START HERE

Getting Real Work out of a Model

You already use ChatGPT, Claude or Gemini. This is how to stop getting average answers.

First edition, September 2026

Addaly

Series editor: Dr Mustafa Mun

Draft, open for academic review

START HERE

Getting Real Work out of a Model

You already use ChatGPT, Claude or Gemini. This is how to stop getting average answers.

Series editor: Dr Mustafa Mun

Addaly

First edition, September 2026 · Draft, open for academic review

Getting Real Work out of a Model

First edition, September 2026, build 62a26075.

Written by AI models to a written standard, with Dr Mustafa Mun as series editor. Exam keys checked blind by a second model: 3,665 of 3,666 agreed. Academic review invited.

The manuscript was drafted with the assistance of a large language model and was edited and published under his name. That is stated plainly here rather than left to be discovered, because a reader is entitled to know how a book they are trusting was made.

How to cite: *Mun, M. (2026). Getting Real Work out of a Model. Addaly. addaly.com/learn/prompting.*

This book is the printed form of the course of the same name at addaly.com/learn/prompting. The website is the living version and is corrected first. Where the two differ, the website is right.

Free to read, free to download, free to print, and free to teach from. There is no paid edition and there never will be.

Every diagram in it is drawn from data by code, not generated as a picture, so the numbers on the page are the numbers in the argument. The answers to every check, test and examination question are at the back.

8 modules · 72 lessons · 30 figures · 8 case studies · 53,059 words · about 10 hours

Brief contents

1

MODULE 1

What a prompt actually is

Explain what happens to a prompt between the send button and the answer — tokenisation, a single re-sent string, sampling, instruction tuning, and the product wrapped around the model — and use that mechanism to predict which prompt changes will help and which are folklore.

Lesson 1 · 6 min

Why the same question gets a better answer

THE ONE THING TO KEEP

A vague question has a vague best answer; detail is what makes a good answer possible at all.

Two prompts, one question

Someone asks:

> *How do I improve my CV?*

They get a list. Use action verbs. Quantify your achievements. Tailor it to the role. Keep it to one page. All true, all useless, because they already knew it.

Now the same person asks:

> *I am applying for a junior data analyst job at a bank in Lagos. My CV is pasted below. The advert asks for SQL, Excel and "stakeholder communication". I have no paid experience — only a six-month university project where I cleaned and analysed 40,000 hospital records in Postgres. Rewrite the top third of the CV so a recruiter sees the SQL work in the first ten seconds. Keep the whole thing to one page. Do not invent any experience I have not described.*
>
> *[full CV pasted]*

Both people want a better CV. The answers are not remotely comparable, because only the second question is answerable.

Why that happens

A language model produces a plausible continuation of the text in front of it. That is genuinely most of what is going on. It has no picture of your life, your file, or your Thursday deadline. It has your words and nothing else.

"How do I improve my CV?" is a question that thousands of articles have already answered. The most plausible continuation is a generic list, because a generic list is what generically follows that sentence. The model is not being lazy. It is being accurate about a vague question.

The second prompt cannot be continued generically. There is one CV, one advert, one awkward gap to cover.

Average in, average out. When someone says a model gave them a bland answer, they have usually asked a question whose honest answer is bland.

What actually did the work

Look at what the second prompt added:

- **Who and where.** Junior, Lagos, banking. What a CV should look like changes by market.
- **The target.** The advert's own three words, quoted.
- **The raw material.** The CV itself, pasted, not described.
- **The hard part.** No paid experience. This is the difficulty of the task. Hiding it would have produced advice for someone else.
- **A finished state.** Top third rewritten, one page, SQL visible in ten seconds.
- **A boundary.** Do not invent experience.

None of that is a trick. It is the briefing you would give a friend who offered to help over lunch.

The incantations are mostly not the thing

A lot of prompting advice is about magic words. Tell it that it is a world-class expert. Offer it a tip. Tell it to take a deep breath. Threaten it politely.

