FREE GUIDE · CLAUDE CODE + OPENAI CODEX

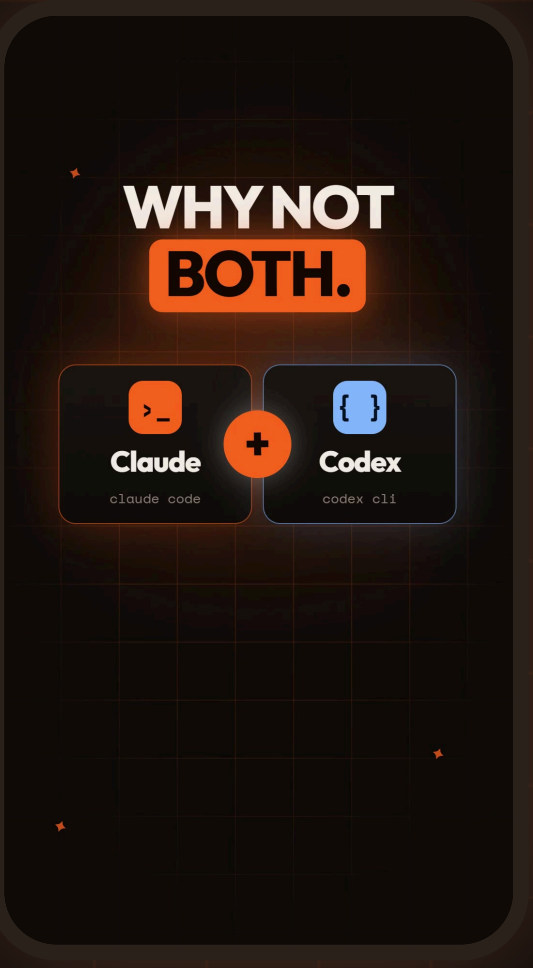
Claude or Codex? Use both.

One rules file that both AI agents follow. Claude builds, Codex reviews. A second model catches what the first one missed. Here's the exact setup, step by step.

⏱ About 15 min to set up

💻 Windows or Mac

📋 Copy-paste files included



The reel this guide goes with

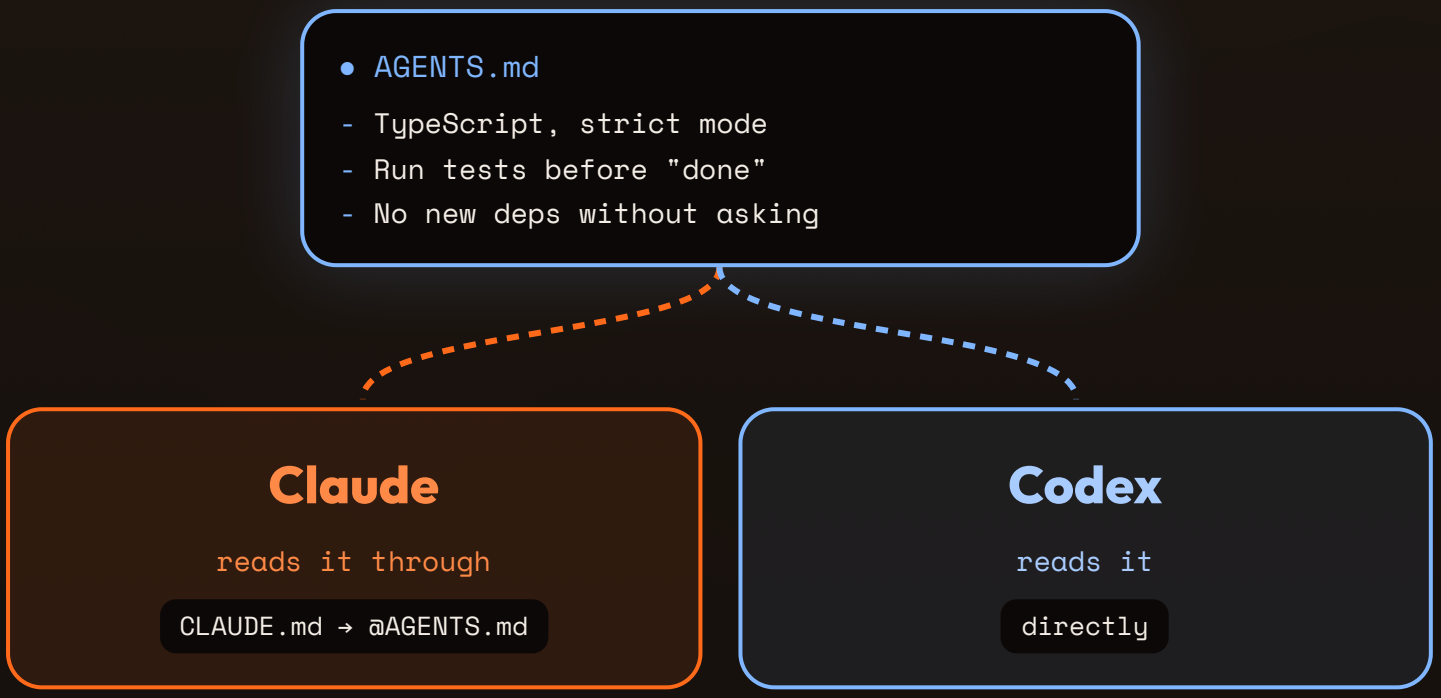
01 · THE BIG IDEA

Two tricks, one workflow

Trick 1: one rules file. Each AI agent reads its own instructions file. Codex reads **AGENTS.md** and Claude reads **CLAUDE.md**. Write your rules once in AGENTS.md, then make CLAUDE.md point to it. Now both agents follow the same rules, and you only update one file.

