ELEVEN VIEWS

FREE RESOURCE

Script to Scenes: The One-Page Map That Turns a 1,200-Word Script Into 35 Remotion Scenes

Paste your script on the left, get a scene-by-scene build order on the right, before Claude writes a frame.

Here is the thing nobody tells you about making AI video: the model is not your bottleneck. Claude can write 40 Remotion scenes in a session without breaking a sweat. The reason your videos come out as 8 sad scenes that all look the same is that you handed Claude a 1,200-word script and said "make this a video." So it guessed. It guessed where the beats were, guessed what should be on screen, guessed how long each shot runs, and then built one giant 700-line component that you cannot edit without it collapsing. You do not have a video problem. You have a chunking problem. This guide is the planning layer that sits between writing the script and building the video. It is one page. You fill it out by hand in about fifteen minutes. Then Claude builds clean, in order, and lands the narration on the same frame as the motion. Below is the rule for breaking a script into scenes, a fillable map template, the exact handoff prompt, the word-count math, and two scripts mapped start to finish so you can see what "done" looks like.

atlas.elevenviews.io

01

Why 35 clean scenes beats 8 do-overs

Picture two ways to spend the same afternoon.

Way one: you paste the whole script into Claude, ask for a video, and get back one massive Composition with eight scenes crammed into a single file. Scene three runs nine seconds with one word on screen. Scene six has four ideas fighting in the same shot. You want to fix the timing on scene five, but every change ripples through the file and breaks two other scenes. You do five do-overs and quit at "good enough."

Way two: you spend fifteen minutes filling out a one-page map first. Each scene gets a row: a timecode, a type, the on-screen text, a motion note, and the asset it needs. Claude reads the map and returns 35 separate scene components in order. Scene five is its own file. You tweak it, re-render that one scene, done. Nothing else moves.

The map is the difference. It is boring. It is the part everyone skips because writing the script felt like the creative work and the map feels like admin. But the map is where the video actually gets designed. The script is what gets said. The map is what gets seen and when. Two completely different jobs, and most people only do the first one.

A good rule of thumb from real builds: a 3 to 5 minute video is 28 to 40 scenes. If your map has 9 rows for a four-minute video, your scenes are too long and the video will feel like a slideshow. If it has 80, you are cutting on every comma and Claude will choke on the render. The sweet spot is one visual idea per scene, 4 to 9 seconds each.

02

The chunking rule: one idea, one breath, one scene

Here is how you cut a script into scenes without overthinking it.

The core rule: a new scene starts when the idea changes OR when the speaker would naturally take a breath. Read your script out loud. Every place you'd pause, draw a slash. Every place the topic shifts, draw a slash. Those slashes are your scene breaks.

Words per scene: aim for 12 to 22 words of narration per scene. At a normal speaking pace of about 150 words per minute, that lands each scene between 5 and 9 seconds. Anything under 8 words is usually a punch line or a stat and should be its own short scene (2 to 4 seconds) for impact. Anything over 25 words is two ideas wearing a trench coat. Split it.

Now decide what becomes on-screen text versus voiceover only. Not every word should appear on screen. Reading along with the narrator word-for-word is exhausting and amateur. Use this split:

On-screen text gets: the key phrase (3 to 7 words, not the full sentence), numbers and stats, names and labels, the one word you want to land.

Voiceover only gets: connective tissue ("and that's why", "so here's the thing"), setup that pays off visually, anything the motion already shows.

Rule of thumb: if the narrator says "we cut render time by sixty percent," the screen shows "60%" big and bold, not the whole sentence. The voice carries the sentence. The screen carries the hit.

Last piece: tag each scene with a type. You only need about six types for most videos, and naming the type up front tells Claude which motion pattern to reach for. Title card, stat/number, bullet reveal, quote, diagram/flow, image/screenshot, transition. Six types covers 90% of explainer and case-study video.

03

The fillable scene map (your one page)

This is the template. One row per scene. Print it, or keep the plain-text version open and type into it. Five columns and that is on purpose. More columns and you'll never finish it.

