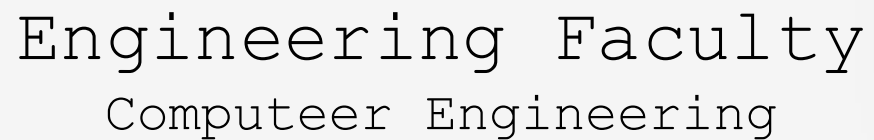


[illegible]

Prepared by: Dr Ercan Ezin

Google I/O

Google I/O registration is now open!

Tune in live May 20th and 21st

[Register now](#)

Hi Ercan,

Get ready to tune into Google I/O, where you can explore the latest innovations from Google! Join us online May 20-21 for live streamed keynotes and sessions covering what's new in AI, Android, web, cloud, and more.

[Register now](#) at no cost and receive your exclusive I/O 2025 badge.



<https://io.google/2025/>

20 Mayıs Salı-20:00-22:00

21 Mayıs Çarşamba-20:00-22:00

INTRODUCTION

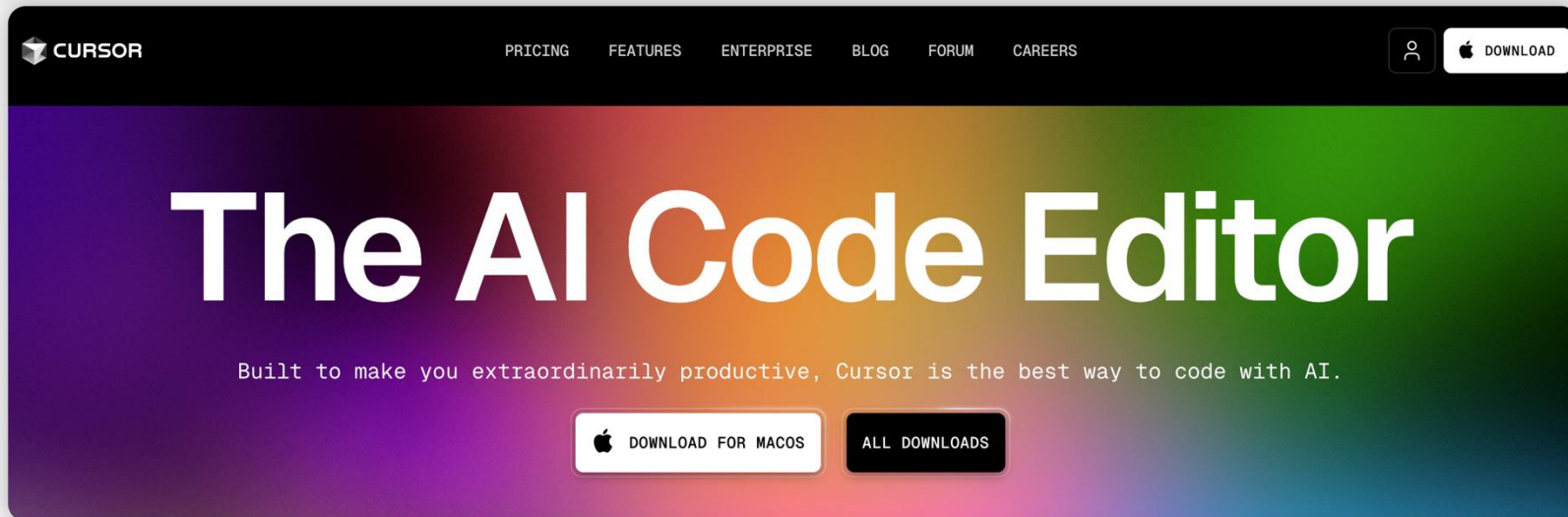
THIS WEEK WE WILL WORK ON

DEV WITH AI POWERED IDEs

ARGUING with ALGORITHMS

How I write code using Cursor: A review

Written October 25, 2024






EVALUATING CURSOR'S VALUE

- In forums relating to AI and AI coding in particular, I see a common inquiry from experienced software developers: Is anyone getting value out of tools like Cursor, and is it worth the subscription price?



