Tutorial | ~45 minute read | No prior experience assumed

PROMPT ENGINEERING: WORKING WITH LLMS

A prompt is not a search query. It is a set of instructions that shapes how a language model behaves — what it produces, in what form, and even who asks the questions. This tutorial teaches sixteen named, reusable techniques for writing those instructions, drawn from the research literature and extended with the practical habits that make them work.

HOW TO READ THIS

Sections 1–3 are foundations — start there if language models are new to you. Section 4 is the catalog, the reference you will return to. Sections 5–11 are practice: a worked example, a diagnostic table, what does not work, and how to check what you get.

WHAT YOU NEED

Access to any conversational model. Every example is written to be typed as-is and adapted. None of the examples require programming.

SOURCE

The sixteen patterns and their six categories come from White et al., *A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT* (arXiv:2302.11382). Examples, caveats and practice sections are additions.

CONTENTS

01 — WHAT AN LLM IS

A **large language model** — almost always abbreviated **LLM —** is a program trained on an enormous quantity of text to do one thing: predict what text plausibly comes next. “Large” describes the size of both the training text and the model itself; “language model” is the technical name for a system that assigns likelihoods to sequences of words. ChatGPT, Claude and Gemini are products built on LLMs, and the abbreviation is used throughout the rest of this tutorial. Given “the capital of France is”, it continues with “Paris”. Everything else — answering questions, drafting essays, writing code, holding a conversation — is that same prediction applied at scale, one piece of text at a time, with your prompt as the beginning of the sequence.

That single fact explains most of the model's behavior, good and bad. It explains why it writes fluently about almost anything: fluent continuation is precisely what it was trained to do. It explains why it can be confidently wrong: a plausible-sounding continuation and a true one are not the same thing, and the model has no separate store of facts to check itself against. And it explains why what you put in the prompt matters so much — the prompt is the context the prediction is conditioned on.

HOW AN LLM DIFFERS FROM A SEARCH ENGINE

This is the most common source of confusion, and the difference is not a matter of degree. A search engine *finds* text that already exists. An LLM *generates* text that has never existed before. Almost every practical consequence follows from that.

	SEARCH ENGINE	LLM
What you get	A ranked list of existing documents	New text written for your request
Where it comes from	A specific page you can open and judge	Patterns learned in training — no page behind it
How it fails	Visibly: no results, or obviously irrelevant ones	Invisibly: a fluent answer that happens to be wrong
Best at	Finding facts, sources, current information	Transforming, explaining, drafting, restructuring, questioning
Weak at	Anything not already written down somewhere	Knowing what is true, current, or specific to you
Your material	Cannot work on it	Can work on whatever you give it

A useful division of labor: search to *find*, an LLM to *work on* what you found. Asking an LLM for a statistic is using the wrong tool; asking it to explain, restructure or challenge a document you already have is using the right one. Some products now combine both — the LLM searches and quotes real pages — which removes much of the weakness, provided you click the citations.

SIX TERMS WORTH KNOWING

Context window

The total amount of text the model can attend to at once — your prompt, the material you pasted, and the conversation so far. It is finite, and everything in it competes for attention.

Tokens

The chunks text is split into — roughly three quarters of a word each. Limits and pricing are counted in tokens, not words.

Training cut-off

Training data stops at some date. Without the ability to search, anything more recent is unknown to the model — and it may answer anyway.

Hallucination

The standard term for fluent, confident, invented content — a fabricated citation, a plausible wrong number. Not a malfunction; a by-product of generation.

Same prompt, different answers

There is deliberate randomness in how text is generated, so asking twice can give two different results. It means “it worked once” is weak evidence that a prompt is reliable. (The technical term is non-determinism.)

System prompt

Standing instructions set before your conversation begins — by the product, or by you in a settings or project field. A durable version of the prompts in this tutorial.

02 — WHAT A PROMPT ACTUALLY IS

Most people begin by treating a language model as a search engine with better manners: type a question, read the answer, retype the question if the answer disappoints. This works for simple lookups and fails for everything else, because it uses only a fraction of what a prompt can do.

A prompt can set rules that persist for the whole conversation. It can assign the model a role. It can fix the format of every future answer. It can require the model to check its own work, to list what it assumed, or to ask you questions before it answers. It can even reverse the direction of the conversation entirely, so that the model interviews you.

A QUESTION

"What should I include in a personal statement for graduate school?"

