



THE MYOS SYSTEM

Make AI work like it actually knows you.

Five plain folders that turn any AI from a clever stranger into a team that gets you, your work, and how you like things done.

No code

No terminal

Any AI



WHY YOUR AI FEELS GENERIC

Your AI is brilliant. It also has amnesia.

Open a new chat and you re-explain who you are, what you're working on, how you like things. So it answers like it's meeting you for the first time — because it is. You're not getting help. You're onboarding a new hire, over and over.

THE FIX IS ALMOST TOO SIMPLE

It can't remember. But it can read.

Put your context into a few plain files and point your AI at them before each task. Now it shows up already knowing you — your voice, your projects, your standards. The files are the memory it never had.

Five folders. One simple question each.

Keep them on your computer. Point your AI at them. That's the system.

1 Core

Who you are and how you work — your identity, your voice, your rules. Start here.

[core/about-you.md](#)

2 References & Playbooks

What you know and how you do it — your knowledge plus reusable step-by-step playbooks.

[playbooks/launch.md](#)

3 Output

Your projects and the work they produce — one folder each.

[output/your-project/](#)

4 Notes

What the system learns over time — what worked, what didn't, project by project.

[notes/lessons.md](#)

5 Agents

Your AI team — named roles, each with a job. A writer, a researcher, a strategist.

[agents/writer.md](#)



WHAT YOU PUT IN, WHAT YOU GET OUT

Plain files in. A team out.

YOU FEED IT

Plain files you own — on your machine, not locked to any single tool. Your context, your voice, your standards.

IT GIVES BACK

A team that runs without you, reuses your playbooks, and gets sharper every week.

Context

Memory

Consistency

Autonomy

Leverage

Compounding

Ownership

THE ONLY HABIT THAT MATTERS

Point your AI at MYOS before every task.

That's the whole discipline. Do it, and your AI stops being a stranger — for good.



HOW THE MEMORY WORKS

Read in. Lessons out.

Two moves make the whole system. Before a task your AI reads your files. After a task it saves what it learned. The loop is what makes it get smarter.

READ — BEFORE A TASK

It loads **core/** plus the notes and references that matter, and shows up already knowing you.

SAVE — AFTER A TASK

It adds one line to **notes/lessons.md** — what worked, what to change. Next time, it remembers.

WHERE SAVING IS AUTOMATIC — AND WHERE IT'S ON YOU

In tools that can write to your files (like Claude Code or Cowork), your AI saves on its own. In a normal chat (ChatGPT, Claude, Gemini) it can **read** your files but not change them — so you paste the line in. Either way, one habit makes it stick: end a task by asking **"What's one line for lessons?"**

MAKE IT AUTOMATIC IN YOUR TOOL

ChatGPT	Make a Project and add your MYOS files to it — new chats read them automatically.
Claude (web / app)	Make a Project and add MYOS to its knowledge.
Claude Code / Cowork	Keep a CLAUDE.md that says "read MYOS first." Fully automatic — it reads and writes.
Any other AI	Start the chat by pasting core/about-you.md.

FROM ZIP TO WORKING IN FIVE MINUTES

Set up MYOS today.

1 Unzip the kit

Open [myos-starter-kit.zip](#) and keep the MYOS folder somewhere on your computer.

2 Fill in Core first

Open [core/about-you.md](#) and replace the brackets with your own words. That one file does most of the work.

3 Add your rules

Do the same in [core/rules.md](#) — how you like things done.

4 Point your AI at it

Wire it the way that fits your tool (previous page). Simplest of all: tell your AI **"Read my MYOS files first, then start."**

5 Let it learn

After a task, add one line to [notes/lessons.md](#) — what worked, what to change. The system gets smarter every week.

KEEP BUILDING

claudedicted.com

More free builds, beginner walkthroughs, and the everyday way into AI.