MY EARLY OBSERVATIONS

- A few months into using Cursor as my daily driver for both personal and work projects, I have some observations to share about whether this is a "need-to-have" tool or just a passing fad, as well as strategies to get the most benefit quickly which may help you if you'd like to trial it.



WHO I AM AND WHO YOU ARE

- I have been writing code for 36 years in a number of languages, but professionally focused on C-heavy computer game engines and Go/Python/JS web development. I am expecting readers to be similarly reasonably comfortable and productive working in large codebases, writing and debugging code in their chosen language, etc. I would give very different advice to novices who might want an AI to teach them programming concepts or write code for them that is way beyond their level!

WHAT IS CURSOR?

- Cursor is a fork of Visual Studio Code (VS Code) which has Large Language Model (LLM) powered features integrated into the core UI. It is a proprietary product with a free tier and a subscription option; however, the pricing sheet doesn't cover what the actual subscriber benefits are and how they compare to competing products. I'll try to clarify that when discussing the features below based on my own understanding, but a quick summary:



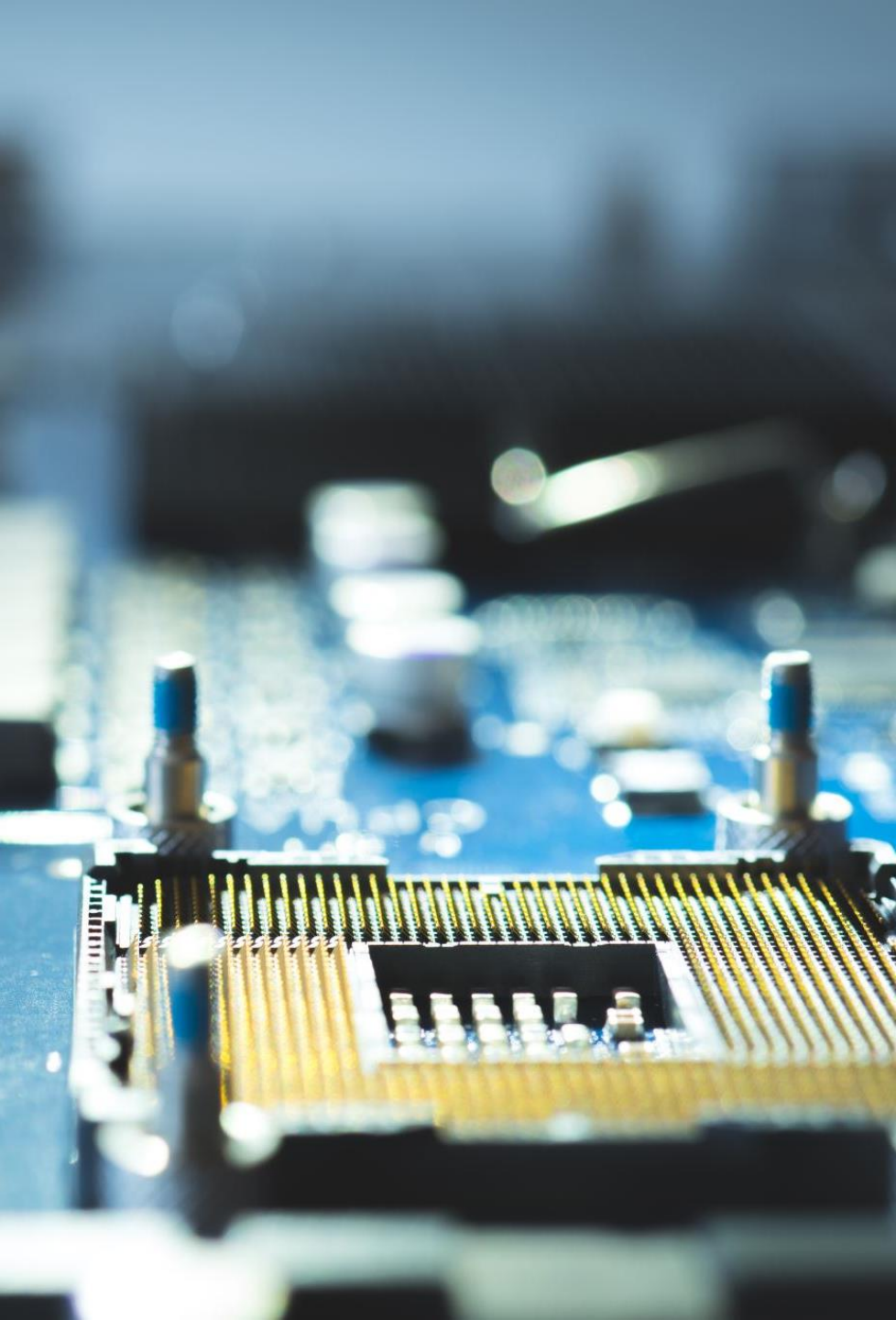
4 FEATURES OF CURSOR IDE

Tab completion: This is a set of proprietary fine-tuned models that both provide code completion in the editor, as well as navigate to the next recommended action, all triggered by the Tab key. Only available to subscribers.

Inline editing: This is a chat-based interface for making edits to selected code with a simple diff view using a foundation model such as GPT or Claude. Available to free and paid users.

Chat sidebar: This is also a chat-based interface for making larger edits in a sidebar view, allowing more room for longer discussion, code sample suggestions across multiple files, etc. using a foundation model such as GPT or Claude. Available to free and paid users.

Composer: This is yet another chat-based interface specifically meant for larger cross-codebase refactors, generating diffs for multiple files that you can page through and approve, also using a foundation model such as GPT or Claude. Available to free and paid users.