Some of those had measurable effects on the weaker models of 2023. Most have shrunk or disappeared as models improved, and several were never more than folklore repeated between blog posts. You will still find them recommended confidently today.

The honest version: the reliable levers are what you are trying to do, the material you are working from, the constraints that make it hard, and the shape you want back. Everything else is decoration.

That is good news. There is no secret language to learn. There is a habit of under-describing your own problem, and you can drop it.

A test that takes one minute

Before you send a prompt, read it as if a competent stranger had sent it to you, with no memory of your week. Ask two questions:

1. Could I start this task from this text alone?
2. What would I have to ask you first?

Every question you would have to ask is a missing line. Add those lines and send it again.

That is the whole method. The rest of this course is about doing each part of it well.

BEFORE YOU MOVE ON

Someone asks a model to write an email to their landlord about a broken water heater and gets something generic. They try again with a longer, more emphatic prompt — "this is really important to me, please do an excellent job, you are a brilliant writer" — and get almost the same email. What is the best explanation?

A. The prompt is still too short. Models respond to length, and it needs to be several paragraphs before the quality improves.

B. The second attempt added emphasis but no new information: nothing about the heater, the flat, how long it has been broken, or what the landlord has already said.

C. Wording never affects the output, so no rewrite of the prompt could have changed anything.

D. The model already produced its best possible answer to that question, so the next step is a different tool or a stronger model.

The answer and why: p. 385

Lesson 2 · 9 min

What the model actually receives

THE ONE THING TO KEEP

The model receives one flat block of text — system prompt, whole transcript, your message — chopped into tokens, with no memory and no formatting, and every prompting technique is a consequence of that shape.

One string, every time

You type a message. The product assembles it into a single block of text and hands that block to the model. The block contains a system prompt you did not write, the whole conversation so far, and your new message at the end.

The model reads it, produces some text, and forgets everything.

That is worth sitting with, because two common beliefs die here.

The model has no memory of your previous turns. It has a transcript. When you send message twelve, messages one through eleven are re-sent with it, every time, as text. The illusion of memory is a text file being re-read. This is why a long conversation gets slower and more expensive, and why an error in turn three is still in the room in turn thirty.

There is no formatting. If you paste a heading in bold, the model does not receive bold. It receives whatever characters the product turned that into, which is usually nothing or a couple of asterisks. Colour, font, cell borders, indentation in your spreadsheet — none of it arrives. If a distinction in your document is carried only by looks, you have to say it in words.

Tokens, and why they matter to you

The block of text is chopped into **tokens** before the model sees it. A token is a chunk of characters that the tokeniser has learned to treat as one unit. Common words are one token. Rarer words split.

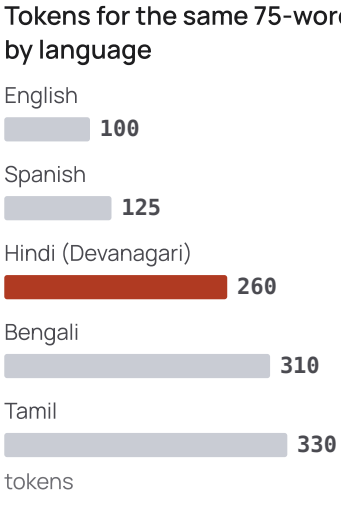
Rough figures for English: one token is about four characters, and 100 tokens is about 75 words. So a 3,000-word document is roughly 4,000 tokens.

Three consequences you will actually meet:

Cost and limits are counted in tokens, not words. When a tool says it accepts 128,000 tokens, that is roughly 96,000 English words — a decent novel. When an API bill says 2 million tokens, that is what you paid for.

Non-English text costs more. The tokenisers used by the big models were fitted mostly on English text, so English is efficient and other scripts are not. The same sentence in Hindi, Bengali, Tamil, Amharic or Thai typically costs two to four times as many tokens as its English translation, sometimes more, because the tokeniser breaks unfamiliar characters into small pieces. This is not a rumour; you can measure it in a minute, and it means a Devanagari document hits the context limit far sooner than an English one of the same length, and costs more per page on a metered API.

