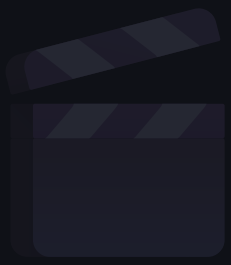

AUDIO & VIDEO PRODUCTION

Video Editing Fundamentals

DaVinci Resolve's Page-Based Workflow

Cuts, Trims, Transitions & Titles

Color Correction & Grading · Fairlight Audio

Capstone: Raw Footage to Final Export

Philip Osztromok

Generated with Claude

Table of Contents

Video Editing Fundamentals — 9 Chapters

OBS Studio's own capstone closed with a recorded file described as "exactly the raw footage the still-outstanding Video Editing Fundamentals course's own capstone will import and edit." This course is that next step — starting with why DaVinci Resolve, specifically, is the tool it teaches.

What Creating YouTube Content Already Covered

Creating YouTube Content's own Editing Software chapter already compared DaVinci Resolve, Premiere Pro, CapCut, and Final Cut at a high level — which tool suits which situation, broad strengths and trade-offs. This course doesn't repeat that comparison. It goes deep specifically on actually using DaVinci Resolve, the tool that chapter already pointed to.

Why DaVinci Resolve Specifically

DaVinci Resolve is genuinely professional-grade — used in real Hollywood film and television post-production pipelines, not merely a tool scaled down for hobbyist YouTubers. It's also genuinely free at a full, capable tier, unlike Premiere Pro's subscription-only model. A paid Studio version exists with additional features, but the free version alone is professionally capable on its own.

What This Course Actually Teaches

The actual workflow: importing footage, editing, color correction and grading, audio, and final delivery. Premiere Pro comparison notes are woven throughout rather than treated as a separate, dedicated comparison — DaVinci Resolve is this course's own primary focus, not one option among several being weighed.

Setting Expectations: A Steeper Learning Curve Up Front

Resolve's own page-based workflow (Chapter 2) is genuinely different from a single-window application like Premiere Pro or CapCut — a real, worthwhile adjustment period for anyone coming from a simpler tool. This pays off with meaningfully more capability once past that initial learning curve, rather than being complexity for its own sake.

ASPECT	DAVINCI RESOLVE (FREE)	PREMIERE PRO
Cost	Free, full capability	Subscription-only
Industry use	Real film/TV post-production	Widely used, especially in video/broadcast
Interface	Page-based (Ch.2)	Single unified timeline window
Color tools	Deep, signature strength (Ch.6)	More basic, historically

WHERE THIS COURSE'S OWN FOOTAGE COMES FROM

The MKV file produced by OBS Studio's own capstone is exactly what gets imported at the very start of this course's own workflow — the two courses connect directly, one producing the raw material the other actually edits.

"FREE" DOESN'T MEAN STRIPPED-DOWN HERE

It's tempting to assume that since Resolve is free, it must be a deliberately limited tool designed to upsell a paid tier. The free version is genuinely used in real professional post-production pipelines, unlike many "free" tools built specifically to nudge users toward paying. The actual free/paid split in Resolve is mostly about enterprise-scale collaborative features — multi-user project sharing, some advanced noise reduction and resolution tools — not crippling the core editing, color, and audio experience most people would ever actually need.

Hands-On Exercises

EXERCISE 1

A friend asks why this course spends an entire chapter on "why DaVinci Resolve" when Creating YouTube Content already compared several editors. Using this chapter's own material, explain what this chapter is actually adding.

View solution

EXERCISE 2

A colleague assumes that since DaVinci Resolve's free version has no cost at all, it must be missing important features compared to paid editors. Using this chapter's own warning box, explain why this assumption doesn't hold up.

View solution

EXERCISE 3

A beginner opens DaVinci Resolve for the first time coming from CapCut and feels immediately overwhelmed by the unfamiliar interface. Using this chapter's own material, explain what's happening and why it's worth pushing through.

View solution

CHAPTER 1 QUICK REFERENCE

- This course builds on, rather than repeats, Creating YouTube Content's own editor comparison — going deep on **using** DaVinci Resolve specifically
- DaVinci Resolve is **professional-grade** (real film/TV pipelines) and **genuinely free** at a fully capable tier
- Premiere Pro comparison notes are woven throughout — Resolve is the **primary focus**, not one of several options
- Resolve's own **page-based workflow** has a real learning curve coming from a simpler tool, but pays off in capability
- The free/paid split is about **enterprise collaboration features**, not crippling the core editing experience

Chapter 1 flagged a real learning curve without explaining it concretely. This chapter is that concrete explanation — the seven pages Resolve organizes its entire workflow around, and how that compares to Premiere Pro's own genuinely different design.

