Coding Nexus · Follow publication

Member-only story

The Best AI Coding Setup I've Ever Used

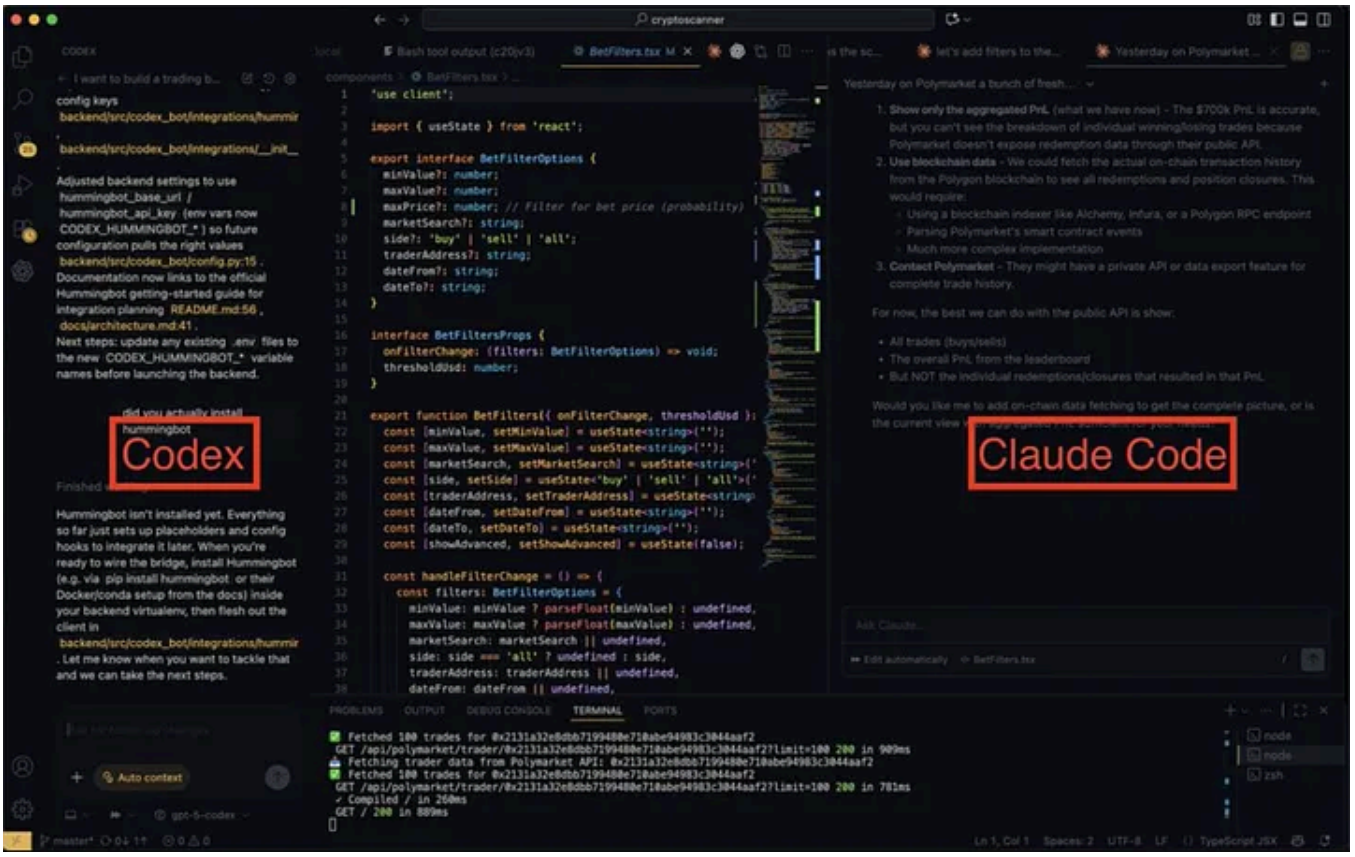
3 min read · 1 day ago

Civil Learning Following

So I accidentally built a dev team last week. Except… the "team" is just me, **Claude Code**, and **Codex**, sitting on two halves of my screen like rival engineers who barely tolerate each other.

And honestly? It's *magic*.
If you've ever wished to move 10x faster without losing your mind, this setup might be the closest thing to real AI pair programming that *feels* human.

Let me show you how it really works — no hype, just proven methods.



. . .

Step 1: Start With Claude Code

Claude Code is like that engineer who can't begin coding until they've created a complete system design doc.
Which — turns out — is a *good* thing.

When I tell Claude Code something like:

> "Build me a Flask app that uploads PDFs and summarizes them."

It doesn't go straight to code.
It *plans*. Like this:

```
# Claude's plan
1. Set up Flask project
2. Add /upload endpoint
3. Extract text using PyPDF2
4. Summarize text via OpenAI API
5. Return the summary to user
```

It's nearly annoyingly organized. But that's the point — Claude provides direction. It's the project manager who actually knows what's going on (rare species, I know).

. . .

Step 2: Pass That Plan to Codex

Now here's where it gets fun. Take that neat plan and toss it at Codex.

Codex will *not* sugarcoat anything.
It'll go:

> "Step 3: PyPDF2 is mid. Try pdfplumber — handles weird PDF encodings better."
> "Step 4: You're gonna hit API limits. Add batching or rate limiting."

Codex is your "been there, done that, not impressed" engineer. And it's *always correct*.
So I copy its feedback and relay it to Claude, like a manager passing along client notes.

Claude updates the plan.
Codex reviews again.
Rinse and repeat.

. . .

Step 3: Let Them Argue. You Watch.

At some point, you realize you're not coding — you're mediating a highly productive AI debate.

