Give Claude context, not a giant prompt

Here's how to set up what I showed in my video. Instead of writing a page of instructions, you give Claude four things before it starts: the goal, room to look around, a chance to interview you, and examples of what good looks like. You won't write any code.

What you need

- A paid Claude plan. Claude Code comes with Pro, Max and Ultra (and with Team and Enterprise). It isn't in the free plan.
- The Claude desktop app. The steps run in **Claude Code**, which is the **Code** tab of the app, not the regular chat.
- One real job you want done, with a folder of files for it.

What “context” means here

Anthropic wrote that after a few years of prompt engineering being the focus of attention, a new term has come to prominence: context engineering. They call it the natural progression of prompt engineering. Their post is about building AI agents, and it’s mostly about what information an agent sees at each step.

What carries over to you is one idea from that post. Don’t pack a prompt with hardcoded, brittle rules. Anthropic says that makes it fragile. But don’t be vague either: they say vague guidance fails to give the model concrete signals. Aim for specific enough to guide Claude, and flexible enough to leave the steps to it.

So this isn’t “short prompts beat detailed ones”. Anthropic’s prompting page still says: “The more precisely you explain what you want, the better the result.” What changes is what you’re precise about. You’re precise about where you’re going and what good looks like, and you leave out the list of steps.

Step 1: open Claude Code

1. Download the Claude desktop app from claude.ai/download, install it and sign in.
2. Click **Code** at the top of the sidebar.
3. Make a new folder for the job, for example “First job” in your Documents, and put the files for the job in it.
4. In the row under the message box, choose **Local**, then click the folder control, choose **Open folder...** and pick that folder.

Keep an eye on what Claude does. If you’d rather approve each step, choose **Manual** in the same row.

Step 2: give it the goal

Tell Claude where you’re trying to get to, not the steps to get there. Then say what “done” looks like, in a way Claude can check.

Anthropic’s Claude Code page puts it this way: make sure Claude knows what done looks like and can check its own work. Give it something it can run that

comes back pass or fail, and it can keep going without you watching. It also says that if you describe the symptom and say what “fixed” looks like up front, you usually save yourself a round of corrections later.

Paste this and fill in the brackets:

```
My goal: [the result you want]. It's done when
[something Claude can check, like a total that
matches, a list with nothing missing, or a
file that exists]. Don't change or delete the
original files.
```

Leave out how to do it. Claude will work that out in the next step.

Step 3: let it look around first

Anthropic’s Claude Code page says Claude starts every task by reading your code, and that it finds your file layout and conventions usually faster than

you could explain them. So you don't need to describe your files. Ask it to look before it starts:

```
Before you start, look through the files
in this folder and tell me what you found
that matters for this job. Don't change
anything yet.
```

If the job needs information from outside your folder, Anthropic's prompting page has two tips for research: give clear success criteria for what a good answer is, and ask Claude to check information across more than one source. You can add a line like:

```
If you need facts from outside this folder,
search for them, check them across more than
one source, and tell me where each came from.
```

Step 4: let it interview you

When you don't know exactly what you want yet, Anthropic says you don't have to work it all out before you prompt. Ask Claude to interview you instead. Their example is: "Interview me until you could write the spec, then write it to SPEC.md."

Paste this:

```
Interview me until you understand what I actually want for this job, then write it down in a file called SPEC.md. Ask one question at a time.
```

Anthropic says Claude comes back with multiple-choice questions about the parts you haven't pinned down, and keeps asking until the answers add up to a spec. Answer them, then read `SPEC.md` and correct anything wrong. Their advice after that: start a fresh session to do the work from the file, so it begins with a clean context and a written definition of done.

Step 5: show it what good looks like

Paste in a few real examples of the standard you expect: a past report, an email you were happy with, a spreadsheet laid out the way you want. Anthropic says examples are one of the most reliable ways to steer Claude's output, and recommends 3 to 5 for best results. They should be:

- **Relevant:** close to the job you're doing now.

- **Diverse:** different enough from each other that Claude doesn't copy an accident they share.

Anthropic's engineering post says to pick a few diverse, canonical examples, and not to stuff a laundry list of edge cases into the prompt. It calls examples "the 'pictures' worth a thousand words."

```
Here are examples of what good looks like for this job: [paste or attach 3 to 5]. Match their standard. Tell me what you noticed they have in common before you start.
```

Not sure your examples are good enough? Anthropic says you can ask Claude to evaluate them for relevance and diversity, or to make more based on yours.

Put it together

Send the four in this order, in one session:

1. The goal and what done looks like (Step 2).
2. "Look through the files first" (Step 3).
3. The interview (Step 4).

4 Your examples (Step 5).

One thing to drop: shouting

If you have old prompts full of capitals and “you MUST”, Anthropic says newer Claude models are more responsive to the system prompt and might now overtrigger on that. Their fix is to soften the language: where you’d write “CRITICAL: You MUST use this tool when…”, write “Use this tool when…” instead.

And give the reason. Anthropic says that explaining why something matters gives Claude more information about your goals and helps it deliver more targeted responses.

Written on 12 November 2026 from Anthropic’s engineering post “Effective context engineering for AI agents”, Claude Code’s best-practices page and Anthropic’s Claude prompting best-practices page. I haven’t run these steps myself while writing this guide. Claude changes fast: if a menu has moved, look for the same words in the app.