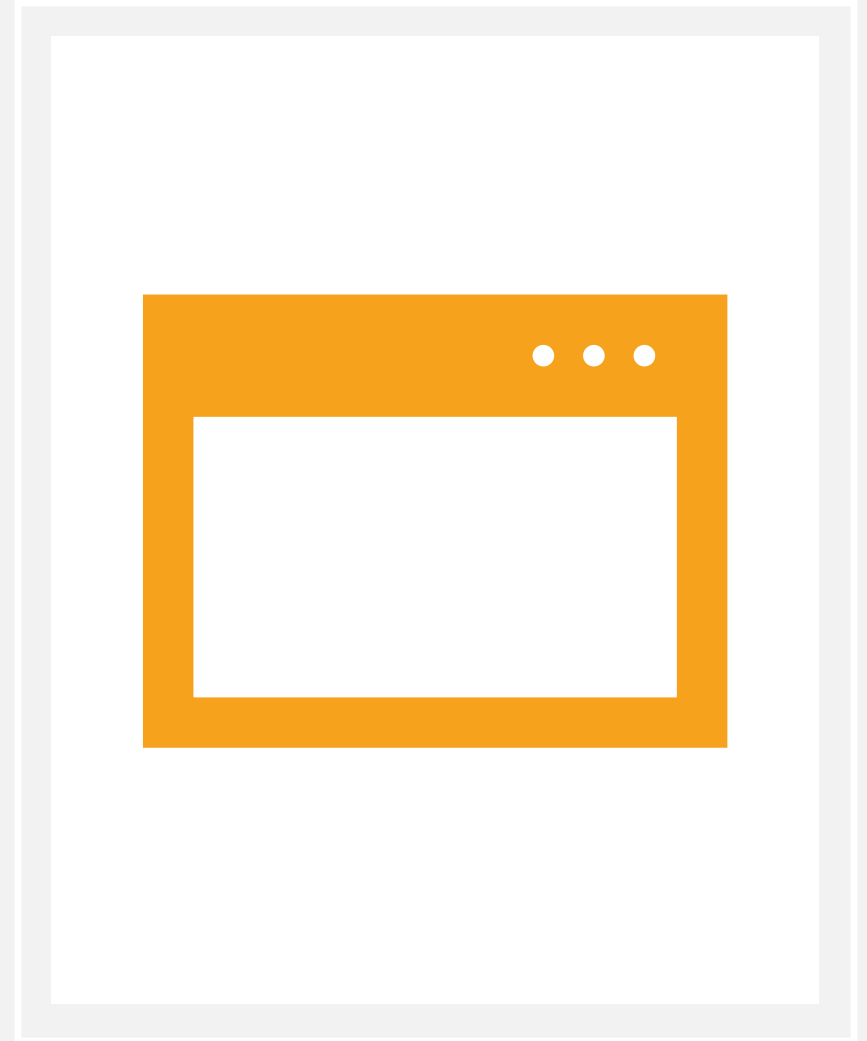


TAB COMPLETION

- While other LLM-powered coding tools focus on a chat experience, so far in my usage of Cursor it's the tab completion that fits most naturally into my day-to-day practice of coding and saves the most time. A lot of thought and technical research has apparently gone into this feature, so that it can not only suggest completions for a line, several lines, or a whole function, but it can also suggest the next line to go to for the next edit.

