frame. Cut them together and see how far one shot goes.

BEFORE YOU MOVE ON

An editor mirrors a shot of a presenter horizontally to fix a screen-direction problem. The shot is a medium close-up against a plain office wall. What should they check before committing, and why does the fix so often fail?

A. Whether the lighting comes from the correct side, since mirroring reverses the key light and makes the shot fail to match its neighbours.

B. Whether the clip was shot at a high enough resolution, since mirroring resamples the image and softens it.

C. Whether the frame contains anything asymmetric — lanyard text, a watch, a shirt placket, a hair parting — because a reversed one of those is instantly readable as wrong.

D. Whether the subject moves at all during the shot, because mirroring reverses the direction of their movement and will therefore contradict the established screen direction of the shot that follows.

The answer and why: p. 454

Lesson 20 · 8 min

Where people decide to stay

THE ONE THING TO KEEP

Viewers answer 'what is this and is it for me' in the first few seconds whether or not you have told them, so state the subject immediately, open on a real image rather than black, and pay off whatever the opening promised.

The shape of the curve

Every platform that shows you a retention graph shows the same shape: a steep drop in the first few seconds, then a gentler slope, then a long tail. The size of that first drop varies enormously between pieces, and it is the single number most under an editor's control.

The mechanism is not mysterious. A viewer arriving at a video is answering one question — *what is this, and is it for me?* — and they will answer it whether or not you have given them the information. If the opening does not answer it, they answer it themselves, usually with "no".

What an opening has to do

Three things, in whatever order the material allows:

1. **Say what this is.** Not the title. The subject, in the first sentence or the first image.
2. **Say why it is worth the next few minutes.** A question, a problem, a surprising claim, a promise of a specific thing.
3. **Establish the register.** Serious or funny, formal or personal, fast or slow. People decide whether they want to be in a room with you very quickly.

A piece that opens with a logo animation, a slow fade up from black, thirty seconds of establishing drone footage and a greeting has done none of the three by the time most of its audience has left.

Figure 2.5

What a working opening has delivered, and when

- **Second 0**
  A real image. This frame is also the thumbnail, so a fade from black gives the platform nothing to show.
- **Second 3**
  The subject is stated, in a sentence or an image. The viewer can now answer what this is.
- **Second 8**
  Why it is worth the next few minutes: a question, a problem, a specific promise.
- **Second 15**
  The register is set, and whatever the opening promised is visibly on its way to being paid.
- **The first tenth**
  The drop-off flattens. After this the curve is about the material rather than the opening.

A logo animation, a fade up and a greeting have done none of the three by the time most of the audience has gone.

The cold open

The most reliable structure: start with the strongest moment in the piece, then go back and begin properly.

In practice, take the best twelve seconds from anywhere in your material, put them at the front, and cut to your actual opening. The moment does not need explaining — it works because it is interesting, and the explanation arrives in the next thirty seconds.

Two cautions. The cold open must be from the actual piece and must actually be paid off; using your best moment as bait and then delivering something else is the mechanism of clickbait, and it trains an audience to distrust you. And it must be a *moment*, not a summary — a summary at the front removes the reason to watch.

The first frame is a thumbnail

Whatever your first frame is, it will be shown as a still somewhere — in a feed, in a preview, as an autoplay poster. A black frame or a fade-from-black gives the platform nothing to show.

Start on a real image. If you want a fade, fade in over the first shot rather than from black into it.

Titles are not an opening

A full-screen title card at the start costs you three to five seconds of the most expensive attention you will ever have, and it tells the viewer something the platform already told them.

If the piece needs a title, put it over the first shot, or a few seconds in once something has happened. The BBC, YouTube's most-watched creators and most feature films all do this. A title sequence at the front is a convention from an era when people had already chosen to sit down.

The specific numbers, honestly

Platform advice circulates as hard rules — "the first three seconds decide everything" — and the honest position is that the numbers move, differ by platform, and are mostly inferred from creator-side dashboards rather than published. What is robust across all of them:

- The drop-off is steepest at the very start and flattens after roughly the first tenth of the piece.
- Openings that state the subject retain better than openings that delay it.

- Retention is measured against whatever the algorithm is optimising this quarter, which changes.

So treat the curve as feedback on your own work rather than as a law. The useful practice is to look at your own retention graph, find the moment of the sharpest drop, and watch the ten seconds before it. That is where you lost them, and it is usually something identifiable: a long pause, a wandering sentence, a section that had not earned its place.

The rest of the piece has openings too

Every section transition is a smaller version of the same decision. When you move from one subject to the next, the viewer re-asks whether to stay. Give each section its own small hook — a question, a surprising image, a change of pace — rather than a flat announcement of what is coming next.

The signposting version ("now let's look at the three main reasons") is worth avoiding for the same reason as the summary: it tells people what they are going to get, which is precisely the information that makes staying optional.

The honest tension

There is a real conflict between retention craft and honesty, and it is worth naming rather than pretending it is not there. Every technique in this lesson can be used to make a piece more accessible to a viewer who would genuinely have valued it, and every one can be used to hold somebody's attention on something that will not repay it.

The distinction is whether the promise gets paid. A cold open that delivers is craft. A cold open that does not is bait, and the fact that it works is not a justification.

Today

Open your last published piece and watch the first fifteen seconds as though somebody sent you the link. Write down what you knew at second three, second eight and second fifteen. If the answer at second eight is still nothing, you have found the edit.

BEFORE YOU MOVE ON

A creator moves their best moment to the front as a cold open and retention improves sharply — but average view duration falls and comments turn negative. What has gone wrong?

A. The cold open is too long, so viewers who stay are reaching the real opening already tired of the piece.
B. The best moment now appears twice, and the repetition is what viewers are objecting to.
C. Retention and average view duration measure essentially the same thing, so one of the two figures is simply being read incorrectly off the creator dashboard.
D. The opening promises something the rest of the piece does not deliver, so people arrive on the strength of the hook and leave when the payoff does not come.

The answer and why: p. 455

CASE STUDY · MODULE 2

Thirty jump cuts, or half a day back in Ranchi

An illustrative case, built from documented patterns.

A four-person health NGO in Ranchi wants a six-minute film about a community health worker for its annual report and its donor page. One camera, one afternoon, one interview.

The material is forty-one minutes of interview with Sunita, who has been a health worker in the same three villages for nine years. It is good. She is funny about the first year, precise about what changed, and quiet about a child she could not get to a hospital in time. The camera was locked off in her front room, one angle, mid-shot, from the moment she sat down to the moment she stood up.

Beside it in the folder are two cutaways, both filmed in the last four minutes before the light went: her register of home visits, and a wide of the lane outside.

The radio edit comes down to five minutes forty of speech that holds together end to end with the screen off. That part goes well. The problem is what it looks like.

To get from forty-one minutes to five and a half, the editor has made about thirty removals. Each one is a jump cut: same framing, same chair, Sunita's head in a slightly different position on either side of the join. Two of the thirty are covered by the register shot, and one by the lane, because a two-second cutaway cannot be used more than about once every ninety seconds without drawing attention to itself.

The NGO's communications lead watches the rough cut and says it looks cheap. She is not wrong, and she is not able to say why. What she is reacting to is that Sunita moves quite a lot while she talks, so the jumps are large, and that after twenty of them the front room has stopped existing as a place.

The editor lays out three options and what each one costs.

Leave the jump cuts and punch in on alternate joins. The source is 4K and the delivery is 1080p, so scaling the incoming clip to 115 per cent and reframing from centred to a third is free in quality. Cost: about two hours. It converts roughly half the jumps into something that reads as a second camera. It does not fix the other half, and it does not give the film a single shot of anything other than Sunita's face.

Go back and shoot pickups. Ranchi to the village is two hours each way on a good day. Sunita does not need to be there for most of the list: the register, her hands, the doorway, the lane, the bicycle, the health sub-centre, the two women she visits, the medicines box. That is a shot list of about eleven items, all of them ten seconds each, all of them achievable in ninety minutes on site. Cost: one full day of the editor's time, about eighteen hundred rupees in travel, and a week of delay, in a month where the annual report has a printing deadline behind it.

Use stills and graphics. The NGO has photographs from the last four years. A held photograph with a slow push is a real cutaway and costs nothing. Cost: the photographs were taken by three different people, two of whom no longer work there, and nobody has a written note of who owns them.

The communications lead's instinct is the first option, because it costs nothing and does not move the deadline. The editor's instinct is the second, because he has counted: a six-minute film wants roughly twelve usable cutaways, he has two, and no amount of scaling manufactures the other ten.

What neither of them can do is invent a reverse angle. Nobody filmed the interviewer, and nobody filmed the women Sunita visits. Those shots are not recoverable by any technique in the edit.

YOUR ANSWERS

1. The communications lead said the rough cut looked cheap. Why is that a symptom report rather than a diagnosis?

2. Why did the punch-in pass fix about half the joins and not the rest?

3. The editor said a six-minute film wants about twelve cutaways. What is that number actually a budget for?

STOP HERE

Write your answers before you turn the page. How it played out starts on p. 108.

CASE STUDY · MODULE 2

How it played out

They did the punch-in pass first, because it was cheap and because it made the rough cut watchable enough to be judged. Then the editor drove out for one day and shot the eleven-item list, which took seventy minutes on site. The finished film has fourteen cutaways in five minutes forty, four of which came from that trip and are the ones the NGO now uses as photographs as well. The photographs from the archive were not used: nobody could establish who took two of them. The annual report slipped by five days. The NGO's next brief specified that a cutaway list would be shot on the same day as every interview, and the editor now sends that list to whoever is holding the camera before the shoot rather than after it.

The questions, discussed

1. The communications lead said the rough cut looked cheap. Why is that a symptom report rather than a diagnosis?

She was describing an effect she genuinely felt and naming a cause she had no way of identifying. The actual mechanism was that thirty jump cuts on a subject who moves a lot gave her eye a new position to re-acquire at every join, and that after twenty of them the room had never been allowed to exist as a place. Acting on the word cheap would have led to better graphics or a grade. Acting on the mechanism led to coverage.

2. Why did the punch-in pass fix about half the joins and not the rest?

A punch-in changes the framing, so the incoming shot reads as a different shot rather than as a discontinuity in the same one. It works when it alternates: punching in twice in a row, or punching in on every join, simply establishes a second framing which then has its own jumps. It also cannot supply a change of perspective, because the lens compression is identical, and it cannot supply an image of anything that is not already in the frame.

3. The editor said a six-minute film wants about twelve cutaways. What is that number actually a budget for?

It is a budget for removals. Every cutaway is a licence to take time out of the interview without a visible join, plus cover for a cough or a fluffed sentence, plus a change of frame that stops one face becoming monotonous. Roughly one usable cutaway per thirty seconds of finished film is what lets an editor compress forty-one minutes to five without the compression itself becoming the thing the viewer watches.

MODULE 2 TEST

Check what stuck

6 questions, one right answer each. This one is for you, not for a record. The answers, and the reason behind each, are in the key on p. 449. If two go wrong, re-read the module before the exam.

1 Someone reaches for a cup. The movement takes about a second and a half. Why is holding the shot to the end of the movement usually a second too long?

A. A shot longer than one second at 25 frames per second reads as a held frame
B. The viewer understood 'reaching for a cup' within a few frames
C. The movement decelerates at the end, so the last frames carry motion blur
D. Most editors cut on the movement rather than after it, so the eye expects it

2 Cutting on action, editors start the incoming close shot two or three frames earlier than where the wide left off, so the movement slightly repeats. Why?

A. It gives the editor handles for a dissolve if the cut does not work
B. Frame rate conversion drops frames at every cut, and the overlap puts them back
C. It hides the fact that the two cameras were at different distances
D. A real body decelerates, so an exact frame match reads as a small stutter

3 A dialogue scene where every picture cut lands on the same frame as its audio cut sounds like a slideshow. What does moving the audio edge past the picture edge change?

A. The audio is compressed less, because fewer cuts means fewer transients
B. The viewer hears the next speaker before the picture, which shortens the scene
C. Sound runs continuously across the join, so the cut has nothing to announce it
D. The picture and sound are no longer linked, so the editor can trim each of them separately

4 A cutaway is the right length and the right moment is inside it, but it should arrive half a second later. Music is already cut to picture. Which trim?

A. Slide
B. Ripple
C. Roll
D. Slip

5 A cut feels wrong and you cannot say why. Murch's ranking suggests where the answer usually is. Where?

A. In rhythm, because rhythm is the only one with a number attached to it
B. In eye-trace, because the eye has to re-acquire the subject at every cut
C. In the bottom two, screen plane and physical space, because those are the ones you can see
D. In the top two, emotion and story

6 Why does a six-minute interview film need roughly twelve usable cutaways rather than two?

A. Because each cutaway must be under two seconds, so more are needed to fill the time
B. Because a cutaway is what lets time be removed without a visible join
C. Because platforms reward a change of frame every thirty seconds
D. Because the same cutaway cannot be used more than once in a single piece

3

MODULE 3

Structure — deciding what the film is

Take unstructured material to a defensible rough cut — a spine stated in one sentence, a chosen shape with a reason, an interview compressed without changing what the speaker meant, a screening run to produce specific notes, and a documented position on every piece of third-party material in it.

Lesson 21 · 8 min

One sentence, or you do not have a film

THE ONE THING TO KEEP

Write what the piece is about as one sentence containing a question or a change, and hold every uncertain clip against it, because the sentence is what makes the decision about a good shot that does not belong.

The test

Before the assembly, write down what the piece is about in one sentence, in plain language, with a verb in it. Not a topic. A claim or a question.

- Topic: "a film about our new clinic". This is a subject heading. Anything can be in it.
- Spine: "a doctor who spent eleven years trying to open a clinic in a district that had none, and what it cost her". Now some material belongs and most does not.

The value is not the sentence. It is that the sentence makes decisions for you. Every clip you are unsure about gets held against it: does this serve the sentence? If not, it goes, however good it is. That single mechanism resolves most of the arguments an edit has with itself.

Why people skip it

Because it feels premature. You have not watched everything, the material might say something you do not expect, and committing early seems like a way to miss the real story.

That instinct is right about *timing* and wrong about *whether*. Write the sentence after the logging pass, not before the shoot. And expect to rewrite it — twice, three times, as the material teaches you what it has. A spine you revise four times is doing its job. The failure is never having one, which is how a twelve-minute film about a clinic ends up containing four minutes about the building contractor.

The two shapes a spine can take

A question. "Why do so many of this town's trained nurses leave within two years?" Good for investigation, explanation and anything with a genuine uncertainty in it. The piece is over when the question has been addressed.

A change. "She started with a rented room and no equipment, and this is what happened." Good for profile, process and anything with a before and an after. The piece is over when the change has completed.

If your sentence is neither — if it is "we look at several aspects of X" — you have a list, not a structure, and lists are the hardest thing to hold an audience through.

Testing a spine against the material

Three checks, all quick.

Can you name the moment it turns? Most pieces have a hinge: the thing goes wrong, the answer arrives, the person changes their mind. Find it in your selects. If there is no hinge anywhere in the material, the piece will be flat no matter how it is ordered, and that is worth knowing on day one rather than day five.

Can you say what the viewer knows at the end that they did not at the start? If the honest answer is "nothing, but it was nice to watch", that is a legitimate kind of piece — a mood film, a montage — but it needs to be built like one, and it needs to be short.

Does the material actually support it? This is where spines die. You want the film to be about her determination, and every single answer she gave was about logistics. Either the sentence changes or you go back and interview her again.

The sentence goes on the wall

Literally. Write it at the top of your notes file, in the project description field, on a sticky note on the monitor. You will drift, because the material is seductive and there are three lovely sequences that do not belong to it.

Editors talk about "killing your darlings". The spine is the thing that tells you which ones are darlings.

Where it gets harder

Client work. The client's sentence and yours may differ. Get theirs in writing at the brief stage — the working-editor block deals with this — and if the material will not support it, say so before you start rather than after.

Corporate and marketing pieces. The spine is often "this product solves this problem", which is fine, and the failure mode is a piece that tries to say six things about the product at once. One sentence, one claim.

Vlogs and personal pieces. The spine is easy to skip because the audience already likes the presenter. It still helps, and the difference between a vlog with a spine and one without is the difference between eight minutes that end and eight minutes that stop.

Events — weddings, conferences, performances. The apparent spine is "what happened", which is chronology, not structure. The better sentence usually involves a person: whose day this was, and what the day was like for them.

A worked example

Material: three hours from a two-day shoot at a small printing workshop.

Figure 3.1

One shoot, three spines, narrowing

A film about a printing workshop
A subject heading. Every frame qualifies, so nothing can be cut.

↓

How offset printing works
A process film. The interviews about the owner's father are now out.

↓

The last press in the district, and the man who will not sell it
A tension, a hinge and an ending. The process footage becomes the evidence.

The third sentence uses less of the material than the first two and produces a better film. That is the normal direction of travel.

- First attempt: "a film about a printing workshop". Useless; every frame qualifies.
- Second: "how offset printing works". Better — now it is a process film, and the interviews about the owner's father are out.
- Third, after watching: "the last offset press in the district, and the man who will not sell it". Now there is a tension, a hinge and an ending, and the process footage becomes the *evidence* rather than the subject.

Notice that the third sentence uses less of the material than the first two, and produces a better film. That is the normal direction of travel.

Today

Take the last thing you edited and write its spine in one sentence, honestly, after the fact. Then list the sections that do not serve it. That list is what a spine would have saved you.

BEFORE YOU MOVE ON

An editor writes the spine "a film about our company's new warehouse" and finds that the rough cut keeps sprawling — every section seems defensible and the piece runs twice its target length. What is wrong with the sentence itself?

A. It is too short to guide decisions; a spine needs to name the audience, the platform and the intended running time before it can be useful for structuring anything.
B. It names a topic rather than a question or a change, so every piece of material about the warehouse satisfies it equally and nothing can be excluded.
C. It is about a place rather than a person, and structures built around locations cannot hold an audience.
D. It was written before the logging pass, so it reflects assumptions rather than what the material actually contains.

The answer and why: p. 457

Lesson 22 · 9 min

Six shapes, and what each one costs

THE ONE THING TO KEEP

Chronology is a choice rather than a default, and each shape has a specific failure — thematic sections give the viewer a clean exit at every join, lists have no momentum, and essays become lectures unless every claim has evidence on screen.

Shape is a choice, not a default

Most unstructured material gets cut chronologically because that is the order it arrived in. Chronology is one option out of several, and it is frequently the weakest. Here are the shapes that recur, what each is good at, and the specific way each fails.

Chronological

Events in the order they happened.

Good for: process films, events with a genuine arc, anything where causation runs forward and the viewer needs to see it.

Fails when: the interesting thing happens at the end. An hour of setup for a two-minute payoff loses its audience before the payoff. The standard repair is a cold open — put the payoff at the front, then "three months earlier".

Thematic

Grouped by subject: here is the money problem, here is the staffing problem, here is the regulation problem.

Good for: explanatory work, anything with several independent strands, interviews where several people address the same topics.

Fails when: nothing connects the sections, so each transition is a full stop and the viewer can leave cleanly at every one of them. Repair: end each section on a question the next one answers.

Problem and solution

Establish something wrong, then work toward the fix.

Good for: tutorials, marketing, case studies, anything instructional. It is the most reliable structure for teaching because it supplies motivation before information.

Fails when: the problem is manufactured. A viewer who does not recognise the problem as theirs disengages immediately, and an exaggerated problem reads as a sales pitch. Repair: state the problem in the audience's own words, not the product's.

The essay

A claim, then evidence, then a conclusion. Often narrated.

Good for: argument, analysis, opinion, video essays.

Fails when: it becomes a lecture with pictures. The remedy is that every claim needs something on screen that *is* the evidence, not a stock shot of somebody typing.

The profile

A person, through what they do.

Good for: almost anything, which is why documentary reaches for it constantly. People are interesting and processes are not, so attaching a process to a person is the most reliable move in the form.

Fails when: the subject is not candid, or the film becomes a tribute. A profile with no difficulty in it is an advertisement.

The list

Five tips, seven mistakes, ten things.

Good for: search-driven and feed-driven work. The viewer knows what they are getting and how long it will take, and they can leave and return.

Fails when: it is the default rather than a choice. A list has no momentum — each item is independent, so there is never a reason to watch the next one. It also flattens material that had a real structure in it. Repair, if you must list: order the items so each builds on the last, and say so.

Choosing

Three questions:

1. **Does the material have a hinge?** If yes, chronological or profile will probably use it. If no, thematic or essay will hide its absence.

2. **What does the viewer want?** Somebody searching "how to fix a dripping tap" wants problem-and-solution starting immediately. Somebody who followed a link about a printing workshop wants a profile.
3. **How long is it?** Under two minutes, almost everything works as a single unbroken idea. Over eight minutes you need sections, and sections need a shape.

Sections, and the joins between them

Whatever shape you choose, a piece over about five minutes breaks into sections of two to four minutes. The joins are where you lose people, so they need work.

Things that make a section join hold:

- **A question at the end of the outgoing section**, answered at the start of the incoming one.
- **A change of texture** — from interview to sequence, from voice to music, from close to wide. Sameness across a join makes the join feel like more of the same.
- **A hard reset**, used sparingly: black, a title, a beat of silence. This works precisely because it is a full stop, so use it where you *want* the viewer to reorient.

What does not work: a slow fade to black and up again, which reads as the end of the piece, and a narrated announcement of what the next section will contain.

Restructure on cards, not in the timeline

When a structure is not working, do not push clips around. Write each section on a card or a line in a document — one line, what it is, how long it is — and rearrange those. A twelve-minute piece is fifteen lines. You can try six structures in twenty minutes on paper and none in twenty minutes on a timeline.

Then implement the winner once.

Today

Take a piece you have made and write its sections as one line each. Then try re-ordering them into a different one of the six shapes above. You will usually find at least one arrangement better than the one you shipped.

BEFORE YOU MOVE ON

A ten-minute explainer is cut thematically into four sections. Retention shows a sharp drop at each section boundary but is healthy within sections. What does that pattern indicate, and what is the structural repair?

A. The sections are too long, and cutting each one shorter will smooth the retention curve across the whole piece.

B. The thematic structure is wrong for the material and it should be recut chronologically so events lead into one another naturally rather than being grouped by subject.

C. Each section ends as a complete unit, so the viewer reaches a natural stopping point four times; ending sections on an unanswered question removes the clean exit.

D. The transitions are visually abrupt, and adding dissolves or title cards at each boundary will keep viewers moving through them.

The answer and why: p. 458

Lesson 23 · 10 min

Compressing what somebody said without changing what they meant

THE ONE THING TO KEEP

You may change what somebody said and not what they meant — so removing a qualification, joining clauses from different answers or cutting the 'no' are fabrications regardless of how clean the join is.

The power you have

A forty-minute interview becomes ninety seconds. You chose which ninety. You removed the qualification, joined two answers, cut the "but" that came after the bit you liked. Nothing you did was technically difficult and none of it is visible in the finished piece.

This is the part of editing with actual ethical weight, and it is barely taught. Broadcasters have codes for it. Most people editing interviews have never read one.

The line

The useful formulation: **you may change what somebody said; you may not change what they meant.**

Shortening, removing repetition, removing an aside, tightening a rambling answer — all normal, all expected by anyone who has been interviewed. Nobody believes the thirty-second clip was thirty seconds in the room.

What crosses the line is anything that alters the proposition:

- **Removing a qualification.** "In most cases, and assuming funding continues, we can do it" becomes "we can do it". The sentence is now a promise it never was.

- **Joining clauses from different answers** so the speaker appears to say something they did not. Broadcast calls the result a *Frankenbite*. The join can be perfect and the result is still a fabrication.
- **Reordering so cause and effect reverse.** They said the delay caused the cost overrun; the cut implies the overrun caused the delay.
- **Cutting the answer to a different question** and placing it under the question you preferred to have asked.
- **Removing the "no".** "Did you approve it?" "No — well, I saw the paperwork." Cut the "no" and you have a confession.

The test to apply

After the cut, ask: **if the speaker watched this, would they say "that is not what I meant"?**

Not "would they be annoyed" — people are often annoyed by accurate reporting. Would they be able to point at a proposition in the cut that they do not hold? If yes, recut.

A second, harsher test for anything contested: could you defend this cut against the full transcript, in front of the person, with a third party reading both? If the answer depends on them not having the transcript, you already know.

What the codes say

The BBC's Editorial Guidelines and Ofcom's Broadcasting Code both permit editing for length and clarity and both prohibit editing that materially misrepresents. India's Cable Television Networks Rules and the various self-regulatory codes for digital news carry similar accuracy provisions. The specific obligations differ by country and by whether you are a broadcaster, a publisher or an individual, and none of this is legal advice — but the underlying standard is remarkably consistent across all of them, because it is not really a legal standard. It is the condition under which interviews remain possible at all.

Techniques that keep you honest

Keep the question in. An answer without its question can be moved anywhere. An answer with its question is anchored.

Mark every join. During the radio edit, place a marker at every point where you joined non-adjacent material. Review that list before you lock. Most problem cuts announce themselves when you look at them as a list rather than in flow.

Use a visible device for a deliberate compression. A jump cut, a brief dissolve, a change of framing — anything that signals to the viewer that time was removed here. It costs nothing and it is the difference between compression and concealment.

Send the cut, where you can. Not editorial control — you are not obliged to give a subject approval over a piece, and in journalism you generally should not. But for corporate, educational and personal work, letting somebody see how they were used is normal courtesy and catches genuine errors.

Keep the rushes and the transcript. For any project where the subject matter is contested, keep the originals until well past publication. The defence against "I never said that" is the recording.

The consent question, separately

Editing ethics assume the person agreed to be filmed. Worth checking that this is true, and that they understood what for.

Somebody who agreed to an internal training video did not agree to a public advertisement. Somebody who spoke about their health to a small community project did not agree to a piece that will be indexed by search engines. Scope matters, and re-use outside the agreed scope needs a new yes.

This lands hardest on vulnerable subjects: patients, children, people in legal difficulty, people who are visibly distressed. The extra question there is not whether they consented but whether they were in a position to weigh what consent meant. If you would not be comfortable showing them the finished piece, that is your answer.

The synthetic line

Several tools will now generate speech in somebody's voice, or alter a recording so that the words change. Applied to an interview, this is not editing. It is fabrication with an editing interface, and there is no length of clip at which it becomes acceptable in a factual piece.

Generating a word the person did not say, smoothing a sentence they fumbled into one they did not utter, translating their answer into a language they do not speak using their own voice — each of these produces a recording of something that did not happen. Several countries now require labelling of synthetic media, and the requirements differ and are moving. Labelling is a floor, not a permission.

Today

Take an interview cut you have already made and list every point where you joined non-adjacent material. Read the original transcript around each one. This takes fifteen minutes and it is the habit worth keeping.

BEFORE YOU MOVE ON

An editor joins two sentences from different parts of an interview into one continuous answer. The join is inaudible, both sentences are the speaker's own words, and neither has been altered. When is this nevertheless a misrepresentation?

A. Whenever non-adjacent material is joined at all, since any splice of separated sentences presents a sequence of speech that never occurred in that form during the interview itself.
B. Only if the speaker objects after publication, since they are the authority on their own meaning and no rule can anticipate it.
C. When the combination asserts a proposition the speaker does not hold — because a condition, a denial or a causal direction was dropped at the join.
D. Only if the two sentences came from answers to different questions, which is what makes the result a fabricated answer rather than a shortened one.

The answer and why: p. 458

Lesson 24 · 8 min

The stages, and how long each really takes

THE ONE THING TO KEEP

Assembly, rough cut, fine cut, lock and finishing exist because work done in one stage is thrown away if the previous stage was not finished, and unscripted material realistically costs three to five hours of editing per finished minute before any finishing.

Why the stages exist

The first lesson of this course named them: assembly, rough cut, fine cut, picture lock, then finishing. They are not bureaucracy. Each one exists because the work done in it becomes worthless if the stage before it is not finished.

- Grade a shot before picture lock, and you re-grade it when the shot gets shorter.
- Mix before picture lock, and every trim moves the mix.
- Add titles before structure is settled, and you re-time every one.
- Cut music in during the assembly, and you start protecting the music.

The cost of getting this wrong is not a mistake you can see. It is doing work twice.

Figure 3.2

The five stages, and what each one is for

- **Assembly**
  Everything usable, in order, ugly. Two to four times the target length.
- **Rough cut**
  Whole sections removed and reordered. Decisions at the paragraph level. This is where the film is made.
- **Fine cut**
  Every join trimmed to the frame, J and L cuts built, pace worked through.
- **Picture lock**
  No more picture changes. The timeline is duplicated and named for the lock.
- **Finishing**
  Sound design and mix, colour, titles, captions, graphics, export.

Work done in one stage is thrown away if the stage before it was not finished. Grade before lock and you regrade every shot you later shorten.

Assembly

Everything usable, in order, roughly. Long and ugly. No transitions, no music, no titles, no colour.

The target: **everything that might be in the film is in the timeline.** If a shot is not here, you will forget it exists.

Expect the assembly to be two to four times your target length. A twelve-minute film assembles at thirty to forty minutes. If your assembly is already close to target length, you have been cutting rather than assembling, and you have made choices without seeing the alternatives.

Time: roughly a day per ten minutes of assembly, from logged material. Much less if you built a selects timeline.

Rough cut

Structure fixed, length roughly right, joins still messy.

This is where the film is made and it is the stage people rush. The work is: remove whole sections, reorder, decide what the piece is. You are making decisions at the paragraph level, not the frame level.

The temptation is to polish a sequence you like. Do not. It will probably be cut, and if it is not, it will be trimmed anyway.

Expect two or three rough cuts. The first is too long, the second has the right structure and is still too long, the third is close.

Time: as long as the assembly again, at least.

Fine cut

Every join trimmed to the frame. J and L cuts built. Pace worked through. The six-frame pass.

Now you are at the frame level, and now the trimming lessons pay for themselves.

Time: about half the rough cut, and it goes fast if the rough cut was honest.

Picture lock

A declaration: no more picture changes. Everybody who needs to sign off has signed off.

It is a real threshold, not a formality, because everything after it is expensive to redo. In a team, lock is announced and the timeline is duplicated and named `_LOCKED`.

If a picture change is genuinely required after lock — and it sometimes is — the honest handling is to make the change *without altering the duration*, so the mix and the titles still line up. Replace a shot, do not shorten one.

Finishing

Sound design, mix, colour, titles, captions, graphics, export. In that order, roughly, though sound and colour can run in parallel because they touch nothing in common.

Time: for a twelve-minute piece, one to three days depending on how much graphics work there is.

The honest ratios

For planning, on documentary-style material with a single editor:

- **Logging**: one hour per hour of rushes, roughly.
- **Assembly to fine cut**: three to five hours of work per finished minute, on unscripted material. A ten-minute film is thirty to fifty hours.
- **Finishing**: two to four hours per finished minute if there are graphics, less if not.

Scripted or tightly briefed work is much faster — one to two hours per finished minute is realistic. Heavily interview-driven documentary can be far slower; ten hours per finished minute is not unusual on complex material.

These numbers exist so you can quote a job. The commonest cause of an unprofitable edit is a quote based on how long it takes to cut a good sequence, multiplied by the number of sequences, with no allowance for the three rough cuts.

Working with someone else's schedule

A client wants to see something early. Show them the **rough cut**, not the assembly — an assembly makes a piece look bad in ways that have nothing to do with how it will turn out, and you will spend the review explaining that.

And tell them what stage they are looking at. "This is a rough cut: structure is what I need your view on, the colour and sound are not done and the titles are placeholders." Without that sentence you will get notes about the font.

Today

Estimate the hours in the last piece you edited, divide by its finished minutes, and write down the number. That is your own ratio, and it is the one to quote from.

BEFORE YOU MOVE ON

An editor's assembly of a planned twelve-minute film comes out at fourteen minutes. Why is that a warning sign rather than a good sign?

A. They made selection decisions without seeing the alternatives in context, so the rough cut has no room left in which to be structured.

B. A fourteen-minute assembly means there was not enough material shot, and the film will have to be padded to reach its target length.

C. Assemblies are supposed to come in shorter than the target rather than longer, so a fourteen-minute assembly is a clear indication that the material was never trimmed on the way in at all.

D. The timeline will be slow to work with at that length, which suggests the editor skipped proxy generation during the assembly stage.

The answer and why: p. 459

Lesson 25 · 8 min

Watching it with someone who has not seen it

THE ONE THING TO KEEP

You cannot watch your own edit because you know what is coming, so run a silent, uninterrupted screening with a stranger to the material and treat every note as a symptom report rather than a diagnosis.

The problem you cannot solve alone

By the third day you have lost the ability to watch your own edit. You know what is coming, so you cannot feel length. You know what the mumbled word is, so you cannot hear that it is unintelligible. You know who everyone is, so you cannot notice that nobody was introduced.

This is not a failure of attention. It is that you have information the audience does not, and there is no way to unknow it. The only reliable instrument is somebody else's first viewing.

Set it up properly

Show it once, uninterrupted, start to finish. No pausing, no commentary, no "this bit isn't finished yet". The whole value is in the experience being continuous.

Tell them what stage it is before you press play, in one sentence: "rough cut, no colour or sound work, titles are placeholders, I want your view on whether it makes sense". This is the difference between notes about structure and notes about the font.

Watch them, not the film. You have seen the film. Sit slightly behind and to the side. Note the timecodes where they shift, look away, check their phone, lean in, or laugh. Those are data and they are more honest than anything they will say afterwards.

Do not defend anything. The moment you explain why a section is there, you have stopped receiving information. Write it down and move on.

The questions that produce useful answers

General questions produce general answers. "What did you think?" gets "it was good". These get specifics:

- **"Tell me what happened, in order."** If they cannot, the structure is not landing. If they leave out a section, that section is not doing anything.
- **"Who was that person?"** Tests whether your introductions work.
- **"Where did you get bored?"** People will tell you honestly if you ask directly and do not react.
- **"Was there anything you did not understand?"** Ask this even when the piece seems simple. The answer is often a term you did not notice you had used.
- **"What was it about?"** Compare their sentence to your spine. The gap is the note.
- **"What would you tell someone about it?"** Reveals what actually stuck.

Avoid: "Did you like the music?", "Was it too long?", "Should I cut the bit about X?". Closed and leading questions produce agreement rather than information.

Reading notes correctly

The most useful principle in receiving creative feedback, and it comes from a book about screenwriting rather than editing: **people are almost always right that something is wrong, and almost always wrong about what it is.**

So treat every note as a symptom report, not a diagnosis.

- "The middle is boring" often means the middle has no question in it, not that its shots are too long.

- "I didn't like her" often means she was introduced badly, or her strongest material comes too late.
- "The music is wrong" often means the pacing is wrong and the music is what they noticed.
- "It's too long" is almost always true, and almost never about the place they say it is. Look at what precedes their boredom, not at the boring bit.

The corresponding discipline: when somebody proposes a fix, thank them and do not implement it. Find the cause.

Who to show it to

Someone who does not know the subject. This is the test that matters, because the audience will not know it either. A colleague who was at the shoot cannot see the piece any better than you can.

Two or three people, separately. A group produces consensus, and consensus hides the individual reactions that are the useful part. One person nodding along changes what the next one says.

Not the client, yet. The client sees a cut you have already fixed with this feedback.

What to do with contradictory notes

Two viewers, opposite reactions. This is normal and does not mean the notes are worthless.

The resolution: where notes conflict, go back to the spine. Which of these two reactions is about the piece you are making? A viewer who wanted more technical depth in a profile is describing a different film, and you may decline the note — but write down that you declined it and why, because if a third person says it, they are not describing a different film, they are describing a gap.

After the screening

Do nothing for a day if the schedule allows. The instinct immediately after notes is to fix everything at once, and half the notes will look different tomorrow.

Then write a list, ordered by size: structural changes first, then section-level, then shot-level. Work top down. A structural change makes several of the smaller notes disappear.

Today

Show something you have made to one person who has not seen it, ask only "tell me what happened, in order", and write down what they leave out.

BEFORE YOU MOVE ON

Three separate viewers say the music in the middle section is wrong. The editor tries two different tracks and both viewers of the new version say the same thing. What is the most likely explanation?

A. The music is genuinely wrong in genre for the piece, and the editor should try a third track from a different category before concluding anything else.
B. Something else in the middle section is not working — most likely its pace or its lack of a question — and the music is simply the most nameable thing people noticed.
C. The music is simply mixed too loud against the dialogue underneath it, so viewers are registering a level problem and reporting it to the editor as a complaint about the choice of track.
D. Three viewers agreeing means the note is reliable, so the section should be cut entirely rather than rescored.

The answer and why: p. 459

Lesson 26 · 8 min

The twenty per cent that has to go

THE ONE THING TO KEEP

A first fine cut is typically a fifth too long, and the material to remove is whatever is justified by your knowledge of the project rather than by its effect on a stranger — so park it rather than deleting it and watch the two versions.

The default failure of every first edit

It is too long. Not slightly — typically by a fifth to a third. This is so consistent that you can treat it as a starting assumption rather than a diagnosis.

The reason is structural, not a lapse of judgement. Every section in your cut is there because you had a reason to put it there. The viewer does not have your reasons; they have only what is on screen. Material that is justified by your knowledge of the project rather than by its effect on a stranger is exactly the material that feels slow.

The exercise

Take your fine cut and remove twenty per cent of its running time. Not by trimming frames — by removing whole units. A section, a sequence, an answer, a shot.

Do it as a duplicate timeline so nothing is lost. Then watch both.

The result, nine times out of ten, is that the shorter one is better and you cannot reconstruct what you removed. That is the finding, and it recalibrates your judgement permanently.

Which twenty per cent

These are the reliable candidates, roughly in order:

Anything that repeats a point already made. Especially second examples. The first example teaches; the second confirms something the viewer already accepted.

Everything before the actual beginning. Most pieces truly start one to two minutes in. Watch your opening and find the first moment where something happens. Everything before it is usually preamble.

The section you had to work hardest on. If a sequence took three days to make work, that is normally a sign the material does not support it. The effort is a symptom.

The shot you are proud of that does not serve the spine. The drone move. The slow-motion detail. The one with the beautiful light. These are darlings and they are the hardest to remove precisely because they are good — the quality is what makes you keep defending them.

Explanations of things nobody asked about. Background, context and history that you found interesting during research.

The ending after the ending. Most pieces end and then continue for twenty seconds. Find the last moment that matters and cut there.

Why it is hard, specifically

Two mechanisms, both worth naming because naming them helps.

Sunk cost. You spent four hours on that sequence. Those hours are gone whether you keep it or not, and keeping a weak sequence does not recover them — it spends the viewer's time as well.

You are the only person who will ever see the alternative. Nobody watching the finished piece knows what was removed. There is no version of the audience that misses your drone shot. The only person who experiences the loss is you, and you are not the audience.

The parking file

The practical defence against the pain: never delete, park.

Make a timeline called `offcuts` or `parked` and drop everything you remove into it. It costs nothing, it takes five seconds, and it removes the finality that makes people fight to keep things.

Twice in a career you will go back and retrieve something from it. The rest of the time it simply makes the removal easy, which is its actual job.

The parked material is also genuinely useful afterwards: it is where the short-form cuts, the social clips and the trailer come from.

When length is not the problem

Two honest exceptions.

When the piece is already short. A ninety-second video cut to seventy-two seconds is a different kind of decision, and at that length the risk is under-explaining rather than over-running.

When the slowness is deliberate and load-bearing. A long take of somebody thinking, a landscape held past comfort, a silence after bad news. These are slow on purpose, and the twenty per cent exercise will suggest cutting them because it cannot tell intention from padding. Protect them consciously — by which is meant: decide, in words, why each one is there, and if you cannot, it goes.

Applying it to somebody else's cut

When you are asked to shorten a piece you did not make, the same order applies, with one addition: ask what the spine is before you start. Without it, the shortening is guesswork and you will remove the thing the piece was for.

Today

Duplicate a finished timeline, remove twenty per cent of it by whole sections, and watch both versions back to back. Whichever wins, you have learnt something you cannot learn by reading.

BEFORE YOU MOVE ON

An editor identifies a sequence that took three days to make work and considers it a reason to keep it in the cut. Why is that reasoning backwards?

A. Because time already spent cannot be recovered by keeping the sequence, and difficulty getting it to work is usually evidence that the material does not support it.
B. Because sequences that take a long time to build accumulate more technical errors, and those errors will be visible to an audience watching at full speed on a large screen.
C. Because the editor is too close to it after three days to judge it, so the decision should simply be deferred to a viewer at the screening.
D. Because anything that required that much work must have been reconstructed from material shot for another purpose, which always shows on screen.

The answer and why: p. 460

Lesson 27 · 9 min

Narration that is not a description of the picture

THE ONE THING TO KEEP

Cut the picture first and write narration into the gaps it leaves, supplying only what the picture cannot show — and read every line aloud while writing, because anything you stumble over the listener will stumble over too.

The commonest fault

"Here we see the workers arriving at the factory." Over a shot of workers arriving at a factory.

The viewer has eyes. Telling them what is on screen wastes the only channel you have that can supply something the picture cannot, and it patronises them in the process.

The rule that fixes most narration: **words and pictures should do different jobs.** The picture shows; the narration supplies what the picture cannot — who these people are, what it costs, what happened next, why it matters, what you cannot see.

Over workers arriving at a factory: "Four years ago there were nine hundred of them. Today there are two hundred and ten."

Write to picture, not before it

The order that works: build the picture edit first, then write narration into the gaps it leaves.

Write the script first and you end up illustrating it — finding a shot for every sentence — and the film becomes a slideshow attached to an essay. Editors call this the radio-with-pictures problem, and it is why so many corporate videos feel like a brochure being read aloud.

The practical method:

1. Cut the sequence with its natural sound.
2. Watch it and note where the viewer will have a question.
3. Write one sentence for each question, in the gap where it arises.
4. Record it, drop it in, and cut the sentences you find you do not need.

Step 4 removes about a third of what you wrote. That third is the sentences that described the picture.

Scratch track first

Record yourself reading the draft on a phone, drop it into the timeline, and edit against it. It costs two minutes and it tells you immediately where the words do not fit the picture — where a sentence needs four seconds and the shot lasts two.

Then rewrite the words to fit the picture, rather than re-cutting the picture to fit the words. The picture was built for a reason.

Replace the scratch with the real recording at the end. Keep the scratch in a muted track until you have; it is the reference for timing.

Writing for the ear

Narration is heard once, at speed, with no ability to go back. That imposes constraints that written prose does not have.

- **Short sentences.** One idea each. A subordinate clause that works on the page is a place to get lost when heard.
- **Say the number in speakable form.** "Just under a third" beats "31.7 per cent" unless the precision matters.
- **Front-load the subject.** "The clinic closed in March" is easier to follow than "In March, following a decision by the district authority, the clinic closed".
- **Read it aloud while writing.** Anything you stumble over, the listener will stumble over. This is the single most effective revision technique for narration and almost nobody does it.
- **Do not write what somebody in the film is about to say.** Narration that pre-empts an interview answer removes the reason to hear it.

Recording it for nothing

You need a quiet room, a microphone close to your mouth, and no processing until afterwards.

- **Distance**: 15 to 20 cm, slightly off-axis so plosives pass beside the capsule rather than into it. A sock or a bit of foam over the mic is a usable pop filter.
- **Room**: the smallest soft space you have. A cupboard of clothes is genuinely the best free vocal booth in most homes, because clothing absorbs reflections.
- **Level**: peaks around -6 dBFS, average around -12 to -15.

- **Take the time**: record each paragraph twice and pick. It is far faster than fixing a bad read.
- **Leave five seconds of silence** at the top of every recording for noise profiling and room tone.
- **Audacity** is free and does everything you need for this. So does Resolve's Fairlight page.

A cheap USB microphone at 20 cm beats an expensive one across a room, for the same reason as every other recording in this course: the ratio of direct to reflected sound.

Editing the read

Narration is edited like any other dialogue: remove the breaths that interrupt, keep the ones that are thinking, cut at the gaps between phrases.

One specific: a narration track sounds better with a *little* more space than feels natural while editing. You are reading along with meaning you already know. Leave a beat after each idea.

Synthetic voices, honestly

Text-to-speech is now good enough to use, and there are three separate questions.

Quality. For neutral informational narration it is usable. For anything carrying emotion, it is still audibly wrong in ways listeners notice without being able to name, because prosody encodes meaning and the model is guessing at meaning.

Disclosure. A synthetic narrator in a factual piece is a fact about the piece. Several jurisdictions now require labelling of synthetic media, the requirements differ, and they are moving.

Whose voice. Cloning a specific person's voice needs that person's explicit, scoped consent — for this project, these uses, this period — and generic stock voices carry their own licence terms about commercial use. Motion, Sound and Music goes into the consent question in detail.

Today

Take any narrated video and transcribe thirty seconds of its narration. Count how many sentences describe what is already visible. That count is the lesson.

BEFORE YOU MOVE ON

An editor writes a full narration script, records it, and then cuts picture to fit it. The result feels like a brochure read aloud despite good footage. What is the mechanism?

A. The narration was recorded before the edit, so its pacing is uniform and cannot respond to the rhythm of the sequences it sits over.

B. Each sentence had to be given a shot, so the picture became illustration of the words, and the two now say the same thing at once.

C. Writing first tends to produce longer sentences, and long sentences are hard to follow when heard once at speed without the ability to go back over them.

D. The footage has been cut too tightly to the narration's timing throughout, so no shot is ever allowed to breathe and the whole piece feels rushed rather than considered.

The answer and why: p. 460

Lesson 28 · 8 min

One shoot, several finished pieces

THE ONE THING TO KEEP

A short is not a trimmed long piece but a separate cut with its own spine, opening on the moment rather than the setup, and the cheapest way to get both is to compose and lock the long version first with the vertical crop already in mind.

The economics

The expensive parts of a video are the shoot and the structural edit. Once those exist, a second and third deliverable cost a fraction of the first — if the first was built with the others in mind.

A typical set from one interview shoot:

- The long piece. Eight to twelve minutes, horizontal, for a website or YouTube.
- Two or three vertical shorts, 30 to 60 seconds each.
- A square or vertical cut of 90 seconds for a feed post.
- A 15-second teaser that links to the long piece.
- Stills, pulled as frames, for thumbnails and posts.

That is five deliverables from one edit. Quoting for one and delivering five is how people lose money; quoting for the set is how the work becomes worth doing.

Short is not a trim of long

The mistake is taking the ten-minute film, cutting forty seconds out of the middle, and posting it. It does not work, because the two forms have different requirements.

The long piece can build. It can introduce people, establish context, hold a question open for two minutes.

The short piece has to be complete in itself. It opens on the thing, delivers one idea, and ends. There is no context to draw on because the viewer has none.

So a short is a separate piece built from the same material with its own spine — usually one moment or one claim from the long piece, given a beginning and an end of its own.

Building the short version

1. Find the single strongest 20 to 40 seconds in the long piece. Usually one answer, one demonstration, one moment.
2. Ask what a stranger needs to know to understand it. Usually one sentence — a name, a place, a number. That becomes a caption or a super, not a spoken introduction.
3. Open on the moment, not on the setup.
4. End on the end of the moment. No sign-off, no "watch the full video", no logo — or if there must be a call to action, one line of text over the last two seconds, not a five-second card.
5. Captions, burnt in, in the middle third of the frame.

Reframing 16:9 to 9:16

The mechanics, from the resolution lesson: duplicate the timeline, change its resolution to 1080×1920, and reframe each shot.

Three approaches, in increasing order of effort:

- **Automatic subject tracking.** Resolve's Smart Reframe, Premiere's Auto Reframe, CapCut's equivalent. Fast, and the output is a first pass — it locks onto the wrong subject in any shot with two people or heavy movement. Check every shot.
- **Static reframe per shot.** Decide one position for each shot and leave it. Usually better than a tracker that wanders.

- **Hand-keyframed pans.** For shots where the subject genuinely crosses the frame. Slow, and only worth it on the few shots that need it.

What cannot be reframed: a two-shot where both people matter. Your options are to crop to one of them, to stack the frame with picture above and captions below, or to use a different shot.

If you also shoot, the cheap fix is to compose with both crops in mind — keep the subject in the central square and avoid putting anything essential at the frame edges. It costs nothing and it makes the vertical version automatic.

Where the assets come from

The parked timeline from the previous lesson is where the shorts live. Material cut from the long piece for being tangential is frequently perfect as a standalone short, because tangential in a structure means self-contained on its own.

Pull thumbnails as frame exports from the graded master, not from the camera original — you want the colour to match the film.

Order of operations

Make the long piece first, to picture lock, then derive. Deriving from an unlocked cut means redoing the derivatives when the master changes.

The exception is when the short is the actual deliverable and the long version is secondary, which is increasingly common in commercial work. In that case build the short first with its own spine, and treat the long piece as the derivative.

A note on the platforms

Specifications for length, aspect and file size differ per platform and change several times a year. Do not learn them; look them up at delivery time from the platform's own help pages, and keep a note of what you used per client in the project's docs folder.

What has been stable for several years: 1080×1920 for vertical, H.264 in an MP4, captions burnt in, and the safe-area rule about the bottom fifth.

Today

Take something you have published and build one 30-second vertical cut from it with its own beginning and end. Note how little of the original structure survives.

BEFORE YOU MOVE ON

A team takes their nine-minute film, exports the best 45 seconds of it unchanged as a vertical crop, and it performs badly despite being the strongest section of the original. What did the export miss?

A. The crop lost image quality, since cropping 16:9 to 9:16 discards most of the frame and leaves too few pixels for a sharp vertical delivery.
B. Vertical platforms re-encode more aggressively, so the file needed a higher bitrate rather than a different edit.
C. The section depended on context established earlier in the film, and a standalone short has to supply its own beginning and end because the viewer arrives with none.
D. The captions were not burnt in, so viewers watching muted in a feed received no information from a section that is carried entirely by speech and by the surrounding narrative setup.

The answer and why: p. 461

Lesson 29 · 10 min

Using material you did not shoot

THE ONE THING TO KEEP

Third-party material is licensed, openly licensed, permitted or covered by a statutory exception — and because exceptions differ fundamentally between fair use and enumerated fair dealing, the widely repeated 'ten seconds is fine' rules have no basis anywhere.

The category

At some point your edit needs something you do not have: a news clip, a photograph, a film extract, a screen recording of a website, a map, a piece of music, a frame from somebody's Instagram.

Every one of those is someone's work. This lesson is about handling that responsibly, and about being honest that the law here is genuinely unsettled in ways that matter.

Four routes, in order of safety

1. Licensed. You paid, or accepted terms, and you have a record. Stock libraries, archive houses, music libraries. The safest and the one that costs money.

2. Openly licensed. Creative Commons, public domain, government open licences. Free, and with conditions that are real obligations: CC-BY requires attribution in a specified form, CC-BY-SA requires your work to carry the same licence, CC-BY-NC forbids commercial use — and a monetised video, a client deliverable or an advertisement for your own services is commercial.

3. Permission. You asked the person and they said yes, in writing, with the scope stated. For material from individuals — a family photograph, somebody's phone video of an event — this is usually the right route and it is usually granted.

4. An exception in law. Fair dealing, fair use, or the equivalent in your country. This is the complicated one.

Where the law is genuinely unsettled

Copyright exceptions differ fundamentally by country, and this is not a detail.

The United States has *fair use*: an open-ended four-factor test — purpose and character including whether the use is transformative, the nature of the work, the amount used, and the effect on the market. It is flexible and it is also unpredictable, because it is decided case by case, after the fact, by a court.

The United Kingdom, India and many Commonwealth countries have *fair dealing*, which works differently: it applies only to an enumerated list of purposes. In the UK these include criticism, review, quotation, news reporting, parody and pastiche. India's Section 52 of the Copyright Act has its own list, which includes private use, criticism or review, and reporting current events. If your use is not on the list, the flexibility of the American test is not available to you, however transformative it is.

Elsewhere the picture varies again, and the internet does not respect borders — a video made in one country is watched in all of them.

Figure 3.3

Two different systems of copyright exception

Fair use, in the United States

- An open-ended four-factor test
- Purpose, nature of the work, amount, market effect
- Any purpose can qualify, including a new one
- Decided case by case, after the fact, by a court

Fair dealing, in the UK and India

- Applies only to an enumerated list of purposes
- Criticism, review, quotation, news reporting
- India's list sits in Section 52 of the Copyright Act
- Off the list, the flexibility is not available to you

This is why the repeated figures of ten seconds or seven notes have no basis in any statute. Nothing here is legal advice; where the stakes are real the question belongs to a lawyer in your jurisdiction.

What this means practically: there is no single answer to "how many seconds can I use". The commonly repeated figures — ten seconds, thirty seconds, seven notes of a song — are folklore with no basis in any statute. This is not a subject where a course can give you a rule, and nothing here is legal advice. Where the material matters and the stakes are real, that is a question for a lawyer in your jurisdiction.

What a course can honestly give you is the shape of the risk and the habits that reduce it.

Habits that reduce the risk

- **Use as little as the point requires.** If you are reviewing a film, use the shot you are discussing, not the scene.

- **Make your use do work.** Commentary, criticism and analysis around the material is what most exceptions are built to protect. Material used as wallpaper is the weakest position in every system.
- **Do not substitute for the original.** If someone could watch your video instead of the original work, you are in the worst part of every test.
- **Attribute, always.** Attribution is not a licence and does not make an infringing use lawful — but its absence makes everything worse, and open licences frequently require it as a condition.
- **Keep a log.** Same file as the music licence log: source, URL, licence or permission, date, what you paid, who granted it. The log is the difference between a claim you can answer in five minutes and one you cannot.

Content ID and platform enforcement, separately

Worth separating from the law entirely, because it operates independently of it.

YouTube's Content ID matches uploaded audio and video against a rights-holder database and raises claims automatically. A claim is not a finding of infringement and a dispute is not a court case. Material you are using entirely lawfully under an exception can be claimed, and material you have licensed can be claimed by the library that licensed it to you. This is an automated matching system, not an adjudication.

The practical consequence is the same as for music: keep records, and expect to use them.

Screen recordings and interfaces

A screen recording of a website or an app contains somebody's copyrighted design, possibly their trademarks, and sometimes other people's content inside it. For tutorials and reviews this is routine and generally uncontroversial. Two things to watch: do not capture other people's personal data in the frame — real names, email addresses, account numbers in a browser tab — and check the terms of service of anything you are recording, because some explicitly restrict it.

Photographs of people

A separate right from copyright. The photographer owns the image; the person in it may have rights over its use for commercial purposes, varying by country. In India the right to privacy and personality rights have been recognised in case law; in much of Europe, image rights are explicit. Using a stranger's face to advertise something is the case that goes wrong most often.

AI-generated material

Whether the output of a generative model is copyrightable, whether training on copyrighted work was lawful, and who bears the risk if an output resembles a specific work — all three are actively disputed and are being decided differently in different countries right now. Treat generated material as carrying unresolved risk rather than as safely free, check what your tool's terms actually grant you, and label it where it is a meaningful part of the work.

Today

Open your last project and write the log you should have kept: every image, clip and track that you did not create, with its source and licence. The ones you cannot account for are the ones to replace.

BEFORE YOU MOVE ON

An editor in India plans to use 25 seconds of a commercial film in a video essay, relying on advice from an American tutorial that the use is transformative and therefore fair. What is wrong with transplanting that reasoning?

A. Transformativeness is a US fair use concept; India's fair dealing applies only to enumerated purposes, so the question is whether the use is on that list.

B. India has no copyright exceptions for the use of film clips at all, so any use of commercial footage requires a licence from the rights holder regardless of the purpose it is put to or how much is used.

C. Twenty-five seconds exceeds the limit that most jurisdictions set for quotation, and staying under ten seconds would place the use safely within the exception.

D. The video will inevitably be watched in the United States as well as in India, so US fair use governs it wherever it was made and the tutorial's advice is therefore sound.

The answer and why: p. 461

CASE STUDY · MODULE 3

The clause that would have fitted the slot

An illustrative case, built from documented patterns.

A freelance editor in Bhopal is cutting a four-minute film for a state department about a rural piped-water scheme. The department's communications officer is the client and signs off the invoice.

The interview is with a junior engineer who has worked on the scheme for six years. Asked whether the remaining villages in the block will be connected, he says this:

> *"In most cases, yes. If the current funding continues into the next financial year, and if the two contractors who left are replaced, we can extend it to the remaining villages. That is the plan."*

Twelve seconds. The film has a four-minute limit set by the slot it is going into, which is a screen in the reception areas of district offices, and the section it belongs to is already thirty seconds over.

The communications officer sends a note with four changes. Three are ordinary: a name spelling, a section that runs long, a request to open on the reservoir rather than the pipes. The fourth asks the editor to use the engineer's answer as "we can extend it to the remaining villages" — four seconds, clean, and it fits the section exactly.

The join is easy. The engineer pauses before "we", and there is a breath after "villages". Cut at the two breaths and the sentence is grammatical, the audio is continuous, and nobody watching could tell. The editor could do it in ninety seconds.

What the cut removes is not filler. It removes two conditions and a hedge. The full answer is a plan contingent on funding and staffing. The short answer is a commitment. A viewer in a district office who has not been connected yet would take one of those as information and the other as a promise, and the engineer did not make a promise.

The editor works this out by applying the test he was taught and has mostly not needed: if the speaker watched this cut, would he be able to point at a proposition in it that he does not hold? The answer is yes. The engineer holds "we can extend it if the funding continues and the contractors are replaced". He does not hold "we can extend it".

The cost of refusing is not abstract. This is the editor's third job for this department and the fourth is under discussion. The communications officer is not acting in bad faith; from where she sits, she has asked for a shorter version of something the engineer said, which is what editing is. She has a slot length and a department that would prefer the film to sound confident. If the editor writes back explaining fabrication and broadcast codes, he is telling a client she has asked him to lie, which is both true and unlikely to be well received.

The cost of agreeing is also not abstract, and it is worth being precise about what it is rather than treating it as a general feeling. The rushes exist. The engineer will see the film, because it plays in the building he works in. And the shape of the sentence — a conditional turned into an assurance about a public service — is exactly the shape that gets repeated back at somebody in a meeting.

The interesting part of the problem is that the editor has three ways to give the client the four seconds she needs, and only one of them involves the cut she asked for.

YOUR ANSWERS

1. The join was inaudible and the sentence was grammatical. Why is that not the standard that matters?

2. Why was offering two alternatives a better move than explaining the principle?

3. What record would have protected the editor if the engineer had later said he never said it?

STOP HERE

Write your answers before you turn the page. How it played out starts on p. 158.

CASE STUDY · MODULE 3

How it played out

The editor sent back two alternative four-second versions rather than a refusal. The first was "If the funding continues, we can extend it to the remaining villages" — four and a half seconds, one condition kept, and it fitted the slot with a frame to spare. The second dropped the answer entirely and used eight seconds from earlier in the interview about what had already been connected, which was uncontested and stronger on screen. He noted in one line, without using the word fabrication, that the shorter version would read as a commitment the engineer had not made and that the reception-area audience would include people waiting to be connected. The communications officer took the first version within the hour. On the next job she asked, unprompted, whether a proposed cut "still says what he said". The editor now keeps a marker at every join in a radio edit where non-adjacent material was joined, and reads that list against the transcript before locking.