The columns:

Timecode: running start time of the scene (00:00, 00:06, 00:11...). You fill this in last, after you know each scene's length from the word count. It is what makes narration and motion line up.

Scene type: one of your six types. Title, Stat, Bullets, Quote, Flow, Image, Transition.

On-screen text: the exact words that appear. Keep it to 3 to 7 words. If you write a full sentence here, you did it wrong.

Motion note: one short instruction. "Number counts up from 0 to 60." "Three bullets stagger in 0.3s apart." "Logo scales in, settles, holds." This is the line Claude turns into animation.

Asset needed: any image, logo, screenshot, or chart this scene requires. Write "none" if it is pure type and motion. This column is your shot list. When you finish the map you know every asset to gather before building.

The discipline that makes it work: fill the On-screen text and Motion note columns as if you are directing, not writing. Short, visual, specific. "Headline fades up over dark background, gold accent on the last word" is a motion note. "Make it look good" is not.

Fill the rows top to bottom from your slashed-up script. One slash, one row. When you're done, count the rows. Sum the seconds. That total should match your target runtime within a few seconds. If it is way off, your scenes are too long or too short and you adjust now, on paper, where it costs nothing.

04

Word-count-to-seconds math (so it lands on the frame)

This is the part that makes your narration and your motion hit the same frame instead of drifting apart by the end of the video. It is simple arithmetic and it saves you an hour of nudging keyframes in the editor.

The base rate: normal voiceover pace is about 150 words per minute, which is 2.5 words per second. So:

seconds = words ÷ 2.5

A 15-word scene = 6 seconds. A 20-word scene = 8 seconds. A 5-word punch line = 2 seconds. Easy.

Now convert to frames, because Remotion thinks in frames, not seconds. At 30fps:

frames = seconds × 30

So a 6-second scene is 180 frames. An 8-second scene is 240 frames. Put the frame count in your handoff so Claude sets each scene's durationInFrames exactly. That is the whole trick: when the scene's duration equals the narration length, the motion finishes when the sentence finishes.

Three real-world adjustments:

Add a 0.5 second pad to scenes that end on a beat or a stat. The number should hold for a moment after the voice stops. So a 2-second stat scene becomes 2.5 seconds (75 frames).

If you talk fast, recalc at 165 wpm (2.75 words/sec). If you use a slower, weightier delivery, use 135 wpm (2.25 words/sec). Time yourself reading one paragraph out loud once and you'll know your real rate.

When you do not have voiceover yet and you're building to a rough timing, the math still gives you scene lengths that feel right, so the video has rhythm even before you record audio.

Keep a running total in the timecode column. Scene 1 starts at 00:00 and runs 6s, so scene 2 starts at 00:06. Scene 2 runs 8s, so scene 3 starts at 00:14. By the last row, your final timecode is your total runtime. If your script is 1,180 words, the math predicts about 7.9 minutes of voiceover at 2.5 wps, so for a tighter 4 to 5 minute cut you trim words on paper before you ever build.

05

The handoff prompt (feed the map, get scenes in order)

Here is the move that keeps Claude from building one giant unmaintainable file: you tell it the file structure you want before you give it the content. You want one component per scene, a parent that sequences them, and shared timing constants. You also want Claude to build in batches so you can review scene 1 through 5 before it commits to a look for all 35.

The structure to ask for:

A scenes/ folder with one .tsx file per scene (Scene01.tsx, Scene02.tsx...).
A Video.tsx parent that uses Remotion's Series or a sequence of Sequence components to play them in order, each with the durationInFrames from your map.
A constants.ts with fps, colors, fonts, and any shared spacing, so the whole video stays consistent and you change the palette in one place.

The creative-direction lock is the other half. Before scenes get built, you give Claude the look once: background color, accent color, font family, the animation feel (snappy vs. slow and weighty), and the easing. Lock that and Claude reuses it across every scene instead of reinventing the style each file. That lock is also what saves tokens, because Claude is not re-deriving the design 35 times. It references constants and builds the next scene.

The swipeable section below has the full handoff prompt and the creative-direction lock prompt ready to paste. Fill the bracketed parts with your map. The key line in it is "build scenes 1 to 8 first, stop, and let me confirm the look before continuing" because that one instruction turns 35 do-overs into one review and a green light.

