

Similar tests, such as RULER (Hsieh et al., 2024), can also be used to evaluate how good a model is at processing long prompts. If the model's performance grows increasingly worse with a longer context, then perhaps you should find a way to shorten your prompts.

System prompt, user prompt, examples, and context are the key components of a prompt. Now that we've discussed what a prompt is and why prompting works, let's discuss the best practices for writing effective prompts.

Prompt Engineering Best Practices

Prompt engineering can get incredibly hacky, especially for weaker models. In the early days of prompt engineering, many guides came out with tips such as writing "Q:" instead of "Questions:" or encouraging models to respond better with the promise of a "\$300 tip for the right answer". While these tips can be useful for some models, they can become outdated as models get better at following instructions and more robust to prompt perturbations.

This section focuses on general techniques that have been proven to work with a wide range of models and will likely remain relevant in the near future. They are distilled from prompt engineering tutorials created by model providers, including [OpenAI](#), [Anthropic](#), [Meta](#), and [Google](#), and best practices shared by teams that have successfully deployed generative AI applications. These companies also often provide libraries of pre-crafted prompts that you can reference—see [Anthropic](#), [Google](#), and [OpenAI](#).

Outside of these general practices, each model likely has its own quirks that respond to specific prompt tricks. When working with a model, you should look for prompt engineering guides specific to it.

Write Clear and Explicit Instructions

Communicating with AI is the same as communicating with humans: clarity helps. Here are a few tips on how to write clear instructions.

Explain, without ambiguity, what you want the model to do

If you want the model to score an essay, explain the score system you want to use. Is it from 1 to 5 or 1 to 10? If there's an essay the model's uncertain about, do you want it to pick a score to the best of its ability or to output "I don't know"?

As you experiment with a prompt, you might observe undesirable behaviors that require adjustments to the prompt to prevent them. For example, if the model outputs fractional scores (4.5) and you don't want fractional scores, update your prompt to tell the model to output only integer scores.