The questions, discussed

1. The join was inaudible and the sentence was grammatical. Why is that not the standard that matters?

Because the test is about the proposition, not the craft. Shortening, removing repetition and tightening a rambling answer are normal and expected by anyone who has been interviewed. Removing a qualification changes what was asserted: a conditional plan becomes an unconditional promise. A perfect join makes that harder to detect, which makes it worse rather than better.

2. Why was offering two alternatives a better move than explaining the principle?

Because the client had a real constraint — four seconds — and no intention to deceive. A refusal leaves her problem unsolved and frames her as dishonest, which invites an argument about motives. Two workable cuts solve the constraint and let the reasoning arrive as a single sentence about how the audience would read it. The editor kept both the line and the client, and the next brief came back with the question already asked.

3. What record would have protected the editor if the engineer had later said he never said it?

The rushes and the transcript, kept until well past publication, plus the marked list of every point where non-adjacent material was joined. The defence against "I never said that" is the recording, and the defence against an accusation of misrepresentation is being able to show the full answer beside the cut and demonstrate that the proposition survived.

MODULE 3 TEST

Check what stuck

6 questions, one right answer each. This one is for you, not for a record. The answers, and the reason behind each, are in the key on p. 456. If two go wrong, re-read the module before the exam.

1 What does a one-sentence spine actually do for an edit?

A. It decides which good material does not belong
B. It tells the viewer what the piece is about in the first ten seconds
C. It fixes the running time before the assembly begins
D. It removes the need for a screening with somebody who has not seen the material

2 A thematically structured piece fails in a particular way. What is it, and what is the repair?

A. The strongest material lands last; the repair is to lift it into a cold open
B. The viewer loses the chronology, and the repair is to put a date on every section
C. Each join is a full stop; end each section on a question the next one answers
D. Sections get longer as the piece goes on; the repair is to cap each at two minutes

3 An answer is cut from "In most cases, and if funding continues, we can do it" to "we can do it". The join is inaudible. Why is that a fabrication?

A. Because the speaker was not shown the cut before publication
B. Because the removed words turned a conditional plan into a commitment
C. Because any cut inside a single answer counts as a Frankenbite
D. Because broadcast codes prohibit removing more than half of any single answer

4 Why is grading a shot before picture lock described as doing the work twice?

A. A graded shot plays back slowly, so the rough cut takes longer to judge
B. Grades are attached to the timeline rather than the clip, so a re-order loses them
C. Colour decisions change once titles and captions are laid over them
D. The shot gets shorter or replaced, and the grade has to be redone

5 A first viewer says "the middle is boring". Read as a symptom report rather than a diagnosis, what does that usually mean?

A. The viewer was the wrong person to show it to, and a second opinion is needed
B. The shots in the middle are held too long and want a six-frame pass
C. The middle has no question running through it
D. The music under the middle section is wrong for the material

6 An editor in India wants ninety seconds of a news broadcast in a video essay, and is told "it is transformative, so it is fair use". Why does that not help?

A. India has fair dealing, which applies only to an enumerated list of purposes
B. Transformative use is a defence only where the material is credited on screen
C. News broadcasts carry a separate statutory right that no exception can override
D. Fair use applies only to material that is already out of copyright

4

MODULE 4

Sound, which is most of it

Deliver a mix a stranger can follow on phone speakers — dialogue at a stated loudness with EQ and compression you can justify, continuous ambience under every join, music ducked by a mechanism you chose, and a measured integrated loudness and true peak rather than a guessed level.

Lesson 30 · 11 min

Bad sound kills a good picture

THE ONE THING TO KEEP

Get the mic within 30 cm, keep peaks near -6 dBFS, record thirty seconds of room tone, and use less noise reduction than you want to.

The asymmetry nobody warns you about

A soft, badly lit, wobbly picture with clean sound reads as low-budget but competent. People watch it to the end. A sharp, well-composed picture with hissy, echoing, uneven sound reads as broken, and people leave in the first fifteen seconds.

This is not taste. Speech intelligibility is effortful: if a listener has to reconstruct words from a noisy signal, they spend attention on decoding rather than on what you are saying, and after twenty seconds of that they stop. Nothing in the picture compensates, because nothing in the picture is doing that work. If time on a project is short, spend it on the audio.

The numbers that actually matter

Digital audio is measured in **dBFS** — decibels relative to full scale — counting down from zero. Zero is a hard ceiling, not a warning: above it, samples are flattened, the waveform develops a square edge, and it crackles. Unlike analogue tape there is no warmth up there and nothing to recover. Clipping is permanent damage.

Working targets for spoken word:

- Dialogue **peaks** around **-6 dBFS**, average level around **-12 dBFS**. That leaves headroom for a laugh or a raised voice.
- Music sitting under dialogue: roughly **-18 to -24 dBFS**, quieter than feels right to you, because you already know the words.
- The final mix is normalised for the platform at export.

The two commonest phone-video faults are opposite versions of one mistake: gain at maximum so everything clips, or recording so quietly that you add 20 dB in post — which raises the hiss and the traffic and the fridge by 20 dB too. The noise floor is baked in at the moment of recording.

Distance beats equipment

If you buy one thing, buy a cheap wired lavalier microphone — around ₹800 to ₹1,500, or ten to twenty dollars — and clip it about 20 to 30 centimetres from the mouth, out of frame. It will beat an expensive microphone two metres away, every time.

The reason is the ratio of direct sound to reflected sound. Sound reaching the mic straight from the mouth falls off quickly with distance; sound bouncing off walls and ceiling is spread evenly through the room and barely falls off at all. Halve the distance and you dramatically improve that ratio. This is why phone video shot across a room sounds like a bathroom: not because the phone mic is bad, but because most of what reaches it has been off a wall first.

Then turn off the fan, the air conditioner and the fridge, close the window, and put a blanket behind the camera if the room is bare. None of it costs anything.

Room tone, and the holes in your dialogue

Record **thirty seconds of the room doing nothing** at every location. Everyone forgets, and everyone regrets it.

Every room has a noise floor — air, hum, distant traffic. When you cut dialogue, that floor stops and restarts at every join, and the ear hears the sudden silence as a hole punched in the scene. It is one of the clearest amateur tells there is. Lay room tone on a track under the whole scene, fill any true gaps with it, and the holes vanish.

While you are there: a **high-pass filter at 80–100 Hz** on every voice track removes rumble, handling noise and air conditioning without touching the speech.

Figure 4.1

Four things that cost nothing and fix most bad sound

Microphone within 30 cm
A cheap wired lavalier close beats an expensive microphone across the room

↓

Peaks around -6 dBFS
Recording quietly bakes the hiss in; going over zero is permanent damage

↓

Thirty seconds of room tone
Every location, every time. It fills the holes your cuts punch in the background

↓

High-pass at 80 to 100 Hz
Removes rumble, footsteps and air conditioning, and touches no speech

Only the fourth happens in the edit. The first three cannot be added afterwards, which is why the noise floor is decided at the moment of recording.

Noise reduction, and exactly how it fails

Classic noise reduction works by spectral subtraction. You give it a sample of noise alone, it builds a profile of how much energy sits in each frequency band, and it subtracts that amount everywhere.

Push it too far and quiet harmonics of the voice fall below the estimate, so they are removed in some frames and kept in others. Those fragments flutter in and out and you get the watery, swirling, underwater artefact engineers call musical noise.

The rule: **if you can hear the noise reduction, there is too much of it.** Six decibels that leaves a little honest hiss beats twenty that leaves a robot. Audiences forgive background noise. They do not forgive artefacts, because artefacts sound like something is wrong with the file.

Free tools: Audacity's noise reduction, and the EQ and dynamics on Resolve's Fairlight page — though some Resolve restoration plug-ins are Studio-only.

The newer speech tools, and where the line is

A different class of tool does not subtract noise. It runs the recording through a model and **resynthesises the voice**. The results can be startling — a phone recording from across a room comes back close to studio.

Two cautions. It is generating audio, so it can shift timbre and invent detail that was never there, which makes it inappropriate wherever the recording is evidence, journalism, or a record of what someone said. And once a tool goes past cleanup into replacing words, changing accents or producing a voice, that is synthetic media: it needs a visible label — several countries, India among them, now require labelling of synthetic content — and somebody's voice needs their written consent. Consent for a likeness or a voice is not a style question. Making Things with AI covers where that stands.

Two traps that catch everyone once

- **Mono recorded into one channel.** A lavalier plugged into a single input often lands on the left channel only, so anyone listening on one earbud hears silence. Set the clip's audio to mono, or copy it to both channels.
- **No sync reference.** If picture and sound came from different devices, clap once, hard, in frame at the top of every take. A clap gives you a single unmistakable spike to line up. Waveform auto-sync works well but fails on noisy or quiet takes, and then you will want that clap.

Today

Play something you made with the picture turned off. If you would not listen to it as a podcast, the picture is not saving it.

BEFORE YOU MOVE ON

An interview recorded near a busy road is cleaned up with heavy noise reduction. The traffic is gone, but the voice now has a swirling, underwater quality. What is happening?

A. The high-pass filter was set too high and has removed the fundamental frequency of the voice, leaving only harmonics.
B. The reduction is subtracting a noise estimate from every band, and quiet voice harmonics near that threshold are being removed in some frames and kept in others, so they flicker.
C. The recording was clipped at source, and noise reduction has exposed the distortion that the traffic was previously masking.
D. The noise profile was sampled from a section containing speech, so the tool is reducing the whole signal evenly.

The answer and why: p. 463

Lesson 31 · 9 min

The music you found is not free

THE ONE THING TO KEEP

Every track needs a licence you can produce on demand, and "royalty free" is a pricing model, not permission.

A word that has cost a lot of people money

"Royalty free" does not mean free. It means no recurring per-play royalty. The licence still has terms, it may still cost money, and it may forbid exactly what you are doing. Plenty of people have read those two words, taken a track, and found out later what the rest of the sentence said.

There are three separate things that can go wrong, and they are worth separating because the fixes differ.

- **You had no licence.** Straightforward infringement.
- **You had a licence, but the wrong one.** Personal use where you needed commercial; one channel where you needed all platforms; one video where the licence was per-project.
- **You had the right licence and got claimed anyway.** This is the one that surprises people, and it is the most common.

Content ID, and why a legitimate licence still gets flagged

YouTube fingerprints uploaded audio and matches it against a database of works submitted by rights holders. A match raises a claim automatically. A claim is not a copyright strike — your channel is not in danger — but it can redirect all revenue from your video to the claimant, and it can block the video in particular countries.

The awkward part: many stock-music libraries register their whole catalogue with Content ID. So you buy a licence, use the track exactly as permitted, and get claimed anyway, because the system does not know about your licence until you tell it. Resolving it means submitting your licence details in a dispute and waiting.

This is not a reason to avoid licensed music. It is a reason to keep records.

Free sources that are genuinely usable

- **YouTube Audio Library** — free, cleared, some tracks require attribution. Safest inside YouTube itself.
- **Free Music Archive**, **ccMixter**, **Incompetech** — Creative Commons, per-track licences, mostly CC-BY.
- **Pixabay**, **Uppbeat**, **Bensound** — free tiers with their own conditions; Uppbeat's free tier requires a credit and a per-track code.
- **Musopen** — public-domain classical, including recordings released freely.

- **Freesound** and **Zapsplat** for sound effects. On Freesound the licence is set per file and varies from CC0 to non-commercial, so check each one individually rather than assuming.
- **BBC Sound Effects** — a large archive free for personal and educational use under its own licence. Read the terms before any commercial use.

Two traps inside Creative Commons specifically:

CC-BY-NC forbids commercial use, and a monetised video is commercial. So is a video advertising your business, and arguably so is a portfolio piece that gets you hired. If money is anywhere near the video, avoid NC.

A public-domain composition is not a public-domain recording. Beethoven died in 1827; the 2019 orchestral recording of the Fifth is a separate copyrighted work with its own performers and label. This catches people constantly. Musopen exists precisely to give you recordings that are actually free.

Keep a licence log

One plain text file per project. For every piece of music and every sound effect, write down: track title and artist, source URL, licence name, the exact attribution string required, the date you downloaded it, and what you paid.

Do this even for free tracks, and especially for free tracks. Eight months later a claim arrives, and the question is not whether you were allowed — it is whether you can prove it in the five minutes you have before you give up and mute the video.

Ducking: making room for the voice

Music under dialogue has to move out of the way. Two methods.

Manual keyframes. Put music on one track and dialogue on another, and pull the music level down by 12 to 15 dB wherever someone speaks, with a short ramp in — around 0.3 seconds — and a slower ramp out, around 0.5 to

0.8 seconds. Slower coming back up sounds natural; snapping back up sounds mechanical.

Sidechain compression. A compressor on the music bus, keyed from the dialogue bus, so the music automatically drops whenever the voice is present. Resolve's Fairlight page has this, as does Reaper and any DAW. It is much faster on a long project and slightly less musical, because it does not know where the music wants to breathe.

Beginners consistently mix music too loud. You already know every word being said, so you cannot hear the masking. The test is blunt: if you cannot understand the dialogue with the music at the level you like, the music is too loud. Play it once on phone speakers, which is where most people will hear it.

Two more things about music

Cut the picture first. If music goes in early you will start cutting to the beat and protecting phrases, and the structure ends up serving the track rather than the meaning. The exception is a montage, where cutting to the beat is the point.

Start the music on an event. A slow eight-second fade-up over nothing announces that you did not know where to begin. Bring it in on a cut, a door, a word.

Generated music

Tools that generate music from a text description are usable and getting better. Three things to keep straight. Check whether your tier grants commercial use — several grant it only on paid plans. Label it as generated where it is a meaningful part of the work, which is increasingly a legal requirement rather than a courtesy. And understand that the training-data disputes around these systems are live and unresolved, so do not assume the output is free of anyone else's rights just because a tool produced it. The shape of the argument is in Making Things with AI; a lawyer in your country is the person who answers it for your case.

Today

Open your last project and write the licence log you should have written then. If you cannot reconstruct where a track came from, that is the answer about how risky it was.

BEFORE YOU MOVE ON

A small business uploads a promotional video using a track from a paid stock library, with a valid commercial licence. Two days later YouTube shows a copyright claim and the video is monetised by someone else. What has most likely happened?

A. The licence covered personal use only, and the business has infringed by using it commercially.

B. Using licensed music without on-screen attribution voids the licence, which triggered the claim.

C. The library registered its catalogue with Content ID, the upload was fingerprint-matched automatically, and the system has no knowledge of the licence until it is submitted in a dispute.

D. The video was flagged as a copyright strike, and the channel is now at risk of deletion after two more.

The answer and why: p. 464

Lesson 32 · 8 min

A track layout that survives a revision

THE ONE THING TO KEEP

Organise audio into dialogue, effects and music busses so a change is one fader rather than forty clips, and level clips with clip gain before any compressor sees them, because a compressor responds to whatever level arrives at it.

Why layout is a technical decision

A mix with everything on two tracks works until the first change. Then the client wants the music swapped, and you discover that the music level was set clip by clip, forty times, and every one of those settings belonged to the old track.

A mix organised into groups survives, because the decisions live in one place per group rather than once per clip.

The standard layout

From the top of the timeline down, or the bottom up — the order is a convention, not a rule, but be consistent:

- **A1–A2: Dialogue.** One track per speaker where you have two people, so each keeps its own processing. A third for narration.
- **A3: Room tone.** A continuous bed under everything, from the previous block.
- **A4–A5: Sound effects.** Spot effects — a door, a click, a phone.
- **A6–A7: Ambience.** Beds: traffic, birds, a room's hum, a crowd.
- **A8–A9: Music.**

Nine tracks sounds like a lot for a five-minute film. It costs nothing, and it means "turn the music down" is one fader rather than a search.

Busses

A **bus** is a channel that several tracks feed into, so processing applied to the bus applies to all of them at once.

The three you want, and they map onto the standard delivery categories:

- **Dialogue bus** — everything spoken. Compression and any overall EQ goes here.
- **Music bus**.
- **Effects bus** — spot effects and ambience together.

Then all three feed the **main** output, where the limiter and the loudness measurement live.

In Resolve's Fairlight page these are called busses and you create them in the Mixer's bus format dialogue. Premiere calls them submixes. Reaper, Audacity's multi-track mode and Ardour — all free or cheap — do the same thing under the same names.

The payoff: sidechain ducking becomes one connection between two busses rather than forty keyframed clips. A loudness problem is fixed with three faders. And the stems below fall out for free.

Stems, and why anybody cares

A **stem** is a bounce of one bus on its own: the whole film's dialogue as a single file, the whole film's music, the whole film's effects.

They exist because of three real situations:

- **Translation.** To dub a film into another language, somebody needs everything *except* the dialogue. That combination — music and effects, no speech — is called an **M&E** and it is a standard deliverable. You cannot produce one from a flattened mix.
- **Revision.** A change to the music six months later needs the dialogue and effects untouched.
- **Broadcast delivery.** Many broadcasters require stems as a matter of course, and will reject a delivery without them.

Export them at the same length as the programme, from the same start timecode, so they line up when dropped back in. Name them `<project>_DIA.wav`, `_MUS`, `_FX`, `_ME`.

Clip gain against track fader

Two different controls that both make things louder, and using the wrong one is the most common mixing mistake.

Clip gain adjusts the level of one clip *before* it reaches any processing on the track. Use it to make all your dialogue clips roughly equal — which is a job you do first, before any other mixing.

The track fader adjusts everything on the track *after* processing. Use it for the overall balance between elements.

The reason the order matters: a compressor on the dialogue track responds to whatever level arrives at it. If clip 1 arrives at -20 and clip 2 at -8, the compressor treats them completely differently and no fader setting fixes both. Level the clips first with clip gain, then compress, then balance with the fader.

The order of work in a mix

1. **Level the dialogue clips** with clip gain so the waveforms look roughly even. Aim for peaks near -12 dBFS at this stage, leaving room for processing.
2. **EQ** each dialogue source — next lesson.
3. **Compress** the dialogue bus — the lesson after that.
4. **Lay room tone and ambience** so there are no holes.
5. **Place effects.**
6. **Bring music in** and duck it.
7. **Set the overall loudness** and put a limiter on the main.

Doing these out of order means redoing them. Compressing before levelling is the classic one.

Naming and colour

Name every track. Colour dialogue one colour, music another, effects a third. On a timeline with thirty audio clips this is the difference between finding a problem and hunting for it.

And lock tracks you have finished with. A locked track cannot be dragged by accident, which is how sync gets broken at 11pm.

Today

Take a project you have mixed on two tracks and rebuild its layout into dialogue, effects and music busses. Then change the music level with one fader. That is the whole argument.

BEFORE YOU MOVE ON

An editor compresses the dialogue bus and then levels the individual clips with clip gain afterwards. Why does the result sound inconsistent no matter how carefully the clip gains are set?

A. Clip gain is applied after the track's processing chain, so the compressor never sees the adjusted levels at all and the changes reach the output unprocessed.

B. Compression should always be applied per clip rather than on a bus, so the fault is in using a bus compressor at all for dialogue that comes from several different sources and recordings.

C. Clip gain changes the waveform display without changing the audible level, so the editor is matching pictures of waveforms rather than actual loudness.

D. The compressor reacts differently to each clip depending on the level arriving at it, so raising a quiet clip afterwards raises a signal that was already compressed differently.

The answer and why: p. 464

Lesson 33 · 9 min

Four EQ moves that make speech intelligible

THE ONE THING TO KEEP

Four moves cover most speech — a high-pass around 80 to 100 Hz, a notch at the mains frequency and its harmonics, a small cut in the 300 to 500 Hz boxy region and a gentle lift around 2 to 3 kHz — applied subtractively before any compressor sees the signal.

What EQ is doing

An equaliser changes how much energy exists in each frequency band. That is all. It cannot add detail that was not recorded, it cannot remove a sound that shares frequencies with the voice, and it cannot fix a bad room.

What it can do is remove energy that is not the voice, and rebalance what remains so the consonants come through. For spoken word that is four moves, and most speech needs no more than three of them.

The human voice, by frequency

Rough map, and worth holding in your head:

Figure 4.2

The speaking voice, band by band

Above 10 kHz: air
Pleasant, and mostly absent from cheap microphones anyway

5 to 8 kHz: sibilance
The s and t sounds. A de-esser works only in this band

1 to 4 kHz: presence
Consonants live here, and this band decides whether words are understood

250 to 500 Hz: boxy
Room resonance. A cut of 3 to 5 dB often clears a whole mix

80 to 250 Hz: body
The fundamental of most voices. Too much of it and the voice booms

Below 80 Hz: nothing useful
Rumble, traffic, handling noise, air conditioning, a chair creaking

Three of the four standard moves are cuts, and they come before the compressor: a compressor triggered by a creaking chair ducks the whole voice with it.

- **Below 80 Hz** — nothing useful in a voice. Rumble, traffic, handling noise, air conditioning.
- **80–250 Hz** — fundamental of most voices. Weight and body. Too much and it is boomy.
- **250–500 Hz** — the "boxy" region. Room resonance lives here and cutting a little often clears a mix.
- **1–4 kHz** — presence and intelligibility. Consonants live here, and this is the band that decides whether words are understood.
- **5–8 kHz** — sibilance. The `s` and `t` sounds.
- **Above 10 kHz** — air. Pleasant, and mostly absent from cheap microphones anyway.

Move one: the high-pass filter

A **high-pass** filter removes everything below a set frequency. Set it at **80 Hz for a male-range voice and 100 Hz for a higher one**, with a slope of 12 or 18 dB per octave.

This is the single most valuable EQ move in video work and it should be on every dialogue track by default. It removes traffic, air conditioning, footsteps through the floor, wind and handling noise, none of which is in the voice.

The test that it is set correctly: sweep it upward until the voice starts to sound thin, then back off by 20 Hz.

Move two: the notch for hum

Mains hum sits at exactly the mains frequency — **50 Hz in India, Europe, Africa; 60 Hz in the Americas** — plus its harmonics at 100, 150, 200 Hz and so on.

It is narrow and it is at a known frequency, which makes it the easiest thing in audio to remove. Use a notch filter — a very narrow, deep cut — at the fundamental and the first two or three harmonics. A high-pass alone will not do it, because the harmonics are inside the voice range.

Many tools have a dedicated hum removal or de-hum module with a 50/60 Hz switch. Audacity has Notch Filter; Resolve's Fairlight has a De-Hummer.

Move three: the boxy cut

If the voice sounds like it was recorded in a cupboard, sweep a narrow boost around **300–500 Hz** until the boxiness is exaggerated, find its centre, then turn that boost into a cut of **3 to 5 dB** with a moderate width.

This is the standard find-it-by-boosting technique and it works for any resonance. What it cannot do is remove the reverberation itself — that is a different problem, dealt with two lessons on.

Move four: the presence lift

A gentle shelf or wide bell of **2 to 4 dB centred around 2–3 kHz** raises the consonants and makes words easier to catch.

Do this last, and do it sparingly. Too much and the voice becomes harsh and fatiguing, which is worse than slightly dull. If you find yourself wanting 8 dB, the problem is that the microphone was too far away and EQ is not the fix.

De-essing, when the lift bites back

Raising presence often raises sibilance with it. A **de-esser** is a compressor that only responds to a narrow high band — typically 5 to 8 kHz — turning down the `s` sounds and leaving everything else.

Set the threshold so the gain reduction meter moves only on the sibilants, by 3 to 6 dB. If it is working constantly, the band is wrong or the threshold is too low, and the voice will sound lisping.

The order, and why

High-pass → hum notch → resonance cut → presence lift → de-ess → compression.

Subtractive moves come first because they reduce the signal reaching the compressor, and a compressor that is being triggered by 60 Hz rumble is ducking the whole voice every time somebody's chair creaks.

What EQ cannot do

Say these plainly, because people spend hours trying:

- **Remove room reverb.** The reflections share every frequency with the direct sound. Dedicated de-reverb tools exist and work partially; EQ does not.
- **Separate two people talking at once.** Both voices occupy the same bands.

- **Remove wind rumble that is inside the voice range.** A high-pass takes the low part; the rest is in the recording permanently.
- **Fix distance.** A voice recorded across a room has a poor ratio of direct to reflected sound, and no filter changes a ratio.

Free tools

Everything above exists in Resolve's Fairlight page at no cost, including the parametric EQ on every channel and the De-Esser and De-Hummer. Audacity has filter curve EQ, notch and a basic de-esser via plug-ins. Reaper's ReaEQ and ReaXComp are excellent and Reaper's evaluation is unlimited. For plug-ins, TDR Nova is a free dynamic EQ that will do the de-essing and the resonance work in one.

Today

Take a voice recording, add a high-pass, and sweep it up from 20 Hz while listening. Find the point where the voice thins, then back off. You now have the only EQ move most dialogue needs.

BEFORE YOU MOVE ON

A voice recorded in a bare room sounds hollow and distant. An editor cuts 4 dB at 400 Hz, which helps slightly, then tries progressively deeper cuts, which make the voice thin without removing the hollowness. Why?

A. The cut is at the wrong frequency and should be found by boosting a narrow band and sweeping until the hollowness is exaggerated.
B. The hollowness is reverberation, which shares every frequency with the direct sound, so an equaliser can only remove the voice along with it.
C. A high-pass filter should have been applied first, and without it the cut at 400 Hz is fighting low-frequency energy that is still present in the signal and masking the result.
D. The microphone's own frequency response is the cause, and only a different microphone in the same room would give a usable recording of that voice.

The answer and why: p. 465

Lesson 34 · 9 min

What a compressor does, and what it does to the gaps

THE ONE THING TO KEEP

Set dialogue compression by the gain reduction meter — 3 to 6 dB on loud words, returning to zero between phrases — and remember that make-up gain raises the noise floor too, so a hissy recording can take less compression than a clean one.

The one-sentence description

A compressor turns down anything louder than a threshold, by a ratio, with a chosen speed of response. It makes nothing louder. What makes things louder is the make-up gain you add afterwards — and that gain raises everything, including whatever noise sits between the words.

That last clause is the whole risk. Motion, Sound and Music covers the general principle; this lesson is about the settings for a speaking voice in an edit.

Why dialogue needs it

A person speaking varies by 20 dB or more between an emphatic word and a trailing-off aside. On good headphones in a quiet room that is fine. On phone speakers on a bus, the quiet half is inaudible and the loud half is the only thing you hear.

Compression narrows that range so the whole sentence sits in the band a phone speaker can deliver. It is not an effect; for spoken-word video it is a delivery requirement.

The five controls

Threshold — the level above which the compressor acts. For dialogue levelled to peaks around -12 dBFS, start the threshold around -**18 dBFS** and adjust.

Ratio — how hard it turns down. 3:1 means an input 3 dB above threshold comes out 1 dB above. For dialogue, **2:1 to 4:1**. Above about 8:1 you are limiting rather than compressing.

Attack — how quickly it responds once the threshold is crossed. **5 to 15 ms** for speech. Too fast and you flatten the consonant transients that carry intelligibility, and the voice goes dull. Too slow and the loud syllable escapes before the compressor acts.

Release — how quickly it stops once the signal drops. **50 to 150 ms** for speech. Too fast and you hear the background noise surge up between words — the artefact called *pumping* or *breathing*, and once you can hear it you cannot stop hearing it. Too slow and the compressor never recovers between phrases, so quiet words after a loud one stay suppressed.

Make-up gain — the level added afterwards to compensate. Set it so the compressed signal is about as loud as the uncompressed one was, then compare with the bypass button.

The number to watch

The gain reduction meter. For dialogue you want it moving **3 to 6 dB on the loud words and returning to zero between phrases.**

If it is pinned at 10 dB and never releases, the threshold is too low and the voice is being squashed flat. If it never moves, the threshold is too high and the compressor is doing nothing.

This single meter is a more reliable guide than any preset, because it shows you the compressor's behaviour rather than its settings.

The noise floor problem, precisely

Here is the mechanism that catches people.

Compression reduces the difference between the loudest and quietest parts. Make-up gain then raises the whole signal. The voice comes up — and so does the air conditioning, the room hum, the traffic, the computer fan.

So a recording with a poor noise floor gets audibly worse with compression, and the worse the original recording, the less compression it can take. Squash a hissy recording by 12 dB and add 12 dB of make-up, and you have made the hiss 12 dB louder than it was.

This is why the recording lessons come before this one. Compression does not create the problem; it reveals how much of one you had.

Two-stage compression

The approach professionals use, and it is gentler than one heavy compressor:

1. **A light compressor on each dialogue track** — ratio 2:1, 3 dB of reduction — to even out that speaker.
2. **A light compressor on the dialogue bus** — ratio 2:1 or 3:1, another 3 dB — to glue everything together.

Two lots of 3 dB sound far more natural than one lot of 6 dB, because each stage is working gently and neither is being asked to catch everything.

The limiter, which is different

A **limiter** is a compressor with a very high ratio and a very fast attack, used as a ceiling rather than as shaping. One goes on the main output, with the ceiling at about -**1 dBFS true peak**, to catch anything that would otherwise clip the encoder.

It should be doing almost nothing. If your limiter is pulling down 6 dB regularly, your mix is too loud and the limiter is now your mix engineer.

Free tools

Resolve's Fairlight has a compressor and limiter on every channel and on every bus, plus a dedicated Dialogue Processor that combines gate, EQ, compression and de-essing with sensible defaults. Reaper's ReaComp is excellent. Audacity has a basic compressor; for anything beyond levelling, use one of the multi-track tools instead.

Today

Put a compressor on a voice recording, set the ratio to 3:1, and lower the threshold slowly while watching the gain reduction meter and listening to the gaps between words. Find the point where you first hear the background surge. That point is your ceiling on that recording.

BEFORE YOU MOVE ON

An editor compresses an interview heavily to make it consistent, and afterwards the background hiss audibly surges up in the gaps between sentences. Which two settings are producing that specific artefact?

A. A slow attack combined with a high ratio, which lets the loud syllables through and then clamps down hard on everything after them.
B. A threshold set too high, so the compressor only engages on the loudest peaks and leaves the quiet sections entirely untreated and therefore noticeably noisier by comparison.
C. A fast release with heavy make-up gain, so the compressor recovers during pauses and the amplified noise floor comes straight up with it.
D. A limiter on the output with its ceiling set too low, which is pulling the whole mix down and raising the relative level of the noise between words.

The answer and why: p. 465

Lesson 35 · 9 min

Loudness, measured rather than guessed

THE ONE THING TO KEEP

Loudness is an energy measurement over time, not a peak, so mix to roughly the destination's integrated LUFS target with true peak at -1 dBTP — mastering louder than the target just means the platform turns you down after you destroyed the dynamics.

Why peak level is the wrong measurement

Two files can both peak at -1 dBFS and one can be twice as loud as the other. Peak level measures the single highest sample. Perceived loudness depends on how much energy there is over time, and on which frequencies it sits in, because the ear is far more sensitive around 2–5 kHz than at the extremes.

So a meter that only shows peaks tells you whether you will clip. It tells you nothing about whether your video is louder or quieter than the one before it in the feed.

LUFS

LUFS — Loudness Units relative to Full Scale — is a measurement that applies a filter modelling the ear's frequency response and then averages energy over time. It was standardised as ITU-R BS.1770 and adopted in Europe as EBU R128. It is the number every platform now normalises against.

Three readings, and they answer different questions:

- **Integrated** — the average over the whole programme. This is the headline number, and it is what platforms use.
- **Short-term** — a three-second window. Useful for spotting a section that is too quiet.

- **Momentary** — a 400 ms window. Useful for finding individual loud moments.

There is also **loudness range (LRA)**, which measures how much the loudness varies across the piece. A very low LRA means everything is at the same level, which is fatiguing. A very high one means quiet sections will be inaudible on a phone.

True peak, separately

dBTP — true peak — estimates the peak of the reconstructed analogue waveform rather than the highest digital sample. These differ: between two samples the actual waveform can rise above both, a phenomenon called inter-sample peaking.

A file that peaks at exactly 0 dBFS can therefore clip when it is converted to analogue, or when it is re-encoded to a lossy format, which every platform does. Leave headroom: **-1 dBTP** is the normal target.

The numbers platforms use

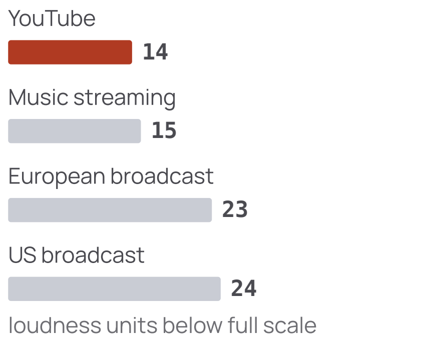
Roughly, as of writing, and these move:

- **YouTube**: normalises toward around -14 LUFS.
- **Spotify, Apple Music**: around -14 to -16 LUFS.
- **Instagram, TikTok and short-form**: broadly similar territory.
- **European broadcast (EBU R128)**: **-23 LUFS**, with true peak at -1 dBTP. This is a regulatory standard, not a suggestion, and broadcasters reject files outside tolerance.
- **US broadcast (ATSC A/85)**: -24 LKFS, which is the same measurement under a different name.

The gap between -14 and -23 is large, and it is the reason a file that passed broadcast QC sounds quiet on YouTube. Ask where the file is going before you set the level.

Figure 4.3

Integrated loudness targets, by destination



Drawn as depth below full scale, so a longer bar is a quieter target. The platform figures move; the broadcast ones are regulatory and files outside tolerance are rejected.

What normalisation means for you

This is the part that changes behaviour.

Platforms turn loud uploads **down**. They generally do not turn quiet ones up by much, if at all. So there is no advantage to mastering loud, and there is a real cost: to get a mix to -6 LUFS you have compressed and limited it heavily, destroying its dynamic range. The platform then attenuates it by 8 dB, and you arrive at the same perceived volume as everyone else with a flatter, more tiring mix.

Mix to roughly the target and let the platform leave you alone.

Measuring it, free

Resolve's Fairlight page has a loudness meter with integrated, short-term, momentary, LRA and true peak, and a preset for each broadcast standard. Open it, play the whole programme from the start, and read the integrated value at the end. It must be the whole programme — an integrated reading of thirty seconds is a reading of thirty seconds.

FFmpeg measures it without opening anything:

```
ffmpeg -i film.mp4 -af ebur128=peak=true -f null -
```

The summary at the end gives integrated loudness, range and true peak. And FFmpeg will also correct it, in two passes for accuracy:

```
ffmpeg -i film.mp4 -af
loudnorm=I=-14:TP=-1:LRA=11:print_format=summary \
       -f null -
# then feed the measured_* values back into a second pass
```

The two-pass form matters: a single-pass `loudnorm` works dynamically and can audibly alter the mix, while two-pass applies a single fixed gain, which is what you want on a finished programme.

YouLean Loudness Meter has a free version that runs as a plug-in in any DAW or NLE.

Dialogue-anchored loudness

One refinement worth knowing. Broadcast practice anchors loudness to the dialogue rather than to the whole mix, because a film with loud music and quiet speech can measure correctly and still be unintelligible.

The practical version: get your dialogue sitting comfortably first, then build music and effects around it, then check the integrated figure. If you have to pull the whole mix down to hit the target, pull the music down rather than the dialogue.

Today

Run the `ebur128` command above on the last video you published. Compare its integrated figure to -14. Whatever the number is, it explains how your video sounded next to everyone else's.

BEFORE YOU MOVE ON

A creator masters a video at -8 LUFS integrated because it sounds punchier in their editor, and uploads it to a platform that normalises to around -14. What happens?

A. The platform attenuates it by roughly 6 dB, so it arrives at the same perceived loudness as everyone else but with the dynamic range already spent on getting to -8.
B. The platform rejects the file, since uploads outside the loudness tolerance fail automated quality control at ingest.
C. The extra loudness survives, because normalisation is applied only to music streaming services and not to video platforms, where creators control their own output levels.
D. The true peak will exceed 0 dBTP during the platform's re-encode, producing audible distortion that would not have occurred at a lower integrated level.

The answer and why: p. 466

Lesson 36 · 8 min

The sounds nobody notices and everybody misses

THE ONE THING TO KEEP

A continuous ambience bed fills every hole your cuts made in the background at once, and spot effects only need to cover the events the viewer can see happening — layered two or three deep and placed a frame early rather than a frame late.

Two categories

Ambience — a continuous bed. Traffic outside a window, a fan, birds, the hum of an office, distant conversation. It establishes that a place exists and it runs under everything.

Spot effects — individual sounds tied to a visible event. A door closing, a cup on a table, a keyboard, a phone buzzing, footsteps.

Audiences do not notice either when they are present. They notice immediately when they are absent, though they usually describe the result as "it feels flat" or "it looks like a video" rather than as a sound problem.

Why absence is so audible

The camera microphone records the room it was in. When you cut between shots filmed in different places, or at different times in the same place, the background changes at every join. When you place a shot with no usable audio at all — a drone shot, a graphic, a still — the background disappears entirely.

Each of those is a hole. The ear tracks the continuity of a space far more attentively than the eye tracks the continuity of a frame, so a break in the background reads as a break in reality.

A continuous ambience bed under the whole scene fills every one of those holes at once. This is the single highest-value hour of sound work in any edit.

Building an ambience bed

1. Find or record a loop of the right room. Thirty seconds is plenty.
2. Place it on its own track, running the entire length of the scene, under everything.
3. Cross-fade the loop points by half a second so the join is inaudible.
4. Set it quietly — typically **-40 to -30 dBFS**, well below dialogue. You should not be able to point at it.
5. Duck it slightly under the loudest dialogue if it is competing, and bring it up in the gaps.

The test: mute it and unmute it. If muting makes the scene sound dead rather than making you notice a sound disappearing, the level is right.

Spot effects

The useful principle: **you do not need every sound, you need the sounds the viewer is looking at.** If a hand puts a cup down in frame and there is no sound, the ear registers a small wrongness. If somebody walks past in the background without footsteps, nobody cares.

So work through the cut once, noting every visible event that should make a noise and does not, and fill those. On a five-minute piece that is usually fifteen to thirty sounds.

Layering is what makes an effect convincing. A door closing is not one sound; it is the handle, the movement of air, the latch and the frame. Two or three layers, slightly offset, sound like a real event. One sound from a library sounds like a sound from a library.

Sync to the frame

A spot effect that lands two frames late reads as wrong even though nobody can say why. Zoom in and place the transient — the sharp attack at the start of the sound — on the exact frame of the visible impact.

A useful trick: place it one frame *early* rather than one frame late. Sound arriving fractionally before the picture reads as tight; arriving after reads as a dub.

Recording your own

The cheapest and most effective sound work available. Your phone records usable spot effects for anything small and close: a cup, a door, paper, a switch, keys, a zip, footsteps on a hard floor.

Record each one six or seven times with slight variation, in the quietest place you have, close to the object. You will end up with a personal library that is better than most free ones because it is specific to the things in your films.

Room tone counts as your own ambience library too. Every location you ever film, record thirty seconds of nothing. Within a year you have a usable collection.

Free libraries

- **Freesound** — enormous, and the licence is set per file and varies from CC0 to non-commercial, so check each one individually.
- **BBC Sound Effects** — a very large archive, free under its own licence for personal and educational use; read the terms before any commercial use.
- **Zapsplat** — free with attribution on the free tier.
- **The Internet Archive** and various public-domain collections.

The same logging discipline applies as for music: source, licence, date, attribution string, in the project's docs folder.

Where it stops being worth doing

A talking-head video shot in one room needs room tone and essentially nothing else. Elaborate sound design on a piece with no visible events is an hour spent on something nobody will perceive.

The things that reliably repay the effort: any cut between locations, any shot with no usable original audio, any visible impact in frame, and any moment where you want the viewer to feel they are somewhere.

Today

Take a scene you have cut and lay one continuous ambience bed under the whole thing at -35 dBFS. Play it with and without. That is a minute of work and it is audible.

BEFORE YOU MOVE ON

A sequence cuts between three locations and a drone shot. The picture is good but the piece feels flat and unconvincing. Which sound problem does that description most likely indicate, and why does it read as a picture problem to the viewer?

A. The dialogue is under-compressed, so levels vary between locations and the viewer is subconsciously straining to hear, which they experience as disengagement rather than as an audio fault.

B. The background changes at every join and vanishes under the drone shot, and the ear tracks spatial continuity closely enough that the break registers as unreality.

C. Spot effects are missing for the visible events in each location, and without them the individual actions on screen have no physical weight to them at all.

D. The music bed is carrying the whole sequence, and music alone cannot establish a location the way recorded sound from the place can.

The answer and why: p. 466

Lesson 37 · 8 min

Making a replaced line sound like it belongs

THE ONE THING TO KEEP

A replaced line sounds wrong because its space is wrong, so match the noise floor first, then the EQ, then add far less reverb than sounds like reverb — and record an impulse response with one clap at every location, because de-reverb is much harder than matching.

The problem

You re-record one line. Or you use a clean take of a sentence from a different part of the interview. Or you have narration recorded in your bedroom placed over footage from a warehouse.

It sounds wrong, immediately and obviously, even though the voice is the same person and the level matches. What has changed is the **space** — the pattern of reflections that tells the ear how big the room is and how far away the speaker was.

What the ear is detecting

Three things, in order of how much they matter:

Reverberation. How long reflections take to decay. A bathroom decays slowly, a bedroom with a bed and curtains decays quickly, outdoors barely at all. A dry line in a reverberant scene sounds like it was pasted on, because it was.

The direct-to-reflected ratio. How close the microphone was. A close-miked line has a high proportion of direct sound and reads as intimate; a line recorded across a room reads as distant even at the same volume.

The noise floor. Every room has one, with its own character. A perfectly silent line in a scene with a fan running is a hole, the same hole the room tone lesson described.

Fixing it, in order

1. Match the noise floor first. Lay the scene's room tone under the replaced line. Half the problem usually disappears here, because the ear stops noticing the silence and stops interrogating the rest.

2. Match the EQ. A room's resonances colour the voice. Put the original and the replacement side by side, listen alternately, and use the EQ moves from earlier in this block to bring the replacement toward the original. Usually this is a small cut or boost in the 200–600 Hz region.

3. Add reverb to match, never to impress. Use a **convolution reverb** if you have one — it applies a measured room's actual response rather than an approximation. Resolve's Fairlight has Reverb; Reaper includes ReaVerbate and the convolution plug-in ReaVerb, both free.

Set it very low. The amount of reverb that makes a line match a room is far less than the amount that sounds like reverb. Start at 5% wet and raise until the line sits, then back off.

4. Match the perceived distance. If the original was further away, roll off some high frequencies as well as adding reverb — air absorbs high frequencies over distance, and a distant sound is duller as well as more reverberant.

Recording your own impulse response

If you can get back to the location, this is the professional method and it is free.

A convolution reverb needs an **impulse response**: a recording of how the room responds to a very short sound. Stand where the subject stood, put the recorder where the camera was, and clap once, sharply. Record the clap and its full decay into silence.

That recording, loaded into a convolution reverb, applies that exact room to anything you put through it. It works startlingly well, and it costs one clap.

Recording the room tone and an impulse response at the end of every shoot takes ninety seconds and solves a class of problem you cannot otherwise solve.

Going the other way: de-reverb

The opposite problem — a line with too much room, that needs to sit in a drier scene — is much harder, because the reflections are mixed into the same signal as the voice.

De-reverberation tools exist and have improved considerably. They work by estimating the reverberant component and subtracting it, and they fail in a specific way: pushed hard, the voice develops a thin, gated quality with the tails of words cut short, and consonants sound processed.

The rule is the same as for noise reduction: **if you can hear the de-reverb, there is too much of it.** A partial improvement that leaves some room is better than a clean-sounding artefact, because audiences forgive a reverberant recording and do not forgive a voice that sounds like a machine.

Free option: Resolve's Fairlight has De-Reverb on some tiers; check yours. Otherwise the options are mostly paid, and the honest advice is to record closer next time.

Narration over location footage

A special case worth naming because it is so common, and the answer is counter-intuitive: **do not match it.** Narration is conventionally understood as coming from outside the scene, so it belongs dry, close and clean. Adding the warehouse's reverb to a voice-over makes it sound like somebody standing in the warehouse, which is not what a narrator is.

Match the room only for speech that is meant to be *in* the scene.

Today

Record a clap in the room you are in, sitting still with the recorder a few metres away. That file is an impulse response, and you have just started a library.

BEFORE YOU MOVE ON

An editor replaces one line in a reverberant warehouse scene with a clean studio recording. Before touching any reverb plug-in, which single step usually removes most of the mismatch, and why?

A. Rolling off the high frequencies on the replacement, since the original was recorded further away and air absorbs high frequencies with distance.
B. Matching the level of the replacement precisely to the surrounding lines, since a difference in volume is what makes an inserted line announce itself to the listener as out of place.
C. Applying a de-reverb pass to the surrounding original lines so that the whole scene matches the dry replacement rather than the other way round.
D. Laying the scene's room tone under the replaced line, because the audible hole is the sudden silence rather than the missing reflections.

The answer and why: p. 467

Lesson 38 · 9 min

What can be repaired, and what cannot

THE ONE THING TO KEEP

Hum, clicks, plosives and level faults are fully repairable; noise, reverb and mild clipping are partly repairable up to the point where the artefact is worse than the fault; and speech under the noise floor or heavily clipped is gone, which is worth saying to a client on day one.

The triage list

Four categories: fully repairable, partly repairable, repairable at a cost, and gone. Knowing which is which saves hours and saves promising a client something you cannot deliver.

Fully repairable

Mains hum. At exactly 50 or 60 Hz plus harmonics. A notch filter or a de-hum module removes it almost completely, because it is narrow, constant and at a known frequency.

Level problems. Too quiet, uneven, one channel only. Clip gain and channel mapping, covered earlier.

Clicks and pops. Individual sample-level glitches. Any click removal tool interpolates across them convincingly, because you are replacing a few milliseconds using the material either side.

Plosives. The thump when a `p` or `b` hits the microphone. A steep high-pass filter applied only to that word — 150 Hz for a few hundred milliseconds — removes the thump and leaves the word. Do it as a clip-level EQ on a short split rather than across the whole track.

DC offset. The whole waveform sitting above or below the centre line. A one-click fix in Audacity and most editors, and worth doing because it costs you headroom.

Partly repairable

Steady background noise. Hiss, air conditioning, a computer fan. Spectral subtraction, covered in the earlier lesson: sample the noise alone, build a profile, subtract. Six to eight decibels of reduction is usually safe. Beyond that you get musical noise — the watery, fluttering artefact where quiet harmonics of the voice fall below the estimate and are removed inconsistently.

The rule stands: **if you can hear the noise reduction, there is too much of it.**

Reverb. Partly, with the caveats from the previous lesson.

Mild clipping. Declipping tools interpolate the flattened tops of the waveform using the shape either side. On brief, mild clipping the result can be genuinely good. On sustained clipping the tool is inventing most of the waveform.

Wind. The low rumble comes out with a high-pass. The roaring, saturated part where the wind overloaded the capsule does not, because the microphone stopped recording the voice during it.

Repairable at a cost

Two microphones with phase problems. When two microphones record the same source from different distances, combining them produces comb filtering — a hollow, phasey sound caused by frequencies cancelling where the two arrivals are out of step.

Three fixes, in order: use one microphone only, which is usually the right answer; nudge one track by the delay between them, which you can measure from the waveform; or invert the polarity of one and see whether it improves, which it will if they were wired oppositely.

Sibilance. De-essing, covered earlier. It works, and overdone it produces a lisp.

Traffic and intermittent noise. A passing motorbike shares frequencies with the voice and is not steady, so spectral subtraction cannot profile it. Options: cut around it and use a different take, cover it with music or ambience, or accept it. Some spectral repair tools let you paint out an isolated event on a spectrogram, which works well for a discrete sound over a gap and badly for one over speech.

Gone

Say these to a client on day one:

Heavy clipping. The peaks were flattened at recording. There is no information in the flat part to recover, only plausible guesses.

Speech below the noise floor. If the voice is quieter than the hiss, no amount of processing separates them, because there is less voice than noise in every band.

Two people talking over each other on one microphone. Source separation models have made real progress here and can sometimes pull voices apart usably. They are also generating audio, with the honesty consequences that carries — which matters in journalism and anywhere the recording is evidence.

Nothing recorded. The take with the microphone switched off. This is more common than anybody admits, and the fix is re-recording, not repair.

The resynthesis question

A newer class of tool runs a recording through a model and regenerates the voice rather than filtering it. The results can be remarkable — a phone recording from across a room returning close to studio quality.

Two cautions, and they are the same ones as elsewhere in this course. The tool is generating audio, so it can shift timbre and invent detail that was never present, which makes it inappropriate anywhere the recording is evidence, journalism or a record of what somebody said. And once a tool moves past cleanup into altering words or producing a voice, that is synthetic media, with labelling requirements that differ by country and are changing, and a consent requirement that is not optional.

The order of operations

Repair before mixing, and repair in this order: de-click → de-hum → de-noise → de-reverb → EQ → compression. Each earlier step removes something the later ones would otherwise amplify.

Today

Take your worst recording and apply exactly one thing: a high-pass at 100 Hz. Then apply six decibels of noise reduction, no more. Compare with the original. That is most of what is available, and it is usually enough.

BEFORE YOU MOVE ON

An interview has a motorbike passing during a key answer. Spectral noise reduction, which worked well on the room's air conditioning, makes no useful difference to the motorbike. Why?

A. The motorbike is louder than the noise reduction's maximum attenuation, so a stronger setting would remove it at the cost of some voice quality.
B. The motorbike occupies the same frequencies as the voice, which means no frequency-domain tool can ever separate the two signals from one another in any circumstances.
C. Spectral subtraction builds a fixed profile from steady noise, and a sound that changes pitch as it passes matches no single profile.
D. Noise reduction only works on noise recorded at a constant level, so the passing vehicle's changing volume defeats the gate that the algorithm depends on.

The answer and why: p. 467

Lesson 39 · 8 min

Mixing when you do not have a studio

THE ONE THING TO KEEP

Check every mix on at least three systems and fix only the faults that appear on more than one, because each system lies in its own direction — and the low-volume test tells you more about balance than any meter.

The honest starting position

You are mixing on laptop speakers, or on earbuds, in a room with a fan. Professional mixing happens in an acoustically treated room on monitors that cost more than your computer. That gap is real and no technique closes it.

What closes most of the *practical* gap is that you are not mixing for a cinema. You are mixing for people who will hear it on a phone, on earbuds, on a laptop, or on a television across a room — which is to say, on worse systems than yours.

What each system hides from you

Laptop speakers have almost nothing below 200 Hz. You cannot hear rumble, hum, or an over-loud bass in the music. Mix on these alone and you will ship a video with a hum in it.

Earbuds and headphones give you detail and stereo but they exaggerate low frequencies and remove the room entirely. A mix that is perfect on headphones is often thin and quiet on speakers, because you set levels in an unnaturally quiet environment.

Phone speakers are the actual target for most feed video: almost no bass, a narrow midrange, mono, and often played in a noisy place. Anything that depends on low frequency is inaudible, and anything quiet is gone.

A television across a room adds distance and whatever the room does.

The multi-system check

The standard professional practice, available to you for free: check the mix on at least three systems, and fix only what is wrong on **more than one**.

That last clause is the important one. Every system lies in its own direction; a fault that appears on all three is in the mix, a fault that appears on one is in the speaker.

A practical set: headphones, laptop speakers, and your phone at arm's length with something else going on in the room.

The checks that catch real problems

Mono. Collapse the mix to mono and listen. Phone speakers are mono, most Bluetooth speakers play mono, and feed autoplay is often mono. Things that vanish in mono are cancelling, usually because a stereo effect or a duplicated track is out of phase. Resolve's Fairlight has a mono button on the monitoring section; most editors have one somewhere.

At low volume. Turn it down until it is barely audible. What survives is what the mix is actually carrying — and if the dialogue disappears before the music does, your balance is wrong. This test is more reliable than any meter for balance.

With something else going on. Play it on your phone with a fan running or a window open. This is the real listening condition for most feed video, and it is brutal about anything quiet.

With the picture off. Covered earlier in the course, and still the single best test of whether the piece holds.

Practical substitutes for a treated room

- **Mix quietly.** Loud monitoring flatters everything and fatigues your ears within twenty minutes. Set a moderate level and leave it, so your sense of "normal" stays calibrated.
- **Take breaks.** Ear fatigue is real and is why mixes get progressively brighter and louder over an evening. Twenty minutes off after an hour.

- **Use a reference.** Open a professionally mixed video of the same kind next to yours, match its perceived loudness, and switch between them. This is the closest free substitute for an accurate room: you are comparing against something you know is right rather than trusting your speakers in absolute terms.
- **Trust the meters for level, your ears for balance.** Loudness and true peak are measurements; your room cannot tell you those reliably, and the meter can.

Delivery targets to keep in your head

Pulling this block together, the numbers to hit:

- Dialogue clip gains levelled to peaks near **-12 dBFS** before processing.
- Music under dialogue around **-18 to -24 dBFS**.
- Ambience beds around **-40 to -30 dBFS**.
- Gain reduction on the dialogue compressor **3 to 6 dB** on loud words.
- Integrated loudness near the destination's target — roughly **-14 LUFS** for platforms, **-23 LUFS** for European broadcast.
- True peak at or below **-1 dBTP**.

Those numbers are a starting point that will be approximately right for most spoken-word video, and every one of them is subordinate to whether a stranger can follow the words on a phone.

The one that matters most

If you do nothing else in this block: level the dialogue, lay room tone under the whole thing, put a high-pass on every voice, and check it on a phone. Four steps, under an hour, and it is the difference most viewers will actually perceive.

Today

Play something you have published through your phone speaker at arm's length with a fan on. Note every moment you could not make out a word. That list is your mix notes.

BEFORE YOU MOVE ON

A mix sounds excellent on headphones and thin and quiet on a phone speaker. The editor raises the overall level to compensate and the platform normalises it back down. What should they have done instead?

A. Rebalanced so the dialogue carries the mix without relying on low frequencies, checking at low volume to see what survives.

B. Boosted the low frequencies, since the phone speaker cannot reproduce bass and the mix needs more of it to compensate for what the driver loses.

C. Applied heavier compression to the whole mix, which reduces the dynamic range so that the quiet parts survive the phone speaker's limited output and the noisy listening environment.

D. Exported a separate mix for phone playback, since one mix cannot serve both headphone and speaker listening.

The answer and why: p. 468

CASE STUDY · MODULE 4

A guest lecture, two ceiling fans, and a microphone at the back

An illustrative case, built from documented patterns.

A commerce college in Pune records a visiting economist's ninety-minute lecture for its online repository. One camera on a tripod at the back of the hall, its own microphone, no external recorder.

The recording is unusable as it stands, and everybody agrees on that within the first thirty seconds of playback.

The hall is a rectangular room with a tiled floor, plaster walls and a high ceiling. Six ceiling fans were running. The camera was eleven metres from the lectern. What the camera microphone captured is therefore mostly the room: a low continuous roar from the fans, a long reverberant tail on every word, and the lecturer's voice arriving at roughly the same level as the air conditioning in the corridor.

The department's media assistant, Farhan, measures what he has. The noise floor sits around minus thirty-eight dBFS. The lecturer's speech peaks around minus twenty-six. That is a gap of about twelve decibels between the voice and the room, and a good part of the voice's energy is reflection rather than direct sound.

He tries the obvious things, in the order he was taught. A high-pass at 100 Hz removes some fan rumble and changes very little, because the fans also produce broadband noise well inside the speech range. Six decibels of spectral noise reduction helps. Twelve decibels produces the watery, fluttering artefact where quiet harmonics of the voice fall below the noise estimate and are removed in some frames and kept in others. A de-reverb pass, pushed hard enough to matter, leaves the ends of words clipped and gives the lecturer a thin, gated quality.

Compression makes it worse in a way that surprises him until he thinks about it: turning the loud words down and adding make-up gain raises the fans by exactly as much as it raises the voice.

That leaves three real options, and a deadline: the repository page is meant to go live before the new term.

Publish it with the light cleanup. Six decibels of noise reduction, a high-pass, gentle compression, captions burnt in and a sidecar file. Honest, quick, and hard work to listen to for ninety minutes. Farhan tests it on his phone with a fan running, which is how a student in a hostel will hear it, and loses whole clauses.

Resynthesise the voice. One of the newer speech tools will take the recording, run it through a model and return something close to studio quality. Farhan tries thirty seconds of it and it is genuinely startling. It is also generating audio: it can shift timbre and invent detail that was never recorded. This is a lecture by a named economist that will be cited, quoted and possibly assessed. What comes back is not a recording of what happened in the room; it is a plausible reconstruction of it.

Ask for a re-record. The lecturer has left Pune. The department could ask her to read the lecture again into a phone with a cheap lavalier at her own desk, and the picture could be held as slides and stills. That is thirty minutes of her time, which she may not give, and it produces a different artefact: a reading, not a lecture, with none of the questions from the hall in it.

The decision is not really between three audio processes. It is about what the file is for.

YOUR ANSWERS

1. Compression made the recording worse. What exactly did it do?

2. Why was resynthesis rejected here when it might be acceptable elsewhere?

3. The department's fix cost nine thousand rupees and half a page of rules. Which half was doing the work?

STOP HERE

Write your answers before you turn the page. How it played out starts on p. 212.

This page is blank so that how it played out stays out of view until you turn the page.

CASE STUDY · MODULE 4

How it played out

Farhan published the lightly cleaned version, with six decibels of noise reduction, a high-pass at 100 Hz, gentle two-stage compression and a proofread caption track, and put a one-line note on the page saying the room audio was poor and captions were provided. He did not resynthesise, on the grounds that the lecture is a record and the tool would have produced audio the economist never spoke. The department then bought two wired lavaliers and a small recorder for about nine thousand rupees, and wrote a half-page rule: for any recorded lecture, a microphone goes on the speaker, the camera microphone is a backup and a sync reference, and thirty seconds of the empty hall is recorded before the audience arrives. The economist's next visit was recorded at close range and needed a high-pass and nothing else.

The questions, discussed

1. Compression made the recording worse. What exactly did it do?

A compressor turns down what exceeds a threshold and makes nothing louder; the make-up gain added afterwards raises everything that is left, including the fans, the corridor and the room tail. Narrowing the range between the loudest and quietest parts and then lifting the whole signal raises the noise floor by the same amount as the voice, so a recording with a poor noise floor can take less compression, not more.

2. Why was resynthesis rejected here when it might be acceptable elsewhere?

Because the recording is a record of what a named person said, and it will be quoted and assessed. A resynthesising tool does not filter the signal, it generates a new one from a model, so it can alter timbre and invent detail that was never present. That is tolerable for a corporate voice-over where nothing turns on exact fidelity, and it is not tolerable where the audio is evidence of what somebody said. The same tool, the same result, a different obligation.

3. The department's fix cost nine thousand rupees and half a page of rules. Which half was doing the work?

Mostly the distance, not the money. The reason a camera microphone eleven metres away sounds like a hall is the ratio of direct sound to reflected sound: direct sound falls away with distance while reflections are spread evenly through the room. A cheap lavalier at twenty-five centimetres changes that ratio more than any processing can afterwards. The recorded room tone and the backup camera track are what make the rest of the edit possible.

MODULE 4 TEST

Check what stuck

6 questions, one right answer each. This one is for you, not for a record. The answers, and the reason behind each, are in the key on p. 462. If two go wrong, re-read the module before the exam.

