

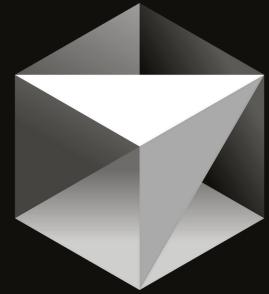
# 17 MUST-HAVE AI TOOLS FOR DEVELOPERS



SLIDE TO EXPLORE ➔



# 1 Cursor IDE



The AI-powered VS Code alternative.

Write, debug, and refactor code with an embedded AI pair programmer.

CURSOR

Pricing   Features   Enterprise   Blog   Docs   Careers   Sign in   Download

# The AI Code Editor

Built to make you extraordinarily productive, Cursor is the best way to code with AI.

[Download for macOS](#)   [All Downloads](#)

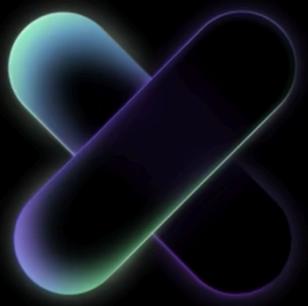
The screenshot shows the Cursor IDE interface. On the left, there's a code editor window for 'lib.rs' with several lines of Rust code. A context menu is open over some selected code. On the right, there are two floating windows: one titled 'Add depth field for tree traversal' containing AI-generated code completion suggestions, and another titled 'lib.rs' showing a detailed breakdown of a Merkle tree struct with its fields and methods. At the bottom, there are status bars for file changes and a search bar.

Follow OM NALINDE to learn more about AI Agents

Repost

2

# GitHub Copilot X



GitHub Copilot X

**Autocomplete on steroids.**

**Converts comments → code, suggests full functions, and integrates deeply with GitHub.**

## GitHub Copilot X: The AI-powered developer experience

GitHub Copilot is evolving to bring chat and voice interfaces, support pull requests, answer questions on docs, and adopt OpenAI's GPT-4 for a more personalized developer experience.

```

GITHUB COPILOT
Here's the corrected code:

import datetime

def parse_expenses(expenses_string):
    """Parse the list of expenses and return the list of triples
    (date, value, currency).
    Ignore lines starting with #.
    Parse the date using datetime.
    Example expenses_string:
        2023-01-02 -34.01 USD
        2023-01-03 2.59 DKK
        2023-01-03 -2.72 EUR
    """
    expenses = []
    for line in expenses_string.splitlines():
        if line.startswith("#"):
            continue
        date, value, currency = line.split()
        expenses.append((datetime.datetime.strptime(date, "%Y-%m-%d"),
                         float(value),
                         currency))
    return expenses

This should fix the issues and correctly parse the expenses data.

What other methods can be used to parse the date in the expenses data
Ask a question or type '/' for topics

