

# Prompt Engineering Tutorial

Supported Models: GPT-4, DeepSeek, Claude, etc.

Lei Zhehao

April 21, 2025

## How to Use This Tutorial

Dear Reader:

This tutorial is designed to help you systematically master prompt engineering techniques. Whether you're just getting started with large language models or looking to improve your existing AI efficiency, you'll find value here.

**How to read this tutorial:**

- **Beginners:** We recommend reading Chapters 1-5 in sequence to grasp the basic concepts and principles
- **Practitioners:** Focus on Chapters 5-7 for practical cases and common pitfalls
- **Developers:** Chapters 8-9 on evaluation methods and industry applications will help you build real systems

The chapter notes at the beginning of each section will help you understand the practical value of that content.

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>What is Prompt Engineering</b>	<b>2</b>
2.1	Background and Importance . . . . .	2
<b>3</b>	<b>Four Core Principles of Prompt Design</b>	<b>3</b>
<b>4</b>	<b>Common Prompt Templates</b>	<b>3</b>
4.1	Zero-Shot Prompt . . . . .	3
4.2	Few-Shot Prompt . . . . .	3
4.3	Chain-of-Thought (CoT) . . . . .	4
4.4	ReAct Framework (Reasoning + Acting) . . . . .	4
4.5	Instruction + Output Template . . . . .	4

<b>5</b>	<b>Comprehensive Prompt Construction Frameworks</b>	<b>4</b>
5.1	ICIO Framework . . . . .	4
5.2	CRISPE Framework . . . . .	5
5.3	BROKE Framework . . . . .	5
5.4	APE Framework . . . . .	5
<b>6</b>	<b>Conclusion</b>	<b>5</b>

# 1 Introduction

*This chapter helps you understand why prompt engineering is so important and how it will change the way you interact with AI.*

In the era of large language models (LLMs), *prompt engineering* has become a core skill for efficiently using these models. With the rapid development of models like GPT-4, Claude, and DeepSeek, knowing how to design prompts that guide models to generate high-quality, controllable outputs has become a key bottleneck and competitive advantage for AI applications.

*Prompt engineering* is not just a technology but also an art, playing a crucial role in:

- **Bridging human-machine interaction gaps:** Helping non-technical users effectively use AI through standardized instructions
- **Maximizing return on investment:** Significantly improving model performance without increasing computational resources
- **Unlocking hidden capabilities:** Discovering the potential of large models in complex reasoning, creative tasks, and more
- **Reducing model risks:** Decreasing the probability of hallucinations, biases, and harmful outputs

**Prompt:** *You are an AI expert. Please explain what "prompt engineering" is in simple terms.*

**Output:** Prompt engineering is the art and science of communicating with large language models. It involves crafting carefully designed inputs (prompts) to guide models to produce high-quality outputs that meet your needs. It's like communicating with a smart but sometimes literal assistant who needs clear guidance.

Think of it as choosing the right bait and technique when fishing - good prompts let you get the most valuable "catch" from a model, rather than struggling with irrelevant results.

## 2 What is Prompt Engineering

*Mastering the definitions and background in this chapter will help you establish a clear understanding of prompt engineering.*

Prompt engineering refers to the **systematic design, optimization, and evaluation** of input text to obtain the desired output from large language models. It combines knowledge from linguistics, cognitive psychology, and machine learning, with the goal of maximizing the potential capabilities of models within limited context windows to achieve **more controllable, efficient, and accurate** human-machine interaction.

### 2.1 Background and Importance

- **Cost Reduction:** Good prompt design can reduce the number of iterations, lowering API call costs.

- **Improved Accuracy:** Clear, specific prompts significantly enhance the relevance and consistency of model outputs.
- **Facilitates Automation:** Structured prompts make programmatic generation and large-scale deployment easier.

### 3 Four Core Principles of Prompt Design

*These four principles form the foundation of all effective prompts. Applying them skillfully will significantly increase your success rate in getting the desired outputs.*

1. **Clear Role (*Role*):** Assigning a specific identity or perspective to the model helps guide its behavior and tone. For example:

You are an experienced Python programmer...

2. **Clear Task (*Task*):** Describe the operation you expect the model to perform in a specific, unambiguous way. Avoid vague instructions. For example, instead of saying "process this data," say:

Please calculate the average of the following numbers: [1, 2, 3, 4, 5]

3. **Sufficient Context (*Context*):** Provide necessary background information, examples (Few-Shot), or input data to help the model understand the task environment. For example, providing reference text for summarization:

Based on the following article [article content], generate a 100-word summary.

