

Creative Strategy in Claude Code

The command and session cheat sheet for ad ideation and scripting

Start here. Everything in the core flow below runs in **Claude Code**, in the workspace you built on Day 1. That is the part everyone in the Accelerator has. You do not need anything else to run your whole creative process here.

If you are in **Genesis**, you also have the **Creative OS** (the Day 2 "Creative Strategy Skill File", the `/cs` modes, and the bots). That is an accelerator that sits on top of this, not a replacement for it. Wherever it speeds a step up, there is a **Genesis only** box. If you do not have Genesis, skip those boxes. Nothing in the core flow depends on them.

1. The three rituals (the part everyone skips)

START: feed it before you write

Before a single hook, give Claude Code the three things it needs. This is the "give it X, Y, Z first" you asked about:

1. **WHO** the client file: `clients/{name}.md` (the offer, the rules, the words this brand will not say)
2. **VOICE** `brand/voice.md` (how the brand sounds, corrected over time)
3. **PROOF and PATTERN** the matching swipes (`swipes/{mechanism}/`), past winners (`winners/`), and any research notes you have dropped in

Paste this to open a session:

```
New writing session for {client}.
Read clients/{client}.md, brand/voice.md, and the swipes in swipes/{mechanism}/.
Here is the angle I want to chase: [one line].
Do not write yet. Tell me what you have and what is missing.
```

The "do not write yet" line is the whole trick. It forces Claude to load your context before it generates, so you get your voice and your swipes, not the model's average.

DURING: save vs promote

This is the question you asked, and it has a clean answer. The workspace splits in two:

- **Workbench** holds the work: `copy/` , `creatives/` . Today's drafts live here. This is **save**.
- **Memory** holds the lesson: `winners/` , `losers/` , `brand/` , `clients/` , `workflows/` . This is **promote**.

The rule of the second time: once is noise, the second time is signal, the third time it should already be a saved play. Promote the lesson that should change your NEXT ad, not every line from today.

You want to	Say this / run this	Where it lands
Drop a standing rule mid-flow ("from now on we never say 'unlock'")	"remember this" or <code>/remember-this</code>	<code>brand/voice.md</code> or the client file
Turn a move you have done three times into one command	<code>/save-as-skill</code>	<code>.claude/skills/</code>
Promote the whole session's lessons at the end	<code>/reflect</code>	<code>winners</code> / <code>losers</code> / <code>brand</code> / <code>workflows</code>

Plain version: **save** = the draft sits in `copy/` . **Promote** = the lesson moves into memory so next session starts smarter.

END: log it so next time starts ahead

Two commands, every working session. This is the "run THIS at the end every time" you asked for:

- `/reflect` the deposit. Claude reviews the session, finds what is worth keeping per the data dictionary, proposes where each lesson goes, files what you approve, and updates today's note. About two minutes.
- `/handoff` saves session state so the next session opens where you left off instead of cold.

If you do nothing else, do `/reflect` . It is the difference between a workspace that compounds and one that just gets bigger.

2. Every command in one place

Claude Code the session loop (everyone has these)

Command	When to run
<code>/vault-setup</code>	Once, at the start. Interviews you and fills in your workspace.
<code>/reflect</code>	End of every session. The promote step.
<code>/remember-this</code>	Mid-session. Files a rule or fact the moment it surfaces.
<code>/save-as-skill</code>	When a move has repeated. Turns it into one command.
<code>/see-your-setup</code>	Every couple of weeks. Audits the workspace so it stays lean.
<code>/upskill</code>	Periodic. Flags stale skills and merges duplicates.

Claude Code working and verification (everyone has these)

Command	When to run
<code>/handoff</code>	End of session. Saves state for the next one.
<code>/plan-as-page</code>	Before a big build. Renders the plan as a clean page first.
<code>/convince-me</code>	Before an important step. Claude restates the plan so you catch a wrong turn early.
<code>/adversarial-review</code>	Before you ship. A find-what-is-wrong skeptic pass.
<code>/create-workflow-diagram</code>	Turn a plain-text workflow into a diagram.

Genesis only: the Creative OS modes

These run inside your Creative OS (the Genesis Creative Strategy tool). They are the upstream thinking: generate here, then bring the chosen output back into your Claude Code workspace to write it in your voice.

Important: `/cs` is not a Claude Code slash-menu command. You type `/cs research` as a message to the Creative OS once it is running, not from the `/` popup. Set the Creative OS up one of two ways:

- **In Claude.ai (web):** make a new Project, paste the Creative Strategy skill file into the project instructions, upload the reference files. Then type `/cs research` as a message.
- **In Claude Code:** install the skill (steps below), then start it by typing `/creative-strategist` (the slash menu lists skills by their full name, so `/cs` will never appear there). Once it is running, type the modes below as a message.

