

# Prompt engineering overview

ⓘ While these tips apply broadly to all Claude models, you can find prompting tips specific to extended thinking models [here](#).

## Before prompt engineering

This guide assumes that you have:

1. A clear definition of the success criteria for your use case
2. Some ways to empirically test against those criteria
3. A first draft prompt you want to improve

If not, we highly suggest you spend time establishing that first. Check out [Define your success criteria](#) and [Create strong empirical evaluations](#) for tips and guidance.

### Prompt generator

Don't have a first draft prompt? Try the prompt generator in the Anthropic Console!

## When to prompt engineer

Ask AI

# ANTHROPIC

Prompt engineering > **Prompt engineering overview**

## Prompting vs. finetuning

## How to prompt engineer

The prompt engineering pages in this section have been organized from most broadly effective techniques to more specialized techniques. When troubleshooting performance, we suggest you try these techniques in order, although the actual impact of each technique will depend on your use case.

1. [Prompt generator](#)
2. [Be clear and direct](#)
3. [Use examples \(multishot\)](#)
4. [Let Claude think \(chain of thought\)](#)
5. [Use XML tags](#)
6. [Give Claude a role \(system prompts\)](#)
7. [Prefill Claude's response](#)
8. [Chain complex prompts](#)
9. [Long context tips](#)

## Prompt engineering tutorial

If you're an interactive learner, you can dive into our interactive tutorials instead!

# ANTHROPIC

Prompt engineering > **Prompt engineering overview**

the prompt engineering concepts found in our docs.

engineering tutorial via an interactive spreadsheet.

Was this page helpful?

 Yes

 No

◀ Legal summarization

Claude 4 best practices ▶

# Claude 4 prompt engineering best practices

This guide provides specific prompt engineering techniques for Claude 4 models (Opus 4 and Sonnet 4) to help you achieve optimal results in your applications. These models have been trained for more precise instruction following than previous generations of Claude models.

## General principles

### Be explicit with your instructions

Claude 4 models respond well to clear, explicit instructions. Being specific about your desired output can help enhance results. Customers who desire the “above and beyond” behavior from previous Claude models might need to more explicitly request these behaviors with Claude 4.

#### Example: Creating an analytics dashboard

##### Less effective:

Create an analytics dashboard

##### More effective:

Ask AI

# ANTHROPIC

Prompt engineering > **Claude 4 prompt engineering best practices**

## Add context to improve performance

Providing context or motivation behind your instructions, such as explaining to Claude why such behavior is important, can help Claude 4 better understand your goals and deliver more targeted responses.

### Example: Formatting preferences

#### Less effective:

NEVER use ellipses

#### More effective:

Your response will be read aloud by a text-to-speech engine, so never use ellipses.

Claude is smart enough to generalize from the explanation.

## Be vigilant with examples & details

Claude 4 models pay attention to details and examples as part of instruction following. Ensure that your examples align with the behaviors you want to encourage and minimize behaviors you want to avoid.

# ANTHROPIC

## Prompt engineering > Claude 4 prompt engineering best practices

formatting in Claude 4 models:

### 1. Tell Claude what to do instead of what not to do

Instead of: “Do not use markdown in your response”

Try: “Your response should be composed of smoothly flowing prose paragraphs.”

### 2. Use XML format indicators

Try: “Write the prose sections of your response in  
<smoothly\_flowin\_g\_prose\_paragraphs> tags.”

### 3. Match your prompt style to the desired output

The formatting style used in your prompt may influence Claude’s response style. If you are still experiencing steerability issues with output formatting, we recommend as best as you can matching your prompt style to your desired output style. For example, removing markdown from your prompt can reduce the volume of markdown in the output.

## Leverage thinking & interleaved thinking capabilities

Claude 4 offers thinking capabilities that can be especially helpful for tasks involving reflection after tool use or complex multi-step reasoning. You can guide its initial or interleaved thinking for better results.

### Example prompt

After receiving tool results, carefully reflect on their quality and determine op-



For more information on thinking capabilities, see [Extended thinking](#).

# ANTHROPIC

Prompt engineering > **Claude 4 prompt engineering best practices**

effective:

Sample prompt for agents

For maximum efficiency, whenever you need to perform multiple independent operations, consider using temporary files or memory-based structures.

## Reduce file creation in agentic coding

Claude 4 models may sometimes create new files for testing and iteration purposes, particularly when working with code. This approach allows Claude to use files, especially python scripts, as a ‘temporary scratchpad’ before saving its final output. Using temporary files can improve outcomes particularly for agentic coding use cases.

If you’d prefer to minimize net new file creation, you can instruct Claude to clean up after itself:

Sample prompt

If you create any temporary new files, scripts, or helper files for iteration, consider deleting them once they’re no longer needed.

## Enhance visual and frontend code generation

For frontend code generation, you can steer Claude 4 models to create complex, detailed, and interactive designs by providing explicit encouragement:

Sample prompt

Don’t hold back. Give it your all.

# ANTHROPIC

Prompt engineering > **Claude 4 prompt engineering best practices**

Add thoughtful details like hover states, transitions, and micro-interactions

“Create an impressive demonstration showcasing web development capabilities”

“Apply design principles: hierarchy, contrast, balance, and movement”

## Avoid focusing on passing tests and hard-coding

Frontier language models can sometimes focus too heavily on making tests pass at the expense of more general solutions. To prevent this behavior and ensure robust, generalizable solutions:

### Sample prompt

Please write a high quality, general purpose solution. Implement a solution that :

Focus on understanding the problem requirements and implementing the correct algo

If the task is unreasonable or infeasible, or if any of the tests are incorrect, |

## Migration considerations

When migrating from Sonnet 3.7 to Claude 4:

- 1. Be specific about desired behavior:** Consider describing exactly what you'd like to see in the output.
- 2. Frame your instructions with modifiers:** Adding modifiers that encourage Claude to increase the quality and detail of its output can help better shape Claude's performance. For example, instead of “Create an analytics dashboard”, use “Create an analytics dashboard. Include as many relevant features and interactions as possible. Go beyond the basics to create a fully-featured implementation.”

# ANTHROPIC

Prompt engineering > Claude 4 prompt engineering best practices

Was this page helpful?

 Yes

 No

< Overview

Prompt generator >

# Use prompt templates and variables

When deploying an LLM-based application with Claude, your API calls will typically consist of two types of content:

**Fixed content:** Static instructions or context that remain constant across multiple interactions

**Variable content:** Dynamic elements that change with each request or conversation, such as:

User inputs

Retrieved content for Retrieval-Augmented Generation (RAG)

Conversation context such as user account history

System-generated data such as tool use results fed in from other independent calls to Claude

A **prompt template** combines these fixed and variable parts, using placeholders for the dynamic content. In the [Anthropic Console](#), these placeholders are denoted with `{double brackets}`, making them easily identifiable and allowing for quick testing of different values.

## When to use prompt templates and variables

You should always use prompt templates and variables when you expect any part of your prompt to be repeated in another call to Claude (only via the API or the [Anthropic Console](#)). [claude.ai](#) currently does not support prompt templates or variables).

Ask AI



Prompt templates offer several benefits:

## ANTHROPIC

**Consistency:** Ensure a consistent structure for your prompts across multiple interactions

Prompt engineering → Use prompt templates and variables

**Testability:** Quickly test different inputs and edge cases by changing only the variable portion

**Scalability:** Simplify prompt management as your application grows in complexity

**Version control:** Easily track changes to your prompt structure over time by keeping tabs only on the core part of your prompt, separate from dynamic inputs

The [Anthropic Console](#) heavily uses prompt templates and variables in order to support features and tooling for all the above, such as with the:

**Prompt generator:** Decides what variables your prompt needs and includes them in the template it outputs

**Prompt improver:** Takes your existing template, including all variables, and maintains them in the improved template it outputs

**Evaluation tool:** Allows you to easily test, scale, and track versions of your prompts by separating the variable and fixed portions of your prompt template

---

## Example prompt template

Let's consider a simple application that translates English text to Spanish. The translated text would be variable since you would expect this text to change between users or calls to Claude. This translated text could be dynamically retrieved from databases or the user's input.

Thus, for your translation app, you might use this simple prompt template:

```
Translate this text from English to Spanish: {{text}}
```

# ANTHROPIC

## Next steps

Prompt engineering > **Use prompt templates and variables**

### Generate a prompt

Learn about the prompt generator in the Anthropic Console and try your hand at getting Claude to generate a prompt for you.

### Apply XML tags

If you want to level up your prompt variable game, wrap them in XML tags.

### Anthropic Console

Check out the myriad prompt development tools available in the Anthropic Console.

Was this page helpful?

 Yes

 No

< Prompt generator

Prompt improver >

## PROMPT ENGINEERING

 Copy page

# Use our prompt improver to optimize your prompts

- ➊ Our prompt improver is compatible with all Claude models, including those with extended thinking capabilities. For prompting tips specific to extended thinking models, see [here](#).

The prompt improver helps you quickly iterate and improve your prompts through automated analysis and enhancement. It excels at making prompts more robust for complex tasks that require high accuracy.

USER

From the following list of Wikipedia article titles, identify which article this sentence came from.  
Respond with just the article title and nothing else.

Article titles:  
{{titles}}

Sentence to classify:  
{{sentence}}

Pre-fill response    Add message pair    Improve prompt Beta    Use our automatic prompt improver



## Before you begin

You'll need:

A [prompt template](#) to improve

# ANTHROPIC

Prompt engineering > **Use our prompt improver to optimize your prompts**

## How the prompt improver works

The prompt improver enhances your prompts in 4 steps:

1. **Example identification:** Locates and extracts examples from your prompt template
2. **Initial draft:** Creates a structured template with clear sections and XML tags
3. **Chain of thought refinement:** Adds and refines detailed reasoning instructions
4. **Example enhancement:** Updates examples to demonstrate the new reasoning process

You can watch these steps happen in real-time in the improvement modal.

# ANTHROPIC

## Prompt engineering > Use our prompt improver to optimize your prompts

### 1. Intent and Deployment Circumstance:

The prompt's intent is to classify a given sentence by identifying which Wikipedia article title it most likely belongs to. This task is likely to be deployed in a context where automatic content classification or information retrieval is needed, such as in a search engine, content management system, or knowledge base.

