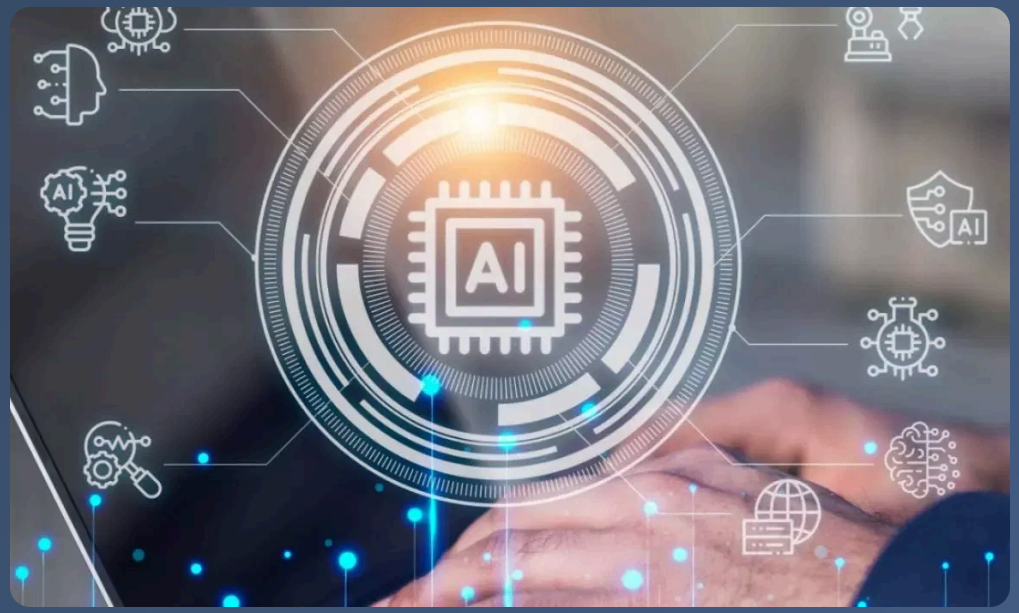
Introduction to AI Coding Tools

For Economics & Finance Students



Why AI Coding Tools Matter for Econ/Finance Students

Data Analysis Acceleration: AI coding tools help economics and finance students analyze large datasets more efficiently, automating repetitive coding tasks and suggesting optimized data processing methods.

Lower Technical Barrier: Students can focus on economic concepts and financial models rather than struggling with syntax and programming details, making technical skills more accessible.

Advanced Modeling Capabilities: AI assistants can suggest statistical methods, econometric approaches, and visualization techniques specifically relevant to economics and finance problems.

Industry-Relevant Skills: Experience with AI coding tools provides a competitive advantage in the job market, as financial institutions increasingly adopt these technologies for research and analysis.

Types of AI Coding Tools

IDE-Based Tools

Integrated directly into code editors, these tools provide real-time suggestions, code completion, and editing capabilities while you write code.

Examples: Cursor, GitHub Copilot, Tabnine

Terminal-Based Tools

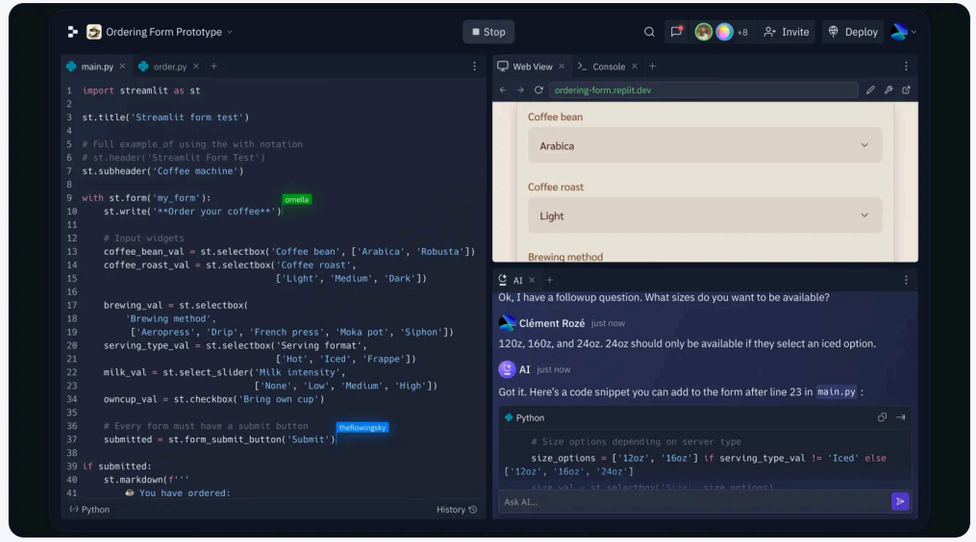
Command-line interfaces that integrate AI assistance into your terminal workflow, helping with code generation, debugging, and documentation.