```

```

add_elements.py M X
add_elements.py > parse_expenses
import datetime

def parse_expenses(expenses_string):
    """Parse the list of expenses and return the list of triples
    (date, value, currency).
    Ignore lines starting with #.
    Parse the date using datetime.
    Example expenses_string:
        2023-01-02 -34.01 USD
        2023-01-03 2.59 DKK
        2023-01-03 -2.72 EUR
    """

    expenses = []
    for line in expenses_string.splitlines():
        if line.startswith("#"):
            continue
        date, value, currency = line.split()
        expenses.append((datetime.datetime.strptime(date, "%Y-%m-%d"),
                         float(value),
                         currency))

return expenses

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

File "/Users/martin/src/samples/add\_elements.py", line 17,  
in parse\_expenses  
 date, value, currency = line.split("#")  
ValueError: not enough values to unpack (expected 3, got 1)

Ln 23, Col 20 Spaces: 4 UTF-8 LF Python 3.11.2 64-bit

3

# LangChain



**Build LLM-powered apps with ease.**

**Framework for connecting data, APIs, and agents in production-ready workflows.**

STACK 1: LangGraph + LangChain + LangSmith + LangGraph Platform

A full product suite for reliable agents and LLM apps

LangChain's products work seamlessly together to provide an integrated solution for every step of the application development journey. When you use all LangChain products, you'll build better, get to production quicker, and grow visibility – all with less set up and friction.

LangChain provides the smoothest path to high quality agents.

Orchestration:  LangGraph

Integrations:  LangChain

Evals + Observability:  LangSmith

Deployment:  LangGraph Platform

STACK 2: No framework + LangSmith

Trace and evaluate any LLM app

LangSmith is framework-agnostic. Trace using the TypeScript or Python SDK to gain visibility into your agent interactions – whether you use LangChain's frameworks or not.

Orchestration: Your choice

Evals + Observability:  LangSmith



**Multi-agent framework for AI collaboration.**

**Let multiple AI agents talk to each other  
and complete complex tasks.**

**AutoGen**

Open-Source Framework for Agentic AI

[aka.ms/autogen](https://aka.ms/autogen)

[autogen@microsoft.com](mailto:autogen@microsoft.com)

[Overview](#) [People](#) [Publications](#) [Downloads](#) [Videos](#) [News & features](#)

AutoGen is an open-source programming framework for building AI agents and facilitating cooperation among multiple agents to solve tasks. AutoGen aims to provide an easy-to-use and flexible framework for accelerating development and research on agentic AI.

Over the past year, our work on AutoGen has highlighted the transformative potential of agentic AI in addressing real-world challenges through agents and multi-agent applications. Building on this progress, we are excited to announce AutoGen v0.4—a significant milestone shaped by learning and valuable feedback from our community of users and developers. This update represents a complete redesign of the AutoGen library, aimed at improving code quality, robustness, generality, and the scalability of agentic workflows.

The initial release of AutoGen generated widespread interest in agentic technologies. At the same time, users faced challenges scaling applications due to limited support for dynamic workflows and debugging tools. Feedback highlighted the need for stronger observability, more flexible collaboration patterns, and for reusable components. AutoGen v0.4 addresses these issues with its asynchronous, event-driven architecture. AutoGen v0.4 adopts a more robust, asynchronous, and event-driven architecture, enabling a broader range of agentic scenarios with stronger observability, more flexible collaboration patterns, and for reusable components.

5

# Pinecone



## Vector database for AI apps.

## Power your RAG pipelines with lightning-fast embeddings search.

Pinecone



DEVELOPER EXPERIENCE

### Scale simplified

Fully managed and serverless for effortless scaling.

### Rapid setup

Launch your vector databases in seconds.

### Serverless scaling

Resources adjust to meet your demand automatically.

### Rock-solid reliability

Trust in consistent uptime for your critical applications.

agent/retriever.py

Quickstart guide

```
1  from pinecone import Pinecone
2
3  pc = Pinecone("<API KEY>")
4
5  pc.Index("semantic-search")
6
7  index.query(
8      namespace="breaking-news"
9      vector=[0.13, 0.45, 1.34, ...],
10     filter={"category": {"$eq": "technology"}},
11     top_k=3
12 )
```



**Scalable vector DB with hybrid search.**

**Enables semantic + keyword search across your data.**

 weaviate [Product ▾](#) [Solutions ▾](#) [Developers ▾](#) [Company ▾](#) [Pricing](#) [!\[\]\(95f36e31cdb1a43835207b277a635db7\_img.jpg\) Try Now](#) [Contact](#)

Your stack's missing piece for RAG, agents, and everything coming next

WHAT ARE YOU BUILDING?

**AI-POWERED SEARCH**

Smart, contextual search across unstructured data

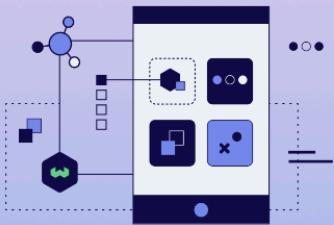
[Learn more](#)



**RETRIEVAL AUGMENTED GENERATION**

Trustworthy chat experiences grounded in your data

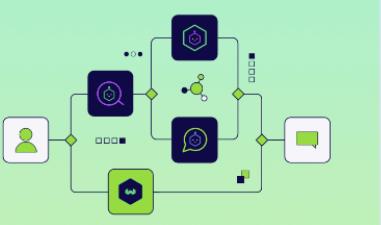
[Learn more](#)



**AGENTIC AI**

Knowledgeable AI agents and agentic workflows

[Learn more](#)





## Benchmark and test LLM apps.

**Evaluate accuracy, reliability, and performance of AI workflows.**

### Evals

Create, manage, and run evals in the OpenAI platform. Related guide: [Evals](#)

### Create eval

```
POST https://api.openai.com/v1/evals
```

Create the structure of an evaluation that can be used to test a model's performance. An evaluation is a set of testing criteria and the config for a data source, which dictates the schema of the data used in the evaluation. After creating an evaluation, you can run it on different models and model parameters. We support several types of graders and datasources. For more information, see the [Evals guide](#).

#### Example request

