

THE PLAYBOOK

Build your own Claude skill

Teach Claude your exact workflow once — the steps, the rules, the voice — and reuse it forever with a single word. Here's how I make mine, plus a fill-in-the-blank prompt that writes one in about five minutes.

WRITTEN BY

Corina · The CV Method

@the_cv_method

A skill is just a saved recipe with a trigger word

Instead of re-explaining the same task every time — the report format, the way you like emails drafted, your brand rules, your review checklist — you write it down once, and Claude runs it your way, consistently.

The best skills are narrow, specific, and honest about their guardrails. A skill that tries to do everything does nothing reliably.

RULE ZERO

**A skill is a recipe with a trigger.
Written once, reused forever.**

If you can't name the trigger and the "done," it's a chat, not a skill
— narrow it until you can.

NARROW

One job, named plainly.

SPECIFIC

Navy and green, not "on-brand."

GUARDED

The nevers, written down.

01

SPOT THE REPEAT

Find the task you explain to AI every week

The report you always reformat, the way you like outreach drafted, your brand rules, your pre-ship checklist. That repetition is the skill. Pick the one task you're most tired of re-explaining — that's the one worth capturing first.

If you've pasted the same instructions to Claude twice, it's already a skill. Go make it one.

02

WRITE THE RECIPE

Brief it like a sharp new hire

The inputs it starts with, the exact steps in order, what "done" looks like, and the guardrails — what it must never do. Be specific: "use our navy and green palette," not "make it on-brand."

Steal from your own past chats — the messages where you corrected Claude are your best rules, already written.

03

NAME IT, TRIGGER IT

A short name and the phrases that launch it

Give the skill a short name and the trigger phrases that should fire it. The trigger is how you — or Claude — call it later without re-explaining a thing.

"draft an outreach email"

"run my review"

"make the monthly report"

04

EXAMPLES AND GUARDRAILS

One good example, one bad — and the hard nevers

A single good example and one bad example teach more than a paragraph of rules — Claude matches patterns. Then spell out the hard nevers: never invent a number, never send without your approval, never use a client's real name.

Examples are the highest-leverage part of the whole skill. Spend your time here.

05

TEST ON A REAL TASK

Run it, watch it drift, fix the recipe

Run the skill on a real job and watch where it drifts from what you wanted. Each miss is a line the recipe is missing — add it. Two or three passes and it's reliable enough to trust.

06

SAVE IT AND REUSE

Drop it in a SKILL.md and call it forever

Save the recipe as a SKILL.md in your skills folder and it's there for good — call it by name, tweak it as your workflow evolves. That's the whole trick: your process, written down once, working for you every time after.

NEXT

The skeleton on the next page fills in the blanks for you — or hands the writing to Claude.

Fill in the blanks, or have Claude fill them for you

A THE SKILL.MD SKELETON

```

---
name: your-skill-name    # lowercase-hyphenated, same as the skill's folder name
description: <one line: what it does and when to use it>
---

## Trigger – say any of these
"<phrase 1>", "<phrase 2>"

## When to use (and when NOT to)
<the situation this is for; the case it should skip>

## Steps
1. <first step – be specific>
2. <next step>
3. <define exactly what "done" looks like>

## Guardrails (hard)
- never <invent a number / send without my approval / use a real client name>
- always <your non-negotiable>

## Example
Good: <a short right answer>
Bad: <a short wrong answer – and why it's wrong>

```

B OR HAVE CLAUDE WRITE IT – PASTE THIS

Turn this into a reusable skill for me.

Here's the task I keep re-explaining to you:

<paste your instructions, or a past chat where you corrected me>

Write it as a SKILL.md with: YAML frontmatter (name + one-line description), trigger phrases, when to use (and when not to), numbered steps, guardrails (the hard "nevers"), and one good + one bad example. Keep it tight and specific to how I actually work – no fluff.

WHERE TO GET SKILLS

OFFICIAL

Official skills plus the skill-creator live at github.com/anthropics/skills — clone it, or in Claude Code run `/plugin marketplace add anthropics/skills` then install a set like document-skills.

ANY SKILL

Install through the plugin marketplace in Claude Code (`/plugin`) or via the API. On Claude.ai (every plan), turn on code execution and upload the skill folder as a ZIP under Customize → Skills. In Claude Code, skills live in `.claude/skills/` (one project) or `~/.claude/skills/` (all projects).

BUILD YOUR OWN

Use the skeleton on page 05 — or the skill-creator from that repo, which scaffolds one for you.

DOCS

support.claude.com → "what are skills" · the open standard: agentskills.io

THE STARTER CHECKLIST

- | | |
|--|---|
| ☐ The task you re-explain most, picked | ☐ The recipe: inputs · steps · done · nevers |
| ☐ A name and the trigger phrases | ☐ One good and one bad example |
| ☐ Tested on a real task, recipe patched | ☐ Saved as a SKILL.md you can reuse |

HONEST AND SAFE You approve before anything sends or ships · no invented metrics · real work stays anonymized · keep secrets out of the skill file · skills propose, you decide.

WANT THIS DONE FOR YOU?

I turn the things you re-explain every week into a clean set of skills

Your steps, your voice, your guardrails — so Claude runs your process the way you would. That's the CV Method.

[Book a discovery call →](#)



hello.thecvmethod.com