Founder Skill — Full Setup Guide

How to install Jake Schincariol's open-source Founder skill pack (github.com/Jakeschincariol/founder-skill) so that one idea gets the whole treatment: board validation, competitor analysis, a 100-buyer consumer panel, pricing, offer, real unit economics and a launch plan, compiled into one business plan with a computed verdict.

1. What you are installing

Eleven skills that hand work to each other through a `founder/` folder in your project. Four of them ship small Python tools (standard library only, no network, no API keys).

Skill	What it does for you
/founder-board	Validation. Three sub-agents score the idea through the Offers, Monopoly and Product lenses, name what kills it, and vote fund / fund-if / pass.
/founder-competitors	Competitor analysis. Direct competitors, substitutes, their prices, positioning and customer complaints, every fact linked.
/founder-consumer	The 100 fake customers. One sub-agent per simulated buyer, each with its own income, habit and objection. Tallies who buys, who passes and why.
/founder-pricing	Turns the panel's price answers into an acceptable price range (Van Westendorp).
/founder-offer	Builds bonuses, guarantee and urgency against the top objections, then re-tests on the panel.
/founder-cfo	Unit economics: margin per sale, break-even per day, year 1 by month, cash needed, payback.
/founder-marketing	Positioning, channels, a dated 30-day campaign, hooks, max cost per customer.
/founder-brand	Names, voice, and the trademark/domain/handle checks to run.

Skill	What it does for you
/founder-ops	Suppliers, staffing, routines, permits to check, risk register.
/founder-launch	A cheap real-world test first, then a dated launch plan.
/founder-plan	Compiles everything into business-plan.md with a verdict: Profitable, Not yet or Incomplete.

2. Where to run it (this decides whether the 100 buyers actually happen)

The board and the consumer panel work by launching sub-agents, and the tools need Python plus a writable project folder. That combination is Claude Code.

- **Claude Code (recommended, full power):** sub-agents, Python tools and the shared `founder/` folder all work. This is the only setup where all of it happens as advertised.
- **Claude app (claude.ai / desktop / mobile):** you can upload individual skills, but expect the panel and board to run inside one conversation rather than as 100 independent agents, and the hand-off between skills is weaker. Fine for a quick look, not for the full run.
- **No install at all:** paste one SKILL.md at the top of a chat and it runs as a mode. You lose sub-agents and the Python tools.

3. Prerequisites

1. Claude Code installed and signed in (see claude.com/claude-code). Check with `claude --version`.
2. Python 3.8 or newer. Check with `python3 --version`. Nothing to pip install.
3. Git, to clone the repo.
4. A plan with headroom. A full panel is roughly 100 sub-agent calls plus 3 for the board. On a lighter plan, start with `--quick` (20 buyers).

4. Install — pick one

Option A: plugin (easiest to update). Inside Claude Code:

```
/plugin marketplace add Jakeschincariol/founder-skill
/plugin install founder-skill@founder-skill
```

Commands then appear namespaced, e.g. `/founder-skill:founder-board`.

Option B: global skills (short command names). In a terminal:

```
git clone https://github.com/Jakeschincariol/founder-skill.git
cp -r founder-skill/skills/founder-* ~/.claude/skills/
```

Commands appear as `/founder-board`, `/founder-consumer`, etc., in every project.

Option C: project-local (good for client work). Copy the same folders into `<your-project>/.claude/skills/` so the pack travels with that one project only.

Verify the install (optional, from the cloned repo):

```
cd founder-skill
python3 -m unittest discover -s tests -v
```

You should see 29 tests pass.

5. Set up a project for one idea

Use one folder per business idea, since every skill reads and writes `founder/` inside it.

```
mkdir ~/ideas/my-idea && cd ~/ideas/my-idea
claude
```

Before the first skill, have these ready (the board asks for anything missing, max three questions):

- What it is, in one sentence
- Who it is for, as specific as possible (age, income, location, habits)
- What it sells and at what price
- Where and how it is sold and delivered
- Budget and constraints

For the CFO step, gather real numbers if you can: cost per unit, monthly fixed costs (rent, salaries, software), startup costs, capacity per day, and a realistic ramp. Supplier quotes beat guesses.

6. Run the full pipeline

Run in this order; each skill reads the previous ones' files.