06

Worked example 1: a 60-second explainer (15 scenes)

Short script, fast turnaround. Here's a real 60-second explainer about a scheduling tool, mapped end to end. Script is about 150 words, which at 2.5 wps is exactly 60 seconds.

Script: "Your team wastes six hours a week just scheduling meetings. Six hours. That's a full work-month every year, gone to back-and-forth emails. Meet Cadence. Drop in your availability once. Share one link. Done. Cadence reads everyone's calendar, finds the overlap, and books it. No more 'does Tuesday work for you' threads. Teams using Cadence cut scheduling time by 80 percent. Setup takes four minutes. It connects to Google, Outlook, and Apple Calendar. Your first three bookings are free. Start today at cadence dot app."

Mapped:

00:00 | Title | "6 hours a week" | Number slams in, screen shakes slightly | none
00:03 | Stat | "A full work-month / every year" | Text stacks, second line fades up | none
00:07 | Transition | "gone to email" | Email icons scatter and fade | email icon
00:10 | Title | "Meet Cadence" | Logo scales in, settles, holds | Cadence logo
00:13 | Bullets | "Availability. Link. Done." | Three words stagger 0.3s apart | none
00:17 | Flow | calendar to overlap to booked | Three boxes connect with animated lines | none
00:23 | Quote | "No more 'does Tuesday work'" | Text types on, strikes through | none
00:28 | Stat | "80% less time" | Counts up 0 to 80, holds 0.5s | none
00:33 | Stat | "4-minute setup" | Number fades up, clock spins | clock icon
00:37 | Image | Google / Outlook / Apple logos | Three logos pop in sequence | 3 calendar logos
00:42 | Title | "First 3 bookings free" | Gold text, gentle pulse | none
00:47 | Title | "cadence.app" | URL fades up, holds | Cadence logo

Twelve scenes, 60 seconds, every asset listed. Notice the on-screen text is never the full sentence. The voice says "teams using Cadence cut scheduling time by 80 percent," the screen says "80%." That contrast is what makes it feel directed instead of captioned.

07

Worked example 2: a 1,180-word script to 34 scenes

This is the real one. A 1,180-word case-study script. At 2.5 wps that is about 7.9 minutes of straight voiceover, too long, so the first job on paper is trimming to a 5-minute target (roughly 750 spoken words) and letting motion carry the rest. After the trim, the map comes out to 34 scenes. Here's how the structure breaks down so you can see the shape of a real one without reading all 34 rows.

The arc (34 scenes, ~5:00 total):

Scenes 1 to 3 (0:00 to 0:18): the hook. A title card, a gut-punch stat about the problem, and a one-line promise. Short scenes, 4 to 6 seconds each, high contrast.

Scenes 4 to 9 (0:18 to 1:10): the problem, deepened. Mostly stat and quote types. The client's pain in numbers. "3 weeks per landing page," "$14k per funnel," "2 freelancers, 0 shipped." Each number is its own scene with a count-up animation.

Scenes 10 to 13 (1:10 to 1:35): the turn. A transition scene, then "Meet the Desk" title, then a flow diagram of how the 25 agents hand work to each other. This is the one diagram-heavy stretch and it gets the most motion-note detail.

Scenes 14 to 24 (1:35 to 3:30): the proof, the meat of the video. Alternating image scenes (real screenshots of shipped work) and stat scenes (the results). This is where the asset column earns its keep. Eleven screenshots, each tied to a specific scene, gathered before the build starts so nobody is hunting for a PNG at 11pm.

Scenes 25 to 30 (3:30 to 4:25): the outcome. Bullets and stats. "40 scenes per session," "1080p in one render," "a twentieth of agency cost." Build momentum with progressively snappier timing.

Scenes 31 to 34 (4:25 to 5:00): the close. Quote, title, CTA, logo hold. Slow the motion back down to let it land.

The lesson from mapping this one: the trim happened on the map, not in the edit. Seeing 34 rows sum to 7:50 told us immediately the script was too long, and cutting words in a table is painless. Cutting them after Claude has built 34 components is not. The map caught it before a single frame existed. That is the whole point of the planning layer. You make the expensive decisions while they're still free.

