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

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.

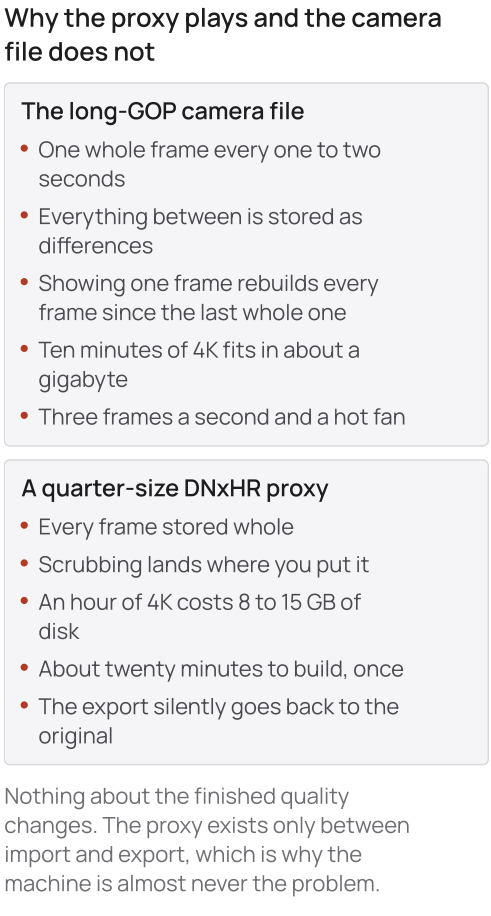
Here is the mechanism, and it is worth understanding once because it explains half of what follows in this course.

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



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.

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

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.

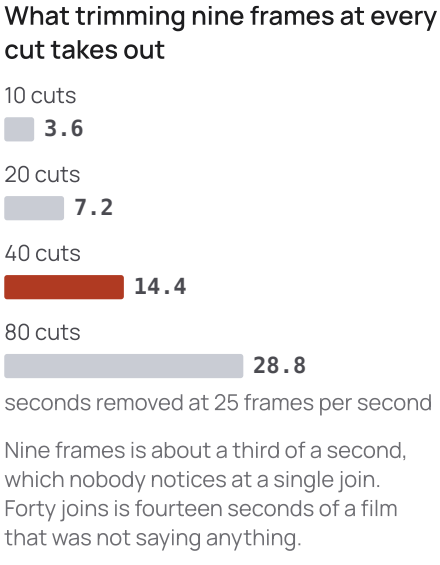
Almost always the problem is length. Nearly every shot is held about two seconds past the moment it stopped saying anything. You cannot see it because you already know what is coming next. The viewer does not, and they are not being patient on your behalf.

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



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.

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

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.

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	CC BY-NC-SA 4.0. Copy, print, share and adapt it for any non-commercial purpose, with credit, under the same licence. There is no paid edition.
THE COURSE	Free to read, with the lessons, the films and the examination, at **addaly.com/learn/video-editing**

Scan to open the course

Addaly

First edition, September 2026 · build 62a26075