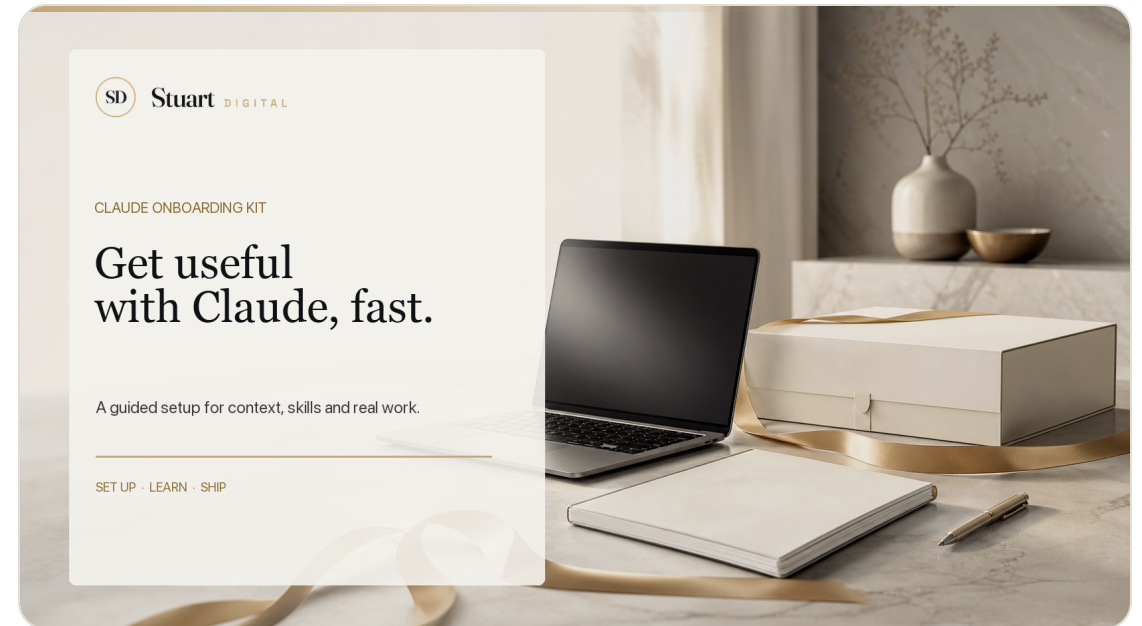


— STAGE 2 · BEGINNER

Wire it into your world.

Stage 1 got Claude working on a file you handed it. Now connect it to where your work actually lives — and learn the habits that keep the answers good and the cost controlled.



Short, practical, and back to work.

01

Recap

02

The vocabulary

03

Connectors

04

Projects & habits

05

What it costs

No new skills today. This is the easy one: settings, connections, and three habits.

What lesson one gave you.

1 • You set Claude up

A Claude folder, a Cowork project, the setup pack dropped in and run once.

2 • The document-to-deliverable loop

Hand Claude a real file → use the right skill → check, refine, deliver.

3 • Your five skills

document-review, meeting-minutes, mom-priority-distiller, executive-summary-report, email-drafter.

4 • You picked your three tasks

The repetitive, low-value, annoying jobs — the first to hand over.

5 • The habits

The small routines that keep the work sharp and the cost down.

Talk it through: Claude knows your firm, your toolkit's loaded, and the loop is yours. Stage two plugs it into where your work actually lives.

The words that make it feel technical aren't technical.

Prompt

What you ask Claude to do.

Model

The Claude brain handling the job.

Token

A small chunk of text Claude reads or writes.

Context window

How much Claude can hold in mind at once.

Artifact

A document, page, or app built beside the chat.

Project

A named workspace with its own files and instructions.

Memory

Useful facts Claude keeps for later conversations.

Hallucination

A confident answer that is not grounded in fact.

Seven more words, then you're fluent.

Connector

A live link to an app you already use.

Skill

Reusable instructions for doing one job well.

Plugin

A packaged capability you can add to Claude.

MCP

Model Context Protocol — the common plug standard that lets any outside app, tool or data source connect to Claude. Every connector you just saw is built on it; MCP is the socket they all fit.

Cowork

The visual workspace for hands-on file work.

Code

The terminal workspace for building software.

Usage window

The rolling allowance shared across your work.

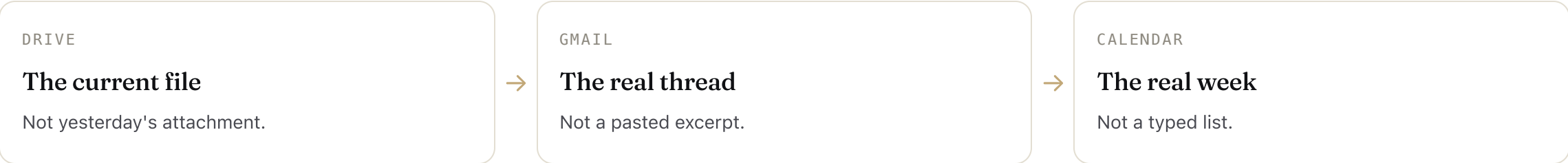
Vocabulary is a map, not a test. You only need enough to know where to click and what to ask.