### 2. Flowchart:

Due to the simplicity of this task, a detailed flowchart isn't necessary. However, here's a basic representation of the process:

```
```mermaid
graph TD
    A[Start] --> B[Receive list of article titles]
    B --> C[Receive sentence to classify]
    C --> D[Compare sentence to each article title]
    D --> E[Identify most likely article title]
    E --> F[Output selected article title]
    F --> G[End]
````
```

### 3. Lessons from Examples:

[Cancel](#)

[Open in Workbench](#)

## What you get

The prompt improver generates templates with:

Detailed chain-of-thought instructions that guide Claude's reasoning process and typically improve its performance

Clear organization using XML tags to separate different components

Standardized example formatting that demonstrates step-by-step reasoning from input to output

Strategic prefills that guide Claude's initial responses

# ANTHROPIC

Prompt engineering > **Use our prompt improver to optimize your prompts**

## How to use the prompt improver

1. Submit your prompt template
2. Add any feedback about issues with Claude's current outputs (e.g., "summaries are too basic for expert audiences")
3. Include example inputs and ideal outputs
4. Review the improved prompt

## Generate test examples

Don't have examples yet? Use our [Test Case Generator](#) to:

1. Generate sample inputs
2. Get Claude's responses
3. Edit the responses to match your ideal outputs
4. Add the polished examples to your prompt

## When to use the prompt improver

The prompt improver works best for:

Complex tasks requiring detailed reasoning

Situations where accuracy is more important than speed

Problems where Claude's current outputs need significant improvement

# ANTHROPIC

Prompt engineering > **Use our prompt improver to optimize your prompts**

## Example improvement

Here's how the prompt improver enhances a basic classification prompt:

**Original prompt**

**Improved prompt**

Notice how the improved prompt:

- Adds clear step-by-step reasoning instructions
- Uses XML tags to organize content
- Provides explicit output formatting requirements
- Guides Claude through the analysis process

## Troubleshooting

Common issues and solutions:

**Examples not appearing in output:** Check that examples are properly formatted with XML tags and appear at the start of the first user message

**Chain of thought too verbose:** Add specific instructions about desired output length and level of detail

**Reasoning steps don't match your needs:** Modify the steps section to match your specific use case

# ANTHROPIC

Prompt engineering > **Use our prompt improver to optimize your prompts**

## Prompt library

Get inspired by example prompts for various tasks.

## GitHub prompting tutorial

Learn prompting best practices with our interactive tutorial.

## Test your prompts

Use our evaluation tool to test your improved prompts.

Was this page helpful?

 Yes

 No

◀ Use prompt templates

Be clear and direct ▶

## Be clear, direct, and detailed

💡 While these tips apply broadly to all Claude models, you can find prompting tips specific to extended thinking models [here](#).

When interacting with Claude, think of it as a brilliant but very new employee (with amnesia) who needs explicit instructions. Like any new employee, Claude does not have context on your norms, styles, guidelines, or preferred ways of working. The more precisely you explain what you want, the better Claude's response will be.

💡 **The golden rule of clear prompting**

Show your prompt to a colleague, ideally someone who has minimal context on the task, and ask them to follow the instructions. If they're confused, Claude will likely be too.

## How to be clear, contextual, and specific

**Give Claude contextual information:** Just like you might be able to better perform on a task if you knew more context, Claude will perform better if it has more contextual information. Some examples of contextual information:

What the task results will be used for

What audience the output is meant for

What workflow the task is a part of, and where this task belongs in that workflow

The end goal of the task, or what a successful task completion looks like

Ask AI

**Be specific about what you want Claude to do:** For example, if you want Claude to output only code and nothing else, say so.

**Provide instructions as sequential steps:** Use numbered lists or bullet points to better Prompt engineering → **Be clear, direct, and detailed** ensure that Claude carries out the task the exact way you want it to.

## Examples

### Example: Anonymizing customer feedback

Notice that Claude still makes mistakes in the unclear prompting example, such as leaving in a customer's name.

| Role                          | Unclear Prompt  | Clear Prompt   |
|-------------------------------|---|--|
| User                          | Please remove all personally identifiable information from these customer feedback messages: {{FEEDBACK_DATA}}  | <p>Your task is to anonymize customer feedback for our quarterly review.</p> <p>Instructions:</p> <ol style="list-style-type: none"> <li>Replace all customer names with "CUSTOMER_[ID]" (e.g., "Jane Doe" → "CUSTOMER_001").</li> <li>Replace email addresses with "EMAIL_[ID]@example.com".</li> <li>Redact phone numbers as "PHONE_[ID]".</li> <li>If a message mentions a specific product (e.g., "AcmeCloud"), leave it intact.</li> <li>If no PII is found, copy the message verbatim.</li> <li>Output only the processed messages, separated by "---".</li> </ol> <p>Data to process: {{FEEDBACK_DATA}}</p> |
| Assistant (Claude's response) | <p>Here are the anonymized customer feedback messages:</p> <ol style="list-style-type: none"> <li>I've been using your software for years, and it's been great. But lately, AcmeCloud has been crashing a lot. Fix it!</li> <li>XXX, your customer support rep, was incredibly helpful. He guided me through the whole process. My email is <u>EMAIL_002@example.com</u> if you want to send a survey.</li> </ol> | <p>CUSTOMER_001: I've been using your software for years, and it's been great. But lately, AcmeCloud has been crashing a lot. Fix it!</p> <p>---</p> <p>CUSTOMER_002: CUSTOMER_SERVICE_001, your customer support rep, was incredibly helpful. He guided me through the whole process. My email is <u>EMAIL_002@example.com</u> if you want to send a survey.</p>  |

| Role             | Unclear Prompt  | Clear Prompt   |
|------------------|---|--|
| <b>ANTHROPIC</b> | <p>Want to send a survey.</p> <p>3. I'm John Smith, and I'm not happy. I've been on hold for 2 hours trying to reach someone at PHONE_001. This is XXX. This is unacceptable!</p> | <p>---</p> <p>CUSTOMER_003: I'm CUSTOMER_003, and I'm not happy. I've been on hold for 2 hours trying to reach someone at PHONE_001. This is unacceptable!</p> |

## Example: Crafting a marketing email campaign

Notice that Claude makes up details to fill in the gaps where it lacks context with the vague prompt.

| Role                          | Vague Prompt   | Specific Prompt   |
|-------------------------------|--|---|
| User                          | <p>Write a marketing email for our new AcmeCloud features.</p>                         | <p>Your task is to craft a targeted marketing email for our Q3 AcmeCloud feature release.</p> <p>Instructions:</p> <ol style="list-style-type: none"> <li>1. Write for this target audience: Mid-size tech companies (100-500 employees) upgrading from on-prem to cloud.</li> <li>2. Highlight 3 key new features: advanced data encryption, cross-platform sync, and real-time collaboration.</li> <li>3. Tone: Professional yet approachable. Emphasize security, efficiency, and teamwork.</li> <li>4. Include a clear CTA: Free 30-day trial with priority onboarding.</li> <li>5. Subject line: Under 50 chars, mention “security” and “collaboration”.</li> <li>6. Personalization: Use {{COMPANY_NAME}} and {{CONTACT_NAME}} variables.</li> </ol> <p>Structure:</p> <ol style="list-style-type: none"> <li>1. Subject line</li> <li>2. Email body (150-200 words)</li> <li>3. CTA button text</li> </ol> |
| Assistant (Claude's response) | <p>Here's a draft marketing email for our new features:</p> <p>Subject: Boost Your</p> | <p>Subject line: AcmeCloud: Enterprise security meets real-time teamwork</p> <p>Email body:</p> <p>Dear {{CONTACT_NAME}},</p>   |

| Role   | Vague Prompt  | Specific Prompt  |
|--|---|--|
| <b>ANTHROPIC</b>   | Business with AcmeCloud's Latest Features   | At {{COMPANY_NAME}}, you're scaling fast and eyeing the cloud. But with growth comes the challenge of keeping data secure while fostering collaboration.                                 |
| Prompt engineering > <b>Be clear, direct, and detailed</b> | Dear Valued Customer,   | That's where AcmeCloud's Q3 release comes in, tailor-made for tech innovators like you.  |
|  | We're excited to introduce three powerful features to AcmeCloud that will revolutionize your workflow:          |  |
|  | 1. Real-Time Collaboration: Work on documents simultaneously with your team. No more version control issues!    | 1. Fort Knox for Your Files: Military-grade encryption now safeguards your data, both in motion and at rest. Your intellectual property stays yours, period.                             |
|  | 2. AI-Powered Insights: Make data-driven decisions with AI that analyzes trends and automates tasks.            | 2. Sync Without Borders: Windows, Mac, or Linux—AcmeCloud speaks all languages. Unified version control means no more "final_v2_REALLY_FINAL.doc".                                       |
|  | 3. Advanced Security Suite: End-to-end encryption, multi-factor auth, and threat detection keep your data safe. | 3. Collaborate at Light Speed: Ditch the attachment ping-pong. Now your devs, designers, and data scientists co-create in real-time, slashing time-to-market.                            |
|  | Limited Offer: Upgrade now for a 20% first-year discount. Plus, refer a friend for an extra 10% off!            | Don't just take our word for it. Experience the AcmeCloud difference with our 30-day trial. And because we know time is your most precious asset, we're throwing in priority onboarding. |
|  | Best,<br>The AcmeCloud Team   | Elevate Your Team Now - Free 30-Day Trial  |

## Example: Incident response

Notice that Claude outputs superfluous text and different formating with the vague prompt.

| Role   | Vague Prompt  | Detailed Prompt  |
|--|---|--|
| <b>ANTHROPIC</b><br>User   | Analyze this AcmeCloud outage report and summarize the key points.  | Analyze this AcmeCloud outage report. Skip the preamble. Keep your response terse and write only the bare bones necessary information.<br><br>List only:<br>1) Cause<br>2) Duration<br>3) Impacted services<br>4) Number of affected users<br>5) Estimated revenue loss. |
| Prompt engineering > <b>Be clear, direct, and detailed</b><br>{REPORT} |   | Here's the report:<br>{REPORT}   |
| Assistant<br>(Claude's response)                                       | <p>Here's a summary of the key points from the AcmeCloud outage report:</p> <p>On May 15th, a misconfigured load balancer update caused significant traffic routing issues, resulting in a service outage from 09:15 to 13:52 UTC. The 4-hour 37-minute disruption impacted our core file syncing service and admin dashboard, rendering them inaccessible. Additionally, the mobile app experienced partial functionality loss. Approximately 68% of our user base, totaling 1.3 million users, were affected. Based on our Service Level Agreements (SLAs) and average revenue per user, we estimate a financial impact of \$420,000 in service credits and potential customer churn.</p> | 1) Cause: Misconfigured load balancer update<br>2) Duration: 4h 37m (09:15-13:52 UTC, May 15)<br>3) Impacted: Core sync, admin dashboard (down); mobile app (partial)<br>4) Affected users: 1.3M (68% of base)<br>5) Est. revenue loss: \$420,000                        |

**Prompt library**  
Get inspired by a curated selection of

**GitHub prompting tutorial**

**Google Sheets prompting tutorial**

prompts for various  
**ANTHROPIC** tasks and use cases.