```
curl https://api.openai.com/v1/evals \
-H "Authorization: Bearer $OPENAI_API_KEY"
-H "Content-Type: application/json" \
-d '{
  "name": "Sentiment",
  "data_source_config": {
    "type": "stored_completions",
    "metadata": {
      "usecase": "chatbot"
    }
  },
  "testing_criteria": [
    {
      "type": "label_model",
      "model": "o3-mini",
      "input": [
        ...
      ]
    }
  ]
}'
```

curl ⌂



# Hugging Face Transformers



Hugging Face

The ultimate ML toolkit.

Evaluate accuracy, reliability, and performance of AI workflows.

Transformers



Transformers acts as the model-definition framework for state-of-the-art machine learning models in text, computer vision, audio, video, and multimodal model, for both inference and training.

It centralizes the model definition so that this definition is agreed upon across the ecosystem. `transformers` is the pivot across frameworks: if a model definition is supported, it will be compatible with the majority of training frameworks (Axolotl, Unslot, DeepSpeed, FSDP, PyTorch-Lightning, ...), inference engines (vLLM, SGLang, TGI, ...), and adjacent modeling libraries (llama.cpp, mlx, ...) which leverage the model definition from `transformers`.

We pledge to help support new state-of-the-art models and democratize their usage by having their model definition be simple, customizable, and efficient.

There are over 1M+ Transformers [model checkpoints](#) on the [Hugging Face Hub](#) you can use.

Explore the [Hub](#) today to find a model and use Transformers to help you get started right away.



Follow [OM NALINDE](#) to learn more about AI Agents

Repost



## Serverless for AI workloads.

**Run ML inference, pipelines, and infra with zero DevOps.**

 **Modal**    [Use Cases](#)    [Playground](#)    [Pricing](#)    [Customers](#)    [Blog](#)    [Docs](#)    [Company](#)    [Log In](#)    [Sign Up](#) 

### Generative AI Inference that scales with you

[View Examples](#)

---

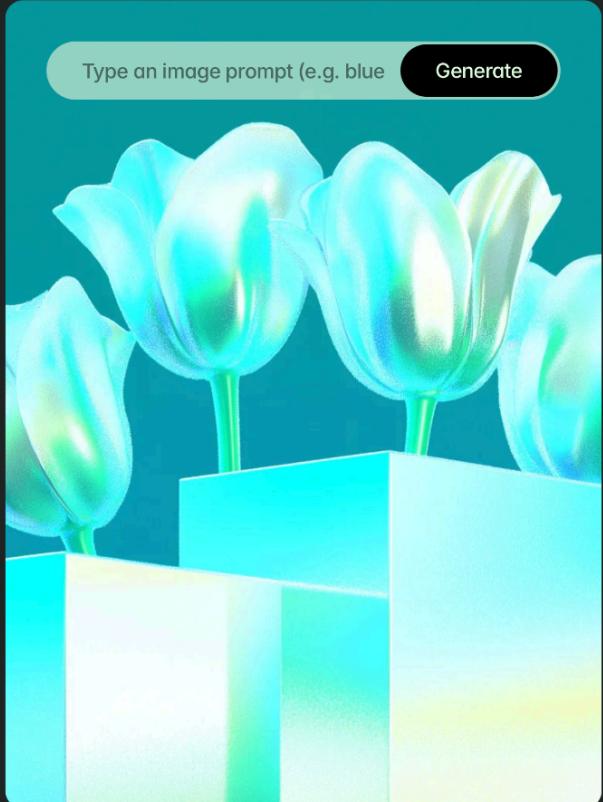
 **Fast cold boots**  
Load gigabytes of weights in seconds with our optimized container file system.

---

 **Bring your own code**  
Deploy anything from custom models to popular frameworks.

---

 **Seamless autoscaling**  
Handle bursty and unpredictable load by scaling to thousands of GPUs and back down to zero.



10

# Vercel v0



## Serverless for AI workloads.

### Run ML inference, pipelines, and infra with zero DevOps.

Community Enterprise Pricing iOS FAQ Sign In Sign Up

## Prompt. Refine. Ship.

Ask v0 to build... +

From the Community

Browse All >

Explore what the community is building with v0.

**Beautiful Shader Experiences**

Create stunning visual experiences with our advanced shader technology. Interactive lighting, real-time rendering, and more. Try it now!

Shaders Hero Section

5510 · 737

**Pointer AI landing page**

9136 · 494

Unleash the Power of AI Agents

Accelerate your development workflow with intelligent AI agents that write, review, and optimize your code.

Try for Free

**M.N.A.Y. - MASTERS OF THE ARSENAL**

OVERVIEW

49% INITIALLY SCOPED

642 POINTS LOST

0 ACCIDENTS

12:01 PM

TUESDAY AUGUST 5, 2023

INDUS AEROS, ARGENTINA

NOTIFICATIONS PAYMENT REQUESTED

Follow OM NALINDE to learn more about AI Agents

Repost

11

# Windsurf



The next-gen AI IDE.

**Combines powerful AI coding assistance with project-aware context for smoother development.**

The screenshot displays the Windsurf AI IDE interface. At the top, there's a navigation bar with links for PRODUCTS, ENTERPRISE, PRICING, BLOG, RESOURCES, COMPANY, and DOWNLOAD. Below the navigation bar is a toolbar with icons for file operations like Open, Save, and Run. The main workspace shows a code editor for a file named "page.tsx". The code contains a function "Home" with components "Header", "Game", and "Footer". A tooltip from the AI agent provides instructions: "Let me check if the components directory exists and what's inside it." Below the code editor, a sidebar shows a list of generated files: "Analyzed app/components" and "Created Header.tsx +14 -0 0 Open diff". The AI agent is described as creating necessary components for a typing speed test application. Step 1 involves creating the "Header" component, which has been successfully created ("Created Header.tsx"). Step 2 involves creating the "Game" component, which is currently being generated ("Creating Game.tsx"). The bottom of the interface features a footer with social media links for LinkedIn, GitHub, and YouTube, along with a "Follow" button and a bio for "OM NALINDE".

Follow OM NALINDE to learn more about AI Agents

Share



## Superfast, multiplayer code editor.

### Enables collaborative coding with blazing speed and integrated AI completion.

[Product](#) [Extensions](#) [Docs](#) [Blog](#) [Pricing](#) [Resources](#) [Sign up](#) [Download](#)

<b>Fast</b> Written from scratch in Rust to efficiently leverage multiple CPU cores and your GPU.	<b>Intelligent</b> Integrate upcoming LLMs into your workflow to generate, transform, and analyze code.	<b>Collaborative</b> Chat with teammates, write notes together, and share your screen and project. All included.
--	--	---

The screenshot shows the Zed editor interface. On the left is a sidebar with a tree view of project files: welcome, workspace, src (containing persistence, dock.rs, items.rs, modal\_layer.rs, notifications.rs, pane.rs, pane\_group.rs, persistence.rs, searchable.rs, shared\_screen.rs, status\_bar.rs, tasks.rs, theme\_preview.rs, toolbar.rs, workspace.rs, workspace\_settings.rs), Cargo.toml, LICENSE-GPL, worktree, zed (contents, resources), and build.rs. The main area shows a file named "zed.rs" with the following code:

