Prompt Engineering / Chain-of-Thought

Pull just the reasoning steps from a model's response, separate from the final answer.
Difficulty: Intermediate
Model: GPT-4 / Claude / Gemini
Use Case: Audit, Debugging, Learning from Model Outputs
Updated: May 2026
Why This Prompt Exists
Models produce long responses with reasoning mixed with final answers. You need the trace alone — but extracting it manually is tedious.

You get:

- responses where reasoning and answer are tangled together
- inability to audit the model's thinking without rereading everything
- no easy way to compare reasoning across multiple responses
- difficulty debugging prompts (can't isolate reasoning from output)
- time wasted separating signal from noise

But structured extraction makes reasoning visible:

- isolated steps: each reasoning step as a separate item
- assumptions extracted: what the model took for granted
- decision points: where the model chose between alternatives
- final answer separated: cleanly distinguished from reasoning
- confidence indicators: where the model was uncertain

Without extraction, reasoning is buried.

This prompt extracts clean reasoning traces from model responses.

The Prompt

```
Assume the role of a reasoning auditor who extracts clean traces from
model outputs.

Your task is to separate reasoning steps from final answers in a
model's response.

Generate:

1. ORIGINAL RESPONSE
   - The full model output

2. REASONING TRACE (steps only)
   - Step 1: [first reasoning step]
   - Step 2: [second reasoning step]
   - Step 3: [third reasoning step]
   - (Continue until reasoning complete)

3. ASSUMPTIONS EXTRACTED
   - What the model assumed without proof
   - Which assumptions are justified vs. questionable

4. DECISION POINTS
   - Where the model chose between alternatives
   - What alternatives were considered (and why rejected)

5. UNCERTAINTY INDICATORS
   - Where the model expressed doubt or qualification
```

- Confidence level per step (if detectable)

6. FINAL ANSWER (isolated)
   - The answer only, without surrounding reasoning

7. REASONING QUALITY ASSESSMENT
   - Is the reasoning complete? (No missing steps)
   - Is the reasoning logical? (Steps follow from previous)
   - Are assumptions explicit? (Or hidden?)
   - Overall quality: High / Medium / Low

INPUTS:

Model response (with reasoning and answer mixed):
[PASTE THE FULL RESPONSE]

Task that produced this response (for context):
[E.G., "Math word problem"]

Expected reasoning structure (if any):
[FREE-FORM / STRUCTURED STEPS / TREE / OTHER]

RULES:
- Preserve the model's exact wording for steps (no paraphrasing)
- Flag when steps are missing (the model jumped without explanation)
- Note when the model contradicts itself across steps
- Distinguish between reasoning and explanation (reasoning = how; explanation = why)
- If reasoning is too tangled to extract cleanly, flag as "poorly

`structured"`

How To Use It

- Run this on model outputs when debugging prompt failures — see where reasoning broke.
- Use extracted reasoning traces to train new prompt engineers (here's how the model thinks).
- Compare reasoning traces across model versions to see if reasoning improved.
- Archive reasoning traces for audit trails (important for regulated industries).
- Share extracted final answers with stakeholders (without the reasoning clutter).

Example Input

Model response:
"Let me think about this. The train travels 60 miles per hour for 2 hours, so distance = rate × time = 60 × 2 = 120 miles. Then it stops for 30 minutes. Then it travels 50 miles per hour for 1.5 hours, so another 75 miles. Total distance = 120 + 75 = 195 miles. So the answer is 195 miles."

Task that produced this response:
"Math word problem — calculate total distance"

Why It Works
Most model outputs bury reasoning inside paragraphs — making it hard to audit or debug.

This framework improves outcomes by forcing:

- step extraction (isolated, numbered reasoning steps)
- assumption identification (what the model took for granted)
- decision point capture (where alternatives were considered)
- uncertainty flagging (where the model wasn't confident)

- final answer isolation (clean separation from reasoning)

Great reasoning extraction doesn’t interpret — it surfaces what the model actually did.

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 Minimal Steps Optimizer