

Scan to ask questions



Contributing to Open Source with Cursor

December 25, 2025

Agenda

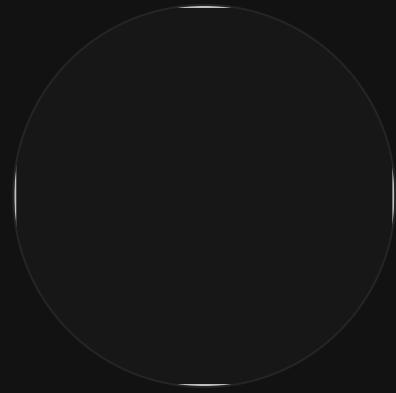
 Workshop

 Q/A

 Hackathon

 Networking

 End



Usman Siddique

Software Engineer at **Motive**

An Analogy: Getting Across Town

Imagine you need to get across town. There are many ways to do it...



Walking



Free



Takes longest



Most effort



Biking



Some cost



Faster



Moderate effort



Driving



Most expensive



Fastest



Least effort



You have the choice between **time**, **money**, **reliability**, and **effort**

Similarly: Building Software



No Tools

- 💰 Free
- ⌚ Slowest
- ⚠️ More errors



IDE & Tooling

- 💰 Some cost
- ⌚ Faster
- ✅ Feedback



AI Tooling

- 💰 Costs more
- ⌚ Fastest
- 🎯 Practice needed



AI tools help you build **faster** with **practice**

Cursor

AI Models = Super Intelligent APIs



Stripe API

Handle payments

```
stripe.charges.create()
```



Twilio API

Send messages

```
twilio.messages.create()
```



AI Model API

Solve any task

```
openai.chat.completions()
```



The key difference: **No guaranteed identical results**

AI Hallucinations 🌟

When AI **confidently generates incorrect information** that seems plausible

🧐 Why does this happen?

- AI predicts next token based on patterns
- Like powerful autocomplete
- It doesn't say "I don't know" - generates what **seems** most likely

💻 In coding, this means...

- ✗ Inventing APIs that don't exist
- ✗ Wrong function signatures
- ✗ Outdated syntax or methods
- ✗ Plausible but buggy logic

✓ **The fix:** Develop a **verification mindset** – every suggestion is a starting point, not a final answer

Understanding Tokens

What is a Token?

A token can be a word or parts of a word:

`hello` → 1 token

`un` `believ` `able` → 3 tokens

 **Why it matters:** Tokens = pricing + speed

Input vs Output Tokens



Input Tokens



Lower cost

Your prompt + context



Output Tokens



2-4x more!

Model's response



Why streaming? AI generates tokens **one at a time**, predicting the next based on previous ones.

Context Window



Your Prompt

What you ask



Chat History

Previous messages



Code Context

Files & snippets

⚠ When context fills up: AI starts "forgetting" earlier parts → answers get worse



Mental Model: Watch the context indicator – **when it's filling up, start a new chat**

Mental Model Summary



You are in control



LLMs amplify your productivity

Non-deterministic



Same prompt → different responses

LLMs hallucinate



You must verify everything

Watch context window



New chat at ~90% filled

Contributing to Open Source with Cursor



Understanding any codebase in minutes

What is Open Source?

Software released under a license where the **copyright holder grants users the right** to **use, study, change, and distribute** the software and its source code to anyone for any purpose.

Major Projects



Linux



VS Code



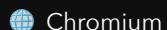
React



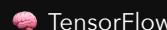
Python



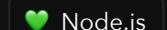
Kubernetes



Chromium



TensorFlow



Node.js



Firefox



Blender



Git



PostgreSQL



My favourites



Dub.co



Roadmap.sh



freeCodeCamp



Build Your Own X

Why Contribute to Open Source? 🤔

If you're doing it just to get a job... DON'T.



Open source requires **patience** and **genuine interest**.

✓ **Do it if you want to...**



Improve your skills



Build a better mindset



Build relationships



Have real impact



Make work public

Where to Find Open Source Projects



Google Summer of Code

Paid summer program for students to contribute to open source



LFX Mentorship

Linux Foundation program with stipends for contributors



MLH Fellowship

12-week internship alternative with real projects



Microsoft FOSS Fund

Microsoft's open source contribution programs



Outreachy

Paid internships for underrepresented groups in tech



GitHub Explore

Find trending repos and good first issues directly



Start small, build your portfolio, and grow your network!

Steps to Your First Contribution

2. Good First Issues

- Look for `good first issue` labels
- Start small, build confidence
- Ask Cursor to explain the issue context

3. CONTRIBUTING.md

- Read the contributing guidelines
- Understand code style & conventions
- Check testing requirements

4. Writing Good Pull Requests



Add "Why" Comments

Don't just explain what the code does – explain **why** it does it.

```
// We use debounce here to  
  
// prevent API spam on  
  
// rapid keystrokes
```



Make PRs Small

- ✓ Reviewed more quickly
- ✓ Reviewed more thoroughly
- ✓ Less blocking on reviews
- ✓ Simpler to roll back



Clear Description

- **What** changed?
- **Why** this approach?
- **How** to test it?
- Link to the **issue**



A good PR tells a **story** – reviewers should understand your thinking without asking questions

Common Mistakes to Avoid

Blindly Trusting AI Output

Always review and understand the code AI generates.
You're responsible for your commits!

Large First PRs

Start small! A 10-line fix is better than a 500-line refactor
for your first contribution.

Ignoring Project Guidelines

Each project has its own conventions. Read
CONTRIBUTING.md and follow the existing patterns.

Not Understanding Your Change

If you can't explain what your code does, don't submit it.
Take ownership of every line.

Bonus: Cursor Cloud Agents + Slack



Work on tasks directly from Slack!

Example message:

```
@cursor fix the login bug in auth.ts and open a PR
```

With Cursor's Slack integration, Cloud Agents can work on your tasks by simply mentioning @cursor with a prompt.

Thank You! 🙏

Let's Connect 