Seven Pages, Seven Dedicated Environments

Media, Cut, Edit, Fusion, Color, Fairlight, Deliver — each one a full, dedicated workspace for one part of the editing process, switched between via tabs at the bottom of the window, rather than one single window trying to accommodate every task at once.

Media Page

The entry point for any project — importing and organizing footage. This is the first stop for any project, with its own full treatment in Chapter 3.

Cut vs. Edit Page

A genuinely unusual design choice: two separate editing pages rather than one. **Cut** is a fast, simplified editing environment built for quick assemblies and simple cuts. **Edit** is the full, traditional multi-track timeline editor with deeper control. Cut is optional — Edit page alone covers everything Cut does and more — but Cut exists specifically for speed on simpler projects where Edit's own greater depth isn't actually needed.

Fusion, Color, Fairlight, Deliver Pages

Each gets its own dedicated chapter later in this course: **Fusion** — node-based compositing and motion graphics (a first look in Chapter 5). **Color** — professional color grading (Chapter 6). **Fairlight** — full audio post-production (Chapter 7). **Deliver** — export and rendering (Chapter 8).

Premiere Pro's Contrasting Philosophy

Premiere Pro instead uses a single, unified window — panels for effects, timeline, color, and audio all exist within one interface, rearranged via panel layout rather than switched via dedicated top-level pages. Neither approach is objectively better; this is a genuine design philosophy difference, not a feature gap between the two.

TASK	RESOLVE'S OWN PAGE	PREMIERE'S OWN APPROACH
Importing/organizing	Media page	Project panel
Quick assembly	Cut page	Same timeline, simplified view
Full editing	Edit page	Timeline panel
Color grading	Color page (dedicated)	Lumetri Color panel
Audio	Fairlight page (dedicated)	Requires Audition for equivalent depth

THIS IS THE LEARNING CURVE CHAPTER 1 MENTIONED

Chapter 1's own "steeper learning curve up front" is exactly this page structure — a genuinely different top-level organization than a single-window editor, now explained concretely rather than left as an abstract warning.

CUT AND EDIT AREN'T REDUNDANT — THEY TRADE DEPTH FOR SPEED DIFFERENTLY

It's tempting to assume that since both Cut and Edit let you edit a timeline, one of them must be pointless. They solve genuinely different needs: Cut trades some depth for speed — ideal for quick social clips or fast assemblies — while Edit trades some speed for full professional control. An editor working on a quick, simple cut might genuinely prefer Cut's own streamlined tools; a complex, precise edit genuinely benefits from Edit's own deeper toolset. The right choice depends on the actual task at hand, not an assumption that one page must be strictly better than the other.

Hands-On Exercises

EXERCISE 1

You need to quickly trim a handful of clips into a rough sequence for a short social media clip, with no need for advanced multi-track editing. Which page from this chapter fits best, and why?

View solution

EXERCISE 2

An editor working on a complex, multi-layer project with precise timing needs argues Cut page is "useless since Edit page can do everything it can and more." Using this chapter's own warning box, explain what this argument misses.

View solution

EXERCISE 3

Explain, in your own words, why Resolve's page-based structure and Premiere Pro's single-window structure are described in this chapter as a design philosophy difference rather than one tool simply having more features than the other.

View solution

CHAPTER 2 QUICK REFERENCE

- Seven pages: **Media, Cut, Edit, Fusion, Color, Fairlight, Deliver** — each a dedicated environment for one part of the process
- **Cut** — fast, simplified editing for quick assemblies; **Edit** — the full, traditional multi-track timeline
- Cut and Edit aren't redundant — they trade **depth for speed** differently, matched to the actual task
- Premiere Pro uses **one unified window** instead of dedicated pages — a genuine design philosophy difference, not a feature gap

Chapter 2 named the Media page as the entry point for any project, without going further. This chapter is that full treatment — how footage actually gets imported, organized, and made smooth to edit with.

The Media Pool

The **Media Pool** is Resolve's own central repository holding every imported clip for a project. The concrete starting point for this entire course: the MKV file produced by OBS Studio's own capstone gets imported here first, becoming this project's own raw material.

Bins: Organizing Footage Within a Project

Bins are folders within the Media Pool itself — organizing footage by shot type, date, or scene, entirely separate from the operating system's own underlying file structure. A genuinely practical habit for any project with more than a handful of clips, well before disorganization becomes a real problem.

Proxies: Editing Smoothly on Lower-Power Hardware

A genuinely important concept for anyone recording high-resolution footage (per OBS Studio's own recording settings) on a machine that struggles to play it back smoothly while editing. A **proxy** is a lower-resolution, easier-to-decode stand-in file, generated automatically from the original footage, used only during editing. When exporting the final file (Chapter 8's own Deliver page), Resolve automatically switches back to the original, full-quality footage — the proxy is purely a working convenience, never present in the final output.