Examples: Claude Code, Gemini CLI, Qodo

Autonomous Agent Tools

AI systems that can independently plan and execute entire coding tasks with minimal human intervention, functioning as virtual software engineers.

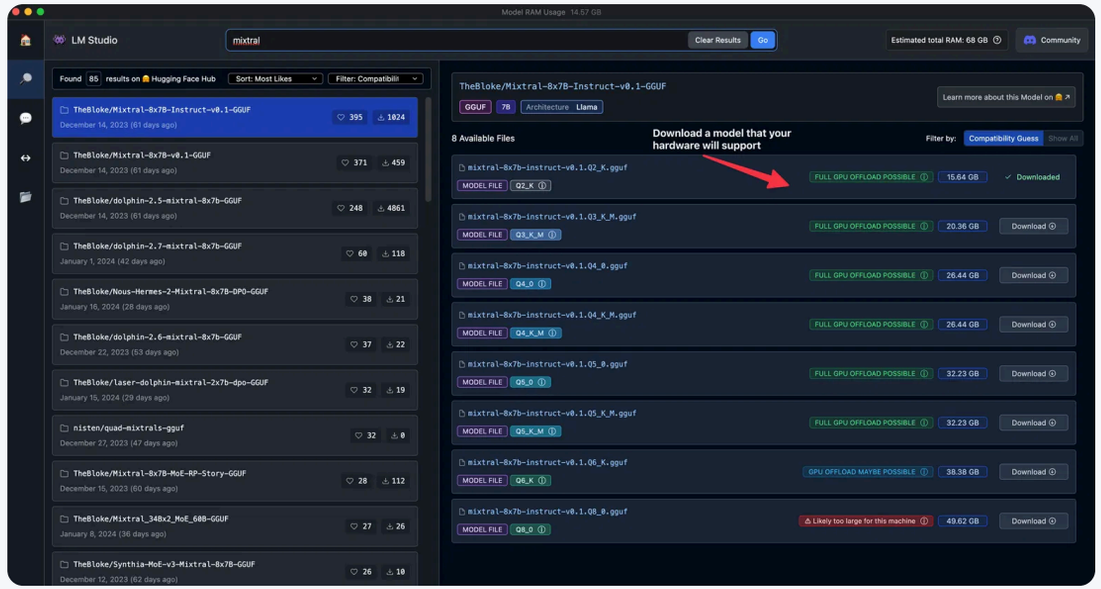
Examples: Devin, AutoGPT, BabyAGI



IDE-Type AI Coding Tools: Cursor

Cursor is an AI-enhanced code editor built to make developers extraordinarily productive. It integrates AI capabilities directly into a familiar IDE environment, making it ideal for economics and finance students working with data analysis code.

- **Intelligent Code Completion:** Predicts your next edit and suggests multi-line code blocks based on context, helping students write statistical analysis and financial modeling code faster.
- **Codebase Understanding:** Analyzes your entire project to provide context-aware suggestions, perfect for navigating complex econometric models or financial simulations.
- **Natural Language Editing:** Update entire functions or classes using simple instructions, allowing students to focus on economic concepts rather than syntax details.
- **Familiar Environment:** Built on VS Code, allowing easy import of extensions, themes, and keybindings that economics and finance students may already be using for Python, R, or other data analysis tools.



Terminal-Type AI Coding Tools: Claude Code

Claude Code is a command-line tool for agentic coding developed by Anthropic. It integrates AI assistance directly into your terminal workflow, providing a flexible, customizable, and scriptable coding environment.

