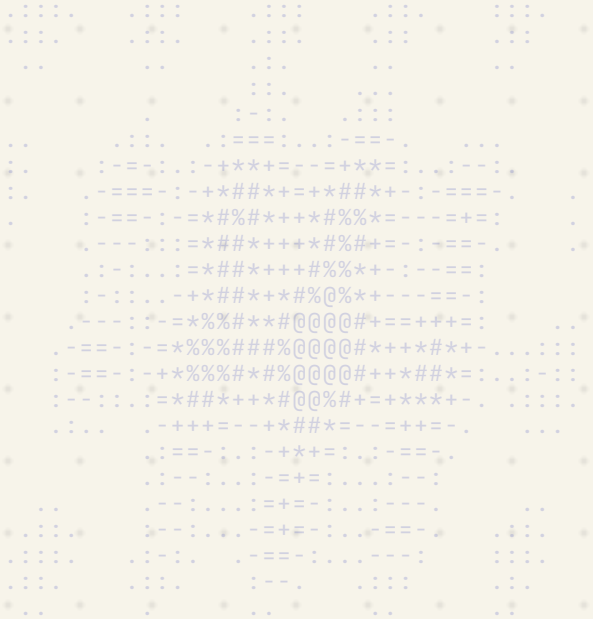
// SECTION 01 · PROMPT PACK

SECRET PROMPTS

Seven prompts that make an AI hand over what it already has and never volunteers: an honest read on you, what a professional would charge, the question you forgot to ask, and the autopsy of your plan failing.

```
> boot opusjake_os
> resource: secret-prompts
> version: v1.0 · 2026
> status: ready_
```

The model is not holding out on you because it is censored. It holds out because default chat is tuned to be agreeable, fast, and finished. Every prompt in here forces it to hand over something it already has and never offers unasked.



HOW TO RUN THESE

These are not task prompts. They run *on* your task — before it, against it, or after it. The task itself is still yours to write.

- Paste each one verbatim. Fill only the `[BRACKETS]`.
- One at a time. Run them in sequence across a session, never stacked into a single message.
- If the answer comes back instantly and agreeably, it did not run. Reply "run that properly, one step at a time" and re-paste.
- Answer honestly when they ask you something. Six of the seven are only as good as what you put in.

> Tested on Claude, ChatGPT, and Gemini. Only the first one depends on the model remembering your past conversations.

01 · THE MIRROR

Every model you use has read more of your unedited thinking than most people in your life. It has never once been asked to describe what it found.

```
Based on everything you know about me from our conversations, answer as a
consultant I am paying to be useful, not liked.

1. What patterns show up in what I ask you — including ones I have never
   pointed out myself?
2. What am I consistently avoiding, deferring, or asking around instead of
   asking directly?
3. What am I better at than I appear to think, based on evidence in our
   conversations rather than encouragement?
4. If I keep operating exactly like this for twelve months, where do I most
   likely end up, and what breaks first?

Quote the specific things I actually said that led to each conclusion. If you
do not have enough evidence for one of these, say so instead of filling it in.
```

Why it works. The quote requirement is the whole prompt. Without it you get a horoscope: flattering, general, true of anyone. Forced to cite your own sentences back to you, the model can only say things you supplied the evidence for.

Run it when you have real history — ChatGPT with memory on, or a Claude Project you have worked inside for weeks. On a cold chat it is a party trick. On six months of context it is uncomfortable.

02 · THE PRICE TAG

The model answers at the level your question implies. A casual ask gets a casual answer, and nothing in the system sets the bar higher. A price does.

```
Before you answer: tell me what a top independent professional would charge
for this exact deliverable, and what they would hand over at that price —
scope, depth, format, and what they would check before sending it.

Then tell me what the cheap version looks like and what is missing from it.

Then deliver the expensive version. If you cannot reach that bar with what I
have given you, tell me what you are missing before you start.
```

Why it works. "Be thorough" is not executable. A price is: it implies a scope, a format, and a standard the model can actually aim at. The scope list is half the value — it is a checklist of what a professional would have included that you never thought to ask for.

Run it when the work leaves your hands. Client deliverables, proposals, anything with your name on it.

03 · THE ONE-QUESTION LOOP

Most bad output is a briefing problem. Ask a model what it needs to know and you get twelve questions in a wall, which you skim and never answer. One question, you answer.

```
Do not start yet. Interview me first.

Ask me ONE question at a time — always the single question whose answer would
most change your output. Wait for my answer before asking the next one.

Keep going until you could do this at the level of someone who has done it a
hundred times. Then stop, say READY, and give me a two-line summary of what
you are about to make.

Rules: no multi-part questions, no lists, no answering your own questions, and
do not start the work until I say go.
```

Why it works. Each answer changes what the next question should be, so the interview adapts instead of front-loading a generic form. And a single question in the chat window gets answered, which is the entire point.