Generating Proxies

Resolve's own built-in proxy generation works by selecting clips in the Media Pool and choosing to generate optimized media at a chosen lower resolution. This is a background process that can take real time depending on footage length and machine power — best started before actually beginning to edit, not partway through a session.

Metadata & Sorting

Basic clip information — resolution, duration, codec — is visible directly in the Media Pool, genuinely useful for quickly identifying content without needing to preview every single clip individually.

CONCEPT	DOES	WHY IT MATTERS
Media Pool	Holds every imported clip	The project's own central footage repository
Bins	Folders within the Media Pool	Real organization for any non-trivial project
Proxies	A lower-res stand-in used only while editing	Smooth editing on underpowered hardware, no quality loss in the final export

WHERE THIS CHAPTER'S OWN WORKFLOW ACTUALLY STARTS

The MKV file remuxed to MP4 in OBS Studio's own Chapter 4 is exactly what gets imported into the Media Pool as the very first step of this chapter's own workflow — the two courses' own capstones connect directly at this exact point.

PROXIES DO NOT LOWER YOUR FINAL EXPORT QUALITY

It's a genuinely common point of confusion to assume using proxies means the final exported video will be lower quality. This isn't true when configured correctly: Resolve automatically relinks to the original, full-resolution footage for the final render at export time (Chapter 8), regardless of what resolution was used to edit smoothly. Proxies exist purely to make the editing experience itself smoother on underpowered hardware — an unfounded fear about quality loss can lead someone to avoid a genuinely useful feature for no real reason.

Hands-On Exercises

EXERCISE 1

Your computer struggles to play back high-resolution footage smoothly while editing, causing choppy playback in the timeline. Using this chapter's own material, explain the feature that solves this, and confirm whether it affects your final exported video's quality.

View solution

EXERCISE 2

A new editor avoids using proxies entirely because they're worried the final video will look worse. Using this chapter's own warning box, explain why this concern is unfounded.

View solution

EXERCISE 3

A project has 60 clips from three different filming sessions, all sitting unsorted in the Media Pool. Using this chapter's own material, explain a practical organizational approach.

View solution

CHAPTER 3 QUICK REFERENCE

- The **Media Pool** holds every imported clip for a project — the entry point for OBS Studio's own recorded footage
- **Bins** organize footage within a project, separate from the OS's own file structure
- A **proxy** is a lower-resolution stand-in used only during editing — **never** present in the final export
- Generate proxies **before** editing begins, since it's a real background process that takes time
- Using proxies has **no effect** on final export quality — Resolve automatically relinks to the original footage at Deliver time

Chapter 3 got footage imported and organized. This chapter is where actual editing begins — the Edit page's own timeline, and the operations used to cut it together.

The Timeline: Tracks and Clips

Video and audio tracks stack similarly to a concept already familiar from elsewhere — Raster Editing Fundamentals' own layers, or OBS Studio's own Scene/Source stacking. Here, clips are placed along tracks in time rather than space. The **playhead** marks the current position being viewed or edited.

The Razor Tool: Splitting a Clip

The **Razor** tool cuts one clip into two separate clips at the playhead's own position — the most basic editing operation, foundational to everything else in this chapter. Premiere Pro calls the identical tool the Razor (or Blade) tool — essentially the same name and function.

Three Kinds of Trim: Ripple, Roll, Slip

A **Ripple** edit trims a clip's own edge and automatically shifts every subsequent clip on the track to close the resulting gap. A **Roll** edit adjusts the cut point between two adjacent clips simultaneously — lengthening one while shortening the other by the same amount, without moving anything else on the timeline. A **Slip** edit changes which portion of the source footage is shown within a clip's own existing duration and position on the timeline, without changing its length or position at all. Three genuinely different operations, easily confused since all three involve "adjusting" a clip in some sense.

Premiere Pro Terminology Mapping

Ripple, Roll, and Slip are used with the exact same names in Premiere Pro — a rare case of near-total terminology parity, unlike some of this course's own earlier comparison points. Worth calling out specifically since it's the exception in this course, not the rule.

EDIT TYPE	WHAT MOVES	WHAT STAYS FIXED
Ripple	Every subsequent clip shifts to close the gap	The trimmed clip's own remaining content
Roll	The shared cut point between two clips	Everything else on the timeline
Slip	Which source footage is shown	The clip's own length and position

THE THIRD TIME THIS EXACT MENTAL MODEL HAS APPEARED

Video/audio tracks stacking clips is genuinely the third time this site's own courses have used the identical "things arranged in an ordered stack, order matters" mental model — layers in Raster Editing Fundamentals, Sources in a Scene in OBS Studio, and now tracks on a timeline here. Reusable across three completely different pieces of software.