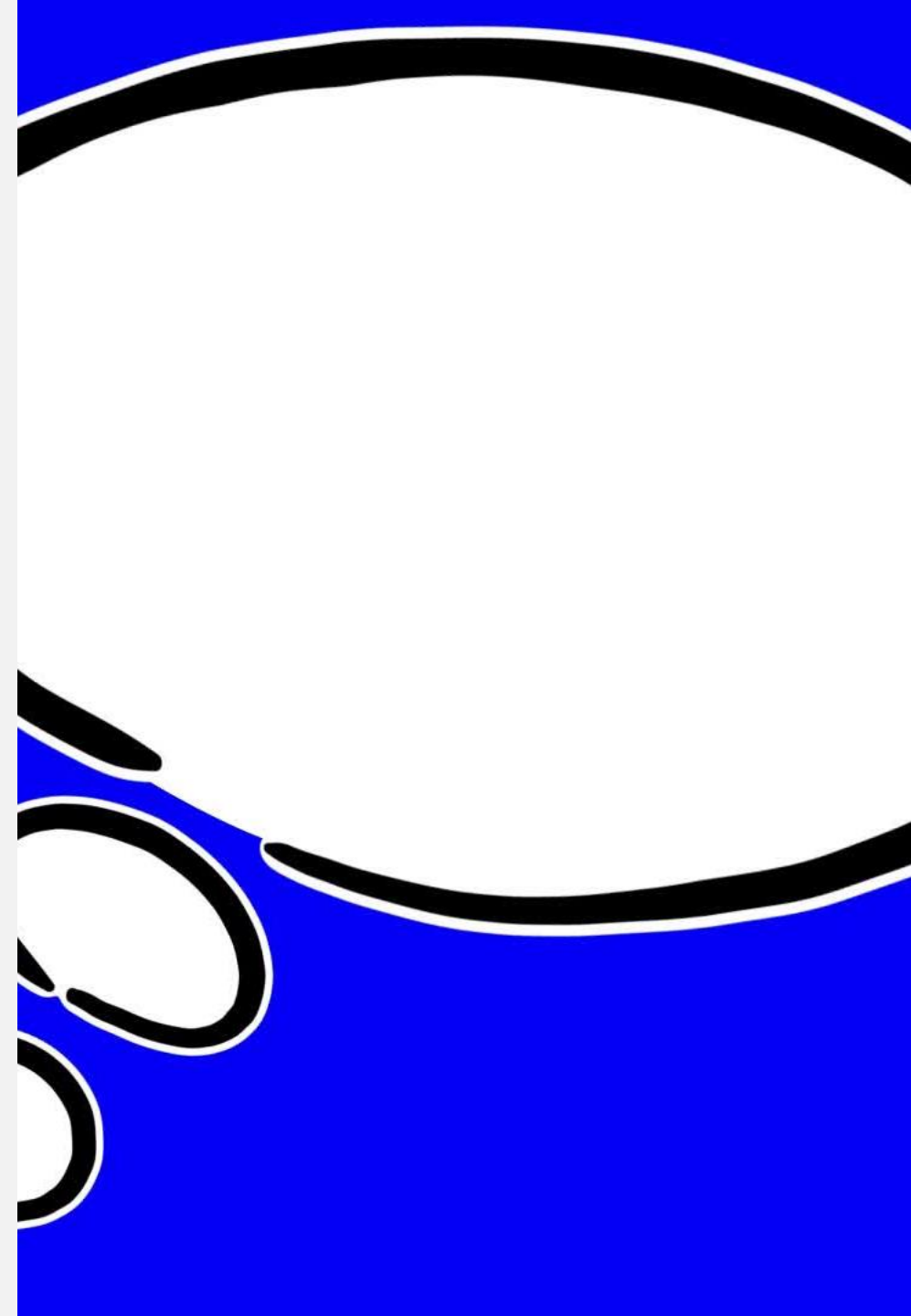
BENEFITS OF TAB COMPLETION

- Sometimes tab completion will independently find a bug and propose a fix. Many times it will suggest imports when I add a dependency in Python or Go.



DRAWBACKS OF TAB COMPLETION

- One is a minor annoyance: Sometimes I don't see the suggestion in time and continue typing, and the completion disappears. My other complaint is the exact opposite situation: Sometimes a completion is dead wrong, and I intentionally dismiss it.