1 Why does a cheap lavalier at 25 cm beat an expensive microphone two metres away?

A. A shorter cable carries less interference from the mains wiring
B. The expensive microphone was designed for music rather than for speech
C. Cheap microphones have a narrower pickup pattern, which rejects the room
D. Direct sound falls away with distance while reflections do not

2 Twelve decibels of spectral noise reduction leaves a watery, fluttering sound around the voice. What happened?

A. The recording was mono, and spectral subtraction needs a stereo reference to work
B. Quiet harmonics of the voice fell below the estimate and came and went
C. The noise profile was taken from a section that also contained some speech
D. The noise reduction ran after compression rather than before it

3 One dialogue clip arrives at the compressor at minus twenty and the next at minus eight. Why does no track fader setting fix both?

A. The fader sits after the compressor, which has already treated them differently
B. The fader works in decibels while clip gain works on a linear scale
C. A fader cannot be automated, so it applies one value to the whole track
D. Compressors ignore the fader entirely and read the level from the source file on disk

4 A hissy recording sounds worse after compression than before it. Why?

A. The release is too long, so the compressor never recovers between phrases
B. The attack is too fast for speech, so the consonants are flattened
C. Make-up gain raises the hiss by as much as it raises the voice
D. Compression adds harmonic distortion, which is audible as hiss

5 Why does mastering a film to minus six LUFS gain nothing on a platform that normalises to around minus fourteen?

A. The platform re-encodes the audio, which restores the original dynamic range
B. The platform turns it down, and the dynamics were already destroyed
C. True peak levels above minus one dBTP are rejected at upload
D. Integrated loudness is measured over three seconds, so a short film is unaffected

6 What does a continuous ambience bed under a whole scene actually fix?

A. It raises the integrated loudness toward the delivery target
B. It signals to the viewer that a new location has been entered
C. It masks the compression artefacts that the platform's encoder introduces
D. It fills the holes that every cut punched in the background

5

MODULE 5

Colour, correction before grade

Take a scene of mismatched shots to a neutral, matched baseline verified on the waveform and parade, apply one look across all of it, and explain why the same LUT lands differently on a log clip and a Rec.709 clip.

Lesson 40 · 10 min

Make it right, then make it feel like something

THE ONE THING TO KEEP

Neutralise and match every shot against the scopes first, because a look applied to unmatched shots produces unmatched results.

Two different jobs that get mixed together

Colour work is two separate operations and doing them in the wrong order wastes most of an evening.

Correction is making the shot right: neutral whites, sensible exposure, matching shots that were filmed minutes apart under drifting light so they belong in the same scene.

Grading is making it feel like something: teal shadows, warm skin, crushed blacks, a bleached daylight look.

Correction first, always, and here is the mechanism. A grade is a set of adjustments built on assumptions about where mid-grey sits, where skin sits and where black sits. If shot one is 600 K too warm and a stop underexposed, and shot two is neutral, the identical grade lands on two different starting points and produces two different results. You end up regrading every shot from scratch and the scene still does not match. Correct first so all your shots agree with each other, then one grade travels across all of them.

Your eyes are lying and the scopes are not

Two things make eyeballing colour unreliable, and neither is about skill.

Chromatic adaptation: look at a warm image for twenty seconds and your visual system rebalances until it looks neutral. Grade for ten minutes and you will keep pushing further in the same direction without noticing. Then you look at it the next morning and it is orange.

And your viewing conditions lie. An uncalibrated laptop panel, a room lit by a yellow bulb, a phone with automatic warmth at night — each shifts what you are seeing by more than the adjustments you are making.

So use the scopes. They are in Resolve's free version, in Kdenlive, and in Premiere.

- **Waveform** for exposure. The vertical axis is brightness. You can see instantly whether highlights are clipping at the top or shadows are crushed against the bottom.
- **RGB Parade** for white balance. Three waveforms side by side, red then green then blue. In a neutral shot the bottoms should line up and the tops should line up. If blue sits higher at the bottom than red and green, your shadows are blue — pull blue lift down until the bases align. This one reading fixes most white-balance problems in about fifteen seconds.
- **Vectorscope** for saturation and hue. There is a diagonal reference line running toward orange, roughly eleven degrees off vertical, and human skin of every colour falls close to it. If your faces are pointing well off that line, something is wrong regardless of how the picture feels.

An order of operations that works

1. **Black point.** Bring the darkest part of the image down until it just reaches the bottom of the waveform without crushing detail you want.
2. **White point.** Bring the brightest part up until it approaches the top without clipping.
3. **Mid-tones.** Now set the overall brightness of faces, which is the only exposure the viewer actually judges.
4. **White balance**, using the parade.
5. **Saturation**, last and modestly.

Only then reach for a look.

Figure 5.1

The order that stops you grading everything twice

Black point
Down until the darkest part just reaches the bottom of the waveform

↓

White point
Up until the brightest part approaches the top without clipping

↓

Mid-tones
The brightness of faces, which is the only exposure a viewer consciously judges

↓

White balance
Align the bottoms, then the tops, of red, green and blue on the parade

↓

Saturation
Last, and modestly

↓

Only now, a look
One grade now travels across every shot in the scene

A look assumes where black, mid-grey and skin sit. Applied to unmatched shots it lands on a different starting point in each one, and you regrade the scene from scratch.

LUTs are not magic and this is why they disappoint

A LUT — lookup table — maps every input colour to an output colour. Apply a film-emulation LUT to a shot and it can look superb or it can look like a bruise, and the difference is almost never the LUT.

Most cinematic LUTs are built to receive a **log** image: a flat, grey, desaturated signal from a camera set to record maximum dynamic range. Feed one a normal, already-contrasty image and it applies a second dose of contrast and saturation on top of the first, blowing highlights and blocking shadows. Feed it a shot that has not been corrected and it faithfully amplifies the colour cast.

So: correct first, then apply the LUT, then dial its strength down — most look better at 50 to 70%.

Related, and worth knowing before you buy a camera: **log recording only helps if the camera has real 10-bit capture.** Log spreads the tonal range thinly so you can redistribute it later. In 8-bit there are only 256 levels per channel to spread, so when you add the contrast back you get visible banding in skies and gradients. Log on an 8-bit phone or entry camera generally costs you more than it gains. Shoot a normal profile, expose it properly, and grade gently.

The washed-out export

You grade a piece, it looks right in the timeline, you export it, and the file looks flat and grey in the player. Almost nobody guesses this correctly the first time.

It is usually a **data-level mismatch**. Video can be tagged as full range, where 8-bit values run 0 to 255, or video range, where black sits at 16 and white at 235. If a file is encoded as one and interpreted as the other, blacks lift and whites dull, and the whole image looks washed. Nothing is wrong with your grade.

Look for a data levels or range setting in your export dialogue, set it to Auto or Video, and check the result in a different player — VLC is free and honest. If it looks right in VLC and wrong in a browser, the tag is the suspect.

The money rule here is unusually strong

Colour is the one area where the free tool is not a compromise at all. DaVinci Resolve's colour page is what feature films are graded on, and it is in the free version. Nothing about node-based grading, power windows, tracking or scopes is held back. If you are learning grading, you are learning it on the same software the professionals use, at no cost.

Kdenlive and Shotcut have basic colour wheels and curves and will handle a straightforward correction. On a phone, colour work is limited to global adjustments and filters; treat it as correction only.

Today

Take one shot, open the RGB parade, and align the bottoms of the three channels. Do nothing else. Compare before and after.

BEFORE YOU MOVE ON

An editor applies a purchased cinematic LUT to footage from a phone. Skies clip to white, shadows go solid black, and skin turns orange. The LUT looks fine in the seller's samples. What is the most likely explanation?

A. The phone footage is 8-bit, and LUTs require 10-bit source data to function at all.
B. The LUT is built for a different colour space and needs to be converted before it can be applied to any footage.
C. The LUT expects a flat log image as input; the phone footage already has contrast and saturation baked in, so the LUT adds a second helping on top.
D. The LUT was applied at 100% strength, and every LUT is designed to be used at partial opacity.

The answer and why: p. 470

Lesson 41 · 8 min

Why grading happens in a graph

THE ONE THING TO KEEP

A node graph lets you choose what each adjustment applies to and in what order, so correct first and style after, name every node, and use the still store to carry one shot's grade across a whole scene.

A stack of adjustments, with a structure

Most editing software applies effects as a list: filter one, then filter two, then filter three. A colour page applies them as a **graph** — a set of connected operations where you choose what feeds what.

The reason is that colour work is not a single chain. You want the exposure correction to apply to everything, the skin-tone adjustment to apply only to the face, and the film look to apply after both but before the vignette. A list cannot express that. A graph can.

Resolve's colour page is node-based and free. If you are working in Kdenlive, Shotcut or CapCut, you are stacking filters in order, which does the first half of this and not the second — that is a real limitation and it is worth knowing which half you are missing.

Serial nodes

The default. Each node receives the output of the previous one and passes its own output on.

`Node 1 → Node 2 → Node 3`

A sensible starting layout for a shot:

1. **Balance.** Neutralise the white point and set black and white points.
2. **Exposure.** Set the mid-tones where faces sit.

3. **Look.** Contrast, saturation, colour shifts.
4. **Secondary.** A key or window for one specific area.
5. **Output.** A final small trim, vignette, grain.

Five nodes, named. Naming them takes ten seconds and means you can come back in a week and change the exposure without hunting through an unlabelled stack.

Why order matters, mechanically

Each node operates on the values it receives. Put a heavy contrast node before your balance node, and the balance node is now working on an image whose shadows have already been crushed — it has fewer distinct values to work with, and the correction it applies is different.

The practical rule: **subtractive and corrective operations first, creative operations after.** Fix, then style. It is the same principle as correction before grade, applied inside a single shot.

Parallel and layer nodes

These exist for the case where you want two adjustments to apply to the same input and then be combined, rather than one feeding the other.

Parallel mixes the outputs of several branches evenly. Useful when you are correcting two different areas of the frame independently — the sky in one branch, the ground in another — and neither should affect the other.

Layer stacks them with a blend mode and an opacity, like layers in an image editor. Useful for compositing a look over the corrected image at a chosen strength.

Most grades never need these. Use them when a serial chain is forcing an adjustment through a correction you did not want it to pass through.

Keys and the alpha input

Every node has an input for the image and an input for a **key** — a mask that says where the node applies.

A key comes from a qualifier (select a colour range), a window (draw a shape), or an external matte. Feed it in and the node's adjustment applies only inside it. This is the mechanism behind every secondary correction, and the next-but-one lesson uses it.

The still store, and why it changes everything

This is the feature that makes shot matching possible, and it is free.

Grab a **still** from a graded shot — in Resolve, right-click the viewer and Grab Still, which stores both the frame and the entire grade. Then:

- **Wipe** the still against another shot to compare them directly, split screen, in the viewer.
- **Apply the grade** from the still to another shot with a middle-click, which copies the whole node tree.

The workflow that follows: grade one shot properly, store it, apply it to every other shot from the same setup, then adjust each individually. On a ten-shot scene this is five minutes rather than an hour.

Copying grades

Three mechanisms, each for a different case:

- **Still store middle-click** — copies one grade to one clip.
- **Ripple all** — applies a change to every selected clip at once. Useful for a look across a finished scene.
- **Groups** — Resolve lets you assign clips to a group and grade the group's pre-clip and post-clip levels, so a change to the group affects all its members while each keeps its own individual grade. This is the correct structure for a scene: individual corrections per shot, one shared look for the group.

Versions

Each clip can hold several complete grades as **versions**, switchable from the right-click menu. Before trying something different, make a new version rather than modifying the one that works.

It costs nothing, and it means the answer to "can we see the other one again" is two clicks rather than an afternoon.

Today

Grade one shot with five named nodes in the order above, grab a still, and apply it to a second shot from the same setup. That is the whole workflow of the block in ten minutes.

BEFORE YOU MOVE ON

An editor places a strong contrast node first in the chain and a white balance node after it. The balance correction behaves oddly and cannot fully neutralise the cast. Why does reordering fix it?

A. The balance node is now working on an image whose shadows and highlights have already been crushed, so it has fewer distinct values to redistribute.
B. Contrast nodes must always be last in a chain, because any operation after them re-expands the range and reintroduces the cast that the balance node has just removed from the image.
C. White balance is a per-clip attribute rather than a node operation, so it is applied before the graph regardless of where the node sits in the chain.
D. Node order does not affect the result mathematically, so the apparent improvement is a difference in how the viewer renders the preview rather than a real change.

The answer and why: p. 471

Lesson 42 · 8 min

The three wheels, and what each one touches

THE ONE THING TO KEEP

Gain multiplies so it sets the white point, lift adds so it sets the black point, gamma is a power curve so it moves the mid-tones with both ends pinned — set them in that order and read every move on the waveform.

Not three brightness controls

The primary colour wheels are labelled **Lift, Gamma, Gain** — or, in some tools, Shadows, Midtones, Highlights. They look like three versions of the same control. They are three different mathematical operations, and using the wrong one is why a correction that should take thirty seconds takes twenty minutes.

Gain: multiply

Gain multiplies every value. Turn it up and white gets brighter, mid-grey gets brighter proportionally, and black — which is zero — stays at zero, because anything times a multiplier is still zero.

So gain is the **white point** control. It sets where the brightest part of your image lands without moving the black.

Lift: add

Lift adds a constant to every value. Turn it up and black rises off zero, mid-grey rises by the same amount, and white rises too — though at the top the effect is proportionally small because white was already large.

So lift is the **black point** control. Lifting the black is how you get a faded, milky look; lowering it is how you get a deep, contrasty one.

In practice most implementations pivot lift around the white point so the top stays put, which is why it feels like a shadows control even though the operation is an addition.

Gamma: a power curve

Gamma applies an exponent. Black stays at zero and white stays at one, because zero and one raised to any power are still zero and one. Everything in between moves.

So gamma is the **mid-tone** control, and it is the one that actually changes how bright a face looks without touching either end. It is the control you reach for most often and the one beginners use least.

Offset, the fourth one

Many tools also have **Offset**, which adds a constant to everything with no pivot at all — the whole image moves up or down together, including the black.

Offset is the right tool for a straightforward white balance on a log image, because it shifts the entire channel rather than reshaping it. Balance with offset in the shadows, then set the other three.

Figure 5.2

Four controls, told apart by which end of the waveform holds still

	White stays put	White moves
Black stays at zero	**Gamma** A power curve. Both ends are pinned and the mid-tones move	**Gain** Multiplies. Zero times anything is still zero, so it sets the white point
Black moves	**Lift** Adds, pivoted near the top, so it sets the black point	**Offset** Adds with no pivot at all. The whole trace slides together

Set gain, then lift, then gamma. With both ends already pinned, gamma redistributes what is between them and cannot undo the first two.

The order that works

1. **Gain** — set the white point. Bring the brightest part of the image up until it approaches the top of the waveform without clipping.
2. **Lift** — set the black point. Bring the darkest part down until it just reaches the bottom without crushing detail you want.
3. Repeat once, because they interact slightly.
4. **Gamma** — set the mid-tones, which in practice means setting the brightness of faces. This is the only exposure the viewer consciously judges.
5. **Balance** — use the colour of each wheel, or offset, to neutralise.

The reason gamma comes third: with black and white pinned, gamma redistributes what is between them without moving either end, so it cannot undo the work of the first two steps.

Reading the waveform while you do it

Every one of these moves has a visible signature on the waveform, which is how you learn what they do:

- **Gain** — the top of the trace moves and the bottom stays. The whole shape stretches from the bottom.
- **Lift** — the bottom moves and the top stays. The shape stretches from the top.
- **Gamma** — both ends stay and the middle of the trace moves up or down.
- **Offset** — the entire trace slides up or down as one.

Make each move while watching the waveform, once, and the abstraction becomes concrete.

Balance on the parade

The RGB parade shows red, green and blue waveforms side by side. For a neutral image:

- The **bottoms** should align. If blue sits higher than red and green at the bottom, your shadows are blue — pull the blue component of lift down until the bases meet.
- The **tops** should align. If red sits higher at the top, your highlights are warm — pull the red component of gain down.

This single reading fixes most white-balance problems in about fifteen seconds, and it is considerably more reliable than your eye, which adapts to a cast within twenty seconds of looking at it.

Where the free tools are

Resolve's primary wheels are in the free version with no restriction, along with offset, log wheels, all the scopes, and the primaries bar mode for typing exact numbers.

Kdenlive has three-way colour wheels and a scope panel. Shotcut has colour grading with shadows, midtones and highlights. Both do the correction work described here perfectly well; what they do not have is the node structure, keys and tracking of the next lessons.

On a phone, colour adjustment is global sliders and filters. Treat it as correction only.

Today

Open one shot, open the waveform, and move gain, lift and gamma one at a time while watching the trace. Five minutes, and you will never confuse them again.

BEFORE YOU MOVE ON

A shot's faces are too dark but the highlights and shadows are already exactly where the editor wants them. Which control changes the faces without disturbing either end, and why?

A. Lift, because it raises the shadows where the under-exposed faces sit and leaves the highlights untouched.
B. Offset, because it shifts the whole signal evenly and preserves the relationships between the tones that are already correct.
C. Gain, because reducing it slightly compresses the top of the range and brings the mid-tones up relative to the highlights.
D. Gamma, because a power curve leaves zero and one fixed and only redistributes the values between them.

The answer and why: p. 471

Lesson 43 · 8 min

Curves, and the shape of contrast

THE ONE THING TO KEEP

A curve maps input values to output values, so a gentle S steepens the mid-tones where faces live and a reverse S produces the faded film look — and Lum versus Sat pulled down in the shadows stops dark noise reading as coloured.

What a curve is

A tone curve maps every input value to an output value. The horizontal axis is what comes in, the vertical is what goes out. A straight diagonal line changes nothing. Bend it and you have changed the relationship between input and output across the whole range.

The wheels give you three broad handles. A curve gives you as many as you want, which makes it both more powerful and easier to ruin an image with.

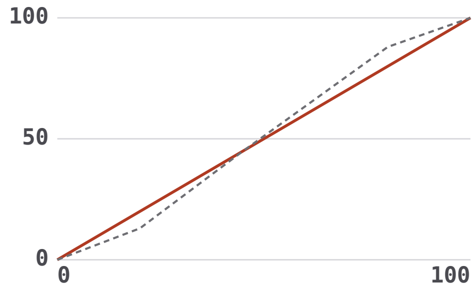
The S-curve

The most useful shape in colour work. Pull the bottom quarter of the line down slightly and the top quarter up slightly, leaving the middle on the diagonal.

What this does: darks get darker, lights get lighter, and the mid-tones — where faces live — get *steeper*, meaning small differences in the middle become larger differences in the output. That is why an S-curve reads as "more contrast" and also as "more detail in the faces", even though no detail was added.

Figure 5.3

What a gentle S-curve does to the middle



Across: Input value, per cent

Up: Output value, per cent

—— Straight line, no change

– – A gentle S-curve

Between the two control points the line is steeper than the diagonal, so small differences in the mid-tones, where faces live, come out larger. No detail was added.

Keep it gentle. Two control points, each moved a small amount. A steep S-curve clips both ends and produces the look of an over-processed phone photo.

The reverse S, or the faded look

Pull the bottom up and the top down. Shadows lift off black, highlights come down from white, and the image loses contrast and gains a matte, filmic quality.

This is the mechanism behind almost every "film emulation" and "faded" look you have seen. It is a small amount of lifted black and rolled-off white. Once you can see it as a curve, you can build any of those looks yourself rather than downloading one.

Where to put the control points

A practical constraint: **do not put a control point where the important values live.** A control point pins that value in place; the curve bends around it.

So for a shot where faces sit around 45% brightness, put your control points at roughly 20% and 80% and leave the region around 45 unpinned, so it can move as a whole rather than being anchored in the middle of the face.

Per-channel curves

Switch the curve to red, green or blue individually, and you are now shaping colour rather than brightness.

Lift the bottom of the blue curve and your shadows go blue. Pull down the top of the blue curve and your highlights go warm. Together, those two moves produce the cool-shadows-warm-highlights split that most commercial grading uses, and you have built it from two control points rather than a preset.

The warning: per-channel curves change saturation as well as hue, because saturation is the distance between channels. Small moves.

Hue-versus curves

Resolve's colour page has a set of curves that take hue as their input, and they solve problems the ordinary curves cannot:

- **Hue vs Sat** — select a hue range and change its saturation. The standard use is taking saturation out of an over-saturated red, or desaturating a distracting green sign without touching anything else.
- **Hue vs Hue** — shift one hue toward another. Used for pulling an orange-red skin tone toward a more natural orange, or moving a foliage green away from yellow.
- **Hue vs Lum** — change the brightness of one hue.

- **Lum vs Sat** — change saturation according to brightness. The standard use is pulling saturation out of the deep shadows, which stops noise in the darks looking coloured and is one of the cheapest ways to make cheap footage look better.

These are all in the free version, and Lum vs Sat on the shadows is worth applying almost every time.

Curves against wheels

When to use which:

- **Wheels** for the primary correction: white point, black point, mid-tones, balance. Broad and predictable.
- **Curves** for shaping: contrast, a specific tonal region, a colour relationship across the range.
- **Hue-versus curves** for isolating one colour without drawing a mask.

A common working grade is: wheels for the correction node, a gentle S-curve for the contrast node, Lum vs Sat for the shadows, and a hue-versus curve if one colour is misbehaving. That is a complete grade and none of it needs a plug-in.

How far to go

The reliable check is the before-and-after toggle, used frequently. Chromatic adaptation means you drift: twenty minutes into a grade your eye has accepted whatever you have been doing and you will keep pushing.

Second check: look at it tomorrow. Almost everybody's grade from last night is too warm and too saturated.

Today

Take one shot, apply a gentle S-curve with two control points, and toggle it on and off. Then apply a much stronger one and see where it starts to clip on the waveform.

BEFORE YOU MOVE ON

An editor applies an S-curve and the faces look better defined, but they describe it as the curve having 'added detail'. What actually happened?

A. The curve sharpened the mid-tone region, which raises acutance at edges and reads as additional detail in the face.
B. The curve is steeper through the mid-tones, so existing small differences there are mapped to larger output differences and become more visible.
C. The curve raised saturation in the mid-tones, and the additional colour separation between skin and background makes the face read as more detailed than it was before.
D. Contrast recovers detail that compression had discarded in the mid-tones, which is why the effect is strongest on heavily compressed footage.

The answer and why: p. 472

Lesson 44 · 9 min

Making a scene belong to itself

THE ONE THING TO KEEP

Grade one hero shot completely and bring every other shot to it on a split wipe, matching black point, white point, mid-tone, balance and saturation in that order on the scopes — and judge the result at playback speed, not frame by frame.

The job

Two cameras, or one camera over two hours, or an interview where the sun went behind a cloud. The shots do not match, and cutting between them makes the viewer aware they are watching a construction.

Matching is the most common colour job in real work and it is almost entirely mechanical. It is not a matter of taste.

Pick the hero shot

Choose one shot in the scene as the reference. Usually the one with the best exposure, the most screen time, or the most important face.

Grade that one properly and completely. Everything else is brought to it. Do not grade the shots in timeline order; grade the hero, then match.

The matching procedure

For each remaining shot, in this order:

1. Find a common element. A face is best, because skin has a narrow expected range and the viewer judges the scene by it. Failing that, a wall, a white object, a piece of clothing that appears in both.

2. Compare on the waveform, not by eye. Put the hero shot's still in the viewer as a split wipe and look at the two waveforms. Match the **black point** first — the bottom of the trace — then the **white point**, then the **mid-tone** level of the common element.

3. Match the balance on the parade. Compare where red, green and blue sit at the bottom and the top, and adjust until the two shots' parades have the same relationship.

4. Match saturation on the vectorscope. Compare the extent of the trace. Two shots with matched exposure and different saturation still do not match.

5. Look at the faces last. Both should sit close to the skin-tone line on the vectorscope, which runs from the centre toward orange at about eleven degrees off vertical. People of every skin colour fall near this line; what varies is how far out along it they sit, not the angle.

The wipe, which is the whole technique

The reason to grab stills is this: you can display the hero shot and the shot you are matching side by side, in the same viewer, with a movable split.

In Resolve, grab a still from the hero, double-click it in the gallery to load it, and drag across the viewer to move the wipe. Now you are comparing two images directly rather than comparing an image to a memory. Memory is useless here — twenty seconds after looking away, your visual system has re-adapted.

The match is complete when you can move the wipe across the join and not see where it is.

Shot match tools

Resolve has an automatic **Shot Match** function: select several clips, right-click, Shot Match to This Clip. It analyses and applies a correction.

It is genuinely useful as a first pass and it is wrong often enough that you check every result. It fails predictably when the two shots contain very different amounts of a colour — matching a wide shot of a green field to a close-up of a face will pull the face green, because the algorithm is matching overall distributions rather than the subject.

Use it, then correct it by hand on the wipe.

The order for a whole scene

1. Grade the hero shot.
2. Copy its grade to every other shot from the **same camera and setup**. Those will be close already.
3. Match each remaining shot on the wipe.
4. Group the whole scene and apply the look to the group, so one adjustment moves all of them together.

5. Play the scene at speed and watch for jumps. Anything you notice at normal speed is a genuine mismatch; anything you can only see by pausing is not worth more time.

That last point is a real time-saver. Matching is judged at 25 frames a second, not frame by frame.

What cannot be matched

- **Different colour temperature of the light source, when it is in frame.** You can match the faces or you can match the window, not both, unless you use a window or a key to treat them separately.
- **Different depth of field.** No colour operation changes lens compression or focus falloff, and a shot at f/1.4 will never cut cleanly against one at f/8 in the same scene.
- **Heavy clipping in one shot.** The information is not there, and pulling the other shot down to match makes both worse.
- **Different motion blur from mismatched shutter.** Mentioned in the frame-rate lesson, and visible in exactly the same way.

When a match is impossible, the editorial answer usually beats the colour answer: put a cutaway between the two shots so they never touch.

Today

Take two shots from the same scene, grab a still from one, and wipe them against each other. Match the black points only. That one step usually does half the work.

BEFORE YOU MOVE ON

An editor uses automatic shot match to bring a close-up of a face in line with a wide shot of a green field. The face comes back with a green cast. What does that reveal about how the tool works?

A. It matched the overall colour distributions of the two frames, and green dominates the wide shot's while the close-up has almost none.
B. The tool sampled the wrong reference frame from the wide shot, and choosing a frame with less field visible in it would produce a correct match between the two clips.
C. The tool cannot analyse skin tones and requires a manual skin-tone qualifier to be set before it will produce a usable match on any shot containing a face.
D. The close-up was shot at a different white balance, and automatic matching cannot correct white balance errors, only exposure differences between clips.

The answer and why: p. 472

Lesson 45 · 9 min

Changing one part of the frame

THE ONE THING TO KEEP

Qualifiers select by colour and windows by shape, both feeding a node's key — and because a qualifier builds its mask from chroma, 4:2:0 footage yields a mask at a quarter the resolution of the picture, which no setting repairs.

Primary against secondary

A **primary** correction applies to the whole image. Everything so far has been primary.

A **secondary** applies to part of it — one colour, one region, one object. Two mechanisms produce the selection: a qualifier, which selects by colour, and a window, which selects by shape.

Qualifiers, and what they actually do

A **qualifier** builds a mask from ranges of hue, saturation and luminance. Pick a colour with the eyedropper, widen the ranges until the thing you want is selected, and the node applies only inside that selection.

Standard uses: pulling down an over-saturated red shirt, changing the colour of a sign, warming skin without warming the background, taking the green out of foliage that is competing with the subject.

The procedure that works:

1. Turn on the **highlight** view, which shows the selection as a matte — white where selected, black where not. You cannot judge a key by looking at the graded result.
2. Sample the colour, then widen hue, then saturation, then luminance, one at a time.
3. Add **blur** to the matte. A hard-edged key always looks cut out; two or three pixels of blur solves it.
4. Add a little **denoise** to the matte if it is crawling.
5. Restrict it with a window if it is also selecting something elsewhere in the frame.

Where qualifiers fall apart, and why

The chroma subsampling lesson at the start of this course pays off here.

A qualifier builds its mask from colour information. In **4:2:0** footage — every phone, every consumer camera — colour is recorded at half the horizontal and half the vertical resolution. So the mask is being built from a quarter as much information as the picture has, and the edges come back blocky, stepped and unstable between frames.

On 4:2:2 or 10-bit footage, qualifiers work well. On 8-bit 4:2:0, expect to blur the matte heavily and to restrict the correction to something subtle. A large hue change on a 4:2:0 key will show its edges.

This is not a failure of the software and no setting fixes it. It is the file.

Windows

A **window** is a drawn shape — circle, rectangle, polygon, or a freehand curve — with a softness setting.

Standard uses: a vignette, brightening a face, darkening a bright window behind a subject, drawing attention to one part of the frame.

The softness setting is the whole craft. A hard-edged window is visible as a shape. Set softness high enough that you cannot see where it ends, then raise the adjustment inside it until it does something. Almost every window in professional work is softer and subtler than a beginner's.

The principle worth holding: **a good window is one the viewer never notices.** If somebody can see that you brightened the face, it is too strong or too hard.

Tracking

A static window on a moving subject is useless, so windows track.

Resolve's tracker analyses the movement of what is inside the window and moves the window to follow, supporting pan, tilt, zoom, rotate and perspective. In practice you place the window on the first frame, press track forward, and check the result.

When it fails — the subject leaves frame, something crosses in front, the movement is too fast — you can switch to frame-by-frame keyframing, or track in reverse from a later frame, or split the shot at the problem point and track the halves separately.

The tracker is in the free version and it is good. It is one of the clearest cases where free Resolve is doing something that used to cost thousands.

Combining a window and a qualifier

The most useful secondary technique. Draw a window around the general area, then qualify inside it.

Example: you want to warm somebody's face. A skin-tone qualifier alone will also select the wooden table and the brick wall. Draw a loose tracked window around the head, qualify skin inside it, and the selection is only the face.

In Resolve, both the window and the qualifier feed the same node's key, and the node's key controls let you combine them.

When not to bother

Secondaries are where grading time disappears. Before spending forty minutes on a key, ask whether the shot is on screen for three seconds in a piece nobody will pause.

The honest order of value: correction and matching first, because everyone perceives those; a simple look second; secondaries last, and only where something is actively distracting. A distracting bright object behind a subject is worth a window. A slightly over-saturated jumper is not.

Today

Draw a soft circular window on a face, raise the brightness inside it by a small amount, and toggle it. Then raise the softness until you cannot tell where it ends.

BEFORE YOU MOVE ON

An editor pulls a skin-tone qualifier on 8-bit 4:2:0 phone footage to warm a face. The matte edges are blocky and flicker between frames. Which response addresses the actual cause?

A. Increase the qualifier's luminance range, since the blockiness indicates the key is dropping parts of the face that fall outside the selected brightness band.

B. Blur and denoise the matte heavily, and keep the correction subtle enough that a low-resolution key can carry it.

C. Switch the node to a layer node so the correction is blended rather than applied directly, which hides the edges of the matte behind the underlying image.

D. Re-render the clip to a 10-bit intermediate codec before keying, so the qualifier has more colour information to work with when it builds the matte.

The answer and why: p. 473

Lesson 46 · 9 min

Colour spaces, and why it looks different in the browser

THE ONE THING TO KEEP

A pixel value only means a colour once you know the space, white point and transfer function — so the washed-out export is usually a full-range file being read as video range, and the browser mismatch is usually gamma 2.2 against 2.4.

The complaint

The grade looks right in the editor. The export looks different in VLC. It looks different again in Chrome, and different again on the phone. Nothing is broken, and there are three separate mechanisms in play.

A colour space is three things

- **Primaries** — which exact red, green and blue the numbers refer to. Rec.709 for HD, Rec.2020 for UHD and HDR, DCI-P3 for cinema and most modern phone screens, sRGB for the web.
- **White point** — what "white" means. Usually D65.
- **Transfer function** — how the stored numbers map to light output. This is the gamma, and it is where most of the trouble lives.

Two files with identical pixel values in different colour spaces are different colours. The values only mean something when you know the space.

The gamma problem, specifically

Rec.709 is defined for broadcast with a display gamma of **2.4**, in a dim viewing environment. sRGB, the web standard, is approximately **2.2**, intended for a brighter room.

That difference is small in the numbers and visible in the shadows: a file graded against 2.4 and displayed at 2.2 has lifted, milkier blacks. Displayed the other way, it has crushed ones.

So a grade that is correct on a calibrated broadcast monitor at 2.4 looks slightly washed in a browser, and the browser is not wrong — it is applying the standard it is supposed to apply.

The practical resolution for most people delivering to the web: **grade at gamma 2.2**, in a normally lit room, on the kind of screen your audience uses, and check the export in a browser. If you are delivering to broadcast, follow their spec, which will say 2.4.

Data levels: the washed-out export

The second mechanism, and this one produces a large, obvious error rather than a subtle one.

Video can be encoded as **full range**, where 8-bit values run 0 to 255, or **video range** (also called legal or limited), where black sits at 16 and white at 235. The extra space at each end exists historically for signal overshoot.

If a file is encoded as one and interpreted as the other, everything shifts. Encode video-range and interpret as full, and blacks lift to grey and whites dull — the classic washed-out export. Encode full-range and interpret as video, and blacks crush and highlights clip.

Figure 5.4

Data levels: how the file was written against what the player assumes

	Player assumes full range	Player assumes video range
Written full range, 0 to 255	**Correct** Black is black and white is white	**Crushed** Blacks crush and highlights clip. Detail is lost at both ends
Written video range, 16 to 235	**Washed out** Blacks lift to grey and whites dull. The classic flat export	**Correct** Black is black and white is white

ffprobe reports color_range as pc for full and tv for video. A blank field means nothing was tagged, and then every player guesses for itself.

The file carries a flag saying which it is. Some encoders write it incorrectly and some players ignore it.

Diagnosis:

```
ffprobe -v error -select_streams v:0 \
  -show_entries
stream=color_range,color_space,color_primaries,color_trc \
  -of default=noprint_wrappers=1 out.mp4
```

`color_range=tv` is video range, `pc` is full. If the field is blank, nothing was tagged and every player will guess.

The fix is in the export dialogue — a Data Levels or Range setting — and the check is to open the export in VLC, which is free and honest, and in a browser. If they agree with each other and disagree with your timeline, the timeline is the odd one out.

HDR, briefly and honestly

High dynamic range uses a different transfer function — PQ or HLG — a wider colour space, and much higher peak brightness. It is genuinely different, not a stronger grade.

Three honest points. Grading HDR properly needs a reference monitor that costs more than most people's entire setup, so grading it on a laptop is guesswork. Platforms handle HDR uploads inconsistently and the tone-mapping down to SDR for viewers without HDR screens is frequently ugly. And for most work — an interview, a tutorial, a short film for the web — HDR buys nothing that a well-graded SDR delivery does not.

It matters if you are delivering to a streamer who asks for it. Otherwise, deliver Rec.709 SDR and know why.

Colour management in Resolve

Two ways to handle mixed sources:

Manual. Colour management off. Put an input transform on each log clip as the first node — a Colour Space Transform from the camera's space to Rec.709 — and grade after it. More steps, and you can see exactly what is happening at every point.

Managed. DaVinci YRGB Color Managed, or ACES. You tell the system each clip's input space and the output space, and it converts everything into a common working space automatically. Fewer steps, consistent handling of mixed cameras, and a pipeline you have to understand to debug.

For most people learning, manual is the better teacher. Switch to managed when you have several camera types in one project and the manual approach is costing you time. As the setup lesson noted, switching mid-project changes the input to every grade you have already built.

Today

Run the `ffprobe` command above on your last export and read the `color_range` field. If it is blank, you have found why the file looks different in different players.

BEFORE YOU MOVE ON

An export looks correct in the editor and noticeably washed out — lifted blacks, dull whites — in a media player. `ffprobe` reports `color_range=pc`. What is the most likely mechanism?

A. The player is applying gamma 2.2 where the grade was made at 2.4, which lifts shadow detail across the image.

B. The file was graded in a wider colour space than Rec.709, so the player is clamping out-of-gamut colours and flattening the image in the process.

C. The file is tagged full range but the player is treating it as video range, so values are being remapped into the 16 to 235 band.

D. The export used a low bitrate, so compression has reduced contrast in the shadows and highlights where the encoder allocated fewest bits to the image.

The answer and why: p. 473

Lesson 47 · 8 min

Building a look instead of downloading one

THE ONE THING TO KEEP

A look is five deliberate decisions — contrast shape, a shadow-highlight colour split, saturation shaped by luminance, a hue shift or two, and grain — so build and save your own rather than applying a LUT that expects a log image it is not receiving.

Why a downloaded LUT usually disappoints

A LUT is a lookup table: every input colour maps to an output colour. It is a fixed transformation with no knowledge of your image.

Most cinematic LUTs are built to receive a **log** signal — flat, grey, wide-range. Feed one a normal, already-contrasty picture and it applies a second dose of contrast and saturation on top of the first. Highlights blow, shadows block, and skin goes orange.

Feed one an uncorrected shot and it faithfully amplifies whatever cast was there.

So: correct first, then apply, then reduce its strength — most look better at 50 to 70%. And check what the LUT expects. A LUT labelled for a specific camera's log profile is not a general-purpose look.

What a look is made of

Almost every look you admire is some combination of five things, and each is one control:

1. Contrast shape. An S-curve, or a lifted black with a rolled-off white. This is the biggest single contributor and it is two control points.

2. A colour split between shadows and highlights. Cool shadows and warm highlights is the commercial default; the reverse gives a colder, more clinical feel. Built with per-channel curves, or with the colour of lift and gain.

3. Saturation shaped by luminance. Lum vs Sat pulled down in the deep shadows and often in the extreme highlights. This stops colour noise in the darks and is present in essentially every professional grade.

4. A hue shift on one or two ranges. Usually pulling greens toward cyan or yellow, and keeping skin close to the reference line.

5. Texture. Grain, and sometimes a very slight softening or bloom.

Build those five deliberately once, save it as a still or a PowerGrade, and you have your own look that you understand and can adjust.

Grain, and why it helps

Adding grain to a digital image seems backwards and it does two useful things.

It **dithers banding.** Smooth gradients in 8-bit footage band into visible steps; a small amount of noise breaks the steps into a texture the eye reads as continuous. This is a genuine technical fix, not an affectation.

And it **unifies sources.** Footage from three cameras with different noise characteristics looks like one thing once a consistent grain sits over all of it.

Keep it subtle — visible as texture when you look for it, invisible when you do not. Resolve has a Film Grain effect; Kdenlive and Shotcut have noise filters; FFmpeg's `noise` filter will do it at export.

The cost: grain is detail that changes every frame, which is exactly what a compressor finds expensive. Heavy grain plus a low upload bitrate produces a mushy result, so if you grain, upload generously.

Halation and bloom

Halation is the soft red-orange glow around bright objects in film stock, caused by light scattering through the emulsion and reflecting back. Bloom is the general spread of highlights.

Both are now standard in "filmic" looks and both are easy to overdo. A little bloom on practical lights in a night scene reads as atmosphere. Bloom on everything reads as a smeared lens.

Matching a reference

The most reliable way to develop a look is to steal one deliberately.

Take a frame from something you admire — a film still, a music video frame, an advertisement. Put it in your gallery beside your shot and wipe between them. Then ask, in order:

- Where is their black point compared to mine?
- Where is their white point?
- What colour are their shadows? Their highlights?
- How saturated are the mid-tones?
- Where does skin sit?

Answer those five on the scopes and you will have reproduced most of the look, and you will understand what you did, which a LUT would not have taught you.

The honest limits

A look cannot fix a bad image. Applied to something soft, noisy or badly exposed, a strong look makes the faults more visible, because contrast and saturation amplify everything including the problems.

A look cannot substitute for lighting. The commonest disappointment: somebody grades flat, evenly lit footage toward a moody film look and gets a dark, murky image rather than a dramatic one. Contrast between light and shadow in a graded image comes from contrast that was in the room.

A strong look dates. The orange-and-teal grade of the late 2000s now reads as a period marker. Restraint ages better.

Today

Take a frame from a film you like, load it into your gallery, and wipe it against one of your own shots. Write down the five answers. That list is a look you can build.

BEFORE YOU MOVE ON

An editor applies a popular film-emulation LUT to well-exposed, standard-profile footage from a phone. Highlights blow out and skin goes orange. What is the mechanism, and what is the fix?

A. The LUT is designed for a different camera manufacturer, so its colour science does not match the phone's and only a LUT built for that phone will work correctly.

B. The LUT expects a flat log input and is adding contrast and saturation on top of an image that already has both.

C. The phone recorded 8-bit, so the LUT has too few levels to work with and is producing banding that reads as blown highlights and shifted skin tones.

D. The footage is in the wrong colour space for the LUT, and a colour space transform node placed after the LUT will convert the result back into Rec.709 correctly.

The answer and why: p. 474

Lesson 48 · 8 min

Legal levels, and the check before you export

THE ONE THING TO KEEP

Broadcast-legal means values between 16 and 235 in 8-bit video range, readable as lines on the waveform — and every quality-control check happens on the exported file in a normal player, because several classes of error exist only in the export.

What "legal" means

Broadcast signals have defined limits. In 8-bit video range, black is **16** and white is **235**, with the space above and below reserved for overshoot in the analogue chain. Values outside that band are **illegal**, and a broadcaster's automated quality control will reject a file that contains them.

For web delivery nothing is rejected, which is why most people never encounter this. It still matters, for two reasons: clipped values at either end are detail you have destroyed, and a file that exceeds the range interacts badly with the range conversions of the previous lesson.

Reading it on the waveform

Set your waveform scope to show values in the broadcast range, and the limits appear as lines.

- Anything flat against the **top** line is clipped highlights. A flat horizontal line at the top of the trace means a region of the image has gone to pure white with no detail in it.
- Anything flat against the **bottom** is crushed shadows.
- A trace that extends beyond either line is illegal.

A small amount of clipping is normal and often intentional — a light source in frame, a reflection off chrome. A wide flat plateau at the top means a face or a sky has lost its detail.

Chroma can be illegal too

Less obvious: a colour can be within the luminance limits and still be out of range, because the combination of luminance and chroma produces an RGB value outside the band.

Heavily saturated reds are the usual culprit. On a vectorscope, colours extending past the target boxes at the corners are the warning.

The fix is a **soft clip** or a broadcast-safe limiter, which rolls off the extremes rather than cutting them. Resolve has both a soft clip control on the primaries and a Broadcast Safe setting in Project Settings with selectable tolerances. Turn it on for broadcast delivery; leave it off for web, where it costs you saturation you are allowed to have.

The quality-control pass

Before any export goes out, run this list. It takes the length of the film plus ten minutes and it catches the errors that are expensive to discover after delivery.

Watch the whole thing, at full screen, without touching anything. Not scrubbing. The whole thing. This is the only step that catches most problems, and it is the one people skip when they are late.

Then check, specifically:

- **Black frames.** A single frame of black between two clips, usually from a gap on the timeline. Zoom out and look for gaps; in Resolve, a gap shows as a break in the track.
- **Audio dropouts.** A track muted, a clip with no audio, a gap in the room tone bed.
- **Sync at the end.** Drift is invisible at the start and obvious at minute eight. Check the last scene specifically.

- **Spelling in every title and super.** Names especially. Getting somebody's name wrong is the error people remember.
- **The caption track**, read end to end against the audio.
- **Levels**: integrated loudness and true peak, measured, from the earlier block.
- **The first and last frame.** Not black, and the last frame holds for long enough to not feel cut off.
- **Placeholder material.** Temporary music, a scratch voice-over, a `LOGO HERE` card, a watermarked stock clip. This is the mistake that reaches a client most often.
- **The export's actual properties.** Run `ffprobe` on the finished file and confirm resolution, frame rate and duration are what you intended. It takes five seconds and it catches the project that was set up wrong three weeks ago.

Watch the export, not the timeline

The important discipline: this check is performed on the **exported file**, played in a normal player, not on the timeline in your editor.

The timeline shows you a rendered preview that may differ from the file in range handling, colour tagging and audio routing. Several classes of error exist only in the export. Watching the timeline and calling it checked is how a washed-out file ships.

A written checklist

Keep it as a text file in your template project and tick it every time. Not because the steps are hard, but because the failure mode is doing this at 1am on the delivery day, from memory, with three of them missed.

Today

Export thirty seconds of something, open it in VLC, and compare it against the timeline. Then run `ffprobe` on it and read the resolution and frame rate. That is the habit.

BEFORE YOU MOVE ON

An editor performs a full quality-control pass by watching the timeline end to end in their editor at full screen, finds nothing wrong, and delivers. The client reports a washed-out picture. What did the method miss?

A. Timeline playback uses the render cache, which is generated at lower quality than the final export and can disguise faults in the delivered file.
B. The editor should have watched at the delivery resolution rather than full screen, since scaling the preview hides artefacts that appear at native size.
C. The preview and the exported file can handle data range and colour tagging differently, so the mismatch exists only in the file.
D. Watching the whole piece is insufficient on its own, and only a waveform scope reading of the exported file can confirm the levels are within the legal band.

The answer and why: p. 474

CASE STUDY · MODULE 5

One shot under a tube light, and the window behind it

An illustrative case, built from documented patterns.

A single editor in Nashik grading a nine-minute film for a private clinic. Two days of filming, mixed daylight and overhead tube lighting, delivery on Friday.

The scene is a two-minute sequence in a consulting room. A doctor explains what the clinic does about follow-up appointments, and a patient — who has consented, on paper, for this one film — describes what changed for her.

Most of it matches without much trouble. The room was lit by a large window on the left and the shots were filmed within forty minutes of each other. The exception is one shot: the patient's answer about the phone call that brought her back, which is the emotional centre of the sequence and the only take of it. It was filmed twenty minutes later, after the sun had gone behind the block opposite, with the overhead tube lights switched on.

On the parade the difference is obvious. The tube light is strongly green in the mid-tones, so the patient's skin sits well off the reference line, and the green is not uniform across the frame: the window behind her still carries the last of the daylight and reads cold. The editor corrects for the green in the skin and the window goes magenta. He corrects for the window and the patient goes green again. There is no single primary correction that fixes both, because there are genuinely two different light sources in the frame.

The technical options are clear and so are their costs.

A qualifier and a tracked window. Draw a soft window around the patient's head and shoulders, track it across the shot, and qualify skin inside it so the correction touches only her face. Then treat the window behind her separately. This is the textbook answer and it works. The complication is the footage: the camera recorded 4:2:0, so the qualifier is building its mask from a quarter of the picture's resolution. Blurring the matte enough to hide the stepped edges means the correction bleeds onto the wall behind her head, and the patient moves, so the window has to track through about forty seconds of small movements. The editor's estimate is an evening, and he has two other scenes to grade.

Accept a compromise primary. Split the difference: leave the skin slightly green and the window slightly magenta, and reduce saturation in both so neither is loud. Twenty minutes. The scene reads as competent, and the one shot the film is built around is the least attractive in it.

Do not let the shots touch. Put a cutaway between the daylight shots and the tube-light shot, so the viewer never sees a direct join. The clinic's own waiting-room signage, the appointment book, a wide of the corridor. Cutting away also means the mismatch is compared across two seconds of something else rather than across a single frame, and the eye is considerably more forgiving of that.

The third option is an editorial answer to a colour problem, and it is the one that keeps getting forgotten by people who are enjoying the colour page.

There is also a decision that is not about this shot at all. The whole project was set up with colour management off, the timeline at Rec.709 gamma 2.4, and every log clip carrying its own input transform. The editor is tempted, standing in front of this mess, to switch the project to managed colour and let it sort the sources out. Doing that would change the starting image under every grade he has already built on the other seven minutes.

YOUR ANSWERS

1. Why could no single primary correction fix both the patient and the window?

2. The qualifier produced stepped, unstable edges. What property of the source file caused that, and what setting fixes it?

3. He was tempted to switch the project to managed colour. Why was that the wrong moment?

STOP HERE

Write your answers before you turn the page. How it played out starts on p. 262.

This page is blank so that how it played out stays out of view until you turn the page.

CASE STUDY · MODULE 5

How it played out

He did the third option first, because it cost ten minutes: a two-second insert of the appointment book, taken from material shot on day one, placed at the join. Then he did a reduced version of the first: a soft tracked window around the patient, a gentle green correction inside it and nothing else, with the matte blurred heavily and the adjustment kept small enough that its edges could not be seen. That took about fifty minutes rather than an evening, because he stopped trying to make the two shots identical and only tried to make the join survive playback at speed. He did not switch to colour management. The scene was signed off without comment, which is the outcome a grade is aiming for. His note to himself afterwards was about the shoot rather than the grade: switch the tube lights off, or switch them on at the start, but do not change the light source halfway through a scene.

The questions, discussed

1. Why could no single primary correction fix both the patient and the window?

Because there were two different light sources in the frame with different colour temperatures. A primary correction applies to the whole image, so any shift that neutralises the green of the tube light also pushes the daylight in the window magenta by the same amount. The only ways out are a secondary that treats one region separately, or an editorial decision that stops the two shots being compared directly.

2. The qualifier produced stepped, unstable edges. What property of the source file caused that, and what setting fixes it?

The footage was 4:2:0, so colour was recorded at half the horizontal and half the vertical resolution of the picture. A qualifier builds its mask from colour, so the mask carries a quarter of the information the image does and its edges come back blocky and crawling. No setting on the keyer fixes it. The practical responses are to blur the matte and keep the correction subtle, or to avoid a colour-based selection entirely.

3. He was tempted to switch the project to managed colour. Why was that the wrong moment?

Because colour management changes the image every grade is built on. Seven minutes of the film had already been corrected against the unmanaged starting point, so switching would have altered the input to all of that work and required regrading it. Managed colour is a decision for the start of a project, or for a project with several camera types where the manual route is costing time. It is not a repair for one difficult shot.

MODULE 5 TEST

Check what stuck

6 questions, one right answer each. This one is for you, not for a record. The answers, and the reason behind each, are in the key on p. 469. If two go wrong, re-read the module before the exam.

1 Every shot in a scene came off one camera in one profile, filmed over two hours in changing light. Why does one LUT applied to all of them produce unmatched results?

A. Most LUTs expect a log image, and only some of the shots were recorded in log
B. A LUT is applied per clip rather than per timeline, so each one is computed separately
C. A LUT maps input values to output values, and the inputs differ
D. LUTs are interpolated, so they behave differently on 8-bit and 10-bit sources

2 A face is too dark, but the black point and the white point are both where you want them. Which control moves the face without disturbing either end?

A. Gamma
B. Offset
C. Lift
D. Gain

3 On the RGB parade the bottoms of the three traces do not align, and blue sits highest. What is wrong, and what fixes it?

A. The image is over-saturated; reduce saturation until the traces meet
B. The file is tagged full range and is being displayed as video range instead
C. The highlights are cool; pull the blue component of gain down
D. The shadows are blue; pull the blue component of lift down

4 A skin qualifier on phone footage produces a blocky, crawling matte. Which of these is true?

A. Widening the luminance range will recover the missing edge detail
B. No setting fixes it; the mask is built from quarter-resolution colour
C. Raising the matte denoise restores the resolution the chroma channels lost
D. Switching the project to 10-bit processing rebuilds the chroma before the key

5 An export has grey blacks and dull whites in one player and looks correct in another. ffprobe reports a blank color_range field. What is happening?

A. Nothing was tagged, so each player is guessing the range
B. The file is 8-bit, so it bands in players that dither and not in those that do not
C. One player is applying the Rec.709 matrix while the other applies Rec.2020's
D. The display gamma differs between the two players, 2.2 against 2.4

6 Why does adding a little grain before export reduce visible banding in a sky?

A. It forces the encoder into a higher bit depth, where the steps are finer
B. Grain raises the bitrate, and more bits mean finer gradients
C. Noise scatters pixels either side of the boundary, so the contour softens
D. The platform's encoder removes the grain again, and takes the banding away with it

6

MODULE 6

Text, captions and graphics

Put text on a moving picture that survives platform compression on a phone — sized and weighted against a stated safe area, timed to a read-aloud test, animated with easing you chose — and ship both burnt-in captions and a sidecar subtitle file, including in a script your editor does not shape correctly.

Lesson 49 · 9 min

Most of your audience has the sound off

THE ONE THING TO KEEP

Burn captions in for feeds and upload a real subtitle file as well, and never publish an auto-transcript you have not read.

The default is silence

Video in a feed autoplays muted. People watch on a bus, in a shared room, in a lecture, next to a sleeping child. A large fraction of the views on anything you post will begin, and often end, without sound.

That is before you count the people who need captions rather than prefer them: deaf and hard-of-hearing viewers, and — very relevant to this classroom — the large number of people watching in their second or third language, who read English considerably faster than they parse spoken English at conversational speed with an unfamiliar accent.

Captions are not an accessibility box to tick at the end. On short-form video they are the primary channel.

Two kinds of caption, and you often want both

Burnt-in (open) captions are rendered into the pixels. They always appear, on any platform, in any player, in autoplay-muted feeds. They cannot be turned off, cannot be translated, cannot be read by a screen reader or indexed by a search engine, and small thin text can be mangled by aggressive compression.

Sidecar subtitles are a separate file — `.srt` or `.vtt` — that the player overlays. They can be switched off, restyled, auto-translated, indexed by search, read by assistive technology, and used to generate a transcript. They do not exist in a muted autoplay preview unless the platform chooses to show them.

For a vertical short: burn them in **and** upload the sidecar file. For a long-form YouTube video or a course: sidecar, always, and burn in only key phrases. Every editor in this course exports SRT, including Kdenlive and Shotcut.

Figure 6.1

Burnt-in captions against a sidecar file

Burnt in, part of the pixels

- Appear in a muted autoplay feed
- Work in any player, on any platform
- Cannot be switched off or translated
- Not indexed, and invisible to a screen reader
- Thin type is mangled by the compressor

A sidecar .srt or .vtt

- Switched off, restyled, auto-translated
- Indexed by search and read by assistive tools
- Becomes a transcript for nothing
- Absent from a muted autoplay preview
- Shapes Indic and Arabic scripts correctly

For a vertical short, ship both. For long-form, the sidecar always, and burn in only the phrases that have to survive a silent feed.

Automatic captions and how they fail

Speech recognition is genuinely good now. On clear speech, in a well-represented accent, with common vocabulary, expect somewhere in the region of 90 to 95% word accuracy.

Accuracy falls off sharply on exactly the things this audience deals with daily: non-native and regionally accented English, code-switching mid-sentence between English and another language, proper nouns, place names, technical terms and acronyms.

And the failure mode is the dangerous part. Recognition does not produce gibberish that you notice. It produces confident, grammatical, wrong sentences that read perfectly and say something else. "Deploy the model" becomes "destroy the model". Nobody proofreads a caption track that scans fine.

So: never publish an auto-caption track unread. Reading a five-minute track takes about four minutes and it is not optional.

Free ways to generate them: YouTube's own automatic captions, which you then edit in Studio; Whisper, an open-weights speech model that runs locally — `whisper.cpp` will transcribe on an ordinary laptop CPU with no GPU and no upload, which matters if the audio is confidential; and various free web transcribers, with the obvious caveat that you are uploading the audio to someone else. Resolve has transcription built in on some versions and tiers, so check yours.

Making captions legible

The specifics matter more than the styling.

- **Two lines maximum**, roughly **37 to 42 characters per line**. Longer lines force the eye to travel and people lose the picture.
- **Reading speed** around **15 to 17 characters per second** is comfortable. Above about 20 people stop reading and start skimming.
- **Minimum time on screen** of about a second, even for a single word, or it flashes.
- **Break lines at grammatical joins.** Break before a preposition or a conjunction, not in the middle of a noun phrase. "The reason the export / failed" reads badly; "The reason / the export failed" reads well.
- **Contrast has to survive a changing background.** Text over video sits on a different colour every frame. Use a semi-opaque box behind it, or a solid outline, or a tight drop shadow. Plain white text with nothing behind it will disappear over a bright sky.
- **Never use a hairline or thin weight.** At phone size, after platform compression, thin strokes break up. Medium or bold, always.

- **Stay out of the interface.** On vertical short-form, the bottom fifth or so of the frame is covered by the username, description and buttons, and the top by the status bar and platform chrome. Put captions in the middle third. Editors lose text to this constantly because the editing canvas has no interface on it.

Titles

The same discipline as Design Foundations, with two additions specific to video.

Keep text inside a safe margin — roughly 5% in from every edge — because players, televisions and platform crops all eat the outer band. And give titles enough time: the honest measure is that you should be able to read the text aloud twice at a normal pace while it is on screen. You will always underestimate this, because you wrote it and you already know what it says.

Resist animating for the sake of it. A title that slides, spins, bounces and glows has spent the viewer's attention on the animation rather than the words.

Free tools

Resolve has a full subtitle track and exports SRT. Kdenlive has a decent subtitle editor. CapCut generates captions automatically and styles them well, which is a large part of why it dominates short-form editing. Subtitle Edit is free, open source, and better at the fiddly work of timing and line-breaking than any video editor. On a phone, CapCut and most mobile editors will auto-caption and burn in; proofread on the phone before exporting.

Today

Take a video you have already published, read its automatic caption track end to end, and count the wrong words. That number is what your silent viewers received.

BEFORE YOU MOVE ON

A creator burns captions into a vertical video and positions them near the bottom of the frame, where they looked balanced on the editing canvas. On the platform, viewers say the captions are unreadable. What went wrong?

A. Burnt-in captions are compressed along with the picture, so the platform's encoder degraded the text.
B. The platform's interface — username, description and buttons — overlays the lower portion of the frame, which the editing canvas does not show.
C. The video was exported at the wrong aspect ratio, so the platform cropped the bottom of the frame away.
D. Sidecar subtitles were also enabled, and the two caption tracks are being drawn on top of each other.

The answer and why: p. 476

Lesson 50 · 9 min

Type that survives a phone and a compressor

THE ONE THING TO KEEP

Size text as a percentage of frame height with a floor around 2.5 per cent, never use a light weight because the compressor destroys thin strokes, give every line its own contrast mechanism against a changing background, and hold it long enough to read aloud twice.

Three things working against you

Text on video faces problems that text on a page does not.

The background changes every frame. White text that is perfectly legible over a dark jacket disappears the moment the subject moves in front of a window.

The compressor hates thin strokes. A hairline serif at 1080p is a few pixels wide. The encoder allocates bits by how much changes between frames, and fine detail over moving video is exactly what it discards first. Thin type comes back broken and fringed.

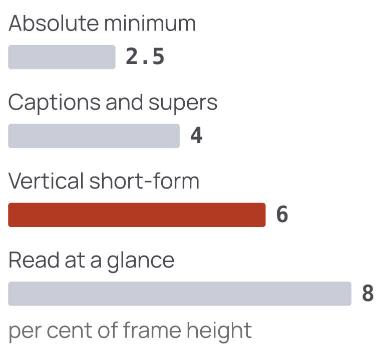
It is being read on a phone. A 1080p frame on a phone held at arm's length subtends roughly the same visual angle as a paperback page, and the reader has a few seconds rather than as long as they like.

Sizing, in numbers

Work in percentages of frame height, because that is what transfers between resolutions.

Figure 6.2

Type size as a share of frame height



At 1080p, 2.5 per cent is 27 pixels and 6 per cent is 65. Check the numbers at the delivery size: type that looks generous on a 4K canvas is half what you thought.