Figure 1.1



The tokenisers were fitted mostly on English, so English is the cheap case. The same passage in Hindi, Bengali or Tamil costs two to four times as many tokens, hits the context limit sooner, and costs more per page on a metered API.

Some tasks are hard for a reason you can now see. Ask a model how many times the letter *r* appears in "strawberry" and it may get it wrong. The word arrived as two or three tokens, not as nine letters. Counting characters inside a token is like being asked to count the strokes in a word you were handed as a picture. The same explains wobbles with reversing strings, rhyme in some languages, and arithmetic on long numbers, which get split into digit groups in ways that vary by model.

You can see your own text tokenised for free. In Python, `pip install tiktoken` and four lines will count them. Several model providers also publish a web page where you paste text and watch it split. Do it once with a paragraph of your own language; it explains more than a diagram would.

The prompt has invisible parts

Your message is not the whole prompt. In front of it sits a system prompt written by whoever built the product, typically a few hundred to a few thousand words of instructions about tone, refusals, formatting and available

tools. Behind it may sit the results of a web search, the contents of a file you attached, or notes the product has saved about you.

So "the same prompt" genuinely is not the same prompt in ChatGPT, Claude, Gemini, Copilot and a raw API call. When a colleague's prompt works for them and not for you, this is usually why, and we come back to it at the end of this module.

Why this matters for prompting

Everything in this course follows from the shape above:

- Because the model receives only the block, anything not in the block does not exist. Hence the next module on context.
- Because the block is re-sent, old mistakes persist and long chats degrade.
- Because there is no formatting, structure has to be carried by words and punctuation you choose deliberately.
- Because it is tokens, not meaning, your budget is finite and your language affects the price.

None of that is a trick. It is what the machine is. Most bad prompting is a reasonable expectation about a different machine.

BEFORE YOU MOVE ON

A finance officer pastes a spreadsheet extract into a chat. In her file, overdue rows are highlighted in red and the model keeps treating them like the others. What is the mechanism behind the failure?

A. The model can see the colours but has been trained to ignore formatting in favour of the text content.

B. Spreadsheets are tokenised differently from prose, so the cell metadata is dropped during tokenisation.

C. Only characters reach the model, so the red fill was never in the block of text it received; the distinction has to be stated in words.

D. The context window filled up, and cell attributes are the first thing discarded when text is truncated.

The answer and why: p. 386

Lesson 3 · 8 min

Why the same prompt gives a different answer

THE ONE THING TO KEEP

Variation between answers comes from the sampler, not from the model changing its mind, so a prompt is only tested when you have run it several times and seen what moved.

The model does not choose a word

At each step the model produces a probability for every token in its vocabulary — tens of thousands of numbers, one per possible next chunk. Something outside the model then picks one. That picker is called the **sampler**, and its settings are why you can send an identical prompt twice and get two different answers.

CASE STUDY · MODULE 1

Six hundred handouts and a page of magic words

An illustrative case, built from documented patterns.

A government polytechnic in Coimbatore, three weeks before the employability module starts, deciding whether to print a prompt handout for six hundred students

The Department of Computer Applications at a government polytechnic in Coimbatore runs a forty-hour employability module for final-year students. Placement season starts in November. For two years the module's most popular handout has been four pages titled "50 Power Prompts", collected by a lecturer, Mr Balan, from social media posts and a video series. It opens with "You are a world-class career coach with 20 years of experience", includes "take a deep breath and work through this step by step", and ends with a note that offering the model a two hundred dollar tip improves results.

Six hundred copies had been quoted for by the campus press. The head of department, Dr Selvi, had approved the spend. Printing was booked for Friday.