Prompt engineering > **Be clear, direct, and detailed** in our docs.

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

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

Was this page helpful?

 Yes

 No

< Prompt improver

Use examples (multishot prompting) >

## PROMPT ENGINEERING

 Copy page

# Use examples (multishot prompting) to guide Claude's behavior

- 💡 While these tips apply broadly to all Claude models, you can find prompting tips specific to extended thinking models [here](#).

Examples are your secret weapon shortcut for getting Claude to generate exactly what you need. By providing a few well-crafted examples in your prompt, you can dramatically improve the accuracy, consistency, and quality of Claude's outputs. This technique, known as few-shot or multishot prompting, is particularly effective for tasks that require structured outputs or adherence to specific formats.

- 💡 Power up your prompts: Include 3-5 diverse, relevant examples to show Claude exactly what you want. More examples = better performance, especially for complex tasks.

## Why use examples?

**Accuracy:** Examples reduce misinterpretation of instructions.

**Consistency:** Examples enforce uniform structure and style.

**Performance:** Well-chosen examples boost Claude's ability to handle complex tasks.

## Crafting effective examples

For maximum effectiveness, make sure that your examples are:

 Ask AI

**Relevant:** Your examples mirror your actual use case.

## ANTHROPIC

**Diverse:** Your examples cover edge cases and potential challenges, and vary enough that

Claude doesn't inadvertently pick up on unintended patterns.

Prompt engineering > **Use examples (multishot prompting) to guide Claude's behavior**

**Clear:** Your examples are wrapped in `<example>` tags (if multiple, nested within `<examples>` tags) for structure.

- 💡 Ask Claude to evaluate your examples for relevance, diversity, or clarity. Or have Claude generate more examples based on your initial set.

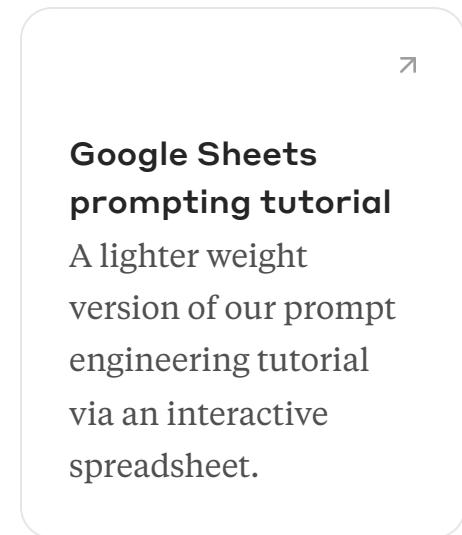
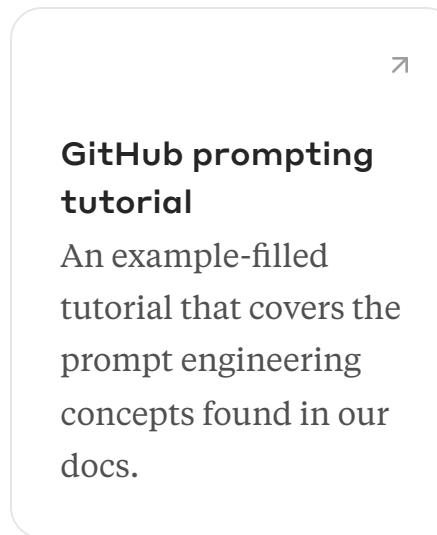
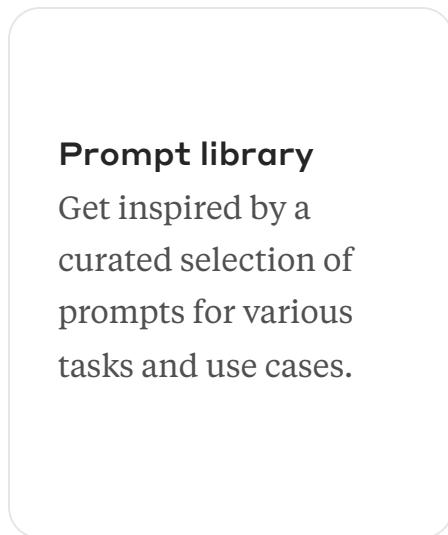
### Example: Analyzing customer feedback

Notice that in the “no examples” version, Claude does not list multiple options for each category, despite its written analysis indicating that there are multiple categories for certain feedback. It also includes a long explanation when we may not want it to.

| Role                          | No Examples  | With Examples   |
|-------------------------------|--|---|
| User                          | <p>Analyze this customer feedback and categorize the issues. Use these categories: UI/UX, Performance, Feature Request, Integration, Pricing, and Other. Also rate the sentiment (Positive/Neutral/Negative) and priority (High/Medium/Low).</p> <p>Here is the feedback: {{FEEDBACK}}</p> | <p>Our CS team is overwhelmed with unstructured feedback. Your task is to analyze feedback and categorize issues for our product and engineering teams. Use these categories: UI/UX, Performance, Feature Request, Integration, Pricing, and Other. Also rate the sentiment (Positive/Neutral/Negative) and priority (High/Medium/Low). Here is an example:</p> <pre>&lt;example&gt; Input: The new dashboard is a mess! It takes forever to load, and I can't find the export button. Fix this ASAP! Category: UI/UX, Performance Sentiment: Negative Priority: High&lt;/example&gt;</pre> <p>Now, analyze this feedback: {{FEEDBACK}}</p> |
| Assistant (Claude's response) | <p>Here's the analysis of the customer feedback:</p> <p>Issue: UI/UX</p>   | <p>Here's the analysis of the customer feedback:</p> <p>1. The new dashboard is a mess! It takes</p>  |

| Role               | No Examples   | With Examples  |
|--------------------|---|--|
| <b>ANTHROPIC</b>   |   |  |
| Prompt engineering | <p>Sentiment: Negative<br/>Priority: High<br/>The customer is frustrated with the new dashboard design, which they find cluttered and slow to load. They specifically call out the missing “export” button, which is likely a critical feature. This issue needs to be addressed as a high priority.</p> <p>Issue: Integration<br/>Sentiment: Positive<br/>Priority: Medium<br/>The customer is pleased with the existing Salesforce integration but would like to see support for Hubspot as well. This is a feature request that could be considered for a future product update.</p> <p>Issue: Pricing<br/>Sentiment: Negative<br/>Priority: High<br/>The customer finds the pricing to be unreasonable, especially for the basic features. They are threatening to switch to a competitor, so this is a high-priority issue that needs to be reviewed.</p> <p>Issue: Performance<br/>Sentiment Negative<br/>Priority: High<br/>The customer is experiencing critical issues with the mobile app, where large file uploads consistently result in crashes. This is a significant problem that impacts the customer’s ability to use the product effectively and should be addressed as a high priority.</p> <p>Issue: UI/UX<br/>Sentiment: Positive<br/>Priority: Low<br/>The customer is overall satisfied with the product, specifically calling out the intuitive UI and excellent customer</p> | <p>forever to load, and I can't find the export button. Fix this ASAP!</p> <p>Category: UI/UX, Performance<br/>Sentiment: Negative<br/>Priority: High</p> <p>2. Love the Salesforce integration! But it'd be great if you could add Hubspot too.<br/>Category: Integration, Feature Request<br/>Sentiment: Positive<br/>Priority: Medium</p> <p>3. Your pricing is ridiculous! \$99/mo for basic features? I'm switching to Competitor X.<br/>Category: Pricing<br/>Sentiment: Negative<br/>Priority: High</p> <p>4. The mobile app keeps crashing when I try to upload large files (&gt;50MB). This is critical for our field agents.<br/>Category: Performance<br/>Sentiment: Negative<br/>Priority: High</p> <p>5. Overall, I'm happy with the product. The UI is intuitive, and the customer support is top-notch. Keep it up!<br/>Category: UI/UX, Other (Customer Support)<br/>Sentiment: Positive<br/>Priority: Low</p> |

| Role             | No Examples   | With Examples   |
|------------------|---|---|
| <b>ANTHROPIC</b> | support. This is a positive piece of feedback that does not require any immediate action. | Prompt engineering → <b>Use examples (multishot prompting) to guide Claude's behavior</b> |



Was this page helpful?

Yes

No

◀ Be clear and direct

Let Claude think (CoT) ➔

# Let Claude think (chain of thought prompting) to increase performance

- 💡 While these tips apply broadly to all Claude models, you can find prompting tips specific to extended thinking models [here](#).

When faced with complex tasks like research, analysis, or problem-solving, giving Claude space to think can dramatically improve its performance. This technique, known as chain of thought (CoT) prompting, encourages Claude to break down problems step-by-step, leading to more accurate and nuanced outputs.

## Before implementing CoT

### Why let Claude think?

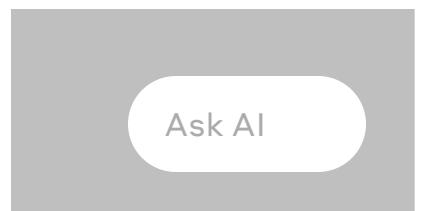
**Accuracy:** Stepping through problems reduces errors, especially in math, logic, analysis, or generally complex tasks.

**Coherence:** Structured thinking leads to more cohesive, well-organized responses.

**Debugging:** Seeing Claude's thought process helps you pinpoint where prompts may be unclear.

### Why not let Claude think?

Increased output length may impact latency.

A large grey rectangular button with rounded corners and a white oval cutout containing the text "Ask AI".

# ANTHROPIC

Prompt engineering > **Let Claude think (chain of thought prompting) to increase performance**  
analysis, writing complex documents, or decisions with many factors.

## How to prompt for thinking

The chain of thought techniques below are **ordered from least to most complex**. Less complex methods take up less space in the context window, but are also generally less powerful.

 **CoT tip:** Always have Claude output its thinking. Without outputting its thought process, no thinking occurs!

**Basic prompt:** Include “Think step-by-step” in your prompt.

Lacks guidance on *how* to think (which is especially not ideal if a task is very specific to your app, use case, or organization)

**Example:** Writing donor emails (basic CoT)

# ANTHROPIC

Prompt engineering > **Let Claude think (chain of thought prompting) to increase performance**

Program information:

```
<program>{{PROGRAM_DETAILS}}  
</program>
```

Donor information:

```
<donor>{{DONOR_DETAILS}}  
</donor>
```

Think step-by-step before you write the email.