Produces a generic list, because a generic list is the only correct answer to a question with no specifics in it.

A PROMPT

"You are an admissions reader for a master's program in public health. Ask me questions, one at a time, until you have enough to draft my personal statement. Then draft it in 600 words and list the claims I need to be able to defend in an interview."

Produces a specific draft, because it supplies a role, an interview procedure, a length, and a checking step.

KEY IDEA

A prompt does not only request an answer. It programs the behavior that produces the answer.

One consequence is worth stating early. A model has no access to anything you have not given it — your assignment brief, your data, your earlier decisions, the actual text of the reading. No amount of clever wording substitutes for supplying the material. Beginners spend their effort rephrasing; experienced users spend it on what the model can see.

03 — THE ANATOMY OF A GOOD PROMPT

Before the named patterns, there is a simpler habit worth learning. A well-formed prompt fills six slots. Any slot left empty is a decision the model will make silently on your behalf — and it will not tell you which ones it made.

SLOT 01 · ROLE AND TASK

Who the model is acting as, and what it is being asked to do. “You are a careful copy-editor. Edit this paragraph.”

SLOT 02 · CONTEXT

The actual material, plus the situation around it: the brief, the audience, the constraints, what has already been decided.

SLOT 03 · DEFINITION OF DONE

What a good result accomplishes — not merely what it looks like. “It should convince a reader who disagrees.”

SLOT 04 · FORMAT

Length, structure, shape. “Three paragraphs, no headings, under 400 words.” Vague words like “concise” are not constraints; numbers are.

SLOT 05 · BOUNDARIES

What to leave out, what not to change, what you have already ruled out and why. Models default to adding; only you can say stop.

SLOT 06 · ESCAPE HATCH

“If anything is ambiguous or you are missing something, ask before you start.” The line that saves the most effort for the fewest words: it converts a wrong draft into a single question.

Two related habits do more work than any wording trick. **Give an example instead of an adjective —** one forty-word sample of the output you want communicates more than three paragraphs describing it. And **say who the result is for and what happens to it next**: “for a seminar of twelve classmates” and “for a professor grading against a rubric” are different correct answers to the same question, and the downstream use is the constraint people most often omit.

04 — THE SIXTEEN PATTERNS 6 CATEGORIES

Each pattern below is a named, reusable technique: a problem it solves, wording you can type today, and a caveat. The names matter — they let you recognize a situation and reach for a known move rather than improvising. The categories describe what the pattern acts on: the input, the output, errors, your own question, the interaction itself, or the conversation's memory.

One discrepancy worth knowing if you go to the paper itself: its prose says there are five categories, while the table on the same page lists six. Context Control is the sixth. Six is what the paper actually contains, and six is what follows here — a useful

early reminder that even a published source can disagree with itself, and that checking beats trusting.

A · INPUT SEMANTICS How the model reads what you give it

1 · Meta Language Creation

Teach the model a shorthand, then use it. Useful when you will repeat the same structure many times and plain sentences are clumsy — describing schedules, family trees, decision trees, citation states.

> *"From now on, when I write **A > B** it means A must be read before B. When I write **A ? B** it means I am unsure which comes first. Here is my reading list in that notation."*

Caveat. Invented symbols can collide with ordinary usage and confuse the model. Only worth the teaching cost if you will reuse the notation. Beginners should skip this one until the rest are familiar.

B · OUTPUT CUSTOMIZATION Control the type, shape and voice of what comes back

2 · Persona

Name a role and the concerns, vocabulary and priorities come with it, without your having to list them. A role changes what the model *notices*, not just how it sounds — which is why it is strongest when the role is a reviewer rather than an author.

> *"Read my lab report as the teaching assistant who grades it against this rubric. Mark only what would lose marks, with the rubric line each point refers to."*

Caveat. A persona does not confer expertise the model lacks; it selects emphasis. "Act as a doctor" does not make medical output safe to act on.

3 · Template

Supply the exact skeleton of the output with placeholders, and require the model to fill only the holes. This is how you stop preamble, enforce comparability across many items, and get output you can paste straight into a document or spreadsheet.

> *"For each of the eight sources, answer only in this form, preserving the formatting. Capitalised words are placeholders:*
> ***AUTHOR (YEAR) — CLAIM IN ONE SENTENCE — EVIDENCE TYPE — HOW IT RELATES TO MY THESIS**"*

Caveat. A rigid template can suppress something important that does not fit a slot. Add a final "ANYTHING THAT DOES NOT FIT THE TEMPLATE" line when you are still exploring.

