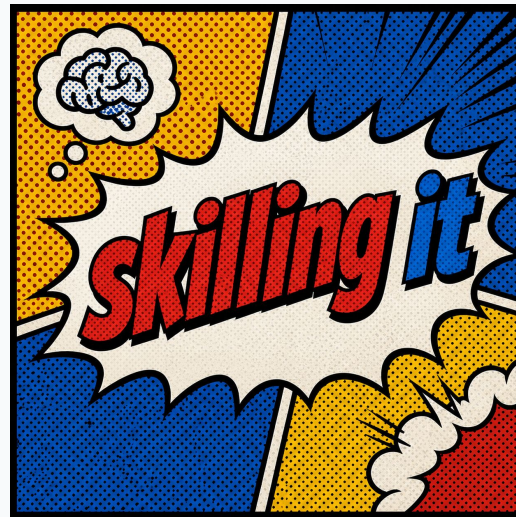


SKilling it!



Don't build agents. Build skills.

What skills are, when you need one, and how to write one well.

At-a-glance

1 A skill is a folder (app for agents)

SKILL.md + references + assets + scripts

2 Progressive disclosure

SKILL.md is the router. References are the depth.

3 Description triggers the skill

Write it for agents, not for humans. Measure invocation.

4 Gotchas are your best content

Build them from real failures over time → learn from experience. Measure performance

5 Do the work first

Live session → red/green/refactor → ship.

START HERE Open your favorite coding agent → do the work → /branch off → /skill-creator → evals → add a gotcha when it fails

THEN /skill-creator built-in evals to measure if it's actually helping. keep improving as you go → gotchas

Golden rules: Curate, don't generate. 2–3 focused skills > one giant skill. Measure performance. Discard what doesn't produce value. Beware of open-source skills (security risk). Don't skill what is already known. Heaviness and models behave differently,

Resources

[SkillsBench](#) — empirical evaluation

[Claude Code Documentation](#) — Skills

[The Complete Guide to Building Skills](#)

[Anthropic](#) — Equipping agents for the real world (Oct 2025)

[Anthropic](#) — How We Use Skills

[Vercel](#) — AGENTS.md outperforms Skills in our evals

[Agent Skills Open Standard](#)

[My skills repo, fork it](#)

Built into Claude Code: [/skill-creator](#) · [/superpowers:writing-skills](#)

SKilling it!

Thank you

Go home and build your skill.
Today.