**Guided prompt:** Outline specific steps for Claude to follow in its thinking process.

Lacks structuring to make it easy to strip out and separate the answer from the thinking.

**Example: Writing donor emails (guided CoT)**

# ANTHROPIC

Prompt engineering > **Let Claude think (chain of thought prompting) to increase performance**

Program information:

```
<program>{{PROGRAM_DETAILS}}  
</program>
```

Donor information:

```
<donor>{{DONOR_DETAILS}}  
</donor>
```

Think before you write the email. First, think through what messaging might appeal to this donor given their donation history and which campaigns they've supported in the past. Then, think through what aspects of the Care for Kids program would appeal to them, given their history. Finally, write the personalized donor email using your analysis.

**Structured prompt:** Use XML tags like `<thinking>` and `<answer>` to separate reasoning from the final answer.

**Example: Writing donor emails (structured guided CoT)**

# Use XML tags to structure your prompts

💡 While these tips apply broadly to all Claude models, you can find prompting tips specific to extended thinking models [here](#).

When your prompts involve multiple components like context, instructions, and examples, XML tags can be a game-changer. They help Claude parse your prompts more accurately, leading to higher-quality outputs.

💡 **XML tip:** Use tags like `<instructions>`, `<example>`, and `<formatting>` to clearly separate different parts of your prompt. This prevents Claude from mixing up instructions with examples or context.

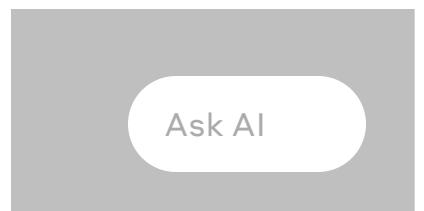
## Why use XML tags?

**Clarity:** Clearly separate different parts of your prompt and ensure your prompt is well structured.

**Accuracy:** Reduce errors caused by Claude misinterpreting parts of your prompt.

**Flexibility:** Easily find, add, remove, or modify parts of your prompt without rewriting everything.

**Parseability:** Having Claude use XML tags in its output makes it easier to extract specific parts of its response by post-processing.

A large grey rectangular button with a white rounded rectangle in the center containing the text "Ask AI".

# ANTHROPIC

Prompt engineering > **Use XML tags to structure your prompts**

## Tagging best practices

- Be consistent:** Use the same tag names throughout your prompts, and refer to those tag names when talking about the content (e.g, `Using the contract in <contract> tags...` ).
- Nest tags:** You should nest tags `<outer><inner></inner></outer>` for hierarchical content.



**Power user tip:** Combine XML tags with other techniques like multishot prompting (`<examples>`) or chain of thought (`<thinking>`, `<answer>`). This creates super-structured, high-performance prompts.

## Examples

### Example: Generating financial reports

Without XML tags, Claude misunderstands the task and generates a report that doesn't match the required structure or tone. After substitution, there is also a chance that Claude misunderstands where one section (like the Q1 report example) stops and another begins.

| Role | No XML Tags  | With XML Tags   |
|------|--|---|
| User | You're a financial analyst at AcmeCorp. Generate a Q2 financial report for our investors. Include sections on Revenue Growth, Profit Margins, and Cash Flow, like with this example from last year: {{Q1_REPORT}}. Use data points from this spreadsheet: {{SPREADSHEET_DATA}}. The report should be extremely concise, to the point, professional, and in | You're a financial analyst at AcmeCorp. Generate a Q2 financial report for our investors. AcmeCorp is a B2B SaaS company. Our investors value |

# ANTHROPIC

## Prompt engineering > Use XML tags to structure your prompts

|   |  |
|---|--|
| <p>Assistant Q2 Financial Report for AcmeCorp Investors</p> <p>Dear Valued Investors,</p> <p>I am pleased to present our Q2 financial report, showcasing AcmeCorp's continued growth and resilience in the B2B SaaS market.</p> <p><b>1. Revenue Growth:</b><br/>Our Q2 revenue reached \$15.2M, a 22% YoY increase. This growth is primarily driven by a 30% expansion in our enterprise client base. However, our SMB segment saw a slight decline of 5%, which we aim to address through targeted marketing initiatives in Q3.</p> <p><b>2. Profit Margins:</b><br/>Gross margin improved to 72% (up 3% from Q1) due to infrastructure optimizations. EBITDA margin, however, dipped to 18% (down 2%) as we invested heavily in R&amp;D for our upcoming AI-driven features.</p> <p><b>3. Cash Flow:</b><br/>Operating cash flow remains strong at \$4.1M. Our cash reserves stand at \$28M, providing a 15-month runway</p> | <p>{[SPREADSHEET_DATA]}</p> <p>&lt;/data&gt;</p> <p>&lt;instructions&gt;</p> <ol style="list-style-type: none"><li>1. Include sections: Revenue Growth, Profit Margins, Cash Flow.</li><li>2. Highlight strengths and areas for improvement.</li></ol> <p>&lt;/instructions&gt;</p> <p>Make your tone concise and professional. Follow this structure:</p> <p>&lt;formatting_example&gt;</p> <p>{[Q1_REPORT]}</p> <p>&lt;/formatting_example&gt;</p> <p>Q2 Financial Report for AcmeCorp Investors&lt;/output&gt;</p> <p>Revenue Growth:<br/>- Q2 revenue: \$15.2M (22% YoY ↑)<br/>- Enterprise clients: 30% ↑<br/>- SMB segment: 5% ↓<br/>Action: Targeted Q3 marketing</p> <p>Profit Margins:<br/>- Gross margin: 72% (3% ↑ from Q1)<br/>Reason: Infrastructure optimizations<br/>- EBITDA margin: 18% (2% ↓)<br/>Reason: R&amp;D investment in AI features</p> <p>Cash Flow:<br/>- Operating cash flow: \$4.1M (strong)<br/>- Cash reserves: \$28M (15-</p> |
|---|--|

# ANTHROPIC

## Prompt engineering > Use XML tags to structure your prompts

| Summary, financials remain robust. We're balancing growth investments with prudent cash management. Our focus for Q3 will be revitalizing SMB growth and optimizing sales efficiency. | Outlook:<br>- Core financials: Robust<br>- Q3 focus: Revitalize SMB, optimize sales     |
|---|---|
| Thank you for your continued trust.   | Management remains committed to balancing growth investments with financial discipline. |

### Example: Legal contract analysis

Without XML tags, Claude's analysis is disorganized and misses key points. With tags, it provides a structured, thorough analysis that a legal team can act on.

| Role | No XML Tags   | With XML Tags  |
|------|---|--|
| User | Analyze this software licensing agreement for potential risks and liabilities: {{CONTRACT}}. Focus on indemnification, limitation of liability, and IP ownership clauses. Also, note any unusual or concerning terms. Here's our standard contract for reference: {{STANDARD_CONTRACT}}. Give a summary of findings and recommendations for our legal team. | <p>Analyze this software licensing agreement for legal risks and liabilities.</p> <p>We're a multinational enterprise considering this agreement for our core data infrastructure.</p> <pre>&lt;agreement&gt; {{CONTRACT}}&lt;/agreement&gt;</pre> <p>This is our standard contract for reference:</p> <pre>&lt;standard_contract&gt; {{STANDARD_CONTRACT}} &lt;/standard_contract&gt;</pre> <pre>&lt;instructions&gt; 1. Analyze these clauses:</pre> |

# ANTHROPIC

## Prompt engineering > Use XML tags to structure your prompts

|           |  |  |
|-----------|--|--|
|           |  | <p>2. Note unusual or concerning terms.</p> <p>3. Compare to our standard contract.</p> <p>4. Summarize findings in &lt;findings&gt; tags.</p> <p>5. List actionable recommendations in &lt;recommendations&gt; tags.</p> <p>&lt;/instructions&gt;</p>   |
| Assistant | <p>The provided software licensing agreement presents several concerning aspects. The indemnification clause is overly broad, potentially exposing us to significant liabilities. The limitation of liability section caps damages at a mere \$500, which is inadequate for our enterprise-level usage. Regarding IP, the agreement grants the vendor joint ownership of any modifications or integrations we create.</p> <p>In summary, this agreement is not in our best interest. It exposes us to undue risk, limits our recourse in case of failures, and compromises our intellectual property rights. I recommend renegotiating these terms or seeking alternative vendors.</p> | <p>&lt;findings&gt;</p> <p>1. Indemnification (Clause 8):</p> <ul style="list-style-type: none"><li>- Issue: Overly broad. We indemnify vendor against “any and all claims” arising from our use.</li><li>- Risk: Could hold us liable even for vendor’s negligence or product defects.</li><li>- Deviation: Our standard limits indemnity to claims arising from our violation of agreement.</li></ul> <p>2. Limitation of Liability (Clause 9):</p> <ul style="list-style-type: none"><li>- Issue: Caps all damages at \$500, including direct damages.</li><li>- Risk: Grossly inadequate for enterprise use. A critical failure could cost millions.</li><li>- Deviation: Our standard allows for direct damages up to 12 months of fees paid.</li></ul> <p>3. IP Ownership (Clause 12):</p> <ul style="list-style-type: none"><li>- Issue: Grants vendor joint ownership of our modifications/integrations.</li></ul> |

# ANTHROPIC

## Prompt engineering > Use XML tags to structure your prompts

return full ownership of our work.

### 4. Unusual Terms:

- Clause 5.3: Vendor can use our usage data for “any purpose.”
- Clause 7.2: Vendor can terminate for “convenience” with only 7 days’ notice.
- Clause 11: No warranties of any kind, even for basic merchantability.

</findings>

### <recommendations>

1. Reject this agreement. Risks far outweigh benefits for an enterprise.

#### 2. Counter-propose:

- Indemnification: Limit to claims arising from our violation of agreement.
- Liability: Cap at 12 months’ fees for direct damages; negotiate for indirect.
- IP: Full ownership of our modifications/integrations.
- Data Use: Restrict to service improvement only.
- Termination: Minimum 90 days’ notice; only for cause.
- Warranties: Demand basic merchantability and fitness for purpose.

3. If vendor refuses, explore alternative providers or in-house solutions.

4. Engage legal counsel for contract negotiation given high stakes.

</recommendations>

# ANTHROPIC

Prompt engineering > **Use XML tags to structure your prompts**

## 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.

Was this page helpful?

 Yes

 No

◀ Let Claude think (CoT)

Give Claude a role (system prompts) ▶

# ANTHROPIC

Prompt engineering > **Let Claude think (chain of thought prompting) to increase performance**

Program information:

```
<program>{{PROGRAM_DETAILS}}  
</program>
```

Donor information:

```
<donor>{{DONOR_DETAILS}}  
</donor>
```

Think before you write the email in `<thinking>` tags. First, think through what messaging might appeal to this donor given their donation history and which campaigns they've supported in the past. Then, think through what aspects of the Care for Kids program would appeal to them, given their history. Finally, write the personalized donor email in `<email>` tags, using your analysis.