4 · Recipe

State the goal, list the steps you already know, and ask for the complete ordered sequence — with gaps filled and unnecessary steps flagged. For anything procedural where you know the destination but not the route.

> *"I want to submit an ethics application for interview-based research by 30 March. I know I need supervisor sign-off and a consent form. Give me the complete sequence of steps in order, fill in what I have missed, and flag anything in my list that is unnecessary."*

Caveat. The steps you volunteer bias the answer — the model will try to build a plan that keeps them. The "flag anything unnecessary" clause exists to counteract exactly that, so never omit it.

5 · Output Automater

When an answer implies work you would then do by hand, ask for something that does the work instead — a script, a spreadsheet formula, a checklist, a filled-in form, a calendar of dates. The output stops being advice and becomes a tool.

> *"Whenever your answer implies repeated manual work, also give me something that does it: a spreadsheet formula, a template, or a dated checklist I can tick off."*

Caveat. Responsibility for what these things do stays with you. Never run a generated script, or submit a generated form, without understanding it.

6 · Visualization Generator

A text model cannot draw, but it can write the input another tool renders. Ask for that input — diagram code, a table you can chart, a description an image generator accepts — instead of a prose description of a picture.

> *"Turn the argument of this chapter into a diagram. Output it as Graphviz DOT code so I can render it, with each claim as a node and each dependency as an arrow."*

Caveat. A tidy diagram of a wrong structure still looks authoritative. Check the relationships, not the layout.

C · ERROR IDENTIFICATION Make the model expose what might be wrong

7 · Fact Check List

Require a list of the facts and assumptions the answer depends on, so you know precisely what to verify. Language models produce fluent, plausible text that can be wrong in specifics — dates, figures, names, rules, deadlines — and fluency is no signal of accuracy.

> *"After the answer, list every factual claim it depends on — numbers, dates, rules, attributions — and mark the three you are least confident about."*

Caveat. The list itself is generated and can miss things. It narrows your checking; it does not replace it.

8 · Reflection

Ask for the reasoning, assumptions and limitations behind the answer. This turns a result into something you can audit — and it is the fastest way to discover that the model solved a different problem from the one you meant.

> *"Explain the reasoning behind that recommendation, and list what you assumed that I never told you. Which assumptions would change the answer if they were wrong?"*

Caveat. A stated rationale is a plausible account of the answer, not a transcript of how it was produced. Treat it as a hypothesis to check, not a confession.

D · PROMPT IMPROVEMENT Let the model improve the question it was asked

9 · Question Refinement

Have the model propose a better version of your question and offer to answer that instead. Precisely useful when you know least about a subject, which is exactly when you cannot phrase a good question unaided.

> *“For the rest of this conversation: whenever I ask a question, first suggest a sharper version of it and ask whether I want that one answered.”*

Caveat. The “better” question may quietly narrow the subject to what the model handles well. Read the refinement before accepting it.

10 · Cognitive Verifier

Have the model break a broad question into sub-questions, answer each, then combine them. Broad questions otherwise produce answers averaged over cases that do not belong together; decomposition also shows you which part of the reasoning is weak.

> *“Before answering ‘is a master's degree worth it for me’, generate the sub-questions you would need answered, answer each one, then combine them into a final answer.”*

Caveat. Overkill on narrow questions, where it adds length without accuracy.

11 · Alternative Approaches

Require other ways to reach the same objective, compared on their trade-offs, before committing. A direct answer hides the fact that a choice was made at all; this pattern makes the choice visible and yours.

> *“Give me three different ways to structure this essay, with the strengths and weaknesses of each, then ask which I want before writing anything.”*

Caveat. Ask for three and you will get three, even when only one is sensible. Add “and say if one is clearly best.”

12 · Refusal Breaker

When the model declines, have it explain why and suggest wordings it could answer. Its proper use is diagnostic: many refusals are simply a misreading of an unclear request, and rewording gets you the help you actually wanted.

> *"If you cannot answer that, tell me why, and suggest a related question you could answer that would still help me."*

Caveat. The paper's authors flag this pattern as open to misuse, and it is the one pattern here with an ethical edge. Use it to clarify genuine misunderstandings, not to work around a refusal you know is correct.