Native Terminal Integration

Works within your existing terminal environment, accessing your shell tools and bash environment without disrupting your workflow.

Context-Aware Assistance

Automatically pulls relevant context from your codebase to provide more accurate suggestions and solutions for economics and finance applications.

Customizable Workflows

Create custom slash commands for repeated tasks like data analysis pipelines or econometric model implementations.

Privacy-Focused Design

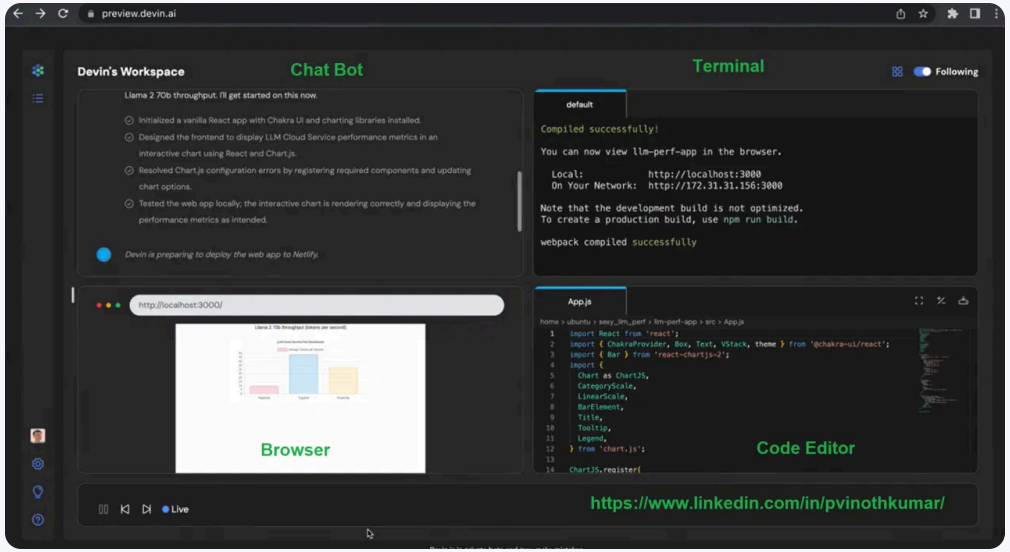
Provides granular control over permissions and data sharing, important when working with sensitive financial data.

Autonomous AI Coding Tools: Devin

What is Devin? An autonomous AI software engineer that can plan and execute entire coding projects with minimal human supervision.

Key Capabilities: Devin can write code, run tests, debug issues, and deploy applications end-to-end while maintaining context across multiple sessions.

Workflow: Users provide a task or ticket, Devin creates a plan, executes the code, tests its work, and delivers a complete solution or pull request.



Real-World Application

Nubank used Devin to refactor millions of lines of code in their ETL system, achieving 8-12x efficiency gains and 20x cost savings compared to manual engineering work.

Applications for Economics Students

Econometric Modeling

AI coding tools can generate complex econometric models in R, Python, or Stata, suggesting appropriate statistical methods and helping debug estimation issues in time series, panel data, and structural models.