```

1 mod app_menus;
2 pub mod inline_completion_registry;
3 #[cfg(any(target_os = "linux", target_os = "freebsd"))]
4 pub(crate) mod linux_prompts;
5 #[cfg(target_os = "macos")]
6 pub(crate) mod mac_only_instance;
7 mod open_listener;
8 mod quick_action_bar;
9 #[cfg(target_os = "windows")]
10 pub(crate) mod windows_only_instance;
11
12 use anyhow::Context as _;
13 pub use app_menus::*;
14 use assets::Assets;
15 use assistant::PromptBuilder;
16 use breadcrumbs::Breadcrumbs;
17 use client::{zed_urls, ZED_URL_SCHEME};
18 use collections::VecDeque;
19 use command_palette_hooks::CommandPaletteHooker;
20 use editor::ProposedChangesEditorToolbar;
21 use editor::{scroll::Autoscroll, Editor, MultiBuffer};
22 use feature_flags::FeatureFlagAppExt;
23 use futures::channel::mpsc, select_biased, StreamExt;
24 use gpu::{
25     actions, point, px, ApplicationContext, AsyncApplicationContext, Context, FocusableView, MenuItem,
26     PathPromptOptions, PromptLevel, ReadGlobal, Task, TitlebarOptions, View, ViewContext,
27     VisualContext, WindowKind, WindowOptions,
28 };
29 pub use open_listener::*;
30 use outline_panel::OutlinePanel;
31 use paths::{local_settings_file_relative_path, local_tasks_file_relative_path};
32 use project::DirectoryLister, Item;
33 use project_panel::ProjectPanel;
34 use quick_action_bar::QuickActionBar;
35 use recent_projects::open_ssh_project;

```

The status bar at the bottom right shows the time as 15:12, the language as Rust, and the frame rate as 120 FPS.

## AI-native cloud IDE.

**Lets you code, run, and deploy entirely in the browser with AI-assisted workflows.**

Interact with Claude Code
Example 1
Example 2

Powerful intelligence
Works where you work
You're in control

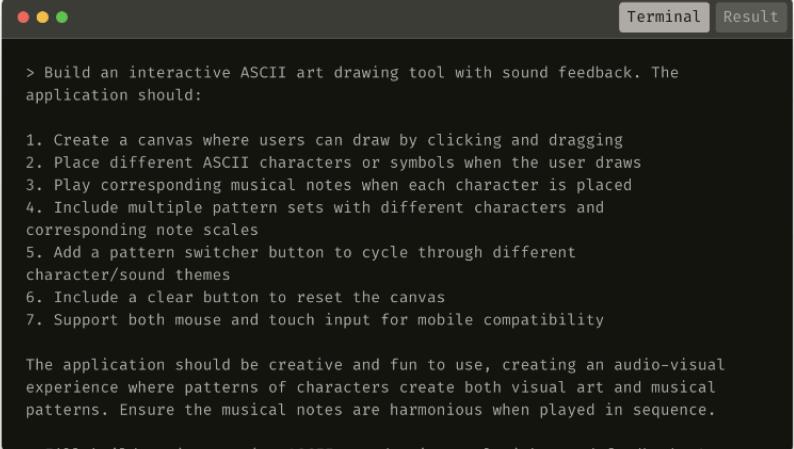
- Uses agentic search to understand your entire codebase without manual context selection
- Makes coordinated changes across multiple files
- Optimized specifically for code understanding and generation with Claude Opus 4.1

**Works where you work**

- Lives right inside your terminal—no context switching
- Integrates with VS Code and JetBrains IDEs
- Leverages your test suites and build systems

**You're in control**

- Never modifies your files without explicit approval
- Adapts to your coding standards and patterns
- Configurable; build on the SDK or run on GitHub Actions

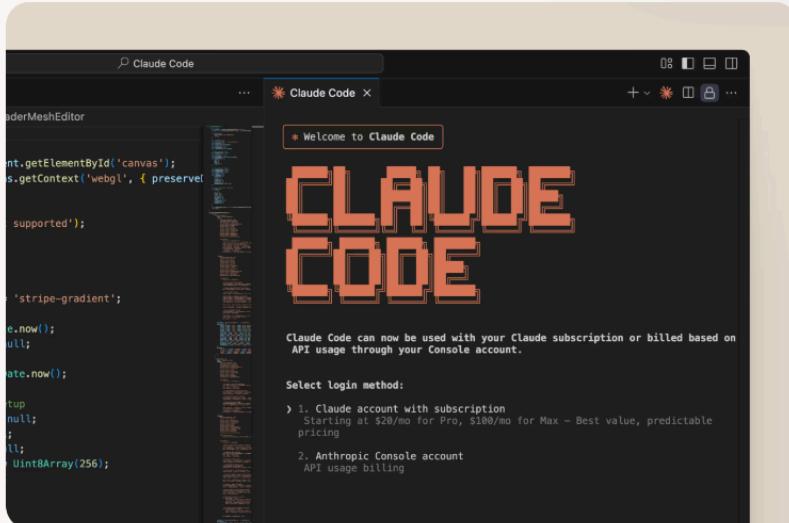


> Build an interactive ASCII art drawing tool with sound feedback. The application should:

1. Create a canvas where users can draw by clicking and dragging
2. Place different ASCII characters or symbols when the user draws
3. Play corresponding musical notes when each character is placed
4. Include multiple pattern sets with different characters and corresponding note scales
5. Add a pattern switcher button to cycle through different character/sound themes
6. Include a clear button to reset the canvas
7. Support both mouse and touch input for mobile compatibility

The application should be creative and fun to use, creating an audio-visual experience where patterns of characters create both visual art and musical patterns. Ensure the musical notes are harmonious when played in sequence.

