20 Prompt Engineering Best Practices for Better Outputs

1. Repeat the request at the end - reinforces and sharpens the model's focus.
2. Use chained reinforcement - restate key criteria with varied wording throughout.
3. Use embedded examples without labels - naturally prime the model with implicit patterns.
4. Use sentence fragments for layout cues - guide structure with short directive phrases.
5. Use intentional redundancy for logic-heavy tasks - reinforce expectations using synonyms.
6. Use [brackets] or ::colons:: to flag variables - enhances clarity for parsing.
7. Ask for a confidence or priority score - encourages evaluative thinking, not just listing.
8. Use temporal anchors - make answers more relevant by tying them to timeframes.
9. Use 'If...then' logic - guide fallback decisions or responses when data is missing.
10. Bookend your tone or format - reinforce style both at the start and end of the prompt.
11. Add a post-prompt disclaimer - e.g., 'Don't hallucinate facts' to reduce made-up data.
12. Use a system prompt frame - e.g., 'You are an expert in X. Your job is to...' for authority.
13. State what *not* to do - adding a line like 'Don't include definitions' prevents distractions.
14. Break prompts into mini-tasks - model handles step-by-step better than bundled tasks.
15. Use pseudo-formatting - like 'Name | Benefit | Audience' to imply structured output.
16. Mention the desired reading time - e.g., 'Give me a summary that takes 30 seconds to read.'
17. Use emotion tags - e.g., 'Make the tone feel like relief, not hype' to guide nuance.
18. Mention what your audience already knows - avoids generic outputs by narrowing scope.
19. Set max length per section - e.g., 'Each point should be under 20 words.'
20. End with 'Review before answering' - nudges the model to self-check and improve output.