Data Cleaning & Preparation

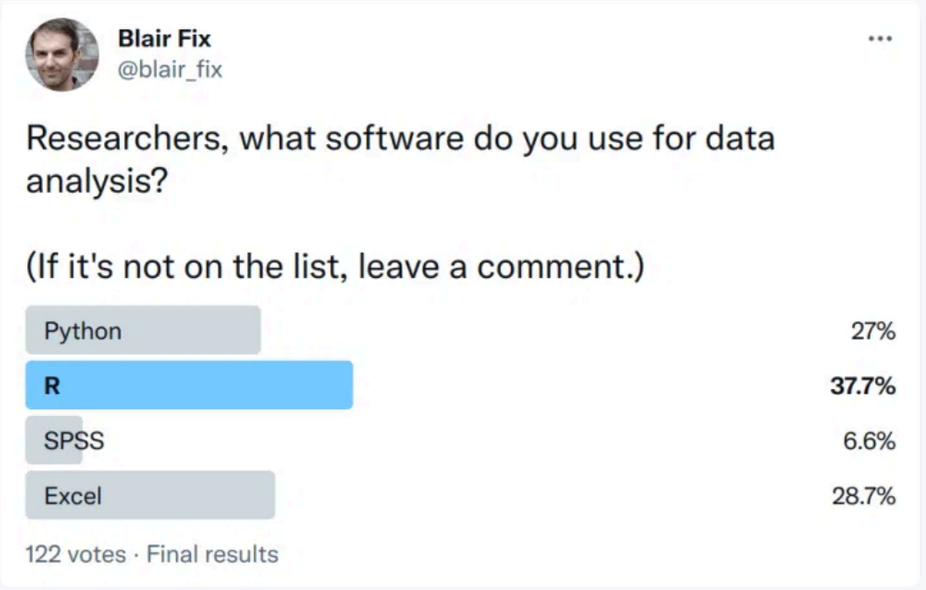
Automate tedious data cleaning tasks for economic datasets, identify outliers, handle missing values, and suggest appropriate transformations based on the data characteristics.

Research Paper Replication

Assist in reproducing published economic research by generating code from paper descriptions, implementing complex economic models, and translating between different programming languages.

Economic Data Visualization

Create effective visualizations for economic analysis, including supply-demand curves, indifference curves, production possibility frontiers, and interactive dashboards for macroeconomic indicators.



Applications for Finance Students

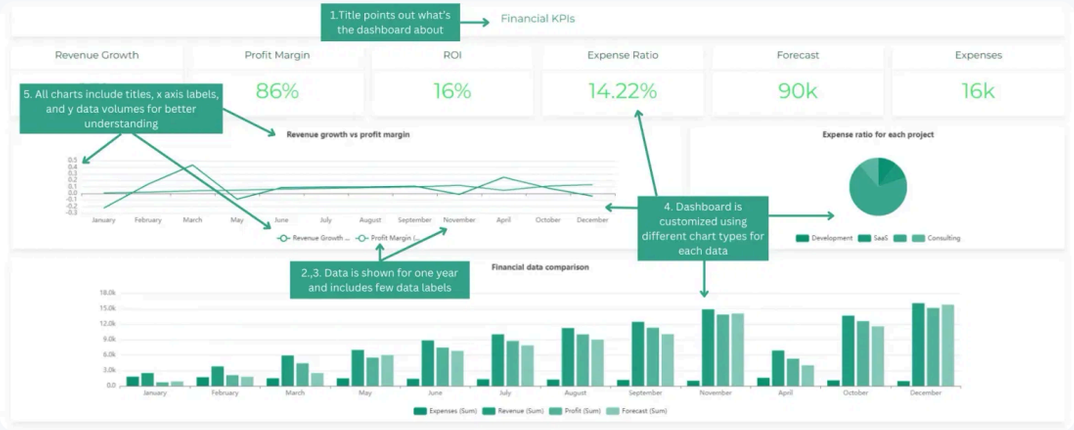
Financial Modeling & Valuation: AI coding tools can help generate and debug complex financial models, DCF analyses, and option pricing models with fewer errors and greater efficiency.

Interactive Dashboards: Create sophisticated financial dashboards and visualizations that update in real-time, making it easier to present financial data to stakeholders.

Algorithmic Trading: Develop and test trading algorithms with AI assistance that can suggest optimizations and identify potential issues in backtesting code.

Risk Analysis: Build more sophisticated risk assessment models that incorporate machine learning techniques for better prediction of market volatility and credit risk.

Financial Reporting Automation: Automate the generation of financial reports and regulatory filings with code that extracts, transforms, and formats financial data.



Getting Started with AI Coding Tools

1. **Choose the Right Tool Type**

   Start with IDE-based tools like Cursor if you're already familiar with code editors. Terminal-based tools like Claude Code are great for command-line users. Autonomous tools like Devin are best for complex projects.

2. **Begin with Small Projects**

   Start by using AI tools for simple data cleaning scripts or basic statistical analyses before moving to complex econometric models or financial simulations.