SUMMARY THOUGHTS

- Whether I'll be using Cursor in a few years or have moved on to another tool, I can't really tell. I am confident that at the time of writing this, Cursor is the best example of the potential of LLM coding assistants, and if you want to explore how this type of tool might be of value I suggest you give it a spin.

The tool for the next generation of developers

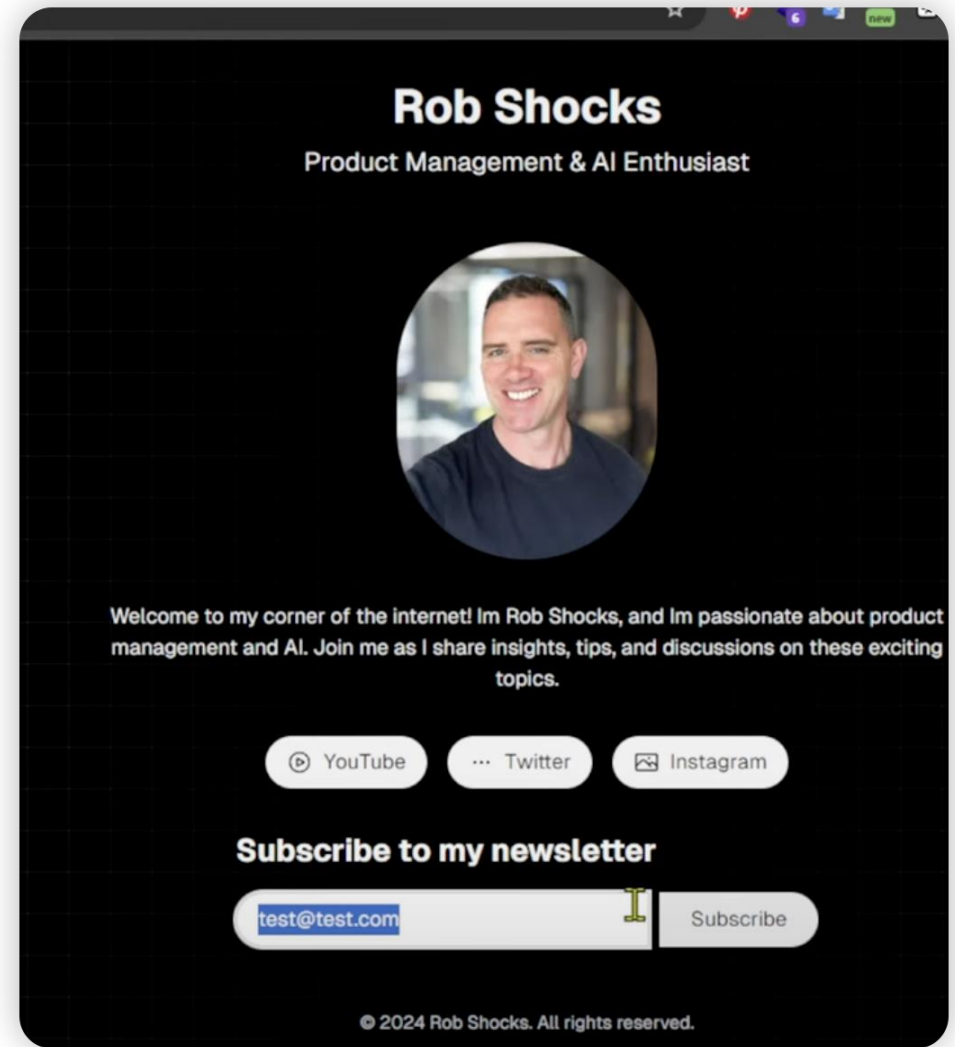
Verified university students get one year of free Cursor Pro.

