PROMPT MASTERY

CRAFTING POWERFUL AI PROMPTS

Table of index

Prompt Mastery: Crafting Powerful AI Prompts

1. Introduction to Prompt Engineering

Introduction

Prompt engineering is the craft of designing, testing, and refining the text inputs (prompts) sent to conversational AI to produce reliable, useful, and safe outputs. It's both art and science: art because good wording often requires intuition and creative framing; science because measurable methods—testing, evaluation, and iteration—drive consistent improvement.

Why it matters: prompt engineering is the single most important lever for improving model output quality without changing code or models. A few carefully chosen words can mean the difference between hallucination and a precise, actionable answer.

(Visual suggestion: cover graphic showing a hand tuning dials labeled “clarity,” “context,” “constraints,” “persona,” and “testing” — annotate each dial with a 1–5 scale.)

Key concepts

- **Prompt:** The input (instructions, context, examples) given to an LLM.
- **Instruction following:** Models are trained to follow instructions; phrasing shapes adherence.
- **Iterative refinement:** Improve prompts by testing, measuring, and refining.
- **Safety & guardrails:** Prompts should reduce risk by specifying allowed behavior and constraints.

Poor vs. strong prompt examples

Bad prompt (vague):

> "Tell me about climate change."

Problems: No audience, no format, no length, no purpose.

Good prompt (specific):

> "You are an environmental science teacher preparing a 10-minute lesson for high-school seniors. Provide a 600–800 word explanation of climate change causes and effects with 3 bullet-point discussion questions and 2 suggested classroom activities. Keep language at a grade-10 reading level."

Why good: defines persona, audience, length, format, and purpose — all boost useful output.

Best practices (summary)

- Start with a clear instruction.
- Provide context and constraints.
- Show an example of the desired format.
- Iterate: test and refine.

Try it Yourself — Exercise 1

1. Take the bad prompt above. Rewrite it for three different audiences: a 10-year-old, a policy maker, and a scientist.
2. For each, list the constraints you added (length, tone, format). Compare outputs and note which constraints had the largest impact.

2. Understanding How Language Models Think

Introduction

To engineer effective prompts you must understand how large language models (LLMs) interpret text. This chapter explains tokenization, context windows, temperature, randomness, and how these model-level features interact with your prompts.

Key concepts

Tokenization

- Models operate on tokens (pieces of text). Tokens are usually smaller than words (subwords).
- Token counts matter: prompts and responses combined must fit in the model's **context window** (max tokens). When you run out of tokens, older context is truncated.

(Visual suggestion: a bar labeled "context window" with colored segments for "system prompt," "conversation history," "current prompt," "response.")

Context window

- The maximum number of tokens the model can attend to at once. Larger windows let you provide more reference material or longer conversations.

Temperature (randomness)

- **Temperature** controls randomness: lower values make outputs more deterministic and focused; higher values increase diversity and creativity. Use:
  - `temperature ≈ 0–0.3` for factual, precise answers.
  - `temperature ≈ 0.7–1.0` for creative writing or ideation.

Top-p / nucleus sampling

- `top_p` (nucleus sampling) restricts token choices to a cumulative probability mass. Lower values reduce randomness in a different way than temperature.

System / user / assistant roles

- In many chat APIs there are message roles (system, user, assistant). The **system** message sets persistent behavior (tone, persona, constraints). Use it to establish high-level guardrails.

How prompts guide reasoning and creativity

- Prompts are instructions + context + examples.
- *Instruction specificity* determines how narrowly the model follows a plan.

- *Examples and demonstrations* (few-shot prompting) teach the model desired formats or styles.

Bad vs. Good: temperature example

Bad (creative but uncontrolled):

> "Write a story about a space explorer." (temperature=1.0)

Better (controlled creativity):

> "Write a 700-word short story about a space explorer returning home, with emotional tension and a twist ending. Use vivid sensory details, but keep language accessible to a general adult audience." (temperature=0.7)

Best practices

- Track token usage and context window.
- Choose temperature/top_p per goal (deterministic vs creative).
- Use system prompts for persistent instructions.

Try it Yourself — Exercise 2

1. Generate three variations of a short review (positive, neutral, negative) using the same prompt but different temperatures (0.0, 0.5, 0.9).
2. Observe how determinism vs creativity changes the output. Report key differences.

3. Six Core Strategies for Better Results

Introduction

OpenAI and industry best practices converge on a small set of strategies that reliably improve results. These are practical, repeatable, and should become your default workflow. OpenAI frames several of these strategies in their prompt engineering guidance.

The Six Strategies

1. **Write clear instructions** — tell the model exactly what you want.
2. **Provide reference text** — give it grounding material for factual tasks.
3. **Split complex tasks into simpler subtasks** — reduce cognitive load for the model.
4. **Give the model time to think** — use chain-of-thought or reflection where appropriate.
5. **Use external tools** — retrieval, calculators, code execution to extend capability.
6. **Test changes systematically** — iterate, measure, compare.

(Visual suggestion: a flowchart showing a prompt lifecycle — design → test → measure → refine — with the six strategies mapped onto the flow.)

Examples & contrasts

- **Without reference text (prone to hallucination):** ask for specific stats on a niche topic without source.
- **With reference text (grounded):** include a short excerpt or URL to source text and ask the model to base answers on it.

Best practices

- Combine strategies: e.g., split a task and provide reference text for each subtask.
- Use tests and example-based evaluation to validate improvements.

Try it Yourself — Exercise 3

Pick a complex task (e.g., "create a marketing plan for a new herbal tea brand"). Apply the six strategies and produce the first draft. Document which strategy produced the largest qualitative improvement.

4. Writing Clear Instructions

Introduction