

LLM Mastery

Master ChatGPT, Claude & Gemini with Advanced Prompting Techniques

Get 10x better results than 99% of users by learning how top LLMs actually work, and the exact prompting frameworks used by professionals.

A Study Guide for AI Learners

Table of Contents

- 1. Introduction: Why Prompting Matters
- 2. Understanding ChatGPT, Claude & Gemini
- 3. The Anatomy of a Great Prompt
- 4. Core Prompting Techniques
- 5. Advanced Prompting Frameworks
- 6. Model-Specific Tips (ChatGPT vs Claude vs Gemini)
- 7. Common Mistakes to Avoid
- 8. Practice Exercises
- 9. Quick-Reference Cheat Sheet

1. Introduction: Why Prompting Matters

Large Language Models (LLMs) like ChatGPT, Claude, and Gemini are incredibly powerful, but the quality of what you get out of them depends heavily on the quality of what you put in. The same model can produce a generic, mediocre answer or a brilliant, tailored one — the only difference is the prompt.

This is called **Prompt Engineering**: the skill of communicating clearly and strategically with an AI so it understands exactly what you want, how you want it, and why. Mastering this skill lets you save time, get more accurate answers, and unlock capabilities most users never discover.

Why 99% of users get mediocre results:

- They write vague, one-line prompts and expect perfect answers.
- They don't give the AI context about their goal or audience.
- They accept the first answer instead of refining it.
- They don't know the model's strengths (or use the wrong model for the task).

2. Understanding ChatGPT, Claude & Gemini

While all three are LLMs, each has a distinct personality, strengths, and ideal use cases.

Model	Made By	Known For	Best Use Cases
ChatGPT	OpenAI	Versatility, plugins/tools, coding, image generation	General tasks, coding, brainstorming, data analysis
Claude	Anthropic	Long context, careful reasoning, natural writing, safety	Long documents, creative writing, research, careful analysis
Gemini	Google	Real-time info, multimodal input, Google ecosystem	Search-integrated tasks, multimodal (image/video), World

Key takeaway: don't just pick one favorite model — match the model to the task. Use Claude for long documents and nuanced writing, ChatGPT for broad tool-integrated tasks, and Gemini when you need real-time or multimodal capability.

3. The Anatomy of a Great Prompt

A strong prompt usually contains up to five elements. You don't need all five every time, but knowing them lets you build precise prompts on demand.

- **Role** — Who should the AI act as? (e.g. 'You are an expert copywriter')
- **Task** — What exactly do you want done?
- **Context** — Background info the AI needs to understand the situation
- **Format** — How should the output look? (bullet list, table, word count, tone)
- **Constraints** — Rules or limits (avoid jargon, stay under 200 words, etc.)

Example — Weak vs Strong Prompt

Weak: "Write about marketing."

Strong: "You are a senior marketing strategist. Write a 150-word Instagram caption promoting a sustainable water bottle to environmentally-conscious millennials. Use a friendly, witty tone and include one call-to-action."

4. Core Prompting Techniques

Zero-Shot Prompting

Asking the model to perform a task with no examples given — relies purely on its training.

```
"Translate this sentence into French: 'Good morning, how are you?'"
```

Few-Shot Prompting

Providing a few examples of the input/output pattern before asking your real question, so the model mimics the pattern.

```
"Q: Happy -> Sad\nQ: Fast -> Slow\nQ: Big -> ?"
```

Chain-of-Thought (CoT) Prompting

Asking the model to reason step-by-step before giving a final answer — dramatically improves accuracy on logic, math, and complex questions.

```
"Solve this step by step: If a train travels 60 km in 45 minutes, what is its speed in km/h?"
```

Role/Persona Prompting

Assigning the AI a specific role or persona to shape tone, expertise level, and perspective.

```
"Act as a senior Python developer reviewing my code for bugs and best practices."
```