3. **Learn Effective Prompting**

   Be specific about economic or financial concepts in your prompts. Include details about the data structure, statistical methods, and desired outputs.

4. **Verify and Understand the Code**

   Always review and understand the generated code. Ask the AI to explain complex parts, especially statistical methods or financial calculations.

Helpful Resources

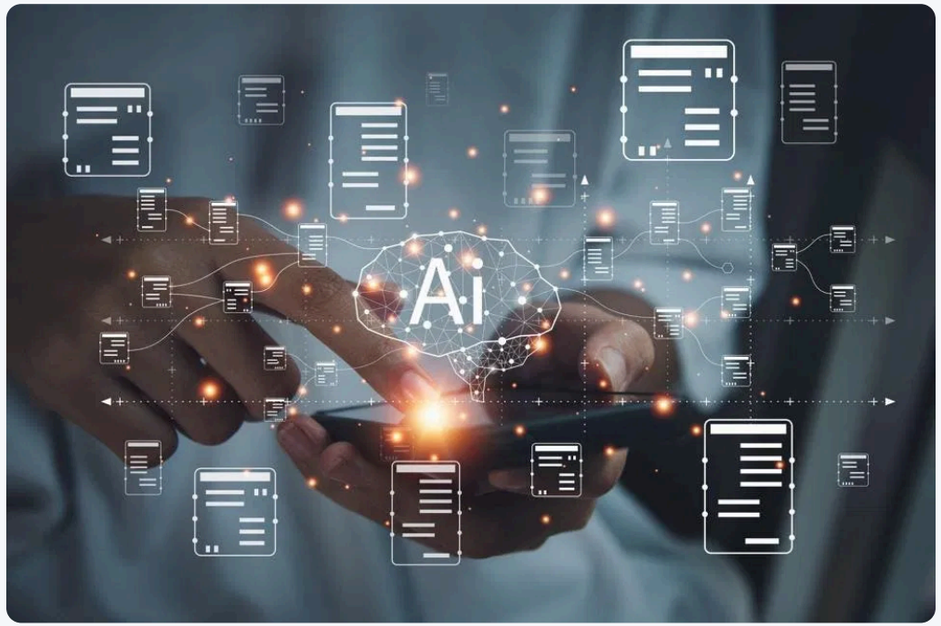
Cursor Academy Claude Code Documentation EconML Library
FinanceBench

Future Trends & Conclusion

Domain-Specific AI Tools: Emergence of AI coding assistants specialized for economics and finance, with built-in knowledge of statistical methods, financial models, and domain-specific libraries.

Enhanced Collaboration: AI tools that can work alongside teams of economists and financial analysts, understanding project context and maintaining consistency across multiple contributors.

Educational Integration: AI coding tools tailored for economics and finance education, helping students learn programming concepts while focusing on domain knowledge.