VERIFY STATUS

CURSOR FOR STUDENTS

<https://www.cursor.com/students>

CURSOR IN ACTION



https://www.youtube.com/watch?v=Tm_2RZm8JB8

PRESENTATION TIME!

Overall 20%

5% Introduction of self and the topic

10% Presentation content(English is favoured)

5% Presentation skills and using English

You have 5 Minutes, make it count!



WORDS OF THE WEEK

- 1- **Abstract**: To consider something independently of its concrete existence; in CS, to extract essential features while ignoring details.
- 2- **Augment**: To make something greater by adding to it; often used when enhancing algorithms or systems.
- 3- **Coherent**: Logically connected and consistent; describes well-structured code or documentation.
- 4- **Concurrency**: The execution of multiple sequences of operations simultaneously or overlapping in time.
- 5- **Convolut ed**: Extremely complex and difficult to follow; warns against over-engineering.
- 6- **Cognizant**: Being aware or informed of something; e.g., “be cognizant of security implications.”
- 7- **Discrepancy**: A lack of compatibility or similarity between two or more facts; useful when debugging or comparing outputs.
- 8- **Empirical**: Based on observation or experiment rather than theory; e.g., “empirical performance data.”
- 9- **Encapsulation**: The bundling of data with the methods that operate on that data; fundamental to object-oriented design.
- 10- **Heuristic**: A practical method not guaranteed optimal but sufficient for immediate goals; widely used in search optimization.
- 11- **Idempotent**: Describes an operation that produces the same result even if applied multiple times; important in APIs.
- 12- **Instantiate**: To create a concrete instance of an abstract data type or class.
- 13- **Intrinsic**: Belonging naturally; e.g., “intrinsic performance of the hardware.”
- 14- **Optimize**: To make as effective or functional as possible, often by improving code or algorithms.
- 15- **Parsimonious**: Characterized by extreme frugality; in CS, preferring the simplest solution with minimal resources.
- 16- **Redundancy**: Duplication of critical components or functions to increase reliability.
- 17- **Robust**: Able to handle error conditions or unexpected inputs without failing.
- 18- **Scalability**: The capability of a system to handle growing amounts of work or to be enlarged.
- 19- **Synergy**: The interaction of elements that when combined produce a total effect greater than the sum of the individual parts.
- 20- **Validate**: To check or prove the accuracy or correctness of something, such as input data or results.



EOF