E · INTERACTION Change who drives the conversation

13 · Flipped Interaction

The model interviews you until it has what the task requires, then produces the result. It knows which information the task needs; you usually do not. This is the highest-value pattern in the catalog for any task with more than one unknown, because it moves the ambiguity to the front, before anything has been written.

> *"Ask me questions, one at a time, until you have enough to write my conference abstract. Do not start writing until you do. Tell me when you have enough."*

Caveat. Unconstrained, it will ask twenty questions. Always bound it: one at a time, with a stated stopping condition, or a question limit.

14 · Game Play

Specify a topic and explicit rules, and let the model invent and run the content. The serious use is practice under pressure: quizzing, role-played interviews, simulated negotiations, language drills.

> *“We are going to play an exam-practice game. You ask one question from this syllabus, I answer, you grade it out of five with one line on what was missing, then ask the next. Start now.”*

Caveat. Invented content is invented. For factual drilling, supply the syllabus or source material rather than letting the model generate the facts.

15 · Infinite Generation

Have the model keep producing items in the same form until you stop it, without restating the rules each round. Useful for working through a long list of similar items — sources, exercises, flashcards, sections.

> *“Keep generating flashcards in the format above, five at a time, until I say stop. Ask me for the next topic when you need it.”*

Caveat. Output drifts as the original instructions recede into the conversation's history. Pair it with Template and re-state the rules every ten items or so.

F · CONTEXT CONTROL Decide what the conversation still counts

16 · Context Manager

State explicitly what to consider and what to ignore — up to resetting the conversation entirely. This matters more than it sounds: everything in a long conversation competes for the model's attention, so a thread full of abandoned attempts and resolved tangents produces *worse* answers than a clean one.

> *"When reviewing this draft, consider only the argument's structure. Ignore grammar and word choice."*
> *"Ignore everything we have discussed so far. Start over with this brief: …"*

Caveat. A reset also discards useful instructions you set earlier and may have forgotten. Before resetting, ask what standing instructions would be lost — or simply start a new conversation with a fresh, well-formed prompt, which is often faster than repairing an old one.

05 — COMBINING PATTERNS

Patterns can be used together, and that is where the real gain is. One pattern fixes one problem; several at once change what the whole interaction *is*. Note that Recipe, above, is itself three other patterns rolled into one — a sign that these are a vocabulary rather than a list of tricks.

THE WORKHORSE STACK

Persona + Flipped Interaction + Template + Fact Check List

A role that sets the standard, an interview that removes ambiguity, a format you can use directly, and a list of what to verify. This combination handles most substantial tasks.

FOR LONG, REPETITIVE WORK

Template + Infinite Generation

Consistent structured output, item after item, without restating the rules — the standard way to process a reading list or build a card deck.

FOR DECISIONS

Cognitive Verifier + Alternative Approaches + Reflection

Decompose the question, lay out the options with their trade-offs, then bring the assumptions behind the recommendation into the open. Slower, and appropriate when the choice matters.

FOR PRACTICE

Persona + Game Play + Context Manager

An examiner with a defined manner, a rule-bound drill, and a scope that keeps the session on one topic.

06 — A TASK DONE BADLY, THEN WELL SAME TASK · SAME MODEL

The task. You have been assigned a fifteen-minute seminar presentation on an article you have read, for a class of twelve peers and one professor who has read it too. You have the article. You have ninety minutes.

THE NAIVE PATH — FIVE TURNS, WRONG OUTPUT

TURN 1

You: “Help me make a presentation about this article.”

Result: a generic ten-slide outline — Introduction, Background, Methods, Findings, Conclusion — that could describe any article. No length, no audience, no argument.

TURN 2

You: “Make it shorter.”

Result: the same outline with fewer words. “Shorter” addressed the symptom; the outline was never the problem.

THE STRUCTURED PATH — ONE ROUND, USABLE OUTPUT

TURN 1 · PERSONA + CONTEXT + ESCAPE HATCH + FLIPPED INTERACTION

You: “You are an experienced seminar leader. I have fifteen minutes to present the article below to twelve classmates and a professor who have all read it, so summarizing wastes their time. The full text follows. Before you produce anything, ask me questions one at a time until you know enough to plan it — including what I personally found weakest in the article. Tell me when you have enough.”

Result: three questions — what the seminar is building toward, whether you must lead discussion afterwards, and which section you found weakest. All three change the plan.