Connectors — Claude reads your real work.



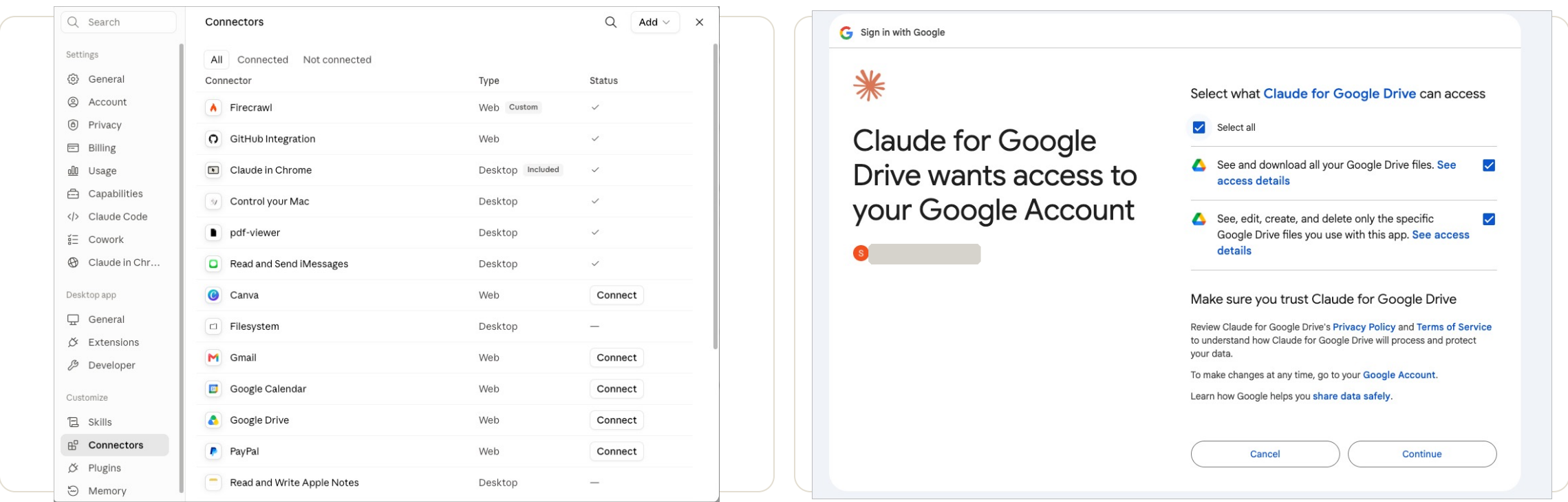
Stop pasting. Start connecting.

Turn on Google Drive and Claude can open the actual file. Turn on Gmail and it can find the actual thread. Connect Calendar and it can work from the real week — without another copy-paste.



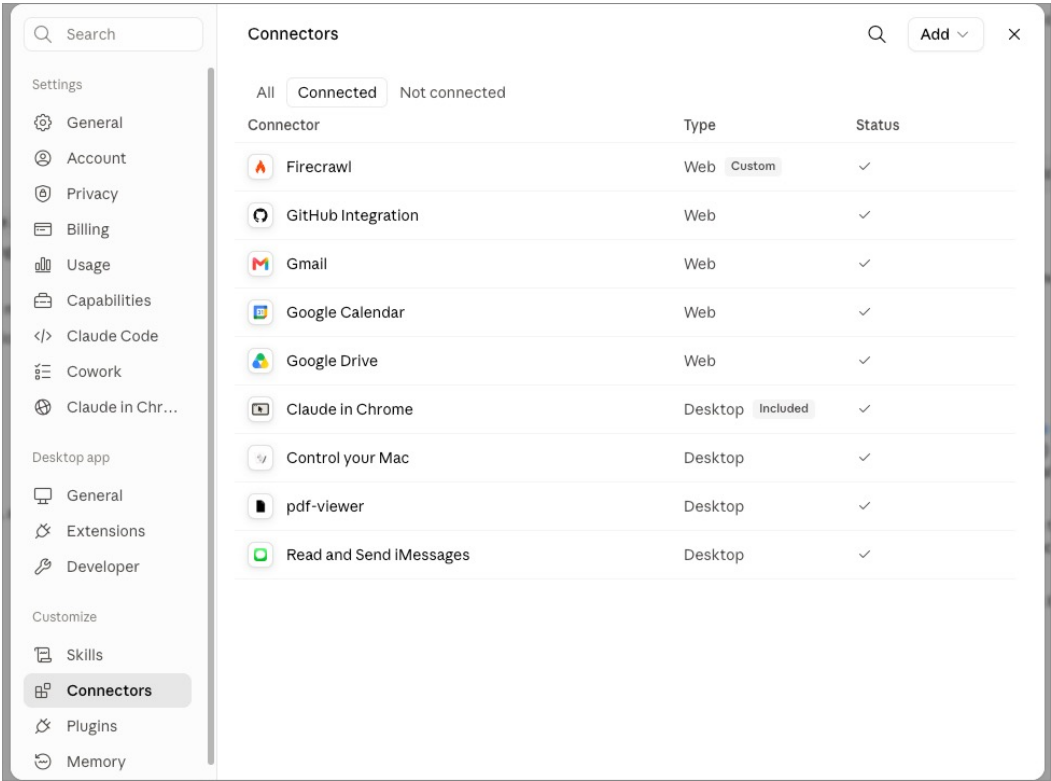
You stay in control. Settings → Connectors. Approve each connection, and switch it off when you do not need it.

