

Prompt to Skill Conversion

Turn the whole back-and-forth conversation into a saved skill you can run with a slash command in a brand new chat.

01 When to use this

Use this the second time you catch yourself typing the same instructions. That is the rule: say it twice, make it a skill. It is rung two of the automation ladder and you never skip it.

02 The prompt

This worked. Consolidate everything we just did into a skill.

Name it `<SKILL_NAME>`.

The SKILL.md description line should say to use this whenever anyone asks to `<TRIGGER_PHRASES`, e.g. create ad creative, ad variants, or run a creative pipeline>.

Bake in everything we settled on during this conversation:

- The locked visual profile: `<PASTE_TUNER_JSON_OR_PATH>`
- The output structure and the contact sheet
- Every correction I made to you along the way

Make it context independent. Assume the person running it has zero history with this conversation. Anything you learned from me that the skill needs, write into the file. Do not leave it implied.

Where I currently supply `<VARIABLE_INPUT`, e.g. the product>, make that a parameter the skill asks for or researches itself. It should work on a product it has never seen.

Write any helper scripts alongside the SKILL.md rather than putting long code inside it.

When you are done, tell me exactly what to type to run it.

03 What it is actually doing

A prompt is a thing you type. A skill is that same thing, written down and given a name.

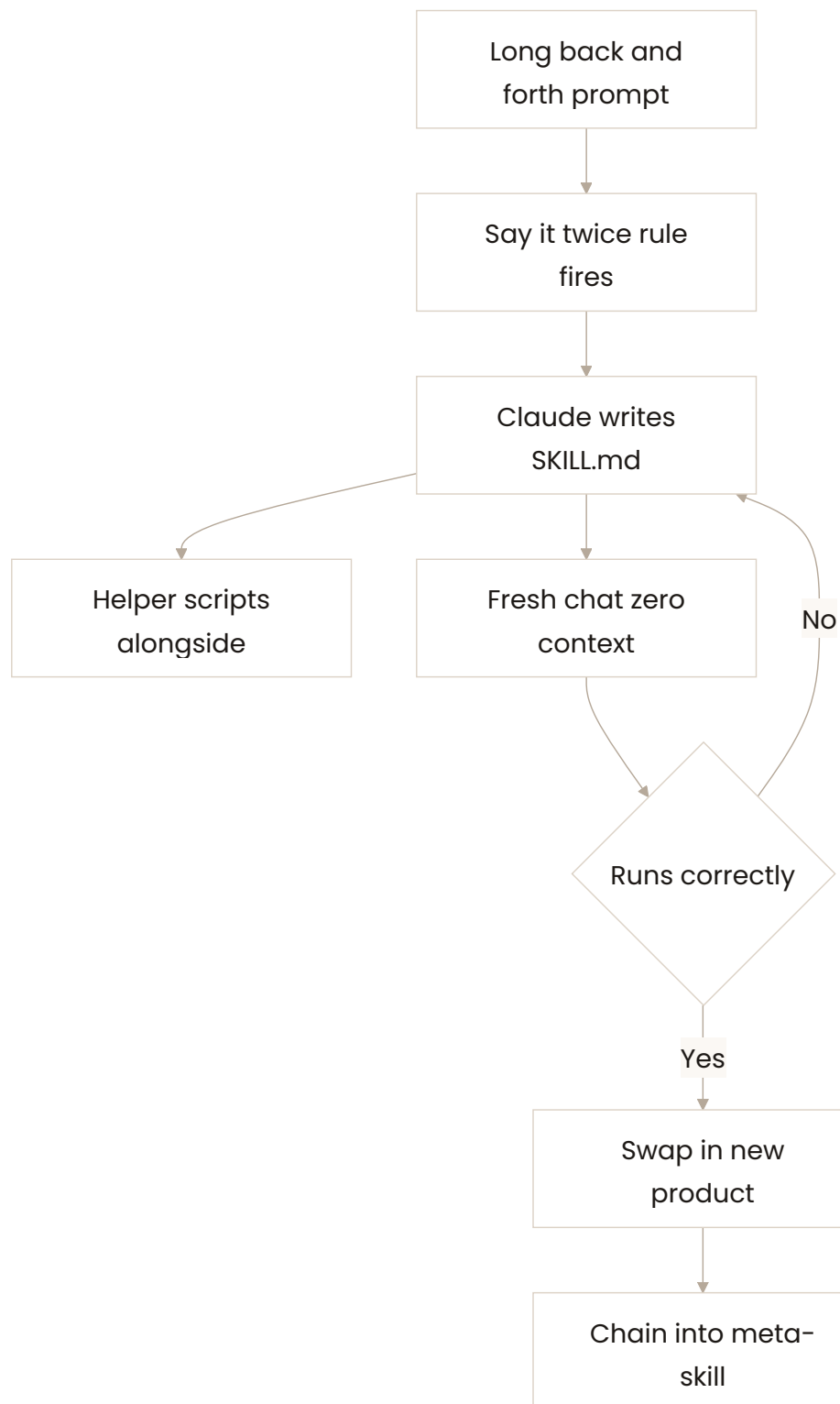
The conversation you just had contains a lot of value that is easy to lose. Every correction you made, every setting you locked, every clarifying question you answered. Ask Claude to consolidate it and all of that goes into a file called SKILL.md, plus any scripts it needs. The first line of that file is a description telling Claude when to reach for the skill, which is how it fires on its own later.