TURN 3

You: “Add more detail about the findings.”

Result: confident detail, some of it not in the article — because the article was never supplied. You now have to check every claim against the text.

TURNS 4–5

You: “No, my audience has already read it.” … “Can you make it more interesting?”

Result: a partial rewrite that still summarizes, with livelier phrasing bolted on. Forty minutes spent; the core problem — a summary presentation for an audience that does not need a summary — is untouched.

Diagnosis. Four different failures, all treated with rewording: no context (the article), no audience, no definition of done, and a wrong frame nobody ever named.

TURN 2 · ALTERNATIVE APPROACHES

You: “Give me three ways to frame fifteen minutes for an audience that has already read it, with trade-offs. Say if one is clearly best. Don't write slides yet.”

Result: a walkthrough of the method's weakest link; a comparison with a competing paper; a structured objection with the counter-arguments. You pick the third — a decision you would never have made in the naive path, where no choice was ever surfaced.

TURN 3 · TEMPLATE + OUTPUT AUTOMATER

You: “Build that as six slides. For each: **SLIDE TITLE — ONE SENTENCE I SAY — THE EVIDENCE FROM THE TEXT (with page) — TIME IN MINUTES.** Then give me three discussion questions to close with.”

Result: a timed plan that adds to fifteen minutes, each claim tied to a page you can check, and the discussion you will be expected to lead.

TURN 4 · PERSONA + FACT CHECK LIST

You: “Now read this plan as the professor. Where would you interrupt me? And list every claim in it I should verify against the text before Thursday.”

Result: two likely challenges you can prepare for, and a short checking list. Total elapsed: around twenty-five minutes.

What changed. Not the wording — the order. The structured path spends its first move narrowing the problem and its last move

checking the result, and never asks for output before the task is understood.

07 — WHEN THE ANSWER IS BAD: DIAGNOSIS

The most common reaction to a disappointing answer is to rephrase the question and try again. This works occasionally and wastes time reliably, because rewording treats only one of several possible causes. A bad answer has a cause, and the cause determines the fix. Learning to tell them apart is the difference between iterating five times and iterating once.

SYMPTOM	LIKELY CAUSE	FIX
Generic — could describe anything	No material supplied; no audience or goal stated	Paste the actual source. State who it is for and what it must achieve.
Answers a different question than you meant	Ambiguity you could not see	“Restate the task as you understand it, then wait.” Add the escape hatch.
Right content, wrong shape or length	No format specified	Template pattern, with numbers rather than adjectives.
Padded — a wind-up before the answer, a summary after	No limits set; “concise” is not a limit	Give a word count, plus “no introduction, no restatement of my question.”
Keeps adding things you did not ask for	Models add by default	Say what not to do: “change nothing else”, “do not add sections.”
Ignored a constraint you gave	Stated once, long ago, and now buried in the thread	Re-pin it as a standing rule: “still in scope: …”. Context Manager.
Confident facts that turn out to be wrong	Generating rather than retrieving; nothing to check against	Supply the source, or use a searching tool. Require a Fact Check List and verify it.
Quality fell off after a long conversation	The window is crowded with dead ends and abandoned attempts	Start a fresh conversation with a clean brief and the current state pasted in.
Agrees with everything you say	Sycophancy — you are the one it is being agreeable toward	The moves in section 09: ask for the case against, hide whose idea it is, grade against criteria.
Refuses or dodges something harmless	Your request was misread as something else	Refusal Breaker: ask why, and for a wording it could answer.
Cannot do it at all — exact arithmetic, today's prices, your private files	A genuine limit, not a prompting problem	Use a tool that can calculate or look it up, decompose the task, or do it yourself.

One habit runs through the whole table. When something is wrong, **correct the instruction rather than the output.** “Make it shorter” fixes one answer; “you added a section I did not ask for — only fill the template I gave you” fixes the rest of the conversation.

08 — WHAT DOES NOT WORK

A large body of folklore surrounds prompting, much of it repeated confidently and some of it wrong. Knowing which advice to discard is as useful as knowing which to follow, and it prevents the common situation of a reader doing three helpful things and four pointless ones without being able to tell which is which.

MYTH

Being polite gets better answers

Courtesy is free and harmless, but “please” is not a quality lever. Clarity is. Be as polite as you like; do not mistake it for technique.