08

Build Kit: turn this guide into video

You have the map. Now turn it into a watchable file. This is the exact order the Desk runs, and you can run it yourself with the same parts.

Start with the four swipeables at the bottom of this guide, because they are the whole pipeline in paste-ready form. The narration script gives you the words. The scene map gives you the build order. The asset checklist tells you what to gather first. The render recipe finishes it. Run them top to bottom and you will not get stuck halfway.

Step 1: lock the look once. Paste the "Creative-direction lock prompt" you already have above. That writes constants.ts with fps 30, background #0A0A0B, gold accent #C9A24B, Fraunces and Inter. Every scene imports from it. You set the palette in one file, not 16.

Step 2: hand Claude the filled map. Use the "Scene map for the companion video" swipeable as your worked example of a finished map, then paste your own into the handoff prompt. Ask for one file per scene and a Video.tsx parent that plays them with Series. Review scenes 1 through 5 before it builds the rest. If the look is right at scene 5, it is right at scene 16.

Step 3: gather assets from the checklist before you record audio. The checklist swipeable lists every logo, screenshot, font file, and prompt doc the companion video needs. Drop them in public/ with the

exact names the scene map references. A scene that calls for desk-flow.svg fails silently if the file is named flow-diagram.svg.

Step 4: record or generate the voiceover from the narration script, time it, and confirm it lands near the target runtime. If your read comes in long, run the "5-minute trim check" before you touch the scenes. Trimming words is free. Trimming built scenes is not.

Step 5: render with the recipe. The "Render and export recipe" swipeable gives you the exact 1080p and 4K commands and the order to run them so audio and motion stay locked to the same frame.

The point of the Build Kit is that nothing in it is guessed. The map decided every scene's length in frames, so the render is just execution. You are not designing in the editor. You designed on the page, and the editor only confirms it.

09

Your companion video guides

The free PDF is the blueprint. The paid track is the Desk building the companion video on camera, in order, with you following along, so you finish with a rendered file and not just notes.

This is a three-part video guide. It is specific to this topic: taking one script and turning it into a folder of clean Remotion scenes that land on frame. No filler, no general "intro to video" detour. Each part ends with something on your drive.

Part 1, The map, in real time (about 12 minutes). You watch a 1,180-word script get cut into scenes using the chunking rule from this guide. Read aloud, slash at every breath and every idea change, sort each line into on-screen text versus voiceover only. By the end the five-column map is filled, the word counts are converted to frames, and the timecodes line up. You walk away with a completed scene map for your own script and the trim done if it ran long.

Part 2, From map to scenes (about 15 minutes). The creative-direction lock goes in first and writes constants.ts. Then the handoff prompt builds scenes 1 through 5, we check the look on camera, and the rest get built in batches. You see why one file per scene means you can re-render scene 9 without touching scene 8. You walk away with a scenes/ folder and a Video.tsx that plays start to finish in the Remotion preview.

Part 3, Audio, timing, and the final render (about 13 minutes). Drop in the voiceover, confirm the narration finishes on the same frame the motion does, fix the one or two scenes that drift, add the 0.5 second pad to the stat scenes, then run the 1080p render and the 4K master. You walk away with a finished MP4, the audio locked to motion, and a render recipe you reuse on the next script without rewatching anything.

What you have at the end of all three: a repeatable pipeline. Script in, scene map filled, constants locked, scenes built in order, voiceover aligned, file rendered at 1080p and 4K. The next video is the same five steps and goes twice as fast because the look is already in constants.ts.

Swipe file

Copy, paste, adjust. These are the exact prompts and templates.

CREATIVE-DIRECTION LOCK PROMPT (PASTE FIRST, BEFORE ANY SCENES)

Before we build scenes, lock the creative direction and put it in a constants.ts I can import everywhere. Use these exact values and reuse them across every scene:

- fps: 30
- Background: #0A0A0B (near black)
- Primary text: #F5F5F0
- Accent (numbers, key words, CTA): #C9A24B (gold)
- Display font: Fraunces. Body font: Inter.
- Motion feel: confident and snappy, not bouncy. Use spring() for entrances with damping ~14, and interpolate with easing Easing.out(Easing.cubic) for fades and slides.
- Default entrance: elements fade up 24px over 0.4s.
- Every scene gets a subtle 1.02x slow scale on its background across its full duration so nothing feels static.