## Examples

### Example: Financial analysis without thinking

Without step-by-step thinking, Claude gives a recommendation that seems reasonable but lacks depth. It doesn't quantify the potential outcomes or consider historical market performance, which are crucial for making an informed decision.

| Role      | Content  |
|-----------|--|
| User      | You're a financial advisor. A client wants to invest \$10,000. They can choose between two options: A) A stock that historically returns 12% annually but is volatile, or B) A bond that guarantees 6% annually. The client needs the money in 5 years for a down payment on a house. Which option do you recommend?   |
| Assistant | Based on the information provided, I would recommend option B, the bond that guarantees a 6% annual return. Here's why: <ol style="list-style-type: none"><li>1. Certainty: The bond offers a guaranteed 6% annual return, which provides certainty in financial planning. This is crucial when the client has a specific financial goal (down payment) with a fixed timeline (5 years).</li></ol> |

# ANTHROPIC

Prompt engineering > **Let Claude think (chain of thought prompting) to increase performance**

~~Market downturns. Longer investment horizons (10+ years) are generally better suited for weathering stock market volatility.~~

While the stock's historical 12% return is attractive, past performance doesn't guarantee future results. The guaranteed 6% from the bond ensures the client will have the funds they need when it's time to make the down payment.

## Example: Financial analysis with thinking

With step-by-step thinking, the advisor provides a much richer analysis. It calculates exact figures for both scenarios, considers historical market volatility, and deeply analyzes the client's risk tolerance. This thorough reasoning leads to a more confident and justifiable recommendation.

| Role      | Content   |
|-----------|---|
| User      | You're a financial advisor. A client wants to invest \$10,000. They can choose between two options: A) A stock that historically returns 12% annually but is volatile, or B) A bond that guarantees 6% annually. The client needs the money in 5 years for a down payment on a house. Which option do you recommend? Think step-by-step.  |
| Assistant | <p>To determine the best investment option for our client, I'll consider several factors step-by-step:</p> <p>&lt;thinking&gt;</p> <ol style="list-style-type: none"> <li>Understand the client's goal and timeline: <ul style="list-style-type: none"> <li>Goal: Down payment for a house</li> <li>Timeline: 5 years</li> <li>Implication: This is a short-term, high-priority financial goal. The client likely cannot afford significant risk or volatility.</li> </ul> </li> <li>Analyze Option A (Stock): <ul style="list-style-type: none"> <li>Pro: Higher historical return (12% annually)</li> <li>Con: Volatile (implying potential for significant gains or losses)</li> <li>Calculate potential 5-year return: <ul style="list-style-type: none"> <li>Best case (12% annually): <math>\\$10,000 * (1.12)^5 \approx \\$17,623.42</math></li> <li>Worst case (market crash): Could lose a substantial portion, e.g., 50% crash = \$5,000</li> </ul> </li> </ul> </li> </ol> |

# ANTHROPIC

Prompt engineering > **Let Claude think (chain of thought prompting) to increase performance**

Guaranteed: \$10,000 (1.00) vs \$10,002.20

## 4. Consider risk tolerance:

- Goal is a house down payment (essential, not discretionary)
- Short timeline (5 years) limits ability to recover from market downturns
- Conclusion: Client likely has low risk tolerance

## 5. Historical perspective:

- Stock's 12% is historical, not guaranteed
- In 5-year periods, stock market can vary wildly:
- 2003-2007 (pre-financial crisis): S&P 500 up ~82%
- 2007-2011 (during crisis): S&P 500 down ~2%
- Conclusion: Past performance doesn't guarantee future results, especially short-term

## 6. Final analysis:

- Option A (Stock): Higher potential return but unacceptable risk of not meeting the down payment goal
- Option B (Bond): Guaranteed to meet ~80% of best-case stock scenario, 100% certainty of having funds

</thinking>

<answer>

I recommend Option B, the bond with a guaranteed 6% annual return. While the stock's historical 12% return is tempting, the bond ensures you'll have \$13,382.25 in 5 years for your house down payment. Given the importance and short timeline of your goal, the stock's volatility poses an unacceptable risk. The bond provides certainty, which is invaluable for such a crucial financial milestone.

</answer>

## Prompt library

Get inspired by a curated selection of

## GitHub prompting tutorial

## Google Sheets prompting tutorial

# ANTHROPIC

Prompt engineering > **Let Claude think (chain of thought prompting) to increase performance**

Was this page helpful?

 Yes

 No

< Use examples (multishot prompting)

Use XML tags >

# Giving Claude a role with a system prompt

💡 While these tips apply broadly to all Claude models, you can find prompting tips specific to extended thinking models [here](#).

When using Claude, you can dramatically improve its performance by using the `system` parameter to give it a role. This technique, known as role prompting, is the most powerful way to use system prompts with Claude.

The right role can turn Claude from a general assistant into your virtual domain expert!

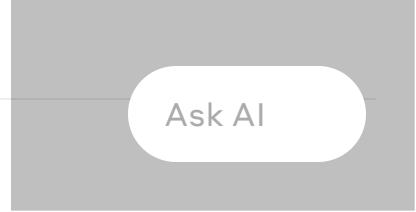
💡 **System prompt tips:** Use the `system` parameter to set Claude's role. Put everything else, like task-specific instructions, in the `user` turn instead.

## Why use role prompting?

**Enhanced accuracy:** In complex scenarios like legal analysis or financial modeling, role prompting can significantly boost Claude's performance.

**Tailored tone:** Whether you need a CFO's brevity or a copywriter's flair, role prompting adjusts Claude's communication style.

**Improved focus:** By setting the role context, Claude stays more within the bounds of your task's specific requirements.



Ask AI

# ANTHROPIC

Prompt engineering > **Giving Claude a role with a system prompt**

```
import anthropic

client = anthropic.Anthropic()

response = client.messages.create(
    model="claude-3-7-sonnet-20250219",
    max_tokens=2048,
    system="You are a seasoned data scientist at a Fortune 500 company.", # <-- role
    messages=[
        {"role": "user", "content": "Analyze this dataset for anomalies: <dataset>"},
    ]
)

print(response.content)
```

 **Role prompting tip:** Experiment with roles! A data scientist might see different insights than a marketing strategist for the same data. A data scientist specializing in customer insight analysis for Fortune 500 companies might yield different results still!

## Examples

### Example 1: Legal contract analysis

Without a role, Claude might miss critical issues:

#### Legal contract analysis without role prompting

# ANTHROPIC

## Prompt engineering > Giving Claude a role with a system prompt

</contract>

Focus on indemnification, liability, and IP ownership.

Assistant Here's a summary of the key points in the software licensing agreement:

1. Indemnification (Section 8):

- We indemnify the vendor against claims arising from our use.

2. Limitation of Liability (Section 9):

- Damages capped at \$500.

3. IP Ownership (Section 12):

- We retain ownership of our modifications.

Overall, the agreement seems standard. The indemnification and liability clauses are typical, and we maintain our IP rights.

With a role, Claude catches critical issues that could cost millions:

### Legal contract analysis with role prompting

| Role | Content   |
|------|---|
| User | You are the General Counsel of a Fortune 500 tech company. We're considering this software licensing agreement for our core data infrastructure:<br><contract><br>{{CONTRACT}}<br></contract> |

Analyze it for potential risks, focusing on indemnification, liability, and IP ownership. Give your professional opinion.

Assistant I've reviewed the software licensing agreement, and I have serious concerns that could expose our company to significant risks:

1. Indemnification (Section 8):

# ANTHROPIC

## Prompt engineering > Giving Claude a role with a system prompt

the agreement.

### 2. Limitation of Liability (Section 9):

- Critical Issue: Damages capped at \$500, including direct damages.
- Risk: Grossly inadequate. A critical failure in our data infrastructure could cost us millions in lost revenue and reputational damage.
- Recommendation: Reject. Negotiate for cap at 12 months of fees for direct damages, and carve-outs for data breaches.

### 3. IP Ownership (Section 12):

- Critical Issue: Vendor gains joint ownership of our modifications and integrations.
- Risk: They could use or sell our proprietary algorithms or data insights to competitors.
- Recommendation: Reject. We must retain full ownership of our work.

Professional Opinion: Do not sign this agreement. It is grossly unfavorable and exposes us to unacceptable levels of risk. If the vendor refuses to renegotiate these terms, I strongly advise exploring alternative providers or accelerating our in-house solution. The potential costs of proceeding with this agreement far outweigh any short-term gains.

## Example 2: Financial analysis

Without a role, Claude's analysis lacks depth:

### Financial analysis without role prompting

| Role | Content  |
|------|--|
| User | Analyze this dataset of our Q2 financials:<br><data><br>{{FINANCIALS}}<br></data><br><br>Highlight key trends and recommend actions. |

# ANTHROPIC

## Prompt engineering > Giving Claude a role with a system prompt

3. EBITDA Margin: 18%, down 2% due to R&D investments.
4. Cash Flow: \$4.1M operating, \$28M in reserves.

Overall, financials look strong with revenue growth and healthy margins. Increased R&D spending is impacting profitability but could pay off long-term. Maintain current course.

With a role, Claude delivers actionable insights:

### Financial analysis with role prompting

| Role      | Content  |
|-----------|--|
| User      | <p>You are the CFO of a high-growth B2B SaaS company. We're in a board meeting discussing our Q2 financials:</p> <pre>&lt;data&gt; {{FINANCIALS}} &lt;/data&gt;</pre> <p>Analyze key trends, flag concerns, and recommend strategic actions. Our investors want aggressive growth but are wary of our burn rate.</p>   |
| Assistant | <p>Board members, here's my analysis of our Q2 financials:</p> <ol style="list-style-type: none"><li>1. Revenue (\$15.2M, +22% YoY):<ul style="list-style-type: none"><li>- Insight: Enterprise growth (+30%) is outpacing overall growth.</li><li>- Flag: SMB segment declined 5%.</li><li>- Action: Reallocate 20% of SMB marketing budget to enterprise sales. Task product to simplify SMB onboarding.</li></ul></li><li>2. Margins (Gross: 72%, +3% QoQ; EBITDA: 18%, -2% QoQ):<ul style="list-style-type: none"><li>- Insight: Infrastructure optimizations boosted gross margins.</li><li>- Flag: R&amp;D spend on AI features dented EBITDA.</li><li>- Action: No change. Our AI roadmap is critical for enterprise stickiness.</li></ul></li><li>3. Cash (\$4.1M OCF, \$28M reserves):<ul style="list-style-type: none"><li>- Insight: 15-month runway at current burn.</li></ul></li></ol> |

# ANTHROPIC

## Prompt engineering > Giving Claude a role with a system prompt

is our growth engine—invest there. But watch SMB churn; it could bleed into enterprise. Our AI bet is sound, but we must convert it to ARR by Q4, or we cut R&D by 30%.

