DEEP TUTORIAL

MOTION GRAPHICS WITH CLAUDE CODE AND HYPERFRAMES

We built a 15 second launch teaser from an empty folder in about 5 minutes: kinetic type, a phone UI, a growing chart, sound effects and a real MP4. Here is every command we ran, the one prompt, what it costs, and every problem we hit with its fix.

BY BOTS ON PAYROLL Tested November 9, 2026 About a 9 minute read

Free skill: https://media.botsonpayroll.tech/tutorials/motion-graphics-skill.zip
Web version: botsonpayroll.tech/guides/motion-graphics

This is the finished piece. Orbit is a made-up habit app, so the brief stays the star. Every frame is one HTML file that Claude Code wrote, and HyperFrames rendered it on our laptop in 13 seconds.

No After Effects, no keyframes by hand, no stock templates. Follow along even if you have never opened a terminal.

What you need, and what it costs

Prices checked on the official pages on November 9, 2026. Plans change, so confirm before you buy. No affiliate links here.

A Mac, Windows or Linux computer — Free

Ours was a Mac. Rendering runs on your own machine.

Claude, with Claude Code — $20/mo

Claude Code is included in Claude Pro: $20 a month, or $17 a month billed yearly. Max starts at $100 a month for heavier use. Source: claude.com/pricing.

HyperFrames — Free

Open source under Apache 2.0, with no per-render fees, and it renders locally. Source: github.com/heygen-com/hyperframes.

Node.js 22 or newer, and FFmpeg — Free

HyperFrames lists both as requirements. Same source.

A voiceover (optional) **$0 to $6/mo**

Our teaser has no voice. If you want one, HyperFrames has a built-in tts command. ElevenLabs' free plan has no commercial license; Starter adds one at $6 a month ($1 the first month). Source: elevenlabs.io/pricing.

The total: if you already pay for Claude Pro, every step below costs nothing extra.

Install, once

Open Terminal (on a Mac: Command and Space, type Terminal, press return) and paste these one at a time.

1. Node and FFmpeg

On a Mac, through Homebrew (install it first with the line on brew.sh if you don't have it):

```
brew install node ffmpeg
node -v          # v22 or higher
ffmpeg -version
```

On Windows, install Node.js 22+ and FFmpeg from their official sites.

2. Claude Code

```
# Mac, Linux, WSL
curl -fsSL https://claude.ai/install.sh | bash
# Windows PowerShell
irm https://claude.ai/install.ps1 | iex
```

Source: Claude Code quickstart. Sign in with your Claude Pro or Max account the first time you run `claude`.

How it works, in one paragraph

HyperFrames treats a video as a web page. You describe scenes in HTML and CSS, and you put every movement on one GSAP timeline (GSAP is the animation library the page loads). HyperFrames then opens the page in a headless browser, steps the timeline one frame at a time, takes a picture of each frame and hands them to FFmpeg with your audio. Because nothing moves on its own clock, every render comes out identical. Claude Code's job is to write that page from your brief, run HyperFrames' checks, and fix what they flag.

The 7 steps we ran

All timings are from our run on November 9. We pinned HyperFrames to version 0.8.145, the one we tested. Newer 0.9 versions exist on npm; if you use one, run check again and expect new rules.

12 s 1. Scaffold an empty project

```
npx --yes hyperframes@0.8.145 init launch-teaser --resolution portrait --non-interactive
cd launch-teaser
```

You get a blank 1080 by 1920 composition, a package.json, and a CLAUDE.md that tells Claude Code to use the HyperFrames skills. Init also adds those agent skills by default. Restart Claude Code in the folder so it loads them.

UNDER 1 MIN 2. Fonts and sound effects

Put your font files in a `fonts` folder. We used Instrument Sans and Geist Mono, both free under the Open Font License. Then make five sound effects with FFmpeg alone, no sound library needed:

```
mkdir -p audio && cd audio
ffmpeg -f lavfi -i "sine=f=1800:d=0.05" -af "afade=t=out:st=0.01:d=0.04" tick.wav
ffmpeg -f lavfi -i "sine=f=520:d=0.18" -af "afade=t=out:st=0.02:d=0.16" pop.wav
ffmpeg -f lavfi -i "anoisesrc=d=0.6:c=pink:a=0.5" -af
"highpass=f=400,lowpass=f=5000,afade=t=in:d=0.25,afade=t=out:st=0.3:d=0.3" whoosh.wav
ffmpeg -f lavfi -i "sine=f=1320:d=0.9" -f lavfi -i "sine=f=1980:d=0.9" -filter_complex
"amix=inputs=2,afade=t=out:st=0.05:d=0.85" ding.wav
ffmpeg -f lavfi -i "sine=f=70:d=0.5" -af "afade=t=out:st=0.03:d=0.45" boom.wav
cd ..
```

ONE MESSAGE 3. Give Claude Code the brief

Start `claude` in the folder and paste the prompt below. Claude Code wrote one index.html of about 190 lines: the styles, five scene blocks, fifteen sound cues and one timeline.

9 S 4. Run check

```
npx --yes hyperframes@0.8.145 check
```

It lints the timeline, renders sample frames, and looks for text off the canvas, overlapping text, contrast and motion problems. Ours passed with 7 warnings on the first round and 1 on the second. Every one is in the fixes below.