The test for whether the skill really worked is simple and strict. Open a brand new chat with zero context. Type the slash command. If it runs correctly, the skill is real. If it needs you to explain things again, the file is incomplete.

Nick proved his ad skill this way. Fresh session, different product, his dialer startup Clarvo. He typed the command and said research clarvo.io. The skill went and found the product information itself, applied the locked format, and produced a contact sheet. That is the real unlock: context independence plus inputs you can swap.

Skills also stack. You can write small ones and have a master skill chain them. Nick's YouTube channel management example runs title candidates, thumbnails, descriptions, keywords, end screen cards, and publishing in sequence. Together they replace an entire role.

04 How it flows



05 Step by step

01 Wait for the second repeat

Do not codify on the first pass. You want the corrections baked in, and corrections only show up once you have run the thing for real.

02 Ask for consolidation, not a rewrite

Tell Claude to consolidate this conversation. It has the whole thread including your corrections. Starting fresh throws that away.

03 Write a sharp description line

The description tells Claude when to use the skill. Name the phrases a person would actually say, like create ad creative or run a creative pipeline.

04 Turn your inputs into parameters

Anywhere you supplied something by hand, make it a slot the skill asks for or researches. That is what lets it work on a product it has never seen.

05 Push code into scripts

Long code inside a SKILL.md makes it hard to read and easy to break. Ask for helper scripts stored beside it.

06 Test in a fresh chat

Open a new session with zero context. Type the slash command. This is the only test that counts.

07 Swap the input to prove it

Run it against a completely different product. If it researches and adapts, you have a skill. If it stalls, something is still living in your head.

08

Chain into a meta-skill later

Once you have several small skills for one job, write one master skill that runs them in order. That is how you replace a whole role instead of one task.

06 **Words explained**

Skill	A saved set of instructions Claude Code can run later with a slash command. Think of it as a program written in plain English.
SKILL.md	The actual file that holds the skill. A description line at the top says when to use it, and the rest says how.
Context independence	The skill works in a brand new chat with no memory of your earlier conversation, because everything it needs lives in the file.
Slash command	Typing a forward slash and the skill name to run it, like typing a shortcut instead of retyping the whole request.
Meta-skill	A skill whose job is to run other skills in order. One command, six jobs done.
Parameter	A blank the skill fills in each run, like which product to use. It is what makes one skill work on many jobs.

07 Watch out for

- Codifying too early bakes in the version before your corrections, so the skill repeats the mistakes you already fixed.
 - A vague description line means Claude never picks the skill up on its own. Name the real trigger phrases.
 - If it only works in the chat where you built it, it is not a skill. It is a transcript.
 - Hard-coding your one product turns the skill into a single-use script. Make the product a parameter.
 - Long code stuffed inside SKILL.md makes the file unreadable and hard to fix later.
-
-

08 How it connects

Batch Generation and Contact Sheet

That batch conversation is exactly the thing you consolidate here. The corrections you made during it become the skill's rules.

Skill to Loop Conversion

A working skill is the only valid starting point for a loop. You cannot schedule something that still needs you to explain it.

Automation Ladder – Prompt Skill Loop Routine

This note is rung two of the four-rung ladder, and the ladder rule is that you never skip a rung.

Creative Pipeline Kickoff

The kickoff deliberately says the words creative pipeline skill up front, because the whole point of prompting first was to produce this file.

Credit. Framework and approach by Nick Saraev, from the prompt to skill rung of the automation ladder in his Zero to One course on Claude Code for marketing. Written up for the Kape vault by BrewedOps.