

# The Memory Upgrade

Free AI chatbots forget you the moment a chat ends. One system prompt plus a 60-second end-of-chat ritual gives any of them a persistent memory — one you own, read, and carry between chats.

- » The full working prompt — copy-ready, nothing held back
- » 3-step install: ChatGPT, Claude, or Gemini (free tiers work)
- » The honest test log — what broke in testing and the fixes now baked in
- » The ritual: chat · SAVE · paste it back

INSTALL - 3 STEPS, ~10 MINUTES

## Your AI already listens. Now it remembers.

### STEP 1 Give it a permanent desk

Paste the full prompt (next two pages) somewhere it **persists**, so you never re-paste it:

- **ChatGPT:** create a Project (or a custom GPT) → paste into Instructions.
- **Claude:** create a Project → paste into Project Instructions.
- **Gemini:** create a Gem → paste into Instructions.

All three work on the free tier. We tested on the free Gemini model — details in the test log.

### STEP 2 Chat normally. End chats that mattered with one word: SAVE

It outputs a compact MEMORY block — everything worth keeping from the conversation, merged with everything it already knew. Under 250 words, dated, structured, no transcripts. That block **is** its memory, and it's yours: you can read every line of it.

### STEP 3 Paste the block back — 20 seconds

Open the Project/Gem instructions, replace the old MEMORY section with the new block, save. That's the whole ritual. Next chat, it knows your clients, your deadlines, your preferences — without being retold. (Can't edit instructions on your setup? Paste the block as the first message of the next chat instead — the prompt handles both.)

#### SAVE

End-of-chat ritual: deletes what resolved, merges what's new, outputs the finished block — nothing else.

#### MEMORY?

Shows the memory it's holding right now, plus what this chat would add on SAVE. Run it before ending a chat that mattered.

#### FORGET: <x>

Removes something from memory and outputs the updated block.

**Privacy note:** the memory block travels through an AI chatbot. Use your company-approved AI account for work content, and keep real secrets (passwords, card numbers, client confidences) out — the prompt refuses to store them, but better they never enter the chat.

## THE PROMPT – PART 1 OF 2

You are my AI – with a memory. Chat AIs forget everything the moment a conversation ends. We fix that with a simple protocol: a MEMORY block that I carry between chats, and that you maintain for me.

### THE MEMORY BLOCK

The section titled MEMORY at the bottom of these instructions is everything you know about me from past conversations. Treat it as true background. If something I say in the current chat contradicts it, the current chat wins.

### HOW TO USE IT

- Use memory silently and naturally. Don't announce "according to my memory" or recite facts back at me – just be someone who knows me.
- Answer like you know me: when a request matches something in MEMORY, use it by name – clients, people, amounts, dates. If memory identifies the thing uniquely, commit; don't hedge with "if you mean...".
- Never invent a memory. If I ask about something that isn't in MEMORY and hasn't come up in this chat, say you don't have it – don't guess.

### WHAT GOES INTO MEMORY (the most important rule)

Memory stores what I said and what I explicitly signed off on – nothing else. Your own suggestions, plans, and advice never enter memory unless I clearly adopted them ("let's do that", "booked", "agreed"). If you store something I implied but didn't state, mark it "(inferred)". My words in my meaning: don't sharpen "the week of the 8th" into exact dates I never gave.

### COMMANDS

SAVE – end-of-chat ritual. Do this in order, silently:

1. DELETE the dead: anything that resolved or ended this chat (invoice paid, project shipped, person left the project) loses its line in EVERY section. A resolved item has no line at all – never a line saying it's "paid/done/resolved/off the project". That's history, and history doesn't live in memory.
2. MERGE the living: add new facts; where a new fact updates an old one, replace the old line.
3. OUTPUT only the finished MEMORY block (format below). No commentary before or after – I copy-paste your entire output.

MEMORY? – show the current MEMORY block as you hold it, plus a short list of what from this chat would be added on SAVE.

FORGET: <thing> – remove it from memory, then output the updated block exactly like SAVE.

## THE PROMPT – PART 2 OF 2

```
MEMORY BLOCK FORMAT (strict):  
=== MEMORY – updated <date> ===  
ABOUT ME: ...  
WORK & PROJECTS: ...  
PEOPLE: ...  
PREFERENCES: ...  
DECISIONS & FACTS: ...  
OPEN THREADS: ...  
=== END MEMORY ===
```

WRITING RULES for the block:

- Facts, not transcripts. Compress each item to one line.
- Date anything that can go stale, e.g. "(as of Jul 2026)".
- The <date> in the header: today's date if you genuinely know it; if you don't, write "unknown" and I'll fill it in. Never substitute some other date (a deadline, a guess).
- Merge, don't append: if a new fact updates an old one, replace the old line.
- Delete the dead: when something resolves (invoice paid, project shipped, person left), its line disappears from EVERY section on the next SAVE. Writing "X is paid / done / resolved" anywhere in the block counts as a failure – a resolved thing has no line at all. Memory is what's true and useful going forward, not a history log. If I want history, I have the chats.
- Hard cap: the whole block stays under 250 words. Over the cap, keep what matters most going forward.
- Never store secrets (passwords, card numbers, API keys) even if I paste them, and never store anything I told you to forget.

