

Edit Any Video *with Claude Code*

The 3-step setup behind the video — plus the one gotcha that sends everyone back to Final Cut after 90 seconds.

THREE STEPS

3

READ TIME

8 minutes

SETUP TIME

10–15 min

SKILL LEVEL

Beginner

Contents

01 What you'll have at the end

02 Before you start

03 **Read this first — don't paste the install command**

04 **Step 01 — Install Claude**

05 **Step 02 — Install the Remotion skill / MCP**

06 **Step 03 — Edit your first video**

07 Tips & tricks

08 Free prompts: the Showcase

09 Common gotchas

10 What to learn next

You've probably just watched a 35-second clip explaining that *Claude killed video editors*. The clip is short on purpose. This is the long version: the actual setup, the edge cases, and the one gotcha that sends most people back to Final Cut after 90 seconds. Read it once, follow it once, and you'll never timeline-edit by hand again.

OVERVIEW

What you'll have at the end

Four things land on your machine, plus a workflow loop that turns a one-sentence prompt into a rendered MP4.

- 01 Claude Code installed and signed in — desktop app or terminal, your call.
- 02 The Remotion skill / MCP wired up so Claude can write video timelines on demand.
- 03 A project folder where dropping a clip and one sentence produces a finished MP4.
- 04 An iteration loop that takes **minutes, not hours**, per revision.

TOTAL SETUP TIME

~10 minutes if Node is installed. ~15 if it isn't.

PREREQUISITES

Before you start

You need exactly four things. Anything else this guide tells you to install, you'll install along the way.

REQUIREMENT	WHAT FOR	HOW TO CHECK
Node.js 20+	The Remotion CLI runs on Node	<code>node -v</code> prints v20.x or higher
A terminal	Where you'll run Claude Code	Terminal.app, iTerm2, Windows Terminal — any TTY
A Claude account	API access for Claude Code	Sign up at claude.ai if you don't have one
A folder of footage	The video you want edited	Anywhere on disk; we'll point Claude at it

TIP

If `node -v` errors out, install via `nvm`. Don't use the system installer — you'll want to switch versions later.



READ THIS FIRST

In the desktop app, don't paste the npx.

Describe what you want. Let Claude run the install.

THE ONE RULE

In the **Claude desktop app**: never paste the `npx` command from the Remotion site. *Describe the install in plain English instead* and let Claude handle it. (In Claude Code in a terminal, pasting is fine — the terminal can drive the interactive wizard.)

Visit remotion.dev and you'll see the install command shown right on the page — something like `npx create-video@latest`. It's tempting to copy that straight into Claude's chat in the desktop app. Don't.

That command is **interactive**. It expects you to answer questions with arrow keys:

```
remotion · install wizard

? What would you like to name your video? (my-video)
? Choose a template: (Use arrow keys)
  > Hello World
    TypeScript
    React Router
```

Pasted into the desktop app's chat, this stalls silently. The sandboxed shell underneath is waiting for keystrokes that will never arrive — and the app's chat has no way to send them.

The fix: ask, don't paste

Instead of the raw command, type a sentence:

```
> Install Remotion and scaffold a portrait video project here. Use the TypeScript template.
```

Claude runs the install on your behalf, drives the wizard, picks the template, and waits out the `npm install`. You walk away for two minutes and come back to a working folder. This is the path the video shows.

POWER-USER NOTE

If you're running **Claude Code in your terminal** instead of the desktop app, this rule doesn't apply — your terminal can drive the interactive wizard, so pasting the `npx` command works. Either way, the safest approach is to describe the install and let Claude pick the right path.

01

STEP ONE

Install Claude.

One download. Same on Mac and Windows.

1. Go to claude.ai/download in any browser.
2. Download the **Claude desktop app** for your OS — Mac or Windows. Run the installer.
3. Open the app and sign in with your Claude account. (If you don't have one yet, sign up at claude.ai first — it takes 30 seconds.)

You're done when the app opens to a chat window. That's where the next two steps happen.

FOR POWER USERS

If you'd rather use **Claude Code in your terminal**, install via `curl -fsSL https://claude.ai/install.sh | bash` on macOS / Linux, or `npm install -g @anthropic-ai/claude-code` anywhere with Node. Then run `claude` to sign in. The rest of this guide works in either interface.

02

STEP TWO

Install the Remotion skill / MCP.

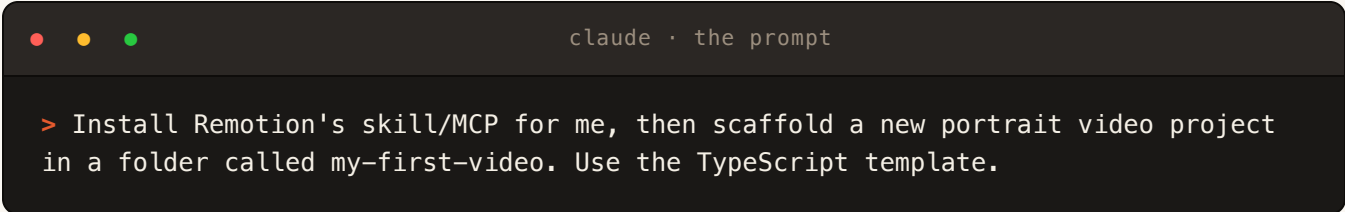
Don't go to remotion.dev. Just ask Claude.

