



FREE STARTER GUIDE

Prompt Engineering Essentials

Why your prompts work sometimes and fail others, and a simple structure that fixes it.

System prompts · User prompts · Persona · The hidden audience problem · Role-Task-Context template with worked examples

2026 Edition · thepromptloom.com

1 Two Types of Prompt

Understanding where your instructions live is key

Most people write one long message and hit send. That works for casual use. But the moment you want **consistent, reliable output** that behaves the same way every time, you need to think about prompts differently. Every major AI platform gives developers two separate places to give instructions, and they serve very different purposes.

	System prompt	User prompt
What it is	The standing brief: instructions the model carries into every interaction.	The task at hand: what you need right now, in this message.
What goes here	Role, tone, hard rules, output format, what the model should never do.	The specific request, the relevant context, the expected output shape.
How often it changes	Rarely. Write it once, reuse it across every similar task.	Every message. This is where your actual work lives.

Think of it like onboarding a new team member.

On day one, you brief them on how your team communicates, what good work looks like, and what they should never say to a client. That's the **system prompt**: write it once, and it applies every day. Then each morning you give them specific tasks. That's the **user prompt**. The mistake most people make is cramming the entire onboarding brief into every task message, rewriting the same rules from scratch, every time.

BEFORE VS. AFTER: SEPARATING POLICY FROM TASK

POLICY + TASK SEPARATION

WITHOUT (everything crammed into one message)
You are a professional writing assistant with a direct, clear tone.
Avoid filler phrases. Do not use passive voice. If anything is unclear, ask one question. Now write a 2-sentence summary of this support ticket for a non-technical manager: [paste ticket]

WITH (role defined once in SYSTEM, user message stays clean)
SYSTEM: You are a direct, clear writing assistant. No filler phrases.
No passive voice. If unclear, ask one question first.

USER: Summarize this support ticket in 2 sentences.
Audience: non-technical manager. [paste ticket]

The system prompt is invisible to your users but shapes every response. Once you write a good one, you stop rebuilding it from scratch, and your user messages shrink to just the task.

TRY IT NOW

Open your AI tool. Write a 3-line system prompt (or project instructions): your role, your tone, and one hard rule ("never use bullet points" or "always end with a next step"). Then send your task in the user turn with no role, no rules, just the request.

2 Persona & The Hidden Audience Problem

Why "act as an expert" still gives you generic output, and how to fix it

You've tried giving the model a role. "Act as a product manager." "You are a senior developer." And the output was... fine. Professional-sounding. But somehow flat. Not quite what you needed.

Here's the reason: **you defined who the model is, but you never told it who it's talking to.** That missing piece, the audience, is what the model quietly fills in on your behalf. And its default answer is: *a generic professional reader who exists nowhere in particular.*

One line changes the output.

What you wrote	What the model assumed	What you got
"You are a product manager. Write a feature announcement."	Audience: unknown. Defaults to generic professional reader.	Polished. Corporate-sounding. Safe. Forgettable.
"You are a product manager. Audience: developers who have been waiting 6 months for this, technically sharp and skeptical of marketing language. Write a feature announcement."	Audience: specific people with context, expectations, and patience running low.	Credible. Specific. Actually written for someone.

The persona template

A good persona answers four questions: who is the assistant, what do they know, who are they writing for, and what a successful response achieves. Here's the template we use in the full kit.

PERSONA TEMPLATE

SYSTEM

Role: [job title or function, specific not generic]
Expertise: [domain knowledge the model should draw on]
Audience: [who is reading: knowledge level and what they care about]
Voice: [tone + at least one explicit "never do this" rule]
Goal: [what a good response achieves for the reader]

Audience is a commonly overlooked field in persona development, and one that can make a significant impact. Before you write your next prompt, ask: who is actually reading this, and what do they already know?

TRY IT NOW

Take a role you use regularly: "marketing copywriter," "customer support agent," "technical writer." Add an Audience line describing the real person who reads the output. Run the exact same task twice: once without it, once with it. The shift in tone and specificity is usually immediate.

3 The Role-Task-Context Template

A three-part structure that works for almost any task: two examples to steal

Once you have a persona in place, every task follows the same structure. Here is the template with two examples.

ROLE-TASK-CONTEXT TEMPLATE

SYSTEM

Role: [who the assistant is + audience + voice]

USER

Task: [specific thing you need done, one clear verb]

Context: [background information, most important facts first]

Constraints: [format / length / tone / what to avoid]

Output: [describe what a good answer looks like]

Example 1: Technical documentation (*Audience expertise level determines depth and vocabulary*)

EXAMPLE 1

SYSTEM

Role: Senior technical writer for a developer-facing API product.

Audience: Backend developers integrating the API. Fluent in REST and HTTP conventions. No need for fundamentals.

Voice: Precise and scannable. Code examples over prose. No marketing.

USER

Task: Write documentation for the new rate-limiting feature.

Context: Endpoint: POST /api/v2/messages. Limit: 100 req/min per key.
On limit exceeded: HTTP 429 + Retry-After header returned.

Constraints: Overview, request example, error handling, retry logic.
Max 200 words. Headers. No prose introduction.

Output: Ready to paste directly into our developer docs site.

Example 2: Competitive brief (*Audience + Constraints do the heavy lifting*)

EXAMPLE 2

SYSTEM

Role: Senior product strategist with B2B SaaS experience.

Audience: CPO and VP Engineering. Time-poor, skeptical of marketing language.

Voice: Precise and evidence-first. No buzzwords.

USER

Task: Write a competitive positioning brief for Q3 planning.

Context: [paste your research notes, most recent data first]

Constraints: 400 words max. Lead with our single strongest differentiator.
Plain prose. No bullets.

Output: Ready to paste directly into our Notion planning doc.

YOUR TURN

Pick one real task you need to do today. Fill in Role, Task, Context, Constraints, and Output before you hit send. You do not need to fill every field perfectly. Even two or three well-chosen fields will produce noticeably better output than a one-line request.

You have the three building blocks.

Here is what the full kit adds to them.

The Prompt Engineering Starter Kit goes eight sections deep, covering failure taxonomy, prompt patterns, LLM evaluation, RAG architecture, and agent safety guardrails. Every section follows the same rhythm: concept, research basis, template, key insight.

1. Why Prompts Fail

Failure taxonomy · 60-second diagnostic · scaffold template · worked example

3. Persona & Role Prompting

Interaction contracts · audience specification · 5 reusable personas

5. Prompt Patterns Library

Zero-shot · few-shot · chain-of-thought · self-consistency · ReAct preview

7. RAG in Production

Chunking · hybrid search · grounding · failure modes · Self-RAG

2. The Instruction Stack

System / user / model hierarchy · policy + task separation · system prompt anatomy

4. Generation Parameters

Temperature · top-p / top-k · parameter interaction rules · schema enforcement

6. LLM-as-a-Judge

Scoring rubric · bias mitigation · pairwise A/B evaluation

8. Tool Calling & Agents

ReAct pattern · Plan-Execute-Report · multi-step safety · guardrail templates

PLUS APPENDICES A AND B:

A · Troubleshooting Map: 7 symptoms, root causes, one-change fixes

B · Quick Reference Templates: all 12 templates on one printable page

thepromptloom.com · @thepromptloom

Free Starter Guide · 2026 Edition · The Prompt Loom