4. **Output Constraints (*Format*):** Specify the expected output format, length, style, etc., for easier parsing by downstream programs or to meet specific scenario needs. For example:

Please return the result in JSON format, including 'name' and 'age' as keys.

### 4 Common Prompt Templates

*These templates are your "prompt toolbox." Choosing the right template for different task types can make your work much more efficient.*

This section summarizes five commonly used templates in practice and provides minimum viable examples (MVPs).

#### 4.1 Zero-Shot Prompt

**Prompt:** *You are an English teacher. Please correct the grammar in the following sentence: "She go to school yesterday."*

**Output:** She went to school yesterday.

#### 4.2 Few-Shot Prompt

**Prompt:** *Classify the following sentiments as positive/negative:*

*Input: I really liked this movie! Output: positive*

*Input: This product is terrible. Output: negative*

*Input: The weather is nice today. Output: ?*

**Output:** positive

### 4.3 Chain-of-Thought (CoT)

**Prompt:** *Question: A farm has 5 chickens and 3 rabbits. How many legs are there in total? Please think step by step.*

**Output:** Thinking: Chickens have 2 legs each, while rabbits have 4 legs each. Calculation: 5 chickens  $\times$  2 legs = 10 legs. 3 rabbits  $\times$  4 legs = 12 legs. Total: 10 + 12 = 22 legs. Answer: There are 22 legs in total on the farm.

### 4.4 ReAct Framework (Reasoning + Acting)

The ReAct framework combines reasoning and action capabilities through a "Thought-Action-Observation" cycle to enhance the model's problem-solving abilities:

- **Thought:** The model analyzes the current state and formulates a solution
- **Action:** Performs specific operations such as searching for information, calculations, etc.
- **Observation:** Records the results of actions and incorporates them into subsequent thinking

### 4.5 Instruction + Output Template

**Prompt:** *Extract the person names and locations from the following text, and output them as a JSON list. Text: John went to New York yesterday, while Mary stayed in Boston.*

**Output:** { "persons": ["John", "Mary"], "locations": ["New York", "Boston"] }

## 5 Comprehensive Prompt Construction Frameworks

*This chapter introduces various professional prompt construction frameworks. Mastering these standardized frameworks will help you design effective prompts more systematically.*

### 5.1 ICIO Framework

**Framework Structure:** Input-Context-Instruction-Output

- **Input:** Provide the raw data or content to be processed
- **Context:** Provide background information and relevant knowledge
- **Instruction:** Clarify specific tasks and requirements
- **Output:** Specify output format and examples

## 5.2 CRISPE Framework

**Framework Structure:** Capacity-Request-Insight-Style-Purpose-Extra

- **Capacity:** Specify the AI's professional role and capabilities
- **Request:** Clarify specific task requirements
- **Insight:** Request in-depth analysis and insights
- **Style:** Specify the tone and style of the output
- **Purpose:** State the ultimate goal of the task
- **Extra:** Supplement with special restrictions or format requirements

## 5.3 BROKE Framework

**Framework Structure:** Background-Role-Objective-Key information-Exclusions

- **Background:** Background information for the project or requirement
- **Role:** Specify the role assumed by the AI
- **Objective:** Clarify the ultimate goal to be achieved
- **Key information:** Provide necessary facts and data
- **Exclusions:** Specify content that is not desired

## 5.4 APE Framework

**Framework Structure:** Approach-Persona-Execute

- **Approach:** Specify the methodology for solving the problem
- **Persona:** Set a professional role or perspective
- **Execute:** Specific execution instructions

# 6 Conclusion

Prompt engineering is evolving from an early collection of "tricks" into a systematic discipline. As large language model capabilities continue to advance and application scenarios expand, mastering prompt engineering has become a core competitive advantage in the AI era.

Through this tutorial, we've systematically introduced the methodology from **basic principles** to **advanced techniques**. It's important to note that prompt engineering is not static knowledge but a dynamically evolving practice. We encourage readers to:

- Build your personal prompt library, accumulating effective templates
- Stay informed about the latest developments in large models and adjust prompt strategies accordingly

- Join communities to exchange experiences and best practices
- Combine prompt engineering with domain-specific knowledge to create unique value

We hope this tutorial will help you seize opportunities in the AI era, fully unleash the potential of large models in research and production scenarios, and create greater social and economic value. Prompt engineering is not just technology, but a bridge connecting human intent with AI capabilities.

**Learning never ends, and trying is the best teacher.**

We wish you success in your prompt engineering journey!