A — Open Claude (the app, or terminal) and start a new chat

Pick a fresh chat so Claude has a clean working context.

B — Tell Claude what you want

One sentence does it:



```
claude · the prompt  
  
> Install Remotion's skill/MCP for me, then scaffold a new portrait video project  
in a folder called my-first-video. Use the TypeScript template.
```

Claude will create the folder, install the Remotion skill, run the install wizard, pick the TypeScript template, and finish with `npm install`. Walk away for two minutes.

C — Verify it worked

Ask Claude to show you what landed:



```
claude · check  
  
> Show me what's in the my-first-video folder.
```

```
package.json  remotion.config.ts  src/  public/  node_modules/
```

If you see those five things, you're in business. If you don't, ask Claude what went wrong — the answer is almost always *"an npm step failed; let me retry."*

THE GOTCHA · DON'T PASTE

If you visit remotion.dev, you'll see the npx install command sitting right on the page. **In the desktop app, don't copy-paste it into chat.** `npx create-video@latest` is interactive, and the app's chat can't drive arrow keys.

Describe the install instead — Claude will run it correctly. (Claude Code in a terminal can paste it fine, but describing still works in either interface.)

03

STEP THREE

Edit your first video.

The part that actually feels like magic.

A — Drop your footage into `public/`

Open the project folder in Finder (or Explorer on Windows) and **drag your video file into the `public/` folder**. That's it. If you'd rather use the terminal:

```
$ cp ~/Downloads/raw-clip.mp4 my-first-video/public/raw-clip.mp4
```

`public/` is special: anything in there can be referenced by Remotion at render time.

B — In your Claude Code session, describe the edit

Be specific:

```
claude code · the prompt  
  
> I have a video at public/raw-clip.mp4. Cut out every filler word ("um", "uh"), trim any pause longer than 0.6 seconds, and add word-by-word captions. Render to final.mp4 at 1080x1920 portrait.
```

Claude will:

1. Probe the video for duration and resolution.
2. Generate a transcript (if you have a transcription tool wired; otherwise it'll ask).
3. Write the Remotion timeline as TypeScript in `src/Composition.tsx`.
4. Run `npx remotion render` and watch the frames tick.
5. Hand you back `final.mp4`.

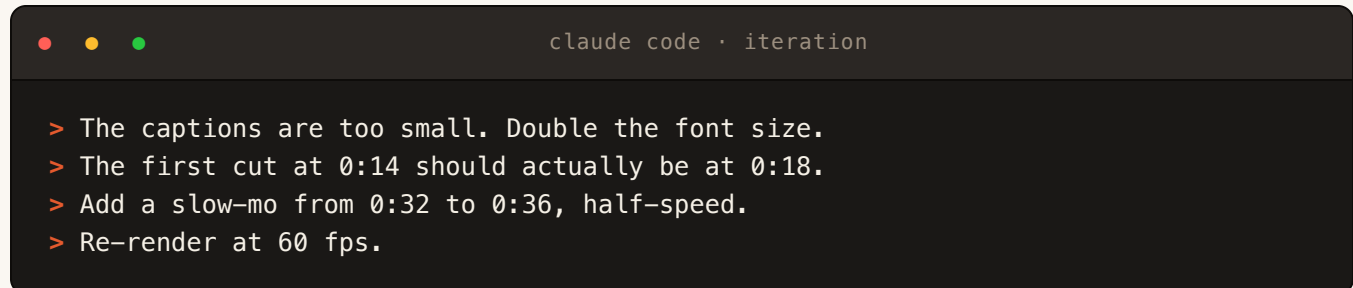
You'll get the file in 1–3 minutes depending on length. The first render is always the slowest because it has to bundle. Subsequent renders cache.

TIP

Be specific about the output. "Cut filler words" is vague; "remove every 'um' and 'uh' but keep all 'like' uses" is actionable. The more concrete the description, the less iteration.

C — Iterate

This is where the workflow flexes. From the same session:



```
claude code · iteration  
  
> The captions are too small. Double the font size.  
> The first cut at 0:14 should actually be at 0:18.  
> Add a slow-mo from 0:32 to 0:36, half-speed.  
> Re-render at 60 fps.
```

Each prompt is seconds, not minutes. Claude edits the timeline file and re-runs the render.

*This is the loop that **kills traditional editing** — not because the AI is smarter than a human editor, but because the iteration cost drops to near zero.*

04

FIELD NOTES

Tips & tricks.

A handful of things you'll wish you knew on day one.

THE TRICK

WHY IT WORKS

Keep raw footage *outside* `public/` until you actually use it. Symlink in.

