

A NO-CODE CRASH COURSE

• Verified June 2026

Skills & Connectors

Teach AI a task once. Give it hands to reach your real apps.

Turn your chat box from a vending machine into a coworker that already knows your standards.

2
features

5 tools

0 code
from you

The vending-machine problem

Most people use AI the way they use a vending machine. Walk up, type the request, take what falls out, walk away. **Tomorrow you type the whole thing again** — the same instructions, the same context, the same "no, our invoices look like *this*."

0

The machine remembers none of it. Every visit starts from zero.

THE ONE IDEA UNDER EVERYTHING

A chat tells AI what to do **this once** . A

Skill teaches it how **every time** . A

Connector gives it **hands** to reach your
real apps.

A working kitchen makes it concrete

CONNECTORS

The kitchen itself

The stove, the knives, the stocked pantry, the fridge full of **your actual ingredients** . Without it a cook can describe a meal but cannot make one.

SKILLS

The recipe cards

Step-by-step instructions for making a dish **your way** , the way your kitchen always makes it — so it comes out the same whoever is cooking.

Give a cook both, and they produce your dish — reliably, without you standing over them. **That is the whole game.**

0

1

PART ONE

The two upgrades

From a thing you type into, to a layer that knows your standards and reaches your tools.

From chat box to operating layer

"I keep re-explaining how."

The accountant pastes the same formatting rules every month.
The marketer pastes the brand voice into every chat. A repeatable task, done your way — only the input changes.

- That shape is a Skill

"I keep copy-pasting from another app."

The numbers live in an Excel file; last quarter sits in an email; the history is in a Drive folder. AI reasons brilliantly — if the information reaches it.

- That cost is a Connector

Put the two together and the chat box becomes a colleague who already knows your standards and can get to your files.

WHAT A SKILL ACTUALLY IS

A folder with a text file in it.

The required file is named `SKILL.md`.

It holds a **name**, a **description**, and your plain-English instructions. That's the minimum — **no code required to start.**

Loaded dynamically

AI doesn't carry every skill in its head. It keeps a short summary of what each one is *for*, and opens the full instructions only when your request matches.

→ *progressive disclosure*

Why dozens of skills can be installed without slowing anything down.

Safe access to the apps where your work lives

It runs on an open standard called **MCP**. You never build it or read it — you click *Connect*. Three facts matter.

1

It inherits your permissions

A connector can never reach anything your own account can't already open.

2

You choose read or write

"Search and summarize" is low-risk.
"Send" or "create" is a write action. Start read-only.

3

You turn them on per chat

Connecting an app makes it available; you still decide which a given conversation may use.

Four cousins, cleanly separated

FEATURE**WHAT IT IS****THE ONE-LINE TEST****Skill**

Reusable *how-to* instructions, loaded only when relevant

"I keep re-explaining *how* to do this."

Connector

Safe access to an external app or data source

"I keep copy-pasting from another app."

Project

A workspace whose files and rules load for *every* chat inside it

"These same files apply to *everything* here."

Custom instructions

Preferences applied broadly to *all* your chats

"I want this true everywhere, always."

In Claude.ai, skills fire automatically

You describe your task in plain language. AI checks it against the short **description** of every skill you've switched on, and loads the match on its own.

Ask "create a PowerPoint about Q3 results" and the PowerPoint skill activates — without you naming it.

When you want to force one

Everywhere: name it in plain English — "use my brand-voice skill."

Cowork & Office add-ins: type / to browse and pick.

Default is automatic ~90% of the time — which is why the description matters so much.

0 2

PART TWO

Use what already exists

Enormous value before you build anything —
switch these on in two minutes.

The skills that come built in

They all ride on one switch — **Code execution & file creation** in Settings. Flip it on, then just ask.

.docx

Word

Professionally formatted documents

.pptx

PowerPoint

Full slide decks from a description

.xlsx

Excel

Real spreadsheets with live formulas

.pdf

PDF

Creating and filling PDFs

Connecting your apps

- 1 Pick a service, click **Connect**, complete the sign-in.
- 2 Enable the connector **for this conversation.**
- 3 Ask for the thing — AI reaches in on its own.

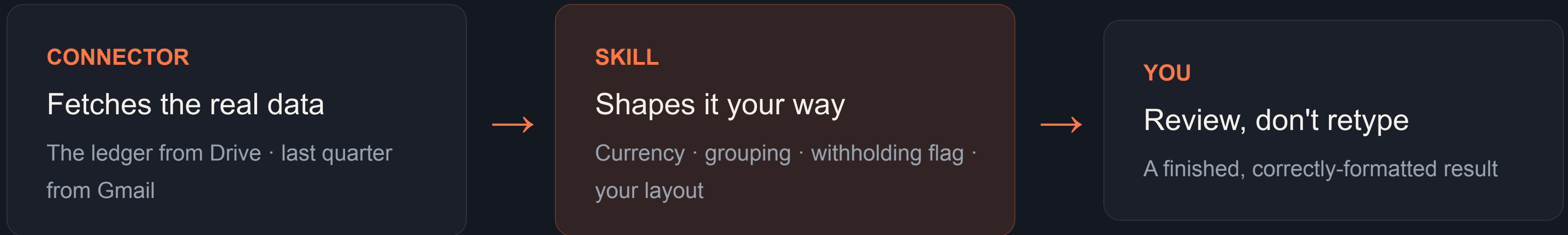
THE DOCTOR'S DEMO LOOP — FAKE DATA

"Make a fake patient intake form and **save it to my Drive** ."

