



# Product Intro & Fundamentals of AI Coding



# Key Resources

## Cursor Docs

[cursor.com/docs](https://cursor.com/docs)

+ AI Docs Assistant

+ “Cursor Learn” Track

what are the benefits of Cursor's semantic search capabilities?

∞ Agent



## Cursor Workshops

[cursor.com/workshops](https://cursor.com/workshops)

Live + Recorded

Developer Productivity Trends from 2025

By Rohan Chandra

📅 Jan 27, 12:00 PM - 12:30 PM EST



Cursor 201

By Jules Mei

📅 Jan 29, 1:00 PM - 2:00 PM EST



Cursor for Financial Services

By Hugo Charé

📅 Feb 3, 12:00 PM - 12:30 PM EST



Cursor for Security Teams

By Travis McPeak

📅 Feb 5, 1:00 PM - 1:45 PM EST



## Cursor Learn

[cursor.com/learn](https://cursor.com/learn)

Updated guidance for 2026.

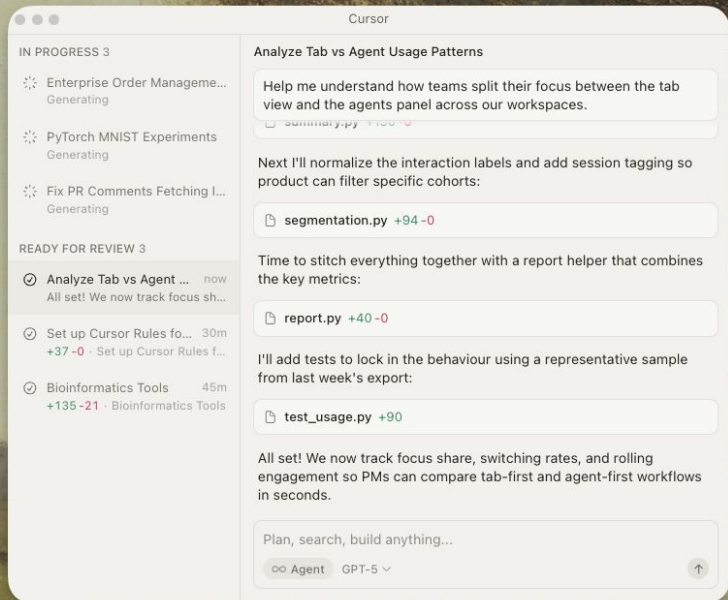


# The Cursor Ecosystem

Cursor is the leading AI coding platform for professional software engineers.

Access all frontier AI models, with built-in enhancements to make them smarter, faster, and more codebase-aware.

It's extremely effective for teams out-of-the-box, *and* allows for limitless customization on top.