Do not restyle per scene. Reference constants.ts. Confirm you've created it and show me the file before building any scenes.

THE HANDOFF PROMPT (PASTE YOUR FILLED MAP INTO THE BRACKETS)

You're building a Remotion video from a scene map I filled out. Build it as separate files, not one giant component.

Structure I want:
- src/scenes/Scene01.tsx ... SceneNN.tsx, one file per scene.
- src/Video.tsx: a parent that plays scenes in order using <Series> with each scene's durationInFrames set to the frame count I give you.
- src/constants.ts: already locked (fps, colors, fonts). Import from it, do not hardcode style.

Rules:
- One visual idea per scene. On-screen text is only the words in the "on-screen text" column, never the full narration.
- Set durationInFrames per scene exactly from my map. Do not guess timing.
- Build scenes 1 to 8 first, then STOP and let me confirm the look before you continue. Do not build all scenes in one shot.

Here is the map. Each row is: scene# | timecode | type | on-screen text | motion note | duration in frames | asset.

[PASTE YOUR FILLED MAP ROWS HERE]

Start with constants.ts confirmation, then Scene01 through Scene08, then Video.tsx wiring those eight, then stop.

PLAIN-TEXT SCENE MAP TEMPLATE (COPY, FILL ONE BLOCK PER SCENE)

VIDEO: [title]
TARGET RUNTIME: [m:ss] FPS: 30 WPM: 150 (2.5 words/sec)

```
--- SCENE 01 ---
Timecode (running start): 00:00
Type: [Title | Stat | Bullets | Quote | Flow | Image | Transition]
On-screen text (3-7 words):
Motion note (one line):
Narration words: [n]   ->   Seconds: [n/2.5]   ->   Frames: [sec x 30]
Asset needed: [none | file name]

--- SCENE 02 ---
Timecode:
Type:
On-screen text:
Motion note:
Narration words:    ->   Seconds:    ->   Frames:
Asset needed:

(repeat the block per scene)

AT THE END:
- Total scenes: [count rows]
- Total seconds: [sum the seconds]
- Matches target within 5s? If not, trim words NOW, on paper.
- Asset list: [collect every non-'none' from the asset lines]
```

THE 5-MINUTE TRIM CHECK (RUN THIS ON ANY OVER-LENGTH SCRIPT)

```
My target runtime is [m:ss]. My script is [N] words. At 2.5 words/sec that's [N/2.5]
seconds of voiceover.

If that's longer than my target:
1. Target spoken words = target seconds x 2.5.
2. Cut [current words - target words] words.
3. Cut from connective tissue first ("and so", "basically", "what this means is"),
then from any sentence whose point the motion will already show on screen.
4. Never cut a number or a proof point. Cut the words around it.

Do this in the script BEFORE filling the scene map. Trimming words in a doc is free.
Trimming scenes after Claude built them is not.
```

COMPANION VIDEO NARRATION SCRIPT (PART 1)

```
COMPANION VIDEO, PART 1: THE MAP, IN REAL TIME
Target runtime: 3:00 (about 450 spoken words at 2.5 words/sec)
Voice: calm, direct, one operator talking to another. No hype.

--- VO ---

Here is the thing nobody tells you about making AI video. The model is not your
bottleneck. Claude can write thirty-five scenes in one session. The reason your
videos come out as eight sad scenes that all look the same is that you handed it a
script and said make this a video. So it guessed.

You do not have a video problem. You have a chunking problem. And the fix is one
```

page you fill out before Claude writes a single frame.

Watch. This is the map. Five columns: timecode, scene type, on-screen text, motion note, and the asset each scene needs. That is the whole instrument. More columns and you never finish it.

