

Ourbit Vendor-Neutral AI - Day 3: Prompting as Clear Instruction

Day 3 - Prompting as Clear Instruction

Programme: Vendor-Neutral AI - 15-Day Certification Programme

Purpose: Treat prompting as instruction design: communicate the task, relevant context and success conditions clearly enough that output can be evaluated.

Learning outcomes

- Turn a vague request into a clear instruction.
- Use task, context, audience, constraints, output format and success criteria deliberately.
- Distinguish useful constraints from unnecessary complexity.
- Recognize when missing information should trigger a question rather than a guess.
- Evaluate output against the instruction instead of accepting it because it sounds polished.

1. Prompting is specification

A prompt is not magic wording. It is a working specification for what you want the system to do. The clearer the specification, the easier it is to judge whether the result succeeded. This principle survives changes in models, vendors and interfaces.

2. Six useful components

Task: the action to perform. Context: information necessary to do it well. Audience: who will use the result. Constraints: boundaries such as scope, length, sources or prohibited assumptions. Format: the shape of the output. Success criteria: what a good result must accomplish.

Not every prompt needs all six components. Add what materially reduces ambiguity. Excessive instructions can conflict or distract just as missing instructions can cause guessing.

3. Separate facts from assumptions

If the model needs information you have not supplied, decide whether it may make a clearly labelled assumption, should ask a question, or must stop. For consequential work, silently filling a gap can be dangerous. A useful instruction might say: "If a required fact is missing, identify it rather than inventing it."

4. Make success testable

"Make this better" does not define better. "Rewrite this explanation for a non-technical reader in under 180 words; preserve all stated facts; define the two technical terms; do not add unsupported claims" gives both the model and reviewer something concrete to work against.

Worked example - vague to testable

Vague: "Write about our project."

Improved specification: "Create a 150-word progress summary for a non-technical stakeholder. Use only the supplied project notes. Cover completed work, the current blocker and the next approved action. Do not infer dates or completion percentages that are not in the notes. Use three short paragraphs."

The improved version does not rely on secret prompt tricks. It clarifies purpose, evidence boundary, audience, content requirements and format.

5. Prompting does not replace evaluation

A strong instruction reduces avoidable failure but cannot guarantee correctness. After generation, compare the result with the success criteria. Check facts, omissions, tone, assumptions and constraints. If the output fails, diagnose why before changing the instruction.

Guided practice

Rewrite this request: "Make me a good comparison of these options." Define the decision purpose, audience, comparison criteria, evidence boundary, treatment of missing information and desired output structure. Then identify one instruction you intentionally did not add because it would not improve the task.

Knowledge check

- Why is prompting better understood as instruction design than as finding magic words?
- Which six components can reduce ambiguity?
- When should a model ask rather than assume?
- Why are success criteria useful?
- Why must output still be evaluated after a strong prompt?

Applied evidence task

A colleague gives an AI assistant the instruction "Analyze our customer feedback and tell us what to do." Produce a safer, testable instruction. You must specify the available evidence, what counts as a theme, how uncertainty is handled, what the AI may recommend, what it must not infer, and the output format. Explain two design choices.

Day 3 takeaway

Good prompting is clear communication plus measurable expectations. State what matters, provide relevant context, constrain real risks, define success, handle missing information explicitly and evaluate the result.

Assessment boundary

This day contributes only to vendor-neutral-ai-15-day. It teaches transferable instruction design rather than commands unique to one vendor.

Learning comes before credentials. Completing this reading alone does not issue a certificate. Programme assessment and practical requirements apply.