

CORNERSTONE AI · FREE RESOURCE

GET CLAUDE ONTO YOUR COMPUTER

Two prompts that move you off the chat window and onto Claude Code, without writing a line of anything.

2

PROMPTS

20

MINUTES

0

CODE WRITTEN

WHAT THIS ACTUALLY IS

If you use Claude or ChatGPT every day, you already know the ceiling. You explain your business, you get a good answer, you close the tab, and tomorrow you explain it again. The tool cannot see your files, so **you become the courier**: copying things in, copying answers back out, all day.

Claude Code is the same Claude, running in a folder on your own computer. It reads what is in that folder and writes back into it. That is the entire difference, and it is a much bigger one than it sounds.

The name is the worst thing about it. You will not write code and you do not need to understand any.

HOW TO USE THIS GUIDE

- 1 Open a **normal Claude chat**, the one you already use, at claude.ai or in the app.
- 2 Copy **Prompt 1** and paste it in. That chat becomes your setup guide and walks you through everything, one step at a time.
- 3 When it tells you to, copy **Prompt 2** and paste it into Claude Code.
- 4 When Prompt 2 has finished, go back to the first chat and reply with the single word **DONE**.

Keep this document open the whole way through. You need both prompts and they go in two different places.

The three pages after this one are the parts people get stuck on, written out in advance. The setup guide covers them too, so you can read them now or skip straight to Prompt 1 and come back when you need them.

If you get stuck at any point, say so in the first chat. That is what it is there for. It knows what you are trying to do and it will work it out with you.

ONE FOLDER IS ONE PROJECT

Claude Code does not sit on your computer the way an app does. It runs **inside one folder**, and whichever folder you point it at becomes its entire world.

It can read every file in that folder and every folder inside it, with nothing to attach and nothing to upload. It cannot see a single thing outside it. Not your desktop, not your downloads, not the folder next door.

That is the whole reason every instruction you will ever read says **open Claude Code in a new folder**. The folder is not where the work gets stored. The folder *is* the project.

WHAT THAT MEANS IN PRACTICE

- **One folder per thing.** Your business gets a folder. A big client gets their own. A one off gets its own. They never see each other, which is exactly what you want.
- **To switch projects, you close it and open it somewhere else.** Nothing carries over and nothing leaks across. There is no project picker to get wrong.
- **Everything it learns, it writes into the folder.** That is why it still knows your business next Tuesday. The memory is sitting on your own disk, in files you can open and read.
- **If it cannot see a file, move the file in.** Drag it into the folder. That is the entire fix, every time, forever.

Pick the folder the way you would pick a filing cabinet, not a scratchpad. Somewhere sensible, named after your business, that you will still be able to find in six months. Everything else in this guide goes inside it.

INSTALL GIT FIRST

Git is the thing that lets Claude keep track of every change it makes, so any of it can be undone. That is all you ever need to know about it, and I will not mention it again.

The setup guide will tell you to do this. It is written out here so you can get it done in the next two minutes and stop thinking about it.

ON WINDOWS

1. Go to git-scm.com/download/win and download the 64 bit setup file.
2. Open the file that lands in your downloads.
3. Click **Next** on every screen, then **Install**. None of the options on the way matter.
4. When it finishes, open the Start menu, type **Terminal** and press Enter.

Check it worked: type `git --version` and press Enter. A version number means you are done.

ON A MAC

1. Press **Command and Space**, type **Terminal** and press Enter. A small window full of text opens.
2. Type `git --version` and press Enter.
3. A box appears offering to install the developer tools. Click **Install** and agree.
4. Let it run. It takes a few minutes and there is nothing to click.

Check it worked: type `git --version` again. A version number means you are done.

If anything on this page throws an error, you are not stuck. Select the whole error message, copy it, paste it into your Claude chat and say: *fix this and explain what went wrong in plain English.*

That works at every step of this guide, not just this one. An error is not a wall. It is just the next thing to paste in.

TWO SETTINGS TO CHANGE

Out of the box, Claude Code stops and asks your permission before it touches a single file. Sensible, and exhausting when it is about to write twelve of them. These two changes are the difference between watching it work and letting it work.

AUTO MODE, SO IT STOPS ASKING

Press **Shift and Tab** together. The mode you are in is named at the bottom of the window, and each press moves you to the next one. Stop when it says auto accept. From then on it writes and edits files without checking in.

Keep pressing and you will reach a mode that stops asking about everything, including running commands on your machine. Leave that one alone.

WHEN TO SWITCH IT BACK OFF

One rule: **watch it the first time, let it work every time after.**

- **Let it work** when it is writing, drafting, editing, reading or organising files. Any of that can be undone. That is what you installed Git for.
- **Watch it work** when something real is on the other end: sending an email, moving money, publishing anything, deleting anything, touching a system a customer can see. Take the permission prompts back and read them.
- **The mode is not a commitment.** Shift and Tab moves you back to asking mode mid job, whenever you want it.

DYNAMIC WORKFLOWS, SO LONG JOBS FINISH FASTER

This lets Claude split a big job across several helpers working at the same time instead of grinding through it alone, which makes long jobs quicker and more thorough. Type `/config`, find dynamic workflows in the list and switch it on. That is the whole change.

YOUR SETUP GUIDE

COPY EVERYTHING BELOW

Paste into: a normal Claude chat

You are going to walk me through setting up Claude Code on my computer.

I run a business. I am not a developer, I have never used a terminal, and I do not know what most technical words mean. Assume I know nothing and do not assume I will recognise a term just because it is common.

How to work with me:

- One step at a time. Give me a single instruction, then stop and wait for me to tell you it worked. Do not give me a numbered list of ten things to go and do.
- Explain a word in plain English before you use it, ideally in terms of running a business.
- If I tell you something is not working, do not repeat the same instruction. Ask me what I can actually see on my screen, and work it out from there.
- Do not be precious about it. If I take a wrong turn, get me back on track without a lecture.