... still build an interactive ASCII art drawing tool with sound feedback. Let me



Works with your IDEs

Claude works directly in VS Code and JetBrains, seeing your entire codebase instead of just isolated snippets. It understands your project structure and existing patterns, making suggestions that actually fit and presenting them directly in your code files. No copying and pasting—just building.






## AI-powered terminal.

**Turns command line into a modern, smart workspace with AI autocomplete and command explanations.**

Introducing Warp Code: the fastest way from prompt to production

Zach Lloyd September 3, 2025

[←](#) Product



Today, we're launching Warp Code — a suite of features for shipping agent-generated code all the way from prompt to production.

With [Warp Code](#) you get:

- **Top-rated coding agent:** #1 on Terminal-bench (52%) and top three on SWE-bench Verified (75.8%, scored with GPT-5) as of Sep 2nd 2025. We built the UI from the ground up to be the best experience for agentic coding.
- **Code review:** Review open changes, ask for modifications, and line-edit code diffs in a dedicated panel
- **Code editing:** A lightweight file viewing and editing experience in Warp with tabbed file viewing, a file tree, and syntax highlighting

15

Replit



Code, host, and ship in one place.

AI-assisted coding + deployment in the cloud → great for prototyping fast.



Products ▾

For Work ▾

Resources ▾

Pricing

Careers

Contact sales

Log in

Sign up

Turn your ideas into apps

What will you create? The possibilities are endless.

Get suggestions

Write a prompt

Make me an automation  
for content creators  
that helps respond to common  
customer inquiries instantly

Integrate with tools they already use

Start building with AI ➤

Follow OM NALINDE to learn more about AI Agents

Repost

16

# Codex



AI co-pilot for full-stack devs.

Helps generate code, debug errors, and accelerate app development across multiple frameworks.

## Codex

Research Preview

A cloud-based software engineering agent that answers codebase questions, executes code, and drafts pull requests.

[Upgrade to try it](#)

[Learn more ↗](#)

Available to users on ChatGPT Plus, Pro, Business, Enterprise and Edu plans

The screenshot shows the Codex AI interface. At the top, there's a header with the Codex logo and a pink circular icon. Below the header, a message says "What are we coding next?". A text box contains a question about updating tests for a new API endpoint. Below the message are two dropdown menus: "monorepo" and "main". To the right are "Ask" and "Code" buttons. A navigation bar at the bottom has "Tasks" and "Archive" tabs, with "Tasks" being active. Three tasks are listed:

- Diagnose issues with status updates (Just now · monorepo) - Status: +76 -47
- Convert non-critical components to React.lazy with Suspense fallbacks (2h ago · monorepo) - Status: Merged, +142 -12
- Fix uncontrolled to controlled warning in ToggleGroup (3h ago · monorepo) - Status: Merged, +158 -25

Follow  OM NALINDE to learn more about AI Agents

 Repost

Google's AI in your terminal.

**Run Gemini models directly from the command line to generate, refactor, and document code.**



