



# Prompt Engineering Best Practices

---





## Core Principles of Effective Prompting

### 1. Be Concise


- Avoid unnecessary details or flowery language.
  - Simple, direct prompts reduce the risk of misunderstanding.
- 

### 2. Be Specific and Well-Defined

- Clearly define the task or desired output.
- **Vague Prompt:**  
 “Tell me about Earth”
- **Improved Prompt:**  
 “Give me five poetic metaphors to describe Earth as a nurturing planet”
- **Why It Matters:**  
Specific prompts guide the model toward your intended purpose, reducing ambiguity and improving relevance and structure.



### Learning Tips:

- **State Intent Clearly:**  
 “Summarize Earth’s atmosphere for a 6th-grade science textbook.”
- **Avoid Overly Broad Prompts:**  
Useful for exploration, but not for goal-oriented tasks.



**Summary:** The more precisely you define what you want, the more accurately the model can respond.

---

### 3. Ask One Task at a Time

- Avoid multi-part or compound questions.
- Break complex tasks into sequential, focused queries.

#### Example Breakdown:

#### Poor Prompt:


“What's the best method of boiling water and why is the sky blue?”

#### Problems:

- Mixes unrelated topics
- Reduced depth in each answer
- Difficult to tailor tone or style

#### Better Prompts (Split):

- “What is the safest and most energy-efficient method to boil water at home, and why?”
- “Explain why the sky appears blue, in simple terms for a 10-year-old.”

 **Tip:** If your prompt feels like two separate questions — split them for clarity and depth.

---

### 4. Turn Generative Tasks into Classification Tasks

- Instead of “Generate X,” ask:

“Which of these X options is better?”

- **Benefits:**

- Greater control
  - Easier evaluation
  - Safer outputs
- 

## 5. Include Examples

- Use example inputs and outputs to guide the model's behavior.
  - Especially effective for:
    - Few-shot prompting
    - Maintaining response structure
    - Teaching style or tone
- 

## Technical Setup for Gemini / Vertex AI in Colab

### ◆ Environment Setup:

- **Installation:**  
Requires installing `google-genai` SDK.
- **Runtime Restart:**  
Necessary after installation to load dependencies.
- **Authentication:**  
Required via `google.colab.auth`.
- **Cloud Project Configuration:**  
Must set project ID and location to access Vertex AI models.
- **Model Loading Example:**  
Common model: `"gemini-2.0-flash-001"` via Vertex AI.



---

## Additional Prompt Engineering Techniques

Technique	Description	Use Case Example
<b>Zero-Shot Prompting</b>	No examples provided	Creative writing, open-ended ideation
<b>One-Shot Prompting</b>	One input/output pair provided	Response style imitation
<b>Few-Shot Prompting</b>	Multiple examples included	Improves consistency, structured answers

---

## Checklist for Effective Prompts

- Ask one question at a time
  - Use clear and specific instructions
  - Avoid hallucinations; encourage fact-checking
  - Use recommendation labels (, ) to judge outputs
  - Present responses in structured formats (tables, lists)
- 

Let me know if you'd like me to provide a downloadable Google Docs version or export this as a PDF!