

# Reasons why the Claude Code should be replaced by the Codex.

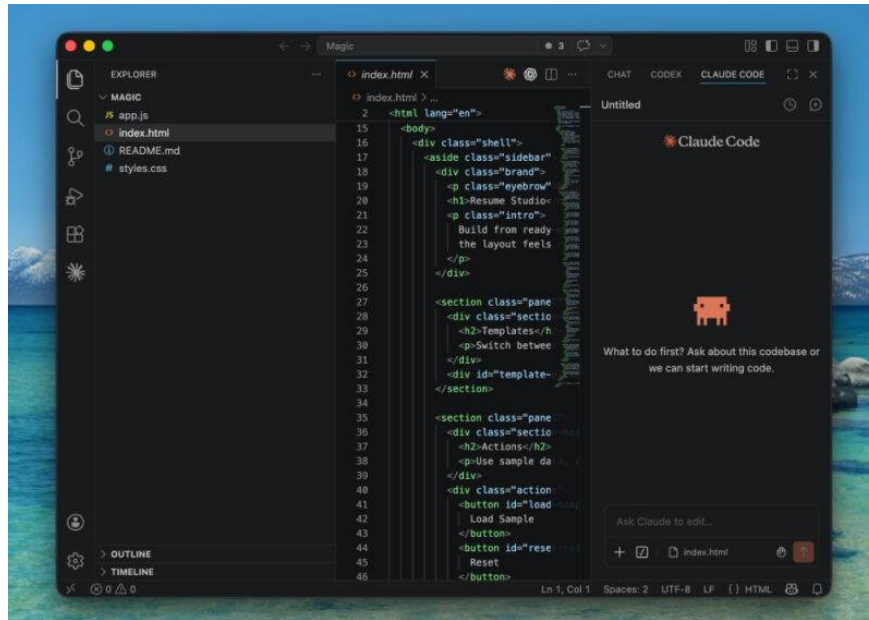
Claude Code remains a powerful tool, especially for complex tasks requiring full context. Its interactive, developer-involved approach can uncover errors in complex refactoring projects.

Many people started using Claude Code as their primary programming assistant, and it met many requirements. It runs locally, reads your files directly, and integrates with your Git setup. It also supports large context windows and even experimental features like agent-based workflows for large refactoring projects. But it also uses a lot of tokens. In one test, Claude Code consumed four times as many tokens as Codex for a similar frontend task. With a \$20/month plan, that number adds up quickly. You can reach the limit much faster than expected, especially if you work continuously. So, some people decided to abandon Claude Code entirely and switch to Codex.

## Claude Code is good, but it has problems.

Claude Code remains a powerful tool, especially for complex tasks requiring full context. Its interactive, developer-involved approach can uncover errors in complex refactoring projects. Because it runs on your computer, it can utilize any local tools or custom hooks you've set up, and by default, it never sends your code to the cloud. You can even write a CLAUDE.md file with project-specific instructions, and Claude Code will read it every time.

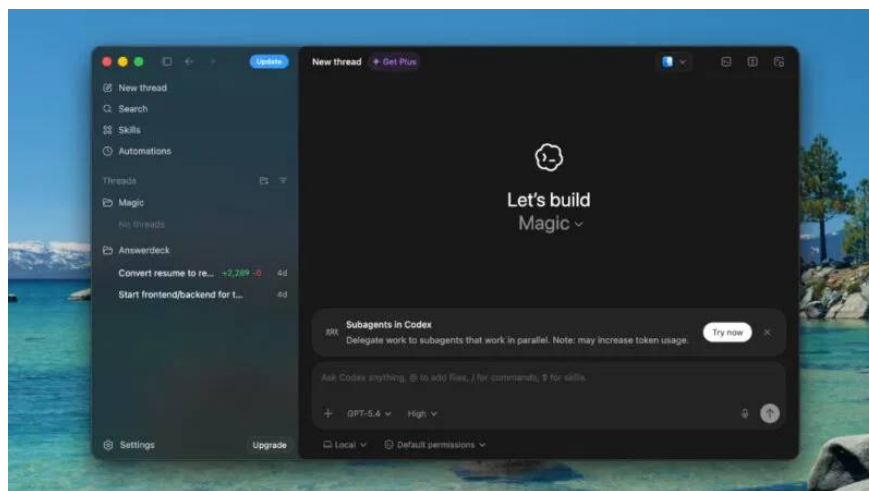
But these features come with trade-offs. One problem is token usage. Claude Code's output is highly detailed, which means it consumes a lot of tokens. For example, in a Figma styling task, Claude consumed 6.2 million tokens compared to Codex's 1.5 million for the same result.