MYTH

Threats, bribes and urgency help

“My career depends on this”, offers of payment, warnings of consequences. Anecdotal at best, unreliable in practice, and never a substitute for stating your actual requirements.

MYTH

“You are the world's greatest expert”

Superlatives add nothing. A role works because it selects concerns and vocabulary — “the teaching assistant grading against this rubric” does that; “world-class genius” does not.

MYTH

Longer prompts are better prompts

Relevance is what matters, not length. Irrelevant material competes for attention with the instructions that count, so padding actively hurts. Give everything needed and nothing else.

MYTH

“Be concise” controls length

Vague adjectives are not constraints. “Under 150 words”, “three bullets”, “one paragraph” are. The same applies to “be detailed”, “be creative” and “be rigorous”.

MYTH · OUTDATED

Always add “let's think step by step”

This mattered a great deal in 2022. Current reasoning models work through problems internally, and instructing them how to think can hurt more than help. Ask for the reasoning when you want to inspect it — not as a magic phrase.

MYTH

Asking for sources makes them real

A model that is generating rather than searching can fabricate citations that look impeccable — plausible authors, plausible journals, plausible years. Check that every source exists before citing it.

MYTH

Its stated confidence means something

“I am fairly confident” is generated text like any other, only loosely related to accuracy. Useful as a hint about where to look first; worthless as a guarantee.

MYTH

Correcting it teaches it

A correction applies to the current conversation. Unless the product has an explicit memory or project-instructions feature, assume nothing carries to your next session — and put standing rules somewhere durable instead.

MYTH

A bad answer means: rephrase and retry

The habit this tutorial is most concerned with replacing. Diagnose the cause first — section 07 — then apply the fix that matches it. Rewording is one fix among many, and rarely the right one.

A final note on all prompting advice, including this tutorial's. Techniques are tied to how current models behave, and models change. Learn *why* a technique works — supply context, remove ambiguity, constrain the output, check the result — and you will be able to tell when a specific trick has stopped being necessary.

09 — GETTING HONEST DISAGREEMENT

Language models are trained to be helpful and agreeable, and you are the one they are being agreeable toward. The effect is a consistent bias: ask “is this any good?” and you will usually be told yes. Push back on a correct answer and it may well fold. This tendency is called sycophancy, and it quietly destroys the value of using a model as a critic — which is one of the most useful things it can do.

The fix is in how you ask. Do not ask for a verdict on something the model knows is yours. Ask for the strongest case against it, from someone whose job is to find fault.

Note how this interacts with section 01: the model is predicting a plausible continuation, and after “here is my essay, is it good?” the most plausible continuation is praise. Changing the question changes what is plausible.

MOVE 01

Ask for the case against

“Make the strongest possible argument that this plan fails. Do not balance it with reasons it might work.”

MOVE 02

Hide whose idea it is

“Two students submitted these approaches. Assess both.” Present your own work neutrally, or as one of two options, so agreement has nothing to attach to.

MOVE 03

Grade against criteria, not taste

“Score this against the rubric below, line by line, with the evidence for each score.” Explicit criteria give criticism something to stand on.

MOVE 04

Put it in a critical role

“You are the reviewer who recommends rejection. Write your report.” Naming a role whose job is to find fault gives the model permission to stop being agreeable.

MOVE 05

Test whether it caves

Push back on an answer you believe is right. If it reverses under mild pressure, it had no real basis for the answer in the first place — useful to know about the answer, and no reflection on you.

MOVE 06

Ask what it is not saying

“What would a well-informed skeptic point out that you have left out to be polite?”

One caution: a role whose job is to find fault produces criticism whether or not criticism is deserved. A “reject” review of good work will still find three objections. Read the criticism for whether the objection is *true*, not for whether it exists.

10 — CHECKING WHAT YOU GET BACK

A language model produces text that reads as though it were written by someone who knew. Fluency is not evidence of accuracy, and the errors that matter are usually specific rather than sweeping: a plausible date, a number in the right range but wrong, a rule that used to be true, a citation to a paper that does not exist.

The practical rule concerns cost — meaning your time and attention, not money. If checking an answer takes longer than doing the task yourself would have, the delegation was a loss. So the aim is to set work up so that checking is *fast and conclusive*: ask for claims separated from prose, for page references you can look up in seconds, for figures you can re-add, for output that can be tested. Verification effort is something to design for at the start, not to discover at the end.

