How to Run Gemini CLI AI Agent on Terminal

Google has announced Gemini CLI, an open-source tool designed to bring their AI models directly into the developer terminal.

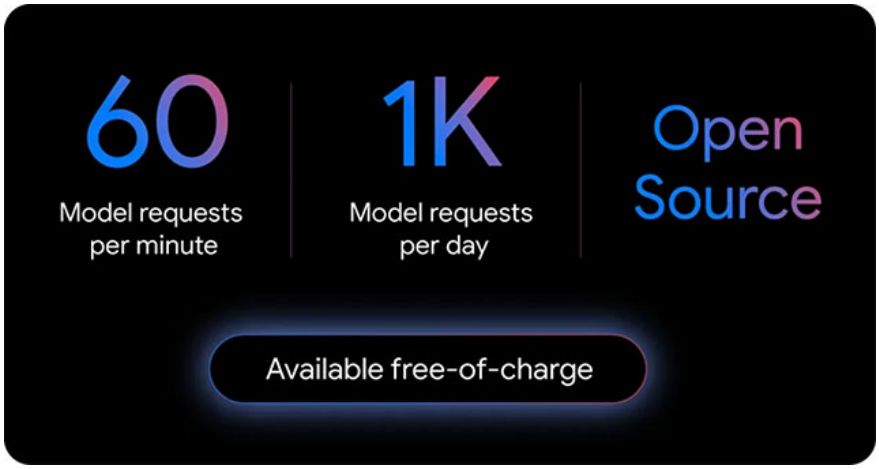
Many people love using the terminal and are always looking for command line tools or utilities that help them perform various tasks directly from the terminal. Recently, Google introduced the open source Gemini CLI AI agent, which brings powerful Gemini AI models right to the command line. Let's dig deeper and see if Gemini CLI is worth exploring.

Note : As with any language model, errors such as misunderstandings, misinterpretations, or ambiguous output can occur. Always check the output carefully, especially for critical tasks, and ensure the output is correct before use.

Google Launches Gemini CLI, Bringing Gemini to the Terminal

Google has announced Gemini CLI, an open-source tool designed to bring its AI models directly into the developer's terminal. The Mountain View giant isn't the first company to think about bringing an AI assistant to the command line. We've seen similar projects from Google competitors in this space, such as OpenAI with its Codex CLI and Anthropic with Claude Code (released last February).

What seems to be Google's main advantage here is its usage limits, which are surprisingly generous for a free service. Logging in with a personal Google account allows users to make up to 60 model requests per minute, and a total of 1,000 free requests per day. This access is directly tied to Gemini 2.5 Pro, Google's powerful model known for its massive context window of up to 1 million tokens, giving individual developers plenty of room to experiment and build without hitting a paywall.



If you're a professional developer who needs to run multiple agents at once or want to use specific models beyond the defaults, Google has other options. You can use Google AI Studio or Vertex AI keys for pay-as-you-go billing, or just purchase the Gemini Code Assist Standard/Enterprise license to unlock additional power.

Here's what you can do with the CLI:

1. Align commands (Ground prompts) with Google Search to retrieve web pages and provide real-time external context to the model
2. Extend the capabilities of the Gemini CLI through built-in support for the Model Context Protocol - (MCP) or packaged extensions
3. Customize prompts and instructions to tailor Gemini to your specific needs and workflow
4. Automate tasks and integrate with existing workflows by calling the Gemini CLI non-interactively in your scripts

Installation is simple, assuming you're already set up for modern web development. To install the Gemini CLI, you'll first need to make sure you have Node.js 18 or higher installed. From there, you can run it directly with **npx** like this:

```
npx https://github.com/google-gemini/gemini-cli
```

Or set it globally for easier access later:

```
npm install -g @google/gemini-cli
```

Since the tool is open source, anyone can review the source code (hosted on GitHub) to understand exactly how it works or verify its behavior. Powered by Yargs and licensed under Apache 2.0, the tool will be familiar to anyone who has built CLI applications with Node.js, making it easy to contribute.

The CLI also shares technology with Google's Code Assist IDE plugin. This creates a more unified experience for developers using Google's AI tools, whether they're working in VS Code or directly in the terminal. This integration works for free tier users and developers subscribed to the Standard or Enterprise plans.

You can find more information in Google's official announcement post .

Gemini CLI Setup

Whether you're using Linux, macOS, or Windows, setting up the Gemini CLI is simple. Since it's distributed as an npm package, you'll need to install Node.js and npm. You can download both packages from the official Node.js website .

