

Is the Codex or Claude Code better for AI in programming?

Many users were once proficient with Cursor Agents. Then Claude Code came along and gradually became a commonly used tool. And now the Codex has appeared. So which AI tool is better for programming?

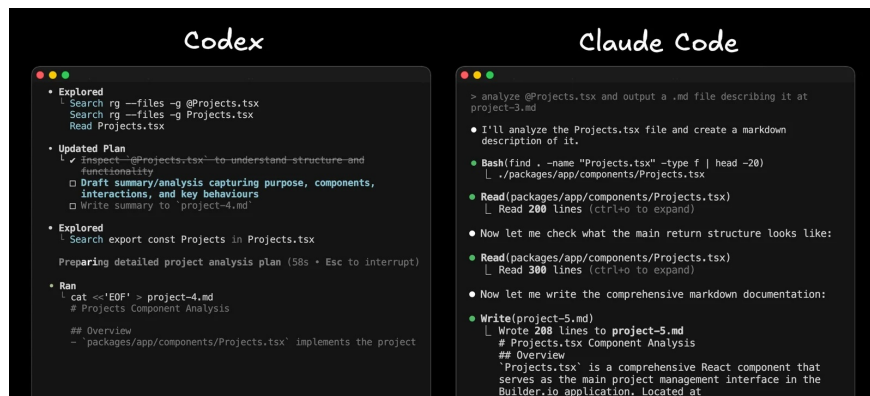
Many users were once proficient with Cursor Agents. Then Claude Code came along and gradually became a commonly used tool. And now the Codex has appeared. So which AI tool is better for programming?

Agents are becoming increasingly similar.

Cursor's latest agent is quite similar to Claude Code's latest agents, and Claude Code, in turn, is quite similar to Codex agents.

Historically, Cursor laid the foundation for many things. Claude Code improved upon them. Cursor copied useful things like to-do lists and better diff formatting, and the Codex adopted many of those as well.

The Codex is so similar to Claude Code that many people wonder if it was trained based on the output of Claude Code.



Here are a few subtle behavioral differences that we might notice:

1. The Codex tends to have slightly slower reasoning, but the token output rate per second (number of tokens displayed per second) is faster.
2. Claude Code tends to be less analytical, but its token generation speed is slightly slower.

3. Within Cursor, switching between models also changes the feel similarly: GPT-5 spends more time on inference, Sonnet spends less time on inference and more time on code output, although the output is slightly slower, especially if you use Opus.

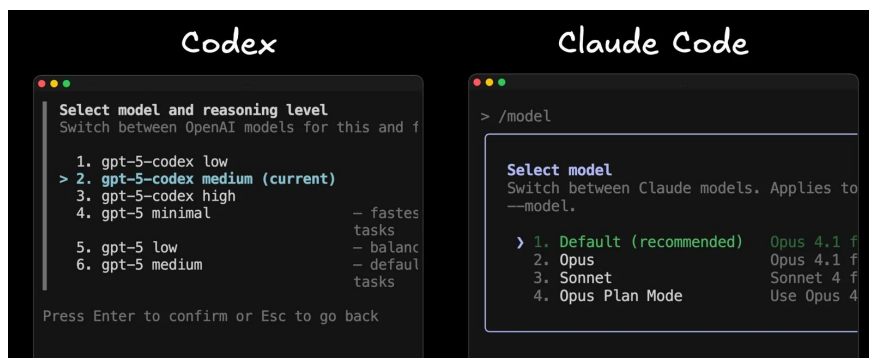
Ultimately, these agents are comparable. Whether you prefer Cursor, Claude Code, or Codex is entirely a personal choice. Many people have a slight preference for Codex or Claude Code, mainly because the company that builds the tool also trains the models and seems to optimize the entire process.