8 S 5. Snapshot and look

```
npx --yes hyperframes@0.8.145 snapshot --at 0,1.5,4.6,6.2,8.5,11.9,14.2 --describe false
```

This writes one PNG per time plus a contact sheet. Pick one time inside each scene and look at every frame yourself. Our worst problem of the night was only visible here.



13 s 6. Render

```
npx --yes hyperframes@0.8.145 render --output renders/orbit-teaser.mp4 --quiet
```

15 seconds of 1080 by 1920 video at 30 frames a second, with sound, 5.7 MB. It rendered locally in 13 seconds.

2 s 7. Fix the loudness

```
ffmpeg -i renders/orbit-teaser.mp4 -c:v copy -af loudnorm=I=-16:TP=-1.5:LRA=11 -c:a aac -b:a
192k renders/orbit-teaser-final.mp4
```

Sound effects on their own measured -32 LUFS, far quieter than anything else in a feed. This pass brought ours to about -15. The video is copied as is.

The prompt we used

Change the words to your own product. Keep the structure: size, colors, fonts, one line per scene, and what to do after.

> **THE BRIEF**
>
> Using /hyperframes, make a 15 second vertical launch teaser (1080x1920) for a habit tracker app called Orbit. Ink background #16171B, orange #FF5A1F and yellow #FFC83D accents, Instrument Sans for words, Geist Mono for labels (the font files are in fonts/). Five scenes: 1) MEET ORBIT, already on screen at frame 0, with an orange ring and a yellow moon orbiting the word; 2) a phone card rises, four habits check off one by one while a streak counts DAY 1 to DAY 30; 3) THIS WEEK: seven bars grow and a percentage counts up to 92% of habits kept; 4) "Never break the chain." with three feature pills flying in; 5) end card: the ring logo spins in, ORBIT, and OUT FRIDAY wiped in with a yellow highlighter. Add the sound effects in audio/ on each beat. Then run check, fix what it flags, and show me snapshots of every scene.

> **AFTER YOU LOOK AT THE SNAPSHOTS**
>
> On frame 0 the tagline touches the ring. Make the ring bigger and the tagline narrower so nothing touches, run check again, then render it and bring the audio up to -16 LUFS.

HyperFrames' own prompting page says the same things we learned: one clear sentence per beat, give every second a job, and state sizes and times as exact numbers.

The problems we hit, and the fixes

An element started in the wrong place when you skip around.
`gsap_repeated_fromto_without_baseline`

Two `fromTo()` animations on one element (our phone card rose in, then slowly zoomed). GSAP applies the second one's start values as soon as the page loads, so a frame before it can show the wrong state. Fix: add `immediateRender: false` to every later `fromTo` on that element.

The same warning on the number counters.
`gsap_repeated_fromto_without_baseline (dwell/hold)`

The DAY 1 to 30 counter and the 92% counter were both `fromTo()` on plain objects. Fix: animate them with `tl.to(counter, {n: 30, onUpdate: ...})` instead.

Pills flying in from off screen were flagged.
`canvas_overflow · escaped_container`

They start off the canvas on purpose. Fix: add `data-layout-allow-overflow` to each one (and to the ring, whose moon pokes past the edge as it turns). Use it only for things that move in or out on purpose.

Two lines of text on top of each other during a scene change.
`content_overlap`

"The chain." was still sliding out while OUT FRIDAY landed. Fix: end the old scene before the new text arrives. We moved scene 4's exit from 12.85 s to 12.6 s and ended it at 13.0 s.

A contrast warning that was wrong.
`contrast 1:1 at 10.833 s`

It sampled a pill while it was half off screen. The snapshot at that exact time showed dark text on a light pill. Look at the frame before you change colors. We left it.

Five "needs subcomposition" warnings.
`nested_structure_needs_subcomposition`

Advice for the HyperFrames Studio timeline view, not a render problem. Safe to leave on a short piece.

Text touching a shape, which check did not catch.

On frame 0 the tagline ran into the orange ring. We only saw it in the snapshot. Fix: ring from 900 to 1000 px, tagline from 52 to 46 px and at most 620 px wide.



Before



After

Silent sound effects.
`media_missing_id`

We hit this one building the YouTube version of this tutorial. An `<audio>` tag with a start time but no `id` is skipped by the renderer, so it plays nothing. check flags it as an error. Fix: give every audio tag its own id, like `<audio id="pop1" src="audio/pop.wav" data-start="5.9">`.

Too quiet.

Sound effects with no music or voice came out at -32 LUFS. Fix: the loudnorm line in step 7.

Run it yourself with our Claude skill

We packed the whole workflow as a Claude skill: a SKILL.md with the steps and rules above, three scripts (start a project with fonts and sound effects, make the sound effects, render plus loudness) and the Orbit teaser as a working example to start from. We tested it in a fresh folder: check passed and it rendered in 13.5 seconds.

1. Download motion-graphics-skill.zip and unzip it.
2. Put the `motion-graphics` folder in `~/.claude/skills/` (or `.claude/skills/` inside a project).

3. Start Claude Code and ask: "Use the motion-graphics skill to make a 10 second teaser for my product." Answer its questions about the beats, colors and last frame.

Skills run with your permissions. Open SKILL.md and the scripts and read them before you use them, like any code from the internet.

AI news + the AI tools worth hiring. @botsonpayroll
Sources: claude.com/pricing · code.claude.com/docs/en/quickstart · github.com/heygen-com/hyperframes · hyperframes.heygen.com · elevenlabs.io/pricing · brew.sh. Everything else is from our own run on November 9, 2026.