- **Absolute minimum for body-sized text**: about **2.5% of frame height**. At 1080p that is 27 px; at 2160p, 54 px.
- **Comfortable for captions and lower thirds**: **3.5 to 4.5%** of frame height. 38–49 px at 1080p.
- **Vertical short-form captions**: larger again, **5 to 7%** of frame height, because they are the primary channel.
- **A title meant to be read at a glance**: 6 to 10%.

If you are designing at 4K and delivering at 1080p, check the numbers at the delivery size. Type that looks generous on a 4K canvas is half the size you thought on the file people watch.

Weight, not size, is usually the fix

A common mistake is making text bigger when the problem is that it is too light. **Never use a hairline, thin or light weight on video.** Medium, semibold or bold, always.

The reason is the compressor. A bold stroke is 6 or 8 pixels wide and survives quantisation; a 2-pixel stroke does not. This is why type that looks elegant in your editor looks broken after upload.

Similarly: avoid high-contrast display serifs, where the thin strokes are very thin. A slab serif or a sturdy sans survives.

Contrast over a moving background

Four mechanisms, in rough order of how often they are the right answer:

1. **A semi-opaque box behind the text.** Black at 60–80% opacity. Ugly in a design sense, completely reliable, and used by every broadcaster on earth for exactly that reason.
2. **A drop shadow**, tight and dark: offset 2–4 px, blur 4–8 px, 60–80% opacity. Nearly invisible as an effect, and it keeps white text readable over most backgrounds.
3. **An outline or stroke**, 2–4 px in a contrasting colour. Standard for short-form captions. It survives any background and it looks like short-form captions, which may or may not be what you want.
4. **A gradient scrim** — a soft dark gradient across the bottom third, under the text. Less heavy than a box and works when the text position is fixed.

What does not work: plain white text with nothing behind it, which will vanish over a bright sky exactly once in every video and you will not notice because you know what it says.

Position and the safe area

From the earlier lesson: keep text within **90% of the frame** — 5% in from every edge.

For vertical short-form specifically:

- The bottom **15–20%** is covered by the caption, username and interface buttons.
- The top **10%** carries the status bar and platform chrome.
- Put anything that must be read in the **middle third**.

Every editor draws these guides. Turn them on, and check a real screenshot from the actual platform if you can, because the safe zones differ between them and change.

Timing

The reliable test: **you should be able to read the text aloud twice, at a normal pace, while it is on screen.**

You will always underestimate this, because you wrote it and you know what it says. A three-word name super needs about two and a half seconds; a sentence needs four or five.

For captions specifically, the numbers from the earlier lesson hold: 15–17 characters per second is comfortable, above 20 people stop reading, and a minimum of about one second on screen even for a single word.

Consistency

Pick two type sizes and one family for a whole piece, and stick to them. The most common amateur tell is not bad typography but *inconsistent* typography — a different size, position and style for every piece of text because each was made separately.

Make one title template and duplicate it. Every editor supports saving a title as a preset or a template; in Resolve you save it to the Media Pool or the Edit page's Titles, in Kdenlive as a title template file.

Today

Export thirty seconds with a line of text in a thin weight and the same line in bold, upload it somewhere that re-encodes, and look at both on a phone. That comparison settles the weight question permanently.

BEFORE YOU MOVE ON

A title in an elegant thin typeface looks crisp in the editor and comes back fringed and broken after upload. Why does increasing the point size not fix it?

A. The platform re-encodes at a lower resolution, so larger text is downscaled by the same proportion and gains nothing from the increase in size.

B. Scaling a title up in the timeline resamples it rather than re-rendering the type, so the larger version is a blurred copy of the small one.

C. The fringing is chroma subsampling on the text's colour edges, which is a property of the delivery format and cannot be affected by typography at all.

D. The encoder discards fine detail that changes between frames, and a thin typeface keeps thin strokes at any size it is set at.

The answer and why: p. 477

Lesson 51 · 8 min

Names, supers and the end of the film

THE ONE THING TO KEEP

A super answers 'who is this' two or three seconds after somebody starts speaking, its second line says why they matter rather than their job title, and every super in a piece comes from one duplicated template — while credits discharge real licence obligations from your log.

The super, and what it is for

A **super** — short for superimposition, also called a lower third or an astons — identifies somebody on screen. Name, and a role or a reason we should listen to them.

Its job is narrow: answer "who is this" at the moment the viewer asks it, and then leave. Everything else about it is decoration.

Motion, Sound and Music covers designing and animating one. This lesson is about the editorial decisions around it, which are the ones that go wrong.

When it appears

On the person's first substantial line, not their first appearance. If somebody walks through a shot at minute one and speaks at minute three, super them at minute three.

Two to three seconds after they start speaking, not on their first frame. The viewer needs a beat to register a new face before they can absorb a name; arriving simultaneously means both are missed.

Once per person per piece, unless the piece is long — over about fifteen minutes, or after a substantial gap, a repeat is helpful.

Never over the end of a sentence that matters. The eye goes to new text on screen and stops listening for about a second. Put supers under the less important part of the answer.

What goes in it

Two lines at most:

```
Priya Ramesh
District Health Officer, Nashik
```

The second line is the one people get wrong. It should say why this person's view matters, not their full job title. "Worked on the scheme for eleven years" is more useful than "Deputy Assistant Director (Programmes)".

And get the spelling right. Send the exact wording to the subject before you lock, or check it against something they wrote themselves. A misspelled name is the error people remember and the one that is most easily avoided.

Consistency is the whole craft

One template, one position, one duration, one animation, for every super in the piece. Duplicate rather than recreate.

The usual failure is that the fourth super is made in a hurry, twenty pixels higher than the others, with a longer second line that wraps. Nobody can say what is wrong; the piece just feels less made.

Position: left-aligned, starting around 10% in from the left, with the baseline around 75–80% down the frame in 16:9. In vertical, higher — the bottom fifth is interface.

Titles inside the piece

Section titles, chapter cards, statements. Two rules.

Keep them on the picture rather than on black where you can. A full-screen card stops the film; text over a continuing shot does not.

Give them the same treatment as supers — same family, same weight, a size step up. Two sizes for a whole piece is plenty.

Credits

The end of the film, and the part most people improvise.

What to include: everybody who worked on it, the subjects who agreed to appear, music with the attribution string each licence requires, archive sources, and thanks. For a commissioned piece, the client's requirements.

Where the licence obligations land. CC-BY requires attribution in the manner specified; some free music tiers require a specific credit line and a code. This is a contractual obligation, not a courtesy, and the credits are usually where it is discharged. Take the exact strings from the licence log you kept.

Roll or cards. A roll — credits scrolling upward — is standard for anything over five minutes. Cards are better for short pieces and for anything watched on a phone, because a roll at readable speed on a small screen is either too fast to read or too long to sit through.

Speed: a credit roll should move slowly enough that a name is on screen for at least three seconds. Most amateur rolls are twice as fast as that.

Do not roll credits over a black screen for ninety seconds on a platform that measures retention. Cards, or credits over a final shot, or a short roll. The last thirty seconds of a video being empty is a measurable cost with no benefit.

The end card

For web delivery, the last few seconds often carry a call to action: subscribe, a link, the next video.

Keep it short — three to five seconds — and put it over something, not over black. And make sure the actual end of the piece has landed first; an end card that arrives before the film has finished saying its last thing undercuts it.

The things that must appear somewhere

Depending on the piece and the jurisdiction, some text is not optional: a sponsorship or paid-partnership disclosure, a label on synthetic or AI-generated content where required, safety warnings for content that could trigger photosensitive epilepsy, and any statutory credit a funder requires. These requirements differ by country and platform and are changing; check them for your case rather than assuming.

Today

Open a piece you have made with more than two supers and compare their positions, sizes and durations. If any differ, you have found the template you should have built.

BEFORE YOU MOVE ON

An editor places each interviewee's name super on the first frame the person appears, and testing shows viewers frequently miss the names entirely. What is the mechanism?

A. The supers are on screen too briefly, and extending each one by two seconds will give viewers time to read them properly.
B. New text draws the eye away from the speaker, so the face and the name compete for the same first second of attention.
C. Viewers on phones cannot read text in the lower third because platform interface elements cover it, regardless of when it appears in the shot.
D. Supers placed on a cut are re-encoded across a keyframe boundary, so the first few frames of the text are degraded and hard to read.

The answer and why: p. 477

Lesson 52 · 8 min

Keyframes, curves and the retime graph

THE ONE THING TO KEEP

Two keyframes and an interpolation is all timeline animation is, so ease every move because nothing physical changes value at a constant rate — and remember that a speed ramp needs easing on the retime frame graph rather than steps on the speed graph.

The mechanism

A **keyframe** records the value of a parameter at a point in time. Two keyframes with different values, and the software interpolates between them, computing a value for every frame in between.

That is all animation is inside an editor: position, scale, rotation, opacity, crop, and any effect parameter, each with a value at a moment in time.

Motion, Sound and Music covers the principles of movement — easing, weight, anticipation. This lesson is the mechanics inside a timeline and the places they go wrong.

Where they live

In Resolve's Edit page, every clip has an Inspector with a diamond next to each parameter. Click it to set a keyframe at the playhead. The keyframes appear in a strip under the clip once you expand it.

The important thing to know: **Resolve has two separate keyframe systems**. Transform, crop and composite keyframes live on the Edit page's Inspector. Colour page keyframes live in the colour page's own keyframe panel, and they are separate — a keyframe on the Edit page will not appear in the colour page's timeline. This confuses everybody once.

Kdenlive puts keyframes in the effects panel with an inline curve editor. Shotcut has keyframe buttons in each filter's panel. Premiere uses the Effect Controls panel.

Linear is the tell

The default interpolation is linear: constant rate of change from the first keyframe to the second.

Nothing physical moves that way. Objects accelerate and decelerate. So a linear move reads as mechanical, and it is the single most reliable sign that a piece was animated by somebody who has not thought about it.

The fix is easing. In Resolve, right-click a keyframe and choose Ease In, Ease Out or Ease In and Out. The keyframe changes shape and the motion gains acceleration.

As a default: **ease out of the start and into the end**. Fast in the middle, slow at both ends.

The curve editor

For more control, the curve editor shows the interpolation as a graph — time on the horizontal axis, value on the vertical — with bezier handles on each keyframe.

What the shape means:

- A **straight line** is linear.
- A curve that starts **flat and steepens** is an ease out of the beginning: slow start, fast finish.
- A curve that **steepens then flattens** is an ease into the end: fast start, gentle arrival.
- An **S** is both, which is the natural-looking default.
- A curve that goes **past the final value and comes back** is an overshoot, which reads as bounce or springiness.

Dragging the handles is more precise than choosing a preset, and it is worth ten minutes to learn because every animation tool everywhere uses the same graph.

The three animations worth mastering

A slow push. Scale from 100% to 105% across a shot's duration, linear or very gently eased. Gives life to a static shot. This is the one you will use most.

A text entrance. Opacity 0 to 100 over 8–12 frames, with a small position move of 10–20 px, eased out. Subtle, professional, and the entire lower-third animation most pieces need.

A masked reveal. Animate a crop or a mask edge across the text so it appears to be wiped on. Free in any editor, and it looks considerably more made than a fade.

The retime curve

Speed changes have their own keyframe graph, and it works differently enough to trip people.

Resolve's **Retime Curve** offers two views of the same information:

- **Retime Frame** — a graph of which source frame plays at which timeline moment. A steeper line means faster playback. This is the one where a speed ramp is a curve rather than a corner.
- **Retime Speed** — a graph of the speed value itself, as a series of steps.

The practical consequence: a speed ramp made with the Speed view alone produces a sudden change at each step, which is visible as a jolt. To ramp smoothly, switch to the Frame view and ease the keyframes, so the rate of change itself changes gradually.

Two traps

Keyframes belong to the clip, not the timeline. Trim the clip and the keyframes move with it, which is usually what you want and occasionally is not — if you trimmed the head, your carefully timed animation now starts at a different point in the shot.

Scaling past 100% costs resolution. A push from 100% to 130% on a 1080p clip in a 1080p timeline is upscaling for most of its duration. On a 4K source at 1080p it is free, which is another reason to shoot larger than you deliver.

Today

Animate a scale from 100% to 110% with linear keyframes, watch it, then ease both ends and watch it again. The difference is the whole lesson.

BEFORE YOU MOVE ON

An editor builds a speed ramp by setting several keyframes in Resolve's Retime Speed view — 100%, then 200%, then 100% — and the result jolts at each change. What does switching to the Retime Frame view allow that the Speed view does not?

A. It plots which source frame plays at each moment, so the keyframes can be eased and the rate of change itself becomes gradual rather than stepped.
B. It shows more keyframes, since the Frame view exposes intermediate keyframes that the Speed view hides from the editor while working.
C. It enables optical flow interpolation, which generates intermediate frames and smooths the transition between the different speed segments of the ramp.
D. It applies motion blur between the speed segments, which disguises the discontinuity at each keyframe where the playback rate changes abruptly.

The answer and why: p. 478

Lesson 53 · 9 min

Graphics without a subscription

THE ONE THING TO KEEP

Fusion inside Resolve, Blender for 3D, Inkscape for vector artwork and a browser for data-driven graphics cover every free route to motion graphics — and one title template you made and reuse beats a downloaded pack everybody else is also using.

What the paid tool is for

Adobe After Effects is the name in every motion-graphics job advertisement, and it is genuinely the standard for complex compositing, expression-driven animation and a plug-in ecosystem nothing else matches.

For the graphics most video work actually needs — a lower third, an animated logo, a chart, a diagram, a simple 3D title, a masked reveal — there is a free route to every one of them. Here is the map.

Fusion, inside Resolve

Resolve's **Fusion page** is a full node-based compositor, included in the free version, and it is the closest free equivalent to After Effects for video work. Where After Effects is layer-based, Fusion is node-based — the same distinction as the colour page, and the same advantages.

What it does: text with full 3D transforms, particle systems, planar and point tracking, keying, rotoscoping, masks, paint, expressions, and a 3D compositing environment.

What it costs: a genuinely steep learning curve. Fusion's interface is unlike everything around it, and the documentation assumes a compositing background. Budget a weekend before it is useful.

The practical route in: learn four nodes — **Text+**, **Transform**, **Merge**, **Background** — which together build any lower third or title card. Add **Tracker** and **Mask** and you can do most of what a real job needs.

Blender, for anything three-dimensional

Free, open source, and the actual answer for 3D text, logo animation, product visualisation and camera moves through a scene.

It also has a video sequence editor and a compositor, neither of which you need if you have Resolve. What you want Blender for is the 3D: extruded text with real lighting and reflections, an object rotating, a camera flying through a title.

The workflow: build in Blender, render with a transparent background as a **PNG image sequence** or a ProRes 4444 / WebM file with alpha, then import into your edit and composite over the picture. Rendering to an image sequence rather than a video file is the safer route — if the render crashes at frame 800 you keep the first 799.

Inkscape, Krita, GIMP

For anything static that then gets animated in the editor.

Inkscape — vector, free, and the right place to make a diagram, a map, a logo or an icon set. Export as PNG at four times the size you need, or export the SVG and render it at any size.

Krita and **GIMP** — raster painting and image editing, free. Krita has frame-by-frame animation built in, which is the free route to hand-drawn animation.

The pattern: make the artwork properly in a design tool, export with transparency, animate in the editor. This is faster than trying to design inside a video editor, whose drawing tools are poor everywhere.

HTML and CSS, for data and repetition

An underused free route. If you can write a little HTML, you can build a graphic in a browser, animate it with CSS, and record or render it.

This is particularly good for anything **repetitive or data-driven** — twenty lower thirds from a spreadsheet, a chart that updates from a CSV, a table of results. Write it once, loop over the data, render twenty PNGs.

Rendering options: a headless browser screenshot per frame, or simply a screen recording if the animation is simple. For a batch of stills, Puppeteer or Playwright in a short script.

FFmpeg, for the mechanical jobs

Burning subtitles, overlaying a PNG, generating a colour background, making a slideshow from stills, adding a watermark. All one-liners:

```
# overlay a transparent PNG at 40px from the bottom left
ffmpeg -i input.mp4 -i logo.png \
  -filter_complex "overlay=40:H-h-40" -c:a copy out.mp4
```

Not animation, but it removes a class of fiddly manual work.

What you actually need

The honest hierarchy for somebody starting:

1. **Title templates in your editor**, duplicated and re-used. This covers 70% of real graphics needs.
2. **Keyframed transforms** on imported PNGs from Inkscape. Another 20%.
3. **Fusion** for the compositing jobs — a tracked mask, a screen replacement, a particle effect.
4. **Blender** when something genuinely has to be three-dimensional.

Most people reach for the complicated tool first and produce worse work more slowly than somebody with one well-made title template.

The trap of the template pack

Downloadable motion-graphics packs are everywhere, free and paid. Two honest cautions. Most are built for After Effects and do not import into anything else, whatever the listing says. And a piece assembled from somebody else's templates looks like a piece assembled from somebody else's templates, because everybody else is using the same ones.

One template you made, consistently applied, beats twenty you downloaded.

Today

Make one lower third in Fusion using Text+, Background, Transform and Merge, and save it to the Media Pool. That is the graphic you will use for the next year.

BEFORE YOU MOVE ON

An editor renders a 3D animated logo in Blender to a compressed MP4 with a black background, then imports it and sets the blend mode to Screen to remove the black. Why is this a worse route than rendering a PNG sequence?

A. MP4 files cannot be imported with transparency, so the blend mode is the only option available and it is therefore the correct approach in this workflow.

B. Blender cannot render alpha channels directly, so compositing against black and using a blend mode is the standard way to get a transparent logo into an edit.

C. A blend mode derives transparency from brightness, so dark parts of the logo go semi-transparent and edge artefacts composite in as haze.

D. PNG sequences render faster than MP4 in Blender, so the image sequence route saves time as well as producing a better result in the final composite.

The answer and why: p. 478

Lesson 54 · 8 min

Subtitle files, as files

THE ONE THING TO KEEP

A subtitle file is plain text with an index, two timecodes and up to two lines, which means find-and-replace fixes a recurring error in one action — and a constant sync offset can be shifted while a growing one means the video's frame rate needs conforming.

What is actually in an SRT

A subtitle file is plain text. Once you know the format you can write, edit, fix and generate them with any tool, which is far faster than clicking in a video editor.

```
1
00:00:02,150 --> 00:00:05,400
The first offset press arrived here
in 1974.

2
00:00:05,800 --> 00:00:08,900
Nobody in the district had seen one.
```

Four parts per cue: an index number, a start and end timecode separated by `-->`, one or two lines of text, and a blank line. Timecodes are `HH:MM:SS,mmm` with a **comma** before the milliseconds.

WebVTT is nearly the same with three differences: the file begins with the line `WEBVTT`, timecodes use a **full stop** before milliseconds, and it supports styling and positioning. VTT is what browsers want; SRT is what most platforms accept.

Editing them as text

Because it is plain text, you can fix a whole file with find and replace. A recurring misheard proper noun — every instance of "Nashik" transcribed as "Nasik" — is one replacement rather than forty clicks.

For timing, **Subtitle Edit** is free, open source, and considerably better at this than any video editor. It shows the waveform, lets you drag cue boundaries against it, checks reading speed and line length, and flags overlaps and gaps. If you work with subtitles regularly, it is the tool to learn.

Aegisub is the other free option, more oriented toward styled subtitles.

Converting and shifting

FFmpeg converts between formats:

```
ffmpeg -i captions.srt captions.vtt
```

And shifts the whole file when the sync is off by a constant amount:

```
# delay every cue by 1.2 seconds
ffmpeg -itsoffset 1.2 -i captions.srt -c copy shifted.srt
```

A constant offset usually means the transcription was made from a file with a different start point. A *growing* offset means the frame-rate problem from the first block, and shifting will not fix it — conform the video first.

Burning in

```
ffmpeg -i film.mp4 -vf "subtitles=captions.srt:force_style=\
'FontName=Arial,FontSize=28,PrimaryColour=&H00FFFFFF,\
OutlineColour=&H80000000,BorderStyle=3,Outline=2,MarginV=90'" \
-c:a copy burned.mp4
```

`BorderStyle=3` gives a box behind the text; `Outline=2` gives a stroke. `MarginV` lifts the text off the bottom, which is what you need for vertical video.

This re-encodes the video, so do it from your master rather than from an already-compressed delivery file.

Quality checks that matter

Run through these before shipping any subtitle file:

- **Reading speed** under about 17 characters per second.
- **Line length** 37–42 characters maximum, two lines maximum.
- **No overlapping cues.** Two cues occupying the same moment renders unpredictably.
- **Minimum duration** of about one second, maximum of about seven.
- **Line breaks at grammatical joins**, not mid-phrase.
- **Cues do not cross a shot change** where avoidable — a subtitle that spans a cut is harder to read.
- **Speaker changes marked** with a dash at the start of each line where two people speak in one cue.
- **Non-speech information** in brackets where it matters: `[phone rings]`, `[laughs]`. This is what makes a caption track usable for a deaf viewer rather than merely a transcript.

Subtitle Edit checks the first five automatically.

Translation

A subtitle file is the unit of translation. Send the SRT, get back an SRT with the same timings and different text.

Two things to watch. Translated text is frequently longer — German and Hindi renderings of an English line commonly run 20–30% longer — so a cue that was at the reading-speed limit in English will exceed it after translation. The translator needs permission to condense, and should be told the character limit.

And machine translation of a subtitle file works reasonably for gist and badly for anything where tone, idiom or a technical term matters. It produces fluent, confident, wrong sentences in exactly the way automatic transcription does, and it should be read by somebody who speaks the language before it ships.

Naming

Platforms pick up language from the filename. `film.en.srt`, `film.hi.srt`, `film.mr.srt`, using ISO 639-1 codes. Get this right and uploads are automatic; get it wrong and your Hindi track is labelled English.

Today

Open an SRT in a text editor and fix one recurring misspelling with find and replace. Then run it through Subtitle Edit's reading-speed check and see how many cues are too fast.

BEFORE YOU MOVE ON

A subtitle file is in sync at the start of a film and roughly two seconds late by the end. An editor applies a constant offset to fix it. Why will that not work?

A. The offset should be applied in the opposite direction, since cues that appear late need their timecodes reduced rather than increased.
B. SRT timecodes are frame-accurate rather than millisecond-accurate, so a fractional offset is rounded and the accumulated rounding produces the drift being observed.
C. The subtitle file must be regenerated from the transcript because a shifted file loses its cue indexing, which is what keeps the cues aligned to the audio.
D. The error grows over time, so it is a rate mismatch rather than a fixed delay, and a constant shift can only make one point in the film correct.

The answer and why: p. 479

Lesson 55 · 9 min

When your editor cannot render the script

THE ONE THING TO KEEP

Indic and Arabic scripts need a shaping engine to reorder and combine glyphs, and many video editors do not have one — so test with a conjunct and a reordering vowel, and use libass through FFmpeg or a sidecar file rather than the title tool.

The problem nobody warns you about

You type a Hindi or Tamil or Bengali caption into your video editor's title tool and it comes out wrong. Not a missing-font box — worse than that. The letters are present and in the wrong shape: conjuncts broken apart, vowel marks sitting after the consonant instead of before it, the matra detached and floating.

This is not a font problem and installing a different font usually does not fix it. It is a **text shaping** problem, and it affects a very large share of this readership.

What text shaping is

In Latin script, rendering text is roughly a matter of drawing one glyph per character, left to right. Complex scripts do not work that way.

In Devanagari, Bengali, Tamil, Telugu, Kannada, Malayalam, Gurmukhi, Odia, Sinhala — and in Arabic, Thai and others — the shape a character takes depends on its neighbours, and characters reorder visually. The vowel sign *i* is typed after its consonant and drawn before it. Two consonants combine into a single conjunct glyph. A consonant takes a different form when it carries a vowel sign.

Doing this correctly requires a **shaping engine** — HarfBuzz is the standard free one — which reads the font's OpenType tables and applies its substitution and positioning rules.

Some video software does not use one, or uses an old or partial implementation. In that case the characters are drawn in typed order with no substitution, which produces exactly the breakage described.

Testing whether your tool shapes correctly

A quick test string per script. Type it, and compare against the same string rendered in your web browser, which does shape correctly.

- **Devanagari**: `□□□□□□□□` — two conjuncts. If you see the halant (the small stroke) visible between letters rather than the letters joined, shaping failed.
- **Devanagari vowel reorder**: `□□` — the *i* sign must appear to the **left** of the ☒. If it is on the right, the engine is not reordering.
- **Tamil**: `□□□□`
- **Bengali**: `□□□`
- **Arabic**: `مرحبا` — letters must join. Isolated forms mean no shaping.

Test once per tool and write down the answer. It is a property of the software, not of the project.

Routes that work

1. Render the text elsewhere and import it as an image. The most reliable route and it works in every editor.

Set the text in Inkscape, GIMP, LibreOffice or a browser — all of which shape correctly — export a PNG with transparency at four times the size you need, and place it on the timeline. For a caption track this is laborious; for titles and supers it is fine.

2. Burn subtitles with FFmpeg using libass. The `subtitles` filter uses libass, which uses HarfBuzz and shapes correctly.

```
ffmpeg -i film.mp4 -vf "subtitles=hindi.srt:force_style=\
'FontName=Noto Sans
Devanagari,FontSize=30,BorderStyle=3,MarginV=100'" \
-c:a copy out.mp4
```

This is the best free route for a full caption track in a complex script, and the output is correct.

3. Render the graphic in a browser. HTML and CSS shape correctly, and a headless browser screenshot gives you a PNG per cue.

4. Ship a sidecar file and let the player shape it. An `.srt` or `.vtt` uploaded to a platform is rendered by the platform's own text engine, which handles complex scripts correctly. This is the simplest answer of all, and it is why a sidecar file is not merely an accessibility nicety for these languages — it is often the only route that renders properly.

Figure 6.3

Routes that shape a complex script correctly, most reliable first

A sidecar file, shaped by the platform
The player's own text engine does the work. Simplest route of all

FFmpeg's subtitles filter
libass uses HarfBuzz, so a whole caption track burns in correctly

Text rendered in a browser
HTML and CSS shape correctly; screenshot one transparent PNG per cue

A PNG set in Inkscape or GIMP
Works in every editor. Fine for supers, laborious for a caption track

The editor's own title tool
Test it first. If the vowel sign in □□ sits on the right, it is not shaping

Test once per tool and write the answer down: this is a property of the software, not of the project. Noto covers the scripts, is free for commercial use and has complete shaping tables.

Fonts

Use the **Noto** family. Google's Noto project covers essentially every script in Unicode, is licensed under the SIL Open Font License, is free for commercial use, and its shaping tables are complete and well tested.

`Noto Sans Devanagari`, `Noto Sans Tamil`, `Noto Sans Bengali`, `Noto Serif Devanagari`, and so on. There are also good free alternatives — Mukta, Hind, Rozha One, Baloo, Yantramanav — most on Google Fonts under the OFL.

Two practical notes. Indic scripts need more vertical space than Latin: vowel marks sit above and below the baseline, so a line height that is comfortable for English clips the matras. Increase line spacing by 20–30%. And a mixed-script caption — Hindi with an English technical term in it, which is how people actually speak — needs a font pairing that looks intentional, or a font family that covers both, which Noto Sans does.

Reading speed

The 17-characters-per-second guideline was derived for Latin script. For Devanagari and other abugidas, a character carries more information — a single syllable cluster where Latin needs two or three letters — so a straight character count understates the reading time available.

There is no settled figure. The honest practice is to test with a reader of the language rather than trusting a number, and to err toward longer cue durations.

Right-to-left

Urdu, Arabic and Hebrew add direction to shaping. A caption containing an English word inside an Urdu sentence is bidirectional text, and getting it right depends on the renderer implementing the Unicode bidirectional algorithm. Most video editors do not. Sidecar files and libass are again the reliable routes.

Today

Type `□□` into your editor's title tool. If the vowel sign is on the right of the consonant, you have just learnt something important about your software.

BEFORE YOU MOVE ON

A Hindi title typed into a video editor shows the vowel sign for 'i' to the right of its consonant instead of the left. The editor installs three different Devanagari fonts and the problem persists. What is happening?

A. The fonts lack the required OpenType substitution tables, and a font with complete Devanagari tables would render it correctly in the same software.
B. The text was typed in the wrong order, and entering the vowel sign before the consonant will place it correctly in the rendered title.
C. The editor is drawing glyphs in typed order without a shaping engine, so no font can supply the reordering that the renderer is not performing.
D. The title tool is rendering at a resolution too low for the vowel sign to be positioned accurately, and working at a larger size will resolve the placement.

The answer and why: p. 479

Lesson 56 · 8 min

Editing a screen recording so it can be read

THE ONE THING TO KEEP

Capture at the delivery resolution so interface elements stay large, punch in with an eased move rather than a cut whenever detail matters, ramp through every moment of waiting, and redact with a tracked solid box because blur on text is not reliably irreversible.

Why they are hard to watch

A tutorial recorded from a screen has a specific set of failures, and all of them have fixes.

Everything is too small. A 27-inch display captured at 4K and delivered at 1080p means a 12-pixel menu label becomes 6 pixels. On a phone it is unreadable.

Nothing tells the viewer where to look. A screen has a hundred elements and the interesting one is a checkbox in the corner.

The cursor teleports. Mouse movement is fast and the viewer cannot follow it.

The pace is the presenter's, not the learner's. You know where the menu is; they are still reading the previous step.

Capture settings

Record at the delivery resolution where you can. Set your display to 1920×1080 before recording, or record a 1080p region of a larger screen. Capturing 4K and downscaling halves the size of every interface element.

On a high-density display, this means changing the scaling so the interface is physically larger and the capture is 1080p of *enlarged* interface, not a shrunken view of a large desktop.

Record at your timeline's frame rate, constant. OBS Studio is free, cross-platform, and records constant frame rate — unlike most built-in screen recorders, which write variable frame rate and produce the drift covered in the first block.

Turn off notifications before you start, and use a clean user account or profile with no personal bookmarks, no open tabs, no account names, no email addresses on screen. This is a privacy matter as much as a tidiness one.

Record system audio separately from your voice if the software lets you, so you can balance them later.

Punch in, relentlessly

The single most valuable editing technique for screen content: scale into the relevant part of the frame.

A 4K capture delivered at 1080p can be scaled to 200% with no loss, which is exactly the headroom needed. Even at 1080p, a punch-in to 150% is usually worth the softness because unreadable text at full size is worse than slightly soft text you can read.

Do it as an animated move — scale and position keyframes, eased, over about 12 to 20 frames — so the viewer follows the movement rather than being teleported. A cut straight to a zoomed view disorients; a quick push does not.

Then pull back out when the detail is done, so the viewer re-establishes where they are.

The cursor

Most screen recorders can highlight the cursor, add a click indicator, or smooth its motion. Turn these on; they cost nothing and they solve the teleport problem.

If your recorder does not, the editing equivalents: a soft circle or arrow animated to the click point, or a brief flash on the click. Keyboard input needs a visual too — a small on-screen key display is standard in software tutorials and several free tools provide one.

Pace and silence

Cut out every moment of nothing: loading, typing a long path, a menu you scrolled past. This is what the speed ramp is for — 400% over a ten-second file copy, with the audio continuing underneath.

And leave a pause after each step completes. Two seconds of the result on screen before moving on. This feels dead while editing and is where the viewer catches up.

The common failure of screen tutorials is the opposite of most video: they are usually too *fast*, not too long, because the presenter already knows the software.

Highlighting

When the viewer must look at one specific thing:

- **A soft rectangular highlight** — the rest of the screen darkened by 30–40%, the region of interest at full brightness. Animated on and off over about 8 frames.
- **An arrow or circle**, drawn simply, appearing briefly.
- **Blur or darken the irrelevant parts**, which also handles any personal information you failed to clear.

Do not use a red circle with a bouncing animation. The movement is more interesting than the thing it is pointing at.

Redaction

If personal data, licence keys or account names appear, blur or box them out. Use a **solid box, tracked**, rather than a blur — a blur applied to text can sometimes be partially reversed, and a mosaic on a short string is not secure. A solid rectangle is.

And check every frame, not just the obvious ones. Browser autocomplete suggestions, notification popovers, window titles and recent-file lists are where this leaks.

Today

Take a screen recording and punch in to 180% on the part that matters, with an eased 15-frame move in and out. Watch it on a phone. That is the difference between a tutorial that works and one that does not.

BEFORE YOU MOVE ON

A tutorial is recorded on a 4K display and delivered at 1080p. Viewers on phones say they cannot read the menus. The editor's instinct is to export at 4K instead. Why is punching in a better answer?

A. Exporting at 4K increases the file size substantially, and the platform will re-encode it back down for most viewers anyway, so the extra resolution is wasted bandwidth.

B. 4K exports take much longer to render and encode, so the time cost outweighs a readability gain that most viewers would not notice on their devices.

C. The text occupies the same small fraction of the frame whatever the delivery resolution, so only enlarging it within the frame helps.

D. Punching in improves the compression efficiency of the file, because a magnified region has less fine detail for the encoder to spend bits describing frame by frame.

The answer and why: p. 480

Lesson 57 · 8 min

Putting numbers on screen honestly

THE ONE THING TO KEEP

A chart on video carries one comparison, revealed in time so the animation directs attention, with labels on the elements rather than in a legend — and because a viewer cannot go back and check the axis, starting a bar chart anywhere but zero is a stronger deception on video than on a page.

Why data on video is different

A chart in a document can be studied. A chart in a video is on screen for four seconds, cannot be paused by most viewers, and has to be understood while somebody is talking over it.

That constraint changes what is possible. Most charts that work on a page do not work on video, and the failure is not aesthetic — the viewer simply does not extract the number.

One number per graphic

The reliable rule. If a graphic is on screen for four seconds, it can carry one comparison.

"Sales rose" — one line, two labelled endpoints. Not twelve series, not a legend, not axis ticks at every interval. Anything the viewer has to decode is time they are not spending on the point.

If you genuinely have five things to say about the data, that is five graphics, revealed in sequence, not one graphic with five things in it.

Reveal, do not present

Animation earns its place here, for a specific reason: a chart that draws itself directs attention in time. The bar grows, so the viewer watches the bar. A static chart appears complete and the eye wanders.

Useful patterns:

- **Bars grow** from the baseline, eased, over 10–20 frames, staggered by 2–3 frames each.
- **A line draws** left to right at roughly reading speed.
- **The label appears after** its bar or point has settled, not with it.
- **The one that matters is highlighted** by colour after the rest have drawn.

All of these are keyframed scale, position and opacity — the techniques from earlier in this block. None needs a specialist tool.

Legibility

The same constraints as all video type, applied to charts:

- **Label directly on the element.** A legend forces a lookup, and a lookup takes a second you do not have. Put "2024" at the end of its line.
- **Bold weights only.** Axis labels in a light weight will not survive compression.
- **Few gridlines, or none.** Thin grey lines at 1 px are precisely what the encoder discards.
- **Round the numbers.** "₹3.2 crore" is read; "₹32,147,882" is not.
- **Do not rely on colour alone** to distinguish series. Around one in twelve men has some degree of colour vision deficiency, and a red-against-green comparison is exactly the common case. Use position, labels or pattern as well.

Honesty

The same obligations as anywhere else, with the added difficulty that a video viewer cannot go back and check.

Start bar charts at zero. A truncated axis makes a 3% difference look like a doubling. On a page a reader might notice the axis label; in four seconds of video they will not.

Do not use area or volume to represent a single quantity. Doubling a circle′s diameter quadruples its area, and the viewer reads area. If you scale an icon by value, scale it by area, not by height.

Say the denominator. ″Complaints up 40%″ means something different if the number went from 5 to 7.

Label the source and the date on screen, in small type. It takes one line and it is the difference between a claim and a citation.

Do not animate between two scales. A bar that grows while its axis also changes is telling the viewer nothing they can read.

The general principle: on video, the viewer′s only defence against a misleading chart is you not making one. The usual excuses — it was in the source, the axis was labelled — do not apply when the graphic is on screen for four seconds under a voice-over.

Building them free

In the editor: a bar chart is rectangles with animated scale. For three or four bars this is genuinely the fastest route and gives you complete control.

Inkscape: draw the chart properly, export layers as PNGs, animate the layers in the editor.

A browser: HTML and CSS, or a charting library, recorded or rendered to PNGs. The right answer when the data changes or there are many charts.

Python with matplotlib: free, and it will render a sequence of PNGs directly, which is the most reproducible route for anything data-driven. Save with a transparent background and composite in the editor:

```
plt.savefig(f"frames/chart_{i:04d}.png", transparent=True,
dpi=200)
```

Spreadsheet exports: LibreOffice Calc will produce a chart you can export as an image. Usable, and it will look like a spreadsheet chart, which is sometimes exactly right for a business context.

Today

Take a chart from a document and redraw it for four seconds of screen time: one comparison, labels on the elements, numbers rounded, source in the corner. Compare the two.

BEFORE YOU MOVE ON

A video chart scales a circular icon's diameter in proportion to a value: 100 units gives a 40 px circle, 200 units gives an 80 px circle. Why does this overstate the difference?

A. The viewer reads area rather than diameter, and doubling the diameter quadruples the area, so a doubling of the value appears as a fourfold increase.

B. Circles are harder to compare than bars in general, so any circular representation of a quantity will be read less accurately than a linear one would be.

C. The larger circle occupies more of the frame, so it holds attention longer and is remembered as more significant than the smaller one.

D. Pixel dimensions do not scale linearly with perceived size on different screen sizes, so the comparison breaks down on a phone.

The answer and why: p. 480

CASE STUDY · MODULE 6

Marathi captions that the editor could not render

An illustrative case, built from documented patterns.

A two-person studio in Pune delivering a ninety-second public-health short for a municipal corporation, to run on the corporation's social accounts. Delivery tomorrow morning.

The brief is specific and reasonable. The film is about a free eye-screening camp, it will be watched almost entirely on phones in a muted feed, and it must carry Marathi captions, because the audience it is for reads Marathi and a good part of it does not read English.

The cut is finished. The mix is finished. The Marathi translation came back from the corporation's own office, checked, in a document. Sneha pastes the first line into her editor's title tool and it comes out wrong.

Not missing-glyph boxes, which would have been obvious. The letters are all there and they are in the wrong shapes. The conjuncts have broken apart, with the halant visible as a separate stroke between letters. In two words the vowel sign that should be drawn to the left of its consonant is sitting on the right. It reads, to somebody who reads Marathi, as nonsense wearing the right letters.

She tries three fonts, including a Noto face, and gets the same result each time. That is the tell: it is not a font problem. The editor's text engine is not shaping. Complex scripts need a shaping engine that reads the font's OpenType tables and applies its substitution and reordering rules, and this tool either does not use one or uses an old one. She confirms it with a two-character test: `□□` renders with the vowel sign on the right.

It is four in the afternoon. The options are these.

Ship it in English. The English captions render perfectly and the film goes out on time. It also fails the brief, and it fails the audience the campaign exists for.

Burn the Marathi captions in with FFmpeg. The `subtitles` filter uses libass, which uses HarfBuzz, which shapes correctly. It is one command with a `force_style` string, and it re-encodes the video, so it has to run from the master rather than from the delivery file. Sneha has never used FFmpeg. Her partner Vikram has, twice. Neither of them has done it under a deadline, and if the command produces something subtly wrong at eleven at night, neither of them reads Marathi well enough to catch it.

Set every caption as an image. Type the Marathi in Inkscape, which shapes correctly, export each cue as a transparent PNG at four times the size, and place them on the timeline. This definitely works and it is entirely mechanical. The film has thirty-one cues. Sneha estimates four minutes a cue with positioning, so about two hours, plus the risk of a typo in any one of thirty-one separate pieces of artwork.

Sidecar only. Upload the video with a `.srt` and let the platform's own text engine shape it. This renders correctly, costs nothing, takes ten minutes, and produces captions that do not exist in a muted autoplay preview — which is precisely where this film lives.

The fourth option is the one that looks cheapest and is the one that most directly defeats the purpose of the job. The first is the one the deadline wants. The decision is really about who the film is for, and there is a fifth thing nobody has thought of yet, which is that they could ask.

YOUR ANSWERS

1. Why did changing the font, including to Noto, not fix the rendering?

2. The sidecar file renders correctly and takes ten minutes. Why was it not the answer on its own?

3. What did asking for an extra few hours actually buy, beyond time?

STOP HERE

Write your answers before you turn the page. How it played out starts on p. 310.

CASE STUDY · MODULE 6

How it played out

Vikram rang the corporation's communications officer at half past four and asked for the morning rather than tomorrow. She gave them until eleven. They then did the second option, and gave themselves the evening for it: Vikram burnt the captions with FFmpeg and libass using Noto Sans Devanagari, and increased the line spacing by twenty-five per cent because the matras were clipping. Sneha sent a phone screenshot of six representative cues, including the conjunct and the reordering vowel, to the officer, who read them and corrected one word. They also uploaded the sidecar `.srt`, so the file exists for search, for translation and for assistive technology. The studio now runs the two-character test on every tool it uses and keeps a note of the answer, and its quotes for any job in an Indic script carry a line about caption rendering.

The questions, discussed

1. Why did changing the font, including to Noto, not fix the rendering?

Because the failure was in shaping rather than in the glyphs. The font contained everything needed; what was missing was an engine to read its OpenType substitution and positioning tables and apply them, so conjuncts were not formed and the vowel sign was not reordered to the left of its consonant. A font supplies shapes and rules. Without a shaping engine such as HarfBuzz, the rules are never applied, whichever font is installed.

2. The sidecar file renders correctly and takes ten minutes. Why was it not the answer on its own?

Because the film's whole audience watches in a muted autoplay feed, and a sidecar track is not shown there unless the platform chooses to display it. The captions were the primary channel for this piece, not an accessibility addition at the end. The sidecar was still uploaded, because it is what makes the film searchable, translatable and usable with assistive technology, but it could not be the only caption.

3. What did asking for an extra few hours actually buy, beyond time?

It bought a proofreading pass by somebody who reads the language. The route they chose renders correctly, but neither of them could have verified that the output was right, and an incorrectly shaped caption looks confident and says something else. Six screenshots sent to a Marathi reader caught a wrong word and confirmed the shaping. The deadline was the constraint that had been pushing them toward the option that failed the brief.

MODULE 6 TEST

Check what stuck

6 questions, one right answer each. This one is for you, not for a record. The answers, and the reason behind each, are in the key on p. 475. If two go wrong, re-read the module before the exam.

1 Why is an unread automatic caption track more dangerous than one full of obvious gibberish?

A. It is indexed by search engines, so the errors travel further
B. It produces fluent, grammatical sentences that say something else
C. Platforms rank videos by caption accuracy, and errors reduce reach
D. Screen readers read automatic captions aloud without any punctuation at all

2 Thin type looks elegant in the editor and breaks up after upload. What is the mechanism?

A. Platforms substitute their own font when the file's font is not embedded
B. Sub-pixel antialiasing is lost when the file is converted to 4:2:0 chroma
C. The editor renders text at 4K and the upload is 1080p, so the strokes halve
D. The encoder discards fine detail over moving video first

3 Captions near the bottom of a vertical video are legible in the editing canvas and unreadable on the platform. Why?

A. Players letterbox 9:16 video, which pushes the captions off the screen
B. The platform crops the bottom of the frame to fit its player
C. The username, description and buttons sit over the bottom fifth
D. The encoder allocates fewer bits to the lower part of the frame

4 A scale from 100 to 110 per cent with two linear keyframes reads as mechanical. Why?

A. Nothing physical changes value at a constant rate
B. Ten per cent is too small a move for the eye to read as real motion
C. Linear keyframes are rendered at the timeline's frame rate rather than the clip's
D. The interpolation is computed in pixels rather than as a percentage

5 A subtitle track drifts further out of sync as the film goes on, and shifting the whole file does not help. Why not?

A. The cues overlap, and overlapping cues render unpredictably
B. The file is WebVTT and the player expects SRT, so the timings are reinterpreted
C. SRT timecodes use a comma before the milliseconds, which some players misread
D. A growing error is a frame-rate problem, not a constant offset

6 A Hindi caption typed into an editor's title tool shows the vowel sign to the right of its consonant. What does that tell you?

A. The font is missing the glyph and a fallback font has been substituted
B. The tool is not applying the font's reordering rules
C. The text was pasted as right-to-left, so the order was reversed
D. The caption file was saved in a legacy encoding rather than in Unicode

7

MODULE 7

Effects, compositing and repair

Choose between stabilisation, retiming, masking, keying, tracking, removal and denoise for a given repair, and state for each one the artefact it produces and the property of the source footage that determines whether it will work.

Lesson 58 · 8 min

Stabilisation, and the crop it costs

THE ONE THING TO KEEP

Stabilisation cancels frame movement and scales up to hide the empty edges, so it always costs 5 to 15 per cent of the frame, cannot remove motion blur already in each frame, and needs a separate rolling-shutter pass or it makes skew more visible rather than less.

What it does, mechanically

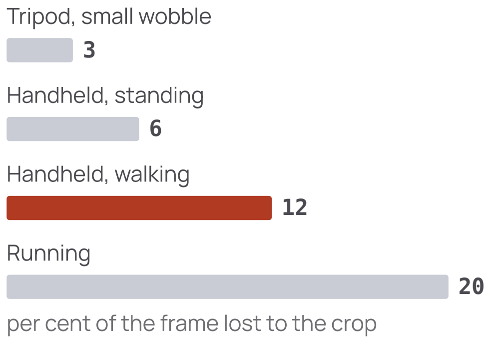
Stabilisation analyses the movement between frames, works out how much of that movement is unwanted shake, and then **moves each frame in the opposite direction** to cancel it.

Moving a frame leaves empty space at the edges. To avoid showing that space, the stabiliser scales the image up and crops. That crop is the price, and there is no version of this that does not pay it.

Typical cost: **5 to 15%** of the frame for handheld walking footage, more for anything violent. This is another reason to shoot larger than you deliver — a 4K source at 1080p can lose 15% and still be at full delivery resolution.

Figure 7.1

What stabilisation costs in frame, by how much shake there was



Typical ranges rather than a specification. A 4K source delivered at 1080p can give up fifteen per cent and still be at full delivery resolution, which is the honest case for shooting larger.

The two kinds

Translation-only. The frame slides horizontally and vertically. Cheap, fast, and sufficient for a locked-off shot with small wobble.

Full transform. Position, rotation, scale and perspective, often with the frame warped non-uniformly to correct rolling-shutter skew. Better results and a larger crop.

Resolve offers three modes on the colour page, and they behave differently enough to be worth knowing:

- **Perspective** — the full model. Best on footage with a lot of parallax and depth.
- **Similarity** — translation, rotation and zoom, no perspective. More stable on footage where the perspective analysis gets confused.
- **Translation** — position only. Use when the others produce a wobbling, swimming result.

If one mode produces swimming, try the next one down. This is the single most useful troubleshooting step and most people never discover it.

Smooth against locked

A second decision: do you want the camera to hold still, or to move smoothly?

Camera Lock removes all movement, producing a shot that looks tripod-mounted. It works on short clips and fails on long ones, because any real drift in the camera eventually exhausts the available crop.

Smooth keeps the intended movement and removes only the high-frequency shake. This is what you want for a walking shot, a pan, or anything where the camera was meant to move. Resolve exposes a Smooth parameter; higher values remove more of the low-frequency motion and cost more crop.

Rolling shutter, and the wobble that is not shake

Most sensors are read out line by line rather than all at once. During a fast pan or a hard vibration, the top of the frame is captured a few milliseconds before the bottom, so vertical lines lean and the image appears to wobble like jelly.

Stabilisation alone makes this **worse** in some cases: cancelling the frame's overall movement leaves the internal skew, now moving against a stationary background, which is more visible than it was.

The fix is a rolling-shutter correction pass, applied before or alongside stabilisation. Resolve has one in the stabilisation controls; the free `ffmpeg` route is `vidstab` plus deshake, which does not model rolling shutter and will leave the skew.

Extreme cases — a propeller, a guitar string, a whip pan — cannot be corrected and stay bent.

When it will not work

Motion blur is baked in. Stabilisation repositions frames; it cannot remove the blur that a shaking camera put into each one. A stabilised shot from a very shaky source is a steady sequence of blurred frames, which looks strange — steady and soft at once.

Nothing trackable in frame. A featureless wall, heavy fog, a night shot that is mostly black. The analysis has no points to follow.

Large moving objects dominating the frame. A car passing close to camera can be tracked instead of the background, and the stabiliser will hold the car still and move the world.

Cuts inside the clip. Stabilise each shot separately. A single analysis across a cut produces nonsense.

Free tools

Resolve's stabiliser is on the colour page in the free version, with all three modes and rolling-shutter correction. It is good.

FFmpeg's `vidstab` is a two-pass filter and genuinely useful for batch work:

```
ffmpeg -i shaky.mp4 -vf vidstabdetect=shakiness=6:accuracy=15 -f
null -
ffmpeg -i shaky.mp4 -vf
vidstabtransform=smoothing=20:zoom=2,unsharp=5:5:0.8 \
       -c:a copy stable.mp4
```

Kdenlive exposes `vidstab` through its interface, which is the easiest free graphical route outside Resolve.

The editorial alternative

Before stabilising, ask whether the shake is a problem. A handheld shot that moves with a person reads as immediacy; the same shot stabilised into unnatural steadiness can read as uncanny. Documentary footage in particular is often better left alone.

And the cheapest fix is sometimes to use the shot for less time. Two seconds of shake reads as energy; twelve seconds reads as a fault.

Today

Stabilise a handheld clip in all three modes and compare. Then look at how much the frame was cropped in each. That number is what the technique costs you.

BEFORE YOU MOVE ON

An editor stabilises a shot taken while walking quickly at dusk. The camera now holds steady, but the result looks wrong — steady and yet oddly soft and dreamlike. What happened?

A. The crop required for stabilisation upscaled the image, and the interpolation has softened the whole frame.
B. Each frame still carries the motion blur the shaking camera put into it, and steady motion with blurred frames reads as unnatural.
C. The stabiliser used Perspective mode and introduced a swimming warp across the frame, which is being perceived as softness rather than as distortion.
D. Low light forced a slower shutter and higher gain, so the source was soft to begin with and stabilisation has simply made that softness easier to notice by removing the distraction of movement.

The answer and why: p. 482

Lesson 59 · 9 min

Retiming, and the two ways of inventing frames

THE ONE THING TO KEEP

Slowing a clip means inventing frames, and the three methods trade differently — duplication stutters but invents nothing, blending ghosts, and optical flow is seamless until its motion estimate is wrong and the image melts at the region it got wrong.

The arithmetic

A clip at 50% speed has to fill twice as many timeline frames as it has source frames. A clip at 200% has to discard half of them. Discarding is easy. Filling is where every artefact comes from.

Three methods, in increasing order of quality and of failure severity.

Nearest frame

Each missing frame is a duplicate of the nearest real one. At 50% every frame plays twice.

The result stutters — it is literally a lower frame rate — but nothing is invented, so nothing is wrong. On material with little movement this is often the cleanest option, and it is what editors reach for when the alternatives look worse.

Frame blending

Each missing frame is a cross-fade between the two real frames either side.

Smoother than duplication, and it produces a visible ghost: two positions of a moving object, each at half opacity. On small movements this reads as motion blur and is acceptable. On a moving hand or a walking person it reads as a double exposure.

Optical flow

The software estimates how every pixel moved between two frames — a motion vector field — and warps the image along those vectors to construct a genuinely new intermediate frame.

When it works, the result is indistinguishable from footage shot at the higher frame rate. It is remarkable.

When it fails, it fails in a specific and recognisable way: the estimate is wrong somewhere, and the image **tears, smears or melts** in that region. Faces distort, fingers stretch, edges wobble.

It fails predictably when:

Figure 7.2

Choosing a retime method from what is in the shot

	Optical flow	Frame blending
Plenty of texture, simple motion	**Seamless** Indistinguishable from footage shot at the higher rate	**Second best** A visible ghost on anything that moves
Fast motion, crossing objects or a plain surface	**Fails visibly** The image tears and melts where the motion estimate was wrong	**Acceptable** Ghosting, but nothing is invented

When neither is acceptable, nearest frame stutters and invents nothing. When that will not do either, the shot cannot be slowed, and that is worth telling a client early.

- **Something enters or leaves frame**, so there is no correspondence to find.
- **Objects cross each other**, and the algorithm cannot tell which is in front.
- **Motion is faster than the search window**, typically a fast pan or a whip.
- **There is little texture**, so there is nothing to match — a plain wall, a sky, fog.
- **The footage is noisy**, so the noise itself generates spurious vectors.
- **Repeating patterns** — a striped shirt, a railing, a brick wall — give the matcher many equally good answers and it picks wrong.

The practical procedure

1. Try optical flow first.
2. Play the shot at full speed and look specifically at hands, faces and edges.
3. If a region melts, try frame blending on that shot.
4. If blending ghosts unacceptably, use nearest frame and accept the stutter.
5. If none is acceptable, the shot cannot be slowed and you need a different shot.

Step 5 is a real outcome and it is worth saying to a client early.

In Resolve, retime processing is set per clip in the Inspector, so you can use optical flow on most shots and drop one problem shot back to nearest frame. There is also a Speed Warp mode, which uses a machine-learning estimator and is generally better — and is a Studio feature on most versions, so check yours.

Shoot the frames instead

The answer that always works. Film at 50 or 60 fps and place the clip on a 25 or 30 fps timeline at 50%. Every frame is real and there is nothing to invent.

This costs nothing but light — a higher frame rate needs a faster shutter and therefore more exposure — and it is why any camera's high-frame-rate mode is worth using for anything you might slow down.

If you shoot at 120 fps, you get clean quarter speed on a 30 fps timeline.

Speed ramps

A change of speed within a shot. Slow into a moment, fast out of it; or a fast whip that decelerates into a reveal.

The mechanics are in the earlier keyframe lesson: use the retime **frame** graph and ease the keyframes, so the rate of change itself is gradual rather than stepped.

Two things that make a ramp work:

- **Put the speed change on a movement**, not on a still moment. The viewer's eye is busy and the transition hides.
- **Add motion blur** if your tool has it. Real fast motion is blurred; a sped-up shot made of sharp frames looks like a fast slideshow.

Audio during a ramp

You cannot ramp speech and keep it intelligible — the pitch shifts and the words smear. The standard handling: unlink the audio, let it play at normal speed over the ramped picture, or drop the sync audio entirely and carry the moment on music or ambience.

Some tools offer pitch-corrected time stretching, which works for modest speed changes on music and poorly on speech beyond about ±10%.

Free tools

Resolve's optical flow retime is in the free version. Kdenlive and Shotcut offer speed changes with frame blending. FFmpeg's `minterpolate` filter does motion-compensated interpolation and is slow but free:

```
ffmpeg -i in.mp4 -vf
"minterpolate=fps=60:mi_mode=mci:mc_mode=aobmc" out.mp4
```

Today

Take a clip of somebody moving, slow it to 40% with optical flow, and look closely at the hands. Then try frame blending. You will see both failure modes in one minute.

BEFORE YOU MOVE ON

An editor slows a shot of a person walking past a striped fence to 40% using optical flow. The person looks fine; the fence smears and wobbles. Why does the fence fail where the person does not?

A. The fence is in the background and therefore moving faster relative to the camera than the person is, which exceeds the algorithm's search window.

B. A repeating pattern gives the motion estimator many equally good matches between frames, so it picks the wrong correspondence.

C. Vertical structures are harder to track than organic shapes because the estimator works on horizontal displacement, which is why fences and railings are the standard failure case.

D. The fence is in sharper focus than the person, so it carries more high-frequency detail and the interpolation has more information to get wrong in the process.

The answer and why: p. 483

Lesson 60 · 8 min

Masks, and the honest cost of rotoscoping

THE ONE THING TO KEEP

A mask is a shape with feather, expand and invert, and animating one by hand costs roughly one to four hours per five seconds of footage — so keyframe sparsely, split the subject into simple parts, and look for a solution that is not rotoscoping first.

A mask is a shape that says where

Draw a shape; an effect applies inside it, or outside it. That is the whole idea, and it underlies windows in grading, blurring a face, replacing a sky, and every composite in this block.

Three properties matter on every mask:

- **Feather** — how soft the edge is. A hard edge always reads as cut out. Almost every mask wants some.
- **Expand or contract** — moving the edge in or out by a few pixels, which is how you remove a fringe of background that came with your selection.
- **Invert** — apply outside instead of inside.

Static, tracked, animated

A **static** mask sits still. Fine for a locked-off shot: a vignette, darkening a blown window, blurring a fixed sign.

A **tracked** mask follows something the tracker can lock onto. Fine for a rigid object moving through frame.

An **animated** mask has its shape keyframed over time, because the thing it covers changes shape as well as position. This is rotoscoping, and it is expensive.

Rotoscoping, honestly

Rotoscoping is drawing a mask around a moving subject, frame by frame, for the length of a shot.

A realistic rate for a person walking, done by hand with a bezier shape and keyframes every few frames: **somewhere between five and thirty seconds of work per frame of footage**, depending on the complexity of the edge. A five-second shot at 25 fps is 125 frames. That is one to four hours for five seconds.

This number is worth knowing because it changes decisions. Rotoscoping is not a thing you casually do to fix a shot; it is a budgeted task, and in professional work it is frequently outsourced to specialists.

What reduces the cost:

- **Keyframe sparsely.** Set a shape every 5 to 10 frames and let interpolation fill between, then fix where it drifts. Do not draw every frame.
- **Split the subject into parts.** A separate shape for the head, the torso, each arm. Each moves more simply than the whole, so each needs fewer keyframes.
- **Track what you can.** Attach a shape to a planar track for any part of the subject that moves rigidly.
- **Use the shot for less time.** Two seconds needs half the work of four.

And the thing that reduces it most: **do not rotoscope if a different solution exists.** A cutaway, a different take, a reframe, or simply not doing the effect.

Hair, motion blur and transparency

The three things a drawn mask cannot represent.

Hair is thousands of fine strands; a bezier shape approximates it as a smooth outline and the result reads as a helmet. The professional answer involves a separate hair matte and considerable effort.

Motion blur means the subject's edge is genuinely semi-transparent across several pixels. A hard shape has no way to express that, so a rotoscoped moving limb looks cut out.

Transparency — glass, smoke, water, a net curtain — is not a shape at all.

When the shot contains any of these, either the effect changes or the shot does.

Machine-assisted rotoscoping

Resolve has a **Magic Mask** that segments a person, or a part of a person, using a machine-learning model: paint a stroke on the subject and it produces a matte and tracks it. Similar tools exist elsewhere, and Segment Anything-derived models power several free implementations.

This is genuinely useful and it changes the economics for a certain class of shot. Two honest caveats. It is good at people and body parts and poor at arbitrary objects. And the matte it produces is soft and imprecise at exactly the edges that matter — hair and motion blur — so it gets you 80% of the way and the last 20% is still hand work.

On Resolve, some Magic Mask features are Studio-only depending on version; check yours before planning around it.

The common jobs

Blurring a face or a licence plate. A tracked shape with a mosaic or blur. Use a solid shape rather than a light blur if the intent is genuine anonymity, and check every frame — the track will slip.

Sky replacement. A luminance key usually beats a drawn mask, because the sky is bright and uniform. Problems are at the horizon and in anything thin against the sky — branches, wires, hair.

Removing something at the edge of frame. Often better solved by a reframe than by a mask.

A split screen. Two rectangular masks and no difficulty.

Today

Draw a mask around a moving object for one second of footage with keyframes every five frames. Time yourself. That number is the one to quote from.