Claude writes something elegant but incomplete.
Codex reviews it like:

```
# "Bro, you didn't handle empty file uploads."
# "Also, missing API key check. Rookie move."
```

Then Claude becomes defensive (in the nicest possible way):

```
def summarize_pdf(file):
    text = extract_text(file)
    if not text:
        raise ValueError("Empty PDF, nothing to summarize.")
    api_key = os.getenv("OPENAI_API_KEY")
    if not api_key:
        raise EnvironmentError("Missing API key.")
    return call_openai_api(text, api_key)
```

And now the code actually runs.

That's the entire loop:
Claude — Codex — Claude — Codex — Done.

. . .

Step 4: Whoever Fixes the Other's Mess Becomes the Lead

Here's my unscientific rule:

> Whichever AI fixed the last bug, gets to be in charge — until they mess up.

Sometimes Claude's "philosophical" energy helps. Sometimes Codex's "no-BS" attitude saves the day. You keep bouncing between them until you stop seeing errors in your terminal.

Feels like managing two chaotic geniuses — but the good kind.

. . .

The Output: 10x Faster Builds

I've built several projects like this — small tools, APIs, even a React dashboard — and I'm not exaggerating when I say this setup **reduces build time by 70–80%**.

You're no longer thinking alone. You're managing a feedback loop that never stops and never complains (unless you count Claude's occasional "I'm sorry, I seem to have misunderstood").

. . .

Bonus Tip: Talk to Them Like Humans

This one's strange but true — if you speak to them casually, they respond better.

Example:

```
# Don't say:
"Rewrite the code."

# Say:
"Hey Claude, Codex says your error handling is weak. Can you fix that?"
```

They'll actually build on each other's context — like real teammates trying to one-up each other.
The quality difference is *wild*.

. . .

Final Thoughts

If you've been treating AI tools like autocomplete on steroids, you're missing out.
The real power is in **coordination**, not generation.

Let Claude plan.
Let Codex critique.
And you — steer the ship.

You'll improve your speed, learn more quickly, and honestly... coding becomes enjoyable once again.

This combo feels less like "using AI" and more like managing your own dev team that never sleeps.

And trust me — the results? Absolutely worth the price.

AI | Ai Tools | Ai Agent | Ai Coding | Coding

Follow

Published in Coding Nexus

8K followers · Last published 17 hours ago

Coding Nexus is a community of developers, tech enthusiasts, and aspiring coders. Whether you're exploring the depths of Python, diving into data science, mastering web development, or staying updated on the latest trends in AI, Coding Nexus has something for you.

Following

Written by Civil Learning

3K followers · 6 following

We share what you need to know. Shared only for information.

Responses (3)

Bgerby

What are your thoughts?

Larrimer Prestosa
6 hours ago

As per ChatGPT Codex no longer exist. Has this been rebranded to something like Cursor.

Reply

Gilang Wicaksono
1 day ago

Me too. My team members are opencode (grok code fast 1) and claude code. They are beast

Reply

Klyde Carpizo
1 day ago

Interesting, might as well build my own AI team for my projects.

Reply

More from Civil Learning and Coding Nexus

In Coding Nexus by Civil Learning

MarkItDown: Convert Anything into Markdown—the Smart Way to Feed LLMs

You know that feeling when you're trying to feed a PDF or a Word document into an LLM, and it just doesn't understand what's going on...

Oct 15 243 4

In Coding Nexus by Civil Learning

The Guy Who Let ChatGPT Trade for Him—and Somehow It Worked

You know how everyone says, “Don’t let AI touch your Money”? Well, someone on Reddit decided to ignore that.

Oct 8 178 8

In Coding Nexus by Algo Insights

4 Open-Source Tools That Made Me Rethink My Dev Setup

I've been coding for a while now. Most of the tools we use every day... they've been the same for years. Editors, browsers, frameworks. Just...

Sep 3 451 9

In Coding Nexus by Civil Learning

Google's New LLM Runs on Just 0.5 GB RAM—Here's How to Fine-Tune It Locally"

A few days ago, Google quietly released a little AI model called Gemma 3 270M.

Aug 15 2.1K 30

See all from Civil Learning

See all from Coding Nexus

Recommended from Medium

In Coding Nexus by Code Coup

Claude Desktop Might Be the Most Useful Free Tool You'll Install This Year

I didn't expect much when I first saw the announcement for Claude Desktop. Another AI wrapper, I thought. Maybe with a shiny UI.

6d ago 246 6

In AI Software Engineer by Joe Njenga

This Viral DeepSeek OCR Model Is Changing How LLMs Work

This DeepSeek OCR model hit an overnight success not seen in any other release—4k+ GitHub stars in less than 24 hours and more than 100k...

Oct 23 272 4

In Towards AI by Gao Dalie (高達烈)

RAG is Not Dead! No Chunking, No Vectors, Just Vectorless to Get the Higher Accuracy

Over the past two years, I have written numerous articles on how Retrieval-Augmented Generation has become a standard feature in nearly all...

Oct 17 699 6

In Dare To Be Better by Max Petrusenko

Claude Skills: The $3 Automation Secret That's Making Enterprise Teams Look Like Wizards

How a simple folder is replacing $50K consultants and saving companies literal days of work

Oct 17 398 6

In Artificial Intelligence in Plain English by Surendra Pandar

Every Paid AI—Now FREE & UNLIMITED (100% Legal)

The founder raised $5.7M to make this possible

Oct 21 47 4

NocoBase

Top 11 Open Source No-Code AI Tools with the Most GitHub Stars

We've compiled a list of 11 open-source no-code tools powered by AI to help you quickly find the platform that best suits your needs.

Oct 21 78