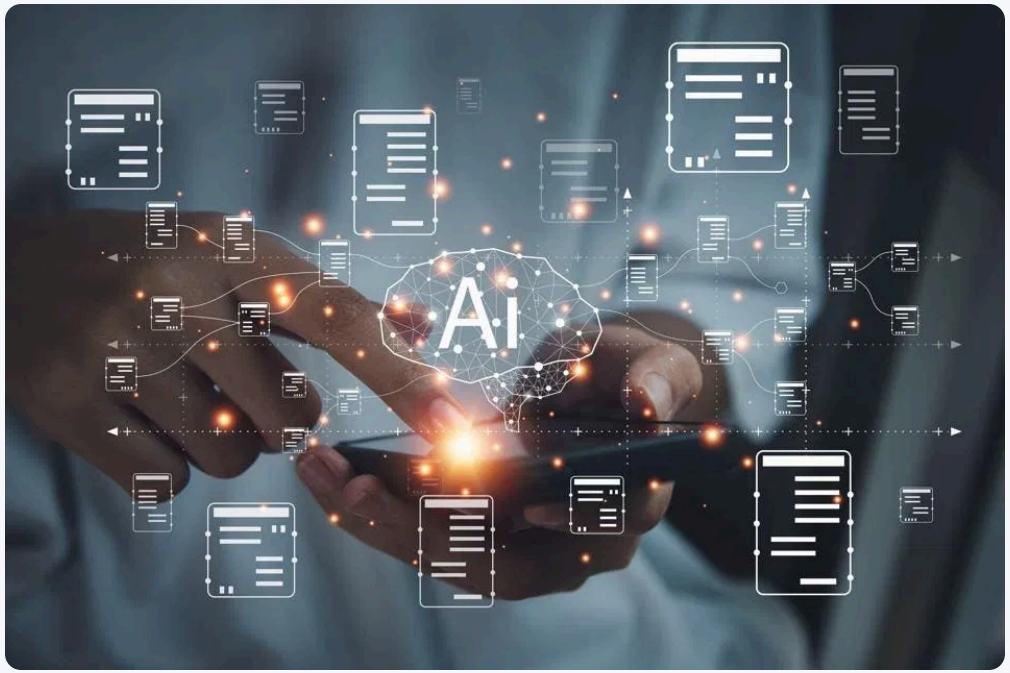
Key Takeaway

AI coding tools are transforming how economics and finance students approach programming tasks. By reducing technical barriers and accelerating development, these tools allow students to focus more on economic theory and financial analysis rather than coding syntax, ultimately preparing them better for an increasingly data-driven industry.

Appendix: Additional AI Coding Tools

This appendix provides installation guides for three additional AI coding tools that can be valuable for economics and finance students. These tools represent different approaches to AI-assisted coding and can complement the tools discussed in the main presentation.

- **OpenAI Codex CLI** - A lightweight coding agent from OpenAI that runs in your terminal, offering AI assistance for coding tasks.
- **Google Gemini CLI** - An open-source AI agent that brings the power of Google's Gemini directly into your terminal.
- **Anthropic Claude Code** - A command line tool for agentic coding developed by Anthropic, providing AI assistance in your terminal.



OpenAI Codex CLI

OpenAI Codex CLI is a lightweight coding agent that runs in your terminal. It provides AI-powered coding assistance directly in your command line interface, making it useful for quick coding tasks and data analysis scripts.

Installation Methods

```
# Using npm (recommended)
npm install -g @openai/codex

# Using Homebrew
brew install codex
```

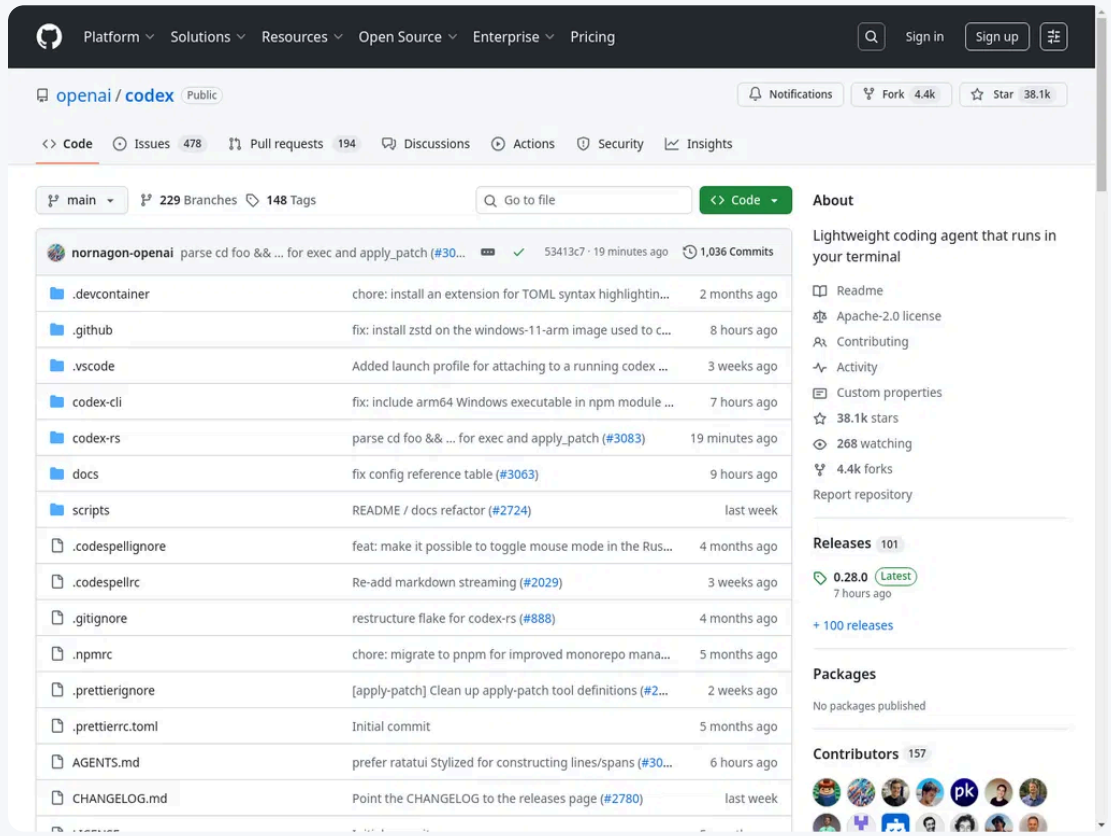
```
# Manual installation from GitHub
# Download from latest GitHub Release:
# https://github.com/openai/codex/releases

# For macOS (Apple Silicon):
# codex-aarch64-apple-darwin.tar.gz

# For macOS (Intel):
# codex-x86_64-apple-darwin.tar.gz

# For Linux:
# codex-x86_64-unknown-linux-musl.tar.gz
```

