

FREE PROMPT KIT FOR MARKETERS AND FOUNDERS

The Fable 5 Handover Prompt Kit

Six copy-paste prompts that turn your most capable AI model into written playbooks your everyday models can follow, long after the included window closes.

By Bige Besikci Yaman · Digital Marketing Consultant · bigebesikci.com

Included Claude Fable 5 access on paid plans runs through July 19, 2026 at 11:59:59 PM PT. From July 20, Fable 5 draws on prepaid usage credits at \$10 per million input tokens and \$50 per million output tokens. Anthropic has said the model is not permanently leaving subscriptions and aims to restore it once compute allows, but nothing is committed beyond July 19.

This kit assumes you will not have cheap Fable access forever. The prompts below do not ask Fable to do your work. They ask Fable to write down how it does your work, in plain instruction files that a cheaper model can read and follow afterwards. Run them in order. Steps 1 through 4 take one focused afternoon.

50% of your weekly plan limit is the Fable ceiling, and it burns that pool faster than Opus or Sonnet. Prompt 6 is the hungriest. Run it last.

READ THIS BEFORE YOU START

You are not training a new model here, and nothing technical is happening behind the scenes. Training a smaller model on a frontier model's outputs is prohibited by Anthropic's terms, and that is not what this is.

What these prompts capture is working knowledge: how the model approaches your tasks, written in plain language, in your own files, for your own workflows. That is exactly what the tooling was built for.

1 • Find what is worth writing down

When: First. Do not skip to writing playbooks before you know which ones matter.

You are currently the most capable model I have access to. Soon I will be running most sessions on cheaper models.

Before we write anything, I want you to study me. Look at everything you have access to: our past conversations, my project files, my published work, my connected tools.

Then give me a ranked list of the 8 to 10 recurring workflows in my work.

Ranking rule: a workflow ranks HIGHER the more its output quality depends on YOUR intelligence rather than on tool access or raw effort. A workflow that any model handles equally well ranks LAST, no matter how often I run it.

For each workflow, tell me in one line each:

- What the workflow is
- What specifically you do that a weaker model would not do
- What breaks in the output when a weaker model runs it

Do not write the playbooks yet. Just show me the ranked list and defend the top three.

The defence matters. If Fable cannot explain what it adds, that workflow does not need Fable.

2 • Turn one workflow into a playbook

When: Once per workflow, starting with your number one. Run this fresh each time.

Write me a complete instruction file for workflow #[N] from your ranked list. This kind of file is often called a skill.

Assume the model reading this file later is less capable than you. It will not read between the lines. It will not push back unprompted. It will not fill gaps with judgment. Everything you currently do without being asked must become a written instruction.

Structure it as:

- A name and a one-line description of when this playbook should be used
- The trigger conditions, phrased as things I might actually type
- The step-by-step procedure, in order, with no steps assumed
- The decision rules: at each point where you would use judgment, write the rule you are actually applying
- The common mistakes: what a weaker model will get wrong here, and the instruction that prevents each one
- A worked example: a real input and the finished output you would produce

Be long where being long prevents a mistake. Be specific about my context, never generic advice.

When you are done, offer to write out the full worked example at length.

Say yes to the worked example every time. Examples transfer better than rules do.

3 • Test it before you lose the reviewer

When: Immediately after each playbook. Not next week.

Here is an instruction file that was written for you to follow.
[paste playbook]

Here is a real task: [paste task]

Run the task following the instructions exactly. Do not improvise. Where the instructions are unclear or leave you guessing, stop and flag it instead of filling the gap yourself.

At the end, list every point where you had to guess.

Run this on Opus or Sonnet, in a fresh conversation, with no other context loaded. Every flagged guess is a hole. Take the list back to Fable and have it patch the playbook.

4 • Upgrade the file that shapes every session

When: After two or three playbooks exist, so Fable can see the pattern.

My standing instructions file is the only thing that carries over between models when I switch. Here it is: [paste your project instructions or custom instructions]

Review it as if you were checking work you inherited from someone less careful than you.

Specifically:

1. Where does it contradict itself?
2. Where does it state a preference but not the reason behind it, so a weaker model will apply it in the wrong situation?
3. Where does it teach a habit that my own stated standards would reject?
4. What am I relying on your judgment for that is not written down anywhere?

Then rewrite it so cheaper models inherit your standards, not just my preferences. Keep it tight. Every line must earn its place, because this file loads into every single session I run.

Question 3 is the one that pays. Frontier models catch contradictions in your own instructions that you have read past a hundred times.

5 • Set up the engine: one brain, many hands

When: Once your playbooks exist. This is how you run them from July 20 onwards.

I want to set up a division of labor for my work, where an expensive model plans and reviews, and cheaper models do the execution.

My recurring workflows are: [paste the ranked list from Prompt 1]

My available models are: [list them]

For each workflow, tell me:

- Which steps genuinely need top-level judgment (the leader)
- Which steps a mid-tier model handles at equal quality (senior worker)
- Which steps a cheap fast model handles fine (junior worker)
- Where the handoffs happen and what gets passed at each one

Rule for the leader: it should never touch raw source material directly. It plans, it delegates, it reviews the returned work. Keeping its attention clear is the whole point.

Give me the full assignment table, then tell me honestly which workflows are not worth splitting because the coordination costs more than it saves.

LAYER	DOES	NEVER DOES
Leader	Plans, breaks work down, reviews, decides	Reads raw files, does the typing, runs the chores
Senior worker	Analysis, drafts, judgment calls inside a lane	Decides scope or approves its own output
Junior worker	Formatting, extraction, repetitive rewrites	Anything where being wrong is expensive

The economics are the surprise here. People running this setup report their bills going down, because the expensive model only pays for thinking.

6 · Build the vault that briefs every future model

When: Last. This is the longest run and the one most likely to eat your weekly allowance.

I want to build a reference vault: a connected knowledge base that will brief every model I run from now on, on any provider, for years.

Sources: [my best performing content, positioning docs, customer research, competitor notes, the reference material in my niche worth knowing]

Build it as a set of linked notes where:

- Each note covers one concept, pattern, or decision rule
- Each note states the principle first, then the evidence, then the exceptions
- Notes link to each other where the concepts genuinely connect
- Anything debatable is marked as debatable, with both sides

Focus on what a weaker model would get wrong without this vault. Skip what any model already knows.

Start with an index of the notes you plan to write and let me approve it before you write them all.

Approve the index before it writes. This is where the allowance goes.

What does not carry over

I want to be precise about the limits, because the internet is currently overselling this pattern.

- A playbook carries over procedure. It does not carry over judgment on genuinely new problems.
- When a task falls outside the playbook, a cheaper model following the playbook still behaves like a cheaper model.
- Structured reasoning prompts narrow the gap. They do not close it.
- Playbooks go stale. When your positioning or your product changes, the playbook is now confidently wrong.

That is fine, because most marketing work is not new. It is the same fifteen workflows executed well, repeatedly. Capture how the best available model runs those fifteen on your specific business and you have converted a temporary access window into a permanent asset. The subscription is rented intelligence. The playbook library is owned infrastructure.

Not sure which workflows are worth writing down?

The SaaS Growth Audit reviews your positioning, funnel, retention, and analytics in one pass and delivers a prioritised action plan in 5 business days.

Contact me · bigebesikci.com/contact