One toggle, one permission.



Settings → Connectors → Google Drive → Connect. Claude asks Google for permission; you grant access.

Now it can see your inbox and your week.



Summarise the Gmail thread with the design team and draft a reply. Claude reads the real thread.

Claude where you already are.

The extension puts Claude beside the browser tab you're using: read this page, pull this table, or draft from this email without leaving it.

1

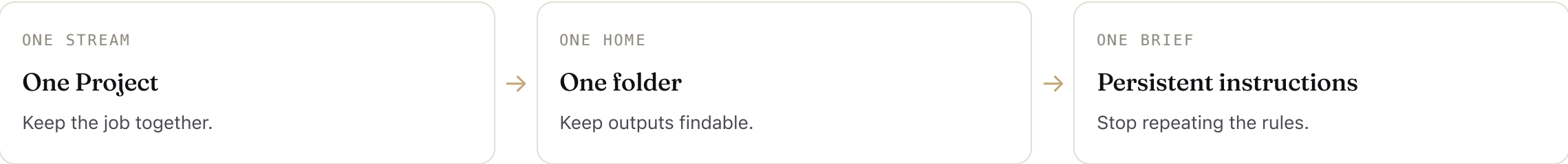
Try the useful version

Read this page. Give me the decision, the evidence, and what I need to do next.

Use it when the page in front of you is the source. Turn it off when it is not part of the job.

The project remembers so you don't repeat yourself.

A Project keeps its own files, instructions, and history. Point it at a real folder and everything Claude makes has a proper home.



Load it once, not every morning.

Project files are cached. Re-uploading the same BOQ into a fresh chat makes Claude re-read the whole thing and burns allowance for no gain.

Cold chat, every day

Upload again. Re-explain the job. Rebuild the context. Get a weaker answer while it catches up.

Project, once

Store the BOQ once. Ask against the same project all week. Better continuity, less waste.

Quality first: the Project that already knows your BOQ gives a better answer than a cold chat you re-fed.

Three habits that keep it good — and cheap.



One topic, one chat.

1

Start fresh for a new job

A long chat buries today's question under yesterday's work.

2

Connectors off when unused

Do not drag your inbox into work that never needed it.

3

Watch the usage bar

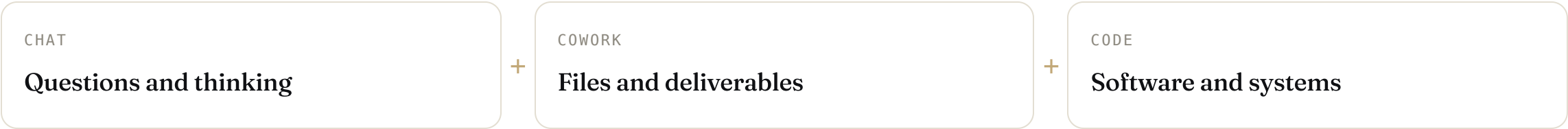
Know when to do the heavy job and when to wait.

Long chats make Claude worse, and you pay more for the worse answer.

Fresh chat = a clearer brief, a better answer, and less waste.

It's one allowance, not three.

Chat, Cowork, and Code draw from the same rolling allowance: the five-hour window and the weekly cap. Heavy use in one leaves less in the others.



Switching model does not reset it. Plan the heavy job for when the usage window is fresh.

What it actually costs.

START HERE · US LIST PRICE

\$0 **\$20**

Free · chat only Pro · Cowork + Code · \$17/yr

NEED MORE? · SAME ACCOUNT, HIGHER LIMITS

\$100 **\$200**

Max 5× · frequent use Max 20× · daily power use

\$20 covers regular use. Lean on it all day and you step up to Max — a choice, not a surprise bill. Cowork and Code are included in Pro, not extra.

Defend the quality first. Cleaner context produces better work. Cost is the second argument.

— WHAT YOU UNLOCKED

You're wired in.

Drive · Gmail · Calendar connected

Project pointed at a folder

One topic, one chat

Connectors off when unused

Load once, not daily

Next: stage 3 chains skills together on your own live project — and shows you how to build one for a job only you have.

— TAKE IT WITH YOU

Keep the settings card. See you at stage 3.

The deck and two-sided settings + habits card are on your personal pack page. There is no skills zip for stage 2 — settings and habits are the lesson.

Deck PDF

Settings + habits card

Scan to add Stuart to your contacts.

can't scan? training.stuardigital.io/card