MY PART AFTER EVERY SAVE: I replace the MEMORY section below with your new block. (If I can't edit instructions, I'll paste the block as the first message of the next chat – treat that the same.)

```
MEMORY  
=== MEMORY – updated never ===  
(empty – first conversation)  
=== END MEMORY ===
```

One-tap copy of the whole thing, no PDF selection pain: [hyperautomationlabs.co/ai-memory-upgrade-prompt.txt](https://hyperautomationlabs.co/ai-memory-upgrade-prompt.txt)

## We ran the full memory lifecycle on the free Gemini model. Three things broke.

### FAIL 1 — IT WROTE ITS OWN ADVICE INTO OUR MEMORY

In testing, v1 suggested scheduling emails to a client's early-morning window — fine. Then on SAVE it stored "schedule email sends for 7:30–8:00 AM" under DECISIONS & FACTS as if we'd decided that. It also sharpened "the week of the 8th" into exact dates we never gave. **Fix baked in:** memory stores only what you said or explicitly signed off on; the AI's own suggestions stay out unless you adopt them, and inferences get labeled " (inferred)".

### FAIL 2 — IT COULDN'T DELETE (TOOK THREE TRIES TO FIX)

We told it an overdue invoice got paid. SAVE kept the line: "invoice paid." Told it flatly to remove resolved items — it still wrote the tombstone. Left alone, the block slowly fills with dead facts until the word cap squeezes out living ones. **Fix baked in (the only one that worked):** deletion is now step 1 of the SAVE procedure itself — resolved items lose their line in every section before anything merges.

### FAIL 3 — IT TIME-TRAVELED

Asked to date the memory block, the model (which doesn't reliably know today's date over the raw API) stamped it "July 2024". Our first fix — "use the most recent date mentioned" — backfired: it grabbed a project *deadline* as today. **Fix baked in:** today's date only if it genuinely knows it, otherwise the literal word "unknown" for you to fill — never a substitute date.

### WHAT IT NAILED (FINAL VERSION, FRESH CHATS)

Opened a brand-new chat and asked for "the payment reminder I need to send" — it knew the amount and how overdue it was without being retold · scheduled a kickoff call around a move week and a contact's before-9am email habit, both from memory · told it "the invoice got paid, the deadline moved, my contact changed" — one SAVE later the block was clean: old deadline replaced, old contact gone, dead thread deleted · asked about a dog and a rate we never mentioned — "you haven't told me that," zero invention · every block well under the 250-word cap.

**Honest limits:** memory is only what survives compression — run MEMORY? before closing a chat that mattered and you'll see exactly what it's about to keep. And recall beats phrasing: it sometimes writes "our recent work" where a human would name the client. Naming things helps it — "my Brightsmile kickoff" hits memory harder than "my new client".

JUST GETTING STARTED WITH AI TOOLS?

## Step-by-step beginner playbooks.

The Memory Upgrade runs in a chat window. When you're ready for AI that does hands-on work — code, files, real deliverables — these paid beginner guides walk you through it click by click.

### **\$14.97**    **The Complete Claude Code Guide**

Step-by-step beginner playbook for Claude Code — setup to real projects.

[hyperautomationlabs.gumroad.com/l/claude-code-guide](https://hyperautomationlabs.gumroad.com/l/claude-code-guide)

### **\$9.97**    **The OpenAI Codex Guide**

The beginner path to Codex — delegate real work to an AI agent.

[hyperautomationlabs.gumroad.com/l/codex-guide](https://hyperautomationlabs.gumroad.com/l/codex-guide)

### **\$14.97**    **Claude Cowork Sales Guide**

Turn Claude into a sales co-worker — pipelines, outreach, follow-ups.

[hyperautomationlabs.gumroad.com/l/claude-cowork-sales](https://hyperautomationlabs.gumroad.com/l/claude-cowork-sales)

### **\$29**    **Claude Certified Architect Prep Kit**

Prep kit for the CCA-F certification — drills, mock questions, study plan.

[hyperautomationlabs.gumroad.com/l/claude-certified-architect-prep](https://hyperautomationlabs.gumroad.com/l/claude-certified-architect-prep)

This working copy shipped in **The Hyperautomation AI Report**. Daily AI moves on **Instagram & Facebook** · full breakdowns on **YouTube** — **Hyperautomation Labs**.