BEFORE YOU MOVE ON

An editor rotoscopes a person's arm as it swings quickly across frame. The mask follows accurately, and the result still reads as cut out. What is the mask unable to represent?

A. The arm's depth relative to the background, which requires a depth matte rather than a two-dimensional shape to composite convincingly.
B. The feathering needed at the edge, which has to vary along the outline rather than being a single uniform value across the whole shape.
C. The motion blur at the fast-moving edge, which is genuinely semi-transparent across several pixels rather than being a hard boundary.
D. The colour spill from the background onto the arm, which persists after masking and gives away the composite to the viewer.

The answer and why: p. 483

Lesson 61 · 9 min

Keying footage that was lit badly

THE ONE THING TO KEEP

A keyer builds its matte from colour, so a garbage matte, an evening-out pass and a denoise before keying do more than any setting on the keyer itself — and a subject in 4:2:0 with green spill in their hair has a hard limit no tool removes.

The position you are usually in

You did not light the screen. Somebody filmed against a green sheet with one lamp, on a phone, and the sheet has creases.

Motion, Sound and Music makes the case that a key is decided on set. This lesson is the other half: what to do when it was not, and how to know when to stop.

What a keyer is doing

A chroma keyer builds a **matte** from colour: pixels close to the key colour become transparent, pixels far from it stay opaque, and pixels in between become semi-transparent.

Three consequences follow directly.

It needs colour resolution. At 4:2:0 the chroma channels are a quarter the resolution of the picture, so the matte is built from a quarter of the information. Edges come back stepped and unstable.

It needs separation. If the screen is unevenly lit — bright in the middle, dark at the edges, which is what one lamp produces — the "green" in the corner is a different colour from the green in the centre, and no single tolerance covers both.

It cannot distinguish spill from screen. Green light bouncing off the screen onto the subject's hair and shoulders is genuinely green, so the keyer takes some of the subject with it.

The repair workflow

In order, and each step should be small.

1. Garbage matte first. Draw a rough mask that excludes everything you do not need — the edges of the sheet, the light stands, the floor. Track it if the subject moves. This removes most of the unevenness before the keyer sees it, and it is the step people skip.

2. Even out the screen before keying. A qualifier or a soft window that lifts the dark corner of the screen toward the centre's brightness. You are not trying to make it look good, only to make it uniform enough for one tolerance.

3. Pull a core matte and an edge matte separately. Set the keyer aggressively to get a solid, clean interior — the core — and separately gently to preserve the semi-transparent edge. Combine them. Most keyers expose this as a clean plate, black point and white point on the matte.

4. Denoise before keying, not after. Noise in the chroma channels generates a crawling, unstable matte. Even a light temporal denoise on the source can transform the result.

5. Despill. Every keyer has a spill suppression control. It works by reducing green where green exceeds the other channels, and overdone it turns skin magenta and hair grey. Small amounts.

6. Match the composite. The subject and the background need the same black point, the same colour temperature, the same grain, and ideally a shared light direction. Most unconvincing composites fail here rather than at the matte — the edges are fine and the subject obviously does not belong.

Luma keys and difference mattes

When there is no green screen at all:

A **luma key** pulls a matte from brightness. It works for a subject against a bright sky or a dark studio, and only when the subject's own brightness does not overlap the background's.

A **difference matte** compares the shot against a clean plate — the same locked-off frame with nothing in it. Anything that differs is the subject. It works well for a locked-off camera and fails on any camera movement, changing light or noise. If you are also shooting, filming five seconds of the empty set at the end of every setup costs nothing and enables this.

When to stop

The honest thresholds:

- If the matte's edges are stepped and blocky on 4:2:0 footage and heavy blurring does not hide it, the shot will not key cleanly. Consider a soft-edged composite where the subject is deliberately blurred at the edge, a smaller subject in frame, or a different approach entirely.
- If the subject is wearing green, or has green in their hair or glasses, those parts will always go transparent. A separate hold-out mask is the only fix, and that is rotoscoping.
- If the screen is out of focus in a way that makes the edge ambiguous, the matte will be ambiguous.

In each of these, the correct answer is often to tell whoever asked that this shot cannot be keyed to the standard they want, and to propose the alternative — a different background approach, a reshoot of that one setup, or dropping the effect.

Free tools

Resolve's 3D Keyer on the colour page is in the free version and is genuinely good. Fusion's Delta Keyer is better still and is also free. Both expose the core/edge separation described above.

Kdenlive and Shotcut have chroma key filters that will handle a well-lit screen and will struggle with a badly lit one. OBS Studio has a live chroma key, useful when you are recording rather than editing.

Today

Find any green-screen clip, pull a key, and then compare the matte on its own against the matte after a garbage mask and a light denoise. The difference is the argument for doing the steps in order.

BEFORE YOU MOVE ON

A key on evenly lit but noisy footage produces a matte whose edges crawl and flicker from frame to frame. Which step addresses the cause rather than the symptom?

A. Blurring the matte more heavily, which averages out the frame-to-frame variation and produces a stable edge.
B. Applying a temporal denoise to the source before the keyer sees it, so the chroma channels stop generating different values each frame.
C. Tightening the keyer's tolerance so that only pixels very close to the key colour are removed, which excludes the noisy values at the boundary.
D. Adding a garbage matte around the subject, which excludes the areas of the screen where the noise is most severe from the key entirely.

The answer and why: p. 484

Lesson 62 · 9 min

Tracking a point, a plane and a camera

THE ONE THING TO KEEP

Point tracking gives position, planar tracking gives a whole surface and survives occlusion, and camera tracking needs real parallax and a rigid scene — so pick the cheapest one that solves the problem, and expect the composite to fail on matching rather than on the track.

Three different problems

Point tracking follows one small feature through a shot and gives you its position over time. Use it to attach a label to a moving object, or to drive a mask.

Planar tracking follows a flat surface and gives you its position, rotation, scale and perspective. Use it to replace what is on a screen, a sign or a wall.

Camera tracking — also called match moving — solves for the three-dimensional path the camera took through the scene, so you can place an object into the space and have it stay put. Use it when something has to sit in the world rather than on a surface.

They are progressively more expensive and progressively more capable. Choose the cheapest one that solves your problem.

Point tracking, and what makes a good feature

A tracker searches a small region of the next frame for the pattern it found in the current one. It works well when the pattern is distinctive and fails when it is not.

Good features: a corner, a high-contrast junction, a small dark mark on a light surface, the corner of a window.

Bad features: a point along a straight edge (it can slide along the edge freely — this is the aperture problem, and it is why edges track badly while corners track well), a repeating pattern, a soft gradient, anything out of focus, anything with motion blur.

When a track fails mid-shot:

- Track backwards from a later frame.
- Split the shot and track each part.
- Move the tracker to a different feature that is rigidly attached to the same object, and offset the result.
- Keyframe by hand through the difficult section.

Planar tracking

The technique for screen replacement, sign replacement, and any mask attached to a flat surface. It tracks the whole surface rather than individual points, so it survives partial occlusion — a hand passing over a phone screen does not break the track.

Mocha is the paid standard. Fusion's Planar Tracker, inside free Resolve, does the same job. Blender's motion tracker also handles planar cases.

The workflow for replacing what is on a screen:

1. Track the plane of the screen across the shot.
2. Attach a corner-pin transform to the track.
3. Put your replacement image through the corner pin.
4. Mask out anything that passes in front — a hand, a head. This part is rotoscoping and it is where the time goes.
5. Match the composite: add the screen's reflections, its brightness relative to the room, a little blur if the shot is not sharp, and grain.

Step 5 is what separates a convincing replacement from an obvious paste. A real screen has glare and is rarely perfectly sharp.

Camera tracking

Solving the 3D camera path requires parallax — the camera must actually move through space, not merely pan. A pure pan from a tripod has no parallax and cannot be solved for position, only rotation.

It also needs the scene to be rigid. Trees in wind, water and crowds all contribute points that move independently, and they degrade the solve.

Blender's motion tracker is free and genuinely capable. The workflow: detect features, solve the camera, check the solve error — under about 1 pixel is good, over 3 is unusable — then place 3D objects in the solved scene and render with a transparent background to composite back.

Fusion has a Camera Tracker as well, in free Resolve.

Budget for this: a straightforward solve on a well-shot moving shot is an hour. A difficult one can consume a day and still fail.

Helping the track at the shoot

If you also film, three things make tracking dramatically easier and cost nothing:

- **Keep the shutter fast enough to limit motion blur** on tracking shots. Blurred features do not track.
- **Put tracking markers in the scene** — small pieces of tape at known positions on a blank wall or a green screen. They give the tracker features where there are none, and they are removed later with the techniques in the next lesson.
- **Move the camera.** For a 3D solve, a slow lateral move gives far more parallax than a pan from the same spot.

Knowing which one you need

- Label follows an object in 2D → **point track**.
- Replace what is on a flat surface → **planar track**.
- Put an object into the room so it stays where you put it → **camera track**.

- Mask follows a rigid object → **point or planar track**.
- Mask follows a person who changes shape → **not a track**; that is rotoscoping.

Today

Track a single point in a moving shot and attach a text label to it. Watch where the track slips. That is what a tracker finds hard, learnt in five minutes.

BEFORE YOU MOVE ON

An editor needs to place a 3D object on a table so it stays put as the camera moves. The shot is a pan from a tripod. Why will a camera track not solve it?

A. The scene is not rigid enough, since a tripod pan reveals more of the room and brings new objects into frame as it goes.

B. Tripod pans introduce rolling shutter skew, which corrupts the feature positions the solver depends on for an accurate result.

C. A pan has no parallax, so there is no positional information for the solver to recover the camera's path in space from.

D. The tracker needs at least eight well-distributed features and a pan across a plain room wall will not supply enough of them for a reliable solve.

The answer and why: p. 484

Lesson 63 · 8 min

Taking something out of a moving shot

THE ONE THING TO KEEP

Removal means covering something with pixels that plausibly belong — try reframing first, patch from the same frame on a locked-off shot and from another frame on a moving one, and always match grain, because a patch that is too clean is what gives a removal away.

The job

A microphone in frame. A light stand in the corner. A logo you do not have the rights to. A tracking marker you put there deliberately. A modern car in a period piece. A person who withdrew their consent after filming.

The mechanism in every case is the same: cover the unwanted thing with pixels that plausibly belong, in a way that holds up while the shot moves.

The order of solutions, cheapest first

1. Reframe. If the object is near an edge and your source is larger than your delivery, scale and reposition. This takes ten seconds and solves a large fraction of real cases. Always try it first.

2. Cut around it. Use less of the shot, or a different take. Also ten seconds.

3. Cover it with something. A graphic, a lower third, a split screen. Sometimes legitimate, often a fudge.

4. Patch from another part of the same frame. If the background behind the object is uniform — a wall, a sky, a road — copy a clean region over it with a soft-edged mask. Works immediately on a locked-off shot.

5. Patch from another frame. If the camera moves, the area behind the object is visible at some other point in the shot. Take that frame, track it into position, and composite it.

6. Clean plate. Film the scene without the object and composite it in. Requires planning at the shoot and is the reliable professional answer.

7. Paint it out frame by frame. The last resort, and it costs what rotoscoping costs.

The locked-off case

When the camera does not move, removal is nearly trivial: copy a clean region of the same frame over the object, soft-mask it, and it holds for the whole shot because nothing moves.

This is why locking the camera off is the single most useful thing you can do for a shot you know needs work.

The moving case

When the camera moves, a static patch slides off. The patch has to move with the shot, which means:

1. **Track** the surface the object sits on — planar track if it is flat, point track if it is simple.
2. **Attach** the patch to that track.
3. **Mask** the patch to only where it is needed, with a soft edge.
4. **Match** the grain, brightness and blur of the patch to the surrounding frame. A clean patch on grainy footage is visible precisely because it is too clean.

Step 4 is where these fail. Add grain to the patch.

Temporal median

A technique worth knowing for a specific case: something moves through an otherwise static shot and you want it gone.

If the camera is locked off and the unwanted thing moves, then for every pixel, the *median* value across many frames is the background — because the object only covers each pixel for a fraction of the time. Computing the per-pixel median over the shot gives you a clean plate with the moving object removed.

This is how "remove the tourists from my photo" features work. FFmpeg can do a rough version with `tmix`; Fusion and most compositors have a temporal median.

It requires a locked-off camera and an object that keeps moving. A person standing still is background as far as the median is concerned.

Machine inpainting

Generative models will now fill a masked region with plausible content, including across a moving shot with reasonable temporal consistency. Several free and paid implementations exist and the capability is improving quickly.

Three honest points.

It is generating content, not recovering it. The pixels behind the object were never recorded. What comes back is a plausible invention, and on anything structured — text, a face, a pattern, architecture — the invention is often visibly wrong in a way that is hard to unsee once noticed.

In factual work this matters. Removing a light stand from a documentary shot is uncontroversial. Removing a person, an object or a piece of text from a record of a real event alters the record, and the fact that the tool made it easy does not change that. Several jurisdictions are moving toward labelling requirements for synthetically altered media, and they differ and are changing.

Temporal consistency is the weak point. Frame-by-frame generation produces flicker, and the tools that address it do so by constraining the generation, which reduces quality. Check the result at speed, not frame by frame.

What is actually hard

- Removing something that crosses **in front of a moving subject**, because the patch has to respect the subject's edge, which is rotoscoping.

- Removing something **reflected** in a surface, which usually requires removing it twice.
- Removing something that **casts a shadow**, because the shadow stays.
- Anything against a **detailed, structured background** — brickwork, text, a crowd.

Today

Take a locked-off shot with something unwanted in it, copy a clean region over it with a feathered mask, and add a little grain to the patch. Five minutes, and it is most of the technique.

BEFORE YOU MOVE ON

An editor patches out a light stand on a handheld shot, tracking the patch to the wall behind it. The track is accurate and the patch is still visible. What is the most likely reason?

A. The patch is too clean — it lacks the grain and noise of the surrounding footage, so it reads as a smooth area in a textured frame.
B. The mask edge is too hard, and increasing the feather until the boundary is imperceptible will integrate the patch with the frame around it.
C. A planar track cannot hold on a handheld shot because the perspective of the wall changes continuously as the camera moves through space.
D. The patch was taken from a frame with different lighting, so its brightness does not match the frame it is composited into at that moment in the shot.

The answer and why: p. 485

Lesson 64 · 9 min

Noise reduction, sharpening, and what neither can do

THE ONE THING TO KEEP

Temporal denoise averages the same pixel across frames so noise cancels and detail survives, spatial denoise averages within a frame so detail goes with the noise — remove chroma noise aggressively and luminance gently, denoise before grading, and re-add grain at the end.

Where noise comes from

A sensor counts photons. In low light there are few photons, so the count is uncertain, and that uncertainty appears as noise. Raising ISO amplifies the signal and the uncertainty together — it does not add noise, it makes the existing shortfall visible.

Two kinds appear in the image:

Luminance noise — brightness varies pixel to pixel. Reads as grain. Broadly acceptable and sometimes pleasant.

Chroma noise — colour varies pixel to pixel. Reads as coloured speckles, usually in the shadows. Always ugly, and much easier to remove because the eye's tolerance for chroma resolution is low anyway.

The practical consequence: **remove chroma noise aggressively and luminance noise gently.** Most beginners do the reverse and end up with a waxy image.

Spatial against temporal

Spatial denoise works within one frame: it averages a pixel with its neighbours, guided by whether they look like they belong to the same region. It removes noise and it also removes fine detail, because fine detail and noise look similar to an algorithm inside a single frame. Push it and skin goes plastic, hair becomes a smear, and texture disappears.

Temporal denoise works across frames: it averages the same pixel over several frames in time. Because noise is random per frame and the real image is not, the noise averages toward zero while the detail survives. This is much better.

Its failure mode is motion. Where something moves, the same pixel is different content in each frame, so averaging produces ghosting or trails. Every temporal denoiser therefore has a motion threshold that decides when to stop averaging, and setting it too high produces smearing behind moving objects.

Figure 7.3

The two denoisers, and where each one fails

Spatial: averages within a frame

- Averages a pixel with its neighbours
- Cannot tell fine detail from noise
- Pushed hard, skin goes plastic and hair smears
- Motion is no obstacle, because it never looks across frames

Temporal: averages across frames

- Averages the same pixel over several frames
- Noise is random per frame, so it cancels
- Real detail is not random, so it survives
- Fails on movement: ghosting and trails behind objects

Temporal first and gently, then a little spatial for what is left. Remove chroma noise aggressively and luminance noise sparingly; most beginners do the reverse and end up with a waxy picture.

The working method: **temporal first, gently, then a small amount of spatial for what remains.**

The free routes

Resolve's temporal and spatial noise reduction on the colour page is **Studio-only** on current versions. This is one of the clearest gaps in the free version, and it is worth saying plainly.

The free alternatives:

```
# hqdn3d: fast, combined spatial and temporal
ffmpeg -i noisy.mov -vf hqdn3d=4:3:6:4.5 -c:v dnxhd -profile:v
dnxhr_hq out.mov

# nlmeans: much better quality, much slower
ffmpeg -i noisy.mov -vf nlmeans=s=3.0:p=7:r=15 -c:v dnxhd -
profile:v dnxhr_hq out.mov
```

The four `hqdn3d` parameters are luma spatial, chroma spatial, luma temporal, chroma temporal. Note that you can raise the chroma values well above the luma ones, which is exactly the asymmetry described above.

Neat Video is the paid specialist and is excellent; it profiles your specific camera's noise and is worth the money if you do a lot of low-light work.

Always denoise **to an intermediate codec**, not back to H.264. Denoising and re-encoding to a lossy format twice is two generations of loss.

Denoise before everything else

Order matters. Denoise first, before grading, because:

- A grade that adds contrast amplifies noise, so denoising afterwards is fighting an amplified problem.
- A keyer or a tracker fed noisy footage produces unstable results.
- Sharpening applied to noise sharpens the noise.

And re-add grain at the end, after grading, if the result looks too clean. Denoise then re-grain is a standard and slightly absurd-sounding workflow, and it works because the grain you add is consistent and controllable while the noise you removed was not.

Sharpening, and what it actually is

Sharpening does not add detail. It increases **acutance** — the local contrast at edges — by finding edges and making the light side lighter and the dark side darker.

That produces a perception of sharpness. It also produces **halos**: a visible bright line on one side of a high-contrast edge and a dark one on the other. Once you can see halos you will see them everywhere, including on television.

Unsharp mask has three controls:

- **Amount** — how strong. Keep it modest.
- **Radius** — how wide the effect extends from the edge. Small values (0.5–1.5 px at 1080p) sharpen fine detail; large values produce halos.
- **Threshold** — how much difference counts as an edge. Raising it stops the filter sharpening noise in flat areas, which is the single most useful control and the least used.

Upscaling

Bicubic and Lanczos interpolate smoothly and produce a soft, honest enlargement. Machine-learning upscalers produce a sharp, plausible enlargement with invented detail.

The honest position on the latter: for archive material, home video and anything where the goal is watchability, they are often a genuine improvement. For anything where the image is evidence or a record, they are generating content — an upscaled face is a guess about what that face looked like — and that is a meaningful difference whatever the result looks like.

What neither can fix

Missed focus. Sharpening raises edge contrast on the blur, producing a crunchy soft image.

Motion blur from a slow shutter. The information is spread across the frame.

Heavy compression artefacts. Sharpening makes blocking more visible, not less. Mild denoise sometimes helps blocking; sharpening never does.

Today

Run the `hqdn3d` command on a noisy clip with chroma values twice the luma values, and compare against the original at 200% zoom. That asymmetry is most of the technique.

BEFORE YOU MOVE ON

An editor applies heavy spatial noise reduction to a low-light interview. The noise is gone and faces look waxy, with hair reduced to a smear. Why does temporal denoise not have this problem to the same degree?

A. Temporal denoise operates only on the chroma channels, leaving luminance detail such as hair and skin texture untouched by the process.
B. Temporal denoise runs at a lower strength by default, so the same slider position removes less of the image in either case.
C. Noise is random between frames while the image is not, so averaging the same pixel over time cancels noise and leaves the detail intact.
D. Temporal denoise uses motion vectors to follow detail as it moves, which means fine structures like hair are tracked and preserved rather than averaged away with the noise around them.

The answer and why: p. 485

Lesson 65 · 8 min

Transitions you build rather than drop in

THE ONE THING TO KEEP

A hard cut is right most of the time and a dissolve means time passed — the transitions worth learning are match cuts, whip pans and masked reveals, which you build from what is in the frame rather than drag from a bin.

The default position

A hard cut is right roughly nineteen times in twenty. A cross-dissolve means time has passed or we have moved to a different mental space, and it means only that.

Everything else in the transitions bin — cubes, page peels, glitch presets, light leaks, star wipes — is a signal that a decision has not been made about what the cut is for. Using one to smooth over a cut that does not work produces a bad cut that now takes longer.

That said, there are transitions worth building, and they are built rather than dragged in.

The dissolve, properly

Two things people get wrong.

Length. The default is usually one second, which is long. For a scene change, 12 to 20 frames is often enough. A long dissolve is a statement about how much time passed; use it when that is what you mean.

Placement. A dissolve centred on the cut point consumes handles from both clips. If either clip is at the limit of its source, the editor either refuses or shortens the dissolve without telling you. The trim lesson's point about handles applies here.

And the mechanism: during a dissolve both images are visible at partial opacity, so a dissolve between two busy shots produces a moment of visual noise. Dissolve between shots that are compositionally simple, or where something in both occupies the same part of the frame.

The match cut

Cutting between two shots where something in the frame — a shape, a movement, a colour — is in the same place in both. The eye follows the continuity and the cut becomes a transformation rather than a break.

A round mirror to a round clock. A hand reaching down to a hand reaching up. A pan that ends moving left cutting to a pan that begins moving left.

This is free, it requires no plug-in, and it is the most satisfying transition in editing. What it requires is that you notice the shape while logging — which is one of the things the logging pass is for.

The whip pan

End the outgoing shot with a fast pan, start the incoming shot with a fast pan in the same direction, and cut in the middle where both are motion-blurred beyond recognition.

If you did not shoot it, you can fake it: add a fast directional blur and a rapid position move over the last 4–6 frames of the outgoing shot, and the mirror of it on the first 4–6 frames of the incoming one. It is four keyframes and a blur, and it holds up.

The masked transition

Something in the frame passes across the lens — a person, a door, a pillar — and at the moment it fills part of the screen, the shot behind it changes.

Build it by masking along the edge of the passing object and revealing the second shot behind it. The mask usually needs to follow the object, which is a track or a short rotoscope.

This is what most "seamless transition" tutorials are actually teaching, and it is genuinely effective when the piece calls for energy.

The morph cut, honestly

A feature in Premiere and, under other names, elsewhere: it attempts to hide a jump cut in a talking head by generating intermediate frames that morph one position of the face into the other.

When it works — small movement, plain background, short gap — it is invisible and very useful. When it fails, the face distorts in a way that is considerably worse than the jump cut would have been.

It is also, mechanically, generating frames of somebody's face that they did not perform. For a corporate talking head this is uncontroversial. For journalism or anything where the recording is a record, it is worth thinking about before it becomes routine.

The practical rule: use it, check it at full speed, and have the jump cut ready.

Audio transitions are not optional

A hard cut in picture with a hard cut in sound is what makes an edit sound like a slideshow. Every audio cut wants a short cross-fade — even two or three frames — to avoid the click that a discontinuity in the waveform produces.

Most editors can apply a default audio cross-fade to every cut at once. Find that setting and turn it on.

And the J and L cuts from the second block are, properly understood, the most important transitions in this course. They do more for how an edit feels than any visual effect.

The test

Before adding any transition, ask: **what does this tell the viewer that a cut would not?**

If the answer is "time passed", use a dissolve. If it is "we are somewhere else now", use a dissolve or a fade. If it is "I wanted this to look more dynamic", cut.

Today

Go through something you have made and replace every transition that is not a cut or a dissolve with a cut. Watch it. Then put back only the ones you miss.

BEFORE YOU MOVE ON

An editor applies a one-second cross-dissolve between two busy, detailed shots to soften a cut that felt abrupt. The result looks worse than the hard cut did. What is the mechanism?

A. The dissolve consumed handles from both clips, so each shot is now showing frames that were trimmed for a reason.
B. Both images are visible at partial opacity throughout, so two detailed frames overlap into a second of visual noise the viewer cannot parse.
C. A one-second dissolve is far longer than the 12 to 20 frames appropriate for a scene change, so it reads as a much larger passage of time than the material supports.
D. Cross-dissolves re-encode both clips across the transition, so compression artefacts appear at exactly the moment the viewer is looking for continuity.

The answer and why: p. 486

Lesson 66 · 8 min

Diagnosing a timeline that has stopped playing

THE ONE THING TO KEEP

Check in order — proxies on, effects bypassed, scopes closed, playback resolution reduced, background renders finished, disk fast enough — because slow playback is almost always one specific effect or one missing toggle rather than an inadequate machine.

Slow is a symptom with several causes

Playback drops to eight frames a second. The fan runs. You conclude the machine is inadequate. Usually one specific thing is responsible, and the diagnosis is quick if you know the order to check.

The order to check

1. Are you playing proxies? The first block covered this. Long-GOP source files are expensive to decode, and a proxy removes the problem entirely. Check that proxy media exists *and* that the toggle to use it is on — these are two separate settings and forgetting the second is extremely common.

2. Is a single effect responsible? Disable all effects — Resolve has a bypass colour grades and fusion effects button; most editors have an equivalent — and play. If it is fine, re-enable one at a time. Nine times in ten the culprit is one node, one effect, or one nested timeline.

The usual suspects, roughly in order of cost: noise reduction, optical flow retiming, motion blur, large blurs, stabilisation before it is cached, any Fusion composition, and a high-resolution PNG scaled down in real time.

3. Is it the scopes? Scopes recompute every frame and can cost a third of your playback performance. Close them when you are not reading them.

4. Is playback resolution set to full? Most editors let you play at half or quarter resolution while keeping full for export. Resolve has Timeline Proxy Mode in the Playback menu. This is a one-click improvement that many people never find.

5. Is something rendering in the background? Cache generation, proxy generation, an export. Check before concluding anything.

6. Is the disk the bottleneck? 4K ProRes runs at roughly 700 Mbit/s, which a slow external drive over a slow connection will not sustain, particularly with two streams for a multicam. If the machine is idle and playback still stutters, suspect the drive.

Render cache

The general solution for the effects problem: let the computer render the expensive sections in the background, and play the rendered result.

Resolve calls this Smart Cache, with modes for Smart, User and Off, plus separate cache settings for Fusion output and for colour output. Smart mode automatically caches anything it considers expensive. User mode caches only what you flag.

On a slower machine, the working method is: set cache to Smart, leave the timeline for a minute after each big change, and play. On a faster one, User mode gives you more control over disk use.

The cache goes to the location you set in project settings — point it at your working drive, not the system disk.

Nested timelines and compound clips

A compound clip or nested timeline is convenient and expensive: the editor renders the nest, then applies whatever is on it, which is two passes. A timeline with six nested compositions is doing six times the work.

Where a nest exists only for tidiness, flatten it. Where it is load-bearing, cache it.

GPU and CPU

Most modern editors do colour, effects and scaling on the GPU and decoding on either. What this means practically:

- **Resolve wants a real GPU** with 4 GB or more of video memory, and more for 4K work. On integrated graphics it runs and it is unpleasant.
- **Hardware decoding** for H.264 and H.265 is usually a checkbox in preferences. Turn it on. On some hardware and some codecs it is unreliable; if playback is glitchy rather than merely slow, try turning it off.
- **RAM** matters for cache. 16 GB is comfortable for 1080p; 32 GB for 4K with effects.

If Resolve is a fight on your machine, **Shotcut and Kdenlive are genuinely lighter** and will edit 1080p comfortably on hardware that Resolve struggles with. That is a real option rather than a consolation prize, and the earlier lessons on cutting, structure and sound apply identically in them.

The optimisation that costs nothing

Work at 1080p even when delivering 4K, and finish at 4K only at the end. The whole edit — structure, cutting, sound — is unaffected by resolution, and a 1080p timeline is roughly four times cheaper to play.

In Resolve you can change the timeline resolution at the end, though as the setup lesson warned, this re-frames every clip whose scaling was set relative to the old frame. The safer version is to work on proxies at full timeline resolution, which achieves the same thing without changing any geometry.

Today

Bypass all effects on your slowest timeline and play it. Then re-enable them one at a time until it stutters. You will find the culprit in about three minutes.

BEFORE YOU MOVE ON

An editor generates proxy media for a whole project and playback is unchanged. What is the most likely explanation before concluding the machine is too slow?

A. Proxy generation is still running in the background, so the machine is doing both jobs at once and will be fine once it finishes.

B. Generating proxies and using them are separate settings, and the toggle telling the editor to play the proxies has not been switched on.

C. The proxy resolution was set too high, so the proxies are nearly as expensive to decode as the original camera files were.

D. Proxies only improve scrubbing performance rather than continuous playback, because continuous playback reads ahead from the original media regardless of the proxy setting.

The answer and why: p. 486

CASE STUDY · MODULE 7

The man in the background who changed his mind

An illustrative case, built from documented patterns.

An independent documentary about a weekly fish market in Kochi, cut by the director herself, three weeks from a festival deadline.

The shot is eleven seconds long and it is the one that opens the film. A handheld move down the length of the market at six in the morning, the light coming in sideways off the water, boxes being carried across frame. It establishes the place in a way nothing else in the rushes does, and every version of the cut has opened with it.

Four days before picture lock, a man who appears in it for about five of those eleven seconds sends a message. He had signed the release on the day. He now works somewhere that he would rather not be associated with the market, he has thought about it, and he asks not to be in the film.

The director's position is that the legal question and the right answer are not the same question. She has a signed release. She also has somebody who has told her plainly that he does not want to be in it, and a film that will be shown to an audience in the city he lives in.

So the shot has to change. There are four ways, and each costs something different.

Cut the shot. Eleven seconds, gone, and with it the only wide that establishes the market as a place. The next best opening is a close-up of ice being poured, which is a good shot and tells the viewer nothing about where they are. Cost: the opening, and about a day of re-cutting the first ninety seconds around it.

Reframe. The source is 4K and the delivery is 1080p, so there is room to scale and reposition. The man walks through the middle third of the frame from left to right, and the camera is moving with him for part of it. A reframe that keeps him out would have to move continuously, would end up at roughly

190 per cent for two seconds, and would lose the two boxes being carried in the foreground that give the shot its depth. Cost: the shot becomes a different and weaker shot, in about two hours.

Remove him. Cover him with pixels that plausibly belong. The camera moves, so a static patch slides off: the surface behind him has to be tracked, the patch attached to the track, masked with a soft edge and matched for grain. Behind him is a row of crates, a wet floor with reflections, and two other people who cross behind him at second six. That last detail is the expensive one — a patch that has to respect the edge of a moving person is rotoscoping, at somewhere between one and four hours per five seconds of footage. Cost: realistically a day and a half, with no guarantee, four days before lock.

Blur or box him. A tracked solid box or a heavy mosaic. It is reliable and it takes an hour. It also announces itself in the opening shot of a documentary, which tells the audience that something has been hidden in the first eleven seconds of the film.

There is a fifth option she almost misses, which is that the market runs every week and the light at six in the morning is the same light.

YOUR ANSWERS

1. The director had a signed release. Why did she treat the request as binding anyway?

2. Why was removal so much more expensive here than in a locked-off shot?

3. What made the re-shoot possible, and what does that suggest about how to log a shoot?

STOP HERE

Write your answers before you turn the page. How it played out starts on p. 358.

CASE STUDY · MODULE 7

How it played out

She re-shot it. Two mornings later she filmed the same move four times in twenty minutes, told the three people in frame what it was for, and got a wider version than the original with better foreground. The re-shot opening is in the film. The removal was never attempted, because a day and a half of tracked patching over a moving person, four days before lock, was not a risk the schedule could absorb. She wrote two things into her own practice afterwards: that a release is the floor and not the ceiling, and that she would find out on the shoot day which shots were load-bearing, because the shot she could least afford to lose was the one she had filmed once.

The questions, discussed

1. The director had a signed release. Why did she treat the request as binding anyway?

Because consent to be filmed is about scope and about what a person understood they were agreeing to, and this man was telling her clearly that the use now in front of him was not the one he had in mind. The release settles the legal question; it does not settle whether showing him is defensible in a film that will screen in the city where he lives and works. Treating the release as the end of the matter would have meant using somebody's face against their stated wishes because a form allowed it.

2. Why was removal so much more expensive here than in a locked-off shot?

Because the camera moved. On a locked-off shot the background behind the object is identical in every frame, so a clean region copied over it with a soft mask holds for the whole shot. With a moving camera the patch has to be tracked onto the surface, matched for grain and brightness, and masked; and where another person crosses in front of the area being patched, the patch has to respect a changing edge, which is rotoscoping and is budgeted in hours per second.

3. What made the re-shoot possible, and what does that suggest about how to log a shoot?

The event was recurring and the conditions repeatable: the market runs weekly and the light at six is the same. The fifth option was almost missed because the problem presented itself as a compositing problem. The practical lesson is to know, at the shoot, which shots are load-bearing and which exist once — and to film the load-bearing ones more than once, since the shot she could least afford to lose was the one she had a single take of.

MODULE 7 TEST

Check what stuck

6 questions, one right answer each. This one is for you, not for a record. The answers, and the reason behind each, are in the key on p. 481. If two go wrong, re-read the module before the exam.

1 Why does stabilisation always cost some of the frame?

A. Moving each frame to cancel the shake leaves empty edges to hide
B. Rolling-shutter correction warps the frame, which shrinks it
C. The tracker discards the outer band of the image before it begins
D. The analysis pass needs a margin of pixels it can measure against

2 A very shaky shot is stabilised. It comes back steady, and every individual frame is blurred. What is the cause?

A. Rolling-shutter correction was applied twice, once on each pass of the analysis
B. The stabiliser reduced the frame rate in order to smooth the motion
C. The motion blur was already inside each frame
D. The perspective mode was used where translation would have been correct

3 A clip slowed with optical flow melts around a striped shirt and is clean everywhere else. Why that region?

A. Stripes are high contrast, so the estimator over-sharpens them
B. A repeating pattern gives the matcher many equally good answers
C. Fabric moves non-rigidly, and optical flow assumes a rigid scene
D. The chroma channels are half resolution, so the vectors are computed at half size

4 Why is rotoscoping described as a budgeted task rather than a quick repair?

A. It cannot be combined with tracking, so every frame is drawn from scratch
B. Machine-assisted masks are Studio features, so the free route is always manual
C. It has to be done in a compositor rather than on the edit page
D. Five seconds of footage costs one to four hours by hand

5 A green screen was lit with one lamp and keys badly. Which step does more than any setting on the keyer?

A. Sharpening the edge of the matte so that the subject's outline is crisp
B. Raising the keyer's tolerance until the whole screen is selected
C. A garbage matte and an evening-out pass before the keyer sees it
D. Increasing the spill suppression until the green edges disappear

6 Removing a light stand from a locked-off shot takes minutes and from a handheld shot takes hours. What is the difference?

A. A static patch slides off, so the patch has to be tracked and matched
B. Handheld shots contain more frames, so more of them must be painted
C. The shot must be stabilised first, and stabilisation crops the area being patched
D. Handheld footage carries motion blur that a patch cannot reproduce

8

MODULE 8

Delivery — the file that leaves the building

Choose a container, codec, bitrate and colour tag for a named destination and justify each from the material and from what the platform's own encoder will do to it, render a master and textless version every later deliverable can be derived from, and diagnose a render that is washed out, drifting, juddering or failing at the same percentage by naming the mechanism responsible.

Lesson 67 · 11 min

Why the upload looks worse than the file

THE ONE THING TO KEEP

The platform re-encodes everything, so upload a generously encoded file at the source frame rate and mix near -14 LUFS rather than louder.

The complaint

Your export looks perfect on your machine. You upload it. The version that comes back has smeared shadows, blocky movement, and bands across the sky. Nothing you did was wrong, and the reason is that you are not uploading a finished video — you are uploading raw material for someone else's encoder.

Understand four things once and export stops being guesswork.

Container and codec are different

The **container** is the box: `.mp4`, `.mov`, `.mkv`, `.webm`. It holds video, audio, subtitles and metadata, and says how they are interleaved. The **codec** is how the video inside was compressed.

An `.mp4` might contain H.264 or H.265 or AV1. That is why "it's an MP4" tells you almost nothing about whether a given player can open it, and why renaming a file extension never fixes anything.

- **H.264 / AVC** — the safe default. Nearly every device on earth decodes it in hardware. Not the most efficient any more, and worth it anyway for anything you hand to another person.
- **H.265 / HEVC** — roughly 30–50% smaller than H.264 at similar quality, and what most modern phones record in. Decode support is good but not universal, and its patent licensing is messy enough that some browsers avoid it.

- **AV1** — royalty-free and more efficient than H.265. Encoding is slow; hardware decode is common on newer devices, absent on older ones. YouTube serves it to clients that can take it.
- **ProRes, DNxHR** — intermediate codecs. Every frame stored whole, files enormous, editing instant. For handing to another editor or archiving a master, never for uploading.

Bitrate is the one that decides how it looks

Bitrate is how many bits per second the encoder may spend. Resolution and codec set the ceiling; bitrate sets what you get.

Roughly, for H.264 uploads (check the platform's current guidance — these numbers move):

- 1080p at 25 or 30 fps: about **8–12 Mbps**
- 1080p at 50 or 60 fps: about **12–18 Mbps**
- 4K at 25 or 30 fps: about **35–45 Mbps**
- 4K at 50 or 60 fps: about **53–68 Mbps**

Some content eats bits far faster than others. Rain, confetti, foliage in wind, water, film grain, smoke and fast camera moves all produce huge change between frames, and the encoder must describe all of it. A talking head against a plain wall at 8 Mbps looks fine; a rainy street at 8 Mbps turns to porridge.

Two practical consequences:

Upload generously. The platform re-encodes whatever you send, so anything you save with a small file is quality thrown away before their encoder starts. Give it two to three times the delivery bitrate.

Upload at a higher resolution than your target. Platforms allocate bitrate by resolution tier, and the 4K tier gets substantially more bits than the 1080p tier. Exporting a 1080p project upscaled to 2160p before upload is a well-known and genuinely effective way to get a better-looking 1080p stream back, particularly for high-motion content. It feels like a cheat. It works because you are moving your video into a more generously funded bucket.

Frame rate and resolution: match, do not convert

Pick the frame rate of your source and stay on it end to end. Shoot 25, cut on a 25 fps timeline, export 25.

Convert and you get judder, because the editor has to invent or discard frames. Putting 30 fps clips on a 25 fps timeline means five frames a second are dropped or blended, which the eye reads as a stutter on every pan.

Regionally, 25 and 50 fps are standard where mains power is 50 Hz — India, most of Europe and Africa — and 30 and 60 where it is 60 Hz. Matching the local mains frequency also avoids rolling flicker under fluorescent and LED lighting.

Slow motion is the exception: shoot 50 or 60 and place it on a 25 or 30 timeline for honest half-speed with real frames.

Audio, and one number that catches people

Export audio as **AAC, 48 kHz, stereo, 192–320 kbps**. There is no reason to go lower and little benefit above.

The number that matters is loudness. Platforms normalise: YouTube brings uploads to somewhere around **-14 LUFS integrated**, and other short-form platforms are in a similar region. Their exact figures change, so treat that as an order of magnitude.

So mastering loud gains you nothing. Crush your mix to -6 LUFS and the platform turns it down by 8 dB: you have destroyed your dynamic range and arrived at the same perceived volume as everyone else. Mix near -14 LUFS and keep true peaks below about -1 dBFS so the encoder has room. Resolve's Fairlight page has a loudness meter; so does any free DAW.

The settings that actually matter, by destination

- **YouTube**: MP4, H.264, high bitrate, source resolution and frame rate, AAC 48 kHz. Consider the upscale-to-4K trick for high-motion work.

- **Instagram, TikTok, Shorts**: MP4, H.264, 1080×1920, under the platform's file size and length limits, captions burnt in. These re-encode hard, so send the best file the app will accept.
- **A client, or another editor**: ProRes 422 or DNxHR SQ in a MOV, plus a separate H.264 for review. Never hand over your only copy as a compressed delivery file.
- **Archive**: keep the project file, the source footage and one high-quality master. Not the YouTube upload.

What you cannot fix at export

Export settings recover nothing. If a shot is soft, noisy, or clipped in the highlights or the audio, a higher bitrate preserves the problem in more detail. Every re-encode loses a little more, so never export an H.264, re-import it, edit and export again — go back to the project.

Today

Export the same thirty seconds twice, at 8 Mbps and 40 Mbps, and watch both full-screen on a phone. If you cannot tell them apart, your content is bit-cheap and you can relax. If you can, you now know why.

BEFORE YOU MOVE ON

Two people upload 1080p videos to YouTube at the same bitrate. One is a talking head against a plain wall and looks clean; the other is a rainy street at night and looks blocky. Both used identical export settings. Why?

A. The rainy footage was probably shot in H.265 and the talking head in H.264, and the mismatch survives re-encoding.
B. Dark footage is compressed more aggressively by the platform because shadow detail is considered less important.
C. The rainy video needed a higher frame rate to capture the motion, and 30 fps was insufficient for the subject.
D. Rain and night footage contains far more change between frames, so the same number of bits per second has to describe much more information and runs out.

The answer and why: p. 488

Lesson 68 · 9 min

The box and the codec are two different decisions

THE ONE THING TO KEEP

A filename tells you the container and nothing about the codec, so repack with a remux rather than a rename, and check a file with ffprobe before you send it anywhere.

The box and the thing inside it

A video file is two separate decisions wearing one filename.

The **codec** is how the pictures and the sound were compressed: H.264, H.265, AV1, VP9, ProRes, DNxHR for picture; AAC, PCM, Opus, MP3 for sound. The **container** is the box that holds those compressed streams side by side, along with the timecode, the track names, the subtitle tracks and the metadata: MP4, MOV, MKV, MXF, WebM.

`holiday.mp4` tells you the box. It tells you nothing about what is inside, which is why "it's an MP4" is not an answer to "what codec is it?" and why one MP4 plays on a phone and another does not.

The confusion is worth clearing up now, because almost every delivery problem in this block is really a container-and-codec problem wearing a different hat.

What each box is for

MP4 is the ISO standard box. It is the safest thing to hand anybody. Browsers play it, phones play it, every platform accepts it. It legally holds H.264, H.265 and AAC, and that is most of what it is used for.

MOV is Apple's QuickTime box, and structurally it is MP4's older sibling — the two share a file format lineage, which is why so many tools treat them interchangeably. MOV is where ProRes lives, and it holds timecode tracks more comfortably than MP4.

MKV (Matroska) is the open box that holds anything: any codec, unlimited subtitle and audio tracks, chapters. It is excellent for archiving and for anything you will watch yourself. Upload one to a client's review system and it may well be rejected, because MKV support outside VLC and media players is patchy.

MXF is the broadcast box. If a spec sheet asks for MXF OP1a, that is a delivery requirement from a system that is older and stricter than you are, and you follow it exactly.

WebM is the browser box for VP9 and AV1. You will mostly meet it as a download from a website rather than as something you make.

Renaming does nothing

The single most common misunderstanding: changing `.mkv` to `.mp4` in Finder or Explorer does not convert anything. The bytes are unchanged. You have put an MP4 label on a Matroska box, and the player either sees through the lie and works, or refuses.

What you actually want is a **remux**: unpack the streams from one box and repack them into another without touching the compression.

```
ffmpeg -i clip.mkv -c copy clip.mp4
```

`-c copy` means "copy the streams, do not re-encode". A two-gigabyte file remuxes in a couple of seconds and loses nothing, because nothing was decoded. Compare that with a re-encode, which takes minutes and always loses a little.

The catch is legality. Not every codec is allowed in every container. H.264 and AAC go into MP4 happily. ProRes in MP4 is not a standard combination and many players will refuse it. FLAC in MP4 is unusual. If a remux fails, ffmpeg will normally say which stream it could not place, and the fix is to re-encode that one stream and copy the rest:

```
ffmpeg -i clip.mkv -c:v copy -c:a aac -b:a 192k clip.mp4
```

Picture copied untouched, audio converted. This is the cheapest repair in the whole delivery chain.

The one MP4 flag worth knowing

An MP4 has an index — the `moov` atom — that says where every frame lives. Encoders often write it at the *end* of the file, because they only know the final layout once they finish.

A file with the index at the end cannot start playing until it has fully downloaded. Uploaded to a web page, it appears broken: a long spinner, then it plays. The fix takes seconds and no quality loss:

```
ffmpeg -i in.mp4 -c copy -movflags +faststart out.mp4
```

Most editors have a "web optimised" or "fast start" checkbox that does exactly this. Tick it for anything going on a website. It does not matter for YouTube or Instagram, which re-encode the file anyway.

Why a colleague cannot open your ProRes

ProRes is a Apple codec, and macOS decodes it everywhere by default. On Windows the situation is better than it used to be and still not universal: recent Windows versions decode ProRes in some applications, VLC and DaVinci Resolve read it regardless, and a random corporate media player may not.

DNxHR is Avid's equivalent and behaves symmetrically — solid everywhere in professional tools, less certain in consumer ones. Both are *intra-frame* codecs, meaning every frame is compressed on its own, which is why they scrub instantly and why the files are enormous.

If you do not know what the other end can open, send H.264 in MP4. It is the format nobody has trouble with.

Honest limits

Container and codec support is not a fixed fact you can memorise once. It shifts with operating system versions, browser versions and hardware. AV1 decoding, for instance, is hardware-accelerated on recent phones and chips and software-only on older ones, where a 4K AV1 file will stutter on a machine that plays 4K H.264 without effort.

The working habit is not to know the compatibility matrix. It is to check the file before you send it, with `ffprobe` or MediaInfo — both free — and to ask the receiving end what they open things in.

Today

Take any file on your drive and run `ffprobe -hide_banner file.mp4`. Read the two lines that say `Stream #0:0` and `Stream #0:1`. You now know its codec, its resolution, its frame rate and its audio format, which is more than the filename ever told you.

BEFORE YOU MOVE ON

Renaming a .mkv file to .mp4 sometimes plays and sometimes does not. What is actually happening?

A. The rename rewrites the file's identifying header, and players that read headers rather than extensions reject the mismatch.

B. Matroska stores its frames in a different order from MP4, so the player has to reorder them and some cannot do it quickly enough.

C. Nothing inside the file changed, so it plays only where the player reads Matroska anyway; a real conversion needs a remux.

D. MP4 requires its index at the start of the file while Matroska writes it at the end, so a renamed file has an index the MP4 parser cannot locate.

The answer and why: p. 489

Lesson 69 · 9 min

Bitrate, and why the same number is generous and stingy

THE ONE THING TO KEEP

Compare bitrates as bits per pixel per second rather than as raw megabits, and expect motion, fine detail and grain to need several times what a static interview needs.

A number with no meaning on its own

Bitrate is bits per second: how many bits the encoder is allowed to spend describing each second of video. Ten megabits per second is 1.25 megabytes per second, which is 75 MB per minute, which is 4.5 GB per hour. That arithmetic is worth doing in your head, because it converts an abstract slider into a file you have to move.

What bitrate is *not* is a quality setting you can quote on its own. "Twenty megabits" means something very different at 1080p25 than at 2160p50, and something different again for a static interview than for a football match.

Bits per pixel: the only fair comparison

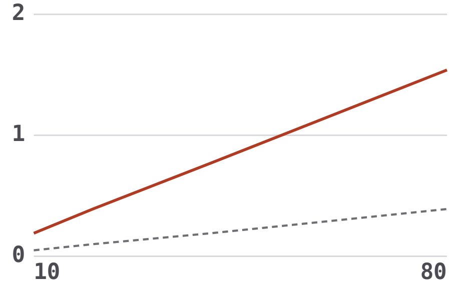
Divide the bitrate by the number of pixels per second and you get a figure you can compare across formats.

At 1080p25: 1920 × 1080 × 25 = 51.8 million pixels per second. Ten megabits across that is about **0.19 bits per pixel**. That is comfortable for H.264 on ordinary material.

At 2160p25 the pixel count is four times higher — 207 million per second. To hold the same 0.19 bits per pixel you need roughly **40 Mbit/s**. This is why exporting 4K at the bitrate that looked fine for 1080p produces a mush of smeared detail, and why people conclude their 4K camera is disappointing.

Figure 8.1

The same megabits, at two resolutions



Across: Bitrate, megabits per second
Up: Bits per pixel
—— 1080p25
- - 2160p25

A 2160p file carries four times the pixels, so it needs four times the bitrate to hold the same bits per pixel. This is why 4K exported at a 1080p number smears, and why people blame the camera.

H.265 and AV1 reach similar quality at roughly half the bitrate of H.264 — the usual claim is 40 to 50% savings, and on real footage it lands somewhere in that range rather than reliably at the top of it. Newer codecs buy their efficiency with computation: an AV1 encode can be many times slower than H.264 on the same machine.

Content changes everything

Compression works by not describing what it can predict. A locked-off interview against a plain wall is almost entirely predictable: the encoder describes the face, copies the wall, and spends very little.

Three things destroy that prediction, and it is worth knowing which, because you can sometimes act on them.

Motion, especially whole-frame motion. A handheld whip pan changes every pixel at once. Confetti, rain, water, leaves in wind and crowd shots are all worst cases.

Fine detail, such as foliage, gravel, fabric weave and sequins.

Grain and noise, which are the cruellest, because noise is random. Random means unpredictable, and unpredictable means the encoder must spend real bits on it or throw it away. This is why a beautifully grainy grade looks like porridge after upload, and why *denoising slightly before export* can make an image look better after compression even though it looks softer before it.

A talking head at 8 Mbit/s can look flawless. The same 8 Mbit/s on a snowstorm falls apart into blocks. Both are 8 Mbit/s.

CBR, VBR and CRF

Three ways to tell an encoder how much to spend.

CBR — constant bitrate — spends the same amount every second regardless of need. It wastes bits on easy passages and starves hard ones. Its one real use is streaming over a fixed-capacity link, and broadcast delivery specs that demand it.

VBR — variable bitrate — is given a target average and allowed to move bits from easy scenes to hard ones. Two-pass VBR analyses the whole file first, then encodes knowing where the difficulty is. Use two-pass when you must hit a file size, and expect it to take roughly twice as long.

CRF — constant rate factor — is the one to reach for when file size is not fixed. You give it a quality number and it spends whatever that quality costs. For x264, CRF 18 is visually near-transparent, CRF 23 is the default and fine for most web work, CRF 28 is visibly soft. Lower is better and each step of about 6 roughly halves or doubles the file.

```
ffmpeg -i master.mov -c:v libx264 -crf 20 -preset slow -pix_fmt
yuv420p \
  -c:a aac -b:a 192k delivery.mp4
```

`-preset slow` tells the encoder to work harder for the same quality target, producing a smaller file at the same CRF. Presets cost time, not quality: `veryslow` might take three times as long as `medium` for perhaps 10% smaller output.

Most editing applications hide CRF behind a "quality" slider, and Resolve's Automatic quality setting on H.264 is doing approximately this. If you want the number itself, export a master and re-encode with ffmpeg.

The rule that saves the most grief

Encode generously into the platform, because the platform will encode again. Every re-encode is a fresh loss, and losses compound — this is **generation loss**, and it is why a file that has been through WhatsApp twice looks like a photocopy of a photocopy.

Practically: if you are uploading, do not pre-compress to save upload time. Give YouTube a 1080p file at 16 Mbit/s or a 4K one at 45 Mbit/s and let its encoder work from something clean. The 400 MB you saved by exporting at 4 Mbit/s buys you visible blocking in every viewer's browser, permanently.

Honest limits

Bitrate ladders on platforms change without announcement, and any specific figure in this lesson is a reasonable default rather than a law. The durable part is the reasoning: bits per pixel, content difficulty, and never handing a compressor something already compressed.

Today

Export the same 30-second clip twice — once at CRF 20, once at CRF 30 — and compare file sizes and the look of the most detailed shot. Then do it again with a clip containing motion and grain. The gap between the two pairs is the lesson.

BEFORE YOU MOVE ON

Two shots are exported in the same 12 Mbit/s 1080p25 file. The interview is flawless; the confetti shot breaks into blocks. Why?

A. Compression spends bits on what it cannot predict, and confetti is unpredictable in almost every pixel of every frame.
B. The confetti shot carries more pixels than the interview, so the same bitrate has to cover a larger picture.
C. Rate limits apply only to the keyframe at the head of each group of pictures, so shots that need more keyframes are starved of bits.
D. Fast motion exceeds the temporal detail the frame rate can represent, so the motion is lost before the encoder sees it and no bitrate recovers it.

The answer and why: p. 489

Lesson 70 · 9 min

4:2:0 and 8 bits: where the colour went

THE ONE THING TO KEEP

Delivery formats keep one colour sample per four pixels and 256 levels per channel, which costs you saturated edges and smooth gradients — and a trace of added grain hides the banding that results.

Your picture has less colour in it than you think

Delivery codecs do not store a full colour value for every pixel. They store brightness for every pixel and colour for only some of them, because the eye is far more sensitive to changes in brightness than to changes in hue. The shorthand for how much colour survives is **chroma subsampling**, written as three numbers.

4:4:4 keeps a colour sample for every pixel. Full colour. Used in graphics work, ProRes 4444, and almost never in delivery.

4:2:2 halves colour horizontally: one colour sample per two pixels across. This is broadcast-grade, and it is what ProRes 422 and DNxHR carry.

4:2:0 halves colour both horizontally and vertically: **one colour sample for every 2 × 2 block of four pixels**. Three quarters of the colour information is gone before the codec even starts compressing.

Almost everything you deliver is 4:2:0, because that is what phone and television decoder chips do in hardware. Your camera probably shot 4:2:0 as well, unless it was an expensive one or you recorded externally.

Figure 8.2

How much colour each scheme keeps

4:4:4, one colour sample per pixel
Graphics work and ProRes 4444. Almost never a delivery format

4:2:2, one per two pixels across
Broadcast grade. ProRes 422 and DNxHR carry it, and masters are kept in it

4:2:0, one per 2 by 2 block
Three quarters of the colour is gone before the codec starts compressing

Almost everything you deliver is 4:2:0, because that is what phone and television decoder chips do in hardware. It shows on red text, thin coloured lines, hard colour edges and green-screen keys.

Where you can see it

Most of the time you cannot, which is why the scheme survives. The places it shows are specific and predictable.

Saturated red on a dark background is the classic. Red carries most of its energy in chroma, so red text or a red line on black smears into a soft fringe. If a title looks crisp in the timeline and mushy after export, this is almost always why.

Thin coloured lines and small text, particularly in screen recordings of code editors with syntax colouring.

Hard colour edges — a graphic with a magenta panel meeting a cyan one will show a stepped, blocky seam under motion.

This has a practical consequence for the text block earlier in this course: heavier weights and larger sizes survive 4:2:0, and hairline coloured type does not. The advice to avoid thin coloured fonts is not taste. It is a codec constraint.

It also explains why keying a green screen shot from a consumer camera is so painful. The key lives entirely in colour, and in 4:2:0 the colour edge is a quarter of the resolution of the picture you are trying to cut out.

Bit depth and banding

Separately from how *many* pixels carry colour is how *finely* each value is recorded.

8-bit gives 256 levels per channel. **10-bit** gives 1024. That sounds like a small difference and it is not, because the visible failure is not a loss of vividness but **banding**: a smooth gradient, such as a sunset or a studio backdrop lit from one side, rendered as a series of flat steps with hard edges between them.

A clear sky at dusk might legitimately need 400 distinct levels between horizon and zenith. At 8-bit you have 256 for the entire range from black to white, and perhaps 40 in the part of the range the sky occupies. The steps become visible, and once you have seen one you cannot unsee it.

Two things make banding worse. Grading is one: stretching contrast on an 8-bit file spreads its existing 256 levels further apart, which is why grading log footage shot in 8-bit is difficult. Compression is the other, because the encoder smooths noise away and noise was the only thing disguising the steps.

The fix nobody expects

Add a small amount of grain before export.

It sounds backwards. It works, and it is the standard trick in every finishing house. Banding is visible because the boundary between two levels is a clean contour your eye traces. Adding noise of roughly half a level of amplitude makes pixels near the boundary randomly fall either side of it, which converts a hard edge into a soft dither. The eye averages it back to a smooth gradient.

In Resolve, a very light Film Grain or a Noise node at low strength over the offending region does it. In free tools, ffmpeg can:

```
ffmpeg -i in.mov -vf "noise=alls=3:allf=t" -c:v libx264 -crf 18
out.mp4
```

Keep it subtle — strength around 2 to 4 in that filter. Enough to break the contour, not enough to notice as texture. The cost is bitrate, because noise is expensive to compress, so use it where the problem is rather than over the whole frame if your tool allows a mask.

The other genuine fix is to deliver 10-bit where the platform supports it, which for HEVC and AV1 it increasingly does. Ten-bit H.264 is possible and poorly supported, so it is not the route.

What to do about it in practice

Shoot the best chroma and bit depth you can afford, because the losses are cumulative and only the first one is recoverable. Grade and composite in a high-precision working space even if you deliver 8-bit 4:2:0 — Resolve's internal processing is 32-bit float regardless of your source, so the quantisation happens once at export rather than at every step.

Keep your masters 10-bit 4:2:2 (ProRes 422 HQ or DNxHR HQX) so that every future delivery is a first-generation encode from good material.

And design graphics knowing where they land: chunky, high-contrast, not relying on saturated red edges against black.

Honest limits

None of this is visible on a phone in daylight, and a good deal of the anxiety about chroma subsampling is misplaced for material that will be watched once on a train. The cases where it genuinely bites are text, graphics, keying and gradients. For a documentary interview in a room, 8-bit 4:2:0 delivery is completely adequate and always has been.

Today

Make a title card: pure red text on black, 1080p, at a thin weight. Export it as H.264 and view it at 100%. Then make the same card in white and compare the edges. That difference is 4:2:0.

BEFORE YOU MOVE ON

A thin red title on black looks soft and fringed after export while the identical title in white stays crisp. What explains the difference?

A. Saturated red sits near the edge of the Rec.709 gamut and exceeds legal levels, so the encoder clips it and the edge collapses.
B. Eight-bit quantisation leaves too few levels to describe a hard red transition, so the edge is rounded off.
C. The title was set at a weight that compresses badly, and the same weight would smear in white too if the background were brighter.
D. Colour is stored at a quarter of the resolution of brightness, and a red edge on black is carried almost entirely by colour.

The answer and why: p. 490

Lesson 71 · 9 min

What the platform does to your upload

THE ONE THING TO KEEP

Your upload is source material for a ladder of renditions you never see, so hand it something clean and uncompressed-twice, match the frame rate, and judge nothing until processing has finished.

Nobody watches the file you uploaded

The earlier lesson established the fact: platforms re-encode everything. This one is about the machinery, because once you know the shape of it you stop guessing about upload settings.

When you upload, the file is decoded and re-encoded into a **ladder** — a set of renditions at different resolutions and bitrates, so a viewer on a weak connection gets 360p and a viewer on fibre gets 2160p, and the player can switch between them mid-stream without stopping. Your file is the source material for that ladder. It is never itself served to anybody.