Winning option: Draw

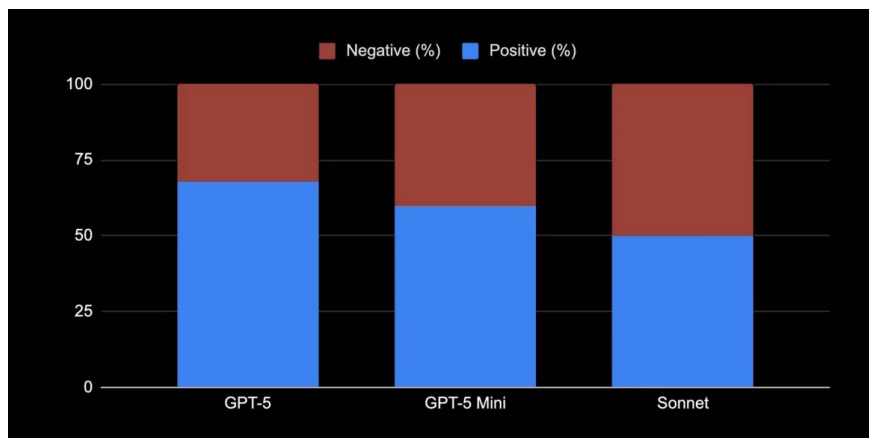
Inference models and controls

Many people quite like the GPT-5 Codex model. It has improved in knowing how much reasoning time is needed for different types of tasks. Over-reasoning for basic tasks is frustrating.

The Codex also allows you to select low, medium, high, or even minimal inference levels for super-fast execution. These options are better than having only two model choices in Claude Code. Cursor, on the other hand, has many options, which sounds good in theory, but can be a bit confusing in practice.



If the company that manufactures the tool trains the models, they will know how to use it best and can offer a reasonable price.



Everyone will have different preferences, so it's a tie here again.

Winning option: Draw

Actual prices and limits

The Codex has standard ChatGPT packages . The Claude Code has standard Claude packages.

On the surface, the prices look pretty similar: Free, packages around \$20, and higher-end packages ranging from \$100 to \$200.

The key point is that the GPT-5 is significantly more efficient than the Claude Sonnet , and especially the Opus. In recent practical use, its quality has been assessed as comparable by most anecdotal and public standards, but the GPT-5 costs only about half as much as the Sonnet, and nearly one-tenth as much as the Opus, meaning the Codex can offer many things at a lower cost.

Providers don't always specify the exact number of tokens required for each package, but in experience, the Codex seems more generous.

Many people can comfortably use Codex's \$20 plan more than Claude's \$17 plan, where limits are reached quickly. Even with Claude's \$100 and \$200 plans, heavy users still hit their limits. With Codex Pro, you almost never hear users complain about exceeding their usage limits.

ChatGPT plans		
Free Explore how AI can help with everyday tasks \$0 / month	Plus Level up productivity and creativity with expanded access \$20 / month	Pro Get the best of OpenAI with the highest level of access \$200 / month

Claude plans		
Free Try Claude \$0	Pro For everyday productivity \$17	Max Get the most out of Claude From \$100

It's also worth noting that this isn't just a 'programming package'. You also get ChatGPT or Claude Chat. With ChatGPT, you also get one of the best image and video creation models, plus more refined products like the ChatGPT desktop application that people use every day.

Claude's desktop application feels slow and more like a basic Electron wrapper. Claude has better MCP integration with many one-click connectors.

Since the number one complaint we hear about programming software is that it's limited in usage, the Codex actually has an advantage.

Winning option: Codex

User experience and permissions

The Codex identifies repositories tracked by Git and defaults to high privileges.

Claude Code's permission system drives many people crazy, and they frequently launch it with the `--dangerously-skip-permissions` option. This is an unnecessary risk, but it causes significant workflow disruption, and the settings aren't saved.

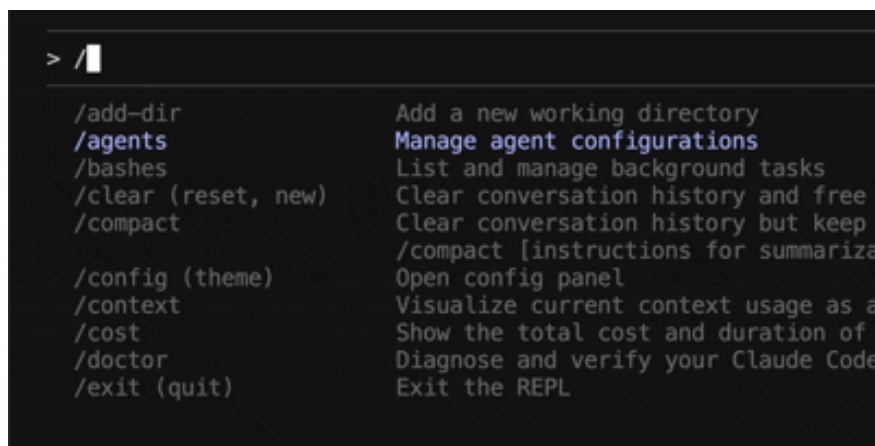
Both terminal user interfaces are fine. Claude Code's terminal UI is slightly nicer and noticeably more mature, ultimately giving you better control over permissions.