After installation, run `codex` to start using the tool. Sign in with your ChatGPT account to use Codex as part of your Plus, Pro, Team, Edu, or Enterprise plan.



Google Gemini CLI

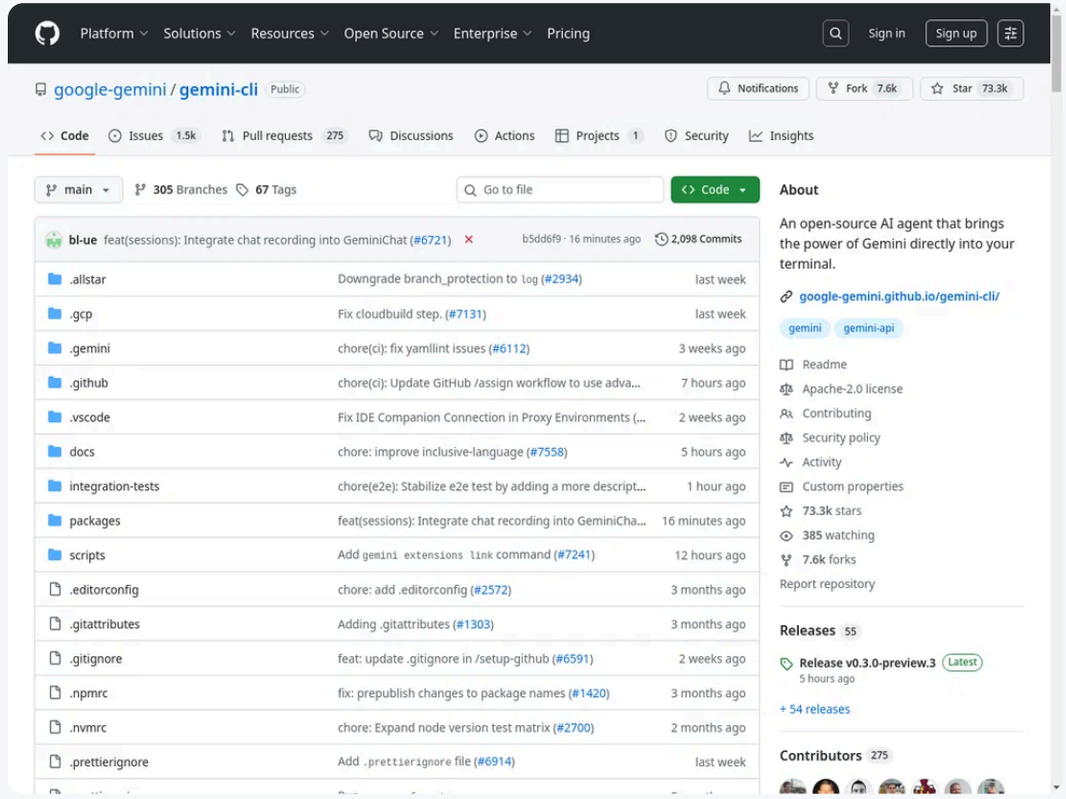
Gemini CLI is an open-source AI agent that brings the power of Google's Gemini directly into your terminal. It provides lightweight access to Gemini, giving you AI assistance for coding, data analysis, and other terminal-based tasks.