What the ladder looks like

A typical ladder runs something like 144p, 240p, 360p, 480p, 720p, 1080p, 1440p, 2160p, each with its own target bitrate. The player picks a rung based on measured bandwidth and screen size, and moves between rungs as conditions change.

Two consequences follow immediately.

First, **the rung most people see is not the top one**. On mobile data, on a phone screen, a lot of viewing happens at 480p or 720p. Your careful 4K grade is being watched at a quarter of a million pixels. This is an argument for large, clear compositions and readable text, which is a craft decision, not a technical one.

Second, **the rungs are not encoded equally**. Platforms reserve their better codecs for their more valuable renditions. YouTube encodes popular and higher-resolution content with VP9 and increasingly AV1, both of which look considerably better per bit than H.264, while a low-view 1080p upload may be served as H.264 only. This is the mechanism behind a piece of folklore you will meet constantly.

The 4K upload trick, stated honestly

The advice circulates in this form: "upload your 1080p video as 4K and the 1080p version will look better."

The mechanism is real. Uploading at 2160p puts the file in a class that receives higher bitrate allocations and better codec treatment across the ladder, so the 1080p rendition served from a 4K upload has sometimes measurably outperformed the same video uploaded at 1080p.

Three caveats, and they matter.

It is **not documented**. Platforms do not publish their allocation rules, which means the behaviour has changed before and will change again.

It **costs you an upscale**. Rendering 1080p material to a 2160p canvas takes four times the pixels, four times the encode time and four times the upload. The upscale itself adds nothing.

And the size of the gain is **small compared to the things you control** — exposure, contrast, noise, motion. If your source is clean and generously encoded, you are already collecting most of what is available.

Treat it as a marginal optimisation to test yourself rather than a rule to follow. That is the honest position, and anyone who states it more confidently than that is guessing too.

Processing happens in stages

Immediately after upload, the platform publishes a low-resolution rendition so the video is watchable at once. The higher rungs arrive over the following minutes, and for a long 4K file, over hours.

This is why a video looks terrible the moment you publish it and fine the next morning, and why the first comment on a lot of channels is somebody complaining about quality. Do not re-export, do not re-upload, and do not judge anything in the first half hour. Check the quality menu in the player: if it only offers 360p and 720p, processing has not finished.

Double compression is the real enemy

Every encode throws information away permanently. Encoding twice throws away what the second encoder thinks is unimportant *in a picture already stripped of detail by the first*, and the artefacts of the first pass — blocking, banding, smeared motion — become structures the second pass faithfully preserves, spending bits on ugliness.

Practical rules that follow:

- Upload from a **master**, not from a file you already exported for something else.
- Never take a file down from one platform and upload it to another. Export a fresh one.
- Do not pre-compress to speed up an upload. A slow upload once is cheaper than permanent damage.
- If you must send through a messaging app, accept that it will be destroyed and export deliberately small so that at least the damage is done once by you and not twice.

Match frame rate, do not convert it

Upload at the frame rate you edited at. A 25p timeline exported as 30p has had frames duplicated by your editor, and the platform then encodes those duplicates faithfully, producing a permanent stutter that no bitrate fixes.

Platforms handle 24, 25, 30, 50 and 60 correctly. There is no advantage to converting, and one obvious way to ruin the motion.

Vertical and the crop that happens without you

Feeds crop. A 16:9 video in a vertical feed is shown letterboxed or cropped to a square or 4:5 frame depending on the surface, and a 9:16 video shown on a landscape player gets pillarboxed. If a video matters on a particular surface, export for that surface rather than hoping.

Reels, Shorts and TikTok all want 1080 × 1920. Feed posts do better at 4:5 (1080 × 1350) than at 1:1 because it occupies more vertical space on a phone. This is the practical reason to keep an ungraded, uncropped master at full frame: reframing from a master is an export, while reframing from a delivered file is a crop of already-compressed pixels.

Honest limits

Everything specific here — codec choices, resolution tiers, processing behaviour — is observed rather than published, and observed behaviour on a platform is a snapshot. The durable rules are the two that follow from compression itself: encode once, and hand the platform the cleanest thing you can.

Today

Open a video of your own on YouTube, press the quality button and note which rungs exist, then open the statistics-for-nerds panel and read the codec line. If it says `vp09`, you have seen the ladder choose something better than H.264 for you.

BEFORE YOU MOVE ON

Uploading a 1080p edit on a 2160p canvas sometimes improves how the 1080p rendition looks. What is the actual mechanism, stated honestly?

A. The upscale triggers the platform's detail enhancement pass, which runs only on 4K uploads and reconstructs texture the 1080p path discards.
B. Higher-resolution uploads receive better codec and bitrate treatment across the whole ladder — undocumented behaviour that has changed before.
C. 4K uploads bypass the transcoding ladder and are served as uploaded, so the second compression never happens.
D. The player defaults to the highest available rendition on a 4K upload, so viewers end up watching 2160p instead of selecting 1080p for themselves.

The answer and why: p. 490

Lesson 72 · 10 min

The numbers, destination by destination

THE ONE THING TO KEEP

Save a named render preset per destination with the frame rate taken from the timeline rather than the preset, because the wrong frame rate is the mistake that survives every other check.

Settings you can actually type in

This lesson is the one to come back to. Everything before it explains why these numbers; here are the numbers.

Two rules first, because they apply to every line below. **Frame rate always matches the timeline.** And **audio is 48 kHz**, in every destination, always — 44.1 kHz is a music convention and mixing the two is how a sync drift starts.

YouTube and long-form web

- H.264, MP4 container, High profile
- 1080p25 or 1080p30: **16 Mbit/s**, VBR two-pass or CRF 18
- 1080p50 or 1080p60: **24 Mbit/s**
- 2160p25 or 2160p30: **45 Mbit/s**
- AAC audio, stereo, **320 kbit/s**, 48 kHz
- Integrated loudness around **−14 LUFS**, true peak **−1 dBTP**

YouTube publishes recommended bitrates in roughly this region and its own encoder will ignore your exact figure anyway; what you are buying is a clean source for it to work from. Going to 50 Mbit/s at 1080p wastes your upload time and changes nothing.

Instagram Reels, TikTok, YouTube Shorts

- **1080** × **1920**, 9:16, H.264, MP4
- 25 or 30 fps, **8 to 12 Mbit/s**
- AAC 192 kbit/s, 48 kHz
- Loudness around −14 LUFS

Keep anything that must be read out of the top 15% and bottom 20% of the frame, where the platform's own interface sits. The safe-area lesson in the text block covers the reasoning; these are the numbers to design against.

Instagram feed

- **1080** × **1350** (4:5) for maximum height in the feed, or 1080 × 1080 square
- Otherwise as above

Client review copies

- H.264, MP4, **1080p at 8 to 10 Mbit/s** — small enough to send, good enough to judge
- **Burnt-in timecode** in a corner, and the version number burnt in with it
- Audio: the current mix, not a placeholder, even if it is rough

The timecode is not decoration. It converts "the bit near the middle where the music is wrong" into "02:14", and that single change is worth more to your schedule than any other habit in this course.

WhatsApp and messaging apps

- **720 × 1280** or 1280 × 720, H.264
- **2 to 3 Mbit/s**, and keep the file under about 16 MB where you can
- AAC 128 kbit/s

These apps re-encode aggressively to a small target regardless of what you send. Exporting small deliberately means the damage happens once, under your control, rather than twice.

Archive master

- **ProRes 422 HQ** (MOV) or **DNxHR HQX** (MOV or MXF), 10-bit
- Full timeline resolution, no platform cropping, no burnt-in captions
- Audio as 24-bit PCM, not AAC

A ten-minute 1080p25 ProRes 422 HQ master runs at roughly 184 Mbit/s, which is about **14 GB**. That is the price of never re-opening the project to make a new deliverable.

Broadcast and agency delivery

Whatever the spec sheet says, exactly. There is a lesson on reading one later in this block. Do not improvise here; a rejected delivery costs a day.

Where the free tools stand

DaVinci Resolve's free version exports H.264 and H.265 in MP4 and MOV, ProRes and DNxHR on the platforms where those are available, up to 3840 × 2160. Some codec and resolution options are reserved for Resolve Studio and the exact list has moved between versions and operating systems — check the render page on your own install rather than trusting a list from the internet, including this one.

Anything Resolve declines to write, **ffmpeg will**, free, on every platform. Export a ProRes or high-bitrate H.264 master from the editor and convert:

```
ffmpeg -i master.mov -c:v libx265 -crf 24 -tag:v hvc1 \
  -c:a aac -b:a 192k out.mp4
```

The `-tag:v hvc1` is needed for H.265 files to play in QuickTime and Safari; without it they play in VLC and nowhere else. **Shotcut** and **Kdenlive** also export the full range of H.264, H.265 and VP9 without restriction, and both run on machines Resolve will not.

The three checkboxes people miss

Frame rate. Set from the timeline, not from the preset. A preset called "YouTube 1080p" often carries 29.97 and will convert your 25p edit silently.

Web optimised / fast start. On, for anything going to a website of your own.

Render at source resolution vs timeline resolution. Some presets quietly upscale or downscale. Read the resolution field, do not assume.

Build presets, once

Every editor lets you save a render preset. Make one per destination, name it properly — `YT-1080p25-16M`, `Reel-1080x1920-10M`, `Master-ProRes422HQ` — and stop retyping numbers at eleven at night, which is when the wrong frame rate gets exported.

In Resolve, set up the Deliver page and press the plus on the preset row. The preset stores the codec, bitrate, audio and resolution, though not the render range, which you still set each time.

Honest limits

Platform recommendations change and the figures above are current practice rather than permanent truth. They are also generous by design: nothing here will look bad, and several of them could be trimmed by a third with no visible difference on a phone. When in doubt, err upward on the upload and downward on anything a person has to download.

Today

Create and save three render presets in your editor: master, web, and review-with-timecode. Name them so that the numbers are in the name.

BEFORE YOU MOVE ON

A 25p edit exported with a preset called "YouTube 1080p" plays with an even stutter on every pan. What went wrong?

A. The preset's bitrate is too low for 1080p, so movement is the first thing to break up under compression.
B. The preset wrote a variable frame rate file, and the player is resampling it unevenly during playback.
C. The preset carried 29.97 fps, so frames were duplicated unevenly to pad 25 up to 30.
D. The preset specified 44.1 kHz audio rather than 48 kHz, so the player resamples and drops video frames to hold sync with the stretched audio.

The answer and why: p. 491

Lesson 73 · 10 min

Six ffmpeg commands worth knowing

THE ONE THING TO KEEP

ffmpeg never modifies its input, so inspecting, remuxing, converting a variable frame rate recording and measuring the loudness of a finished file are all free experiments.

The tool that does the jobs your editor refuses

ffmpeg is a free command-line program that reads and writes essentially every video and audio format there is. Every editor you have used has some version of it, or something like it, buried inside. Learning six commands from it removes a whole class of problem from your week: the file that will not import, the client who needs it 4 MB smaller, the screen recording that drifts out of sync, the codec your free editor declines to write.

Install it with `brew install ffmpeg` on macOS, `sudo apt install ffmpeg` on Debian or Ubuntu, `winget install ffmpeg` on Windows, or from the official downloads page. It has no interface. That is the point: it does one job per line and leaves the machine free.

The general shape is always the same:

```
ffmpeg -i input.mov  [what to do]  output.mp4
```

Input first, options, output last. The last filename is always the one being written, and ffmpeg will ask before overwriting.

1. Find out what a file actually is

```
ffprobe -hide_banner clip.mp4
```

The two lines beginning `Stream #0:0` and `Stream #0:1` tell you codec, resolution, frame rate, pixel format and audio sample rate. When somebody says "it won't import", this is the first thing to run, and it is usually the whole diagnosis. For a tidier answer:

```
ffprobe -v error -select_streams v:0 \
  -show_entries
stream=codec_name,width,height,r_frame_rate,pix_fmt \
  -of default=noprint_wrappers=1 clip.mp4
```

2. Change the box without touching the picture

```
ffmpeg -i clip.mkv -c copy clip.mp4
```

Seconds, not minutes, and no quality loss at all, because nothing is decoded. Covered in the container lesson; it belongs on this list because it is the command you will use most.

3. Re-encode to a specific quality

```
ffmpeg -i master.mov -c:v libx264 -crf 20 -preset slow \
  -pix_fmt yuv420p -c:a aac -b:a 192k out.mp4
```

`-crf 20` is the quality dial — lower is better, 18 is near-transparent, 23 is the default. `-preset slow` trades encoding time for a smaller file at the same quality.

`-pix_fmt yuv420p` is the single most useful flag on this page. A file encoded as `yuv444p` or `yuv422p` plays perfectly in VLC and refuses to play in QuickTime, Safari and most browsers. When a file works on your machine and not on the client's, check the pixel format first.

4. Trim without re-encoding

```
ffmpeg -ss 00:01:30 -i in.mp4 -t 60 -c copy out.mp4
```

Start at one minute thirty, take sixty seconds, copy the streams. Instant, lossless, and **it will not land on the exact frame you asked for**. With `-c copy` the cut must begin on a keyframe, and keyframes in a long-GOP file are typically one or two seconds apart, so the start can move by up to that much. For a rough extract this is fine. For a frame-accurate cut, drop `-c copy` and let it re-encode.

5. Fix a variable frame rate recording

Screen recorders and phones often write **variable frame rate** files: 30 fps when the screen is still, 12 fps when nothing moves. Editors assume a constant rate, so the audio holds and the picture drifts, and by minute eight the lips are a third of a second late.

```
ffmpeg -i screen.mp4 -vf "fps=30" -c:v libx264 -crf 18 \
  -c:a copy fixed.mp4
```

Do this before importing, every time, for anything that came from a screen recorder, a phone screen capture or a game capture tool. The setup block warned about drift; this is the command that cures it.

6. Measure loudness on the exported file

```
ffmpeg -i final.mp4 -af loudnorm=print_format=summary -f null -
```

It prints integrated loudness, loudness range and true peak for the actual delivered file rather than for your timeline. The sound block covered what the numbers mean. This is how you check them on the thing you are about to send, which is the only place the check counts.

Two more worth keeping

Extract the audio for a transcription service, or for a sound editor:

```
ffmpeg -i film.mov -vn -c:a pcm_s24le audio.wav
```

Make a contact-sheet-style still every ten seconds, for logging or for thumbnails:

```
ffmpeg -i film.mov -vf "fps=1/10" stills/frame_%04d.jpg
```

How to be safe with it

ffmpeg never modifies its input. Everything it does produces a new file, so the worst outcome of a wrong command is a wasted file and a wasted minute. That makes it a good place to experiment.

Two habits keep it painless. Work in the folder the files are in, so filenames stay short. And when a command fails, read the **last three lines** of the output rather than scrolling back through the banner — the actual error is nearly always there, and it nearly always names the stream it could not handle.

Honest limits

ffmpeg is not an editor and this is not a route around learning one. It has no timeline, no scopes and no sense of what a cut is. Its filter syntax is genuinely awkward, the documentation is written for people who already understand it, and half the answers you will find online are for versions from a decade ago. Use it for the six jobs above and reach for the editor for everything else.

Today

Run `ffprobe` on the last file a client sent you. Then remux it to MP4 with `-c copy` and time how long it takes.

BEFORE YOU MOVE ON

A trim with `-ss 00:01:30 -t 60 -c copy` produces a file that starts about a second later than asked. Why?

A. With the streams copied rather than re-encoded, the cut can only start on a keyframe, which may be a second or two away.

B. Placing `-ss` before `-i` performs an approximate seek, and moving it after the input would make the copy frame-accurate.

C. Copying streams discards the source timecode track, so the output is timed from the file's own first frame instead of from the requested point.

D. The container rounds start times to whole seconds whenever no re-encode takes place, so any fractional start position is discarded.

The answer and why: p. 491

Lesson 74 · 9 min

The master, and the versions to make while the project is open

THE ONE THING TO KEEP

Render a full-frame, textless, intra-frame master plus stems at the end of every project, because deriving a new deliverable from a file always works and deriving one from a project only works while the application, fonts, plugins and media all still resolve.

One export that everything else comes from

There is a difference between the file you send somebody and the file you keep. The one you keep is the **master**, and making one properly is the difference between a five-minute job and a two-day one when a request arrives in November for a square version with the logo changed.

A master is a single, high-quality, destination-neutral render of the finished programme. Every delivery is made from it. If your only surviving artefact is the project, then producing anything new requires the project to still open, the media to still relink, the fonts to still be installed and the plugins to still exist. A master requires none of that. It is a file. Files keep.

What format, and why it is not H.264

Masters are written in **intra-frame** codecs: ProRes 422 HQ, DNxHR HQX, or their higher variants. Every frame is compressed independently, there is no prediction between frames, and the quality is high enough that a second encode from it is indistinguishable from an encode from the timeline.

This matters because of **generation loss**. Encode a file with H.264, then encode the result again, and the second encoder faithfully preserves the first one's blocking while adding its own. Three generations in, faces have plastic edges. Intra-frame masters at high bitrate are not immune in theory and are effectively immune in practice: the losses are far below what the next encode will throw away anyway.

The arithmetic is unsentimental. ProRes 422 HQ at 1080p25 runs about **184 Mbit/s**, so ten minutes is roughly **14 GB**, and an hour is about 83 GB. At 4K, multiply by four. A 4 TB drive holds a lot of ten-minute films and not many hour-long ones, which is a real budgeting consideration rather than an abstraction.

If storage is genuinely tight, **ProRes 422** rather than HQ is about two thirds of the size and visually very hard to separate for delivery purposes. **ProRes LT** is thinner again and starts to matter if the file will be graded further. Do not master to H.264 to save space; master to a lighter ProRes instead.

What resolution and frame rate the master is in

The highest you will ever need, and the frame rate you shot.

If the timeline is 1080p25, master at 1080p25 even if a client wants 4K later — upscaling at master stage bakes in an interpolation you cannot remove, whereas upscaling at delivery stage is a decision you can revisit with better tools next year.

If you shot 4K and delivered 1080p, master at 4K if you still have the media and the disk, because the reframe request is the most common late request there is.

Never master at a platform's aspect ratio. Master at the full frame, and crop at delivery.

The versions worth making at the same time

While the project is open and your head is in it, render the variants that are expensive to reconstruct later.

The textless master. The same programme with every title, lower third, subtitle and logo switched off. It costs one extra render now. It is the only way a translated version, a re-branded version or a corrected-spelling version gets made without rebuilding the graphics. Broadcast specs frequently require it, and clients who did not ask for it are delighted when it exists.

The clean version, meaning graded picture with no burnt-in captions, distinct from the caption-burnt web version.

Audio stems, at minimum three: dialogue, music, effects. The sound block covered how to route them. Rendering them now means a future dub into another language is possible at all; without an M&E stem — music and effects with dialogue removed — it is not.

A stills export of three or four frames at full resolution, for thumbnails, press and your own portfolio.

These four together are perhaps fifteen minutes of render time at the end of a project, and they are what separates a deliverable archive from a folder of one file.

Intermediates during the edit

A master is an end-of-project artefact. The same idea appears mid-project under a different name.

When a sequence is finished and expensive — a heavily composited title sequence, a long graded montage, a stabilised drone shot with noise reduction — render it out as an intermediate in the same intra-frame codec and re-import it as a clip. The timeline plays it instantly, the effects are frozen, and nothing recomputes.

Keep the original nodes and layers in a parked timeline so the intermediate can be remade. The rule is: freeze what is finished, keep the recipe.

This is also how you work across applications. A shot sent to Blender or Fusion for compositing goes out as an image sequence or a ProRes clip and comes back the same way, because intermediates are the only currency two different programmes reliably share.

The one thing people get wrong

They make the master *after* burning in captions or cropping for a platform. Then the master carries decisions that belong to one destination, and every other destination inherits them: a square crop with a caption block baked into the middle of the frame, from which no landscape version can ever be made.

Master first, clean, full frame, no text. Then derive.

Honest limits

Masters are large, and for a lot of ordinary work — a weekly vlog, a social clip, a wedding highlight nobody will re-cut — a good H.264 at 40 Mbit/s is a perfectly reasonable keeper and costs a fortieth of the space. The question is whether anybody will ever encode from it again. If the answer is genuinely no, the intra-frame master is ceremony.

Today

Take the last thing you finished, open it, and render a textless version. Note how long it took. That number is what you will be comparing against the day somebody asks for the titles in Tamil.

BEFORE YOU MOVE ON

Four months after delivery a client asks for the same film with titles in Tamil. Which artefact makes that cheap?

A. The graded H.264 web delivery, because a replacement title can be masked over the old one and the file exported again.
B. The project file, because the titles are live text and can simply be retyped.
C. The audio stems, because a dub into another language is the expensive part of a localisation and the picture can be left exactly as it was delivered.
D. The textless master, because the new titles are laid over clean picture with nothing to relink.

The answer and why: p. 492

Lesson 75 · 10 min

Seven ways a render lies to you

THE ONE THING TO KEEP

Render a sixty-second test from the hardest minute of the timeline and open it in two players, because gamma shifts, hardware artefacts, sync drift and file size all declare themselves in a minute and cost an evening after four hours.

Seven failures and the mechanism behind each

An export that fails loudly is easy. The dangerous ones succeed and are wrong, and they are wrong in a small number of recognisable ways. Each of these has a cause you can name, which means each has a fix you can apply rather than a setting you can flail at.

1. The exported file looks washed out, or darker, than the timeline

This is the classic Mac gamma shift: Resolve shows the grade correctly, QuickTime plays the render noticeably lifted or crushed, and you regrade to compensate and now it is wrong everywhere else.

The cause is **colour tagging**. An H.264 or ProRes file carries metadata describing its colour primaries, transfer function and matrix. Some players trust that metadata, some ignore it and assume Rec.709, and some apply a further correction of their own. When the file is untagged or mis-tagged, two players disagree about the same bytes.

Fix it at the source: in Resolve's Deliver page, set the colour tag explicitly — Rec.709, Gamma 2.4 for broadcast or Rec.709-A for Apple-ecosystem web delivery — rather than leaving it automatic. Then verify by opening the render in **two different players**, and in a browser. Do not regrade to chase one player. You will be chasing it for ever.

2. Audio drifts out of sync over a long export

Not a constant offset, which would be a sync error from the edit. A drift that starts at zero and grows to half a second by the end.

The cause is almost always a **variable frame rate source**, or a sample-rate mismatch: 44.1 kHz audio in a 48 kHz timeline, resampled by something that rounded. Screen recordings, phone captures and game capture are the usual suspects.

Fix by converting sources to constant frame rate before import — the ffmpeg lesson has the command — and by keeping every asset at 48 kHz.

3. Green, black or garbled frames, usually near the start

The cause is **hardware encoding or decoding**. GPU encoders are enormously faster than software ones and occasionally produce broken output, particularly with older drivers or unusual resolutions.

Fix by switching the render to software encoding — in Resolve, the Advanced Settings on the Deliver page have a hardware acceleration toggle; in ffmpeg, use `libx264` rather than `h264_nvenc` or `h264_videotoolbox`. It is slower and it is correct. If the corruption is in playback rather than export, turn hardware **decoding** off in preferences instead.

4. The export stops at 43% every time

A render that fails at the same point is telling you exactly where the problem is. Something at that timecode is unreadable: a partially copied file, a clip with a corrupt tail, a still image in a format the renderer cannot handle, a missing font.

Find it by bisecting. Set the render range to the first half. If it succeeds, the fault is in the second half; halve again. Four or five renders locates a single clip. Then re-transcode that clip to ProRes with ffmpeg and replace it — a transcode reads the file once, and it either succeeds, in which case you have a clean replacement, or it fails at the same frame, in which case you know the media itself is damaged and can go back to the card.

5. Silence on one platform, sound everywhere else

The cause is **track layout**. An export with four separate mono tracks plays as expected in a professional player and may be reduced to the first track, or to silence, by a consumer one. Or a stereo mix rendered with content only on the left plays silent in the right ear on headphones and perfectly on a phone, whose single speaker sums both.

Fix by rendering web deliveries as a single stereo pair, and checking the output on headphones before sending. Keep the multi-track version for broadcast, where the spec asks for it.

6. Motion judders although nothing juddered in the timeline

Two causes, and they look similar.

Frame rate conversion. A 25p timeline rendered at 29.97 must invent 4.97 frames per second, which it does by duplicating, and duplication under motion reads as a stutter. Match the timeline.

Field order. Interlaced material rendered with the fields swapped plays as a rapid back-and-forth judder on anything moving. If you are working with old broadcast material, check upper field first against lower field first, and prefer deinterlacing to progressive at import so the problem cannot reach the render at all.

7. The file is enormous, or absurdly small

Enormous usually means an intra-frame codec was selected by accident — ProRes where H.264 was intended — or that the quality slider was left at maximum with a bitrate ceiling removed.

Absurdly small usually means the render range was set to an in-and-out point you forgot about, and you have exported forty seconds of a nine-minute film. Always check the **duration** of the finished file against the timeline duration before sending anything. It takes two seconds and it catches a whole category of embarrassment.

The habit that prevents most of this

Render a **60-second test** from the most difficult part of the timeline before committing to the full export. The difficult part means: the graded shot, the composited title, the section with the imported screen recording, the place where the music changes.

A minute of render tells you about gamma, about audio, about hardware artefacts and about file size per second, from which you can extrapolate the whole. Discovering a gamma shift after a four-hour 4K render is a particular kind of evening.

Honest limits

Some export failures are genuinely bugs in a specific version on specific hardware, and no amount of reasoning locates them. When you have bisected, switched to software encoding, checked the sources and it still fails, the remaining moves are: update or roll back the application, render in sections and join them, or render a ProRes master and convert with ffmpeg. The last of those works startlingly often, because it takes the editor's encoder out of the chain entirely.

Today

Render a 60-second test from the hardest minute of your current timeline. Open it in two different players and compare the brightness.

BEFORE YOU MOVE ON

An export fails at 43% on every attempt. What does the repeatability tell you, and what do you do?

A. The machine exhausts memory once the render cache fills, so clearing the cache and restarting the application will get past it.

B. Something at that timecode is unreadable, and halving the render range repeatedly will find which clip.

C. The GPU is overheating after a fixed amount of work, which lands at roughly the same progress figure each run.

D. The output has reached a container size limit, which occurs at a fixed number of gigabytes and therefore at the same proportion of any given render.

The answer and why: p. 492

Lesson 76 · 9 min

Checking the file, not the timeline

THE ONE THING TO KEEP

Watch the rendered file once at speed without stopping, then measure its duration, loudness, codec and first and last frame, because every verification you did on the timeline was of something nobody receives.

The timeline is not the deliverable

Everything you have verified so far, you verified on a timeline: the scopes, the loudness meter, the captions, the cut. None of that is what anybody receives. The deliverable is a file, and the file can differ from the timeline in colour, in loudness, in duration, in track layout and in what the first frame is.

So the last act of any job is a check performed **on the rendered file**, opened as a stranger would open it. It takes about twice the running time of the film and it is the highest-value half hour in the process.

Pass one: watch it, as an audience member

Open the file in an ordinary player. Full screen. Do not scrub, do not take notes on the timeline, do not touch anything. Watch the whole thing once, at speed, on the device it is meant for — the phone, if it is a phone.

This is the only opportunity you get to experience the film rather than inspect it, and you will lose it the moment you pause. What you are listening for is whether you understood it, whether you were bored anywhere, and whether the sound carried in the room you were in.

Write down what you felt afterwards, not during.

Pass two: the technical sweep

Now inspect. In order, because the order is roughly cheapest-first.

Figure 8.3

Three passes over the rendered file

Watch it once, at speed
Full screen, no scrubbing, on the kind of device it is meant for

↓

The technical sweep
Duration, first and last frame, loudness measured on the file, ffprobe

↓

The other machine
A phone at half brightness, a laptop's own speakers, a television across a room

Every check before this was performed on a timeline, and nobody receives a timeline. Crushed shadows, dialogue lost on a phone speaker and type that is illegible at arm's length surface here and almost nowhere else.

Duration. Does the file length match the timeline length, to the second? A mismatch means a render range, a missing tail, or a frame rate conversion.

First frame and last frame. Step to frame one. Is it black, or is it the last frame of the slate? Step to the end. Does the film end cleanly, or is there a frame of the next shot, or two seconds of silence you did not intend? Head and tail are the two places nobody looks and everybody notices.

Loudness, measured on the file. Not the timeline meter, which measured what the mix bus was doing, but the exported audio:

```
ffmpeg -i final.mp4 -af loudnorm=print_format=summary -f null -
```

Read `Input Integrated` and `Input True Peak` and compare them with the target for the destination. A render can be quieter than the timeline if a bus was excluded, or louder if a limiter was bypassed on export.

Scrub the joins. Jump to each cut you were unsure about, and to every point where picture and sound change together. Compression artefacts and sync problems both live at transitions.

Check the file's own properties. `ffprobe` or MediaInfo: codec, resolution, frame rate, pixel format, audio channels and sample rate. Compare against what you intended, line by line. This is where you catch the 29.97 preset.

Captions. If burnt in, check they sit inside the safe area on the actual platform, not in your editor's guides. If a sidecar file, open the SRT in a text editor and read the first and last cues, then load it in the player alongside the video and check the timing has not drifted.

Pass three: the other machine

Play it somewhere that is not your edit suite. A phone at half brightness. A laptop with the built-in speakers. A television, if the job is for one.

Three things surface here and almost nowhere else: **shadows that were readable on a graded monitor and are now a black rectangle**, **dialogue that was clear in headphones and is not on a phone speaker**, and **text that was elegant at a metre and is illegible at arm's length**.

The colour block's lesson on legal levels covers the measurement side of this. This is the human side, and they catch different faults.

The checklist to keep

Before anything leaves your machine:

1. Duration matches.
2. Watched start to finish, once, without stopping.
3. First frame and last frame are what you intended.
4. Integrated loudness and true peak measured on the file.
5. Codec, resolution and frame rate confirmed with `ffprobe`.

6. Captions checked in position or the sidecar opened and read.
7. Played on a phone.
8. Filename is the agreed convention, with version and date.
9. The master, textless version and stems are rendered and filed.
10. The music licence and any archive permissions are saved in the project folder.

Items 9 and 10 are not about this delivery. They are about the request that arrives in four months, and they are the two people skip.

When somebody else is checking

If a broadcaster or platform runs an automated quality check, it tests things a human never would: a single illegal luminance value, a frame of black longer than permitted, an audio channel below a silence threshold, a photosensitivity risk from rapid flashing. A rejection comes back as a timecode and a rule number, and it is usually one frame.

That is not an insult to your grade. It is a machine doing exactly what it was asked. Fix the frame, re-render, resubmit.

Honest limits

No checklist catches a film that does not work. The technical pass tells you the file is correct; it says nothing about whether the second act is long, and the two get confused precisely when you are tired and want to be finished. If the first pass told you that you were bored at four minutes, no amount of `ffprobe` addresses it, and the honest move is to go back to the timeline rather than to ship a technically flawless version of something you already know is slack.

Today

Take the last file you delivered. Run the loudness command on it and check the first and last frame. If either surprises you, that is the check earning its place.

BEFORE YOU MOVE ON

The timeline meter read −14 LUFS, but the rendered file measures −19. What is the most likely explanation?

A. Editor loudness meters report peak level rather than integrated loudness, so the timeline figure was never the same quantity.
B. The AAC encoder normalises audio during compression, which lowers the delivered level.
C. The render contained a different signal from the one the meter saw — a muted bus, a bypassed limiter, a different track layout.
D. Integrated loudness is defined over a whole programme, and the timeline meter only accumulates from the playhead onwards, so it never reports a true figure for the film.

The answer and why: p. 493

Lesson 77 · 9 min

The project that still opens in six months

THE ONE THING TO KEEP

Learn JKL and three-point editing, name folders by date and version, and back up the project file separately from the footage.

Two problems, one lesson

The first is speed: most beginners edit with a mouse, dragging clip edges by hand, and are three to five times slower than they need to be. The second is survival: a client emails in March about a video you made in September, and you cannot open it, or you can open it and half the clips are offline.

Both are solved by habits that cost nothing and take an afternoon to adopt.

Learn five keys, not fifty

Shortcut lists are useless because you cannot absorb forty at once. These are the ones that change how the work feels, and they are close to identical across Resolve, Premiere, Avid, Kdenlive and Shotcut.

J, K, L. L plays forward, J plays backward, K stops. Tap L twice for double speed, three times for faster. Hold K and tap L to step forward one frame. Your left hand lives here and you never touch the playhead with the mouse again.

I and O. Mark in, mark out. This is how you select a piece of a clip without dragging anything.

The split key. Cuts the clip at the playhead. In Resolve it is Ctrl/Cmd+B; in Premiere it is Ctrl/Cmd+K. Learn the one for your editor and stop reaching for the blade tool.

That is five. Use only those for a week and your speed roughly doubles.

Three-point editing, which is the real unlock

This is how professional editors actually work and almost no self-taught editor discovers it.

1. In the source viewer, mark **in** and **out** on the piece of the clip you want.
2. In the timeline, mark **in** where it should land.
3. Press **insert** (which pushes everything downstream later) or **overwrite** (which replaces what is there).

Three points define the fourth. You never drag a clip to the timeline and then trim it, which is two operations and a guess. In Resolve, insert and overwrite default to F9 and F10 on the edit page.

Once this is in your hands, an assembly of a one-hour interview takes twenty minutes instead of two hours.

A folder structure that survives

Decide this **before** you import anything, because moving files afterwards breaks every link in the project.

```
2026-03-12_Sharma_ClinicFilm/
  01_footage/
  02_audio/
  03_graphics/
  04_project/
  05_exports/
  06_docs/
  proxies/
```

Date first, in year-month-day, so folders sort chronologically without any effort. Numbered subfolders so they stay in a sensible order rather than alphabetical. Exports named `ClinicFilm_v01.mp4`, `v02`, `v03` — never `final`, never `final_final`, never `FINAL_USE_THIS`. A version number tells you which is newest six months from now; the word "final" tells you nothing and is always a lie.

Put a plain text file in `06_docs` holding the music licence log from the earlier lesson, the client's brief, the delivery specifications, and the passwords-free notes about who said what. That file is what makes a revision request in March survivable.

The Resolve trap that eats projects

Worth stating plainly because it catches people who thought they had backups.

DaVinci Resolve stores projects in an internal database, not as a loose file in your project folder. Copying the whole footage folder to an external drive therefore backs up **none of your edit**. You have to explicitly export the project — File, Export Project, producing a `.drp` — or export the whole database.

Do it at the end of every session. The `.drp` is a few megabytes. Also open Preferences and confirm project backups and auto-save are on, and note where they are being written.

Premiere, Kdenlive and Shotcut do store a project file in the folder, but the same discipline applies: the project file is small, the footage is large, and they need different backup strategies for exactly that reason.

Backups, cheaply

The standard is **3-2-1**: three copies of the data, on two different kinds of media, one of them somewhere else physically. That sounds like a corporate policy. Here is the student version, and it costs the price of one external drive:

- Copy one: your working drive.
- Copy two: an external hard disk you plug in at the end of each week. External drives are cheap per terabyte and irreplaceable footage is not.
- Copy three, off-site: the **project file, the licence log and the exported master** in any free cloud storage. Those are megabytes, not terabytes. You cannot afford to sync 500 GB of footage to the cloud, and you do not need to — footage from a shoot can often be reshot or is already backed up on the camera cards, while the edit represents forty hours of decisions that exist nowhere else.

Keep camera cards unformatted until the edit is delivered, if you can afford the cards. It is the cheapest insurance in this list.

Two speed habits worth the effort

Render cache. When playback stutters on a section with effects, render that region in the background so the timeline plays smoothly and you can judge the timing. It is called render cache or smart cache in Resolve, and preview rendering elsewhere.

Make a template project. Bins already created, timeline at your usual resolution and frame rate, title styles and caption styles saved, export presets set. Duplicate it at the start of each job. Twenty minutes of setup, once, that you never do again.

Today

Export your current project file to the folder next to your footage, then write your file structure down and use it on the next thing you shoot. And spend a week on JKL. Nothing else in this course pays back faster.

Next, if you want to go further: Motion and Audio for titles that move and mixes that hold together, Visual Storytelling for what to point the camera at, and AI Video for what generative tools can and cannot yet do in a real edit.

BEFORE YOU MOVE ON

An editor finishes a Resolve project, copies the entire project folder — footage, audio, graphics, exports — to an external drive, then wipes the laptop. Reinstalling Resolve, the timeline is gone. Why?

A. Resolve keeps projects in an internal database rather than as a file inside the project folder, so copying the folder captured the media but not the edit.
B. The proxies were not copied, so Resolve cannot rebuild the timeline without them.
C. The clips were relinked to new paths on the external drive, and Resolve discards a timeline when every clip goes offline.
D. The project was saved in a newer version of Resolve than the reinstalled copy, so it could not be read.

The answer and why: p. 493

CASE STUDY · MODULE 8

It looks fine here and washed out there

An illustrative case, built from documented patterns.

A two-person production business in Tiruppur delivering a three-minute product film to a knitwear exporter. The editor works on a Mac; the client watches on a Windows laptop in the office.

The grade took most of a day and it is good. Deep blacks in the warehouse, clean whites on the fabric, skin on the two machinists sitting where it should on the vectorscope. The editor, Deepak, exports an H.264 review copy and sends the link.

The reply comes back in forty minutes: the colour has gone flat, the blacks are grey, and it looks worse than the version they saw last week. Could he put the contrast back.

Deepak opens the same file on his own machine and it looks correct. He opens it in VLC and it looks correct. He opens it in his browser and it is slightly lifted. The client's screenshot, taken on the office laptop, shows a noticeably washed-out image with grey blacks.

Two mechanisms could be doing this and he has to tell them apart, because the fixes are opposite.

The first is a **gamma difference**. Rec.709 is defined for a display gamma of 2.4 in a dim room. The web standard is approximately 2.2. A grade that is correct at 2.4 looks slightly washed in a browser, and the browser is applying the standard it is supposed to apply. That difference is real and it is small, and it lives in the shadows.

The second is a **data-level mismatch**, and it produces a large, obvious error rather than a subtle one. A file can be written full range, with 8-bit values running 0 to 255, or video range, with black at 16 and white at 235. If it is written as one and read as the other, everything shifts: video-range read as full crushes, full-range read as video lifts the blacks to grey and dulls the whites. The file carries a flag; some encoders write it wrongly and some players ignore it.

`ffprobe` on the export returns a blank `color_range` field. Nothing was tagged. Every player is guessing, and the client's player is guessing differently from his.

So he has a choice, and the shape of it matters more than this one job.

Regrade to match the client's screen. Add contrast until the screenshot looks right. It would take twenty minutes and the client would be happy today. It would also make the film wrong in VLC, wrong on his own machine, wrong on the exporter's website, and wrong on any player that reads the tag correctly. He would be chasing one player for ever, and the next client would have a different one.

Fix the tagging and prove it. Set the colour tag explicitly on the render rather than leaving it automatic, re-export, run `ffprobe` to confirm the field is populated, and check the result in two different players and a browser. It costs an export and an hour. The client did not ask about tagging and will not care about the explanation.

The second cost is the awkward part. The client's complaint was "put the contrast back", and answering it with a paragraph about transfer functions is a way of being right and being ignored.

YOUR ANSWERS

1. How did the blank `color_range` field distinguish between the two possible causes?

2. Why would regrading to match the client's screen have been the expensive option, given that it was faster?

3. The client asked for contrast and got a file with a colour tag. What made that answer land rather than sound evasive?

STOP HERE

Write your answers before you turn the page. How it played out starts on p. 418.

This page is blank so that how it played out stays out of view until you turn the page.

CASE STUDY · MODULE 8

How it played out

He set the colour tag explicitly on the Deliver page, re-exported, confirmed with `ffprobe` that `color_range` now read `tv`, and checked the file in VLC, in QuickTime and in Chrome on his own machine before sending it. He wrote two sentences to the client: that the first file had been sent without a colour tag so different players were interpreting it differently, and that the new one was tagged and should look the same everywhere. It did. He then added three lines to the studio's checklist: tag the export explicitly rather than leaving it automatic, run `ffprobe` on every finished file before it leaves, and watch every review copy in a second player. The next time a client said the colour looked wrong it turned out to be the office laptop's night-shift setting, which the checklist also now asks about.

The questions, discussed

1. How did the blank `color_range` field distinguish between the two possible causes?

A gamma difference between 2.4 and 2.2 is a small shift in the shadows and would not produce grey blacks in a screenshot. A missing range tag means no player has been told how to interpret the values, so each one guesses, and a mismatch between how the file was written and how it is read produces the large lift the client saw. The blank field was positive evidence for the second mechanism and against the first.

2. Why would regrading to match the client's screen have been the expensive option, given that it was faster?

Because it would have baked a correction for one player's mistake into the film itself. Every player that reads the tag correctly would then have shown an over-contrasted image, including the exporter's own website and any later deliverable derived from the same master. The fault was in the file's metadata, not in the grade, so the only fix that survives is one applied where the fault is.

3. The client asked for contrast and got a file with a colour tag. What made that answer land rather than sound evasive?

That the file demonstrably looked right on the client's own laptop afterwards, and that the explanation was two sentences rather than a lecture. A complaint is a symptom report: the client correctly observed that something was wrong and proposed the wrong cause. Fixing the cause and showing the result addresses the complaint; explaining transfer functions instead would have been accurate and useless.

MODULE 8 TEST

Check what stuck

6 questions, one right answer each. This one is for you, not for a record. The answers, and the reason behind each, are in the key on p. 487. If two go wrong, re-read the module before the exam.

1 Renaming a .mkv file to .mp4 does not make it play. What does a remux do that renaming does not?

A. It rewrites the file's colour tag so that players interpret it correctly
B. It moves the index to the front of the file so playback can start early
C. It re-encodes the video into a codec the container is allowed to hold
D. It unpacks the streams and repacks them into the other box

2 A 1080p25 export at 10 Mbit/s looks clean. The same material at 2160p25 and 10 Mbit/s smears. Why?

A. H.264 is not efficient enough at 4K and needs H.265 to work at all
B. There are four times as many pixels to describe with the same bits
C. 4K files are always 10-bit, and 10-bit needs a higher bitrate
D. The encoder spends most of the bitrate on the extra chroma samples that 4K carries

3 A carefully grained grade turns to porridge after upload. What is the mechanism?

A. Noise is random, so the encoder must spend real bits on it or discard it
B. Grain is added after the colour tag, so it is encoded at video range
C. Grain raises the true peak of the luminance signal beyond what the codec permits
D. The platform applies its own noise reduction before encoding

4 A video looks terrible ten minutes after publishing and fine the next morning, with no re-upload. What happened?

A. The browser cached the preview render and served that until it expired overnight
B. The upload was re-encoded a second time after the first attempt failed
C. Only the low rungs of the ladder had finished processing
D. The platform throttles bitrate until a video passes a view threshold

5 Why is a master rendered textless and at the full frame rather than at the delivery crop?

A. Textless files compress better, so the archive takes less space
B. Baked-in decisions belong to one destination and cannot be removed
C. Burnt-in captions cannot be re-encoded without generation loss
D. Platforms reject any file that already carries burnt-in captions inside it

6 You played the whole programme through the timeline's loudness meter and read minus fourteen integrated. ffmpeg on the exported file reports minus nineteen. What does that tell you?

A. The file was encoded at 44.1 kHz, which lowers the measured loudness
B. The platform's normalisation has already been applied inside the exported file
C. The measurement is unreliable because ffmpeg and the editor use different standards
D. Something in the render path differed from the mix bus

EXAMINATION

Editing Video — final examination

This paper covers the whole course:

- getting footage into a state you can work with
- the cut itself
- structure
- sound
- colour
- text and captions
- effects and repair
- the file that leaves the building

56 questions, the whole pool. The site draws 25 for a sitting, pass mark 70%. Work any 25 under your own conditions. Answers from page 494.

1 Module 1 · Getting the footage into a state you can work with

A green screen was lit evenly and filmed on a phone. The key is still blocky along every edge. What is the limit?

A. The green channel is compressed more heavily than red and blue in H.264 encoding
B. The phone recorded 4:2:0, so the matte is built from quarter-resolution colour
C. The bitrate was too low, so the edges macroblocked before the key was pulled
D. 8-bit capture gives 256 levels, which is too few for a clean matte

2 Module 1 · Getting the footage into a state you can work with

What does shooting 4K for a 1080p delivery actually buy in the edit?

A. A higher bitrate allocation from the platform when the file is uploaded
B. Better colour, because 4K recording modes use 4:2:2 on most consumer cameras
C. A sharper picture, because more pixels always means more visible detail
D. Room: punch-ins, reframes, the stabilisation crop and a vertical version

3 Module 1 · Getting the footage into a state you can work with

Why is the image scaling behaviour in project settings described as expensive to change once an edit is under way?

A. Changing it re-frames every clip whose size does not match the timeline
B. It controls the proxy resolution, so every proxy has to be rebuilt
C. It is locked to the project's colour management setting and moves with it
D. It is stored per clip, so changing it means re-importing the media

4 Module 1 · Getting the footage into a state you can work with

Editing is described as a search problem. What follows practically from that?

A. A transcript replaces the watch pass entirely for anything built on speech
B. Scrubbing is faster than watching, so the logging pass should be done at speed
C. You cannot choose a better shot from material you never formed a memory of
D. Smart bins remove the need for a watch pass, since they gather clips automatically

5 Module 1 · Getting the footage into a state you can work with

For which material does transcript-first editing produce something coherent to listen to and nothing to watch?

A. A lecture recorded in one take with slides
B. A vlog with a presenter talking to camera throughout
C. An interview with two speakers on separate microphones
D. A process film where somebody's hands are the story

6 Module 1 · Getting the footage into a state you can work with

A file will not import into your editor. VLC plays it perfectly. What does that establish?

A. The file is damaged and needs untrunc to rebuild its index
B. The file is intact and the editor lacks the codec
C. The file has a variable frame rate, which VLC tolerates and editors do not
D. The audio stream is in a layout the editor cannot decode, which blocks the import

7 Module 1 · Getting the footage into a state you can work with

Footage sits on a working drive and on a second partition of the same drive. Why is that not two copies?

A. Checksums cannot be computed across two partitions of a single physical disk
B. Partitions share one file table, so a deletion removes both at once
C. One drive failure takes both of them
D. The operating system caches writes, so the second copy may be incomplete

8 Module 2 · The cut itself

Two people are talking and the camera crosses the line between two shots. What does the viewer experience?

A. The scene stops reading as a conversation, because both look the same way
B. The cut reads as a jump, because the two framing sizes do not match
C. The eye loses the subject and re-acquires it, which costs about half a second
D. The audio appears to come from the wrong side unless the mix is corrected too

9 Module 2 · The cut itself

A talking head is cut with thirty jump cuts and the speaker moves a great deal between takes. Why does the convention stop working here?

A. The platform's encoder cannot predict frames across a jump, so it blocks
B. The teleport is large, so the eye complains
C. Jump cuts are acceptable in short-form only, never in long-form video
D. Each jump resets the viewer's retention decision, which shortens watch time

10 Module 2 · The cut itself

If the meaning of a shot is produced by the shot next to it, what does that give an editor?

A. The order of shots matters less than the quality of each one

B. A montage of unrelated images will always read to viewers as an argument

C. Any shot may be placed anywhere, since the context supplies the meaning

D. You can build events you never filmed

11 Module 2 · The cut itself

A politician's promise is cut directly to a shot of a closed factory. What is the honest objection?

A. It asserts a relationship without stating it

B. The factory shot is archive and will need a licence before it can be used

C. Cutting to a static image breaks the rhythm the interview had established

D. The two shots were filmed at different times, which is a continuity error

12 Module 2 · The cut itself

In the working method for cutting a conversation, what happens before any picture decision?

A. Colour matching the two angles so the cut does not draw attention

B. Choosing which angle carries each line of the exchange

C. The radio edit, the breaths, and a bed of room tone

D. Placing the music so the scene has a rhythm to cut against

13 Module 2 · The cut itself

Mirroring a shot horizontally to manufacture a reverse angle fails visibly in which case?

A. Any frame where the camera was handheld rather than locked off
B. Any frame recorded in 4:2:0, because the chroma is sampled asymmetrically
C. Any frame that has been stabilised, because the crop is not symmetrical
D. Any frame containing text, a watch, a hair parting or a door handle

14 Module 2 · The cut itself

What distinguishes a cold open that is craft from one that is bait?

A. Whether it runs under twelve seconds
B. Whether the promise it makes gets paid
C. Whether it is graded to match the rest of the piece
D. Whether a title card follows it within the first thirty seconds of the piece

15 Module 3 · Structure — deciding what the film is

Why is a first fine cut reliably a fifth to a third too long?

A. Clients ask for more content than the agreed running time allows
B. The assembly was two to four times the target and the trimming was never finished
C. Material justified by the editor's knowledge feels slow to a stranger
D. Most editors trim frames rather than removing whole sections

16 Module 3 · Structure — deciding what the film is

What is the actual job of a timeline called parked or offcuts?

A. It removes the finality that makes people fight to keep things
B. It stores the graded versions of shots that were cut from the film
C. It satisfies broadcast delivery requirements for unused material and rushes
D. It keeps a backup in case the project file is corrupted

17 Module 3 · Structure — deciding what the film is

Why is narration written after the picture edit rather than before it?

A. A scratch track cannot be recorded until the cut is locked
B. Written first, it gets illustrated, and the film becomes a slideshow
C. The running time is unknown until the picture is finished
D. A voice-over recorded early will not match the room tone of the finished mix

18 Module 3 · Structure — deciding what the film is

Over a shot of workers arriving at a factory, which line is doing the job narration exists for?

A. The factory opens at six every morning, as it has for decades
B. The workers make their way in through the main gate as the shift begins
C. Here we see the workers arriving at the factory
D. Four years ago there were nine hundred of them

19 Module 3 · Structure — deciding what the film is

Why is a thirty-second vertical cut not simply a trimmed version of the long piece?

A. The short has no context to lean on and has to be complete in itself
B. Vertical delivery needs a different frame rate and a lower bitrate
C. The long piece is graded and the short is derived from the camera original
D. Platforms reject a vertical file that has been cropped from a horizontal timeline

20 Module 3 · Structure — deciding what the film is

A track licensed CC-BY-NC is used under a video advertising the editor's own services. Why is that a problem?

A. Content ID will claim the track regardless of the licence, which blocks the video
B. NC licences require attribution in a specific form that most people omit
C. Advertising your own services is a commercial use
D. NC licences expire after a year unless renewed with the rights holder

21 Module 3 · Structure — deciding what the film is

Somebody agreed to appear in an internal training video. The same interview is now wanted for a public advertisement. What follows?

A. A credit in the end titles discharges the obligation
B. It is permitted provided the person is not identified by name on screen
C. The original release covers it, because the footage is unchanged
D. The scope has changed, so it needs a new yes

22 Module 4 · Sound, which is most of it

What does the phrase royalty free actually mean on a music library?

A. The track may be used in any project without payment
B. There is no recurring per-play royalty
C. The composer has waived their moral rights in the work
D. The track is registered with Content ID and will therefore not be claimed

23 Module 4 · Sound, which is most of it

Beethoven died in 1827. Why can a 2019 orchestral recording of his Fifth still be somebody's copyright?

A. Copyright in a composition is renewed whenever it is performed in public
B. Public domain applies only in the country where the composer died, not elsewhere
C. The recording is a separate work, with its own performers and label
D. Orchestras register their arrangements as new compositions each time

24 Module 4 · Sound, which is most of it

Why can a film not be dubbed into another language from a flattened mix?

A. The dialogue cannot be separated from the music and effects
B. The flattened mix is normalised, so new dialogue will not match its level
C. Dubbing requires the original timecode, which a mix does not carry
D. Lip sync in the new recording cannot be checked against a stereo file

25 Module 4 · Sound, which is most of it

On a compressor set for dialogue, what should the gain reduction meter be doing?

A. Staying still, because a correctly set compressor never needs to act at all
B. Moving 3 to 6 dB on loud words and back to zero between phrases
C. Sitting steadily at about 10 dB through the whole take
D. Moving only on the sibilants, by about 3 dB each time

26 Module 4 · Sound, which is most of it

A re-recorded line sounds pasted into the scene. What is matched first?

A. The EQ, to match the resonances the room added to the original
B. The level, so the replacement peaks at the same value as the surrounding lines
C. The reverb, to match the decay of the original room
D. The noise floor, by laying the scene's room tone under it

27 Module 4 · Sound, which is most of it

Why is narration over location footage deliberately not matched to the room in the picture?

A. A narrator is understood to be outside the scene
B. Convolution reverb cannot be applied to a mono voice track
C. The location's impulse response was recorded after the shoot rather than during it
D. Reverb would push the integrated loudness above the destination's target

28 Module 4 · Sound, which is most of it

You turn the mix down until it is barely audible and the dialogue disappears before the music does. What does that tell you?

A. The loudness range is too wide for the destination's delivery specification
B. The dialogue needs more compression to survive at low level
C. The balance is wrong
D. The music is in stereo and the dialogue is in mono

29 Module 5 · Colour, correction before grade

Why does putting a heavy contrast node before the balance node change what the balance node is able to do?

A. Contrast nodes are computed last regardless of their position in the graph
B. Balance nodes read the source clip rather than the previous node's output
C. The graph is evaluated in parallel, so the order of two nodes is undefined
D. A node works on what it receives, and the shadows are already crushed

30 Module 5 · Colour, correction before grade

What makes the still store the feature that turns shot matching from an hour into five minutes?

A. It stores a frame that can be exported as a reference for the client
B. It carries the whole node tree, and it wipes against another shot
C. It analyses two shots automatically and computes the difference between them
D. It saves each shot's grade as a version that can be recalled from the timeline

31 Module 5 · Colour, correction before grade

Matching a shot to a graded hero shot, what is compared first?

A. The overall colour balance, on the RGB parade display
B. The saturation, on the vectorscope
C. The black point, on the waveform
D. The skin tone against the reference line

32 Module 5 · Colour, correction before grade

An automatic shot match pulls a close-up of a face green when matched to a wide of a green field. Why?

A. It matches overall distributions rather than the subject
B. The tool assumes both clips came from the same camera and profile
C. The face occupies too little of the frame for the analysis to find any skin tones
D. The wide was filmed at a different colour temperature from the close-up

33 Module 5 · Colour, correction before grade

What is the mark of a good power window?

A. Its edge is hard enough to contain the correction precisely
B. The viewer never notices it
C. It tracks without a single keyframe being placed by hand
D. It is drawn tightly around the subject so that nothing else is affected

34 Module 5 · Colour, correction before grade

A flat horizontal line along the top of the waveform means what?

A. The white balance is correct, because all three channels agree
B. The file is tagged video range and is being displayed as full range instead
C. The image has been graded within broadcast-safe limits
D. A region of the picture is clipped and holds no detail

35 Module 5 · Colour, correction before grade

Flat, evenly lit footage is graded toward a moody film look and comes back dark and murky rather than dramatic. Why?

A. Contrast between light and shadow has to be in the room
B. The footage was 8-bit, so the shadows banded when they were pushed
C. The grade was applied before the shots had been matched to one another
D. The LUT expected a log image and received a Rec.709 one

36 Module 6 · Text, captions and graphics

When should a name super appear?

A. At the end of their answer, so that it does not compete with what they are saying
B. On the person's first appearance on screen
C. Two or three seconds after they start speaking
D. At the same moment their first line begins

37 Module 6 · Text, captions and graphics

The second line of a name super should carry what?

A. The organisation they work for and its location
B. The date and the place at which the interview was recorded
C. The person's full job title and department
D. Why their view matters

38 Module 6 · Text, captions and graphics

Why are credits sometimes a contractual obligation rather than a courtesy?

A. Platforms require a credit roll on any video longer than five minutes
B. Some licences specify an attribution string as a condition of use
C. Contributors hold moral rights that only an on-screen credit can satisfy
D. A film without credits cannot be entered into most festival programmes

39 Module 6 · Text, captions and graphics

A speed ramp built in the Retime Speed view jolts at every step. What fixes it?

A. Adding motion blur across the ramp so the change is less visible to the eye
B. Reducing the number of keyframes so there are fewer steps to see
C. Easing the keyframes on the Retime Frame graph instead
D. Turning on optical flow so the intermediate frames are smoother

40 Module 6 · Text, captions and graphics

Why render a Blender title to a PNG image sequence rather than straight to a video file?

A. A crash at frame 800 leaves the first 799 intact
B. Image sequences import faster into a video editor's timeline
C. PNG is lossless while every video codec used for compositing is not
D. Video files cannot carry an alpha channel at all

41 Module 6 · Text, captions and graphics

Why record a screen at the delivery resolution rather than at the display's full resolution?

A. Most editors will not import a screen recording larger than 1080p without a transcode
B. Capturing 4K and downscaling halves the size of every interface element
C. Screen recorders write variable frame rate at higher resolutions only
D. A 4K capture cannot be punched into without losing sharpness

42 Module 6 · Text, captions and graphics

Why is a truncated bar-chart axis a stronger deception on video than on a page?

A. Animated bars grow from the baseline, which exaggerates the difference further
B. A chart on screen for four seconds is usually shown without any axis at all
C. Compression removes the axis labels before the viewer can read them
D. The viewer cannot go back and check the axis

43 Module 7 · Effects, compositing and repair

A carefully rotoscoped arm moving quickly still looks cut out. Why?

A. The mask edge is hard, and a moving limb's real edge is semi-transparent
B. The mask was keyframed too sparsely, so it drifts between keyframes
C. The feather was set too low for the resolution of the footage
D. The shot was denoised after masking rather than before, so the edge crawls

44 Module 7 · Effects, compositing and repair

Filming five seconds of the empty set at the end of every setup enables which technique?

A. A planar track of the background surface
B. A luminance key pulled from the brightest part of the frame
C. A difference matte against a clean plate
D. An optical flow estimate with fewer spurious vectors

45 Module 7 · Effects, compositing and repair

Why does a tracker slide along a straight edge and lock onto a corner?

A. Edges move faster than corners when the camera pans across them
B. A corner contains more pixels inside the tracker's search region than an edge does
C. An edge carries more contrast than a corner, which confuses the search
D. Along an edge, many positions match equally well

46 Module 7 · Effects, compositing and repair

A pan from a tripod cannot be solved for camera position. Why not?

A. The scene is not rigid enough, because a pan reveals trees and traffic
B. There is no parallax: the camera never moves through space
C. The rotation is too fast for the solver's search window to follow
D. A tripod introduces vibration that the solver reads as real movement

47 Module 7 · Effects, compositing and repair

You ask for a one-second dissolve and the editor quietly makes it shorter. Why?

A. Dissolves are capped at twelve frames in most editors by default
B. One of the two clips is a compound clip, which cannot be dissolved across
C. The clips had no handles beyond their in and out points
D. The timeline frame rate does not divide the requested duration evenly

48 Module 7 · Effects, compositing and repair

Playback has dropped to eight frames a second. What is checked first?

A. Whether proxies exist and the toggle to use them is on
B. Whether the GPU has enough video memory for the timeline resolution
C. Whether a background render or cache generation is currently running
D. Whether the disk can sustain the data rate the timeline needs

49 Module 7 · Effects, compositing and repair

A locked-off shot of a street has people walking through it and you want the street empty. Which technique?

