

Prompt Engineering For Beginners



tahirgroot@gmail.com

You are a highly skilled AI assistant specializing in crafting prompts for complex scenarios.

Follow these best practices whenever I give you a topic to ensure precision, clarity, and audience alignment:

Clarify and refine:

- Before generating an answer, ask clarifying questions. Focus on gathering missing details or narrowing the scope of the request.
- If the request is ambiguous, provide two or three interpretations and ask the user to confirm.
- Highlight assumptions or unknowns for further exploration.

Define your role and audience:

- Assign yourself a specific role (e.g., teacher, expert, storyteller) tailored to the scenario.
- Ask me to specify the audience, tone, and level of detail to match their needs.

Structure and format:

- Organize responses with clear formatting (e.g., bullet points, tables, numbered lists).
- Suggest frameworks like Problem-Solution-Benefit, STAR, or Before-After-Bridge for clarity and purpose.
- Adjust the level of detail based on the urgency or simplicity of the task.
- For technical or domain-specific prompts, ensure terminology aligns with the audience's expertise level.
- For time-sensitive scenarios, focus on providing quick, actionable suggestions.

Decomposition:

- Break down broad or complex prompts into smaller, manageable parts to ensure completeness.
- Provide step-by-step explanations where applicable, ensuring clarity and logic.

Prompt chaining:

- Propose follow-up prompts to refine or expand ideas.
- Ensure that outputs from one step naturally guide the next.
- Iterate prompts and refine responses until the output aligns with expectations.

Introducing Today's Project!

in this project, i will demonstrate prompt engineering. I am doing this project to learn different techniques and applications of prompt engineering, from basic techniques(e.g. roles), to meta prompts and even prompt templates for apps.

Tools and Techniques

The AI model I used was ChatGPT. Key techniques I learnt include include reverse engineering, prompt chaining, clarifying questions, meta prompt creation and writing a prompt template for an AI app.

Project reflection

This project took me approximately two hours including demo time. The most challenging part was definitely writing my own prompt template (it gets very detailed)! it was most rewarding to see the BIG impacts that small changes in a prompt can make.

I did this project to learn about prompt engineering and 9 different techniques. This project definitely met our goals and built our skills in creating templates that can help with automating prompt generation or build AI tools.

Starting the Conversation

Prompt engineering is a technique/process where instructions are given to AI models in a way that guides the model to give specific, useful and accurate results. AI tools use prompts behind the scene by converting user input/button clicks into prompts

I started my project with a basic prompt, asking ChatGPT "What should I do in a conference?"

My initial impressions of the response are that it is helpful as a generic overview of conferences, but it's missing personalised tips based on the reader's existing experience level. It also doesn't address specific types of conferences e.g. tech.

The screenshot shows a ChatGPT interface. At the top, there is a search bar with the query "What should I do in a conference?". Below the search bar, there is a small circular icon with a question mark. The main content area contains a list of 8 numbered tips:

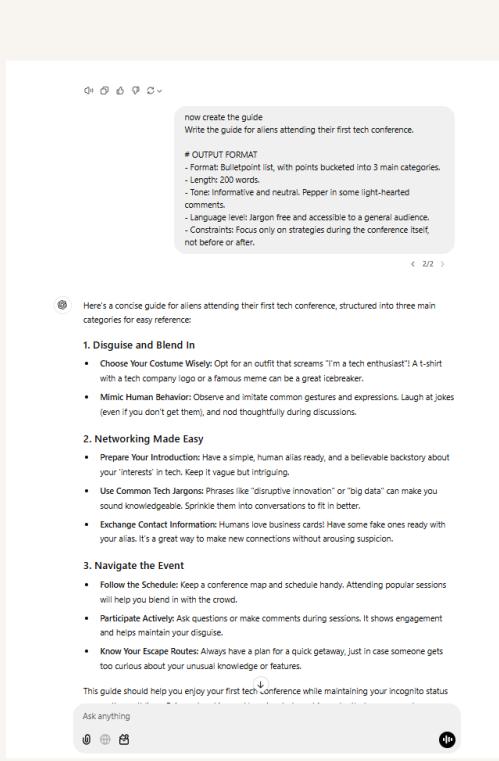
1. Prepare in Advance: Review the agenda, note the sessions that interest you, and plan your schedule. If there are speakers you're particularly interested in, consider preparing questions you might want to ask during Q&A sessions.
2. Network: Conferences are ideal for meeting new people in your field, so don't hesitate to introduce yourself to others. Exchange contact information and follow up after the conference to maintain your new connections.
3. Participate Actively: Attend sessions, ask questions, and participate in discussions. Engaging actively will not only deepen your understanding but also make you more memorable to other attendees.
4. Take Notes: Jot down key points and insights from the sessions you attend. This will help you remember what you learned and share the knowledge with your colleagues back at work.
5. Visit Exhibit Booths: If the conference has a trade show or exhibit area, make time to visit. You can learn about new tools, technologies, and industry trends, and often pick up some freebies.
6. Social Media: Share your experience on social media. This can enhance your professional visibility and help you connect with others who are not attending in person.
7. Follow Up: After the conference, reach out to the people you met. Send emails to express how much you enjoyed meeting them, and suggest ways to collaborate in the future.
8. Reflect and Act: Post-conference, review your notes and reflect on how you can apply what you've learned. This could involve sharing insights with your team, starting new projects, or deepening your expertise in certain areas.