`board → competitors → consumer → pricing → offer → cfo → marketing → brand → ops → launch → plan`

1. **/founder-board** <your idea in a few lines> — writes `founder/idea.md` and `founder/board/`, returns three scores, kill risks and votes. The "fund if" conditions become your checklist.
2. **/founder-competitors** — writes `founder/competitors.md`. Run this before the panel: competitor complaints make the simulated buyers sharper.
3. **/founder-consumer --quick** — 20 buyers first. Claude shows you the customer profile (`founder/customer.json`) and the pitch (`founder/pitch.md`) before dealing. Correct them here; a wrong customer makes the whole panel wrong.
4. **/founder-consumer** — the full 100 once the quick run looks sensible (details in section 7).
5. **/founder-pricing** — reads the four price answers per buyer and gives the acceptable range against your price and competitors.
6. **/founder-offer** — answers the top objections and re-tests on a quick panel to see if the buy rate moves.
7. **/founder-cfo** — builds `founder/numbers.json` with you and runs the unit-economics tool. Every estimate is flagged as an estimate.
8. **/founder-marketing**, **/founder-brand**, **/founder-ops**, **/founder-launch** — campaign, naming, operations and a dated launch that starts with a real-world test.
9. **/founder-plan** — compiles `founder/business-plan.md` with the computed verdict.

You can run any skill on its own too. `/founder-consumer --quick` on a raw idea is the fastest gut check.

7. Making the 100-buyer panel run smoothly

- **Turn on accept-edits mode** before the run (press Shift+Tab in Claude Code until it shows accept edits). Each buyer writes one answer file; without this you approve 100 file writes by hand.
- **It runs in waves of 10.** Claude Code runs about 10 sub-agents at a time, so 100 buyers is 10 waves. Let it finish; don't interrupt mid-wave.

- **Same seed, same buyers.** The cards are dealt from a recorded seed, so you can re-run after changing the pitch or price and compare like for like. Use `--seed` to change the crowd.
- **Use real customer data when you have it.** Interviews, reviews, survey answers or WhatsApp chats make `customer.json` far better than guesses. Tell Claude which parts are guesses.
- **What you get back:** the buy/pass split, buy rate by segment and behaviour, the top reasons people pass in their own words, what would flip a no, and price answers for the pricing step. Files land in `founder/panel/results.md` and `results.json`.

8. Troubleshooting

- **Command not found:** with the plugin install, use the namespaced name (`/founder-skill:founder-consumer`). With the copy install, restart Claude Code after copying.
- **Tool path not expanded:** if Claude can't find `panel.py` or `unit_economics.py` , tell it to use the skill's base directory that Claude Code printed when the skill loaded.
- **Some buyers didn't answer:** the skill runs `panel.py check` , then `prompts` again for the missing ones only. A buyer that fails twice is left out and counted.
- **A buyer printed JSON in chat instead of writing the file:** Claude files it with `panel.py save <id>` ; it rejects anything invalid.
- **Usage limits hit mid-panel:** stop, then run `check` and `prompts` later. Only the missing buyers re-run.
- **CFO says a field is missing:** that is by design; it never guesses a number. Supply it.

9. Read the results honestly

- The 100 buyers are simulated. They are excellent at surfacing objections, segments and wording, and poor at predicting real conversion. Language models lean agreeable, so treat the buy rate as an upper bound.
- Confirm the big calls with real people: a waitlist, a pre-sale or a landing page with a price. /founder-launch starts with exactly that.
- Never use panel quotes as testimonials.
- The board applies published frameworks; it does not speak for Hormozi, Thiel or Jobs.
- Numbers are only as good as the inputs. Check costs with suppliers and structure, payroll and tax with an accountant. Not financial, legal or tax advice.

10. Copy-paste kickoff

Open Claude Code in a fresh idea folder and paste:

```
/founder-board I want to launch <what it is> for <who, specifically> in
<where>, selling <product> at <price>. Budget is <amount>, and <any
constraint>.
```

Then, in order: `/founder-competitors`, `/founder-consumer --quick`, `/founder-consumer`, `/founder-pricing`, `/founder-offer`, `/founder-cfo`, `/founder-marketing`, `/founder-brand`, `/founder-ops`, `/founder-launch`, `/founder-plan`.

The finished plan is `founder/business-plan.md`.