RIPPLE ISN'T JUST "THE NORMAL WAY TRIMMING WORKS"

It's easy to assume Ripple is simply how trimming always behaves, and be genuinely surprised when a simple trim unexpectedly shifts every other clip on the track. Ripple's own automatic gap-closing behavior is a deliberate, powerful feature for maintaining a tight edit with no gaps — but it can also produce an unwanted cascading shift if the editor didn't intend for later clips to move at all. Knowing which trim mode is actually active before trimming avoids this surprise entirely.

Hands-On Exercises

EXERCISE 1

You want to trim a few frames off the end of one clip and add them to the beginning of the very next clip, adjusting the shared cut point between them without moving anything else on the timeline. Which edit type from this chapter fits, and why?

View solution

EXERCISE 2

An editor trims the end of a clip expecting only that one clip to change, but every clip after it on the timeline unexpectedly shifts earlier. Using this chapter's own warning box, explain what likely happened.

View solution

EXERCISE 3

You have a clip on the timeline at exactly the right length and position, but you realize the wrong section of the original source footage is being shown within it. Which edit type from this chapter fits, and why wouldn't Ripple or Roll be appropriate here?

View solution

CHAPTER 4 QUICK REFERENCE

- The **Razor** tool splits one clip into two at the playhead — identical in name and function to Premiere Pro's own tool
- **Ripple** — trims and shifts everything after to close the gap; **Roll** — adjusts a shared cut point, nothing else moves; **Slip** — changes which source footage shows, length/position unchanged
- Ripple/Roll/Slip use the **exact same names** in Premiere Pro — a rare case of full terminology parity
- Tracks stacking clips is the **same ordered-stack mental model** as layers (Raster Editing Fundamentals) and Scenes/Sources (OBS Studio)
- Know which trim mode is active **before** trimming — Ripple's automatic gap-closing can cause an unwanted cascading shift

Chapter 4 covered cutting and trimming clips together. This chapter adds visual polish on top — transitions between clips, and text titles — plus a first glimpse of the tool waiting when basic titles aren't enough.

Cross-Dissolves: The Basic Transition

Dragging a transition onto the boundary between two clips applies it there directly. A **cross-dissolve** blends the outgoing clip into the incoming clip over a short duration — conceptually the same idea as OBS Studio's own Fade transition between Scenes, just applied to clips already sitting on a timeline rather than live Scenes being switched in real time.

Other Basic Transitions

A **Dip to Color/Black** fades through a solid color rather than directly between two clips, often used to signal a scene change or a time skip. A **Wipe** has one clip visually push the other out. Each one signals a slightly different kind of narrative break to the viewer, not just a stylistic preference between otherwise interchangeable effects.

Basic Text Titles

The Edit page's own built-in title tool handles simple text overlays directly — font, size, position, and basic animation presets. Fast, and genuinely sufficient for straightforward captions, lower-thirds, or simple title cards.

A First Look at the Fusion Page

When basic titles aren't enough, the **Fusion** page offers node-based compositing for genuinely custom motion graphics, animated logos, or complex multi-element titles. This is only a brief preview here, not a deep dive — Fusion's own node-based approach is a fundamentally different way of building visual content than a simple text tool, worth knowing exists even before learning it in real depth.

Choosing Basic Titles vs. Fusion

A genuine decision, similar in spirit to Chapter 2's own Cut-vs-Edit page choice: a simple lower-third or caption doesn't need Fusion's own added complexity; a custom animated intro or branded graphic genuinely benefits from it.

NEED	RIGHT TOOL
A simple caption or lower-third	Edit page's own built-in title tool
A custom animated logo or intro	Fusion page
A blend between two clips	Cross-dissolve
A scene change or time skip	Dip to Color/Black

THE SAME UNDERLYING IDEA, APPLIED DIFFERENTLY

A cross-dissolve here and OBS Studio's own Fade transition are genuinely the same underlying concept — blending from one thing to another over time — just applied differently: OBS's Fade blends between live Scenes in real time, while this cross-dissolve blends between clips already sitting on a fixed timeline.

MORE TRANSITIONS DON'T MAKE A VIDEO LOOK MORE PROFESSIONAL

It's tempting to assume using lots of different, fancy transitions makes an edit look more polished. Professional editing typically uses transitions sparingly and purposefully — per this chapter's own material, each one is meant to signal something specific to the viewer, not decorate the edit. Overusing flashy transitions indiscriminately is a common beginner tell that often looks less polished, not more. Restraint, not variety, is usually the actual mark of skilled editing.

Hands-On Exercises

EXERCISE 1

You need a simple text caption identifying a speaker's name and title at the bottom of the screen. Which tool from this chapter fits, and why would reaching for Fusion be unnecessary here?

