The One-Prompt Video Studio Playbook

The prompts, approval gates and budget caps I use to run Open Montage from Claude Code, from first prompt to finished video.

What Open Montage is and why I use it

Open Montage is a free, open source video production system that runs inside your AI coding assistant. I use it with Claude Code. You describe a video in plain language, and a team of AI agents does the research, writes the script, generates visuals, pulls real B-roll, adds music, voiceover and sound effects, then delivers a finished video.

The usual way to make a video is a chain of handoffs. One person researches, one writes, one records the voice, one edits, and each handoff costs time or money. Open Montage puts those jobs in one pipeline that you steer from a chat window.

This guide is the playbook I wish I had on day one: how the pipeline works, the one thing that surprises people, the prompts to copy, the setup, and a fix table. The repo is released under the AGPLv3 license, and every API key is optional.

01

How it actually works

1. You open the Open Montage folder in Claude Code and type one prompt. Claude reads the project files and picks a pipeline, such as an animated explainer, a cinematic trailer or a screen demo.
2. Every pipeline runs the same stages in order: research, proposal, script, scene_plan, assets, edit, compose. Each stage has a director skill, a markdown file in the skills folder that teaches the agent how to do that one job.
3. Before anything gets rendered, the agent checks its own work with ffprobe, frame sampling, audio level analysis and subtitle checks. Then it hands you the video, and you decide what to change.

02

The one thing almost everyone gets wrong

People expect a one-prompt tool to run start to finish without them. It does not, and that is on purpose. The proposal, the script, the scene plan, the generated assets and the publish step each pause and wait for your sign-off.

> **WATCH OUT**
>
> When the agent stops, it is waiting for you, not stuck. The gate holds until you reply in chat. Read what it is asking, answer it, and the run continues. Asset generation pauses on a storyboard with takes, prompts and the cost of each asset, so approve the visuals before the render, not after.

---

03 The prompts and the files

SOURCE CODE

The full repo:

https://github.com/calesthio/OpenMontage

These prompts come straight from the Open Montage README. Paste any of them into your assistant after setup. The first two cost nothing, because they use free tools only.

ZERO KEYS: ANIMATED EXPLAINER

```
Make a 45-second animated explainer about why the sky is blue
```

FREE REAL-FOOTAGE DOCUMENTARY

```
Make a 90-second documentary montage about what a city feels like at 4am. Use real
footage only, no narration, elegiac tone.
```

START FROM A REFERENCE VIDEO

```
Here's a YouTube short I love. Make me something like this, but about CRISPR for high
school students.
```

PRODUCT TEASER (NEEDS AN IMAGE OR VIDEO PROVIDER)

```
Make a product launch teaser for a fictional smart water bottle called AquaPulse
```

CINEMATIC TRAILER (FULL SETUP)

```
Create a cinematic 30-second trailer for a sci-fi concept: humanity receives a warning
from 1000 years in the future
```

Open Montage also reads instruction files in the repo root, so your assistant knows the rules before it starts. Claude Code reads CLAUDE.md, and AGENTS.md is the shared operating guide for the other assistants.

.ENV (ALL OPTIONAL)

```
# Free to get
PEXELS_API_KEY=
PIXABAY_API_KEY=
UNSPLASH_ACCESS_KEY=

# Paid or gateway keys, add only what you need
FAL_KEY=
ELEVENLABS_API_KEY=
OPENAI_API_KEY=
GOOGLE_API_KEY=
```

SITES SHOWN IN THE VIDEO

- Open Montage on GitHub
  https://github.com/calesthio/OpenMontage
- Open Montage website
  https://openmontage.video
- Open Montage on YouTube
  https://www.youtube.com/@OpenMontage

Set it up in Claude Code or Codex

1. Check the requirements first. You need Python 3.10 or newer, FFmpeg, Node.js 18 or newer, and an AI coding assistant. On a Mac, install FFmpeg with: brew install ffmpeg. The HyperFrames renderer needs Node.js 22 or newer.
2. Clone the repo and go into it: git clone https://github.com/calesthio/OpenMontage.git, then cd OpenMontage.
3. Run the setup: make setup. If you do not have make, ask Claude Code to run the manual install steps from the README for your system. It creates a Python virtual environment, installs requirements.txt, runs npm install in the remotion-composer folder, installs piper-tts and copies .env.example to .env.
4. Open the OpenMontage folder itself in Claude Code, or in Codex. The skills it needs already sit inside the repo in the skills folder, with the stage directors in skills/pipelines, so there is nothing to copy into your own skills folder.
5. Add an API key only when you want a tool that needs one. Ask Claude to add it to the .env file in the project. Every key is optional.
6. Run the first test. Type: make demo, to render the zero-key demo videos. Or paste the sky is blue prompt from section 4.
7. A good first run looks like this. The agent opens the Backlot board for you, a local page that fills in as the pipeline runs. It then stops at the proposal and waits for your answer in chat. Reply, approve each gate, and the final video renders.

💡 QUICK TIPS

- Start with the zero-key prompts. Once you have seen one full run, adding a paid provider makes more sense.
- If you want to watch a production live without making one, the README includes: python scripts/backlot_simulate_run.py

---

04

The commands I actually run

CHEAPEST FIRST DRAFT

```
Make a [45]-second animated explainer about [TOPIC] for [AUDIENCE]. Use free tools
only, and tell me the estimated cost before you generate any assets.
```

> **REMAKE A VIDEO I LIKE**
>
> ```
> Here's a [YouTube Short / Reel / TikTok] I love: [LINK]. Make me something like this, but about [YOUR TOPIC]. Keep the pacing and the hook style, change the topic and the visuals, and show me the cost before you start.
> ```

> **PRODUCT TEASER**
>
> ```
> Make a [30]-second product launch teaser for [PRODUCT NAME], [ONE-LINE DESCRIPTION]. Show me the storyboard for approval before you render anything.
> ```

> **TRAILER FOR A CONCEPT**
>
> ```
> Create a cinematic [30]-second trailer for [CONCEPT]. Pause at the script and again at the scene plan so I can approve each one.
> ```

> **CHECK WHAT MY SETUP CAN DO**
>
> ```
> List which tools and providers are available in this setup, which ones need a key I have not added, and which pipeline fits [VIDEO IDEA].
> ```

When it breaks

ERROR	FIX
npm install fails with ERR_INVALID_ARG_TYPE on Windows	Run: npx --yes npm install
make: command not found	Use the manual install steps from the README instead of make setup.
HyperFrames will not run	It needs Node.js 22 or newer. The general minimum for the project is 18.
The agent stops and nothing happens	It is waiting at an approval gate. Reply in chat with your answer and it continues.
The video looks plain, with no AI visuals	With zero keys you only get the free tools. Add an image or video provider key to the .env file for generated visuals.
The cost is higher than I expected	Budget modes are observe, warn and cap. The default total cap is 10 dollars, and the default per-action approval threshold is 50 cents. Ask for the estimate before assets are generated.
I am not sure which provider to use	Read docs/PROVIDERS.md in the repo for setup, pricing and free tiers.

Ready to go deeper?

This playbook is the short version. Inside the hub you get the full build walkthroughs, the prompt library, and the systems I use to ship this in a weekend.

Join the Build →

https://hub.digicuratoragency.com/join