

EVERYONE · 5 MINUTE READ

Context engineering: giving AI what it needs to know

Good results depend less on clever wording and more on giving the assistant the right background, examples, documents and definition of good.

What it is

Context is everything the AI assistant can see when it answers: your instructions, the chat so far, any documents you add and any saved instructions. **Context engineering** means choosing that material on purpose, so the assistant works from the right facts rather than guesses.

Early advice on AI focused on clever wording ("prompt engineering"). Wording still helps, but as assistants such as Claude, ChatGPT, Microsoft Copilot and Gemini improved, the bigger difference came from what you give them to work with. A short, plain request with the right material often beats a long, clever request with none.

Think of a chef. A great recipe (the prompt) helps, but if the fridge is empty or full of the wrong ingredients, the meal will disappoint. Context engineering is stocking the fridge: the right ingredients, not every ingredient.

Why it matters

- The assistant knows nothing about your organisation, customers or standards unless you tell it.
- With the right material, answers become specific to your situation instead of generic.
- Too much material can bury the important details.
- Good context, saved once, can be reused by the whole team.
- Shared material must still follow your data rules: only approved tools and approved information.

How to do it

1. **Give background:** who you are, who the output is for, and why it is needed.
2. **Add examples** of past work you were happy with, so it can copy the style and structure.
3. **Attach the source documents** the answer should rely on, if your organisation allows it, and say "use only these".
4. **Define what good looks like:** length, tone, must-have points, and what would make it wrong.
5. **Cut what is not needed:** remove old versions, unrelated pages and long email chains.
6. **Use projects or saved instructions** where your tool offers them, so standing background does not need retyping.
7. **Keep a context pack:** a short document you maintain and reuse for repeat work.
8. **Review and update** the pack when things change, and check outputs as usual.

Example

You write a monthly update for senior managers.

Before:

Write the monthly update for October.

Result: vague, invented highlights.

After, with a context pack pasted in or saved in a project:

CONTEXT PACK: Monthly management update
Audience: senior managers, read in under two minutes.
Purpose: show progress, risks and decisions needed.
Structure: 1) Headline, 2) Three wins, 3) Two risks, 4) Decisions needed, each with an owner.
Tone: factual, no adjectives like "excellent" or "huge".
Good looks like: [paste last month's approved update]
Rules: use only the notes below. If a figure is missing, write [missing] instead of guessing.

THIS MONTH'S NOTES:
[paste team notes]

Then simply ask: "Draft the October update from the notes." The draft follows your structure and flags gaps. You still check every figure against the source before it goes out.

Common mistakes

- **Polishing wording while giving no material.** Fix: spend the time finding the right documents and examples instead.
- **Pasting everything just in case.** Fix: include only what the task needs.
- **Forgetting to say what "good" means.** Fix: add a short checklist or a model example.
- **Letting a context pack go stale.** Fix: give it an owner and a review date.
- **Adding confidential documents to an unapproved tool.** Fix: check your organisation's rules first and use made-up examples when testing.

Try it this week

1. Choose one task you do at least monthly.
2. Spend ten minutes writing a context pack: audience, purpose, structure, tone, one good example, rules.
3. Run the task once with only a short request, and once with the pack.
4. Compare the two drafts and note what the pack fixed.
5. Save the pack where you and your team can find it.

Going further

- **Keep context packs in Markdown**, plain text with simple headings and bullets, which assistants read easily (see guide 06).
- Projects or workspaces in many assistants let you store documents and standing instructions for a topic; set one up for each repeat task.
- Put the most important instructions at the start and repeat critical rules near the end of long material.
- Label each pasted item clearly ("POLICY", "EXAMPLE", "NOTES") so the assistant knows its role.
- Context window: the limited amount an assistant can hold at once. Very long chats may drop or compress earlier material, so restate key points or start a fresh chat with a summary.
- Connecting AI to your files through approved connectors (for example using MCP, an open standard for linking AI to tools and data) is context engineering at scale, and needs the same care about access.

Key words

- **Context:** all the information the assistant can see when it answers.
- **Context window:** how much an assistant can hold in view at once.
- **Context engineering:** choosing and organising that information on purpose.
- **Context pack:** a reusable document with the background, examples and rules for one task.
- **Project:** a workspace in an assistant that keeps documents and instructions together.
- **Saved instructions:** standing guidance the assistant follows in every chat.