Then: "Find it and summarize the meds into a **SOAP-style note** ."

One connector, both directions — AI wrote a file in, then searched and read it. No download, no copy-paste.

Skills + Connectors, working together



The accountant's monthly close: a ~~two-hour~~ ritual becomes a **two-minute** review.

Which do I need? Diagnose by friction

I re-explain **how** every time

• Skill

I copy data **from an app**

• Connector

I fetch real data **and** shape it my way

Both

I just want a **one-off** answer

Neither

Two cautions: a task you do once is just a good prompt. And a connector you don't need is **a door you've left open** — scope is safety.

03

PART THREE

Build your own skill

It sounds like it needs programming. It doesn't — you describe it, and AI writes the file.

THE FASTEST PATH

Let AI write it for you

Anthropic ships a skill for *making* skills, called **skill-creator**. You describe what you want in plain English; it asks a few clarifying questions, then generates a complete, correctly-formatted skill.

THE MINDSET

You are the client, not the contractor.

A good client doesn't lay bricks. They write a clear brief, check the result, and keep what they paid for. A skill is that, made permanent — commissioned once, reused forever.

Anatomy of a SKILL.md

```
--- ← frontmatter (always loaded) name:  
client-monthly-summary description: Prepares a  
monthly client  
    financial summary.  
Use when the user  
    asks for a "client summary" or  
    "monthly close".  
--- # Body — loaded only when matched ##  
Instructions 1. Format amounts in the reporting  
    currency, with separators.  
2. Group line items by expense head.  
3. Flag payments above the threshold.
```

1 Frontmatter

Name + description. The *only* thing always loaded — how AI decides relevance.

2 Body

The instructions, opened only when the description matches your request.

Rules: file named exactly SKILL.md · folder in kebab-case · no XML tags
· description under ~1024 chars.

The description is the whole game

AI reads *only* the description to decide relevance.

The formula: **what it does + when to use it + the exact phrases you'd actually say.**

VAGUE — NEVER FIRES

"Helps with reports."

"Brand stuff for marketing."

SHARP — FIRES EXACTLY WHEN IT SHOULD

"Prepares a monthly client summary. Use when the user asks for a **'client summary'**, **'monthly close'**, or **'month-end report'**."

"Writes social captions in our brand voice. Use when drafting **Instagram, LinkedIn, or X captions** ."

Fires too eagerly? Add a **negative trigger**: "Do NOT use for one-off questions — only full month-end reports."

Test, then iterate

A

Test triggering

Try the phrases that *should* activate it.
Try unrelated requests and confirm it does *not* take over.

B

Test the output

Run it on a real input, including one deliberately awkward case — a payment exactly on the threshold, a messy ledger.

C

Bring failures back

Hand the failure to skill-creator: "It double-counted reversals — net them out before grouping."

Never triggers → description too vague. Add the words you actually use.

Triggers wrongly → description too broad. Narrow it, or add a negative trigger.

04

PART FOUR

One skill, five places

Write it once. The recipe hangs in many kitchens,
even if each arranges its shelves a little differently.

A skill written once is portable

Claude.ai

Web & mobile — your main tool. Buttons and toggles.

Cowork / OpenWork

Desktop — acts on the real files on your computer.

Claude Code / OpenCode

Terminal — for people who work with code.

PORTABLE — WRITE ONCE, RUNS ACROSS TOOLS

The same `SKILL.md` even runs in OpenAI's Codex CLI and Google's Gemini CLI. Core portable; edges differ per tool.

NOT PORTABLE

ChatGPT's Custom GPTs and Gemini's Gems are handy — but locked to one product.

05

PART FIVE

Use it safely

Treat a skill from a stranger like a contract you're about to sign — and a connector like a key you're about to hand over.

Two real risks, one checklist

Malicious skills

A skill is a text file AI will follow. Hidden instructions can cause **prompt injection** or **data exfiltration**.

Over-broad access

Grant write access carelessly and AI can change or send things on your behalf — a wrong edit, an early email, a deleted record.

THE SAFE-USE CHECKLIST

- Install from **trusted sources** — the built-ins and official directory.
- Have AI **read a skill** before you enable it.
- Start connectors **read-only**; earn up to write.
- Scope to the **smallest folder** the task needs.

Five lines to carry out the door

- 01 A chat is one-time; a **Skill** is the repeatable how; a **Connector** is hands to reach your apps.
- 02 A Skill is a folder with a `SKILL.md` — name, description, instructions. No code to start.
- 03 In Claude.ai skills fire automatically — so the **description** is the most important thing you write.
- 04 Diagnose by friction: re-explaining how → Skill; copy-pasting → Connector; both → both.
- 05 Care scales with capability: trusted sources, read before enabling, start read-only.

Try this now — about thirty minutes

1

Trigger a built-in

Turn on file creation, then ask for a one-slide deck. Watch the skill fire without you naming it.

2

Connect one app

Connect Drive read-only and ask it to find a document and pull the three numbers that matter.

3

Build your first skill

Pick your most-repeated task. Ask skill-creator to build it. Pressure-test the description both ways.

Teach AI once. Connect it to your apps. **Stop starting from zero.**