IDE

Terminal/CLI

Agent API

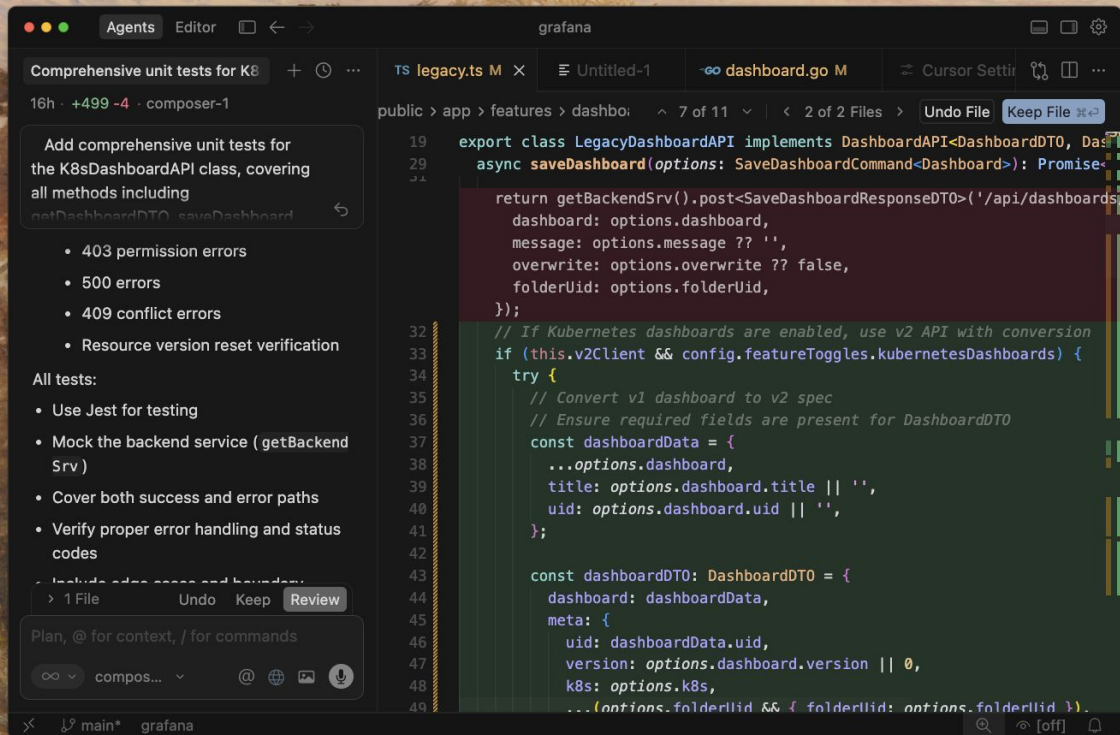
Browser Mode

Cloud Agents

Custom Skills & Subagents

# (A few!) Ways of Working

## Code-Centric



The screenshot shows a code editor window with a commit message on the left and code on the right. The commit message is:

```
Comprehensive unit tests for K8sDashboardAPI class, covering all methods including netDashboardDTO.saveDashboard
```

- 403 permission errors
- 500 errors
- 409 conflict errors
- Resource version reset verification

All tests:

- Use Jest for testing
- Mock the backend service (getBackendSrv)
- Cover both success and error paths
- Verify proper error handling and status codes