Installing a skill in Claude Code (the reliable way). Do not upload the zip and ask Claude to "install it." It will often say it did when it did not. Add the folder manually, the same way as Day 1:

1. Unzip the skill file. You get a `creative-strategist` folder.
2. Drag that folder into your workspace's `.claude/skills/` folder, right next to `reflect`, `remember-this`, and the rest. (In Finder, press `Cmd+Shift+.` to reveal the hidden `.claude` folder.) You should end up with `.claude/skills/creative-strategist/SKILL.md`.
3. Fully quit and reopen Claude Code. It only loads skills on startup.
4. Type `/creative` and pick it. If it shows up, you are set.

Command	When to use
<code>/cs research</code>	Day one on a new client. M.A.P. research + the C.A.S.H. framework.
<code>/cs segments</code>	Have research, need to define who you are targeting.
<code>/cs concepts</code>	Have segments, need 100+ concept ideas (with the Infinite Ad Concepts bot).
<code>/cs coverage</code>	Have concepts, want to see the gaps you have not tested.
<code>/cs write</code>	The methodology view of writing, hook standards enforced.
<code>/cs quiz</code>	Building a quiz funnel.
<code>/cs proof</code>	Upsells, downsells, post-purchase.

Not sure where you are? Type `/creative-strategist` with nothing after it and it routes you.

3. Watch it run: blank to a finished ad

A full pass on a made-up product, `ZenMag` (a magnesium glycinate sleep supplement), done entirely in Claude Code. Every prompt is real text you can type. Where Genesis speeds a step up, there is a box.

Step 1. Load the desk. In your workspace:

```
New writing session for ZenMag.
Read clients/zenmag.md, brand/voice.md, and the swipes in swipes/surprising-culprit/.
Do not write yet. Tell me what you have on this brand and what is missing.
```

Claude reads your files and tells you it has the voice rules and two matching swipes, and is missing your proof. You point it at the proof line in the client file.

Step 2. Ideate from your material, not a blank page.

```
Give me 10 angles for ZenMag, grounded in the swipes you just read and the
research in research/. No blank-page guesses. Lead with curiosity, not pain.
```

You pick one: **the Hidden Culprit.** "It is not stress keeping you up at 3am. Stress burns through a mineral your body needs to stay asleep."

Genesis only. Want a bigger bank first? Run `/cs concepts` with the Infinite Ad Concepts bot in your Creative OS, then paste the strongest ideas back into this prompt so Claude grounds them in your swipes.

Step 3. Aim it.

```
For that angle, who is the sharpest segment and what awareness level are they?
Use the research, do not invent an audience.
```

You land on **tired-but-wired professionals, 35 to 50, problem-aware.**

Genesis only. `/cs segments` plus Build a Buyer gives you the deeper avatar. Carry the segment summary back here.

Step 4. Hooks first, stress-tested.

```
Write 5 hooks for this angle and segment.
Run Luke's Vicious test on each one and tell me which survive.
```

Claude writes five, scores them, and kills the weak ones out loud: *"There is a mineral that helps you sleep' fails, weakest read is 'a thing helps a thing.' Rewrite: 'Your 3am wake-up is a mineral problem, not a stress problem.'"*

Step 5. The script, only after the hook survives.

```
Take the winning hook and write the full short-form video script.
Hook, then payoff, then mechanism, then offer. Keep it caveman simple.
```

Step 6. Check it before it ships.

```
/adversarial-review
```

Then the simplification gate:

```
Rewrite any dense section. Same ideas, same order, simple words.  
Swap Latin-root words for Germanic ones.
```

Step 7. Bank the lesson.

```
/reflect
```

Claude proposes: promote the winning hook structure to `winners/` with the WHY, note the dead "mineral helps you sleep" line in `losers/`, and add "always name a specific time anchor (3am) in sleep hooks" as a voice rule. You approve. Next ZenMag session, Claude already knows all of it.

That is the loop. Blank screen to a finished, checked ad, written in your voice, with the workspace smarter than when you started.

4. The non-negotiables (Luke's standards, on every hook)

These are the Day 2 principles. Ask for any of them by name in Claude Code:

- **The Vicious test.** Strip the hook to its weakest reading. If the weakest read is "something happens when you do something," it fails. A vicious hook has stakes and consequences.
 - **The Hourglass.** The second the hook lands, the sand falls. Get to the point in one or two lines. No "stop scrolling for 60 seconds."
 - **The Caveman standard.** Germanic words, not Latin. "Sweat" not "perspire." "Bad" not "severe." If it looks dense, it is dense.
 - **The Identity rule.** Name the ad in two or three words ("the 3am ad," "the ice cube ad"). Those words go in line one. If you cannot name it, the concept is muddled.
 - **Anti-pain default.** Do not open with "you used to be able to sleep through the night." That is what every competitor and every AI writes. Lead with curiosity.
-

Claude Code Accelerator. Keep this next to your workspace.