For example, if you are running Linux, first download and install fnm (Fast Node Manager) using the following command:

```
curl -o- https://fnm.vercel.app/install | bash
```

Next, use fnm to download and install Node.js:

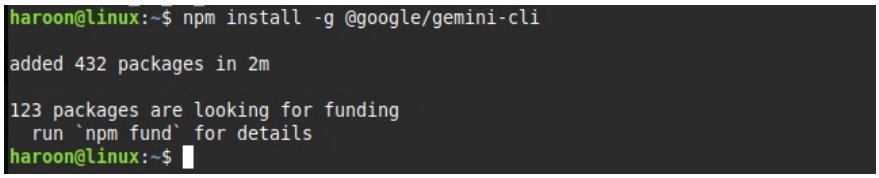
```
fnm install 22
```

That's it! You have successfully downloaded and installed Node.js. To check the version number, run the following command:

```
node -v npm -v
```

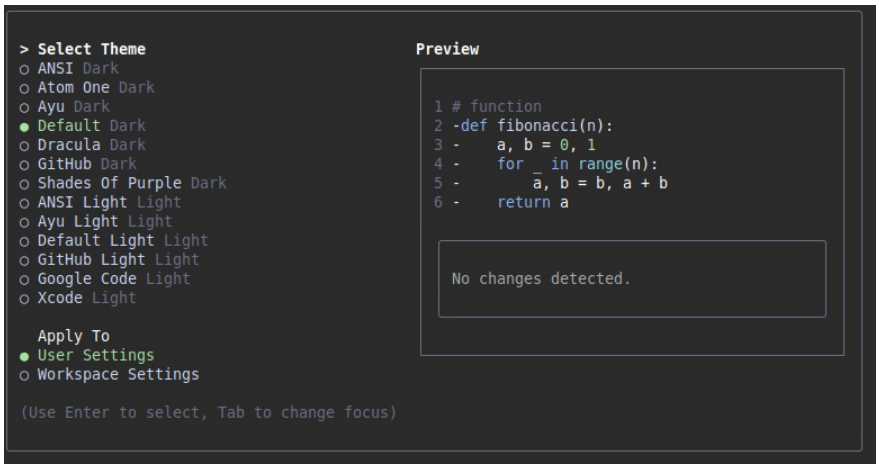
Once installed, use npm (Node Package Manager) to install Gemini CLI system-wide:

```
npm install -g @google/generative-ai
```



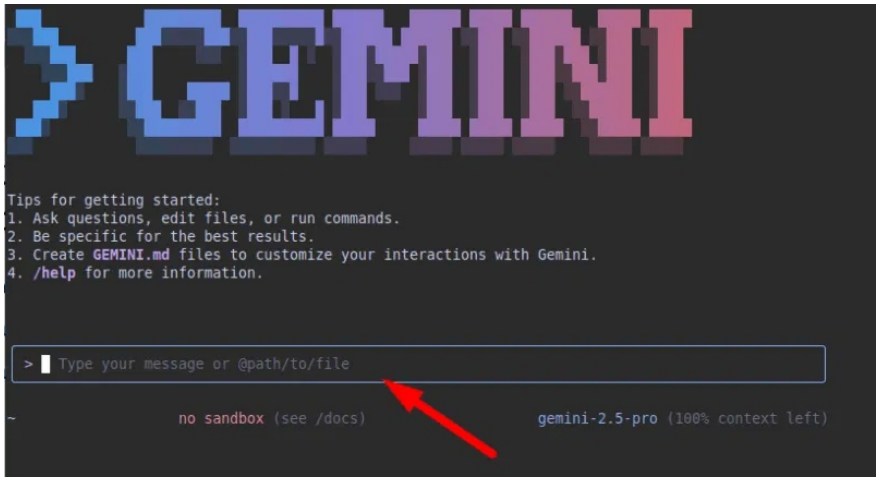
The -g option installs the package system-wide, allowing the gemini command to be accessible from almost anywhere in the terminal.

Once installed, run the gemini command in the terminal to launch the Gemini CLI. The first step is to choose the interface you want.



Next, choose your authentication method. Choose to sign in with a Google account to access the free plan, which should be enough for most users. If you're working on a larger project, you can upgrade to a paid plan for higher limits.

This will open a new browser window where you will need to provide your Google account credentials and grant the necessary permissions to Gemini CLI. Once successfully authenticated, you are ready to start using it!