```

    ➜ GEMINI

Tips for getting started:
1. Ask questions, edit files, or run commands.
2. Be specific for the best results.
3. ./help for more information.

> write a short paragraph about why Gemini CLI is awesome
+ I will start by searching the web for "Gemini CLI" to understand its main features and purpose. Following that, I'll examine the README.md file for a project-specific overview. I will then consult docs/index.md and docs/cli/commands.md to gain a deeper understanding of its capabilities. Finally, I will write a short paragraph summarizing why Gemini CLI is awesome.

-- GoogleSearch Searching the web for: "Gemini CLI features and purpose"

: Uncovering Gemini's Awesome (esc to cancel, 21s)

Using 3 GEMINI.md files
~/code/gemini-cli (release*)      no sandbox (see /docs)      gemini-2.5-pro (99% context left)

```

Gemini CLI is an open-source AI agent that brings the power of Gemini directly into your terminal. It provides lightweight access to Gemini, giving you the most direct path from your prompt to our model.

### 🚀 Why Gemini CLI?

- 🎯 **Free tier:** 60 requests/min and 1,000 requests/day with personal Google account
- 🧠 **Powerful Gemini 2.5 Pro:** Access to 1M token context window
- 🔐 **Built-in tools:** Google Search grounding, file operations, shell commands, web fetching
- 📡 **Extensible:** MCP (Model Context Protocol) support for custom integrations
- 💻 **Terminal-first:** Designed for developers who live in the command line
- 🏷️ **Open source:** Apache 2.0 licensed

**Interested in  
more content like this?**

**Follow me :  
OM NALINDE**