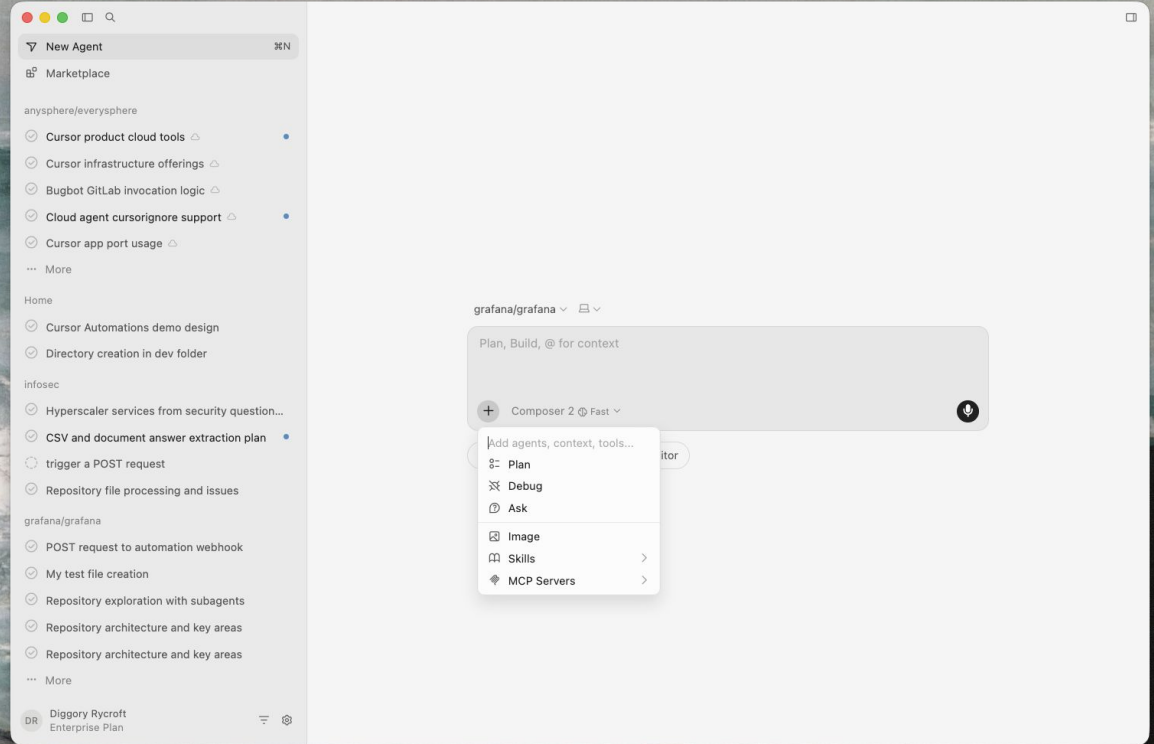
The code on the right is in TypeScript and shows a class `LegacyDashboardAPI` implementing `DashboardAPI`. The `saveDashboard` method is highlighted in green. The code is as follows:

```
19 export class LegacyDashboardAPI implements DashboardAPI<DashboardDTO, DashboardCommand> {
20   async saveDashboard(options: SaveDashboardCommand<Dashboard>): Promise<DashboardDTO> {
21     return getBackendSrv().post<SaveDashboardResponseDTO>('/api/dashboards', {
22       dashboard: options.dashboard,
23       message: options.message ?? '',
24       overwrite: options.overwrite ?? false,
25       folderUid: options.folderUid,
26     });
27   }
28 }
29
30 // If Kubernetes dashboards are enabled, use v2 API with conversion
31 if (this.v2Client && config.featureToggles.kubernetesDashboards) {
32   try {
33     // Convert v1 dashboard to v2 spec
34     // Ensure required fields are present for DashboardDTO
35     const dashboardData = {
36       ...options.dashboard,
37       title: options.dashboard.title || '',
38       uid: options.dashboard.uid || '',
39     };
40   } catch {
41     // ...
42   }
43   const dashboardDTO: DashboardDTO = {
44     dashboard: dashboardData,
45     meta: {
46       uid: dashboardData.uid,
47       version: options.dashboard.version || 0,
48       k8s: options.k8s,
49       ...(options.folderId && { folderId: options.folderId })
50     }
51   };
52   return getBackendSrv().post<SaveDashboardResponseDTO>('/api/dashboards/v2', dashboardDTO);
53 }
54 }
```