Therefore, although both are fairly evenly matched, the Codex generally feels more fundamental, so the victory leans toward the Claude Code.

Winning option: Claude Code

Features

Claude Code has more features: auxiliary agents, custom hooks, and many configuration options. Cursor also has quite a few features, but Codex is the most limited.



```
> /  
  
/add-dir          Add a new working directory  
/agents          Manage agent configurations  
/bashes          List and manage background tasks  
/clear (reset, new) Clear conversation history and free  
/compact         Clear conversation history but keep  
/compact [instructions for summariza  
/config (theme)  Open config panel  
/context         Visualize current context usage as a  
/cost            Show the total cost and duration of  
/doctor         Diagnose and verify your Claude Code  
/exit (quit)    Exit the REPL
```

However, the strength of the Codex is that it's open source, so you can customize it in any way you want or learn from it to develop your own agent.

But in short, if you want a lot of features (including some really useful ones), Claude wins.

Winning option: Claude Code

Instruction file: Agents.md vs. Claude.md

One thing that frustrates people about Claude Code is that it doesn't support the standard Agents.md, only Claude.md.

Tools like Cursor, Codex, and Builder.io all support Agents.md. It's inconvenient to maintain a separate file for Claude when everything else adheres to this standard.

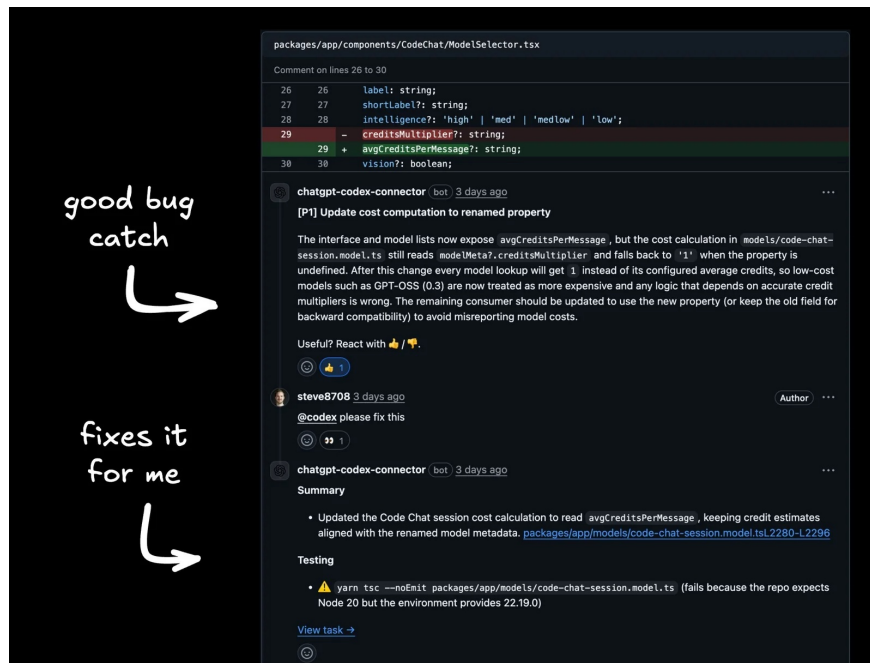
Winning option: Codex

The big difference: GitHub integration

This is the main reason why people prefer the Codex.

Some people briefly tried integrating Claude Code's GitHub repository. The Builder.io development team found it terrible. The reviews were very lengthy without spotting obvious bugs. You couldn't comment and ask for helpful bug fixes. It offered no value.

The Codex's GitHub application is the opposite. Install it, enable automatic code review for each repository, and it actually finds hard-to-detect errors. It comments directly, you can request it to fix the bug, the tool works in the background, then lets you review and update the pull request right there, and then merges.



Importantly, the user experience feels similar to that on the terminal. Prompts that work on the CLI also work on the GitHub UI. The same model, the same configuration, the same behavior. That consistency is crucial.

The second contender is Cursor's Bugbot. It finds bugs well and offers helpful "fix on the web" or "fix in Cursor" options. You won't regret choosing either one. Many still prefer the Codex because of its price, model integration capabilities, and consistency with the CLI workflow.

Winning option: Codex

Currently, the best option is the Codex. Its integration with GitHub is excellent, the pricing and limitations are reasonable, the modeling options suit how people work, and consistency from start to finish is crucial.

But honestly, you really can't go wrong with any of these options right now. If you prefer Claude Code or Cursor, that choice should absolutely be respected.

You finished reading the article "**Is the Codex or Claude Code better for AI in programming?**" 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.