Installation Methods

```
# Run instantly with npx (no installation required)
npx https://github.com/google-gemini/gemini-cli
```

```
# Install globally with npm
npm install -g @google/gemini-cli
```

```
# Install globally with Homebrew (macOS/Linux)
brew install gemini-cli
```



- Node.js version 20 or higher
- Compatible with macOS, Linux, or Windows

Anthropic Claude Code

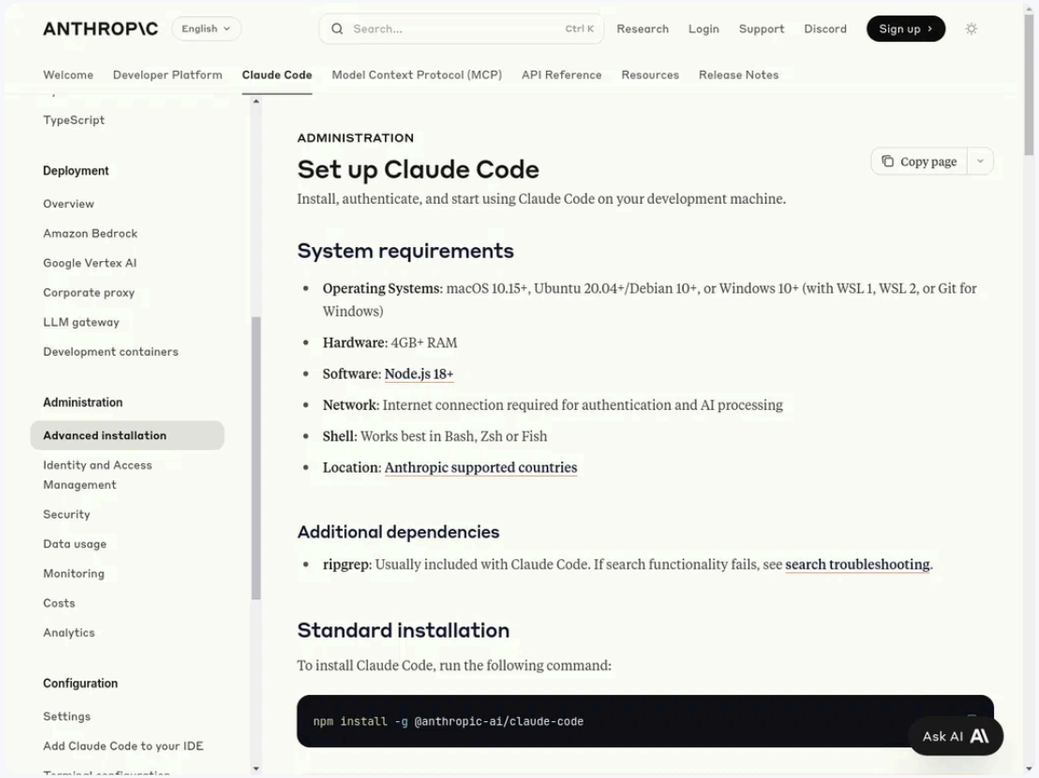
Claude Code is a command line tool for agentic coding developed by Anthropic. It integrates AI assistance directly into your terminal workflow, providing a flexible, customizable, and scriptable coding environment.

Standard Installation

```
# Using npm (standard method)
npm install -g @anthropic-ai/claude-code
```

```
# Native binary installation (Beta)
# For macOS/Linux:
curl -fsSL https://claude.ai/install.sh | bash

# For Windows PowerShell:
irm https://claude.ai/install.ps1 | iex
```



- OS: macOS 10.15+, Ubuntu 20.04+, or Windows 10+ (with WSL)
- Hardware: 4GB+ RAM
- Software: Node.js 18+