A. A difference matte built against a clean plate filmed on the day
B. A per-pixel median across many frames
C. A tracked patch taken from another frame of the shot
D. Machine inpainting of a mask drawn around each person

50 Module 8 · Delivery — the file that leaves the building

A file plays in VLC and refuses to play in Safari and most browsers. Which setting is the first suspect?

A. The audio codec, which should be PCM rather than AAC
B. The colour range flag, which should be set to full rather than video
C. The container, which should be MOV rather than MP4
D. The pixel format, which should be yuv420p

51 Module 8 · Delivery — the file that leaves the building

A video on your own web page shows a long spinner and then plays normally. What fixes it?

A. Moving the index to the front of the file

B. Converting the file from H.265 to H.264 for browser support

C. Re-encoding with two-pass VBR so that the final file size is predictable

D. Raising the bitrate so the first seconds load faster

52 Module 8 · Delivery — the file that leaves the building

ffmpeg -ss 00:01:30 -i in.mp4 -t 60 -c copy out.mp4 starts about a second later than asked. Why?

A. The audio and video streams carry different start timestamps inside the file

B. The -ss flag is placed before -i, which seeks inaccurately by design

C. With -c copy the cut has to land on a keyframe

D. The container's timebase rounds the start point to the nearest second

53 Module 8 · Delivery — the file that leaves the building

A render fails at the same percentage every time. What is the efficient diagnosis?

A. Rebuild the cache and clear the optimised media for the project

B. Render in sections and join them afterwards with a stream copy in ffmpeg

C. Switch to software encoding and run the whole render again

D. Bisect the render range until one clip is isolated

54 Module 8 · Delivery — the file that leaves the building

A 25p timeline exported through a preset named YouTube 1080p judders on every pan. What happened?

A. Hardware encoding was enabled, which drops frames on long pans by design
B. The preset carried 29.97, so frames were duplicated
C. The bitrate in the preset was too low for the motion in the piece
D. The preset upscaled the timeline, and the scaler softened the movement

55 Module 8 · Delivery — the file that leaves the building

Somebody copied their whole footage folder to an external drive and believes their Resolve edit is backed up. What is wrong?

A. Copying media breaks the relative paths the project uses to find its clips
B. The proxies were not copied, so the timeline will come back offline
C. Resolve keeps projects in a database, not in the folder
D. The cache folder holds the render decisions and was not included

56 Module 8 · Delivery — the file that leaves the building

What makes three-point editing faster than dragging a clip onto the timeline and trimming it?

A. Three marks define the fourth, so there is no guess to correct
B. Insert and overwrite are rendered faster than a drag operation
C. It places the clip at the playhead, which removes any need for snapping
D. It bypasses the source viewer, so fewer windows are involved

Answers

The key is grouped by module. Each entry gives the page of the question, the question cut short, the right answer and why. For a lesson check it also says why each wrong option tempts. That part is the teaching, not the marking.

Module 1 · Getting the footage into a state you can work with

The module starts on p. 11.

MODULE 1 TEST

Module 1 test, Q1, p. 58
The same 4K phone clip plays at three frames a second from the original file and smoothly from a quarter-size proxy. What is the mechanism?
Answer C: The camera file stores frames as differences, so each one must be rebuilt
Why: H.264 and H.265 are long-GOP formats: they store a whole frame once every one to two seconds and describe everything between as differences. To display the frame at four minutes twelve, the decoder finds the last whole frame and reconstructs every frame since. Scrubbing makes it do that continuously. A proxy in an intra-frame codec stores every frame complete, so the frame you asked for is simply there.

Module 1 test, Q2, p. 58
A camera recorded 8-bit 4:2:0 in a log profile. The shot is corrected and contrast is added back. What appears?
Answer A: Banding in skies and smooth gradients
Why: Log spreads a wide brightness range thinly across whatever levels the file has. At 8-bit there are 256 per channel, so adding the contrast back stretches the gaps between levels until they are visible as steps. Macroblocking comes from a starved bitrate and fringing from 4:2:0 chroma; both are real faults with different causes. This is why log is a 10-bit feature, and why a well-exposed standard profile beats log on an 8-bit camera.

Module 1 test, Q3, p. 58
Filming at 30 fps indoors on a 50 Hz supply produces bands crawling up the frame. The same camera outdoors shows nothing. Why?
Answer D: Daylight does not pulse, so every frame catches the same amount of light
Why: Fluorescent tubes and most cheap LED panels pulse with the mains, twice per cycle, so 100 times a second on a 50 Hz supply. At 30 fps the shutter interval does not divide evenly into that pulse, so each frame catches a slightly different amount of light and the difference reads as a beat or as bands. The sun does not pulse, so outdoors there is nothing to beat against. Either match the frame rate to the mains, or set a shutter that is a whole multiple of the flicker period.

Module 1 test, Q4, p. 59
A card was copied by dragging it to a drive. Every file appeared and the folder sizes matched. Three weeks later one clip is green...
Answer B: That the write completed, not that the copy matches the source
Why: A plain copy writes bytes and reports that the write finished. Nothing reads the destination back. A failing cable, reader or card can therefore produce a file of the right size containing the wrong bytes, and the failure is silent until you need the clip. A checksum fingerprints the source and the copy and compares them, which is what rsync --checksum does during the copy and what a stored manifest lets you repeat months later.

Module 1 test, Q5, p. 59

A phone clip is in sync at the clap and half a second late by minute ten. What is the usual cause, and what fixes…

Answer A: A variable frame rate file; transcode it to constant rate before importing

Why: Phones and screen recorders often write variable frame rate: the file claims 30 fps while the actual interval between frames wanders. The editor assumes a constant rate, so picture and sound diverge steadily rather than sitting at a fixed offset. Compare r_frame_rate with avg_frame_rate in ffprobe; if they differ, transcode the whole shoot to constant frame rate at offload. Clock drift between two devices does the same thing more slowly and is fixed by conforming one clip's duration.

Module 1 test, Q6, p. 59

In DaVinci Resolve, why is the timeline frame rate described as a setting you cannot change later?

Answer C: It greys out as soon as any media exists in the project

Why: The frame rate is a project setting, not a timeline setting, and it locks the moment the first clip lands in the project. After that the honest routes are a new project at the right rate or a new timeline inside the old one, which leaves you converting anyway. Re-framing every scaled clip is true of timeline resolution, which is changeable and expensive for a different reason: every punch-in you set relative to the old frame moves.

THE LESSON CHECKS

Lesson 1, p. 16

An editor working on a laptop enables proxy media, and playback becomes smooth. A colleague warns that the final video will now be low resolution…

Answer A: Proxies are only used during playback; at export the editor relinks to the original camera files, so output resolution and quality are unchanged.

Why: Camera files store frames as differences, not pictures, so edit a proxy and let the export go back to the original.

B The colleague is right, and the…: Assumes proxy state at export is a manual step the editor must remember rather than something the software resolves.

C Quality drops slightly, because the colour…: Confuses the reduced preview image with the data the adjustments are actually applied to at render time.

D It depends on the proxy resolution…: Treats proxy resolution as something that propagates into the render instead of a preview-only setting.

Lesson 2, p. 21

An editor is handed 4K phone footage of a presenter in front of a green sheet. The lighting was even and the sheet is clean...

Answer B: Phone footage is 4:2:0, so the colour information the keyer builds its matte from exists at half the horizontal and half the vertical resolution of the picture.

Why: A camera file carries four fixed decisions — whole frames or differences, how much colour was kept, how many brightness levels, and how many bits per second — and each one sets a hard limit on what the edit can do later.

A The bitrate is too low, so...: Right family of problem, impossible fix: re-encoding cannot add back detail that was never written, and a higher target bitrate on a lossy source only preserves the existing artefacts more faithfully.

C The clip is long-GOP, so frames...: Names a real property that causes playback and re-encode problems, but the blockiness of a matte follows from chroma resolution, not from inter-frame prediction.

D The green sheet is the wrong...: Treats a resolution limit as a sampling choice; picking a better key colour improves separation but cannot raise the resolution of the colour channels.

Lesson 3, p. 25

A creator in Delhi shoots indoors at 30 fps under LED panels and gets horizontal bands crawling up the frame. The same camera outdoors in...

Answer D: The LEDs pulse at 100 Hz on 50 Hz mains, which does not divide evenly into a 30 fps frame interval; shooting 25 fps or setting shutter to 1/50 or 1/100 cancels it.

Why: Pick one frame rate from the mains frequency where you shoot, set every camera and the timeline to it, and keep shutter at roughly 180 degrees — because mismatched rates cost you frames at every conversion and mismatched shutter costs you flicker you cannot remove.

A The sensor is overheating under continuous...: Attributes a lighting-interference pattern to a thermal fault, which produces noise and shutdowns rather than regular moving bands.

B The LED panels are too dim...: Confuses noise with a periodic artefact — gain noise is random per frame, while flicker bands are regular and move at a steady rate.

C The camera's rolling shutter is scanning...: Rolling shutter does make the bands appear as bands rather than as whole-frame pulsing, but the cause is the light's pulse rate, and no post-process restores brightness that was never recorded.

Lesson 4, p. 29

An editor has a project where 80% of the material is 1080p and 20% is 4K. They set the timeline to 4K so as not…

Answer B: Committed the majority of the film to a 200% upscale, and taken on a heavier timeline, for the benefit of a fifth of the shots.

Why: Shoot larger than you deliver so the edit has room to punch in, stabilise and reframe, set one image-scaling behaviour at the start of the project, and keep everything that must be read inside 90 per cent of the frame.

A Made a reasonable trade: the 4K…: Misses that the 1080p clips are not left alone — on a 4K timeline they are upscaled 200% for their whole duration, which is invented detail, not neutral handling.

C Ruined the 4K clips, because a…: Overstates the harm: a 4K clip on a 4K timeline is a one-to-one map and is not resampled at all.

D Made no difference to the picture…: Assumes a final downscale undoes an intermediate upscale, when the upscale has already replaced real detail with interpolated detail that the downscale then averages.

Lesson 5, p. 32

Halfway through grading a documentary in Resolve, an editor switches the project from DaVinci YRGB to DaVinci YRGB Color Managed because a tutorial recommended it…

Answer B: Every existing grade was built on a differently transformed starting image, so the grades are still applied but are now landing on inputs they were not made for.

Why: Frame rate, image scaling and colour management are decided before the first import and are expensive to reverse, so configure a template project once and duplicate it rather than choosing settings under time pressure.

A Nothing but a re-render of the…: Treats colour management as a viewer preference; it changes the image each node receives, so it is upstream of every grade rather than downstream of them.

C The grades are deleted, because switching…: Overstates the damage: nodes and their settings survive the switch, which is precisely why the result is confusing rather than obviously broken.

D Only the log clips in the…: Assumes management is a targeted fix for log footage, when it converts everything into a working space including material that was already in the timeline space.

Lesson 6, p. 36

A shooter copies a card by dragging it to an external drive, confirms the folder sizes match exactly, then formats the card to keep shooting…

Answer A: Bytes may have been corrupted in transit while the file length stayed correct, and there is now only one copy, so nothing can be compared against.

Why: A drag-and-drop copy reports success without reading anything back, so verify irreplaceable footage with a checksum manifest at offload and keep the material in two places on two devices before any card is formatted.

B None in practice — if every…: Equates size with content: a file can be written at the right length with corrupted bytes inside, which is precisely the failure a size comparison cannot see.

C The drive might not have finished…: Names a real phenomenon that a proper eject already handles, and which would show as missing or truncated files rather than as silent corruption.

D The camera's sidecar metadata folders were…: Identifies a genuine and common mistake, but a folder-level size comparison would have caught missing sidecar files, whereas it cannot catch altered content.

Lesson 7, p. 40

Two editors have the same three-hour interview. One builds a selects timeline of forty minutes before cutting; the other starts assembling directly from the bins…

Answer C: Because every timeline decision is really a question about what else exists, and the second editor has to answer it by scrubbing rather than by remembering.

Why: Watch everything once before cutting anything, marking moments with written notes and building a selects timeline, because editing is a search problem and you cannot search material you never formed a memory of.

A Because the selects timeline is shorter…: Mistakes an organisational advantage for a performance one; playback speed is not what makes assembly slow on a three-hour interview.

B Because a selects timeline removes the…: Overstates what a string-out does — nothing is deleted, and the alternatives remain available, which is part of why the method is safe to use.

D Because assembling directly from the bins…: Invents a technical difference; both approaches reference the same source files in exactly the same way.

Lesson 8, p. 44

An interview is shot on a phone with a separate audio recorder. The clap syncs perfectly, but by minute eight the lips lead the sound…

Answer B: The phone recorded variable frame rate, so the editor's constant-rate assumption makes picture and sound diverge steadily.

Why: Give every take one sharp sync point at the start, use waveform sync as a first pass and check it at a hard consonant, and convert variable-frame-rate phone footage to constant rate at offload because drift is usually the file, not the clocks.

A The recorder's clock drifted; speed-stretching the…: A real cause with the right fix, but crystal drift over eight minutes is typically a few frames, not twelve — the size of the error points elsewhere.

C The audio was recorded at 44.1…: Would produce drift, but a 44.1-in-48 mismatch is an 8.8% error — the sound would be obviously fast and the pitch obviously raised within seconds.

D The waveform sync tool locked onto…: A mis-locked sync is wrong from the first frame, whereas this error starts at zero and grows, which is the signature of a rate problem rather than an offset one.

Lesson 9, p. 48

An editor builds a paper edit from an automatic transcript of an accented interview and cuts the timeline straight from it. What is the specific…

Answer B: Recognition errors are confident and grammatical, so a misheard sentence can cause the editor to discard material that actually said what they needed.

Why: For speech-led work, transcribe everything and build the structure in text with timecodes attached, because reading is four times faster than listening and rearranging a paragraph costs a keystroke where rearranging a timeline costs a minute.

A The transcript's timecodes will be slightly…: A real nuisance that fine trimming fixes; it changes the polish of the cut rather than which material ends up in it.

C Automatic transcripts omit filler words, so…: Most systems include fillers, and removing them is usually the goal — this describes a convenience, not a structural risk.

D The model will quietly normalise the…: Recognition does tend toward standard forms, but the consequence is a mismatched caption track rather than a wrong editorial decision about which lines exist.

Lesson 10, p. 52

Lesson 10, p. 52

A clip imports and plays, but at minute six the audio is clearly ahead of the picture, and the gap grew steadily from zero. `ffprobe`...

Answer C: The nominal and actual rates disagree, so the file is variable frame rate and the editor's constant-rate assumption is accumulating error.

Why: Most broken footage falls into four fixable classes — variable frame rate, an unsupported codec, a missing index, and a mis-mapped audio channel — and the ones that are genuinely unrecoverable are clipped highlights, clipped audio and missed focus, which are worth naming to a client on day one.

A The file is damaged and needs...: Confuses a correctly written variable-rate file with a corrupt one; an unindexed file typically will not import or seek at all rather than drifting smoothly.

B The clip was recorded at 29.97...: A real and separate problem, but 29.97 would read as `30000/1001`, and a fixed-rate mismatch produces a constant ratio rather than the wandering average shown here.

D The audio stream is at the...: Sample-rate mismatch does cause drift, but it would show in the audio stream's parameters, and would not explain two disagreeing video rate fields.

Module 2 · The cut itself

The module starts on p. 61.

MODULE 2 TEST

Module 2 test, Q1, p. 110
Someone reaches for a cup. The movement takes about a second and a half. Why is holding the shot to the end of the movement…

Answer B: The viewer understood 'reaching for a cup' within a few frames

Why: Recognition is much faster than completion. The shot delivers its information almost immediately and then continues for the length of the physical action, so roughly a second of it is doing nothing. Do that at forty cut points and you have added most of a minute of nothing to the film. The editor cannot feel it, because they already know what comes next; the viewer is not being patient on anyone's behalf.

Module 2 test, Q2, p. 110
Cutting on action, editors start the incoming close shot two or three frames earlier than where the wide left off, so the movement slightly repeats…

Answer D: A real body decelerates, so an exact frame match reads as a small stutter

Why: A person performing the same action twice does not accelerate and decelerate identically, and the second camera never quite matches the first. Cutting at the mathematically identical frame therefore tends to read as a tiny hitch. Overlapping by a few frames means the eye is busy following the motion at exactly the moment the join happens, and the join disappears. Matching the subject's position in the frame across the cut does the rest.

Module 2 test, Q3, p. 111
A dialogue scene where every picture cut lands on the same frame as its audio cut sounds like a slideshow. What does moving the audio…

Answer C: Sound runs continuously across the join, so the cut has nothing to announce it

Why: In a J cut the incoming sound starts before its picture; in an L cut the outgoing sound continues over the incoming picture. Either way the audio does not change at the moment the picture does, so there is no simultaneous discontinuity for the viewer to notice. Unlinking audio and video is the mechanical step that lets you do it, not the reason it works.

Module 2 test, Q4, p. 111
A cutaway is the right length and the right moment is inside it, but it should arrive half a second later. Music is already cut…

Answer A: Slide

Why: A slide moves a clip along the timeline while the clips either side absorb the change: the one before gets longer, the one after gets shorter. The clip's own content and duration are untouched and nothing downstream moves, so the music stays where you put it. A ripple would shift everything after it and break the music; a roll moves a join rather than a clip; a slip changes which part of the clip you see without moving it.

Module 2 test, Q5, p. 111

A cut feels wrong and you cannot say why. Murch's ranking suggests where the answer usually is. Where?

Answer D: In the top two, emotion and story

Why: Murch weighted emotion at about half the value of a cut and story at about a quarter, leaving the four continuity criteria to share the rest. The practical consequence is that four times out of five a cut that feels wrong is landing on the wrong beat or failing to advance anything, while you are inspecting the geography. Walk the list from the bottom because the lower ones are visible, but expect the answer at the top.

Module 2 test, Q6, p. 112

Why does a six-minute interview film need roughly twelve usable cutaways rather than two?

Answer B: Because a cutaway is what lets time be removed without a visible join

Why: Every removal from a single-camera interview is a jump cut unless something covers it. Two seconds of hands, the room or the object lets three minutes disappear invisibly, and also covers a cough or a fluffed line and breaks up a single face. About one per thirty seconds of finished film is what makes a forty-minute interview compress to six minutes without the compression becoming the thing the viewer watches. They can repeat, just not next to each other.

THE LESSON CHECKS

Lesson 11, p. 65

An editor removes eight frames from the end of every shot in a rough cut. Most cuts improve, but three feel abrupt and wrong. What...

Answer C: Those shots were still delivering something in their final frames — a reaction completing, a movement resolving — so the information was cut off rather than the dead air.

Why: Trim every shot to the frame where it stops giving new information, and cut to the reaction more often than you think.

A The frame rate is wrong on...: Reaches for a technical setting when the effect described is about content, not timing arithmetic.

B Shortening always damages a cut, and...: Treats pace as an illusion rather than accepting that most shots genuinely run past their information.

D Those three cuts needed a dissolve...: Assumes a transition effect can substitute for a shot that no longer contains what the cut depended on.

Lesson 12, p. 69

A vlogger removes every pause from a piece to camera, producing forty jump cuts. Viewers say it feels frantic and claustrophobic. The vlogger adds a…

Answer D: The complaint is about the framing never changing and the space never being established, which a transition cannot supply; varying the shot size and adding cutaways is what the sequence lacks.

Why: Cut on movement, keep the camera on one side of the line, and let sound cross the join before the picture does.

A Cross-dissolves are only correct between different…: Treats dissolve placement as a rule violation rather than asking what the viewer was actually reacting to.

B The dissolves reduce perceived sharpness, and…: Attributes a structural complaint about space and rhythm to an image-quality side effect.

C Dissolves increase total running time, which…: Assumes the problem was length, when the viewers described texture and confinement rather than pace.

Lesson 13, p. 74

An editor has cut a montage to music, with every shot landing on a beat. One shot in the middle shows the subject's hand entering…

Answer B: Slip the clip six frames, which changes which part of the source is shown while its position and duration stay exactly as they are.

Why: Four named operations move a cut — ripple changes duration and shifts everything downstream, roll moves the join only, slip changes the content inside a clip, slide changes its position — so decide which question you are asking before you touch the edge.

A Ripple trim six frames off the…: Fixes the content but shortens the clip, so every later shot moves six frames earlier and the whole montage falls off the beat.

C Roll the incoming edit six frames…: Roll moves the join, so the previous shot gets six frames longer and its own beat relationship breaks instead.

D Slide the clip six frames earlier…: Moves the clip's position, so the shot no longer starts on its beat even though the neighbours keep the total duration.

Lesson 14, p. 79

In a cut scene, a subject's glass is half full in the wide and full in the close-up, and the cut lands exactly on the...

Answer D: Keep the cut, because the emotional beat outranks the physical continuity and moving it trades the most valuable property of the cut for the least.

Why: A cut can satisfy six things — emotion, story, rhythm, eye-trace, screen plane and physical space — and when they conflict, the audience forgives a broken geography far more readily than a wrong emotional beat.

A Move it, because continuity errors accumulate...: Correctly notes that errors accumulate, then acts on it in the one case where the cost is highest — trading the scene's strongest moment for a detail almost nobody registers.

B Move it and fix the emotion...: Treats music as a repair for structure; a score can amplify a beat that exists but cannot relocate the moment the audience was given to feel it.

C Cut away to an insert of...: A real technique, made wrong by 'always' — inserting a shot at the emotional turn removes the audience from the face at exactly the moment they need to be on it.

Lesson 15, p. 83

A tutorial is recut at short-form pace — a visual change every one to two seconds — and its retention graph improves markedly, but viewers...

Answer B: Retention measures whether the frame kept changing, while comprehension needs a conceptual unit to stay on screen long enough to be processed, and the recut shortened both together.

Why: Average shot length gives you a number for pace, but rhythm lives in the variation around it — put long shots where the viewer should settle and short ones where they should be pushed, and cut speech at the breath rather than inside the phrase.

A The faster cuts introduced continuity errors...: Assumes the damage is in the joins; the comprehension loss happens even when every cut is technically clean, because the problem is the interval rather than the match.

C Retention graphs are unreliable on short...: Dismisses the measurement instead of explaining the conflict, and the retention gain from frequent visual change is a well-observed effect rather than an artefact.

D The audio has been cut to...: Describes a real side effect of aggressive cutting, but viewers reporting they could not follow the argument is a comprehension failure rather than a complaint about delivery.

Lesson 16, p. 87

An editor needs to compress a two-minute process into twenty seconds but only has one continuous wide shot of it. Why does adding a close...

Answer D: Because cutting to the insert removes the viewer from the continuous action, so any amount of elapsed time can pass behind it without a visible discontinuity.

Why: A sequence is built out of fragments the viewer stitches into a continuous action, so shoot every action whole from three distances and join the shots with 'but' and 'therefore' rather than 'and then'.

A Because the insert is higher resolution...: Invents a technical difference; the insert is usually the same format, and resolution is not what makes a time cut acceptable.

B Because speeding a shot up introduces...: Names a real artefact of retiming, but a clean speed-up on a locked-off shot can be artefact-free and still reads unmistakably as fast motion.

C Because an insert re-establishes the geography...: Reverses the roles: a close insert removes geography rather than supplying it, and it is a wide shot that re-establishes space.

Lesson 17, p. 91

An editor cuts an interview, places every picture cut exactly on its corresponding audio cut, and the result feels mechanical despite each individual cut being...

Answer C: Offset the picture edges from the audio edges, so sound is continuous across each join and the cut has nothing to announce itself with.

Why: Cut a conversation as audio first, move every join to a breath, lay room tone underneath, and only then decide what the viewer looks at — which is the listener at least as often as the speaker.

A Add a short cross-dissolve at every...: Softens the symptom at the cost of the cut, and a dissolve at every join in a dialogue scene announces itself far more than the hard cuts did.

B Lengthen every shot slightly so the...: Addresses pace when the complaint is about coincidence — the same rate of cutting feels fluid as soon as picture and sound stop landing together.

D Cut to the listener rather than...: Good advice about shot choice that does not touch the cause: cutting to a different face still lands picture and sound on the same frame.

Lesson 18, p. 95

An editor needs to remove ninety seconds from the middle of a single-camera interview. They have a two-second insert of the subject's hands. Why does...

Answer A: The insert removes the subject from frame, so the viewer has no reference to notice the discontinuity against, whereas a dissolve keeps the same face visibly changing position.

Why: Cutaways and inserts are not decoration — they are the licence to remove time, hide a mistake and change rhythm, and an interview needs roughly one usable one for every thirty seconds of finished film.

B The insert restores continuity of motion...: Misstates the mechanism: the insert has nothing to do with matching the subject's motion, and dissolves blend moving images rather than static ones.

C A dissolve reduces image quality at...: Introduces an export concern that is invisible at normal bitrates and unrelated to whether the join reads as a jump.

D The insert lasts two seconds, which...: Attributes the effect to duration rather than to what is on screen — a two-second dissolve would still show the face jumping, and a six-frame insert still works.

Lesson 19, p. 99

An editor mirrors a shot of a presenter horizontally to fix a screen-direction problem. The shot is a medium close-up against a plain office wall...

Answer C: Whether the frame contains anything asymmetric — lanyard text, a watch, a shirt placket, a hair parting — because a reversed one of those is instantly readable as wrong.

Why: One camera yields several usable shots through punch-ins, reframes, ramps, freezes, stills and graphics — but each repair has a cost, and a missing reverse angle of a real person is not among the things any of them can supply.

A Whether the lighting comes from the...: A real continuity concern, but a reversed key is usually subtle and often acceptable — it is not what makes viewers notice a flip.

B Whether the clip was shot at...: Mirroring is a lossless pixel reordering and does not resample anything, so resolution is not at stake.

D Whether the subject moves at all...: Reverses the purpose of the technique: changing screen direction is exactly what the flip was performed to achieve.

Lesson 20, p. 104

A creator moves their best moment to the front as a cold open and retention improves sharply — but average view duration falls and comments...

Answer D: The opening promises something the rest of the piece does not deliver, so people arrive on the strength of the hook and leave when the payoff does not come.

Why: Viewers answer 'what is this and is it for me' in the first few seconds whether or not you have told them, so state the subject immediately, open on a real image rather than black, and pay off whatever the opening promised.

A The cold open is too long...: Length of the hook affects pacing but would not produce complaints, and a shorter hook does not address a promise the piece fails to keep.

B The best moment now appears twice...: Repeating the moment is standard practice and usually passes unremarked; the objection is about what the rest of the piece delivered, not about seeing a clip twice.

C Retention and average view duration measure...: They are genuinely different — retention is the shape of the curve and duration is the area under it — and the two moving in opposite directions is exactly the signal worth attending to.

Module 3 · Structure — deciding what the film is

The module starts on p. 113.

MODULE 3 TEST

Module 3 test, Q1, p. 160

What does a one-sentence spine actually do for an edit?

Answer A: It decides which good material does not belong

Why: The value is not the sentence, it is that the sentence makes decisions. Every clip you are unsure about is held against it, and a good shot that does not serve it goes. Without one, a twelve-minute film about a clinic ends up containing four minutes about the building contractor, and there is no basis on which to argue otherwise. The spine is what tells you which of your favourite shots are darlings.

Module 3 test, Q2, p. 160

A thematically structured piece fails in a particular way. What is it, and what is the repair?

Answer C: Each join is a full stop; end each section on a question the next one answers

Why: Thematic sections are independent by construction, so nothing carries the viewer from one to the next: every join is a clean place to leave. Giving each section an unanswered question that the next one takes up restores a reason to continue. A change of texture across the join helps too. The other failures listed are real, but they belong to chronological and list structures.

Module 3 test, Q3, p. 160

An answer is cut from "In most cases, and if funding continues, we can do it" to "we can do it". The join is inaudible...

Answer B: Because the removed words turned a conditional plan into a commitment

Why: You may change what somebody said; you may not change what they meant. Shortening and tightening are expected by anyone who has been interviewed. Removing a qualification alters the proposition: the speaker holds a plan contingent on funding, and the cut has them holding an unconditional promise. The test is whether they could point at a claim in the cut that they do not hold, and a perfect join makes that harder to detect rather than more acceptable.

Module 3 test, Q4, p. 161

Why is grading a shot before picture lock described as doing the work twice?

Answer D: The shot gets shorter or replaced, and the grade has to be redone

Why: The stages exist because work done in one becomes worthless if the stage before it was not finished. Before lock the structure is still moving: shots are shortened, swapped and cut entirely, and every hour of grading spent on material that leaves the film is an hour spent twice. The same argument applies to mixing, titling and cutting music in early, which is where you start protecting the track instead of the film.

Module 3 test, Q5, p. 161

A first viewer says "the middle is boring". Read as a symptom report rather than a diagnosis, what does that usually mean?

Answer C: The middle has no question running through it

Why: People are almost always right that something is wrong and almost always wrong about what it is. Boredom in the middle is usually structural: nothing has been set up that the viewer wants resolved, so there is no reason to stay. Trimming frames treats the symptom. Look at what comes immediately before the point where their attention went, because that is where the piece lost them.

Module 3 test, Q6, p. 161

An editor in India wants ninety seconds of a news broadcast in a video essay, and is told "it is transformative, so it is fair...

Answer A: India has fair dealing, which applies only to an enumerated list of purposes

Why: Fair use is the American test: open-ended, four factors, any purpose can qualify, and transformativeness carries real weight. Fair dealing works differently. India's Section 52 lists the permitted purposes, and a use that is not on the list does not become permitted by being transformative. This is also why the folklore figures of ten seconds or seven notes have no basis anywhere, and why anything with real stakes is a question for a lawyer in your jurisdiction.

THE LESSON CHECKS

Lesson 21, p. 118

An editor writes the spine "a film about our company's new warehouse" and finds that the rough cut keeps sprawling — every section seems defensible...

Answer B: It names a topic rather than a question or a change, so every piece of material about the warehouse satisfies it equally and nothing can be excluded.

Why: Write what the piece is about as one sentence containing a question or a change, and hold every uncertain clip against it, because the sentence is what makes the decision about a good shot that does not belong.

A It is too short to guide...: Confuses a spine with a brief — those things belong in the brief, and adding them to the sentence does not give it the discriminating power it lacks.

C It is about a place rather...: Overgeneralises — plenty of strong pieces are built around a place, provided the sentence says something about it rather than merely naming it.

D It was written before the logging...: Names a real sequencing mistake, but a topic sentence written after logging would sprawl in exactly the same way.

Lesson 22, p. 122

A ten-minute explainer is cut thematically into four sections. Retention shows a sharp drop at each section boundary but is healthy within sections. What does...

Answer C: Each section ends as a complete unit, so the viewer reaches a natural stopping point four times; ending sections on an unanswered question removes the clean exit.

Why: Chronology is a choice rather than a default, and each shape has a specific failure — thematic sections give the viewer a clean exit at every join, lists have no momentum, and essays become lectures unless every claim has evidence on screen.

A The sections are too long, and...: Addresses duration when the drops are located precisely at the joins; shorter sections would simply produce more of the same boundaries.

B The thematic structure is wrong for...: Reaches for a different shape when the diagnosis points at the joins, and chronology carries its own failure modes that the material may not survive.

D The transitions are visually abrupt, and...: Treats a structural full stop as a visual problem — a title card at a boundary makes the stopping point more explicit rather than less.

Lesson 23, p. 127

An editor joins two sentences from different parts of an interview into one continuous answer. The join is inaudible, both sentences are the speaker's own...

Answer C: When the combination asserts a proposition the speaker does not hold — because a condition, a denial or a causal direction was dropped at the join.

Why: You may change what somebody said and not what they meant — so removing a qualification, joining clauses from different answers or cutting the 'no' are fabrications regardless of how clean the join is.

A Whenever non-adjacent material is joined at...: Sets the bar where no interview could be cut to length; joining separated material is ordinary practice, and the objection has to be about meaning rather than adjacency.

B Only if the speaker objects after...: Outsources the judgement to a reaction that may never come, and people frequently object to accurate cuts while missing genuinely distorting ones.

D Only if the two sentences came...: Uses a proxy that is neither necessary nor sufficient: two sentences from the same answer can still be joined to invert its meaning, and material from different questions can be combined faithfully.

Lesson 24, p. 132

An editor's assembly of a planned twelve-minute film comes out at fourteen minutes. Why is that a warning sign rather than a good sign?

Answer A: They made selection decisions without seeing the alternatives in context, so the rough cut has no room left in which to be structured.

Why: Assembly, rough cut, fine cut, lock and finishing exist because work done in one stage is thrown away if the previous stage was not finished, and unscripted material realistically costs three to five hours of editing per finished minute before any finishing.

B A fourteen-minute assembly means there was…: Possible, but the assembly length alone cannot distinguish a thin shoot from an editor who cut as they assembled — and the second is far more common.

C Assemblies are supposed to come in…: Inverts the convention: an assembly is expected to run several times the target, and a short one is the thing worth investigating.

D The timeline will be slow to…: Confuses an editorial signal with a performance one; fourteen minutes is trivial for any machine and the concern is about decisions, not playback.

Lesson 25, p. 136

Three separate viewers say the music in the middle section is wrong. The editor tries two different tracks and both viewers of the new version…

Answer B: Something else in the middle section is not working — most likely its pace or its lack of a question — and the music is simply the most nameable thing people noticed.

Why: You cannot watch your own edit because you know what is coming, so run a silent, uninterrupted screening with a stranger to the material and treat every note as a symptom report rather than a diagnosis.

A The music is genuinely wrong in…: Keeps testing the thing the note named after two attempts have failed, which is the standard way of spending a week on the wrong variable.

C The music is simply mixed too…: A real and common confusion worth checking, but it would normally present as trouble hearing the words rather than as a consistent objection to the music itself.

D Three viewers agreeing means the note…: Correct that repeated notes are reliable signals, wrong that reliability tells you the cause — agreement about a symptom says nothing about which change fixes it.

Lesson 26, p. 140

An editor identifies a sequence that took three days to make work and considers it a reason to keep it in the cut. Why is…

Answer A: Because time already spent cannot be recovered by keeping the sequence, and difficulty getting it to work is usually evidence that the material does not support it.

Why: A first fine cut is typically a fifth too long, and the material to remove is whatever is justified by your knowledge of the project rather than by its effect on a stranger — so park it rather than deleting it and watch the two versions.

B Because sequences that take a long…: Invents a quality correlation; a difficult sequence is often technically immaculate precisely because so much attention went into it.

C Because the editor is too close…: Right that proximity distorts judgement, but it converts a diagnosis into a deferral rather than acting on what the difficulty itself indicates.

D Because anything that required that much…: Overreaches with 'always' and with an assumption about provenance that the difficulty alone does not establish.

Lesson 27, p. 144

An editor writes a full narration script, records it, and then cuts picture to fit it. The result feels like a brochure read aloud despite…

Answer B: Each sentence had to be given a shot, so the picture became illustration of the words, and the two now say the same thing at once.

Why: Cut the picture first and write narration into the gaps it leaves, supplying only what the picture cannot show — and read every line aloud while writing, because anything you stumble over the listener will stumble over too.

A The narration was recorded before the…: A genuine consequence of the order used, but flat pacing would make the piece monotonous rather than make it feel like illustrated text.

C Writing first tends to produce longer…: True about writing for the ear and unrelated to the specific complaint, which is about the relationship between words and pictures rather than about sentence length.

D The footage has been cut too…: Describes a pacing symptom that often accompanies this mistake, but loosening the cuts leaves the words and pictures still duplicating each other.

Lesson 28, p. 148

A team takes their nine-minute film, exports the best 45 seconds of it unchanged as a vertical crop, and it performs badly despite being the…

Answer C: The section depended on context established earlier in the film, and a standalone short has to supply its own beginning and end because the viewer arrives with none.

Why: A short is not a trimmed long piece but a separate cut with its own spine, opening on the moment rather than the setup, and the cheapest way to get both is to compose and lock the long version first with the vertical crop already in mind.

A The crop lost image quality, since…: Real when the source is 1080p, but a 4K source crops to vertical at full delivery resolution, and the piece would still fail for reasons unrelated to sharpness.

B Vertical platforms re-encode more aggressively, so…: Aggressive re-encoding is genuine and worth compensating for, but it degrades the picture rather than explaining why a strong section stopped working as a standalone piece.

D The captions were not burnt in…: A serious and common omission, but it describes a delivery failure that would apply equally to a properly built short.

Lesson 29, p. 154

An editor in India plans to use 25 seconds of a commercial film in a video essay, relying on advice from an American tutorial that…

Answer A: Transformativeness is a US fair use concept; India's fair dealing applies only to enumerated purposes, so the question is whether the use is on that list.

Why: Third-party material is licensed, openly licensed, permitted or covered by a statutory exception — and because exceptions differ fundamentally between fair use and enumerated fair dealing, the widely repeated 'ten seconds is fine' rules have no basis anywhere.

B India has no copyright exceptions for…: Overstates the position: Section 52 does provide exceptions including criticism and review, so the issue is which list applies rather than whether any exception exists.

C Twenty-five seconds exceeds the limit that…: Repeats the durational folklore the lesson identifies as baseless; no statute specifies a number of seconds in either system.

D The video will inevitably be watched…: Gets the cross-border problem backwards — reaching multiple jurisdictions means more laws may apply, not that the most permissive one controls.

Module 4 · Sound, which is most of it

The module starts on p. 163.

MODULE 4 TEST

Module 4 test, Q1, p. 214

Why does a cheap lavalier at 25 cm beat an expensive microphone two metres away?

Answer D: Direct sound falls away with distance while reflections do not

Why: Sound arriving straight from the mouth drops off quickly with distance. Sound that has bounced off walls and ceiling is spread evenly through the room and barely drops off at all. Halving the distance therefore improves the ratio of direct to reflected sound dramatically, and that ratio is what makes speech sound close rather than like a bathroom. It is a property of the room and the geometry, not of the microphone's price.

Module 4 test, Q2, p. 214

Twelve decibels of spectral noise reduction leaves a watery, fluttering sound around the voice. What happened?

Answer B: Quiet harmonics of the voice fell below the estimate and came and went

Why: Spectral subtraction builds a profile of how much energy sits in each frequency band when only noise is present, then subtracts that amount everywhere. Pushed too far, quiet parts of the voice fall under the estimate and are removed in some frames and kept in others, and those fragments flutter. That is musical noise. Six decibels that leaves a little honest hiss is better, because audiences forgive background noise and do not forgive artefacts.

Module 4 test, Q3, p. 215

One dialogue clip arrives at the compressor at minus twenty and the next at minus eight. Why does no track fader setting fix both?

Answer A: The fader sits after the compressor, which has already treated them differently

Why: A compressor responds to whatever level reaches it. The clip at minus eight is well over the threshold and is squashed; the clip at minus twenty may not trigger it at all. By the time the signal reaches the fader the difference has already been baked in, and one fader value cannot undo two different treatments. Level the clips with clip gain first, which acts before the processing, then compress, then balance with the fader.

Module 4 test, Q4, p. 215

A hissy recording sounds worse after compression than before it. Why?

Answer C: Make-up gain raises the hiss by as much as it raises the voice

Why: A compressor makes nothing louder; it turns down what crosses the threshold. The make-up gain added afterwards raises everything that is left, including the air conditioning, the fan and the room. So the worse the original noise floor, the less compression a recording can take, and squashing a hissy clip by twelve decibels and adding twelve back leaves the hiss twelve decibels louder than it was. Compression does not create the problem; it reveals how large it was.

Module 4 test, Q5, p. 215

Why does mastering a film to minus six LUFS gain nothing on a platform that normalises to around minus fourteen?

Answer B: The platform turns it down, and the dynamics were already destroyed

Why: Platforms attenuate loud uploads and generally do not raise quiet ones by much. Getting a mix to minus six means heavy compression and limiting, so the dynamic range is gone before the file leaves your machine; the platform then takes eight decibels off and you arrive at the same perceived volume as everybody else with a flatter, more tiring mix. Mix near the destination's target and keep true peak at or below minus one dBTP.

Module 4 test, Q6, p. 215

What does a continuous ambience bed under a whole scene actually fix?

Answer D: It fills the holes that every cut punched in the background

Why: Every cut joins two different moments, and the background noise level differs slightly at each. The ear tracks the continuity of a space more attentively than the eye tracks the continuity of a frame, so those small breaks read as holes punched in the scene. One continuous bed under everything fills all of them at once. Set it so low you cannot point at it: the test is that muting it makes the scene sound dead rather than making you notice a sound stop.

THE LESSON CHECKS

Lesson 30, p. 168

An interview recorded near a busy road is cleaned up with heavy noise reduction. The traffic is gone, but the voice now has a swirling…

Answer B: The reduction is subtracting a noise estimate from every band, and quiet voice harmonics near that threshold are being removed in some frames and kept in others, so they flicker.

Why: Get the mic within 30 cm, keep peaks near -6 dBFS, record thirty seconds of room tone, and use less noise reduction than you want to.

A The high-pass filter was set too…: Blames a filter that cuts a fixed low band, which produces thinness rather than fluctuating artefacts.

C The recording was clipped at source…: Confuses permanent clipping damage, which sounds like crackle, with an artefact introduced by the processing itself.

D The noise profile was sampled from…: Identifies a real setup error, but an over-inclusive profile dulls the voice consistently rather than making it fluctuate.

Lesson 31, p. 172

A small business uploads a promotional video using a track from a paid stock library, with a valid commercial licence. Two days later YouTube shows…

Answer C: The library registered its catalogue with Content ID, the upload was fingerprint-matched automatically, and the system has no knowledge of the licence until it is submitted in a dispute.

Why: Every track needs a licence you can produce on demand, and "royalty free" is a pricing model, not permission.

A The licence covered personal use only…: Assumes an automated match implies the licence was wrong, when the terms were stated to be commercial.

B Using licensed music without on-screen attribution…: Treats attribution, a condition of some free licences, as the trigger for an automated fingerprint match.

D The video was flagged as a…: Conflates a monetisation claim, which is automated and low-stakes, with a strike, which is a formal legal notice.

Lesson 32, p. 176

An editor compresses the dialogue bus and then levels the individual clips with clip gain afterwards. Why does the result sound inconsistent no matter how…

Answer D: The compressor reacts differently to each clip depending on the level arriving at it, so raising a quiet clip afterwards raises a signal that was already compressed differently.

Why: Organise audio into dialogue, effects and music busses so a change is one fader rather than forty clips, and level clips with clip gain before any compressor sees them, because a compressor responds to whatever level arrives at it.

A Clip gain is applied after the…: Reverses the signal path: clip gain is pre-fader and pre-processing, which is exactly why it is the right tool for levelling before compression.

B Compression should always be applied per…: Per-clip compression is occasionally useful but is not the issue here, and a bus compressor on levelled dialogue is standard practice rather than a mistake.

C Clip gain changes the waveform display…: Clip gain genuinely changes level and the display reflects it; the confusion described would make the technique useless in every order.

Lesson 33, p. 182

A voice recorded in a bare room sounds hollow and distant. An editor cuts 4 dB at 400 Hz, which helps slightly, then tries progressively...

Answer B: The hollowness is reverberation, which shares every frequency with the direct sound, so an equaliser can only remove the voice along with it.

Why: Four moves cover most speech — a high-pass around 80 to 100 Hz, a notch at the mains frequency and its harmonics, a small cut in the 300 to 500 Hz boxy region and a gentle lift around 2 to 3 kHz — applied subtractively before any compressor sees the signal.

A The cut is at the wrong...: The right technique for finding a resonance, and it will locate the room's strongest mode — but the remaining hollowness is reverberation rather than a single peak.

C A high-pass filter should have been...: Correct about the order of operations and about the value of a high-pass, but low-frequency rumble is not what makes a voice sound hollow and distant.

D The microphone's own frequency response is...: Blames the equipment when the problem is the ratio of direct to reflected sound — the same microphone at 20 cm in that room would sound fine.

Lesson 34, p. 186

An editor compresses an interview heavily to make it consistent, and afterwards the background hiss audibly surges up in the gaps between sentences. Which two...

Answer C: A fast release with heavy make-up gain, so the compressor recovers during pauses and the amplified noise floor comes straight up with it.

Why: Set dialogue compression by the gain reduction meter — 3 to 6 dB on loud words, returning to zero between phrases — and remember that make-up gain raises the noise floor too, so a hissy recording can take less compression than a clean one.

A A slow attack combined with a...: Describes settings that cause a different artefact — escaped transients followed by a dip — rather than noise rising in the silences.

B A threshold set too high, so...: A threshold that is too high means the compressor barely acts at all, which cannot produce a surge — the artefact requires the compressor to be working hard and then releasing.

D A limiter on the output with...: A low ceiling does squash a mix, but a limiter reduces peaks rather than lifting quiet passages, so it cannot raise the noise floor in the gaps.

Lesson 35, p. 191

A creator masters a video at -8 LUFS integrated because it sounds punchier in their editor, and uploads it to a platform that normalises to…

Answer A: The platform attenuates it by roughly 6 dB, so it arrives at the same perceived loudness as everyone else but with the dynamic range already spent on getting to -8.

Why: Loudness is an energy measurement over time, not a peak, so mix to roughly the destination's integrated LUFS target with true peak at -1 dBTP — mastering louder than the target just means the platform turns you down after you destroyed the dynamics.

B The platform rejects the file, since…: Describes broadcast delivery, where tolerances are enforced and files are rejected — consumer platforms normalise rather than refuse.

C The extra loudness survives, because normalisation…: Video platforms normalise too, which is precisely why the loudness race that once existed in music has largely stopped mattering on them.

D The true peak will exceed 0…: A genuine risk of hot masters and worth guarding against with a -1 dBTP ceiling, but it is a separate failure from what normalisation itself does to the file.

Lesson 36, p. 195

A sequence cuts between three locations and a drone shot. The picture is good but the piece feels flat and unconvincing. Which sound problem does…

Answer B: The background changes at every join and vanishes under the drone shot, and the ear tracks spatial continuity closely enough that the break registers as unreality.

Why: A continuous ambience bed fills every hole your cuts made in the background at once, and spot effects only need to cover the events the viewer can see happening — layered two or three deep and placed a frame early rather than a frame late.

A The dialogue is under-compressed, so levels…: Level inconsistency between locations is real and worth fixing, but it produces difficulty hearing rather than a sense that the places are not real.

C Spot effects are missing for the…: Missing spot effects do flatten specific moments, but the complaint here spans whole locations including a drone shot with no events in it.

D The music bed is carrying the…: Reasonable observation about what music can and cannot do, but it describes a symptom of the missing ambience rather than the cause of the flatness.

Lesson 37, p. 199

An editor replaces one line in a reverberant warehouse scene with a clean studio recording. Before touching any reverb plug-in, which single step usually removes…

Answer D: Laying the scene's room tone under the replaced line, because the audible hole is the sudden silence rather than the missing reflections.

Why: A replaced line sounds wrong because its space is wrong, so match the noise floor first, then the EQ, then add far less reverb than sounds like reverb — and record an impulse response with one clap at every location, because de-reverb is much harder than matching.

A Rolling off the high frequencies on…: A real and useful distance cue, but it addresses timbre rather than the absence that the ear notices first.

B Matching the level of the replacement…: Level matching is necessary and is usually already done, and a perfectly level-matched dry line in a reverberant scene still sounds obviously pasted in.

C Applying a de-reverb pass to the…: Technically possible and almost always the wrong direction: de-reverb is lossy and artefact-prone, so it risks a whole scene to save one line.

Lesson 38, p. 203

An interview has a motorbike passing during a key answer. Spectral noise reduction, which worked well on the room's air conditioning, makes no useful difference…

Answer C: Spectral subtraction builds a fixed profile from steady noise, and a sound that changes pitch as it passes matches no single profile.

Why: Hum, clicks, plosives and level faults are fully repairable; noise, reverb and mild clipping are partly repairable up to the point where the artefact is worse than the fault; and speech under the noise floor or heavily clipped is gone, which is worth saying to a client on day one.

A The motorbike is louder than the…: Treats it as a matter of degree; the failure is that the profile does not describe the sound at all, so more attenuation removes the wrong energy.

B The motorbike occupies the same frequencies…: Overlapping frequencies do make separation hard, but spectral repair can paint out an overlapping event in some cases — the decisive problem is that the noise is not stationary.

D Noise reduction only works on noise…: Names level rather than spectrum and invokes a gate that spectral subtraction does not use, though it lands near the right intuition about steadiness.

Lesson 39, p. 207

A mix sounds excellent on headphones and thin and quiet on a phone speaker. The editor raises the overall level to compensate and the platform...

Answer A: Rebalanced so the dialogue carries the mix without relying on low frequencies, checking at low volume to see what survives.

Why: Check every mix on at least three systems and fix only the faults that appear on more than one, because each system lies in its own direction — and the low-volume test tells you more about balance than any meter.

B Boosted the low frequencies, since the...: Compensating for a speaker's missing band by adding energy there wastes headroom on frequencies that speaker will never reproduce, and ruins the mix everywhere else.

C Applied heavier compression to the whole...: Heavier compression does help intelligibility on small speakers, but applied to the whole mix it raises music and noise alongside speech and does not fix the balance.

D Exported a separate mix for phone...: Platform-specific mixes exist in broadcast, but they are not deliverable to a feed and they avoid rather than solve a balance that should work on both.

Module 5 · Colour, correction before grade

The module starts on p. 217.

MODULE 5 TEST

Module 5 test, Q1, p. 264

Every shot in a scene came off one camera in one profile, filmed over two hours in changing light. Why does one LUT applied to...

Answer C: A LUT maps input values to output values, and the inputs differ

Why: A lookup table is a fixed transformation with no knowledge of your image. If one shot is a stop under and six hundred kelvin warm and another is neutral, the identical mapping lands on two different starting points and produces two different results. Neutralise and match the shots first, on the waveform and the parade, and then one look travels across all of them. That is the whole argument for correction before grading.

Module 5 test, Q2, p. 264

A face is too dark, but the black point and the white point are both where you want them. Which control moves the face without...

Answer A: Gamma

Why: Gamma applies an exponent. Zero and one raised to any power are still zero and one, so black and white are pinned while everything between them moves. Gain multiplies, which sets the white point; lift adds, which sets the black point; offset adds with no pivot at all, so the whole trace slides. This is why gamma comes third in the order and why it is the control beginners reach for least and need most.

Module 5 test, Q3, p. 264

On the RGB parade the bottoms of the three traces do not align, and blue sits highest. What is wrong, and what fixes it?

Answer D: The shadows are blue; pull the blue component of lift down

Why: The parade shows red, green and blue side by side. The bottom of each trace is that channel's black point, so a blue trace sitting higher at the bottom means the darkest parts of the picture carry more blue than red or green. Lift is the black point control, so pulling its blue component down brings the bases together. The tops are the white point and belong to gain. This one reading fixes most white balance problems in about fifteen seconds.

Module 5 test, Q4, p. 265

A skin qualifier on phone footage produces a blocky, crawling matte. Which of these is true?

Answer B: No setting fixes it; the mask is built from quarter-resolution colour

Why: A qualifier selects by colour, and in 4:2:0 footage colour is recorded at half the horizontal and half the vertical resolution, so one sample covers a block of four pixels. The mask is therefore built from a quarter of the information the picture holds, and its edges come back stepped and unstable. Blurring the matte and keeping the correction subtle hides it; nothing repairs it, because the limit is in the file rather than in the software.

Module 5 test, Q5, p. 265

An export has grey blacks and dull whites in one player and looks correct in another. ffprobe reports a blank color_range field. What is happening?

Answer A: Nothing was tagged, so each player is guessing the range

Why: A file is written either full range, 0 to 255, or video range, 16 to 235, and it carries a flag saying which. Read video-range data as full and blacks lift to grey and whites dull; read full-range data as video and blacks crush. A blank flag means nothing was told to the player, so different players guess differently. A gamma difference is real but subtle and lives in the shadows; this is the large, obvious version.

Module 5 test, Q6, p. 265

Why does adding a little grain before export reduce visible banding in a sky?

Answer C: Noise scatters pixels either side of the boundary, so the contour softens

Why: Banding is visible because the boundary between two quantisation levels is a clean contour your eye can trace. Noise of roughly half a level makes pixels near that boundary fall randomly on either side of it, which turns a hard edge into a dither the eye averages back to a smooth gradient. The cost is bitrate: noise is expensive to compress, so grain plus a stingy upload gives you a different problem.

THE LESSON CHECKS

Lesson 40, p. 222

An editor applies a purchased cinematic LUT to footage from a phone. Skies clip to white, shadows go solid black, and skin turns orange. The…

Answer C: The LUT expects a flat log image as input; the phone footage already has contrast and saturation baked in, so the LUT adds a second helping on top.

Why: Neutralise and match every shot against the scopes first, because a look applied to unmatched shots produces unmatched results.

A The phone footage is 8-bit, and…: Correctly identifies a bit-depth limitation but misapplies it — 8-bit shows banding under heavy grading, not blown highlights.

B The LUT is built for a…: Reaches for a colour-space conversion when the visible symptom is a double dose of contrast on an already-contrasty image.

D The LUT was applied at 100%…: Names a real remedy but treats reduced opacity as the design intent rather than a way to soften an already-mismatched transform.

Lesson 41, p. 226

An editor places a strong contrast node first in the chain and a white balance node after it. The balance correction behaves oddly and cannot…

Answer A: The balance node is now working on an image whose shadows and highlights have already been crushed, so it has fewer distinct values to redistribute.

Why: A node graph lets you choose what each adjustment applies to and in what order, so correct first and style after, name every node, and use the still store to carry one shot's grade across a whole scene.

B Contrast nodes must always be last…: States a rule that does not exist — plenty of grades put operations after contrast, and the issue is what the balance node receives rather than a fixed position for contrast.

C White balance is a per-clip attribute…: Confuses a camera raw setting, which is genuinely pre-graph, with the balance controls on a node, which operate exactly where they sit.

D Node order does not affect the…: Colour operations are not commutative — applying contrast then balance produces a different image from balance then contrast, which is the whole reason order is a decision.

Lesson 42, p. 231

A shot's faces are too dark but the highlights and shadows are already exactly where the editor wants them. Which control changes the faces without…

Answer D: Gamma, because a power curve leaves zero and one fixed and only redistributes the values between them.

Why: Gain multiplies so it sets the white point, lift adds so it sets the black point, gamma is a power curve so it moves the mid-tones with both ends pinned — set them in that order and read every move on the waveform.

A Lift, because it raises the shadows…: Lift adds a constant, so it raises the black point as well as the faces — the shadows the editor wanted to keep would move.

B Offset, because it shifts the whole…: Preserving relationships is exactly the problem: offset moves black and white along with the faces, so neither end stays where it was.

C Gain, because reducing it slightly compresses…: Gain multiplies, so lowering it darkens the faces along with the highlights rather than raising them.

Lesson 43, p. 236

An editor applies an S-curve and the faces look better defined, but they describe it as the curve having 'added detail'. What actually happened?

Answer B: The curve is steeper through the mid-tones, so existing small differences there are mapped to larger output differences and become more visible.

Why: A curve maps input values to output values, so a gentle S steepens the mid-tones where faces live and a reverse S produces the faded film look — and Lum versus Sat pulled down in the shadows stops dark noise reading as coloured.

A The curve sharpened the mid-tone region…: Confuses tonal contrast with spatial sharpening — a tone curve operates per pixel value and has no knowledge of edges at all.

C The curve raised saturation in the…: An S-curve does tend to increase apparent saturation as a side effect, but that changes colour separation rather than revealing tonal differences within the face.

D Contrast recovers detail that compression had…: Nothing recovers discarded information; on heavily compressed footage a steep curve exposes the compression artefacts rather than undoing them.

Lesson 44, p. 240

An editor uses automatic shot match to bring a close-up of a face in line with a wide shot of a green field. The face…

Answer A: It matched the overall colour distributions of the two frames, and green dominates the wide shot's while the close-up has almost none.

Why: Grade one hero shot completely and bring every other shot to it on a split wipe, matching black point, white point, mid-tone, balance and saturation in that order on the scopes — and judge the result at playback speed, not frame by frame.

B The tool sampled the wrong reference…: A different frame would shift the result but not the method — any frame of that wide shot is dominated by the field.

C The tool cannot analyse skin tones…: Invents a prerequisite; shot match makes no attempt to identify subjects at all, which is exactly the limitation on display here.

D The close-up was shot at a…: Shot match does adjust balance as well as exposure, so a white balance difference is within its scope rather than outside it.

Lesson 45, p. 244

An editor pulls a skin-tone qualifier on 8-bit 4:2:0 phone footage to warm a face. The matte edges are blocky and flicker between frames. Which...

Answer B: Blur and denoise the matte heavily, and keep the correction subtle enough that a low-resolution key can carry it.

Why: Qualifiers select by colour and windows by shape, both feeding a node's key — and because a qualifier builds its mask from chroma, 4:2:0 footage yields a mask at a quarter the resolution of the picture, which no setting repairs.

A Increase the qualifier's luminance range, since...: Widening luminance changes what is selected, not the resolution at which the selection is computed — the edges stay blocky and more of the frame gets included.

C Switch the node to a layer...: Blending reduces the strength of the whole correction uniformly and does nothing about the stepped boundary where the matte transitions.

D Re-render the clip to a 10-bit...: Transcoding to a higher bit depth preserves what is there and creates no new chroma samples, so the key is built from exactly the same information.

Lesson 46, p. 249

An export looks correct in the editor and noticeably washed out — lifted blacks, dull whites — in a media player. `ffprobe` reports `color_range=pc`. What...

Answer C: The file is tagged full range but the player is treating it as video range, so values are being remapped into the 16 to 235 band.

Why: A pixel value only means a colour once you know the space, white point and transfer function — so the washed-out export is usually a full-range file being read as video range, and the browser mismatch is usually gamma 2.2 against 2.4.

A The player is applying gamma 2.2...: A real cause of shadow differences between environments, but it produces a subtle shift in the darks rather than dull whites as well.

B The file was graded in a...: Gamut clamping desaturates and shifts hues rather than lifting blacks and dulling whites uniformly across the tonal range.

D The export used a low bitrate...: Low bitrate produces blocking and banding in busy or dark areas, not an even lift of the black point and a reduction of the white point.

Lesson 47, p. 253

An editor applies a popular film-emulation LUT to well-exposed, standard-profile footage from a phone. Highlights blow out and skin goes orange. What is the mechanism…

Answer B: The LUT expects a flat log input and is adding contrast and saturation on top of an image that already has both.

Why: A look is five deliberate decisions — contrast shape, a shadow-highlight colour split, saturation shaped by luminance, a hue shift or two, and grain — so build and save your own rather than applying a LUT that expects a log image it is not receiving.

A The LUT is designed for a…: Camera-specific LUTs exist, but the decisive property is whether the input is log or display-referred, not which brand recorded it.

C The phone recorded 8-bit, so the…: Bit depth genuinely limits how far a grade can be pushed, but it produces banding in gradients rather than clipped highlights and a hue shift on skin.

D The footage is in the wrong…: Right that a transform is involved and wrong about the order — a transform after the LUT operates on an image the LUT has already misinterpreted.

Lesson 48, p. 257

An editor performs a full quality-control pass by watching the timeline end to end in their editor at full screen, finds nothing wrong, and delivers…

Answer C: The preview and the exported file can handle data range and colour tagging differently, so the mismatch exists only in the file.

Why: Broadcast-legal means values between 16 and 235 in 8-bit video range, readable as lines on the waveform — and every quality-control check happens on the exported file in a normal player, because several classes of error exist only in the export.