Run it when the task is bigger than the prompt — anything you would spend fifteen minutes briefing a freelancer on.

04 · THE PREMORTEM

"What could go wrong" gets you a list of generic risks. Assume it already went wrong and the same model gets specific and unkind.

```
It is [SIX MONTHS] from today. [THE PLAN] failed — not slowly, obviously.

Write the autopsy. For each cause of death: what actually happened, the first
warning sign we saw and explained away, and the cheapest thing we could have
done this week to prevent it.

Rank causes by probability times damage, not by how dramatic they are.

Include at least one failure that is my fault, one that is nobody's fault, and
one that comes from the thing I am currently most confident about.

End with the single check I should put in my calendar, and the date.
```

Why it works. This is prospective hindsight. A 1989 study found that imagining an outcome has already happened produced roughly 30 percent more explanations for it than forecasting the same event did — and more specific ones. Gary Klein turned the effect into the premortem, and it survives the jump to a language model intact.

The confidence clause is where it pays. Your biggest risk is never on your list, because you are not watching it.

Run it when money or months are about to move. Before you sign, hire, launch, or commit a quarter.

05 · THE REVERSE PROMPT

You do not have to get better at prompting if the model writes the prompt.

```
Do not answer my question yet.

First, write the prompt I should have written to get the best possible answer
— including the context I forgot to give you, the constraints I should have
set, the output format I should have asked for, and the questions I did not
know to ask.

Show me that prompt. Mark anything you had to guess about with [ASSUMPTION].

Then run it.
```

Why it works. The model has seen an enormous number of well-specified briefs and can reconstruct one from a vague ask. The `[ASSUMPTION]` tags are the real output: each one is a fact only you have. Correcting three of them is usually the difference between a generic answer and yours.

Run it when you keep getting reasonable answers to the wrong question. It is also the fastest way to learn prompting — read what it writes about your own ask.

06 · THE PANEL

Ask one question, get one averaged answer: the safe middle of everything the model has read. Contested questions do not have a safe middle.

```
Do not give me the consensus answer.

Convene three specialists who would genuinely disagree about [TOPIC]. Name
each one's discipline and the bias that comes with it. Have each give their
read, then attack the other two by name.

Then bring in a fourth voice — the chair — who rules: who is right, what the
other two are right about anyway, and the single piece of evidence that would
flip the call.

No false balance. If one of them is simply correct, say so.
```

Why it works. Assigning positions forces the model to generate the arguments that live at the edges instead of collapsing them into the mean, and the cross-attack surfaces assumptions the consensus buries. The last line stops it from ending in a diplomatic non-answer.

Run it when the decision is genuinely contested — pricing, architecture, strategy, hiring — and the easy answer arrived suspiciously fast.

07 · THE HANDOFF

Every long session ends with everything in the context window and nothing on disk. "Summarize this chat" gets you a recap. This gets you a working document.

```
Write the handoff brief that would let a competent stranger — or a fresh AI
with zero memory of this conversation — pick this up cold and continue without
asking me anything.

Include: the goal in one line; every decision we made and the reasoning behind
it; the options we rejected and why they died; the current state of every file,
asset, or deliverable; open questions and who has to answer them; and the exact
next action, specific enough to start in five minutes.

Write it for someone competent who was not here. Do not recap the conversation
itself.
```

Why it works. The rejected options are the part you always lose. Six weeks later you re-propose the thing you already killed and spend a day re-learning why. Banning the recap is what turns a transcript into a document.

Run it when you close a long session, hand work to somebody else, or notice the model forgetting the top of the thread — which has its own early-warning trick.

THE RUNNING ORDER

THE MOMENT	THE PROMPT
Starting anything bigger than one message	03 · One-Question Loop
Good answers, wrong question	05 · Reverse Prompt
Work that leaves your hands	02 · Price Tag
Before money or months move	04 · Premortem
The easy answer came too easy	06 · Panel
Closing a long session	07 · Handoff
Once a quarter, on your own	01 · Mirror

On a real project they run in sequence: **05** to fix the ask, **03** to fill the gaps, **02** to set the standard, then the work, then **04** before you commit to it and **07** when you stop. Five prompts, one deliverable, and you never wrote a brief.

Two of these belong in your settings rather than your clipboard. Put the standard from **02** into ChatGPT's custom instructions, a Claude Project, or your `CLAUDE.md`, and it applies to everything without a paste. The rest are situational by design — a premortem you run every day stops being a premortem.

Want the one-word versions that flip behavior mid-conversation instead of full blocks? That is The Code Book. Want the model to tell you how much of its answer it actually believes? That is The Truth Prompt. More builds every week at opusjake.ai.