Before anything else, ask me whether I am on Windows or a Mac, and wait for my answer.

Then get me through these seven milestones, in this order:

1. Install the Claude desktop app. Claude Code is part of it, so there is nothing separate for me to go and buy.
2. Install Git. Say what it is in one sentence and then never mention it again: it keeps a history of every change made in a folder, so nothing is ever permanently lost and I can always go back. Give me the steps for my machine only, and the one thing to type to check it worked. If a step throws an error, tell me to copy the whole message and paste it back to you rather than trying to read it myself.
3. Have me create a new folder for my business, somewhere I will find it again. Help me decide where to put it and what to call it. Explain while I do it that Claude Code works inside one folder at a time: it sees everything in the folder I open it in and nothing outside it, so this folder is the project.
4. Show me how to open that folder in Claude Code.
5. Get me off the constant permission prompts and into auto mode. Explain it first: out of the box it asks before every single change, which is sensible but exhausting when it is about to create a dozen files, and auto accept lets it write and edit files without asking. Walk me through switching to it, then tell me to leave the mode after that one alone, because it stops asking before it runs commands on my machine too. Ask me what I can see at the bottom of my screen rather than assuming which version I have. Then give me the rule for switching back: let it work when it is writing and editing files, watch it work when something real is on the other end, like sending an email, moving money, publishing or deleting.
6. Turn on subagents. Explain what one is first: instead of working through a long job alone, Claude can send several helpers at different parts of it at once and bring the results back, which makes big jobs faster and more thorough. Then walk me through opening settings with the `/config` command and switching on dynamic workflows. Read what is actually on my screen with me rather than telling me what the option is called, because it moves.
7. Tell me to paste my second prompt into Claude Code. I already have it saved, so you do not need to write it for me. Make it clear that it goes into Claude Code, not back into this chat.

Then stop and wait for me.

Before I go off to do the last step, tell me clearly: when the second prompt has finished running, I must come back to this chat and reply with the single word DONE.

When I reply DONE, explain three things, still in plain English, with no jargon:

- What I now have that I did not have this morning. Be concrete about the folder and the files in it, and what they are for.
- How to use it day to day. Give me three specific things to try this week, using my own business as the example, not a generic one.
- Why this is different from typing into a chat window. Be honest about what it is not good for, as well as what it is.

TEACH IT YOUR BUSINESS

Paste this into Claude Code, once the first chat tells you to. Answer the interview properly. This is the part that does the actual work.

COPY EVERYTHING BELOW**Paste into: Claude Code**

Set this folder up as the working home for my business, so you and I can work in it together from now on. I am not a developer, so explain anything you need me to understand in plain English.

Do this in three parts, in order.

Part 1. Interview me.

Before you create a single file, interview me properly about my business. Ask one question at a time and wait for my answer. Push where an answer is thin, and ask the obvious follow-up rather than moving on.

Cover at least:

- What the business does, in the words a customer would use, and who it is for
- Who I turn away, and why
- What I sell, what it costs, and what changes the price
- How I actually work: what a normal week looks like, what tools I use, who else is involved and what they do
- How I talk. Ask me to paste in two things I have written that sounded like me
- My rules: what I always do, what I never do, and what I would never say
- What is on my plate right now, and what I am trying to get to

Keep going until you could brief a new hire on my business without me in the room. Then tell me you think you have enough, and ask what you have missed.

Part 2. Write it down.

Create a folder called context. Write what you learned into separate plain-English files inside it, one per subject, so I can open and edit any one of them later without having to read the rest.

Write them for a person to read, not for a machine. Then show me what you wrote and let me correct it.

Part 3. Set the standing rules.

Create a file called CLAUDE.md in the top level of this folder. In it, instruct yourself to:

- Read everything in context before answering anything, at the start of every session, without being asked
- Keep this workspace tidy. If files are piling up loose in the top folder, or a folder is turning into a junk drawer, say so and propose a better structure before it gets worse rather than after
- Ask me before creating anything I did not ask for
- Write in my voice, using what you captured about how I talk
- Never invent a number or a fact about my business. If it is not in context and I have not told you, ask me

When all three parts are done, tell me plainly what you built, where it is, and what each file is for.

Then tell me to go back to my Claude chat and reply DONE.

THE RESULT

WHAT YOU END UP WITH

A folder on your own computer with your business written down inside it, in files you can open and edit yourself. Not a profile buried in a settings screen. Plain text, on your machine, that you own.

```
your-business/  
  CLAUDE.md      the standing rules Claude reads first, every time  
  context/       your business, in plain English  
    what-we-do.md  
    clients.md  
    pricing.md  
    how-i-work.md  
    voice.md  
    rules.md
```

From then on you stop briefing it. It already knows. You open that folder, say what you want, and it works from the same context every single time.

A FEW HONEST NOTES

- **This is not a coding tool.** You will not write code and you do not need to understand any. The name is genuinely the worst thing about it.
- **It is the same Claude.** Same model, nothing extra to buy. The difference is where it runs and what it can see.
- **Nothing here leaves your machine** except what you send to Claude in the normal way, exactly as it does when you use the chat.
- **The context is a first draft.** It gets better every time you correct it. That is the point of it being editable text rather than something locked in a settings screen.
- **Give the interview real answers.** A thin interview produces thin context, and thin context is why people say AI gives generic output.

If it breaks, or the setup guide sends you in circles, email me and I will unstick you: cristo@cornerstone-ai.pro

There is no call attached to this and nothing to buy. Run it, get set up, never speak to me. That is a completely fine outcome.