On the Tuesday a student called Aarthi came to Mr Balan with a problem he was not expecting. She had used prompt 14 — the world-class career coach one — to write a covering letter for a testing role at a firm in Chennai. The letter was fluent and said nothing. She had then written four lines of her own naming the advert's three requirements and the one project she had actually done, pasted her CV underneath, and asked again. The second letter was better, and she wanted to know why the handout had not told her to do that.

Mr Balan did the thing the handout did not. He took ten past adverts and ten real CVs from students who agreed to it. He ran the handout's prompt over all ten, three times each, and scored every letter out of four on a checklist: does it name a requirement from the advert, does it cite something from the CV, does it avoid inventing experience, is it under three hundred words. Then he ran a plain version with no persona, no breathing, no tip, with the advert and the CV pasted in full.

The handout version scored 21, 19 and 22 out of 40. The plain version scored 31, 33 and 30.

Neither total was the interesting part. The interesting part was that three runs of the same unchanged prompt differed by three marks on their own. That meant any single before-and-after demonstration — the kind that fills the posts the handout was assembled from — could show whatever the person running it was hoping to see, and could show it honestly, without anybody lying.

He found a second thing he had not gone looking for. Two students had drafted in Tamil and asked for a translation, and both had run into the tool's message limit on a document that in English would have fitted easily. The same passage in Tamil was costing roughly three times as many tokens. The students working in their first language had a smaller working space than the students working in their second, and nothing in the interface said so.

So there was a decision, and both sides of it cost something real.

Reprinting meant rewriting the handout in three weeks, which the campus press could not absorb. The next slot was late November, by which point the module would be half over and six hundred students would have started placements with no material at all. Part of the approved spend had already gone on plates.

Not reprinting meant handing six hundred students a document that Mr Balan could now demonstrate was worse than a page of plain advice, in a module whose entire purpose was to get them hired.

Dr Selvi's objection was not that he was wrong. It was that "we measured it and the magic words did nothing" is a hard sentence to say about a document the department has taught from for two years, and that the handout was popular precisely because it looked like secret knowledge. A page that says paste the advert, paste your CV, name the hard part and say what you will not invent looks like something the students already knew.

YOUR ANSWERS

1. Mr Balan's plain prompt scored higher on all three runs. Why was the three-mark spread between runs of the same prompt more important than the gap between the two versions?

2. The handout's persona line was not merely useless. What was it costing?

3. Aarthi's four extra lines were not a technique she had been taught. What were they doing, in terms of the six slots?

STOP HERE

Write your answers before you turn the page. How it played out starts on p. 52.

This page is blank so that how it played out stays out of view until you turn the page.

CASE STUDY · MODULE 1

How it played out

They cancelled the four-page handout and printed a single A4 sheet, which the press could fit on the Friday slot because it was one plate rather than four. The sheet had six headings — the task, the material, the examples, the constraints, the shape, the escape hatch — and, on the back, one worked covering letter built slot by slot from a real advert and a real CV.

The magic-word page was not thrown away. Mr Balan kept it as the module's second exercise: students run three of its prompts and the plain version over the same three adverts, three times each, and score them on the four-point checklist themselves. Roughly a third of each cohort now reports that a phrase they were sure had helped moves the score by less than the gap between two runs of the identical prompt.

The Tamil token finding went into the module as a separate ten-minute demonstration, because it changes what students do rather than what they believe: draft long documents in whichever language you are working in, but expect a Tamil or Malayalam paste to fill the window two to three times faster than the English equivalent, and split it.

The honest cost was borne by the first cohort. The new sheet went out without the two extra pages of worked examples Mr Balan wanted, because there was no time to write them, and the November batch had to be taught those from the board. Dr Selvi's other prediction was also correct: the sheet is less popular. Students who used to ask for more prompts now ask for more examples, which is more work for the department and better for them.

The questions, discussed

1. Mr Balan's plain prompt scored higher on all three runs. Why was the three-mark spread between runs of the same prompt more important than the gap between the two versions?