A Timeline playback uses the render cache…: Cache quality is a real setting worth checking, but it usually makes the preview worse than the export rather than concealing a range error.

B The editor should have watched at…: Viewing scale affects how visible compression artefacts are and has nothing to do with the black and white points of the file.

D Watching the whole piece is insufficient…: A scope on the export is a genuinely useful check, but the failure here is that no property of the export was examined at all, not which instrument was used.

Module 6 · Text, captions and graphics

The module starts on p. 267.

MODULE 6 TEST

Module 6 test, Q1, p. 312
Why is an unread automatic caption track more dangerous than one full of obvious gibberish?
Answer B: It produces fluent, grammatical sentences that say something else
Why: Recognition does not fail by producing nonsense you would notice. It substitutes a confident, well-formed alternative: deploy the model becomes destroy the model, and the sentence still scans. Nobody proofreads a track that reads fine. Accuracy also falls sharply on accented English, code-switching, proper nouns and technical terms, which is most of what this audience records. Reading a five-minute track takes about four minutes.

Module 6 test, Q2, p. 312
Thin type looks elegant in the editor and breaks up after upload. What is the mechanism?
Answer D: The encoder discards fine detail over moving video first
Why: An encoder allocates bits by how much changes between frames, and fine detail over moving video is the first thing it gives up. A hairline stroke two pixels wide does not survive quantisation; a bold stroke six or eight pixels wide does. That is why the fix for text that breaks up is usually weight rather than size, and why high-contrast display serifs with very thin strokes are a poor choice on video.

Module 6 test, Q3, p. 312
Captions near the bottom of a vertical video are legible in the editing canvas and unreadable on the platform. Why?
Answer C: The username, description and buttons sit over the bottom fifth
Why: The frame is intact; the platform's own interface is drawn on top of it. The bottom fifteen to twenty per cent carries the caption, username and buttons, and the top tenth carries the status bar and platform chrome. The editing canvas has no interface on it, which is why this is missed constantly. Anything that must be read goes in the middle third, and it is worth checking a real screenshot from the platform.

Module 6 test, Q4, p. 313
A scale from 100 to 110 per cent with two linear keyframes reads as mechanical. Why?
Answer A: Nothing physical changes value at a constant rate
Why: Linear interpolation means the value changes by the same amount every frame, so the move begins at full speed and stops dead. Objects in the world accelerate and decelerate, so the eye reads a constant rate as machinery. Easing out of the start and into the end gives the move acceleration, and the curve editor's bezier handles let you shape it precisely. It is the single most reliable tell of animation done without thinking about it.

Module 6 test, Q5, p. 313

A subtitle track drifts further out of sync as the film goes on, and shifting the whole file does not help. Why not?

Answer D: A growing error is a frame-rate problem, not a constant offset

Why: Shifting a file adds the same amount to every cue, which repairs a constant offset such as a transcription made from a file with a different start point. An error that grows means the timebase itself disagrees, which is the variable frame rate problem from the first block wearing a different hat. Conform the video to a constant frame rate first; the subtitles will then hold with a single shift or with none.

Module 6 test, Q6, p. 313

A Hindi caption typed into an editor's title tool shows the vowel sign to the right of its consonant. What does that tell you?

Answer B: The tool is not applying the font's reordering rules

Why: In Devanagari the vowel sign i is typed after its consonant and drawn before it, and conjuncts combine two consonants into one glyph. Applying those rules needs a shaping engine reading the font's OpenType tables. Where the tool has none, the characters are drawn in typed order, so the letters are all present and in the wrong shapes. Changing the font will not help. Test with ☒☒ once per tool and write the answer down.

THE LESSON CHECKS

Lesson 49, p. 272

A creator burns captions into a vertical video and positions them near the bottom of the frame, where they looked balanced on the editing canvas…

Answer B: The platform's interface — username, description and buttons — overlays the lower portion of the frame, which the editing canvas does not show.

Why: Burn captions in for feeds and upload a real subtitle file as well, and never publish an auto-transcript you have not read.

A Burnt-in captions are compressed along with…: Names a real risk of burnt-in text but does not explain why position specifically caused the failure.

C The video was exported at the…: Assumes a crop, which would remove the captions entirely rather than make them hard to read.

D Sidecar subtitles were also enabled, and…: Imagines a conflict between caption types, when a sidecar file is not what the creator produced here.

Lesson 50, p. 276

A title in an elegant thin typeface looks crisp in the editor and comes back fringed and broken after upload. Why does increasing the point...

Answer D: The encoder discards fine detail that changes between frames, and a thin typeface keeps thin strokes at any size it is set at.

Why: Size text as a percentage of frame height with a floor around 2.5 per cent, never use a light weight because the compressor destroys thin strokes, give every line its own contrast mechanism against a changing background, and hold it long enough to read aloud twice.

A The platform re-encodes at a lower...: Platforms do serve lower resolutions to some viewers, but a larger glyph survives a downscale better rather than worse, so this predicts the opposite result.

B Scaling a title up in the...: True if the title is scaled as an image, but re-typing at a larger point size renders fresh vectors and the fringing still appears.

C The fringing is chroma subsampling on...: Chroma subsampling does soften coloured edges and contributes, but white text on dark backgrounds carries most of its edge in luminance, where weight is decisive.

Lesson 51, p. 280

An editor places each interviewee's name super on the first frame the person appears, and testing shows viewers frequently miss the names entirely. What is...

Answer B: New text draws the eye away from the speaker, so the face and the name compete for the same first second of attention.

Why: A super answers 'who is this' two or three seconds after somebody starts speaking, its second line says why they matter rather than their job title, and every super in a piece comes from one duplicated template — while credits discharge real licence obligations from your log.

A The supers are on screen too...: Duration matters and is often too short, but here the text is missed at its start rather than cut off partway through being read.

C Viewers on phones cannot read text...: A genuine and serious constraint on vertical video, but it would make supers unreadable rather than missed at a specific moment in the timing.

D Supers placed on a cut are...: Encoding at a cut does allocate bits differently, but a super that persists for seconds is fully resolved long before a viewer would finish reading it.

Lesson 52, p. 284

An editor builds a speed ramp by setting several keyframes in Resolve's Retime Speed view — 100%, then 200%, then 100% — and the result…

Answer A: It plots which source frame plays at each moment, so the keyframes can be eased and the rate of change itself becomes gradual rather than stepped.

Why: Two keyframes and an interpolation is all timeline animation is, so ease every move because nothing physical changes value at a constant rate — and remember that a speed ramp needs easing on the retime frame graph rather than steps on the speed graph.

B It shows more keyframes, since the…: Both views show the same keyframes; the difference is what the graph's axis represents, not how many points are visible.

C It enables optical flow interpolation, which…: Optical flow is a separate setting that affects how frames are generated at any speed, and it will not remove a discontinuity in the speed itself.

D It applies motion blur between the…: Motion blur can soften the appearance of fast motion but is not what the Frame view provides, and blurring a jolt leaves the jolt.

Lesson 53, p. 288

An editor renders a 3D animated logo in Blender to a compressed MP4 with a black background, then imports it and sets the blend mode…

Answer C: A blend mode derives transparency from brightness, so dark parts of the logo go semi-transparent and edge artefacts composite in as haze.

Why: Fusion inside Resolve, Blender for 3D, Inkscape for vector artwork and a browser for data-driven graphics cover every free route to motion graphics — and one title template you made and reuse beats a downloaded pack everybody else is also using.

A MP4 files cannot be imported with…: States a real format limitation as though it justified the method, when the alternative is to render a format that does carry alpha in the first place.

B Blender cannot render alpha channels directly…: Blender renders alpha readily via RGBA output, so the premise is wrong and the technique is a workaround for a problem that need not exist.

D PNG sequences render faster than MP4…: Image sequences are more robust to a crash mid-render, but encoding speed is not the reason to prefer them and does not address the transparency problem.

Lesson 54, p. 292

A subtitle file is in sync at the start of a film and roughly two seconds late by the end. An editor applies a constant…

Answer D: The error grows over time, so it is a rate mismatch rather than a fixed delay, and a constant shift can only make one point in the film correct.

Why: A subtitle file is plain text with an index, two timecodes and up to two lines, which means find-and-replace fixes a recurring error in one action — and a constant sync offset can be shifted while a growing one means the video's frame rate needs conforming.

A The offset should be applied in…: Direction matters, but reversing a constant shift on a progressive error simply moves the point at which the error is zero.

B SRT timecodes are frame-accurate rather than…: SRT timecodes carry milliseconds and rounding at that scale could not accumulate to two seconds across a film.

C The subtitle file must be regenerated…: Index numbers are cosmetic and most players ignore them entirely, so shifting a file does not break anything structural about it.

Lesson 55, p. 298

A Hindi title typed into a video editor shows the vowel sign for 'i' to the right of its consonant instead of the left. The…

Answer C: The editor is drawing glyphs in typed order without a shaping engine, so no font can supply the reordering that the renderer is not performing.

Why: Indic and Arabic scripts need a shaping engine to reorder and combine glyphs, and many video editors do not have one — so test with a conjunct and a reordering vowel, and use libass through FFmpeg or a sidecar file rather than the title tool.

A The fonts lack the required OpenType…: Incomplete font tables do cause script failures, but a mainstream Devanagari font has those tables — the software is not reading them at all.

B The text was typed in the…: Reverses the encoding model: Devanagari is stored in logical order with the vowel after its consonant, and the renderer is supposed to reorder it for display.

D The title tool is rendering at…: Rendering size affects fidelity of the glyph shapes but not which side of a consonant a mark is placed on, which is a logic decision rather than a resolution one.

Lesson 56, p. 302

A tutorial is recorded on a 4K display and delivered at 1080p. Viewers on phones say they cannot read the menus. The editor's instinct is…

Answer C: The text occupies the same small fraction of the frame whatever the delivery resolution, so only enlarging it within the frame helps.

Why: Capture at the delivery resolution so interface elements stay large, punch in with an eased move rather than a cut whenever detail matters, ramp through every moment of waiting, and redact with a tracked solid box because blur on text is not reliably irreversible.

A Exporting at 4K increases the file…: Bandwidth is a real consideration and platforms do serve lower tiers, but it does not address why a phone viewer still cannot read the text.

B 4K exports take much longer to…: Render time is a practical annoyance rather than an argument about legibility, and it would not matter if the 4K export actually solved the problem.

D Punching in improves the compression efficiency…: A magnified image does compress somewhat differently, but efficiency is a side effect rather than the reason the technique makes text readable.

Lesson 57, p. 306

A video chart scales a circular icon's diameter in proportion to a value: 100 units gives a 40 px circle, 200 units gives an 80…

Answer A: The viewer reads area rather than diameter, and doubling the diameter quadruples the area, so a doubling of the value appears as a fourfold increase.

Why: A chart on video carries one comparison, revealed in time so the animation directs attention, with labels on the elements rather than in a legend — and because a viewer cannot go back and check the axis, starting a bar chart anywhere but zero is a stronger deception on video than on a page.

B Circles are harder to compare than…: True as a general point about perceptual accuracy, and it does not explain the specific systematic overstatement that this scaling produces.

C The larger circle occupies more of…: Attention and memory effects are real but describe a different mechanism from a quantifiable error in how the magnitude is encoded.

D Pixel dimensions do not scale linearly…: Both circles scale together with the frame, so their ratio to each other is unchanged on any screen size.

Module 7 · Effects, compositing and repair

The module starts on p. 315.

MODULE 7 TEST

Module 7 test, Q1, p. 360

Why does stabilisation always cost some of the frame?

Answer A: Moving each frame to cancel the shake leaves empty edges to hide

Why: The stabiliser works out how much of the frame-to-frame movement is unwanted and shifts each frame the other way to cancel it. Shifting a frame exposes empty space at the edges, so the image is scaled up and cropped to cover it. That crop is the price and nothing avoids it: five to fifteen per cent for handheld walking footage, more for anything violent. It is one of the honest arguments for shooting larger than you deliver.

Module 7 test, Q2, p. 360

A very shaky shot is stabilised. It comes back steady, and every individual frame is blurred. What is the cause?

Answer C: The motion blur was already inside each frame

Why: Stabilisation repositions whole frames. It cannot remove the blur that the camera's movement put into each individual exposure, because that blur is part of the pixels. The result is a steady sequence of soft frames, which looks strange precisely because the two impressions contradict each other. Mode choice matters for swimming and warping; it does nothing about blur that was recorded.

Module 7 test, Q3, p. 360

A clip slowed with optical flow melts around a striped shirt and is clean everywhere else. Why that region?

Answer B: A repeating pattern gives the matcher many equally good answers

Why: Optical flow estimates how each pixel moved between two frames and warps the image along those vectors to build a new one. A repeating pattern matches equally well at several offsets, so the estimate is ambiguous and the algorithm picks one that is wrong, and the invented frame tears in that region. The same failure appears on railings and brickwork, and for related reasons on plain surfaces, fast motion and crossing objects.

Module 7 test, Q4, p. 361

Why is rotoscoping described as a budgeted task rather than a quick repair?

Answer D: Five seconds of footage costs one to four hours by hand

Why: A five-second shot at 25 fps is 125 frames, and a hand-drawn animated mask runs at roughly five to thirty seconds of work per frame depending on the edge. Knowing the number changes decisions: it makes a cutaway, a different take or a reframe worth looking for first. Keyframing sparsely and splitting the subject into simple parts reduces it; machine-assisted masks get you most of the way and leave the hair and the motion blur.

Module 7 test, Q5, p. 361

A green screen was lit with one lamp and keys badly. Which step does more than any setting on the keyer?

Answer C: A garbage matte and an evening-out pass before the keyer sees it

Why: One lamp means the green in the corner is a different colour from the green in the centre, and no single tolerance covers both. A garbage matte removes the light stands, the floor and the edges of the sheet before the keyer sees them, and an evening-out pass lifts the dark corner toward the middle so one tolerance can work. Widening the tolerance instead starts eating the subject, and heavy despill turns skin magenta.

Module 7 test, Q6, p. 361

Removing a light stand from a locked-off shot takes minutes and from a handheld shot takes hours. What is the difference?

Answer A: A static patch slides off, so the patch has to be tracked and matched

Why: On a locked-off shot the background behind the object is identical in every frame, so a clean region copied over it with a soft mask holds for the whole shot. When the camera moves, the patch has to be attached to a track of the surface it sits on, masked, and matched for grain and brightness. If anything passes in front of the patched area, that edge has to be rotoscoped, which is where the hours actually go.

THE LESSON CHECKS

Lesson 58, p. 320

An editor stabilises a shot taken while walking quickly at dusk. The camera now holds steady, but the result looks wrong — steady and yet…

Answer B: Each frame still carries the motion blur the shaking camera put into it, and steady motion with blurred frames reads as unnatural.

Why: Stabilisation cancels frame movement and scales up to hide the empty edges, so it always costs 5 to 15 per cent of the frame, cannot remove motion blur already in each frame, and needs a separate rolling-shutter pass or it makes skew more visible rather than less.

A The crop required for stabilisation upscaled…: A real cost of a heavy crop, but a 10 per cent upscale produces mild softening rather than the specific mismatch between steady motion and blurred frames.

C The stabiliser used Perspective mode and…: Perspective mode genuinely can swim on some footage, but swimming reads as a shifting warp in the geometry rather than as uniform softness.

D Low light forced a slower shutter…: Dusk footage is often soft and noisy, but the described effect is specifically that the frames are blurred while the camera is not moving, which no amount of pre-existing softness explains.

Lesson 59, p. 325
An editor slows a shot of a person walking past a striped fence to 40% using optical flow. The person looks fine; the fence smears...

Answer B: A repeating pattern gives the motion estimator many equally good matches between frames, so it picks the wrong correspondence.

Why: Slowing a clip means inventing frames, and the three methods trade differently — duplication stutters but invents nothing, blending ghosts, and optical flow is seamless until its motion estimate is wrong and the image melts at the region it got wrong.

A The fence is in the background...: Parallax does make background motion faster in some shots, and here it is the pattern rather than the speed that defeats the matcher.

C Vertical structures are harder to track...: Invents a directional bias; motion estimation searches in two dimensions, and railings fail because of ambiguity rather than orientation.

D The fence is in sharper focus...: More detail generally helps a matcher rather than hindering it — a featureless region is the harder case, not a detailed one.

Lesson 60, p. 329
An editor rotoscopes a person's arm as it swings quickly across frame. The mask follows accurately, and the result still reads as cut out. What...

Answer C: The motion blur at the fast-moving edge, which is genuinely semi-transparent across several pixels rather than being a hard boundary.

Why: A mask is a shape with feather, expand and invert, and animating one by hand costs roughly one to four hours per five seconds of footage — so keyframe sparsely, split the subject into simple parts, and look for a solution that is not rotoscoping first.

A The arm's depth relative to the...: Depth information helps with defocus and occlusion, but a flat matte composites convincingly all the time when the edge is right.

B The feathering needed at the edge...: Variable feather is a genuine refinement available in better tools, and it still cannot describe an edge spread across many pixels by movement.

D The colour spill from the background...: Spill is a real giveaway in keying work, where a coloured screen contaminates the subject — an ordinary background does not spill in the same way.

Lesson 61, p. 333

A key on evenly lit but noisy footage produces a matte whose edges crawl and flicker from frame to frame. Which step addresses the cause…

Answer B: Applying a temporal denoise to the source before the keyer sees it, so the chroma channels stop generating different values each frame.

Why: A keyer builds its matte from colour, so a garbage matte, an evening-out pass and a denoise before keying do more than any setting on the keyer itself — and a subject in 4:2:0 with green spill in their hair has a hard limit no tool removes.

A Blurring the matte more heavily, which…: Blur softens the edge spatially within each frame but does nothing across time, so a crawling edge continues to crawl at a softer radius.

C Tightening the keyer's tolerance so that…: Tightening tolerance shrinks the matte and leaves a fringe of background, and the remaining boundary pixels still vary frame to frame.

D Adding a garbage matte around the…: A garbage matte is a valuable early step for uneven screens, but it cannot help at the subject's own edge, which is where the crawling is visible.

Lesson 62, p. 337

An editor needs to place a 3D object on a table so it stays put as the camera moves. The shot is a pan from…

Answer C: A pan has no parallax, so there is no positional information for the solver to recover the camera's path in space from.

Why: Point tracking gives position, planar tracking gives a whole surface and survives occlusion, and camera tracking needs real parallax and a rigid scene — so pick the cheapest one that solves the problem, and expect the composite to fail on matching rather than on the track.

A The scene is not rigid enough…: Newly revealed geometry does complicate a solve, and it is not what makes a tripod pan unsolvable for camera position.

B Tripod pans introduce rolling shutter skew…: Rolling shutter does degrade solves on fast pans and is a correctable nuisance rather than a fundamental obstruction.

D The tracker needs at least eight…: Feature count is a real practical constraint, but adding markers to the wall would not make a pure rotation solvable for position.

Lesson 63, p. 341

An editor patches out a light stand on a handheld shot, tracking the patch to the wall behind it. The track is accurate and the…

Answer A: The patch is too clean — it lacks the grain and noise of the surrounding footage, so it reads as a smooth area in a textured frame.

Why: Removal means covering something with pixels that plausibly belong — try reframing first, patch from the same frame on a locked-off shot and from another frame on a moving one, and always match grain, because a patch that is too clean is what gives a removal away.

B The mask edge is too hard…: A hard edge is a common fault worth fixing, but a well-feathered patch that is still noise-free remains visible as a smooth region.

C A planar track cannot hold on…: Planar tracking handles changing perspective well, and the question states the track is accurate, so the problem lies after the track.

D The patch was taken from a…: A real matching failure and one of the things to check, though it presents as a visible brightness step rather than as a patch that is subtly too clean.

Lesson 64, p. 346

An editor applies heavy spatial noise reduction to a low-light interview. The noise is gone and faces look waxy, with hair reduced to a smear…

Answer C: Noise is random between frames while the image is not, so averaging the same pixel over time cancels noise and leaves the detail intact.

Why: Temporal denoise averages the same pixel across frames so noise cancels and detail survives, spatial denoise averages within a frame so detail goes with the noise — remove chroma noise aggressively and luminance gently, denoise before grading, and re-add grain at the end.

A Temporal denoise operates only on the…: Temporal denoise acts on both channels; the asymmetry between chroma and luma is a choice you make in the settings rather than a property of the method.

B Temporal denoise runs at a lower…: Default strengths vary by tool and are not the mechanism — a heavy temporal setting still preserves detail that a heavy spatial one destroys.

D Temporal denoise uses motion vectors to…: Motion compensation is used by better temporal denoisers to decide where averaging is safe, but the noise cancellation itself comes from randomness over time rather than from tracking detail.

Lesson 65, p. 350

An editor applies a one-second cross-dissolve between two busy, detailed shots to soften a cut that felt abrupt. The result looks worse than the hard…

Answer B: Both images are visible at partial opacity throughout, so two detailed frames overlap into a second of visual noise the viewer cannot parse.

Why: A hard cut is right most of the time and a dissolve means time passed — the transitions worth learning are match cuts, whip pans and masked reveals, which you build from what is in the frame rather than drag from a bin.

A The dissolve consumed handles from both…: A real consequence of dissolves and a good reason to check handles, though it would show as unwanted content at the edges rather than as a muddy overlap.

C A one-second dissolve is far longer…: Length is worth correcting and changes what the transition means, but shortening it does not remove the overlap problem between two busy frames.

D Cross-dissolves re-encode both clips across the…: Encoding does allocate bits differently across a dissolve, and at sensible bitrates that is invisible compared with the overlap itself.

Lesson 66, p. 354

An editor generates proxy media for a whole project and playback is unchanged. What is the most likely explanation before concluding the machine is too…

Answer B: Generating proxies and using them are separate settings, and the toggle telling the editor to play the proxies has not been switched on.

Why: Check in order — proxies on, effects bypassed, scopes closed, playback resolution reduced, background renders finished, disk fast enough — because slow playback is almost always one specific effect or one missing toggle rather than an inadequate machine.

A Proxy generation is still running in…: Worth checking and it resolves itself, whereas a setting left off will not improve however long the editor waits.

C The proxy resolution was set too…: A real mistake that yields a smaller improvement than expected, but even half-resolution intra-frame proxies decode far more cheaply than long-GOP originals.

D Proxies only improve scrubbing performance rather…: Invents a distinction: when proxies are in use the editor plays them for both scrubbing and playback.

Module 8 · Delivery — the file that leaves the building

The module starts on p. 363.

MODULE 8 TEST

Module 8 test, Q1, p. 420
Renaming a .mkv file to .mp4 does not make it play. What does a remux do that renaming does not?
Answer D: It unpacks the streams and repacks them into the other box
Why: Renaming changes the label and not a byte of the contents. A remux, ffmpeg -i in.mkv -c copy out.mp4, takes the compressed streams out of one container and writes them into another without decoding anything, so it takes seconds and loses nothing. It fails only where a codec is not legal in the target container, and the repair is to re-encode that one stream and copy the rest. Moving the index is a separate flag, movflags faststart.

Module 8 test, Q2, p. 420
A 1080p25 export at 10 Mbit/s looks clean. The same material at 2160p25 and 10 Mbit/s smears. Why?
Answer B: There are four times as many pixels to describe with the same bits
Why: 1080p25 is about 51.8 million pixels a second and 2160p25 is about 207 million. Ten megabits across the first is roughly 0.19 bits per pixel, which is comfortable; across the second it is under 0.05. To hold the same quality you need around 40 Mbit/s. Comparing bitrates as bits per pixel rather than as raw megabits is the only fair comparison, and it is why people blame a 4K camera for an export setting.

Module 8 test, Q3, p. 420
A carefully grained grade turns to porridge after upload. What is the mechanism?
Answer A: Noise is random, so the encoder must spend real bits on it or discard it
Why: Compression works by not describing what it can predict. Noise is unpredictable by definition, so an encoder either spends a large share of its budget describing it or throws it away and spends the budget on blocks. This is why a light denoise before export can make an image look better after compression even though it looks softer before it, and why if you grain, you upload generously.

Module 8 test, Q4, p. 421
A video looks terrible ten minutes after publishing and fine the next morning, with no re-upload. What happened?
Answer C: Only the low rungs of the ladder had finished processing
Why: Your upload is source material for a ladder of renditions at different resolutions and bitrates. The platform publishes a low rung immediately so the video is watchable, and the higher ones arrive over the following minutes, or hours for a long 4K file. Check the quality menu: if it offers only 360p and 720p, processing has not finished. Re-exporting and re-uploading at this point solves nothing and costs a generation.

Module 8 test, Q5, p. 421

Why is a master rendered textless and at the full frame rather than at the delivery crop?

Answer B: Baked-in decisions belong to one destination and cannot be removed

Why: A master exists so that every future deliverable can be derived from a file rather than from a project that must still open, relink and find its fonts. A square crop with captions baked into the middle of the frame is a master from which no landscape version and no translated version can ever be made. Master clean, full frame and textless, then crop and caption at delivery, and render the stems while the project is still open.

Module 8 test, Q6, p. 421

You played the whole programme through the timeline's loudness meter and read minus fourteen integrated. ffmpeg on the exported file reports minus nineteen. What does…

Answer D: Something in the render path differed from the mix bus

Why: Both meters implement the same measurement, so a five-decibel gap is a real difference between what the mix bus was doing and what was written to the file. The usual causes are a bus excluded from the render, a limiter bypassed on export, or a track layout that collapsed differently. It is also the argument for measuring on the file rather than on the timeline: nobody receives a timeline.

THE LESSON CHECKS

Lesson 67, p. 369

Two people upload 1080p videos to YouTube at the same bitrate. One is a talking head against a plain wall and looks clean; the other…

Answer D: Rain and night footage contains far more change between frames, so the same number of bits per second has to describe much more information and runs out.

Why: The platform re-encodes everything, so upload a generously encoded file at the source frame rate and mix near -14 LUFS rather than louder.

A The rainy footage was probably shot…: Attributes the difference to the source codec, when both were re-encoded to the same delivery format regardless.

B Dark footage is compressed more aggressively…: Invents a content-aware policy instead of recognising that bitrate is consumed by change, not by brightness.

C The rainy video needed a higher…: Confuses temporal sampling with compression capacity — more frames would consume more bits, not fewer.

Lesson 68, p. 373

Renaming a .mkv file to .mp4 sometimes plays and sometimes does not. What is actually happening?

Answer C: Nothing inside the file changed, so it plays only where the player reads Matroska anyway; a real conversion needs a remux.

Why: A filename tells you the container and nothing about the codec, so repack with a remux rather than a rename, and check a file with ffprobe before you send it anywhere.

A The rename rewrites the file's identifying…: Confuses the extension with the file's internal header; renaming in a file browser touches neither the header nor the streams.

B Matroska stores its frames in a…: Invents a structural difference in frame ordering; both containers carry the same coded frames in the same order.

D MP4 requires its index at the…: Mixes up the faststart problem, which is real but concerns where the index sits in a genuinely valid MP4, not a renamed Matroska file.

Lesson 69, p. 378

Two shots are exported in the same 12 Mbit/s 1080p25 file. The interview is flawless; the confetti shot breaks into blocks. Why?

Answer A: Compression spends bits on what it cannot predict, and confetti is unpredictable in almost every pixel of every frame.

Why: Compare bitrates as bits per pixel per second rather than as raw megabits, and expect motion, fine detail and grain to need several times what a static interview needs.

B The confetti shot carries more pixels…: Blames pixel count when both shots are the same resolution; the difference is entirely in the content.

C Rate limits apply only to the…: Invents a rule about keyframes; rate control allocates across every frame, and a moving shot does get more keyframes but that is a consequence rather than the cause.

D Fast motion exceeds the temporal detail…: Confuses shutter and frame rate with compression; raising the bitrate visibly improves the confetti shot, which it could not do if the detail were already gone.

Lesson 70, p. 382

A thin red title on black looks soft and fringed after export while the identical title in white stays crisp. What explains the difference?

Answer D: Colour is stored at a quarter of the resolution of brightness, and a red edge on black is carried almost entirely by colour.

Why: Delivery formats keep one colour sample per four pixels and 256 levels per channel, which costs you saturated edges and smooth gradients — and a trace of added grain hides the banding that results.

A Saturated red sits near the edge…: Legal levels are a genuine delivery concern, but clipping crushes a value rather than softening an edge, and the white title carries the same hard luminance edge without suffering.

B Eight-bit quantisation leaves too few levels…: Points at bit depth, which produces banding across smooth gradients rather than softness at a high-contrast edge.

C The title was set at a…: Weight does matter for legibility after compression, but this does not explain why the identical weight in white survives the same export.

Lesson 71, p. 387

Uploading a 1080p edit on a 2160p canvas sometimes improves how the 1080p rendition looks. What is the actual mechanism, stated honestly?

Answer B: Higher-resolution uploads receive better codec and bitrate treatment across the whole ladder — undocumented behaviour that has changed before.

Why: Your upload is source material for a ladder of renditions you never see, so hand it something clean and uncompressed-twice, match the frame rate, and judge nothing until processing has finished.

A The upscale triggers the platform's detail…: Invents an enhancement step; no stage of the pipeline adds detail that was not in the upload.

C 4K uploads bypass the transcoding ladder…: Every upload is transcoded into renditions; nothing is ever served to viewers exactly as uploaded.

D The player defaults to the highest…: Describes rendition selection, which is driven by bandwidth and screen size, and would not explain why the 1080p rendition itself measures better.

Lesson 72, p. 391

A 25p edit exported with a preset called "YouTube 1080p" plays with an even stutter on every pan. What went wrong?

Answer C: The preset carried 29.97 fps, so frames were duplicated unevenly to pad 25 up to 30.

Why: Save a named render preset per destination with the frame rate taken from the timeline rather than the preset, because the wrong frame rate is the mistake that survives every other check.

A The preset's bitrate is too low…: A real cause of mush on moving shots, but blocking looks different from an even stutter and raising the bitrate would not remove this one.

B The preset wrote a variable frame…: Editors write constant frame rate on export; variable frame rate is a problem you import, not one you create at the render page.

D The preset specified 44.1 kHz audio…: Conflates an audio problem with a picture one; a sample-rate mismatch produces a slow drift, not a repeating judder on every pan.

Lesson 73, p. 396

A trim with `-ss 00:01:30 -t 60 -c copy` produces a file that starts about a second later than asked. Why?

Answer A: With the streams copied rather than re-encoded, the cut can only start on a keyframe, which may be a second or two away.

Why: ffmpeg never modifies its input, so inspecting, remuxing, converting a variable frame rate recording and measuring the loudness of a finished file are all free experiments.

B Placing `-ss` before `-i` performs an…: The position of `-ss` does change seeking behaviour, but with `-c copy` the keyframe constraint still applies wherever you put it; only dropping `-c copy` gives frame accuracy.

C Copying streams discards the source timecode…: Invents a timecode mechanism; the offset comes from the structure of the compressed frames, and the same shift happens in files carrying no timecode at all.

D The container rounds start times to…: Proposes a rounding rule that does not exist; the shift is to the nearest keyframe, which rarely falls on a whole second.

Lesson 74, p. 400

Four months after delivery a client asks for the same film with titles in Tamil. Which artefact makes that cheap?

Answer D: The textless master, because the new titles are laid over clean picture with nothing to relink.

Why: Render a full-frame, textless, intra-frame master plus stems at the end of every project, because deriving a new deliverable from a file always works and deriving one from a project only works while the application, fonts, plugins and media all still resolve.

A The graded H.264 web delivery, because…: Covering an old title is visible against any moving background, and it re-encodes an already-compressed file for a second generation.

B The project file, because the titles…: True only while the application version, fonts, plugins and media all still resolve — which is precisely the thing that fails four months later.

C The audio stems, because a dub…: Stems are needed for a dub and do nothing at all about text burnt into the picture, which is what was asked for.

Lesson 75, p. 404

An export fails at 43% on every attempt. What does the repeatability tell you, and what do you do?

Answer B: Something at that timecode is unreadable, and halving the render range repeatedly will find which clip.

Why: Render a sixty-second test from the hardest minute of the timeline and open it in two players, because gamma shifts, hardware artefacts, sync drift and file size all declare themselves in a minute and cost an evening after four hours.

A The machine exhausts memory once the…: Memory exhaustion is worth ruling out, but it depends on what else is running and rarely repeats to the same percentage on every attempt.

C The GPU is overheating after a…: Thermal failures move with ambient temperature and with what else is loaded, so they do not repeat to the same frame.

D The output has reached a container…: Modern MP4 has no practical limit at these sizes, and a size ceiling would move with the bitrate rather than sit at 43% of every attempt.

Lesson 76, p. 409

The timeline meter read −14 LUFS, but the rendered file measures −19. What is the most likely explanation?

Answer C: The render contained a different signal from the one the meter saw — a muted bus, a bypassed limiter, a different track layout.

Why: Watch the rendered file once at speed without stopping, then measure its duration, loudness, codec and first and last frame, because every verification you did on the timeline was of something nobody receives.

A Editor loudness meters report peak level…: Editor meters do compute integrated loudness to the same standard; the discrepancy is about what signal reached the render, not about which quantity was displayed.

B The AAC encoder normalises audio during…: Lossy audio encoding can shift true peak by a fraction of a decibel, but it does not renormalise the programme loudness by five.

D Integrated loudness is defined over a…: Describes how a meter accumulates, which is fixed by resetting and playing the whole timeline, and does not explain a file that differs from a correctly measured timeline.

Lesson 77, p. 413

An editor finishes a Resolve project, copies the entire project folder — footage, audio, graphics, exports — to an external drive, then wipes the laptop…

Answer A: Resolve keeps projects in an internal database rather than as a file inside the project folder, so copying the folder captured the media but not the edit.

Why: Learn JKL and three-point editing, name folders by date and version, and back up the project file separately from the footage.

B The proxies were not copied, so…: Treats proxies as part of the edit rather than as disposable preview files regenerated from the originals.

C The clips were relinked to new…: Describes an offline-media problem, which shows red clips on an intact timeline rather than no timeline at all.

D The project was saved in a…: Blames version incompatibility, which would produce an error on opening a project rather than the absence of one.

Examination

The paper starts on p. 423.

Examination, Q1, p. 424
A green screen was lit evenly and filmed on a phone. The key is still blocky along every edge. What is the limit?
Answer B: The phone recorded 4:2:0, so the matte is built from quarter-resolution colour
Why: A keyer builds its matte from colour. In 4:2:0 one colour sample covers a 2 by 2 block, so the mask carries a quarter of the picture's resolution and its edges come back stepped and unstable between frames. Lighting the screen well removes the other problems and cannot remove this one, because the limit is in the file. This is the single most common reason a key that looked achievable on set is not achievable in the edit.

Examination, Q2, p. 424
What does shooting 4K for a 1080p delivery actually buy in the edit?
Answer D: Room: punch-ins, reframes, the stabilisation crop and a vertical version
Why: Resolution is a count of pixels, not a grade of picture, and the platform will compress both. What the extra pixels buy is headroom in the frame: scaling to 200 per cent and still delivering full resolution, reframing a badly composed shot, absorbing the 5 to 15 per cent a stabiliser takes, and cropping 16:9 to 9:16 without going soft. The cost is storage, heat and a heavier timeline.

Examination, Q3, p. 424
Why is the image scaling behaviour in project settings described as expensive to change once an edit is under way?
Answer A: Changing it re-frames every clip whose size does not match the timeline
Why: Scale to Fit, Scale to Fill and Centre Crop decide what happens to any clip whose dimensions differ from the timeline. Change the setting after you have punched in on forty shots and all forty move, because their scaling was set relative to the old behaviour. It belongs in the ten minutes of setup before the first import, alongside frame rate, resolution, audio sample rate and cache location.

Examination, Q4, p. 425
Editing is described as a search problem. What follows practically from that?
Answer C: You cannot choose a better shot from material you never formed a memory of
Why: Every decision in a timeline is really the question of what should be here instead, and that question cannot be answered about material you cannot recall. Editors who skip the watch pass spend the week scrubbing, which is slow and also unreliable, because a shot's value is usually in something you cannot see at eight times speed. A written marker saying best answer re cost is findable; one saying good is not.

Examination, Q5, p. 425

For which material does transcript-first editing produce something coherent to listen to and nothing to watch?

Answer D: A process film where somebody's hands are the story

Why: The method works by letting the words decide the structure, which is exactly right for speech-led work and useless where the picture carries the meaning. A dance piece, an action sequence, a silent montage and a process film built on somebody's hands all fail the same way: the transcript produces an argument, and the argument is not what the piece was about.

Examination, Q6, p. 425

A file will not import into your editor. VLC plays it perfectly. What does that establish?

Answer B: The file is intact and the editor lacks the codec

Why: VLC plays things editors refuse, so a clean playback tells you the data is readable and the problem is on the editor's side. Check the codec with ffprobe; editors commonly refuse 10-bit HEVC, some AV1 variants and unusual pixel formats. The universal answer is to transcode to an intermediate such as DNxHR HQ or ProRes 422. If VLC also refuses it, the file itself is damaged and you are in repair territory.

Examination, Q7, p. 426

Footage sits on a working drive and on a second partition of the same drive. Why is that not two copies?

Answer C: One drive failure takes both of them

Why: The rule is two copies on two devices, because the failure being guarded against is a device failure. Two partitions are one piece of hardware with one set of bearings and one controller. A working drive plus a laptop's internal disk counts; a working drive plus its own second partition does not. Cards left unformatted until delivery are the cheapest third copy there is.

Examination, Q8, p. 426

Two people are talking and the camera crosses the line between two shots. What does the viewer experience?

Answer A: The scene stops reading as a conversation, because both look the same way

Why: Screen direction is how the frame encodes relationship. Keep every camera on one side of the line between two people and A always looks right while B always looks left, so the audience knows without thinking that they are looking at each other. Cross it and both face the same wall. The repairs are a neutral shot in between, a shot taken on the line, or a moving shot in which the camera is visibly seen to cross.

Examination, Q9, p. 426

A talking head is cut with thirty jump cuts and the speaker moves a great deal between takes. Why does the convention stop working here?

Answer B: The teleport is large, so the eye complains

Why: Audiences read a jump cut in a talking head as honesty about compression, which is why the convention works at all. It stops working when the subject has moved far between the two frames, because then the eye is tracking a face that has teleported, and when there are so many that the space is never allowed to exist. The usual repair is to punch in to 110 or 115 per cent on every second or third jump and to alternate with a cutaway.

Examination, Q10, p. 427

If the meaning of a shot is produced by the shot next to it, what does that give an editor?

Answer D: You can build events you never filmed

Why: Editing is juxtaposition rather than selection. Six fragments filmed in a different order over an hour are read as one continuous action, so a sequence can compress eight seconds of walking into two and the viewer reconstructs the rest. The same mechanism is what makes montage as argument powerful and dangerous: cutting two things together asserts a relationship without stating it, and if it would be a lie in words it is a lie in pictures.

Examination, Q11, p. 427

A politician's promise is cut directly to a shot of a closed factory. What is the honest objection?

Answer A: It asserts a relationship without stating it

Why: Montage as argument produces a third idea out of two images, and the claim it makes is never spoken, which is precisely what makes it easy to use for something untrue. The test is straightforward: if the juxtaposition would be a lie stated in words, it is a lie in pictures, and the fact that you never said it out loud is not a defence. Licensing and continuity are real questions and are not this one.

Examination, Q12, p. 427

In the working method for cutting a conversation, what happens before any picture decision?

Answer C: The radio edit, the breaths, and a bed of room tone

Why: Build the conversation as audio first: cut for meaning, move every join to a breath, and lay room tone under the whole scene so the joins have nothing to announce themselves with. If it does not hold with the screen off, no picture will fix it. Only then ask what the viewer should be looking at, and the answer is the listener at least as often as the speaker.

Examination, Q13, p. 428

Mirroring a shot horizontally to manufacture a reverse angle fails visibly in which case?

Answer D: Any frame containing text, a watch, a hair parting or a door handle

Why: A flip reverses everything in the frame, so any asymmetry gives it away: lettering of any kind, a clock, a watch on the usual wrist, a shirt buttoning the standard way, a parting, a ring, a steering wheel. In practice it is usable on plain-background close-ups and abstract b-roll and almost nowhere else. Check the frame carefully before committing, because the tell is invisible until somebody sees it and then permanent.

Examination, Q14, p. 428

What distinguishes a cold open that is craft from one that is bait?

Answer B: Whether the promise it makes gets paid

Why: Every technique that improves retention can make a piece more accessible to somebody who would have valued it, and can equally hold attention on something that will not repay it. The distinction is the payoff. A cold open that delivers the thing it promised is craft; one that uses the best moment as bait and delivers something else trains an audience to distrust you, and the fact that it works is not a justification.

Examination, Q15, p. 428

Why is a first fine cut reliably a fifth to a third too long?

Answer C: Material justified by the editor's knowledge feels slow to a stranger

Why: Every section is in the cut because you had a reason to put it there. The viewer does not have your reasons; they have only what is on screen. Material that is justified by your knowledge of the project rather than by its effect on a stranger is exactly the material that feels slow, which is why the twenty per cent exercise removes whole units rather than frames, and why the shorter version usually wins.

Examination, Q16, p. 428

What is the actual job of a timeline called parked or offcuts?

Answer A: It removes the finality that makes people fight to keep things

Why: Twice in a career you will retrieve something from it. The rest of the time its job is psychological and practical at once: dropping a sequence into a parked timeline costs five seconds and turns deletion into storage, so the argument about whether to keep it stops being an argument. It is also where the shorts, the social clips and the trailer come from later.

Examination, Q17, p. 429

Why is narration written after the picture edit rather than before it?

Answer B: Written first, it gets illustrated, and the film becomes a slideshow

Why: Write the script first and you spend the edit finding a shot for every sentence, which produces an essay with pictures attached. Cut the sequence with its natural sound, note where the viewer will have a question, and write one sentence into each gap. About a third of what you wrote then turns out to be describing the picture, and you cut it.

Examination, Q18, p. 429

Over a shot of workers arriving at a factory, which line is doing the job narration exists for?

Answer D: Four years ago there were nine hundred of them

Why: Words and pictures should do different jobs. The picture already shows workers arriving, so narration that describes it spends the only channel that can supply what the picture cannot, and patronises the viewer while doing so. A number, a consequence, a cost, a piece of history, something outside the frame: that is what narration is for.

Examination, Q19, p. 429

Why is a thirty-second vertical cut not simply a trimmed version of the long piece?

Answer A: The short has no context to lean on and has to be complete in itself

Why: The long piece can build: introduce people, establish context, hold a question open for two minutes. The short arrives to a viewer who has none of that, so it opens on the moment rather than the setup, delivers one idea and ends. It is a separate cut with its own spine, usually built around one answer or one demonstration, with whatever a stranger needs to know carried as a caption rather than as a spoken introduction.

Examination, Q20, p. 430

A track licensed CC-BY-NC is used under a video advertising the editor's own services. Why is that a problem?

Answer C: Advertising your own services is a commercial use

Why: The NC condition forbids commercial use, and a monetised video is commercial, as is a video advertising a business, and arguably so is a portfolio piece that gets you hired. If money is anywhere near the video, avoid NC entirely. Attribution is a separate condition of CC-BY, and Content ID is a separate system again that operates independently of whether your use was permitted.

Examination, Q21, p. 430

Somebody agreed to appear in an internal training video. The same interview is now wanted for a public advertisement. What follows?

Answer D: The scope has changed, so it needs a new yes

Why: Consent is scoped. Somebody who agreed to an internal training video did not agree to a public advertisement, and somebody who spoke about their health to a small community project did not agree to something indexed by search engines. Re-use outside the agreed scope needs a new agreement. The question bites hardest with vulnerable subjects, where the further test is whether they were in a position to weigh what consent meant.

Examination, Q22, p. 430

What does the phrase royalty free actually mean on a music library?

Answer B: There is no recurring per-play royalty

Why: It is a pricing model, not permission. The licence still has terms, it may cost money, and it may forbid exactly what you are doing: personal use where you needed commercial, one channel where you needed all platforms, one video where the licence was per project. Read the rest of the sentence, and keep a log with the track, the source, the licence, the required attribution string, the date and what you paid.

Examination, Q23, p. 431

Beethoven died in 1827. Why can a 2019 orchestral recording of his Fifth still be somebody's copyright?

Answer C: The recording is a separate work, with its own performers and label

Why: A public-domain composition is not a public-domain recording. The score is free; the performance, the engineering and the master are a new work with their own rights and their own term. This catches people constantly, and it is why Musopen exists: to supply recordings that are actually free rather than scores that are.

Examination, Q24, p. 431

Why can a film not be dubbed into another language from a flattened mix?

Answer A: The dialogue cannot be separated from the music and effects

Why: A dub needs everything except the speech, which is the music and effects stem, the M&E. Once the three busses have been summed into one file, the speech is inside the same signal as the room, the traffic and the music, and no processing separates them reliably. Render stems at the end of every project, at the same length and start timecode as the programme, and the future dub is possible at all.

Examination, Q25, p. 431

On a compressor set for dialogue, what should the gain reduction meter be doing?

Answer B: Moving 3 to 6 dB on loud words and back to zero between phrases

Why: The meter shows behaviour rather than settings, which makes it a better guide than any preset. Pinned at 10 dB and never releasing means the threshold is too low and the voice is being squashed flat. Never moving means the threshold is too high and nothing is happening. Moving only on the sibilants is what a de-esser does, which is a different job in a different band.

Examination, Q26, p. 432

A re-recorded line sounds pasted into the scene. What is matched first?

Answer D: The noise floor, by laying the scene's room tone under it

Why: A perfectly silent line in a scene with a fan running is a hole, and the ear interrogates everything else about the line because of it. Lay the scene's room tone under the replacement and half the problem disappears before you touch a filter. Then match the EQ, then add far less reverb than sounds like reverb, and finally roll off some top if the original was further away.

Examination, Q27, p. 432

Why is narration over location footage deliberately not matched to the room in the picture?

Answer A: A narrator is understood to be outside the scene

Why: Adding the warehouse's reverb to a voice-over makes it sound like somebody standing in the warehouse, and a narrator is not standing anywhere. Narration belongs dry, close and clean. Match the room only for speech that is meant to be in the scene: a replaced line, a line lifted from another take, a piece of dialogue re-recorded afterwards.

Examination, Q28, p. 432

You turn the mix down until it is barely audible and the dialogue disappears before the music does. What does that tell you?

Answer C: The balance is wrong

Why: Turning the level down strips away everything the mix is not actually carrying, so what survives is what it is built on. If the speech goes before the music, the music is too loud, and no meter will tell you that as clearly. It is the most reliable free balance test there is, and it takes ten seconds on whatever you already have.

Examination, Q29, p. 433

Why does putting a heavy contrast node before the balance node change what the balance node is able to do?

Answer D: A node works on what it receives, and the shadows are already crushed

Why: Each node operates on the values it is handed. Crush the shadows first and the balance node has fewer distinct values left in them, so the correction it applies is a different correction. This is the same principle as correction before grade, applied inside one shot: subtractive and corrective operations first, creative operations after, and name every node so you can find them in a week.

Examination, Q30, p. 433

What makes the still store the feature that turns shot matching from an hour into five minutes?

Answer B: It carries the whole node tree, and it wipes against another shot

Why: A grabbed still holds the frame and the entire grade. Middle-click applies that grade to another clip, so every shot from the same setup starts from the hero's correction rather than from nothing. Dragging across the viewer wipes the still against the shot you are matching, which lets you compare two images directly instead of comparing an image to a memory that has already re-adapted.

Examination, Q31, p. 433
Matching a shot to a graded hero shot, what is compared first?

Answer C: The black point, on the waveform

Why: Black point, then white point, then the mid-tone level of a common element, then balance on the parade, then saturation on the vectorscope, and the faces last. The order matters because each step changes what the next one is reading. Judge the result at playback speed: anything you notice at normal speed is a genuine mismatch, and anything you can only see by pausing is not worth more of your evening.

Examination, Q32, p. 434
An automatic shot match pulls a close-up of a face green when matched to a wide of a green field. Why?

Answer A: It matches overall distributions rather than the subject

Why: The algorithm compares the statistics of two whole frames. A frame that is mostly field is mostly green, so bringing the close-up's distribution toward it pushes the face green. That is a predictable failure rather than a bug. Use the tool as a first pass, then correct it by hand on a split wipe, which is where you are comparing the thing you actually care about.

Examination, Q33, p. 434
What is the mark of a good power window?

Answer B: The viewer never notices it

Why: A hard-edged window reads as a shape drawn on the picture. Set the softness high enough that you cannot see where it ends, then raise the adjustment inside until it does something. Almost every window in professional work is softer and subtler than a beginner's, and if somebody can see that you brightened a face, it is either too strong or too hard.

Examination, Q34, p. 434
A flat horizontal line along the top of the waveform means what?

Answer D: A region of the picture is clipped and holds no detail

Why: A flat plateau at the top is a region that has gone to maximum with nothing left in it. A small amount is normal and often intentional: a light source in frame, a reflection off chrome. A wide flat plateau means a face or a sky has lost its detail, and no grade recovers it, because the values were never recorded. The same reading at the bottom is crushed shadows.

Examination, Q35, p. 434
Flat, evenly lit footage is graded toward a moody film look and comes back dark and murky rather than dramatic. Why?

Answer A: Contrast between light and shadow has to be in the room

Why: A grade redistributes the light that was recorded. If the room was lit evenly, there is no separation between a bright side and a dark side to bring out, so pulling the whole image down produces darkness rather than drama. The other options describe real failures with other causes, and a strong look applied to a weak image tends to make its faults more visible, not less.

Examination, Q36, p. 435
When should a name super appear?

Answer C: Two or three seconds after they start speaking

Why: The viewer needs a beat to register a new face before they can absorb a name, so a super arriving on the first frame of speech means both are missed. It also belongs on the first substantial line rather than the first appearance: somebody who walks through a shot at minute one and speaks at minute three is supered at minute three, and never over the end of a sentence that matters.

Examination, Q37, p. 435

The second line of a name super should carry what?

Answer D: Why their view matters

Why: Worked on the scheme for eleven years tells a viewer why to listen. Deputy Assistant Director, Programmes tells them nothing they can use in the two and a half seconds the super is on screen. The super's whole job is to answer who is this at the moment the question arises, and then leave; everything else about it is decoration, including the animation.

Examination, Q38, p. 435

Why are credits sometimes a contractual obligation rather than a courtesy?

Answer B: Some licences specify an attribution string as a condition of use

Why: CC-BY requires attribution in the manner specified, and several free music tiers require a particular credit line and a per-track code. Those are conditions of the licence, so failing to include them is a breach rather than an oversight. Take the exact strings from the licence log you kept during the edit, which is why the log is kept in the project's docs folder rather than in your memory.

Examination, Q39, p. 435

A speed ramp built in the Retime Speed view jolts at every step. What fixes it?

Answer C: Easing the keyframes on the Retime Frame graph instead

Why: The Speed view shows the speed value as a series of steps, so a ramp built there changes rate abruptly at each one. The Frame view graphs which source frame plays at which timeline moment, and easing its keyframes makes the rate of change itself gradual, which is what a ramp is. Optical flow and motion blur improve the frames; they do not smooth the curve.

Examination, Q40, p. 436

Why render a Blender title to a PNG image sequence rather than straight to a video file?

Answer A: A crash at frame 800 leaves the first 799 intact

Why: A long render that fails partway through has produced nothing if it was writing a single video file, and eight hundred usable frames if it was writing a sequence. ProRes 4444 and WebM both carry alpha perfectly well, so that is not the reason. Render with a transparent background, bring the sequence into the edit, and composite over the picture.

Examination, Q41, p. 436

Why record a screen at the delivery resolution rather than at the display's full resolution?

Answer B: Capturing 4K and downscaling halves the size of every interface element

Why: A 12-pixel menu label captured at 4K and delivered at 1080p becomes six pixels, which is unreadable on a phone. Set the display to 1920 by 1080 before recording, or record a 1080p region, or raise the interface scaling so the capture is 1080p of an enlarged interface rather than a shrunken view of a large desktop. Then punch in on anything that matters, with an eased move rather than a cut.

Examination, Q42, p. 436

Why is a truncated bar-chart axis a stronger deception on video than on a page?

Answer D: The viewer cannot go back and check the axis

Why: A reader can study a chart, notice the axis label and adjust. A viewer has four seconds, somebody talking over it, and no ability to pause in most players. Starting a bar chart anywhere but zero makes a three per cent difference look like a doubling, and on video the only defence the viewer has is you not doing it. The same applies to area, volume and undeclared denominators.

Examination, Q43, p. 437

A carefully rotoscoped arm moving quickly still looks cut out. Why?

Answer A: The mask edge is hard, and a moving limb's real edge is semi-transparent

Why: Motion blur means the subject's edge is genuinely spread across several semi-transparent pixels, and a bezier shape has no way to express that: it is either inside or outside. The same limit applies to hair, which is thousands of strands a smooth outline turns into a helmet, and to glass, smoke and net curtains, which are not shapes at all. When a shot contains any of these, either the effect changes or the shot does.

Examination, Q44, p. 437

Filming five seconds of the empty set at the end of every setup enables which technique?

Answer C: A difference matte against a clean plate

Why: A difference matte compares the shot against a frame of the same set with nothing in it: whatever differs is the subject. It costs five seconds on the day and works well on a locked-off camera, failing on any camera movement, changing light or heavy noise. The same clean plate is also the reliable professional answer to object removal, which is why it is worth the five seconds twice over.

Examination, Q45, p. 437

Why does a tracker slide along a straight edge and lock onto a corner?

Answer D: Along an edge, many positions match equally well

Why: A tracker searches the next frame for the pattern it found in this one. A patch of straight edge looks the same at every position along that edge, so the match is ambiguous and the point drifts. A corner is distinctive in both directions, so there is one best answer. The same reasoning rules out repeating patterns, soft gradients, anything out of focus and anything with motion blur.

Examination, Q46, p. 437

A pan from a tripod cannot be solved for camera position. Why not?

Answer B: There is no parallax: the camera never moves through space

Why: A three-dimensional solve works out depth from the way near objects shift more than far ones as the camera moves. A pure rotation gives every point the same apparent motion regardless of distance, so there is no depth information to recover and only the rotation can be solved. For a camera track, a slow lateral move gives far more usable parallax than a pan from the same spot.

Examination, Q47, p. 438

You ask for a one-second dissolve and the editor quietly makes it shorter. Why?

Answer C: The clips had no handles beyond their in and out points

Why: A dissolve centred on the cut needs unused material from both clips to overlap, exactly as a roll trim does. If either clip has been trimmed to the bounds of its source there is nothing there, so the editor either refuses or shortens the transition without saying so. It is the same reason to keep shooting a few seconds before and after every take.

Examination, Q48, p. 438

Playback has dropped to eight frames a second. What is checked first?

Answer A: Whether proxies exist and the toggle to use them is on

Why: Long-GOP source files are expensive to decode, and proxies remove the problem entirely. Two settings are involved and forgetting the second is extremely common: the proxy media has to exist, and the toggle to play it has to be on. Only after that is it worth bypassing effects one at a time, closing the scopes, reducing playback resolution, and finally suspecting the drive.

Examination, Q49, p. 438

A locked-off shot of a street has people walking through it and you want the street empty. Which technique?

Answer B: A per-pixel median across many frames

Why: If the camera is locked off and everything unwanted keeps moving, then each pixel shows background most of the time, so the median value across many frames is the empty street. It is how remove-the-tourists features work, and it requires both conditions: a static camera and objects that do not stop. Somebody standing still is background as far as the median is concerned.

Examination, Q50, p. 438

A file plays in VLC and refuses to play in Safari and most browsers. Which setting is the first suspect?

Answer D: The pixel format, which should be yuv420p

Why: A file encoded as yuv444p or yuv422p is perfectly valid and plays in VLC, and browsers and QuickTime will not touch it. Adding -pix_fmt yuv420p to the ffmpeg command, or choosing a standard delivery preset in the editor, fixes it. It is the classic works-on-my-machine export, and it costs one flag.

Examination, Q51, p. 439

A video on your own web page shows a long spinner and then plays normally. What fixes it?

Answer A: Moving the index to the front of the file

Why: An MP4's moov atom says where every frame is, and encoders often write it at the end because that is when they know the layout. A file with the index at the end cannot start playing until it has fully downloaded. ffmpeg -c copy -movflags +faststart rewrites it in seconds with no quality loss, and most editors have a web optimised checkbox that does the same thing.

Examination, Q52, p. 439

ffmpeg -ss 00:01:30 -i in.mp4 -t 60 -c copy out.mp4 starts about a second later than asked. Why?

Answer C: With -c copy the cut has to land on a keyframe

Why: Stream copy does not decode anything, so it cannot construct a frame that was stored as a difference. The cut therefore begins at the nearest keyframe, and in a long-GOP file those are one to two seconds apart. For a rough extract that is fine; for a frame-accurate cut, drop -c copy and let it re-encode, which costs time and gives you the exact frame.

Examination, Q53, p. 439

A render fails at the same percentage every time. What is the efficient diagnosis?

Answer D: Bisect the render range until one clip is isolated

Why: A failure at a fixed point is telling you that something at that timecode is unreadable: a partially copied file, a corrupt tail, an unsupported still, a missing font. Render the first half; if it succeeds the fault is in the second, and four or five renders isolate one clip. Then transcode that clip to ProRes, which either gives you a clean replacement or fails at the same frame and tells you the media itself is damaged.

Examination, Q54, p. 440

A 25p timeline exported through a preset named YouTube 1080p judders on every pan. What happened?

Answer B: The preset carried 29.97, so frames were duplicated

Why: Presets often carry a frame rate, and the wrong one is the mistake that survives every other check. Rendering 25p as 29.97 means inventing about five frames a second by duplication, and duplication under motion reads as a stutter that no bitrate repairs. Set the frame rate from the timeline every time, and confirm it on the finished file with ffprobe before anything leaves.

Examination, Q55, p. 440
Somebody copied their whole footage folder to an external drive and believes their Resolve edit is backed up. What is wrong?
Answer C: Resolve keeps projects in a database, not in the folder
Why: Resolve stores projects in an internal database rather than as a loose file beside the footage, so copying the media folder backs up none of the edit. Export the project as a .drp at the end of every session, or export the whole database. The file is a few megabytes, which is also why the off-site copy in the 3-2-1 rule is the project, the licence log and the master rather than the rushes.

Examination, Q56, p. 440
What makes three-point editing faster than dragging a clip onto the timeline and trimming it?
Answer A: Three marks define the fourth, so there is no guess to correct
Why: Mark in and out on the source, mark in on the timeline, and press insert or overwrite: the fourth point is implied, so the clip lands exactly where it belongs the first time. Dragging is two operations and a guess, and the guess is then corrected by eye. With JKL and the split key this is most of the reason a professional assembly of an hour-long interview takes twenty minutes rather than two hours.

MAKE THINGS

Editing Video

Cuts, sound, colour and export — taught on DaVinci Resolve, which costs nothing.

WHAT IT TEACHES	A working course in the craft of editing.
WHAT IS INSIDE	Eight modules and 77 lessons, 28 figures drawn from data, eight case studies, eight module tests, a 56-question examination, and every answer with the reason behind it.
LICENCE	
THE COURSE	Free to read, with the lessons, the films and the examination, at **addaly.com/learn/video-editing**

Addaly

First edition, September 2026 · build 62a26075

Scan to open the course