Another challenge is the interactive workflow. Claude Code shows you each planned change and waits for your approval before proceeding. This gives you control but means you can't let it execute a task on its own without proofreading. For quick bug fixes or writing simple functions, this can feel cumbersome. In fact, you have to say "No, continue" a lot, which disrupts the workflow. Finally, Claude Code's Pro plan has a fixed usage limit. With heavy use, the \$20 plan often runs out quickly, forcing users to upgrade to the more expensive Max plan.

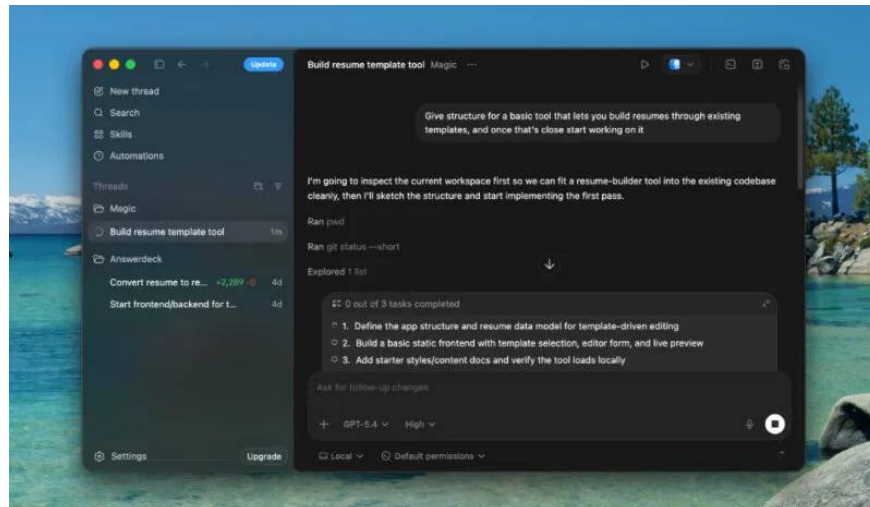
## The Codex turned out to be better than we thought.

The latest version of the Codex has addressed many of Claude Code's weaknesses. Firstly, it has proven highly capable of automating programming tasks. You describe your goal in English, and the Codex will then plan and execute it automatically. In tests, the Codex handled tasks such as generating sample code, refactoring functions, and even complete features well.

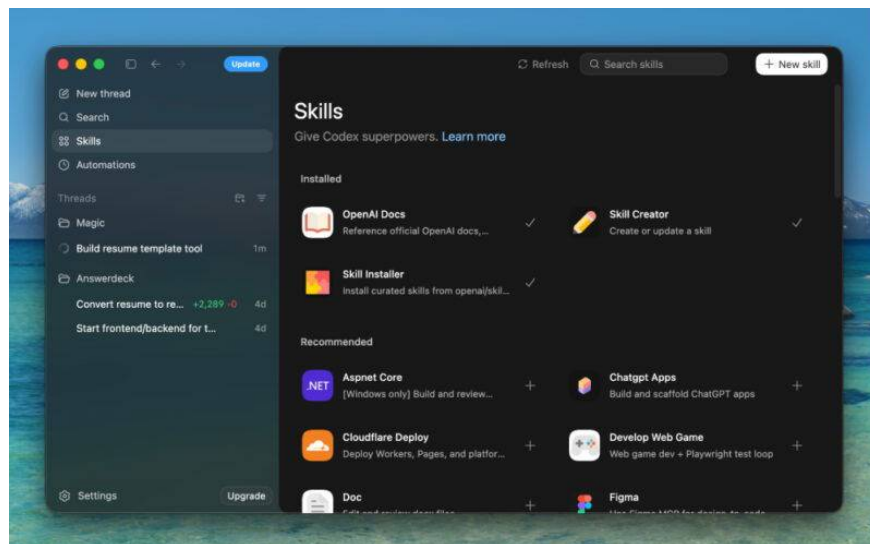


It also has a larger context window – it pulls your entire repository at hand when working, and uses a difference-based context strategy so that long work sessions can continue without losing track. The Codex output is

generally excellent. It typically produces concise, well-functioning code instead of lengthy comments.