# (A few!) Ways of Working

Code-Centric

**Agent-First**

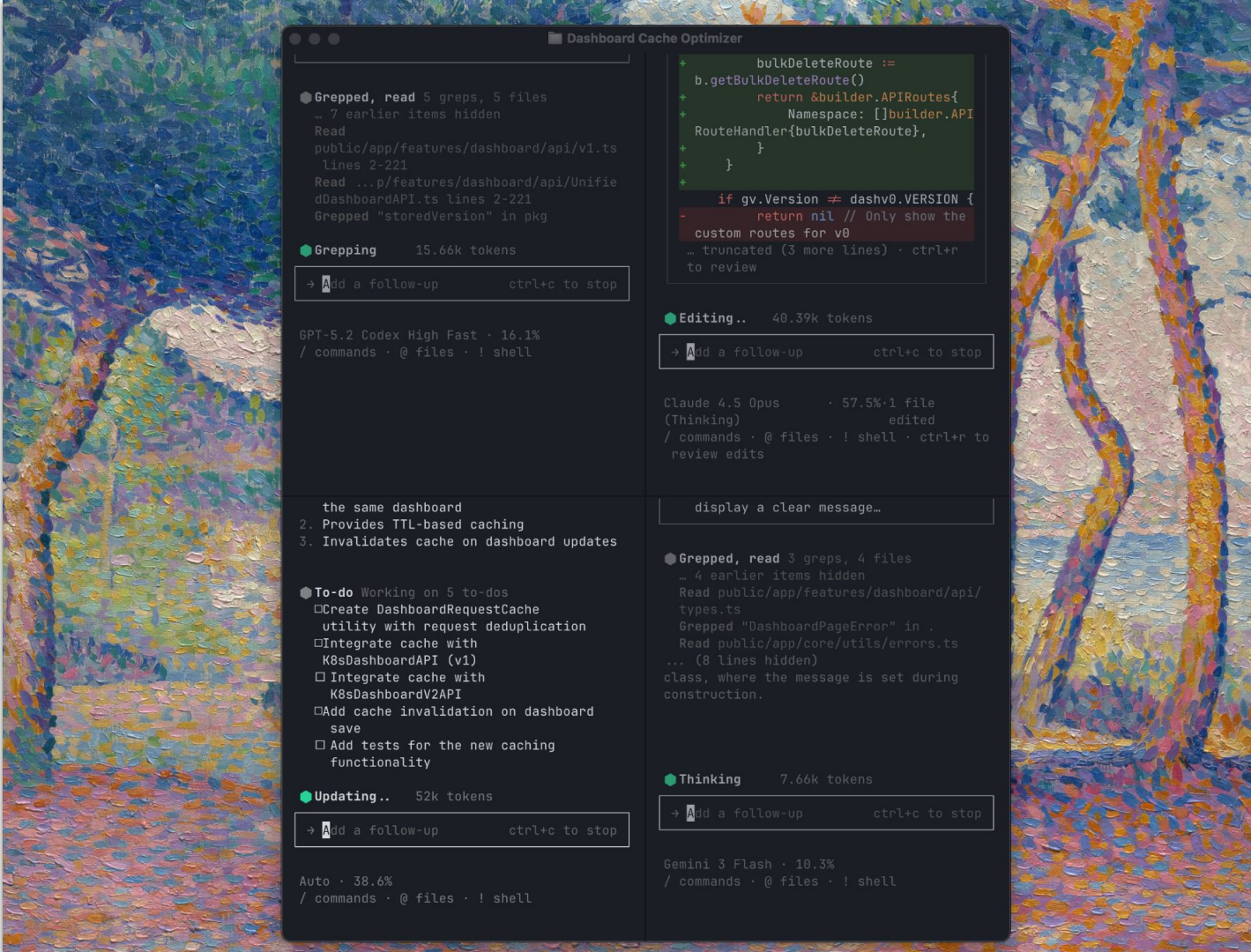


# (A few!) Ways of Working

Code-Centric

Agent-First

Multi-Agent  
Terminals



```
● Grep'ed, read 5 greps, 5 files
... 7 earlier items hidden
Read
public/app/features/dashboard/api/v1.ts
lines 2-221
Read ...p/features/dashboard/api/Unifie
dDashboardAPI.ts lines 2-221
Grep'ed "storedVersion" in pkg
```

● Grep'ing 15.66k tokens

→ Add a follow-up ctrl+c to stop

GPT-5.2 Codex High Fast · 16.1%  
/ commands · @ files · ! shell

- the same dashboard
- 2. Provides TTL-based caching
- 3. Invalidates cache on dashboard updates

- To-do Working on 5 to-dos
  - Create DashboardRequestCache utility with request deduplication
  - Integrate cache with K8sDashboardAPI (v1)
  - Integrate cache with K8sDashboardV2API
  - Add cache invalidation on dashboard save
  - Add tests for the new caching functionality

● Updating.. 52k tokens

→ Add a follow-up ctrl+c to stop

Auto · 38.6%  
/ commands · @ files · ! shell

```
+ bulkDeleteRoute :=
+ b.getBulkDeleteRoute()
+   return &builder.APIRoutes{
+     Namespace: []builder.API
+     RouteHandler{bulkDeleteRoute},
+   }
+ }
```

```
- if gv.Version ≠ dashv0.VERSION {
+   return nil // Only show the
+   custom routes for v0
+ ... truncated (3 more lines) · ctrl+r
+ to review
```