In sum: Double down on enterprise, streamline SMB, optimize sales, and monetize AI.  
Questions?

### 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.

Was this page helpful?

Yes

No

Use XML tags

Prefill Claude's response

# ANTHROPIC

Prompt engineering > **Giving Claude a role with a system prompt**

**PROMPT ENGINEERING** Copy page

# Prefill Claude's response for greater output control

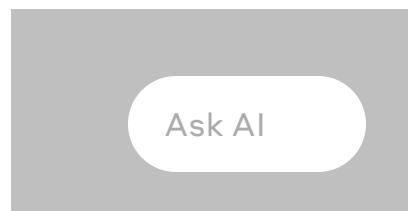
- ➊ While these tips apply broadly to all Claude models, you can find prompting tips specific to extended thinking models [here](#).
- ➋ Prefilling is only available for non-extended thinking modes. It's not currently supported with extended thinking.

When using Claude, you have the unique ability to guide its responses by prefilling the `Assistant` message. This powerful technique allows you to direct Claude's actions, skip preambles, enforce specific formats like JSON or XML, and even help Claude maintain character consistency in role-play scenarios.

In some cases where Claude is not performing as expected, a few prefilled sentences can vastly improve Claude's performance. A little prefilling goes a long way!

## How to prefill Claude's response

To prefill, include the desired initial text in the `Assistant` message (Claude's response will continue from where the `Assistant` message leaves off):

Ask AI

# ANTHROPIC

## Prompt engineering > Prefill Claude's response for greater output control

```
model="claude-opus-4-20250514",
max_tokens=1024,
messages=[
    {"role": "user", "content": "What is your favorite color?"},
    {"role": "assistant", "content": "As an AI assistant, I don't have a favo
]
)
```

 The prefill content cannot end with trailing whitespace. A prefill like "As an AI assistant, I " (with a space at the end) will result in an error.

## Examples

### Example 1: Controlling output formatting and skipping the preamble

 Power user tip: Prefilling { forces Claude to skip the preamble and directly output the JSON object. This is cleaner, more concise, and easier for programs to parse without additional processing.

### Example: Structured data extraction without prefilling

| Role | Content  |
|------|--|
| User | Extract the name, size, price, and color from this product description as a JSON object:<br><br><description><br>The SmartHome Mini is a compact smart home assistant available in black or white for only \$49.99. At just 5 inches wide, it lets you control lights, thermostats, and other connected devices via voice or app—no matter where you place it in your home. This affordable little hub brings convenient hands-free control to your smart devices.<br></description> |

# ANTHROPIC

## Prompt engineering > Prefill Claude's response for greater output control

```
"name": "SmartHome Mini",
"size": "5 inches wide",
"price": "$49.99",
"colors": ["black", "white"]
}
``
```

I've extracted the following details from the product description:

- **Name:** SmartHome Mini
- **Size:** 5 inches wide
- **Price:** \$49.99
- **Colors:** Available in black and white

The JSON object provides a structured format that's easy for programs to parse and use. Let me know if you need any modifications or have any other questions!

## Example: Structured data extraction with prefilling

| Role                                | Content  |
|-------------------------------------|--|
| User                                | Extract the name, size, price, and color from this product description as a JSON object:<br><br><description><br>The SmartHome Mini is a compact smart home assistant available in black or white for only \$49.99. At just 5 inches wide, it lets you control lights, thermostats, and other connected devices via voice or app—no matter where you place it in your home. This affordable little hub brings convenient hands-free control to your smart devices.<br></description> |
| Assistant<br>(prefill)              | {  |
| Assistant<br>(Claude's<br>response) | "name": "SmartHome Mini", "size": "5 inches wide", "price": "\$49.99", "colors": [   |

# ANTHROPIC

Prompt engineering > **Prefill Claude's response for greater output control**

## Example 2: Maintaining character in roleplay scenarios

 **Role-play tip:** Prefilling a bracketed `[ROLE_NAME]` can remind Claude stay in character, even for longer and more complex conversations. This is especially powerful when combined with role prompting in the `system` parameter.

### Example: Maintaining character without role prompting

| Role  | Content   |
|---|---|
| User (50 conversation turns in with “Sherlock Holmes” Claude) | What do you deduce about the owner of this shoe?<br><image>muddy_shoe.jpg</image>   |
| Assistant   | As an AI assistant, I would be happy to analyze the image of a muddy shoe as if you had provided it. Here’s my deduction... |

### Example: Maintaining character with role prompting

| Role  | Content  |
|---|--|
| User (50 conversation turns in with “Sherlock Holmes” Claude) | What do you deduce about the owner of this shoe?<br><image>muddy_shoe.jpg</image>  |
| Assistant (prefill)   | [Sherlock Holmes]  |
| Assistant (Claude’s response)                                 | Ah, what have we here? A most curious specimen, Watson! Let us examine this shoe with a discerning eye. The mud, you see, is not your common |

# ANTHROPIC

## Prompt engineering > Prefill Claude's response for greater output control

Now, observe the laces! They are tied with a particular knot favored by those in the medical profession—doubtless a habit acquired during years of tying surgical gowns. But the scuff marks on the toe box tell a different tale: they are consistent with the kicking motion one might employ in the martial arts, specifically Baritsu.

We may conclude, then, that our absent shoe owner is a doctor, possibly a surgeon, who resides in Whitechapel but frequents Hampstead Heath—perhaps for a secret Baritsu practice, away from prying eyes. A most intriguing case, Watson! We must make haste; the game is afoot!

### 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.

Was this page helpful?

 Yes

 No

◀ Give Claude a role (system prompts)

Chain complex prompts ▶

# ANTHROPIC

Prompt engineering > **Prefill Claude's response for greater output control**

# Chain complex prompts for stronger performance

- ⓘ 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?

- Accuracy:** Each subtask gets Claude's full attention, reducing errors.
- Clarity:** Simpler subtasks mean clearer instructions and outputs.
- 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.

[Ask AI](#)

# ANTHROPIC

Prompt engineering > **Chain complex prompts for stronger performance**

## How to chain prompts

- 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.

# ANTHROPIC

Prompt engineering > **Chain complex prompts for stronger performance**

## Prompt 1

| Role      | Content  |
|-----------|--|
| User      | <p>Summarize this medical research paper.</p> <pre>&lt;paper&gt;{{RESEARCH_PAPER}}&lt;/paper&gt;</pre> <p>Focus on methodology, findings, and clinical implications.</p>   |
| Assistant | <p>Methodology: Double-blind RCT with 500 Type 2 diabetes patients. Group A {{SUMMARY}} (n=250) received novel GLP-1/GIP dual agonist; Group B (n=250) received standard GLP-1 agonist. 24-week trial, primary endpoint: HbA1c reduction.</p> <p>Findings:</p> <ul style="list-style-type: none"> <li>- Group A: 1.8% mean HbA1c reduction</li> <li>- Group B: 1.1% mean HbA1c reduction</li> <li>- 68% of Group A vs. 52% of Group B achieved HbA1c &lt;7%</li> <li>- Group A: 5.2kg mean weight loss</li> <li>- Group B: 3.1kg mean weight loss</li> </ul> <p>Safety: Similar adverse event profiles, no serious events attributed to treatment.</p> <p>Clinical Implications: Dual agonist shows superior glycemic control and weight loss compared to standard therapy. Could be a game-changer, especially for obese diabetics.</p> |

## Prompt 2

| Role | Content   |
|------|---|
| User | <p>Your task is to provide feedback on a research paper summary. Here is a summary of a medical research paper:</p> <pre>&lt;summary&gt; {{SUMMARY}} &lt;/summary&gt;</pre> <p>Here is the research paper:</p> <pre>&lt;paper&gt; {{RESEARCH_PAPER}} &lt;/paper&gt;</pre> |

# ANTHROPIC

## Prompt engineering > Chain complex prompts for stronger performance

- Accurately noted safety profile and lack of serious adverse events.
- Assertion of “game-changer” is justified by significantly better outcomes.

### 2. Clarity: A-

- Concise and well-structured with clear sections.
- Good use of comparative language (e.g., “up 3% from Q1”) for context.
- Minor issue: Acronyms like “RCT,” “GLP-1,” and “GIP” not defined. Consider spelling out first use.

### 3. Completeness: B+

- Covers all requested areas: methodology, findings, clinical implications.
- Missing: Baseline characteristics of groups (were they well-matched?).
- Missing: Any secondary endpoints (e.g., blood pressure, lipids).
- Consider: Long-term implications or need for further research.

Overall, the summary is accurate and clear, providing key insights for a quick grasp of the study. To elevate to A+, define acronyms, mention group baseline stats, note any secondary outcomes, and briefly touch on the need for long-term studies.

## Prompt 3

| Role | Content  |
|------|--|
| User | Your task is to improve a paper summary given feedback. Here is the first draft of a medical research paper:<br><summary><br>{{SUMMARY}}<br></summary> |
|      | Here is the research paper:<br><paper><br>{{RESEARCH_PAPER}}<br></paper>   |
|      | Here is the feedback:<br><feedback><br>{{FEEDBACK}}<br></feedback>   |
|      | Update the summary based on the feedback.  |

# ANTHROPIC

## Prompt engineering > Chain complex prompts for stronger performance

Groups were well-matched for age, BMI, and baseline HbA1c.

Findings:

- Group A: 1.8% mean HbA1c reduction
- Group B: 1.1% mean HbA1c reduction
- 68% of Group A vs. 52% of Group B achieved HbA1c <7%
- Group A: 5.2kg mean weight loss
- Group B: 3.1kg mean weight loss

Secondary endpoints showed improvements in blood pressure and lipid profiles for both groups, with Group A showing slightly better results.

Safety: Similar adverse event profiles, with no serious events attributed to either treatment.

Clinical Implications: The dual GLP-1/GIP agonist demonstrates superior glycemic control and weight loss compared to standard GLP-1 therapy. These results suggest it could be a significant advancement, particularly for obese diabetics. However, longer-term studies are needed to confirm sustained efficacy and safety, as well as to investigate potential benefits on cardiovascular outcomes.

