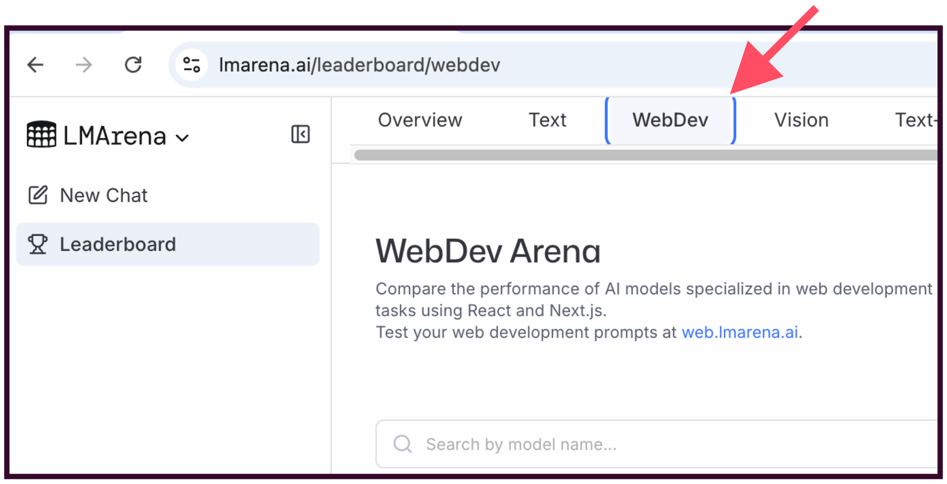
Coding Model Alternatives

ChatGPT isn't the only option out there. While it's one of the top models for generating code, there are other strong contenders. The best part? They're easy to find!

- The Chatbot Arena LLM Leaderboard, developed and maintained by teams at UC Berkeley, is an open-source platform designed to evaluate AI models based on human preferences. Users ask questions to two models (either randomly selected or specifically chosen) and then rate which one performed better. With over 2,000,000 votes collected, it offers valuable insights into model performance. You can check out the leaderboard at https://lmarena.ai/leaderboard.

  Select the "WebDev" category to see which models come out on top for coding.



WebDev Arena

Compare the performance of AI models specialized in web development tasks using React and Next.js.
Test your web development prompts at web.lmarena.ai.

Coding-Specific Tools

As you've seen, ChatGPT can be a powerful coding partner, but having to manually copy and paste code from your chats to IDLE is tedious and error-prone. Fortunately, there are tools that are specifically made to integrate AI into the coding process. They come in four categories: prompt-only, AI IDE, asynchronous agent, and CLI AI agent. Generally speaking, each subsequent category requires more technical expertise to use, but multiplies developer productivity more.

Software engineers and AI engineers will generally start with tools from the AI IDE category.

	prompt only	AI IDE	async agent	CLI AI agent
tech expertise required	near zero	some	more	most
dev productivity multiplied	least	more	most	most

Prompt-Only Tools

These tools work mainly through natural language prompts and visual designs from apps like Figma (https://www.figma.com/design/). They're self-contained environments that let you build and deploy a web app within a single tool.

While these tools let you access the generated code, they're really built for so-called vibe coding: rapid development diven entirely by natural language prompts.

Tools like this are simple to use because they don't require extensive programming knowledge and they combine everything into one package. However, they're limited to relatively simple web-based apps and offer little choice as to languages and frameworks.

Examples include Bolt (https://bolt.new/), Loveable (https://lovable.dev/), Replit (https://replit.com/), and v0 (https://v0.app/).

AI IDEs

An AI integrated development environment (AI IDE) is an IDE with a built-in AI agent. AI IDEs can look over an entire codebase and make changes throughout, working alongside you to edit, run, and test your code.

If you're starting with an existing codebase, you'll often just open the project folder in the AI IDE and let it analyze your code. For new projects, you can prompt in a chat window as you've done when working with ChatGPT. The AI IDE will write code, organize files in appropriate folders, and provide you access to edit the code if needed.

When using AI IDEs, it's often better to start new projects with a software requirements specification (SRS). An SRS is a technical document that describes a system's functional and non-functional requirements and interfaces to other systems. This gives the AI agent a more structured starting point for generating code.

AI IDEs offer more flexibility and support more complex projects than prompt-only tools, but they do require users to understand IDEs. They also require knowledge of version control systems like Git (https://git-scm.com/), which help developers track code changes and collaborate without interfering with each other's work.

Examples include:

Cursor (https://cursor.com/): This is the most popular AI IDE. It offers detaled control of the development process, rich debugging tools, and the ability to deeply customize the interface.

GitHub Copilot (https://github.com/features/copilot): GitHub, a site built on Git for storing and sharing code repositories, developed an AI coding extension for the popular VS Code IDE. GitHub Copilot is notable for its flexible review methods and voice command support.

Windsurf (https://windsurf.com/): Windsurf fully integrates the AI agent into the IDE, providing for a clean interface. It also features extensive task automation capabilities.

An IDE is an application that offers tools for writing, debugging, running, and deploying code. IDLE, included with the standard Python installation, is a basic example of an IDE. More advanced IDEs, such as Visual Studio, Visual Studio Code, Eclipse, and IntelliJ IDEA, come with a wide range of features. Be sure to explore their documentation to learn how they integrate AI tools.

AI agents are AI tools that can act autonomously. Actions include searching the internet, using specialized tools, updating code, etc.

GitHub Copilot is not the only AI extension for VS Code. Check out Qodo, Tabnine, Cody, and Pythagora.

Asynchronous Agents

Both prompt-only tools and AI IDEs are designed to be used interactively. Asycnhronous agents, on the other hand, are designed to be given complex tasks and then set to work over a long period of time (minutes to hours), freeing developers to work on other things while their code is being updated.

In addition to working independently, async agents are typically invoked through a chatbot-like interface, allowing you to use a browser or even a mobile device to work on your codebase. You can also set multiple agents to work in parallel.

Async agents represent a step up in capability from AI IDEs, but demand even more knowledge of how to use Git and GitHub.

Examples include Codex (https://chatgpt.com/features/codex), Devin (https://devin.ai/), and Claude Code Web (https://code.claude.com/docs/en/claude-code-on-the-web).

CLI AI Agents

Command-line interface AI agents are just like async agents, except that they're invoked from a command-line interface. Why does this matter? For experienced developers, the command line enables fast, efficient text-based workflows. Additionally, you can script invocation of CLI AI agents, automating tasks and improving efficiency. Finally, CLI agents can be run from a terminal window inside an IDE, extending async agent capabilities to non-AI IDEs.

Examples include Claude Code (https://claude.com/product/claude-code), Codex CLI (https://developers.openai.com/codex/cli), and Gemini CLI (https://geminicli.com).