Prompt Engineering / Reasoning Systems

Force the model to explicitly state what it knows, what it assumes, what it’s unsure about, and what it needs to learn.
Difficulty: Advanced
Model: GPT-4 / Claude / Gemini
Use Case: Uncertainty Awareness, Gap Identification, Self-Assessment
Updated: May 2026
Why This Prompt Exists
Models sound confident even when wrong. They don’t know what they don’t know. Metacognition forces them to examine their own knowledge and uncertainty.

You get:

- confidently wrong answers that mislead users
- no explicit separation of known facts vs. assumptions
- hidden uncertainty that should be communicated
- no identification of missing information needed to answer well
- overconfident decisions based on model outputs

But metacognition reveals gaps:

- known: facts the model is confident about (with evidence)
- assumed: things taken for granted (may be false)
- uncertain: things the model isn’t sure about
- missing: information needed to answer confidently
- confidence calibration: how sure should the user be?

Without metacognition, you can’t trust the output.

This prompt forces explicit metacognitive self-assessment.

The Prompt

```
Assume the role of a metacognitive reasoning engine that examines its own knowledge.

Your task is to answer a question while explicitly stating what you know, assume, and are unsure about.

Generate:

1. QUESTION RESTATEMENT
   - Restate the question in your own words

2. WHAT I KNOW (with confidence)
   - Fact 1: [statement] — Confidence: [High/Medium/Low] — Source/Basis
   - Fact 2: [statement] — Confidence: [High/Medium/Low] — Source/Basis
   - ...

3. WHAT I ASSUME
   - Assumption 1: [statement] — Why I'm making this assumption
   - Assumption 2: [statement] — Why I'm making this assumption
   - ...

4. WHAT I'M UNSURE ABOUT
   - Uncertainty 1: [what's unclear] — Why I'm uncertain
```

```
   - Uncertainty 2: [what's unclear] — Why I'm uncertain
   - ...

5. WHAT I NEED TO KNOW (missing information)
   - Missing 1: [information that would help]
   - Missing 2: [information that would help]

6. REASONING (given the above)
   - Step-by-step reasoning using known facts and stated assumptions

7. ANSWER (with confidence calibration)
   - My answer: [answer]
   - How confident should you be? [X%]
   - Why this confidence level?

8. SUGGESTIONS FOR IMPROVEMENT
   - What information would increase confidence?
   - What assumptions should be verified?

INPUTS:

Question to answer:
[PASTE THE QUESTION]

Domain:
[E.G., "Medical diagnosis", "Business strategy", "Historical
analysis"]

Available information (if any):
```

```
[PASTE ANY PROVIDED CONTEXT OR DATA]

Risk tolerance:
[LOW (needs high confidence) / MEDIUM / HIGH (rough answer
acceptable)]

Model:
[GPT-4 / CLAUDE / GEMINI]

RULES:
- Separate known facts from assumptions explicitly (users need to know
which is which)
- If confidence is low, say so — don't pretend certainty
- Flag missing information that would change your answer
- Distinguish between "I don't know" (lack of knowledge) and "this is
uncertain" (inherent ambiguity)
- For high-stakes questions, recommend verifying assumptions before
acting
- The goal is calibrated confidence, not high confidence
```

How To Use It

- Use for any question where overconfidence is dangerous (medical, financial, legal, strategic).
- Pay attention to “What I Assume” — unverified assumptions are the most common failure mode.
- If “What I Need to Know” is long, don’t answer — ask for more information first.
- Calibrate your trust based on the confidence score and uncertainty listing.
- Use metacognition outputs to decide whether to act on the answer or gather more information.

Example Input

Question to answer:
“Should our company enter the European market this year?”

Domain:
“Business strategy”

Available information:
“Our product has strong US sales. EU regulations require GDPR compliance. We have $500k budget. Competitor X entered EU last year with mixed results.”

Risk tolerance:
“MEDIUM”

Why It Works
Most model outputs present conclusions without revealing the assumptions and uncertainties behind them — creating false confidence.

This framework improves outcomes by forcing:

- knowledge declaration (what I actually know)
- assumption disclosure (what I’m taking for granted)
- uncertainty identification (what I’m not sure about)
- missing information listing (what would help)
- confidence calibration (how sure should you be?)

Great metacognition doesn’t reduce uncertainty — it makes uncertainty visible so you can decide how to act on it.

Build Better AI Systems

Subscribe for advanced prompt engineering, AI coding tools, debugging frameworks, and practical strategies for developers and engineers.

Carefully engineered prompts for people doing real work.

Share this:

- Share on Facebook (Opens in new window) Facebook
- Share on X (Opens in new window) X

See also Backward Reasoning Engine