Because it sets the size of the effect he can detect at all. Variation between runs comes from the sampler, not from the model changing its mind, so a difference smaller than the run-to-run spread is invisible without careful

measurement. That is exactly the size of most of the effects reported for magic phrases, and it means a single before-and-after demonstration proves nothing either way. It also tells him what to do next: to claim a small improvement he would need many more items, not more confidence.

2. The handout's persona line was not merely useless. What was it costing?

The place where the useful information should have gone. Every sentence of role-play and motivation occupies context and dilutes the instructions that matter, and in this case it stood in for the advert and the CV — the two things the model could not possibly get anywhere else. A prompt that opens with three sentences of costume and then asks a vague question is worse than the same prompt with those sentences replaced by the material.

3. Aarthi's four extra lines were not a technique she had been taught. What were they doing, in terms of the six slots?

She filled the two slots the handout left empty. Pasting the CV and the advert supplied the context; naming the three requirements and the one project she had actually done set a finished state rather than a topic. The persona had filled none of the six. Her letter improved because the question stopped being answerable generically, not because she found a better phrasing.

MODULE 1 TEST

Check what stuck

6 questions, one right answer each. This one is for you, not for a record. The answers, and the reason behind each, are in the key on p. 384. If two go wrong, re-read the module before the exam.

1 A wrong assumption you introduced in turn three of a chat keeps colouring answers at turn thirty, even after you corrected it twice. What is the mechanism?

A. The whole transcript is re-sent as text with every message.
B. The model keeps a running memory of the chat and updates it after each of your turns.
C. Your corrections adjusted the model's weights, and weight changes take several turns to settle.
D. The product caches the first answer and reuses parts of it to keep replies consistent.

2 A prompt produced correctly formatted output the first time you ran it. You are about to run it over two hundred customer messages. What should you do first?

A. Run it five times over the same three messages and see what moves.
B. Add the phrase "be precise and consistent" so the format holds across a long batch.
C. Nothing — a prompt that produced the right format once will produce it two hundred times.
D. Lower the temperature by rewriting the prompt in shorter and more literal sentences.

3 You give eleven constraints in one prompt. The answer satisfies eight, silently drops three, and reads as though it satisfied all eleven. Why does nothing announce the omission?

A. The dropped constraints were probably contradictory, and contradictions are always resolved silently.

B. Constraints beyond about eight exceed the model's working memory and are truncated before it reads them.

C. The product strips instructions it judges to be stylistic rather than substantive.

D. Instruction-following is a trained behaviour, and nothing checks the output against your request.

4 A prompt opens "You are an expert lawyer with 20 years of experience. Review my contract." Changing twenty to thirty years does nothing. What is the honest account of what the role is doing?

A. It selects a more capable internal expert mode, which saturates after about twenty years.

B. It has no effect at all, since personas were removed from instruction tuning some years ago.

C. It sets register, and adds no competence; "review" is the part that has not been specified.

D. It primes domain vocabulary, which is the main reason specialist prompts outperform general ones.

5 Someone shows you one before-and-after pair proving that adding "take a deep breath and work through this step by step" improved an answer. Why is that not evidence?

A. The phrase came from a paper that has since been retracted by its authors.

B. Two runs of the identical prompt already differ by more than the effect being claimed.

C. Encouraging phrases work only on reasoning models, and the demonstration used an ordinary one.

D. Any improvement from a phrase is temporary, because providers patch known prompt tricks within weeks.

6 A colleague's prompt works for her and not for you. Before you rewrite the words, what should you check?

A. Whether she is paying for a subscription tier that exposes a different sampling temperature.

B. Whether her phrasing uses vocabulary the model was more heavily trained on.

C. Whether she ran it more times than you did and kept only the best result.

D. Whether her product has search, reads files in full, or is injecting memory.

> *My colleague sent a rude email about the project and I want to reply firmly but professionally.*

Every important word there is your interpretation. *Rude* could be a sarcastic aside, a copied-in manager, or a flat sentence you have read four times in a bad mood. *The project* could be anything. The model now writes a reply to an email it has never seen, and it will be generic, because it is answering the only thing it was given: a category of situation.