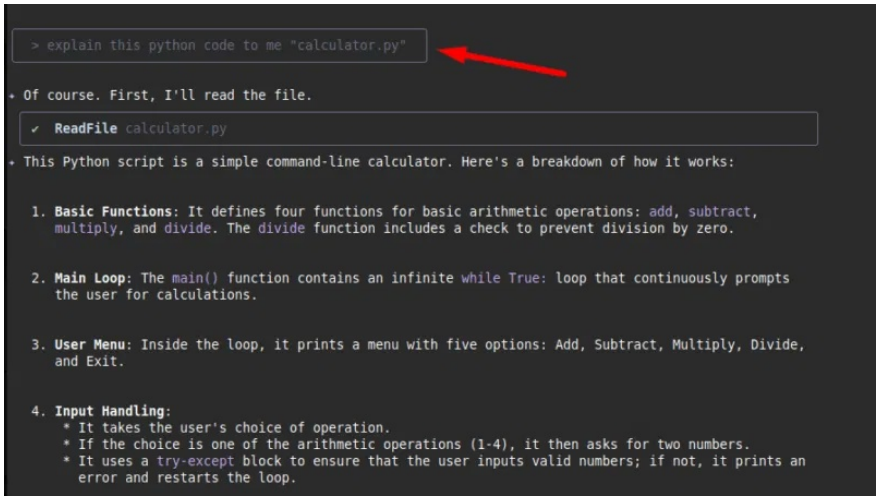
Before you begin, take a moment to explore the built-in help system. Type **gemini --help** into the terminal. You'll get a full list of available commands and options.

Real-world example of using Gemini CLI

You can do almost anything with Gemini CLI as you would with any other chatbot. However, its main advantage is its seamless integration with your system. This integration makes it easy to perform many tasks in your daily development workflow.

For example, with the Gemini CLI, you can ask it to generate and save code to a specific location, and perform various operations on that code directly in the terminal. Additionally, you can use it to explain existing code. If you come across a complex function that you don't understand, you can simply ask for an explanation. Just open the Gemini CLI and type:

```
gemini > Explain this Python code to me "path/to/your/file.py"
```



It will provide a clear and concise explanation of the code's functionality.

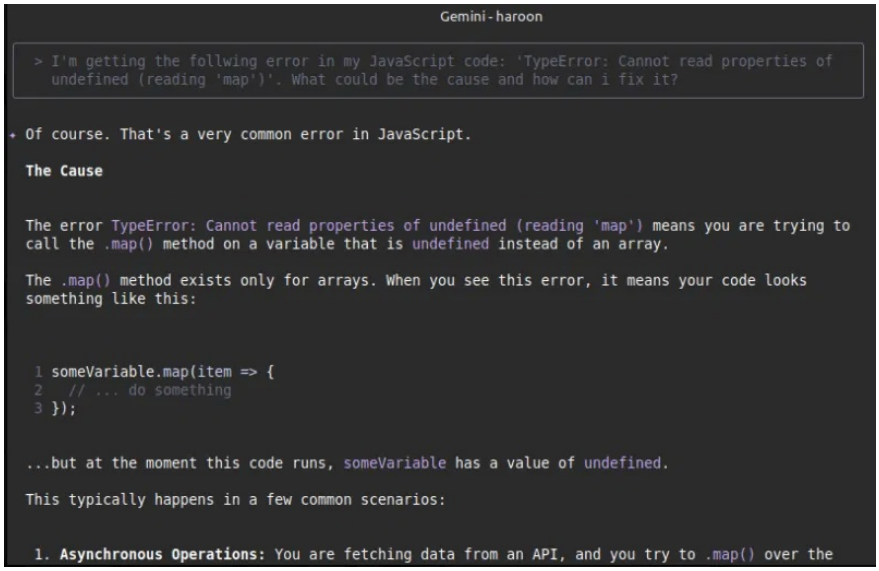
Let's say you have a folder containing a set of images that you want to rename based on their creation date. Instead of writing a script or doing it manually, try:

```
gemini > Rename all JPEG files in this directory to include their creation date
```

Gemini CLI will parse files, extract metadata, and rename them (e.g. photo.jpg to 2025-07-09_photo.jpg). This is a significant time saver for repetitive tasks.

Additionally, you can also get instant debugging help using Gemini CLI. Just paste the error message or you can also provide a screenshot of the error and ask for analysis and potential solutions.

```
gemini > I'm getting the following error in my JavaScript code: 'TypeError: Canno
```



Additionally, the Gemini CLI can support Git operations. It can write commit messages, create pull request descriptions, or summarize code changes — saving you time while maintaining a clean version history. These are just a few examples. You can explore many more uses, such as creating Bash scripts, generating

documentation, organizing files, and more.

As AI continues to evolve and improve, tools like the Gemini CLI or Warp AI terminal will become more powerful and more tightly integrated into our development workflows. Getting familiar with these tools now means you'll be ready to take advantage of more advanced features when they're released.

So you should give Gemini CLI a try. It's easy to set up, free to use for most personal needs, and will likely become your new favorite development companion.

You finished reading the article "**How to Run Gemini CLI AI Agent on Terminal**" 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.

© 2019 TipsMake.com