## Examples

### Example: Analyzing a legal contract (without chaining)

Note that Claude misses the instruction to provide proposed changes in its email draft.

| Role | Content  |
|------|--|
| User | You're our Chief Legal Officer. Review this SaaS contract for risks, focusing on data privacy, SLAs, and liability caps.<br><br><contract> |

# ANTHROPIC

## Prompt engineering > Chain complex prompts for stronger performance

Assistant After reviewing the SaaS contract, I've identified several concerns:

1. Data Privacy (Section 7):

- Issue: Vendor has broad rights to use client data for “service improvement.”
- Risk: Could lead to unauthorized data sharing or monetization.

2. SLAs (Section 5):

- Issue: 99.5% uptime guarantee is below industry standard.
- Risk: Excessive downtime could impact our operations.

3. Liability (Section 9):

- Issue: Cap set at 3 months of fees.
- Risk: Inadequate for potential data breaches or service failures.

Draft email:

Dear {{VENDOR}},

After reviewing the SaaS agreement, we have concerns:

1. Data Privacy: We require that our data only be used for direct service provision, not broad “service improvement.” Please amend Section 7.

2. SLAs: 99.5% uptime is insufficient. We need 99.99% with clearly defined penalties. Please revise Section 5.

3. Liability: The current cap of 3 months’ fees is inadequate. We propose a cap of 12 months’ fees, with no cap for data breaches or willful misconduct.

We value our potential partnership and believe these changes will set us up for success. Please send a revised agreement.

Best regards,  
{{CLO\_NAME}}

## Example: Analyzing a legal contract (with chaining)

# ANTHROPIC

## Prompt engineering > Chain complex prompts for stronger performance

User: You're our chief legal officer. Review this SaaS contract for risks, focusing on data privacy, SLAs, and liability caps.

```
<contract>
{{CONTRACT}}
</contract>
```

Output your findings in <risks> tags.

- |           |  |
|-----------|--|
| Assistant | <ol style="list-style-type: none"> <li>1. Data Privacy (Section 7): <ul style="list-style-type: none"> <li>- Issue: Vendor has broad rights to use client data for “service improvement.”</li> <li>- Risk: Could lead to unauthorized data sharing or monetization of client data.</li> <li>- Impact: Potential breach of client trust, regulatory fines (GDPR, CCPA).</li> </ul> </li> <br/> <li>2. SLAs (Section 5): <ul style="list-style-type: none"> <li>- Issue: 99.5% uptime guarantee is below industry standard (typically 99.99%).</li> <li>- Risk: Up to 43.8 hours of downtime per year vs. 52.6 minutes at 99.99%.</li> <li>- Impact: Operational disruptions, revenue loss, client dissatisfaction.</li> </ul> </li> <br/> <li>3. Liability (Section 9): <ul style="list-style-type: none"> <li>- Issue: Liability cap set at 3 months of fees.</li> <li>- Risk: Inadequate for potential data breaches, IP theft, or prolonged service failures.</li> <li>- Impact: Insufficient compensation for damages that could far exceed the cap, like class-action lawsuits or regulatory fines.</li> </ul> </li> </ol> |
|-----------|--|

## Prompt 2

Role	Content
User	<p>Draft an email to a SaaS product vendor outlining the following concerns and proposing changes. Here are the concerns:</p> <pre>&lt;concerns&gt; {{CONCERN}}</pre>
Assistant	<p>Dear {{VENDOR}},</p> <p>After a thorough review of the proposed SaaS agreement, our legal team has identified several high-priority concerns that must be addressed before we can proceed:</p> <ol style="list-style-type: none"> <li>1. Data Privacy (Section 7):</li> </ol>

# ANTHROPIC

## Prompt engineering > Chain complex prompts for stronger performance

### 2. SLAs (Section 5):

- Current: 99.5% uptime guarantee.
- Concern: Up to 43.8 hours of downtime per year, risking operational disruptions.
- Proposed Change: 99.99% uptime with tiered service credits. E.g., 10% credit for <99.99%, 25% for <99.9%, 50% for <99.5%.

### 3. Liability (Section 9):

- Current: Cap at 3 months of fees.
- Concern: Grossly inadequate for data breaches, IP theft, or prolonged outages.
- Proposed Change: 12 months' fees for general liability. Uncapped for data breaches, willful misconduct, IP infringement.

These changes are non-negotiable for us. They align with industry standards and our commitment to data security, operational excellence, and fiduciary responsibility. We value the potential of our partnership, and addressing these will set a strong foundation.

Please have your legal team review and incorporate these changes. We're happy to discuss further to ensure mutual understanding and agreement.

Best regards,  
{{CLO\_NAME}}

## Prompt 3

Role	Content
User	Your task is to review an email and provide feedback. Here is the email: <email> {{EMAIL}} </email>  Give feedback on tone, clarity, and professionalism.
Assistant	1. Tone: - Professional and assertive without being confrontational. - Use of “non-negotiable” is strong but justified given the criticality of the issues. - Collaborative language (“we value,” “happy to discuss”) balances the firmness, maintaining a constructive tone.  2. Clarity:

# ANTHROPIC

## Prompt engineering > Chain complex prompts for stronger performance

### 3. Professionalism:

- Focuses on mutual benefit and industry standards, not just our demands.
- Mentions broader responsibilities (data security, operational excellence, fiduciary duty), elevating the discussion.
- Offers further discussion, showing commitment to understanding and resolving issues.

Overall, the email strikes the right balance of assertiveness and collaboration. It clearly articulates our concerns, proposes specific solutions, and maintains a professional, partnership-oriented tone. No revisions needed; proceed with sending.

## Example: Multitenancy strategy review

### Prompt 1

Role	Content
User	<p>As a senior solutions architect, review and analyze this multitenancy strategy for our new enterprise SaaS product.</p> <pre>&lt;strategy&gt; {{STRATEGY}} &lt;/strategy&gt;</pre> <p>Focus on scalability, security, and cost-effectiveness.</p>
Assistant	<p>1. Scalability:</p> <p>{ANALYSIS}</p> <ul style="list-style-type: none"> <li>+ Database sharding by tenant reduces contention, supports high growth.</li> <li>+ Microservices allow independent scaling of components.</li> <li>- Shared app servers could bottleneck under extreme load.</li> <li>- No mention of auto-scaling or load balancing for app tier.</li> </ul> <p>2. Security:</p> <ul style="list-style-type: none"> <li>+ Tenant data isolation via sharding mitigates data leaks.</li> <li>+ JWT with short expiry and refresh tokens is industry best practice.</li> <li>- No mention of encryption (at rest, in transit) or key management.</li> <li>- IAM strategy lacks detail on roles, permissions, or RBAC.</li> </ul>

# ANTHROPIC

## Prompt engineering > Chain complex prompts for stronger performance

Two discussions of related instances of spot instances for cost savings.

### Prompt 2

Role	Content
User	<p>Draft a strategy review document for engineering leadership based on this analysis of a multitenancy strategy.</p> <pre>&lt;strategy&gt; {{STRATEGY}} &lt;/strategy&gt;</pre> <pre>&lt;analysis&gt; {{ANALYSIS}} &lt;/analysis&gt;</pre> <p>Include an executive summary, detailed analysis, and recommendations.</p>
Assistant	<pre># Multitenancy Strategy Review: SaaS Enterprise Product {{STRATEGY_DOC}}</pre> <p>## Executive Summary</p> <p>The proposed strategy demonstrates a solid foundation in scalability and security principles but has gaps in implementation details. It excels in data isolation and component-level scaling but lacks robust app-tier scaling and advanced security measures. Cost optimizations are present but may falter at high scale. With targeted enhancements, this strategy can effectively support our enterprise SaaS ambitions.</p> <p>## Detailed Analysis</p> <p>### 1. Scalability: B+</p> <p>Strengths:</p> <ul style="list-style-type: none"> <li>- Database sharding by tenant is excellent. It minimizes query latency, supports high data growth, and facilitates seamless per-tenant scaling.</li> <li>- Microservices architecture allows scaling individual components based on demand, crucial for handling diverse enterprise workloads.</li> </ul> <p>Weaknesses:</p> <ul style="list-style-type: none"> <li>- Shared application servers risk becoming a bottleneck. As tenant count and traffic grow, these could degrade performance across all tenants.</li> </ul>

# ANTHROPIC

## Prompt engineering > **Chain complex prompts for stronger performance**

... SECURITY ...

### Strengths:

- Sharded databases provide strong tenant data isolation, mitigating risks of data leaks or unauthorized access between tenants.
- JWT with short expiry and refresh tokens align with OWASP best practices, reducing token vulnerability windows.

### Weaknesses:

- No mention of data encryption at rest or in transit (TLS 1.3). This is non-negotiable for enterprise data.
- Key management strategy is missing. How are encryption keys, JWT secrets, and DB credentials managed and rotated?
- IAM lacks depth. We need RBAC with well-defined roles, attribute-based access control (ABAC) for fine-grained permissions, and integration with enterprise identity providers (SAML, OIDC).

### ### 3. Cost-effectiveness: B

### Strengths:

- Shared app servers reduce idle capacity costs, good for steady-state traffic.
- Serverless for infrequent tasks (e.g., reporting) is smart; pay only when these resources are used.

### Weaknesses:

- Per-tenant databases may not be cost-effective at scale compared to a well-designed multi-tenant database with robust data partitioning.
- No mention of cost-saving measures like reserved instances for predictable workloads or spot instances for fault-tolerant tasks.
- Lack of app-tier auto-scaling could lead to over-provisioning, increasing costs.

### ## Recommendations

**1. App Tier Resilience:** Implement auto-scaling with Kubernetes or cloud-native services (EKS, GKE). Deploy across availability zones with a CDN for static assets.

**2. Security Hardening:**

- a. Encrypt data at rest (AES-256) and in transit (TLS 1.3).
- b. Use a managed KMS (AWS KMS, GCP KMS) for key management.
- c. Enhance IAM with RBAC, ABAC, and SAML/OIDC for enterprise SSO.

# ANTHROPIC

## Prompt engineering > Chain complex prompts for stronger performance

c. Implement proactive monitoring and right-sizing to avoid overprovisioning.

4. **Proof of Concept (PoC):** Before full implementation, conduct a PoC with a high-scale simulation:
  - a. Test sharded DB vs. optimized multi-tenant DB for cost and performance.
  - b. Stress-test app tier with simulated enterprise-level traffic spikes.
  - c. Conduct third-party security audit, focusing on data isolation and access controls.

With these enhancements, our multi-tenancy strategy will be enterprise-grade: scalable, secure, and cost-effective. The PoC will validate our design choices and instill confidence in our enterprise clients.