Paste the email. The reply changes completely, because the model can see the one sentence that actually stung and the three that were perfectly normal.

The three-layer rule

For any task involving a document, a message, a file or an error, there are three layers, and people habitually supply the wrong one.

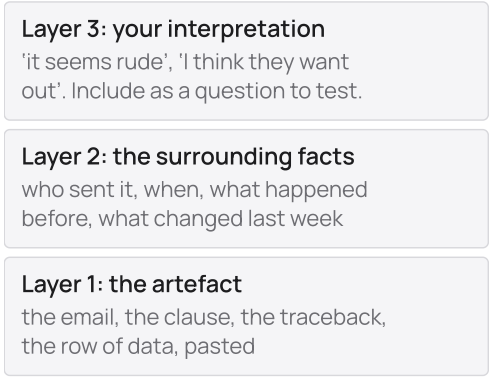
Layer 1 — the artefact. The email. The contract clause. The traceback. The row of data. The photograph of the letter. **Layer 2 — the surrounding facts.** Who sent it, when, what happened before, what changed recently. **Layer 3 — your interpretation.** It seems rude. It looks like a bug in the loop. I think they are trying to get out of the contract.

Layers 1 and 2 are what the model cannot get anywhere else. Layer 3 is worth including *as* an interpretation — "I read this as hostile, am I right?" — but never as a substitute for layer 1.

The single most common failure in this whole course is a prompt made entirely of layer 3.

Figure 2.1

The three layers of any document task



Layers one and two are what the model cannot get anywhere else. Layer three belongs in the prompt as a question, never as the input. The commonest failure in the course is a prompt made entirely of layer three.

What "the source" means in different jobs

A bug. The failing code, the full traceback, one line of the input data. Not "it crashes on the sales file". A traceback names the file, the line and the exception type; those three facts often solve it outright.

A document question. The clause, plus enough either side that it is not quoted out of context. If the document is long, paste the section and say what the document is.

A data question. The header row and five real rows, not a description of the columns. Column names carry information you have stopped noticing — `amt_inr`, `created_at_utc`, `status_2` — and the model will use them.

A writing task. The draft, the brief you were given, and one piece of work you consider good. Do not describe the house style; show it.

A decision. The actual options, with their actual numbers. Two quotes, both pasted, beat "one is cheaper but slower".

EXAMINATION

Getting Real Work out of a Model — final examination

This paper covers the whole course:

- what happens to a prompt between the send button and the answer
- the material you supply and how you mark it
- teaching by example
- the shape of the output
- making a model work the problem
- turning a lucky prompt into a reliable one
- checking an answer
- applying all of it to the work you actually have

56 questions, the whole pool. The site draws 25 for a sitting, pass mark 70%. Work any 25 under your own conditions. Answers from page 428.

1 Module 1 · What a prompt actually is

A model miscounts the letter r in "strawberry". Which fact about the input explains it?

A. Counting is arithmetic, and arithmetic is performed by a separate component that this task bypasses.

B. Short words are stripped of repeated characters during preprocessing to save context.

C. Spelling questions are answered from memorised word lists rather than from the text you sent.

D. The word arrived as two or three tokens, not as nine separate letters.

2 Module 1 · What a prompt actually is

You set temperature to zero and still get two different answers from an identical prompt. What is the honest explanation?

A. The provider silently raises the temperature when a request looks creative rather than factual.

B. Temperature zero applies to the visible answer only; any internal reasoning still samples.

C. Zero is a nominal setting and providers clamp it to a small positive value for output quality.

D. Batched execution changes the order of floating-point additions, flipping near-tied tokens.

3 Module 1 · What a prompt actually is

Which sentence should be deleted from a prompt, by the audit rule for decorative detail?