View solution

EXERCISE 2

A beginner editor applies a different flashy transition (spins, flips, zooms) between every single clip in their video, believing this makes it look more professional. Using this chapter's own warning box, explain why this is likely to backfire.

View solution

EXERCISE 3

Explain why a cross-dissolve in this course's timeline-based editing and OBS Studio's own Fade transition between Scenes are described as the same underlying concept, despite being used in genuinely different contexts.

View solution

CHAPTER 5 QUICK REFERENCE

- **Cross-dissolve** — blends outgoing into incoming clip, the same underlying idea as OBS Studio's own Fade
- **Dip to Color/Black** and **Wipe** each signal a different kind of narrative break, not interchangeable stylistic choices
- Edit page's own **built-in title tool** handles simple captions/lower-thirds fast; **Fusion** handles genuinely custom motion graphics
- Choosing between basic titles and Fusion is a real decision matched to the actual need, like Chapter 2's own Cut-vs-Edit choice
- **Restraint, not variety**, is the real mark of professional editing — overusing flashy transitions is a common beginner tell

Chapter 2 named the Color page as one of Resolve's own seven dedicated environments. This chapter is the full treatment — and one of Resolve's own genuine signature strengths, not just one page among equals.

Color Correction vs. Color Grading: A Real Distinction

Color correction fixes technical problems — exposure, white balance, matching shots from different cameras or lighting to look consistent. A corrective, "fix a problem" step. **Color grading** is a creative, stylistic choice applied on top of already-corrected footage — establishing a mood or look, not fixing anything broken. Two genuinely different goals, often confused as the same thing.

The Color Wheels

Lift/Gamma/Gain (also called Shadows/Midtones/Highlights) are the primary color correction tools — adjusting color and brightness independently across three separate tonal ranges. The foundational tool used for both correction and grading alike.

Nodes: Resolve's Own Non-Destructive, Layered Color Approach

The Color page uses a node-based structure, similar in spirit to Fusion's own node system from Chapter 5. Each node applies one distinct correction or effect, chained in sequence — genuinely powerful for isolating and adjusting individual steps independently, for instance a corrective node followed by a separate creative-grade node, each fully editable on its own without affecting the other.

Scopes: Reading Color Objectively, Not Just by Eye

Waveform, **vectorscope**, and **histogram** are technical monitoring tools showing the actual, objective color and brightness data in a shot — since a monitor's own display settings, room lighting, or simple eye fatigue can all make an image look different than it technically is. Professional colorists rely on scopes rather than trusting their eyes alone for critical decisions like matching shots.

LUTs (Look-Up Tables)

A **LUT** applies a pre-defined color transformation — useful for quickly applying a consistent style, or converting a camera's own flat/log color profile into a more natural-looking starting point. A real, genuine shortcut, though professional grading still typically involves manual adjustment on top of a LUT, not relying on one alone as the finished result.

Why This Is One of Resolve's Own Signature Strengths

Resolve began life as a dedicated color-grading tool, well before it became a full editing application — the Color page's own real depth directly reflects that history. Premiere Pro's own Lumetri Color panel covers similar basic ground but historically lacked Resolve's own node-based depth and dedicated scopes, though Adobe has closed some of that gap over time — an honest comparison, not an exaggerated one.

CONCEPT	RESOLVE	PREMIERE PRO
Dedicated color environment	Color page	Lumetri Color panel
Structure	Node-based, chained corrections	Stacked effects, historically less node-like
Scopes	Dedicated, built-in	Available, historically less integrated
Origin	Began as a dedicated color tool	Began as a general NLE

THE SECOND TIME THIS COURSE HAS USED NODE-BASED THINKING

The Color page's own node structure is genuinely the same underlying mental model already introduced with Fusion in Chapter 5 — a chain of individually editable steps, rather than one flattened effect. Recognizing this pattern once means recognizing it again here.

CORRECTION AND GRADING ARE NOT THE SAME STEP, AND ORDER MATTERS

It's tempting to treat color correction and color grading as two names for the same activity. Conflating them can cause a real workflow mistake: applying a stylistic, mood-driven grade before actually correcting technical problems — mismatched white balance or exposure between shots — makes those technical problems genuinely harder to see and fix afterward, since the creative grade is now obscuring them. Correction should generally happen first, grading second, not the reverse.

Hands-On Exercises

EXERCISE 1

Two shots in the same scene were filmed with slightly different white balance settings and look noticeably mismatched when cut together. Is this a color correction or color grading problem, using this chapter's own definitions? Explain your answer.

View solution

EXERCISE 2

An editor applies a moody, desaturated cinematic grade to a scene first, then notices the shots still look slightly mismatched in exposure but finds it much harder to identify and fix the mismatch than before the grade was applied. Using this chapter's own warning box, explain what went wrong with their workflow order.