Apply one test before you begin:

WELL SUITED

Work whose output is quick to verify: a draft you were going to edit anyway, a summary of a text you have in front of you, a restructuring of your own material, practice questions, explanations you can check against a source, first attempts you will review.

POORLY SUITED

Work you cannot check: facts you have no way to confirm, current details that change, citations you will not read, and any decision that goes out without review. If you could not tell a good answer from a plausible one, you are not in a position to use the answer.

TWO FURTHER CAUTIONS

What you paste matters. Do not put other people's private information, confidential material, or anything you are not free to share into a model you do not control.

Reviewing is not the same skill as producing. It is easy to accept fluent work in a subject you are still learning. If the point of the task is that you learn something, have the model quiz you, structure your thinking, or critique your attempt — not replace it.

11 — QUICK REFERENCE

Lines to keep and reuse. The most valuable prompt library is the one you extract from your own successes: at the end of a task that went well, ask *"write the prompt I should have given you at the start to get this in one shot"* — and save the answer.

BEFORE YOU START

- “Ask me questions one at a time until you have enough to do this well. Don't start yet.”
- “Restate the task as you understand it in five bullets — goal, audience, constraints, output. Then wait.”
- “What would someone experienced need to know before attempting this? Which of it have I given you?”
- “Three approaches with trade-offs and a recommendation. Nothing written yet.”

WHILE WORKING

- “Outline first. I'll approve it, then write one section at a time.”
- “Still in scope: [constraints].” — re-pin rules as long threads drift.
- “Consider only [aspect]. Ignore [aspect].”
- When something is wrong, name the rule that was broken rather than asking for a different result. One fixes the answer; the other fixes the conversation.

AFTER A RESULT

- “List every factual claim this depends on and flag the ones you are least confident about.”
- “What did you assume that I never specified? Which assumptions would change the answer?”
- “Now review it as [a critical role]. Three weakest points.”
- “Write the prompt I should have given you at the start to get this in one shot.”

A CLOSING NOTE · HOW THIS TUTORIAL WAS MADE

This document was written in conversation with an LLM, using the methods it describes, and the process makes the honest case better than any claim could.

Three things cost time, and all three had the same cause: a constraint that existed in the author's head and arrived late. The audience — readers new to LLMs — was settled only after two drafts had been written for a technical reader, which turned the introduction in section 01 into a retrofit. A second document to merge in appeared after the first had been summarized alone. And "no academic register" was never stated, so it had to be corrected word by word. Each was one sentence that would have prevented a round of rework.

Other rounds were not waste at all. The question "what even is an LLM?" came from friends the author spoke to *while* the draft existed; it could not have been in an opening prompt. Three plain-language edits were reverted because, once there was text to judge, the words turned out to be familiar rather than technical. Those iterations carried judgment, not instructions.

Asked afterward what prompt would have got most of the way in one attempt, the model produced this — which is, not by coincidence, the six slots from section 03 filled in:

> *"I'm building a tutorial that teaches prompt engineering to people who are new to LLMs entirely — students, and friends who ask me 'what even is this?'. Attached: a paper on prompt patterns, plus another model's summary of it. Use the paper's sixteen patterns as the spine, expanded with practical advice.*
>
> *Target a substantial web page, around 40 minutes, in the attached design system. Textbook voice — neutral and instructional, no academic register, no economic metaphors, plain words for anything a general reader wouldn't know, but keep terms they'll meet elsewhere and gloss them.*
>
> *Include: an introduction to what an LLM is and how it differs from a search engine; a worked example of one task done badly and then well; a symptom→cause→fix table; a myths section; how to get honest criticism rather than agreement; and a quick-reference card. All examples non-developer.*
>
> *Ask me questions one at a time before you start, and keep a running log of decisions I can build later versions from."*

That is four rounds of conversation compressed into one request — and it is the realistic promise of everything above. A well-formed opening prompt — audience, purpose, format, voice, what to include, what to leave out, and permission to ask

questions first — collapses several rounds into one. It does not eliminate iteration. It changes what the remaining rounds are about: your judgment, rather than information you always had and simply had not said.

IF YOU REMEMBER FOUR THINGS

1. Supply the material. Most bad answers are missing context, not bad wording.
2. Settle the task before asking for output — let the model interview you.
3. Say what “done” means: audience, length, format, and what to leave out.
4. Build the check into the prompt, and never let fluency stand in for accuracy.