A. "The reader is a school governor who is not a lawyer and has ten minutes."

B. "I value clarity and precision, so please be thorough in your analysis."

C. "If the contract does not say, write 'not stated' rather than inferring."

D. "The budget is 400,000 pesos and the export rules changed last week."

4 Module 1 · What a prompt actually is

Of the six slots a working prompt fills, which one most often prevents a confidently invented answer?

A. The examples, because a demonstrated boundary is more precise than a stated one.

B. The constraints, because a checkable limit is honoured more reliably than a preference.

C. The escape hatch, because it makes "I cannot" an available response.

D. The output shape, because a named container leaves no room for unsupported commentary.

5 Module 1 · What a prompt actually is

Which of these has held up across models and tasks, rather than being folklore from 2023?

A. Telling the model that the task is very important to your career.

B. Assigning a persona with a stated number of years of experience.

C. Offering a tip, which raises effort by changing the reward framing.

D. Naming the finished state rather than the topic.

6 Module 1 · What a prompt actually is

"How do I improve my CV?" returns generic advice. What is the most accurate description of why?

A. The model is being accurate about a vague question, whose honest answer is generic.

B. The model is conserving effort, since short questions are routed to a smaller model.

C. The topic is over-represented in training data, so specific advice has been averaged away.

D. The question has no output format, and unformatted requests default to a list of tips.

Module 1 · What a prompt actually is

The module starts on p. 11.

MODULE 1 TEST

Module 1 test, Q1, p. 54

A wrong assumption you introduced in turn three of a chat keeps colouring answers at turn thirty, even after you corrected it twice. What is…

Answer A: The whole transcript is re-sent as text with every message.

Why: There is no memory. Every message re-sends the system prompt, the entire conversation so far, and your new text as one flat block. Turn three is therefore still physically present at turn thirty, competing with your correction rather than being overwritten by it — which is why starting again with the correction written into the first message beats a fifth firmer correction at the bottom of a long chat.

Module 1 test, Q2, p. 54

A prompt produced correctly formatted output the first time you ran it. You are about to run it over two hundred customer messages. What should…

Answer A: Run it five times over the same three messages and see what moves.

Why: One success is a single draw from a distribution. The sampler makes two runs of an identical prompt differ, so a prompt that gives the right format four times in five will break roughly forty rows out of two hundred. Five runs over the same few items shows you what your prompt actually determined and what you left to chance, and it is the only thing that turns a prompt that got lucky into a prompt that was tested.

Module 1 test, Q3, p. 55

You give eleven constraints in one prompt. The answer satisfies eight, silently drops three, and reads as though it satisfied all eleven. Why does nothing…

Answer D: Instruction-following is a trained behaviour, and nothing checks the output against your request.

Why: Obedience was trained by examples and human preference, not built in as a mechanism. The model has learned the shape of following instructions, so it satisfies most and drops some, and there is no internal checklist comparing the finished text against what you asked. That is why the fix is structural: rank the constraints, keep them few and checkable, or add a separate pass that checks the draft and quotes the deciding words.

Module 1 test, Q4, p. 55

A prompt opens "You are an expert lawyer with 20 years of experience. Review my contract." Changing twenty to thirty years does nothing. What is…

Answer C: It sets register, and adds no competence; "review" is the part that has not been specified.

Why: A role changes vocabulary, reading level and pacing — real effects, sometimes exactly what you wanted. It cannot add knowledge, because there is no expert mode to switch on. The failure in this prompt is the verb: "review" names no finished state, so it stands in for whatever you actually wanted. Naming the verb, the object, the constraints, the finished state and what to do when stuck is the work; the costume is the cheap part.

Module 1 test, Q5, p. 56

Someone shows you one before-and-after pair proving that adding "take a deep breath and work through this step by step" improved an answer. Why is...

Answer B: Two runs of the identical prompt already differ by more than the effect being claimed.

