Chain complex prompts for stronger performance

Copy page

While these tips apply broadly to all Claude models, you can find prompting tips specific to extended thinking models here.

When working with complex tasks, Claude can sometimes drop the ball if you try to handle everything in a single prompt. Chain of thought (CoT) prompting is great, but what if your task has multiple distinct steps that each require in-depth thought?

Enter prompt chaining: breaking down complex tasks into smaller, manageable subtasks.

Why chain prompts?

1. **Accuracy**: Each subtask gets Claude's full attention, reducing errors.
2. **Clarity**: Simpler subtasks mean clearer instructions and outputs.
3. **Traceability**: Easily pinpoint and fix issues in your prompt chain.

When to chain prompts

Use prompt chaining for multi-step tasks like research synthesis, document analysis, or iterative content creation. When a task involves multiple transformations, citations, or instructions, chaining prevents Claude from dropping or mishandling steps.

Remember: Each link in the chain gets Claude's full attention!

Debugging tip: If Claude misses a step or performs poorly, isolate that step in its own prompt. This lets you fine-tune problematic steps without redoing the entire task.

1. **Identify subtasks**: Break your task into distinct, sequential steps.
2. **Structure with XML for clear handoffs**: Use XML tags to pass outputs between prompts.
3. **Have a single-task goal**: Each subtask should have a single, clear objective.
4. **Iterate**: Refine subtasks based on Claude's performance.

Example chained workflows:

- **Multi-step analysis**: See the legal and business examples below.
- **Content creation pipelines**: Research → Outline → Draft → Edit → Format.
- **Data processing**: Extract → Transform → Analyze → Visualize.
- **Decision-making**: Gather info → List options → Analyze each → Recommend.
- **Verification loops**: Generate content → Review → Refine → Re-review.

Optimization tip: For tasks with independent subtasks (like analyzing multiple docs), create separate prompts and run them in parallel for speed.

Advanced: Self-correction chains

You can chain prompts to have Claude review its own work! This catches errors and refines outputs, especially for high-stakes tasks.

Example: Self-correcting research summary

Examples

Example: Analyzing a legal contract (without chaining)

Example: Analyzing a legal contract (with chaining)

Prompt library

Get inspired by a curated selection of prompts for various tasks and use cases.

GitHub prompting tutorial ↗

An example-filled tutorial that covers the prompt engineering concepts found in our docs.

Google Sheets prompting tutorial ↗

A lighter weight version of our prompt engineering tutorial via an interactive spreadsheet.

Amazon Bedrock

Google Cloud's Vertex AI

Startups program

Company

Anthropic

Careers

Economic Futures

Research

News

Responsible Scaling Policy

Security and compliance

Transparency

Help and security

Availability

Status

Support

Discord

Terms and policies

Privacy policy

Responsible disclosure policy

Terms of service: Commercial

Terms of service: Consumer

Usage policy