Start with the script. Read it out loud. Every place you naturally take a breath, draw a slash. Every place the idea changes, draw a slash. Those slashes are your scene breaks. That is the entire chunking rule.

Aim for twelve to twenty-two words of narration per scene. At a hundred and fifty words a minute, that lands each scene between five and nine seconds. Anything under eight words is a punch line or a stat. Give it its own short scene, two to four seconds, so it hits.

Now the split that separates clean from amateur. Not every word goes on screen. Reading along with the narrator word for word is exhausting. On-screen text gets the key phrase, three to seven words, plus numbers, names, and the one word you want to land. Voiceover carries the connective tissue and anything the motion already shows.

Here is the math that makes narration and motion hit the same frame. Seconds equals words divided by two point five. Frames equals seconds times thirty. A fifteen-word scene is six seconds, a hundred and eighty frames. Put that frame count in the map. When the scene's duration equals the narration length, the motion finishes when the sentence finishes.

One adjustment. Add half a second of pad to any scene that ends on a stat. The number should hold after the voice stops.

Fill the timecodes last, once you know each scene's length. They are just the running total. Now narration and motion line up by the end instead of drifting apart.

That is the map. Fifteen minutes by hand. The script is what gets said. The map is what gets seen, and when. Two different jobs. Most people only do the first one.

In Part two, we lock the look and turn this map into a folder of scenes. Fill yours out first. Bring it with you.

--- END VO ---

SCENE MAP FOR THE COMPANION VIDEO (PART 1)

VIDEO: Companion Video Part 1, The Map In Real Time
TARGET RUNTIME: 3:00 FPS: 30 WPM: 150 (2.5 words/sec)
Format: scene | on-screen text | motion note | asset needed

01 | "The model is not your bottleneck" | Headline fades up 24px over 0.4s, holds | none
02 | "8 sad scenes" | Four faint scene-cards stack flat, dim and gray | none
03 | "You have a chunking problem" | "video" strikes through, "chunking" slides in gold | none
04 | "One page. Before a single frame." | One-page sheet scales in 1.02x, settles | none

05 | "5 columns" | Five column headers stagger in 0.3s apart | map-template.png
06 | "Read it out loud" | Script lines fade up one at a time | sample-script.png
07 | "Slash every breath. Slash every idea." | Slash marks draw in between lines, gold | none
08 | "12 to 22 words per scene" | Range bar sweeps left to right, settles | none
09 | "5 to 9 seconds" | Two numbers count up from 0, hold | none
10 | "Under 8 words = its own scene" | Short card snaps in, slight screen shake | none
11 | "Not every word goes on screen" | Full sentence dims, 4 key words stay lit gold | none
12 | "On screen: phrase, numbers, names" | Three labeled chips stagger in | none
13 | "Voiceover: the connective tissue" | Text fades up under a soft waveform | none
14 | "seconds = words / 2.5" | Formula types in character by character | none
15 | "frames = seconds x 30" | Second formula slides up under first | none
16 | "15 words = 6 sec = 180 frames" | Three values chain left to right, gold on the last | none
17 | "+0.5s pad on every stat" | A stat holds, then a half-second bar extends | none
18 | "Fill timecodes last" | Timecode column fills top to bottom 00:00, 00:06 | none
19 | "Now they land on the same frame" | Narration bar and motion bar align, snap together | none
20 | "Fill yours out. Bring it to Part 2." | Eleven Views logo scales in, settles, holds | elevenviews-logo.svg

ASSET AND RESOURCE CHECKLIST (PART 1 COMPANION VIDEO)

COMPANION VIDEO PART 1, ASSET AND RESOURCE CHECKLIST
Gather everything before you build. A scene that calls a missing file fails quietly.