One rules file, both agents

Write the rules once. Both agents read them.



Trick 2: a built-in second opinion. When Claude finishes a change, it runs **codex exec**, which asks Codex to review the work without opening its chat window. Codex lists what looks wrong, and Claude fixes it. Two different models check each other's work.

The build → review loop

It runs by itself every time you ask for a change.



Who does what

Give each agent one clear job.

THE BUILDER

Claude Code

- Plans the change with you
- Writes and edits the files
- Runs your tests
- Fixes what the review finds

THE REVIEWER

Codex

- Reads the change with fresh eyes
- Spots bugs and missing checks
- Flags code that isn't used
- Lists problems, doesn't rewrite

Prefer it the other way round? Swap the roles. The shared rules file works both ways.

02 · BEFORE YOU START

What you need

Two AI plans and two free tools.

Claude Code PLAN

Needs a Claude plan that includes Claude Code. Install guide at code.claude.com/docs.

Codex CLI PLAN

OpenAI's coding agent. Sign in with a ChatGPT plan that includes Codex, or an OpenAI API key.

Node.js FREE

Installs Codex. Get the **LTS** version from nodejs.org.

Git FREE

Tracks what changed, so Codex knows what to review. Get it from git-scm.com.

03 · DO THIS ONCE PER PROJECT

Step by step

You'll paste a few commands into a **terminal**. On Windows, open **PowerShell** from the Start menu. On Mac, open **Terminal**. Paste each command and press Enter.

1 Install Claude Code and Node.js ~5 min

Install Node.js from nodejs.org (click LTS, then Next until it finishes). Install Claude Code with the one-line command on code.claude.com/docs. Then check Node works:

```
node -v
```

> **You should see** a version number like v24.x. If you get "not recognized", open a new terminal and try again.

2 Install Codex and sign in ~3 min

This installs OpenAI's Codex CLI. Then type **codex** once and follow the sign-in screen.

```
npm install -g @openai/codex
codex
```

> **After signing in,** close Codex (press Ctrl + C) and come back to the terminal.

3 Open your project and turn on Git ~1 min

Go to your project folder (replace the path with yours), then let Git start tracking changes:

```
cd path\to\your-project
git init
```

> **Already using Git?** Skip **git init**. If Git ever asks who you are, copy the two "git config" lines it shows you, add your name and email, and run them.

4 Write your rules once: AGENTS.md ~2 min

Create a file called **AGENTS.md** in your project folder and paste this in. Change the rules to fit your project. Codex reads this file automatically.

• AGENTS.md

```
# Project rules

- Use TypeScript in strict mode.
- Run the tests before saying a task is done.
- Don't add new packages without asking first.
- Keep changes small and explain what you changed.
```

> **Easiest way:** open Notepad, paste it, then File → Save as, set "Save as type" to **All files** and name it **AGENTS.md**.

5 Point Claude to it: CLAUDE.md ~2 min

Create **CLAUDE.md** in the same folder. The first line pulls in AGENTS.md, so Claude follows the same rules. The rest tells Claude to get a Codex review after every change.

• CLAUDE.md

```
@AGENTS.md

## Second opinion
When you finish a change, get a review from Codex by running:
codex exec "Review the current git diff for bugs, missing checks and unused
code. List the problems only. Don't change any files."
Then fix the problems that are real, and tell me what Codex found and what
you fixed.
```

> **The @ line matters: @AGENTS.md** tells Claude Code to read that file too. Change your rules in AGENTS.md only.

6

Build something every day

Start Claude Code in your project folder and ask for a change like you normally would. Claude builds it, asks Codex for a review, fixes what's real and tells you what happened.

```
claude
```

EXAMPLE PROMPT

Add a contact form to the homepage with name, email and message fields. When you're done, follow the second-opinion step in CLAUDE.md.

> **Want a review by hand?** Run the same **codex exec** line from CLAUDE.md in your terminal any time.

How the reel is built (28 s)

Short scenes, one idea each, and a clear ask at the end.



- Orange always means Claude
- Blue always means Codex
- Every visual lands on its spoken word
- One "riser + impact" on "best combo"

04 · QUICK ANSWERS

FAQ

Do I need both subscriptions?

Yes. Claude Code needs a Claude plan, and Codex needs a ChatGPT plan that includes it or an OpenAI API key. Node.js and Git are free.

Will Codex change my code?

Not with this setup. The review line asks Codex to list problems only and not change any files. Claude makes the fixes, so one agent stays in charge of your code.

Can I swap the roles?

Yes. Let Codex build and ask Claude to review. Because both agents read the same AGENTS.md, the rules stay identical either way.

Does this slow me down?

A review adds a minute or two per change. For small edits, you can tell Claude to skip it ("no review needed for this one").

Why does it need Git?

Git keeps a record of what changed since your last save point. That record, called the "diff", is exactly what Codex reviews.

YOU'RE SET

Two models. One workflow.

Set it up once, and every change you make gets checked twice.