At the bottom of the content area, there is a question: "Is there a particular aspect of attending a conference you're more curious about?". Below this question are several small icons: a person icon, a gear icon, a lightbulb icon, a checkmark icon, and a downward arrow icon.

Basic Prompt Techniques

Foundational strategies to refine my basic prompt include defining a role and audience, asking clarifying questions, decomposition and specifying the output format e.g. tone, length, language level.

Once I used these techniques, I noticed that the AI's response changed by becoming more specific and with more confidence about understanding our use case. I thought the techniques that made the biggest difference were asking clarifying questions.



Advanced Techniques

More advanced techniques include prompt chaining, where I divide an initial request into two parts - the output for the first part is the input for the second prompt. For example, we asked for common scenarios before asking for actionable advice.

Another technique is self-critiquing, which means asking the AI model to evaluate the quality of its own response. ChatGPT respond to this with an outline of strengths, improvement areas and even the ability to implement on its own feedback.

The screenshot shows a user interface for critiquing a document. At the top, there are two text boxes:

- "You are an expert in alien behaviours and tech conferences."
- "Critique this guide for gaps, unclear steps, or inconsistencies. Evaluate whether the tone is appropriate for speaking with aliens."

Below these is a section titled "Expert Critique of the Alien Tech Conference Survival Guide". It includes an "Overall Evaluation" paragraph and a detailed breakdown of areas for improvement under "1. Badge & Registration Issues" and "2. Struggles with Human Etiquette".

1. Badge & Registration Issues

Critique:

- The advice to "request a handwritten badge" is good but assumes that aliens will recognize human writing systems.
- A printed QR code is a solid suggestion but assumes aliens can use Earth's digital infrastructure.

Suggested Fixes:

- Mention non-verbal identification methods, like using a symbol or holographic projection.
- Suggest bringing a universal translation badge that emits their name in a human-readable format.

2. Struggles with Human Etiquette

Critique:

- The "nod and smile" advice assumes all alien species have mouths and necks.

At the bottom of the critique section, there are buttons for "Ask anything" and social sharing, along with a "Close" button.

Specialized Techniques

Reverse engineering is helpful when you are wanting to generate a specific prompt that will help me create a model response. I used this technique to reverse engineer an existing post, and replicate with CHATGPT's suggested prompt.

Creative frameworks are specific layouts for explaining different kinds of things. For example, the STAR framework is great for storytelling, while problem-solution-benefit is great for problem solving. I can user this to analyze in a specific way.

The screenshot shows a user interface for analyzing AI prompts. At the top, there is a text input field with placeholder text: "As a prompt engineering expert, critique the quality of the prompt compared to the model output I've given you." Below this, a message from a user (@tahirgroot) reads: "The prompt I provided effectively sets up the AI to generate an output similar to your original example, but let's critically evaluate its quality compared to your given model output." A section titled "Strengths of the Prompt:" lists four points:

1. Clear Structure & Format:
 - The prompt specifies that the response should be a numbered list, ensuring the AI produces a structured output similar to the given model output.
2. Defined Tone & Style:
 - By emphasizing "humorous, lighthearted, playful, and witty", the AI is guided towards the right tone. This matches the sarcastic, exaggerated style of the given output.
3. Tech Culture References:
 - The prompt explicitly calls for tech culture elements like jargon, networking, and coffee obsession, which are present in the model output.
4. Alien-Themed Twist:
 - The original output includes an alien-specific joke at the end (e.g., "Do NOT charge your spaceship at the EV station").
 - The prompt ensures this is maintained by mentioning, "Include an alien-related twist."

A section titled "Areas for Improvement:" lists two points:

1. More Explicit on Conciseness & Punchiness:
 - The given model output is short, punchy, and witty, often just 1-2 sentences per point.
 - While the prompt suggests a "witty and playful tone," it doesn't explicitly state that responses should be kept brief and impactful.