Instruction-Based Prompting

Giving direct, clear, and structured instructions rather than vague questions.

```
"List 5 pros and 5 cons of remote work in a two-column table."
```

Iterative Refinement

Treating the first response as a draft, then asking follow-up prompts to refine, shorten, or adjust tone.

```
"Make this more concise and add a touch of humor."
```

5. Advanced Prompting Frameworks

The RTF Framework (Role - Task - Format)

A simple 3-part structure: state the Role the AI should play, the Task to complete, and the Format of the output. Great for quick, effective prompts.

```
"Role: You are a nutritionist. Task: Create a 3-day meal plan for muscle gain.  
Format: Present as a table with meals, calories, and protein."
```

The CO-STAR Framework

- **Context** — background information
- **Objective** — the specific goal
- **Style** — writing style to emulate
- **Tone** — emotional tone (formal, casual, persuasive)
- **Audience** — who is this for
- **Response format** — structure of the output

Self-Consistency Prompting

For tricky reasoning problems, ask the model to solve the same problem multiple ways or double-check its own answer before finalizing it. This reduces errors on math and logic tasks.

```
"Solve this problem, then verify your answer using a different method before giving  
your final response."
```

Prompt Chaining

Breaking a complex task into a sequence of smaller prompts, where the output of one becomes the input of the next. Useful for research, long-form writing, and multi-step projects.

```
Step 1: "Brainstorm 10 blog topic ideas about productivity." Step 2: "Pick the best  
one and create an outline." Step 3: "Write the full article based on this outline."
```

Negative Prompting

Explicitly telling the model what to avoid — tone, content, structure — sharpens the output.

```
"Write a product description. Avoid buzzwords like 'revolutionary' or  
'game-changing'. Do not use exclamation marks."
```

6. Model-Specific Tips

ChatGPT

- Great with custom instructions/memory — set persistent preferences once.
- Use it for code with explicit language and library versions.
- Ask it to use tools (browsing, code execution) explicitly when needed.

Claude

- Excels with long documents — paste large files and ask for structured analysis.
- Responds very well to clearly labeled sections (Role, Context, Task) in the prompt.
- Great for nuanced writing — ask it to 'think step by step' for reasoning-heavy tasks.

Gemini

- Strong for tasks needing up-to-date or real-time information.
- Good multimodal support — combine text prompts with images for analysis.
- Integrates well with Google Workspace (Docs, Sheets) for productivity tasks.

7. Common Mistakes to Avoid

- Being too vague ('write something good') instead of specific.
- Not specifying format, length, or tone.
- Giving up after one bad response instead of refining the prompt.
- Overloading a single prompt with too many unrelated tasks.
- Not providing examples when the task is unusual or pattern-based.
- Ignoring the context window — pasting irrelevant information that dilutes focus.

8. Practice Exercises

Try rewriting these weak prompts using the techniques from this guide:

- 1. "Write a poem." — Add role, topic, style, and length.
- 2. "Help me with my resume." — Add context (job title, industry) and format.
- 3. "Explain quantum computing." — Add audience level (5-year-old vs. graduate student).
- 4. "Give me marketing ideas." — Add product, target audience, and budget constraints.
- 5. "Fix my code." — Add language, error message, and desired behavior.

Tip: After rewriting each prompt, test it in ChatGPT, Claude, and Gemini and compare the results. This builds real intuition for how each model responds.

9. Quick-Reference Cheat Sheet

Goal	Technique to Use
Need step-by-step accuracy	Chain-of-Thought Prompting
Want consistent tone/style	Role/Persona Prompting
Task follows a pattern	Few-Shot Prompting
Complex multi-step project	Prompt Chaining
Need to avoid certain content	Negative Prompting
Want a well-rounded, structured prompt fast	RTF or CO-STAR Framework
Double-checking a tricky answer	Self-Consistency Prompting

Keep practicing! Prompt engineering is a skill that improves with deliberate experimentation. The more precisely you communicate your intent, the more powerful these tools become.