We can use LLMs to refine prompts for better accuracy and usefulness. Clear instructions and detailed context lead to more precise responses. Anthropic’s experts share 15 key tips for crafting effective and reliable prompts.

1. **Communicate clearly and precisely** when writing prompts. The ability to clearly state tasks and describe concepts is crucial.
2. Be willing to iterate rapidly, sending many prompts to the model in quick succession. Good prompt engineers are comfortable with **constant back-and-forth refinement**.
3. **Consider edge cases** and unusual scenarios when designing prompts. Think about how your prompt might fail in atypical situations.
4. **Test your prompts with imperfect, realistic user inputs**. Don’t assume users will provide perfectly formatted or grammatically correct queries.
5. Read and analyze model outputs carefully. **Pay close attention to whether the model is following instructions as intended**.
6. Strip away assumptions and clearly communicate the full set of information needed for a task. **Break down the task** systematically to ensure all necessary details are included.
7. Think about the “theory of mind” of the model when writing prompts. **Consider how the model might interpret your instructions** differently than intended.
8. **Use version control and track experiments** when working with prompts. **Treat prompts like code** in terms of management and iteration.
9. **Ask the model** to identify unclear parts or ambiguities in your instructions. This can help refine and improve your prompts.
10. Be precise without overcomplicating. **Aim for clear task descriptions without building unnecessary abstractions**.
11. **Consider the balance between typical cases and edge cases**. While handling edge cases is important, don’t neglect the primary use case.
12. Think about how prompts integrate into larger systems. **Consider factors like data sources, latency, and overall system design**.
13. Don’t rely solely on writing skills; **prompt engineering requires a mix of clear communication and systematic thinking**. Good writers aren’t necessarily good prompt engineers, and vice versa.
14. When working with customers, help them understand the realities of user input. **Guide them to consider** **real-world usage patterns** rather than idealized scenarios.
15. **Practice looking at data and model outputs extensively**. Familiarize yourself with how the model responds to different types of prompts and inputs.

In-context learning lets a model use examples within a prompt to perform tasks without extra training. Few-shot prompting is a type of in-context learning that provides a small set of examples (usually 2–5) to help the model generalize and handle complex tasks. Unlike zero-shot prompting, which gives no examples, few-shot prompting improves performance by showing input-output pairs as guides, allowing the model to learn patterns and apply them to similar new tasks.

**Book Reference**: <https://learning.oreilly.com/library/view/building-llms-for/9798324731472/index_split_015.html>