View solution

EXERCISE 3

An editor insists their shot looks properly exposed just from viewing it on their laptop screen, without checking any scopes. Using this chapter's own material, explain the risk in relying on this alone.

View solution

CHAPTER 6 QUICK REFERENCE

- **Color correction** fixes technical problems; **color grading** is a creative choice applied afterward — genuinely different goals
- **Lift/Gamma/Gain** — the primary color wheels adjusting shadows/midtones/highlights independently
- **Nodes** chain individually editable correction/grading steps — the same mental model as Fusion (Chapter 5)
- **Scopes** (waveform, vectorscope, histogram) show objective data — a monitor/eye alone can be misleading
- **LUTs** are a real shortcut, not a substitute for manual adjustment on top
- Correct **first**, grade **second** — grading before correcting makes technical problems harder to spot and fix

Chapter 2 named Fairlight as one of Resolve's own seven dedicated pages. This chapter is the full treatment — connecting directly back to OBS Studio's own audio material and forward to Audio Editing & Production Basics' own dedicated course.

What Fairlight Actually Is

Fairlight is a full, dedicated audio post-production environment built directly into Resolve — not a bolted-on afterthought. Genuinely comparable in depth to a standalone Digital Audio Workstation (DAW), sitting right alongside video editing without ever needing to leave the application.

Basic Audio Editing on the Timeline

Trimming and adjusting audio clips uses the exact same edit tools already learned in Chapter 4 — Razor, Ripple, Roll, and Slip all apply to audio clips too, not just video. A genuinely reusable skill, rather than a separate audio-specific toolset to learn entirely from scratch.

Track-Level Mixing

Per-track volume and pan work similarly in spirit to OBS Studio's own Audio Mixer (that course's own Chapter 3), just applied here to a fixed, already-recorded timeline rather than live Sources. **Fader automation** lets volume change smoothly over time rather than staying at a single fixed level — used, for instance, to duck background music under narration.

Basic Fairlight Effects

EQ and **Compression** are available directly on the Fairlight page — the exact same concepts covered conceptually in OBS Studio's own Chapter 3 (Noise Gate/Noise Suppression) and covered in far more depth in Audio Editing & Production Basics' own dedicated chapters. Here, though, they're applied in post-production on an already-recorded file, not live during capture.

Premiere Pro's Contrasting Approach: A Separate Application

Premiere Pro has only basic audio tools built directly into its own timeline. Genuinely deep audio work in the Premiere ecosystem requires switching to **Adobe Audition**, a separate application entirely. Resolve's own Fairlight page keeps this all inside one single application, with no context-switching required at all — a real, concrete advantage worth naming plainly here, not just diplomatic even-handedness for its own sake.

TASK	RESOLVE (FAIRLIGHT)	PREMIERE PRO
Basic audio trim/mix	Built directly into the timeline	Built directly into the timeline
Deep audio post-production	Fairlight page, same application	Requires switching to Adobe Audition
EQ/Compression	Directly in Fairlight	Available, more limited without Audition

WHERE DEDICATED AUDIO DEPTH STILL LIVES

Fairlight covers audio well enough for most video projects directly inside Resolve — but Audio Editing & Production Basics' own dedicated Audacity coverage goes further on production-specific techniques (frequency-focused cleanup, more nuanced compression) for anyone wanting to do more involved audio work, especially audio-only content like a podcast.

FAIRLIGHT EXISTING DOESN'T MEAN A DEDICATED AUDIO TOOL IS NEVER NEEDED

It's tempting to assume that since Fairlight exists, there's no reason to ever use a separate, dedicated audio tool. Fairlight is genuinely capable for audio tied to a video project, but audio-focused content — podcasts, voiceover-only work — or highly specialized audio tasks may still benefit from a dedicated tool's own deeper, audio-only-focused feature set (Audio Editing & Production Basics' own material). Fairlight's own real strength is convenience and depth *within* a video-editing workflow specifically, not necessarily surpassing every dedicated audio tool at every audio-specific task.

Hands-On Exercises

EXERCISE 1

You need to lower background music's volume specifically during a narration segment, then bring it back up afterward, rather than setting one fixed volume level for the entire track. Which Fairlight concept from this chapter fits, and why?

View solution

EXERCISE 2

A Premiere Pro editor wants to do genuinely deep audio cleanup on their project's dialogue track. Using this chapter's own material, explain what they'd need to do that a Resolve editor wouldn't.

View solution

EXERCISE 3

Someone starting a dedicated, audio-only podcast asks whether they should just use Fairlight for everything since it's already inside Resolve. Using this chapter's own warning box, explain a reasonable answer.