FONTS (install locally and load in Remotion):
[ ] Fraunces (display) - https://fonts.google.com/specimen/Fraunces - weights 400, 600
[ ] Inter (body) - https://fonts.google.com/specimen/Inter - weights 400, 500, 600
[ ] Load both with @remotion/fonts or staticFile woff2 in public/fonts/

IMAGE / GRAPHIC ASSETS (drop in public/, exact names matter):
[ ] map-template.png - the five-column blank map, screenshot from this guide
[ ] sample-script.png - the 1,180-word script as styled text block
[ ] elevenviews-logo.svg - gold-on-dark logo, transparent background
[ ] (no other assets; scenes 01-19 are pure type and motion, asset = none)

PROMPT DOCS (keep open in a tab while building):
[ ] Creative-direction lock prompt (from this guide's swipeables) - paste first
[ ] The handoff prompt (from this guide's swipeables) - paste with filled map
[ ] The 5-minute trim check (from this guide) - only if your read runs long

CODE / PROJECT SCAFFOLD:
[ ] Remotion project initialized (npm create video, Blank template)
[ ] src/constants.ts - written by the lock prompt, holds fps + palette + fonts
[ ] src/scenes/ - empty folder, one Scene##.tsx per row above
[ ] src/Video.tsx - parent with <Series>, durationInFrames per scene

AUDIO:
[ ] Part 1 voiceover - record from the narration script, or generate TTS
[ ] Export as voiceover-part1.wav (48kHz) into public/audio/

[ ] Timed read confirmed near 3:00; if long, run trim check before mapping

PALETTE LOCK (copy into constants.ts, do not hardcode elsewhere):
[ ] fps: 30
[ ] background: #0A0A0B
[ ] primaryText: #F5F5F0
[ ] accent: #C9A24B
[ ] displayFont: Fraunces / bodyFont: Inter

RENDER AND EXPORT RECIPE (PART 1 COMPANION VIDEO)

COMPANION VIDEO PART 1, RENDER AND EXPORT RECIPE
Run in this order. Do not render before audio is aligned, or you render the drift twice.

ORDER OF OPERATIONS:
1. Confirm constants.ts has fps 30 and the locked palette. Everything reads from here.
2. Preview the whole video: npx remotion studio. Watch scenes 1–20 play through once.
3. Drop voiceover-part1.wav into the timeline (an <Audio> in Video.tsx, src=staticFile).
4. Walk the stat and formula scenes (08, 09, 14–17). Confirm each number finishes before the cut. Add the 0.5s pad (15 frames) to any that clip.
5. Re-check total duration matches the audio length within ~10 frames. Fix the one or two scenes that drift by nudging durationInFrames, not the audio.
6. Only now render.

1080p (delivery / social), H.264 MP4:
```
npx remotion render Video out/part1-1080p.mp4 \
  --codec=h264 \
  --width=1920 --height=1080 \
  --crf=18 \
  --pixel-format=yuv420p
```
(crf 18 = high quality, small enough to upload. Keep yuv420p so it plays everywhere.)

4K master (archive / re-edit source), H.264 MP4:
```
npx remotion render Video out/part1-4k.mp4 \
  --codec=h264 \
  --width=3840 --height=2160 \
  --crf=16 \
  --pixel-format=yuv420p
```
(Render the 4K from the SAME composition. If your scenes use rem/percentage layout off constants.ts, they scale clean with no per-scene edits.)

VERTICAL CUT (optional, 9:16 for shorts):
- Add a second composition in Root.tsx at 1080x1920 reusing the same scenes.
- Re-center title and stat scenes; bullets stack tighter. Render at --crf=18.

FINAL CHECKS BEFORE YOU SHIP:
[ ] Audio and motion still locked at the last scene, not just the first
[ ] Gold accent reads #C9A24B on a real screen, not muddy
[ ] No scene under 45 frames (1.5s) unless it is an intentional punch

```
[ ] Filenames: part1-1080p.mp4 and part1-4k.mp4 in out/
[ ] Keep the 4K as the master; re-export 1080p from it only if you must avoid re-
render
```

Want this built into your business?

If you can write the script but freeze at the build, that's the exact gap Eleven Views fills. The Desk runs about 25 Claude-powered agents that take a script through this map and out the other side as rendered 1080p or 4K scenes, dropped into your edit, at roughly a twentieth of what an agency charges. Want it done for you, or want the Desk running inside your own shop? Book a build or ask about licensing the Desk at elevenviews.io. Bring the script. We'll bring the 34 clean scenes. Book a call at atlas.elevenviews.io/book.