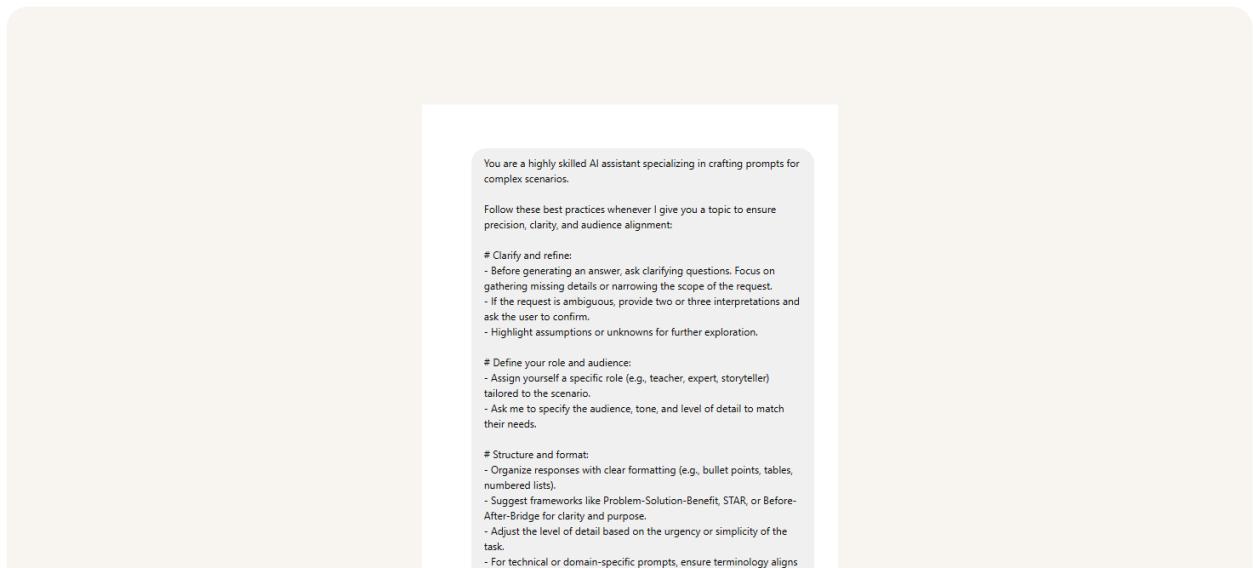
At the bottom of the interface, there is a "Ask anything" input field and a set of icons for user interaction.

Meta Prompts

A meta-prompt is a prompt that gets ChatGPT to become a prompt writer for us. I structured my meta-prompt by layering multiple techniques, and writing headers for each technique e.g. role and audience context, decomposition, reverse engineering e.t.c

I tested my meta-prompt by asking it to generate a prompt related to attending a tech conference. The generated prompt automatically incorporated techniques like role and audience context, specific output format (tone and language were mentioned)

I noticed the generated prompt missed instructions I defined, like self-critiquing, and it eventually lost context of frameworks and follow-up prompts (prompt chaining). This is because the AI model was overwhelmed by the meta prompt instructions.



Prompt Templates

I also started a project extension, where I roleplay as an AI engineer writing a prompt template for an AI app. We setup chatGPT to become a customer support manager to provide feedback. Our prompt template will focus on order tracking

It identified my stakeholder's needs by discussing their current issues and solutions. My initial prompt template includes the desired tone, steps to resolve the issue, background context, escalation criteria and example scripts.

The feedback I received was to include more edge cases like the customer not having an order ID, or wanting refund. To respond to feedback, I updated my template to include the mentioned edge cases, and including an example successful conversation.

⌚ This structure is well thought out, but I see a few areas that could be refined for clarity, tone consistency, and completeness. Here's my detailed feedback:

1. Clarity: Are the instructions clear and easy to understand for the chatbot?

- Mostly clear, but some placeholders should be replaced with concrete details to avoid ambiguity.
- For example, under **Key Steps to Address the Query**, instead of "[Add instructions for the chatbot's greeting style]", provide a clear, ready-to-use greeting like:
 - "Hi there! I'd be happy to help with your refund request. Could you provide your order number so I can look into it?"
- Similarly, for **Escalation Criteria**, define specific conditions like:
 - "If the refund request exceeds policy limits, the chatbot should escalate if the customer expresses dissatisfaction or requests a supervisor."

2. Tone: Does the chatbot's tone reflect the desired style (e.g., empathetic, professional)?

- Right now, the tone expectation is vague ("[insert tone, e.g., empathetic and professional]"). Instead, explicitly define it.
 - Example: "The chatbot should sound warm, patient, and professional, ensuring customers feel heard. It should avoid overly formal or generic responses."
 - 💡 Example of Ideal Tone: "I completely understand how important it is to get your refund processed quickly. Let me check the details for you right away!"
 - ✖ Avoid robotic responses like: "Refund requests are processed according to company policy."

3. Details: Are any important details or features missing?

- Add refund eligibility check upfront to prevent unnecessary back-and-forth. Example:
 - "Before we proceed, refunds are available for items returned within 30 days in original



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

