// SECTION 01 · SYSTEM PROMPT

THE TRUTH PROMPT

One system prompt that makes any AI check its logic, sort facts from guesses, catch its own bias, and stamp every answer with a confidence score. Stop treating confident tone as proof.

```
> boot opusjake_os
> resource: truth-prompt
> version: v1.0 · 2026
> status: ready_
```

AI answers everything in the same voice. A fact it can source and a guess it invented arrive in identical confident prose, and you cannot hear the difference. This prompt installs a checkpoint. Before anything reaches you, the model audits its own logic, sorts what it knows from what it is guessing, checks whether it is just agreeing with you, then puts a number on how much you should trust it.

WHY IT WORKS

The problem is not that models lie. It is that they never flag which parts are shaky. Fluent delivery is the style default, so a claim the model half-remembers from training data gets the same polish as one it could defend in court. You end up grading tone instead of truth.

The fix is already inside the model. In 2022 Anthropic published a paper called "Language Models (Mostly) Know What They Know." The finding: ask a model to judge the probability that its own answer is true, and larger models score themselves with real calibration. The self-grading ability exists. Default chat just never invokes it. The Truth Prompt invokes it on every message.

The reason it needs to be this specific: "be accurate and unbiased" is a useless instruction, the same way "sound more human" is useless. The model has nothing to execute. Three named checks with named outputs give it a procedure, and models are good at procedures. Logic gets attacked, facts get binned, bias gets asked about directly, and the score has defined bands instead of vibes.

The score earns its keep comparatively. Whether 82 "really" means 82 percent matters less than this: when one answer says 95 and the next says 55, you know exactly where your checking time goes.

THE PROMPT

Paste this at the start of a session. It holds for the whole conversation.

```
TRUTH MODE. From here on, run every answer through three checks before it reaches me. Show me results, not process.

1. LOGIC. Draft your answer, then attack it. Does the conclusion follow from the premises? Kill anything resting on a leap, a false choice, or correlation dressed up as cause. If two of your claims contradict each other, resolve it before answering.

2. FACTS. Sort every factual claim into three bins: VERIFIED (settled knowledge, or you can name the source), RECALLED (from training data, could be stale or garbled), GUESS (plausible pattern-match, unchecked). Label RECALLED and GUESS claims where they appear. Never deliver them in the same voice as VERIFIED. If you have web access, check the load-bearing claims. Flag anything that may have changed since your training cutoff.

3. BIAS. Ask yourself: are you agreeing because I clearly want you to? Defaulting to the safe, mainstream take? Would you answer differently if I argued the opposite? If yes, say so in the answer.

Then answer normally, and end every substantive answer with:

TRUTH CHECK
CONFIDENCE: NN/100, plus one line on what the number rests on
SHAKIEST CLAIM: the single claim most likely to be wrong
FLIP CONDITION: what evidence would change this answer

Score honestly: 90-100 verified or provable. 70-89 solid, minor inference. 50-69 plausible, verify before acting. Below 50 is a lead, not an answer: name what to check first.

If my idea is flawed, tell me straight and tell me why. Wrong and clear beats wrong and polite.
```

THE ONE LINER

No time for the full block? This version gets you the number in any chat:

```
Before you answer: check your logic, label which claims are verified vs recalled vs guessed, check whether you are just agreeing with me, then end with CONFIDENCE: NN/100 and your shakiest claim.
```

The full version is still the one to install. The three bins are what keep the score honest, because the model has to look at its own sourcing before it picks a number.

READ THE SCORE

The bands are instructions to you, not just labels.

- **90 to 100.** Verified or provable. Act on it.
- **70 to 89.** Solid with some inference. Act, but know which claim was inferred. It is named in the answer.
- **50 to 69.** The model is telling you to verify before you act. Listen. This band is the whole reason the prompt exists: claims you would have shipped now arrive pre-flagged.
- **Below 50.** Brainstorm output. Treat every claim as a lead, and the model has already told you what to check first.

The two extra lines do the legwork. SHAKIEST CLAIM is your verification queue sorted for you: one claim, google it first. FLIP CONDITION is the search query, already written.

One behavior note. Never punish low scores. If you react badly every time the model says 55, it learns inside that conversation to inflate, and the whole instrument drifts. A 55 that saves you from acting on a bad number is the prompt doing its job. Thank it and go verify.

And the honest limit: the score is self-reported. The calibration research says it correlates with truth, not that it guarantees it. What the prompt reliably kills is the uniform confident voice, which is the thing that gets people burned. For money, health, or legal questions, verify the big claims even at 90.

MAKE IT PERMANENT

The paste dies with the session. Install it where you actually work.

- **ChatGPT.** Settings, then Personalization, then Custom Instructions. Put the full prompt in the "how to respond" box. Every new chat starts armed.
- **Claude.** Drop it into a Project's instructions. Every chat in that project runs truth mode without you asking.
- **Claude Code.** Add it to the CLAUDE.md in your repo, or `~/.claude/CLAUDE.md` to make it global. Now your coding agent flags shaky claims about your own stack.
- **Gemini.** Saved info, or bake it into a Gem.

Once it is installed, "truth mode" becomes a handle. Say "truth check that" to audit any single claim mid-conversation, or "truth check your last answer" to run it retroactively on something that smelled off.

WHEN IT EARNS ITS KEEP

Run it always if you like the texture. But these are the moments it pays for itself:

- **Numbers, quotes, and citations.** The classic hallucination zone. Watch how many arrive labeled RECALLED.
- **Anything after the training cutoff.** The prompt forces the model to admit its data may predate the answer instead of bluffing through.
- **"Is this a good idea?"** The bias check is the star here. You find out whether you are getting analysis or applause.
- **Code that touches production.** A 60 on "this migration is safe" is the cheapest incident review you will ever run.
- **Long sessions.** Instructions get buried as context grows. If the TRUTH CHECK block stops appearing, the rest of the prompt is gone too, which makes it a built-in canary. Restate it or start clean.

Tonight's version: paste the prompt, re-ask the last factual question you acted on this week, and look at the number. If it comes back under 70, you know what to do. Pair it with the humanizer so the answers that survive the truth check also read like you. For more build with AI playbooks, follow opusjake.ai.