Why: That phrase was found by an automatic search for prompt wordings, scored on one maths benchmark, on one model of 2023. Whatever it is worth now, its effect is typically smaller than the run-to-run variation you get from an unchanged prompt — so a single pair can show whatever the person running it hoped to see, honestly and without anyone lying. Detecting a small effect needs ten items with known answers, three runs of each version, and a count.

Module 1 test, Q6, p. 56

A colleague's prompt works for her and not for you. Before you rewrite the words, what should you check?

Answer D: Whether her product has search, reads files in full, or is injecting memory.

Why: Your prompt is one layer of what reaches the model. In front of it sit a system prompt you did not write, any tools the product has, whatever a file upload actually became, saved notes about the user, and sometimes per-message routing to a different model. A prompt that behaves differently in two products usually differs there rather than in the words, and four questions about the layers resolve it faster than an afternoon of rewriting.

THE LESSON CHECKS

Lesson 1, p. 15

Someone asks a model to write an email to their landlord about a broken water heater and gets something generic. They try again with a...

Answer B: The second attempt added emphasis but no new information: nothing about the heater, the flat, how long it has been broken, or what the landlord has already said.

Why: A vague question has a vague best answer; detail is what makes a good answer possible at all.

A The prompt is still too short...: Longer prompts genuinely do tend to work better, so length gets mistaken for the active ingredient instead of the detail that usually travels with it.

C Wording never affects the output, so...: Once you learn that flattery and pleading do nothing, it is easy to overcorrect into thinking phrasing is irrelevant — when phrasing that carries information changes the answer entirely.

D The model already produced its best...: When effort does not help, blaming the model is the natural move — but between the two attempts the actual question never changed.

Lesson 2, p. 19

A finance officer pastes a spreadsheet extract into a chat. In her file, overdue rows are highlighted in red and the model keeps treating them…

Answer C: Only characters reach the model, so the red fill was never in the block of text it received; the distinction has to be stated in words.

Why: The model receives one flat block of text — system prompt, whole transcript, your message — chopped into tokens, with no memory and no formatting, and every prompting technique is a consequence of that shape.

A The model can see the colours…: Assumes formatting arrives and is then de-prioritised; nothing about the colour ever reaches the model at all.

B Spreadsheets are tokenised differently from prose…: Blames the tokeniser for something that happened earlier: the paste operation already discarded the fill colour.

D The context window filled up, and…: Invents a priority order for truncation; the colour was absent from the first token, not evicted later.

Lesson 3, p. 23

A shop owner has a prompt that pulls order number, date and total from pasted emails. It worked perfectly in his three trials. He runs…

Answer A: Three successes at a default temperature do not show the spread of outputs the same prompt produces, and the failure rate was always there.

Why: Variation between answers comes from the sampler, not from the model changing its mind, so a prompt is only tested when you have run it several times and seen what moved.

B The batch was too large, and…: Each email is a separate request; there is no accumulating fatigue across a batch.

C The emails in the batch were…: A real risk, but it does not explain why the *same* prompt on the *same* input can behave differently; the sampler is the mechanism being tested here.

D Long-running jobs hit rate limits, and…: Attributes the pattern to infrastructure; a rate limit produces errors, not quietly reformatted dates.

START HERE

Getting Real Work out of a Model

You already use ChatGPT, Claude or Gemini. This is how to stop getting average answers.

WHAT IT TEACHES	A practical course for people who use AI chat tools every week and quietly suspect they are getting less than the tools can give. Eight modules and seventy-two lessons on what actually changes an answer: context, tasks, examples, format, reasoning, and honest iteration.
WHAT IS INSIDE	Eight modules and 72 lessons, 30 figures drawn from data, eight case studies, eight module tests, a 56-question examination, and every answer with the reason behind it.
LICENCE	CC BY-NC-SA 4.0. Copy, print, share and adapt it for any non-commercial purpose, with credit, under the same licence. There is no paid edition.
THE COURSE	Free to read, with the lessons, the films and the examination, at **addaly.com/learn/prompting**

Addaly

First edition, September 2026 · build 62a26075

Scan to open the course