View solution

CHAPTER 7 QUICK REFERENCE

- **Fairlight** is a full, DAW-comparable audio post-production environment built directly into Resolve
- Chapter 4's own edit tools (**Razor/Ripple/Roll/Slip**) apply directly to audio clips too — a reusable skill
- **Fader automation** lets volume change smoothly over time, e.g. ducking music under narration
- Premiere Pro requires a **separate application (Audition)** for equivalent audio depth — a real, concrete Resolve advantage
- Fairlight's strength is convenience **within a video workflow** — dedicated audio-only work still benefits from a dedicated tool

Chapter 2 named Deliver as the final of Resolve's own seven pages, and Chapter 3 promised proxies never affect final quality without explaining the actual mechanism. This chapter is both — the real export step, and where that proxy promise is actually kept.

The Deliver Page

The **Deliver** page is where a finished timeline actually becomes a real, deliverable video file — render settings, output location, and format/codec selection all live here.

Render Settings vs. Project Settings

A genuinely common point of confusion: **project settings** control the resolution and framerate the timeline itself uses while editing, while **render settings** on the Deliver page control what actually gets exported — which can differ from the project's own settings if needed, for instance rendering at a lower resolution than the edited timeline. Understanding these as two separate, independently configurable things avoids confusion when an export doesn't match expectations.

Codecs & Containers

Another genuinely common point of confusion: a **container** (like MP4 or MOV) is the file format wrapping the data; a **codec** (like H.264 or H.265/HEVC) is how the actual video data inside is compressed. The same container can hold different codecs, and vice versa. **H.264** remains the most broadly compatible codec across platforms and devices; **H.265/HEVC** offers better compression at the same quality, but with less universal playback support.

Matching Delivery Settings to YouTube's Own Requirements

Cross-referencing Creating YouTube Content's own Editing Workflow chapter directly: YouTube publishes recommended upload specifications for resolution, bitrate, and codec. Resolve's own built-in YouTube export preset simplifies this by pre-configuring settings already matched to those recommendations, rather than needing to look them up and configure everything manually each time.

Where Proxies Get Swapped Back

Revisiting Chapter 3's own promise concretely: during the actual render process here on the Deliver page, Resolve automatically uses the original full-resolution footage rather than whatever proxy was used for smooth editing. This is the actual mechanism behind Chapter 3's own claim that proxies never affect final quality — it happens right here, at render time.

CONCEPT	CONTROLS
Project settings	Resolution/framerate the timeline uses while editing
Render settings	What actually gets exported — can differ from project settings
Container (MP4/MOV)	The file format wrapping the data
Codec (H.264/H.265)	How the video data inside is compressed

THE VIDEO-EDITING-SIDE VERSION OF A FAMILIAR IDEA

OBS Studio's own MKV-to-MP4 remuxing was producing a final, properly-formatted deliverable file at the end of a process, matched to whatever's actually needed next. This chapter's own export step does the identical job for video editing — producing a finished file matched to a YouTube upload, rather than whatever format was convenient during editing itself.

A CORRECT PREVIEW DURING EDITING DOESN'T GUARANTEE A CORRECT RENDER

It's tempting to assume that since the timeline already plays back correctly during editing, whatever gets rendered out will automatically look and sound right too. A mismatched or poorly chosen render setting — the wrong resolution, an incompatible codec/container combination, or a render happening before proxies fully relink to full-resolution footage — can produce a genuinely different, sometimes broken result than what was seen during editing. Always review the actual rendered output file before considering a project finished, rather than assuming the editing-time preview guarantees the final file is correct.

Hands-On Exercises

EXERCISE 1

A project was edited at 4K resolution, but the final export needs to be delivered at 1080p for a client's specific requirement. Explain, using this chapter's own material, why this is possible without re-editing the timeline itself.

View solution

EXERCISE 2

An editor exports a project, uploads it to YouTube, and immediately considers the job finished without ever opening the rendered file. Using this chapter's own warning box, explain the risk in skipping this step.

View solution

EXERCISE 3

A project was edited using proxies for smoother playback on a lower-power laptop. Explain what actually happens to those proxies during the final render, using this chapter's own material.

View solution

CHAPTER 8 QUICK REFERENCE

- **Project settings** (editing timeline) and **render settings** (Deliver page) are two separate, independently configurable things
- **Container** (MP4/MOV) wraps the data; **codec** (H.264/H.265) compresses it — not the same concept
- Resolve's own **YouTube export preset** pre-configures settings matched to that platform's own recommendations
- Proxies get **swapped back for full-resolution footage** right here, at render time — the actual mechanism behind Chapter 3's own promise
- Always **review the actual rendered file** — a correct editing-time preview doesn't guarantee a correct final render

Every prior chapter covered one piece of Resolve's own workflow. This capstone takes the exact footage OBS Studio's own capstone produced and edits it into a finished, publish-ready video — touching every chapter's own tool in the order a real edit would actually use them.