## Prompt 3

Role	Content
User	<p>Grade this strategy review document for clarity, actionability, and alignment with enterprise priorities.</p> <pre>&lt;priorities&gt; {{PRIORITIES}} &lt;/priorities&gt;  &lt;strategy_doc&gt; {{STRATEGY_DOC}} &lt;/strategy_doc&gt;</pre>
Assistant	<p>1. Clarity: A</p> <ul style="list-style-type: none"> <li>+ Clear structure with Executive Summary, Detailed Analysis, and Recommendations.</li> <li>+ Analysis uses a grading system (B+, B-) for quick grasp of strengths/weaknesses.</li> <li>+ Technical terms (RBAC, ABAC, TLS) are explained or contextualized, aiding non-technical leadership.</li> <li>+ Visual aids (bullet points, indentation) enhance readability.</li> </ul> <p>2. Actionability: A-</p> <ul style="list-style-type: none"> <li>+ Recommendations are specific and technical (e.g., “Implement auto-scaling with Kubernetes”).</li> <li>+ PoC plan with clear objectives provides a roadmap.</li> <li>+ Missing: Timelines, resource requirements, or team assignments for each recommendation.</li> </ul>

# ANTHROPIC

## Prompt engineering > **Chain complex prompts for stronger performance**

- Scalability: Discusses handling enterprise-level traffic spikes.
- Security: Focuses on data isolation, encryption, and enterprise IAM.
- Cost: Strategies align with enterprise FinOps (reserved instances, right-sizing).
- + PoC with “high-scale simulation” and security audit shows commitment to enterprise-grade quality.
- + Language resonates with enterprise concerns: “unacceptable risk,” “enterprise-grade,” “instill confidence.”

Overall, the document excels in clarity and enterprise alignment. The actionability is strong but could be elevated to A+ with a “Next Steps” section. The technical depth, coupled with clear explanations, makes this highly effective for both technical and non-technical leadership. Great work!

### **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.

Was this page helpful?

 Yes

 No

# ANTHROPIC

Prompt engineering > **Chain complex prompts for stronger performance**

# Long context prompting tips

- 💡 While these tips apply broadly to all Claude models, you can find prompting tips specific to extended thinking models [here](#).

Claude's extended context window (200K tokens for Claude 3 models) enables handling complex, data-rich tasks. This guide will help you leverage this power effectively.

## Essential tips for long context prompts

**Put longform data at the top:** Place your long documents and inputs (~20K+ tokens) near the top of your prompt, above your query, instructions, and examples. This can significantly improve Claude's performance across all models.

- 💡 Queries at the end can improve response quality by up to 30% in tests, especially with complex, multi-document inputs.

**Structure document content and metadata with XML tags:** When using multiple documents, wrap each document in `<document>` tags with `<document_content>` and `<source>` (and other metadata) subtags for clarity.

### Example multi-document structure

**Ground responses in quotes:** For long document tasks, ask Claude to quote relevant parts of the documents first before carrying out its task. This helps Claude cut through the “noise” of the rest of the document's contents.

# ANTHROPIC

## Example quote extraction

### Prompt engineering > Long context prompting tips

You are an AI physician's assistant. Your task is to help doctors diagnose patients.

```
<documents>
  <document index="1">
    <source>patient_symptoms.txt</source>
    <document_content>
      {{PATIENT_SYMPTOMS}}
    </document_content>
  </document>
  <document index="2">
    <source>patient_records.txt</source>
    <document_content>
      {{PATIENT_RECORDS}}
    </document_content>
  </document>
  <document index="3">
    <source>patient01_appt_history.txt</source>
    <document_content>
      {{PATIENT01_APPOINTMENT_HISTORY}}
    </document_content>
  </document>
</documents>
```

Find quotes from the patient records and appointment history that are relevant to the patient's symptoms.

[Prompt library](#)

[GitHub prompting tutorial](#)

[Google Sheets prompting tutorial](#)

Get inspired by a **ANTHROPIC** curated selection of prompts for various tasks and use cases. > **Long context prompting tips**

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

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

Was this page helpful?

 Yes

 No

< Chain complex prompts

Extended thinking tips >

**PROMPT ENGINEERING**

# Extended thinking tips

 Copy page

This guide provides advanced strategies and techniques for getting the most out of Claude's extended thinking features. Extended thinking allows Claude to work through complex problems step-by-step, improving performance on difficult tasks.

See [Extended thinking models](#) for guidance on deciding when to use extended thinking.

## Before diving in

This guide presumes that you have already decided to use extended thinking mode and have reviewed our basic steps on [how to get started with extended thinking](#) as well as our [extended thinking implementation guide](#).

## Technical considerations for extended thinking

Thinking tokens have a minimum budget of 1024 tokens. We recommend that you start with the minimum thinking budget and incrementally increase to adjust based on your needs and task complexity.

For workloads where the optimal thinking budget is above 32K, we recommend that you use [batch processing](#) to avoid networking issues. Requests pushing the model to think above 32K tokens causes long running requests that might run up against system timeouts and open connection limits.

Extended thinking performs best in English, though final outputs can be in [any language](#) [Claude supports](#).

 Ask AI

# ANTHROPIC

Prompt engineering > Extended thinking tips

## Prompting techniques for extended thinking

### Use general instructions first, then troubleshoot with more step-by-step instructions

Claude often performs better with high level instructions to just think deeply about a task rather than step-by-step prescriptive guidance. The model's creativity in approaching problems may exceed a human's ability to prescribe the optimal thinking process.

For example, instead of:

User

Think through this math problem step by step:

1. First, identify the variables
  2. Then, set up the equation
  3. Next, solve for x
- ...

Consider:

User

Try in Console

Please think about this math problem thoroughly and in great detail.

Consider multiple approaches and show your complete reasoning.

Try different methods if your first approach doesn't work.

That said, Claude can still effectively follow complex structured execution steps when needed. The model can handle even longer lists with more complex instructions than previous versions. We recommend that you start with more generalized instructions, then

# ANTHROPIC

Prompt engineering > Extended thinking tips

**Multishot prompting** works well with extended thinking. When you provide Claude examples of how to think through problems, it will follow similar reasoning patterns within its extended thinking blocks.

You can include few-shot examples in your prompt in extended thinking scenarios by using XML tags like `<thinking>` or `<scratchpad>` to indicate canonical patterns of extended thinking in those examples.

Claude will generalize the pattern to the formal extended thinking process. However, it's possible you'll get better results by giving Claude free rein to think in the way it deems best.

Example:

User

Try in Console

I'm going to show you how to solve a math problem, then I want you to solve a sim

Problem 1: What is 15% of 80?

`<thinking>`

To find 15% of 80:

1. Convert 15% to a decimal:  $15\% = 0.15$
2. Multiply:  $0.15 \times 80 = 12$

`</thinking>`

The answer is 12.

Now solve this one:

Problem 2: What is 35% of 240?

## Maximizing instruction following with extended thinking

# ANTHROPIC

Prompt engineering > **Extended thinking tips**

↳ Executes those instructions in the response

To maximize instruction following:

Be clear and specific about what you want

For complex instructions, consider breaking them into numbered steps that Claude should work through methodically

Allow Claude enough budget to process the instructions fully in its extended thinking

## Using extended thinking to debug and steer Claude's behavior

You can use Claude's thinking output to debug Claude's logic, although this method is not always perfectly reliable.

To make the best use of this methodology, we recommend the following tips:

We don't recommend passing Claude's extended thinking back in the user text block, as this doesn't improve performance and may actually degrade results.

Prefilling extended thinking is explicitly not allowed, and manually changing the model's output text that follows its thinking block is likely going to degrade results due to model confusion.

When extended thinking is turned off, standard `assistant` response text prefill is still allowed.

- ⚠️ Sometimes Claude may repeat its extended thinking in the assistant output text. If you want a clean response, instruct Claude not to repeat its extended thinking and to only output the answer.

## Making the best of long outputs and longform thinking

# ANTHROPIC

## Prompt engineering > Extended thinking tips

Increase both the maximum extended thinking length AND explicitly ask for longer outputs

For very long outputs (20,000+ words), request a detailed outline with word counts down to the paragraph level. Then ask Claude to index its paragraphs to the outline and maintain the specified word counts

**⚠️** We do not recommend that you push Claude to output more tokens for outputting tokens' sake. Rather, we encourage you to start with a small thinking budget and increase as needed to find the optimal settings for your use case.

Here are example use cases where Claude excels due to longer extended thinking:

### Complex STEM problems

Complex STEM problems require Claude to build mental models, apply specialized knowledge, and work through sequential logical steps—processes that benefit from longer reasoning time.

**Standard prompt**    **Enhanced prompt**

User

Try in Console

Write a python script for a bouncing yellow ball within a square,  
make sure to handle collision detection properly.  
Make the square slowly rotate.

# ANTHROPIC

Prompt engineering > Extended thinking tips

## Constraint optimization problems

Constraint optimization challenges Claude to satisfy multiple competing requirements simultaneously, which is best accomplished when allowing for long extended thinking time so that the model can methodically address each constraint.

Standard prompt   Enhanced prompt

---

User

Try in Console

Plan a week-long vacation to Japan.

- ➊ This open-ended request typically results in only about a few seconds of thinking time.

## Thinking frameworks

Structured thinking frameworks give Claude an explicit methodology to follow, which may work best when Claude is given long extended thinking space to follow each step.

Standard prompt   Enhanced prompt

---

# ANTHROPIC

Prompt engineering > Extended thinking tips

THE PERSONALIZED MEDICINE MARKET BY 2027.

Begin with:

1. A Blue Ocean Strategy canvas
2. Apply Porter's Five Forces to identify competitive pressures

Next, conduct a scenario planning exercise with four distinct futures based on regulatory and technological variables.

For each scenario:

- Develop strategic responses using the Ansoff Matrix

Finally, apply the Three Horizons framework to:

- Map the transition pathway
- Identify potential disruptive innovations at each stage

① By specifying multiple analytical frameworks that must be applied sequentially, thinking time naturally increases as Claude works through each framework methodically.

## Have Claude reflect on and check its work for improved consistency and error handling

You can use simple natural language prompting to improve consistency and reduce errors:

1. Ask Claude to verify its work with a simple test before declaring a task complete
2. Instruct the model to analyze whether its previous step achieved the expected result
3. For coding tasks, ask Claude to run through test cases in its extended thinking

Example:

# ANTHROPIC

Prompt engineering > Extended thinking tips

- n=0
- n=1
- n=5
- n=10

And fix any issues you find.

## Next steps

### Extended thinking cookbook

Explore practical examples of extended thinking in our cookbook.

### Extended thinking guide

See complete technical documentation for implementing extended thinking.

Was this page helpful?

 Yes

 No

< Long context tips

Define success criteria >