Metric Coders Prompt Engineering Cheat Sheet

Created by Suhas Bhairav

1. Start with the End Goal – clarify output format, tone, and style.

2. Use Clear Role Instructions – frame the model’s role explicitly.

3. Break Down Complex Tasks – split into smaller, sequential steps.

4. Specify Output Constraints – define word count, format, structure.

5. Leverage Few-Shot Examples – guide the model with 1–3 examples.

6. Use Step-by-Step Reasoning – request thought process, then final answer.

7. Avoid Ambiguity – replace vague terms with specific instructions.

8. Control Verbosity – state ‘concise’ or ‘elaborate’ as needed.

9. Use Structured Outputs – prefer JSON or Markdown for parsing.

10. Add Context Continuously – carry context/history across prompts.

11. Specify the Target Audience – tailor complexity and tone.

12. Force Validation – ask the model to verify its output.

13. Use Guardrails – limit scope with explicit constraints.

14. Chain Prompts for Complex Workflows – plan → generate → refine.

15. Control Creativity Levels – choose factual vs. imaginative outputs.

16. Enforce Style Consistency – define style guides for uniformity.

17. Ask for Multiple Variations – generate several options for testing.

18. Add Negative Instructions – clarify what to avoid.

19. Tune Temperature & Sampling – balance creativity and accuracy.

20. Iterative Refinement – feed outputs back for improvement.

21. Combine with RAG/APIs – enhance with external knowledge.

22. Prevent Hallucinations – instruct the model to stay factual.

23. Test with Edge Cases – validate reliability with unusual inputs.

24. Benchmark and Iterate – A/B test prompts for performance.

25. Document & Modularize – treat prompts as reusable assets.