Step 1 — Importing the Footage (Chapter 3)

The MP4 file remuxed from OBS Studio's own MKV recording is imported directly into the Media Pool and placed into its own bin — the concrete handoff point between the two courses.

Step 2 — Generating Proxies for Smooth Editing (Chapter 3)

Since this is high-resolution tutorial screen capture, proxies are generated before editing begins, letting the timeline play back smoothly regardless of the editing machine's own power — with no effect on the eventual export quality.

Step 3 — Assembling a Rough Cut on the Cut Page (Chapter 2)

Since this is a straightforward, single-application tutorial with no complex multi-track needs, the Cut page's own fast, simplified tools are used first to assemble a rough sequence quickly.

Step 4 — Refining on the Edit Page (Chapters 2 & 4)

Moving to the Edit page for precise control, Ripple and Roll trims tighten pacing and remove dead air or mistakes the rough Cut-page assembly left in.

Step 5 — A Simple Intro Transition and Lower-Third Title (Chapter 5)

A cross-dissolve eases into the main tutorial content, and a basic lower-third title identifies the tutorial's own topic — deliberately using the Edit page's own built-in title tool rather than Fusion, since nothing here needs custom motion graphics.

Step 6 — A Light Color Correction Pass (Chapter 6)

Since this is screen and webcam footage rather than cinematic content, only a light correction pass is applied — fixing the webcam's own slightly uneven exposure and white balance. No creative grade is applied at all; per Chapter 6's own correction-vs-grading distinction, a tutorial genuinely doesn't need a stylistic mood, only accurate, consistent color.

Step 7 — Cleaning Up Audio on the Fairlight Page (Chapter 7)

A light EQ and compression pass refines the narration track further — a second pass on top of what OBS Studio's own live Noise Gate and Noise Suppression filters already handled during recording, not a replacement for that earlier cleanup.

Step 8 — Exporting With the YouTube Preset (Chapter 8)

Resolve's own built-in YouTube export preset on the Deliver page handles resolution, bitrate, and codec matched to YouTube's own recommendations. Per Chapter 8's own warning box, the actual rendered file is reviewed directly before the project is considered finished.

CAPSTONE STEP	CHAPTER IT DRAWS FROM
Step 1 — Importing footage	Chapter 3
Step 2 — Generating proxies	Chapter 3
Step 3 — Rough cut on Cut page	Chapter 2
Step 4 — Refining on Edit page	Chapters 2 & 4
Step 5 — Transition and title	Chapter 5
Step 6 — Color correction	Chapter 6
Step 7 — Audio cleanup	Chapter 7
Step 8 — Export and review	Chapter 8

IF DEEPER AUDIO WORK WERE NEEDED

If this tutorial's audio needed cleanup beyond what Fairlight's own light pass covers here, Audio Editing & Production Basics' own dedicated Audacity material — still outstanding in this same subject — is exactly where that deeper work would happen next.

A WELL-EDITED VIDEO DOESN'T GUARANTEE IT PERFORMS WELL

This capstone produces a genuinely well-edited, publish-ready video — clean cuts, accurate color, clear audio, correctly exported. But whether that video actually finds an audience depends on decisions this course never covers: thumbnails, titles, SEO, upload timing, and audience growth, all covered in this site's own YouTube subject (`yt1`-`yt3`). Editing well and growing a channel are genuinely separate skill sets, and this course's own scope stops at the finished, correctly exported file.

Hands-On Exercises

EXERCISE 1

Explain why this capstone's own Step 6 applies only a light color correction pass with no creative grade, referencing Chapter 6's own correction-vs-grading distinction directly.

View solution

EXERCISE 2

A viewer of this capstone asks why Step 7's audio cleanup is described as "a second pass" rather than the first time this footage's audio was ever cleaned up. Explain the answer, referencing OBS Studio's own material directly.

View solution

EXERCISE 3

Write a short chapter-attribution summary (2-3 sentences) explaining how this capstone's own step-by-step shape compares to OBS Studio's own capstone, and why both courses independently arrived at the same kind of shape.

View solution

COURSE COMPLETE

Video Editing Fundamentals — 9 of 9 chapters complete. Audio Editing & Production Basics remains outstanding as the final course in the Audio & Video Production subject.

CHAPTER 9 QUICK REFERENCE

- This capstone edits **OBS Studio's own recorded footage** from import through a finished export, touching every prior chapter's own tool in a natural order
- The edit order mirrors real practice: **import, generate proxies, rough cut, refine, add polish, correct color, clean audio, then export and review**
- Only a **light correction pass** was needed, no creative grade — an honest scope match to tutorial content, not cinematic footage
- Editing well and growing a channel are **separate skill sets** — this course's scope ends at a correctly exported file