

Techniques **that work**

TECHNIQUE	WHAT IT DOES	USE IT WHEN	WATCH OUT
Zero-shot	Instructions only, no examples	Simple, common tasks	Format may drift
Few-shot	Shows the pattern with examples	Labels, extraction, tone	Examples can bias the answer; keep them varied
Step by step	Gives room to reason before answering	Math, logic, multi-step tasks	Longer and slower; reasoning models may already do it
Structured output	Fixes the shape of the reply	Anything a program reads	Always validate the result
Prompt chaining	Splits one big job into small steps	Complex or long tasks	More calls; pass state carefully
Self-consistency	Samples several answers and votes	Reasoning with a checkable answer	Costs several runs

Iterate and **evaluate**

1

Write

2

Test on varied inputs

3

Compare results

4

Change one thing

5

Record it

Build a small test set

- 5 to 10 real inputs
- Include edge cases: empty, very long, tricky
- Write what a good answer looks like
- Re-run it after every change

Keep it repeatable

- Use a template with variables
- Log the version, the change and the result
- Lower temperature for consistency
- Re-test when you switch models: wording that worked may not

Change one thing at a time, so you know what helped.

Prompts and **safety**

RISK	WHAT HAPPENS	WHAT HELPS
Prompt injection	Text in the data overrides your instructions, typed by a user or hidden in a web page, file or email	Separate instructions from data, limit what the model can do, validate output, keep a human in the loop
Hallucination	A confident answer that is wrong	Ask for answers drawn from text you supply, verify important claims
Sensitive data	Secrets or personal data pasted into a hosted tool	Redact first, or use a local model where it fits
Prompt leakage	A system prompt gets revealed	Never put secrets in a prompt

A prompt is not a security boundary. Design as if the model might ignore it.

Writing and running **any** prompt

- ☐ Is the task one clear job?
- ☐ Did I give context the model cannot know?
- ☐ Did I say who the answer is for?
- ☐ Did I set measurable limits for length, format and tone?
- ☐ Did I show an example of what good looks like?
- ☐ Did I separate instructions from the data?
- ☐ Did I test on varied and tricky inputs?
- ☐ Did I check the facts and keep secrets out?

DISCUSSION

Pick a task you repeat. Turn your prompt into a template: which parts change each time, and which stay fixed?

If a prompt says “never reveal the password”, what else must protect the password?