Claude often tried to match the original structure with lots of comments, while the Codex 'just settled' with minimal text. For example, when asked to write unit tests or bug fixes, the Codex provided quick patches. It could even automatically generate pull requests via GitHub integration. This completely changed how code review and CI/CD were viewed – it was possible to tag @Codex and receive automated reviews or bug fixes without having to write any pipelines yourself.



You can also use the Codex CLI, an open-source and easy-to-install tool. Simply run:

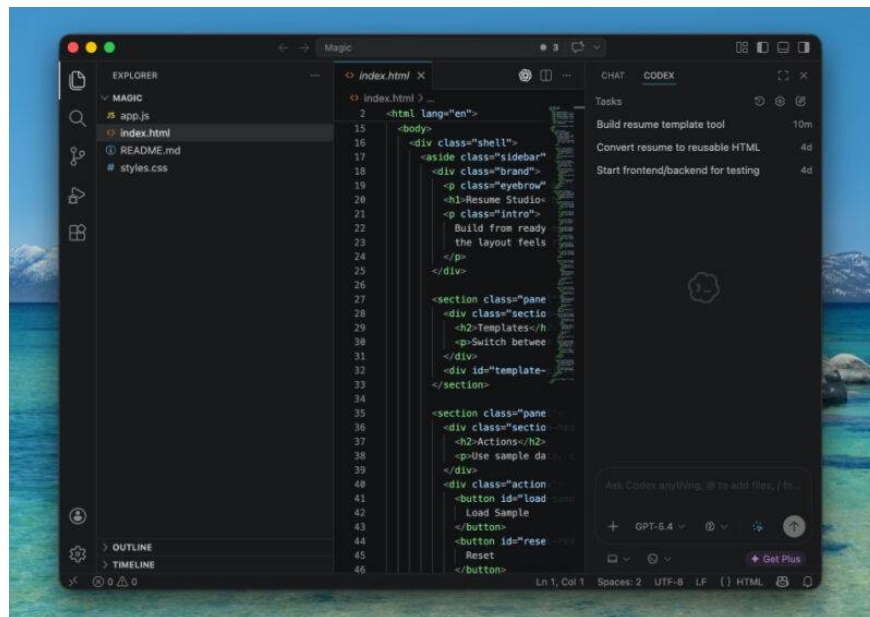
```
npm install -g @openai/codex codex "refactor this module to use async/await"
```

The CLI has modes like 'suggested' and 'fully automatic', so you can choose the level of autonomy you want. Additionally, a nice touch is that the Codex reads the AGENTS.md file if you have one, which is an open standard, so any existing project instructions will be carried over. Finally, while Claude Code has limited official tools, the Codex now has an official VS Code extension and a macOS application (Windows support coming soon). This means you can use the Codex in the cloud or on your computer flexibly, giving you the flexibility that Claude Code doesn't offer.

# Using the Codex inside VS Code

One of the best features of the Codex is its integration with VS Code. The official extension brings the AI chat panel directly into your editor.

Install the extension from the VS Code Marketplace by adding the OpenAI Codex extension. It appears as a Codex icon in the sidebar. Clicking the icon will open the chat panel, prompting you to log in with your ChatGPT account (Plus or Pro) or API key. After logging in, Codex defaults to starting in Agent mode, meaning it can read files in your open project, run commands, and even write code after requesting permission.



From there, you can ask programming questions in plain English. For example, you can highlight a function and ask it to explain its functionality, or type something like 'write tests for all endpoints'. This extension is context-aware, so it reads the open files and highlighted code to provide a suitable answer.

When the Codex suggests edits, the extension will display the differences. It also integrates with Git, making it easy to manage changes. You can also adjust the approval mode. The extension lets you choose between Chat-only mode, which doesn't make code changes; Agent mode, which asks for permission before making changes; and Full Access mode, which makes changes without prompting. Overall, Agent mode strikes a good balance between convenience and security.

If you're still unsure which AI tool to pay for, this comparison of ChatGPT Plus and Claude Pro will help. By the way, check out these 14 ChatGPT alternatives .

You finished reading the article "**Reasons why the Claude Code should be replaced by the Codex.**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.