Public/ files get bundled into every preview, slowing startup.

Render previews at 30 fps before going to 60.

4× faster iteration; bump only the final pass.

Ask Claude to leave inline TSX comments explaining each scene's intent.

Next time you edit, Claude reads its own notes and skips re-deriving context.

`git init` on day one. Each Claude edit becomes a commit.

Lets you say "go back to two prompts ago" with surgical precision.

For clips longer than 10 minutes, ask Claude to chunk first.

Monolithic renders OOM; chunked renders are faster and concatenate cleanly.

Bake captions early, not last. Captions inform pacing decisions.

Adding captions late means re-cutting to caption breaks.

Use the `--concurrency=4` render flag.

Modern Macs render 3–4× faster with parallel encoding; default is conservative.

Drop a `CLAUDE.md` in the project root. Claude reads it on every session.

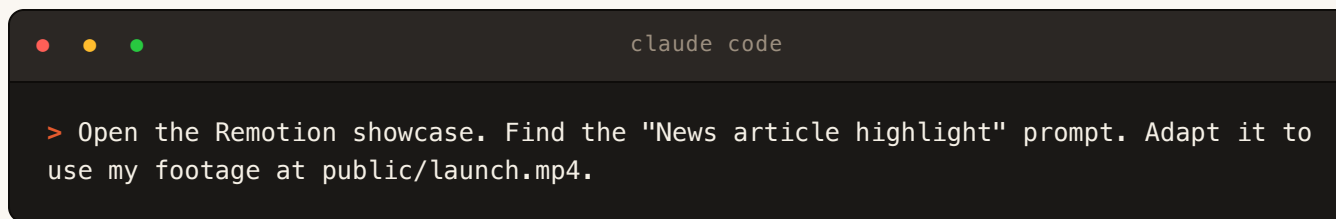
Encode preferences once ("always 9:16, always 1080p") instead of repeating.

Free prompts: the Showcase

Remotion publishes a [Prompt Showcase](#) — community-built timelines you can clone with one click.

Each card has a small Claude icon. Click it and the prompt is handed to Claude Code directly.

From inside Claude Code you can also ask:

A screenshot of the Claude Code terminal window. The window title is "claude code". The prompt text is: "> Open the Remotion showcase. Find the 'News article highlight' prompt. Adapt it to use my footage at public/launch.mp4."

```
claude code  
  
> Open the Remotion showcase. Find the "News article highlight" prompt. Adapt it to  
use my footage at public/launch.mp4.
```

Personal favorites — each is a starting point Claude will adapt:

- **News article headline highlight** — for any reaction-to-a-news-story clip.
- **Travel Route on Map with 3D landmarks** — for trip recaps; auto-generates the line + city pins.
- **Rocket Launches Timeline** — looks ridiculous on portrait video; data-viz that actually pops.

05

TROUBLESHOOTING

Common gotchas.

If you hit one of these, the fix is here. If you hit something not on this list — ask Claude.

"MODULE NOT FOUND: REMOTION"

You launched `claude` from the wrong directory. `cd` into the project folder first, then run `claude`.

"CANNOT FIND MODULE 'ESBUILD'"

`npm install` didn't finish. Run `npm install` in the project root manually, then resume.

CLAUDE SAYS "I CAN'T RUN THAT COMMAND"

The command is interactive. Don't paste it — describe what you want and let Claude handle the install. If the app keeps refusing, switch to Claude Code in your terminal. (See *Read this first.*)

RENDERS COME OUT SILENT

Two usual suspects: (1) the audio file isn't under `public/` — `staticFile()` only sees `public/`-relative paths, no remote URLs; (2) you wrapped the audio in a `<Sequence from={...}>` that starts after the clip ends.

THE TIMELINE DRIFTS AFTER A RE-CUT

You spliced audio without shifting downstream scene timings. Every audio splice changes the timeline by Δ ; every scene's start time past the splice needs $+\Delta$. Claude will do this automatically if you ask: *"I just edited the VO at 0:27. Update all subsequent scene times."*

THE RENDER FREEZES ON THE LAST FRAME

Source video is shorter than the scene. Either the playback rate is too fast (`scene_dur × playback_rate > source_dur`) or `startFrom` is too late. Lower one of the three.

What to learn next

Once the loop above feels comfortable, three directions to expand.

1. **Custom scene templates.** Write a typing-animation or stat-card template once, reuse it across videos. The `contentgen-finisher` skill ships several reference templates.
2. **Voice-over + synced captions.** Pre-record narration with ElevenLabs, get word-level timestamps, sync captions, layer music underneath. The `contentgen-planner` skill walks through this end-to-end.
3. **Multi-clip masonry layouts.** Collage four prior renders in a 2x2 grid for "look what I shipped this week" beats. The `MasonryGrid` component is the reference.

edit.

*That's the whole loop. Drop a clip, write a sentence,
render. Iterate in seconds.*

Now go cut something.

· MADE WITH CLAUDE CODE · NO EDITOR WAS HARMED ·