● Editing.. 40.39k tokens

→ Add a follow-up ctrl+c to stop

Claude 4.5 Opus · 57.5%·1 file  
(Thinking) edited  
/ commands · @ files · ! shell · ctrl+r to  
review edits

display a clear message...

```
● Grep'ed, read 3 greps, 4 files
... 4 earlier items hidden
Read public/app/features/dashboard/api/
types.ts
Grep'ed "DashboardPageError" in .
Read public/app/core/utls/errors.ts
... (8 lines hidden)
class, where the message is set during
construction.
```

● Thinking 7.66k tokens

→ Add a follow-up ctrl+c to stop

Gemini 3 Flash · 10.3%  
/ commands · @ files · ! shell

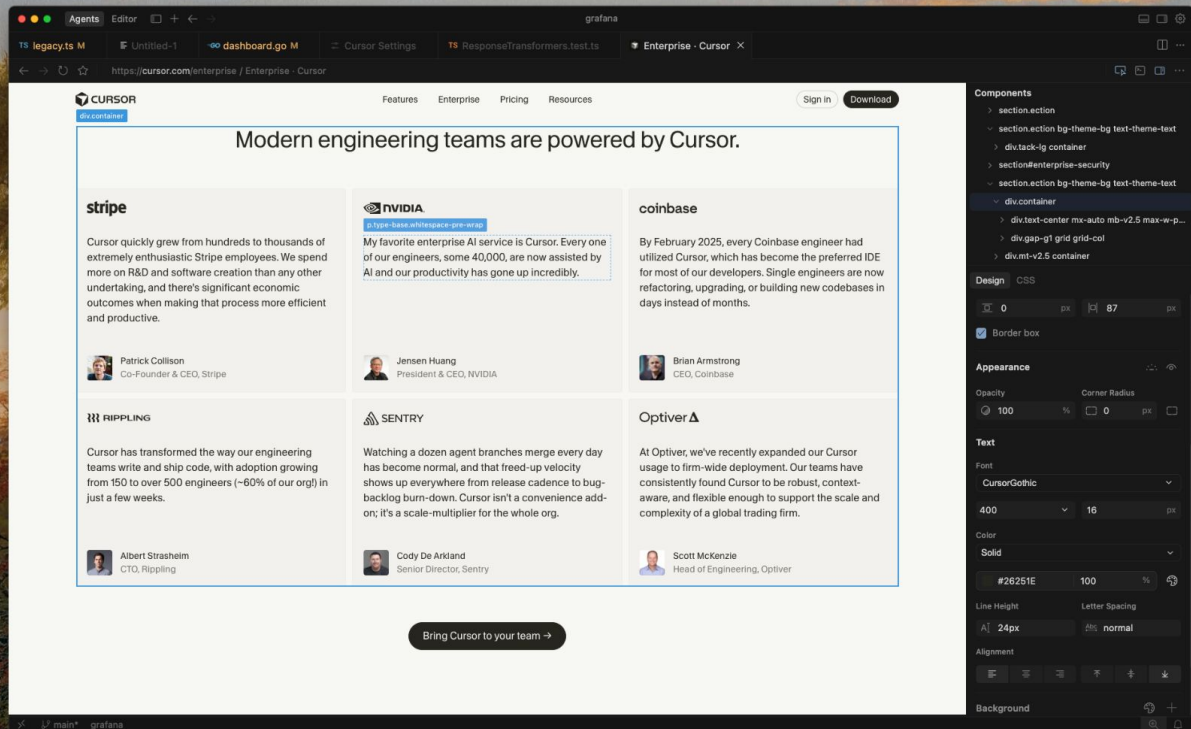
# (A few!) Ways of Working

Code-Centric

Agent-First

Multi-Agent  
Terminals

Frontend Design  
& Testing



The screenshot displays the Cursor website's 'Enterprise' page. The main heading reads 'Modern engineering teams are powered by Cursor.' Below this, there are six testimonials arranged in a 2x3 grid, each featuring a company logo, a quote, and a photo of a Cursor user. The companies featured are Stripe, NVIDIA, Coinbase, Rippling, Sentry, and Optiver. A 'Bring Cursor to your team' button is located at the bottom center of the testimonial grid. On the right side of the browser window, a 'Components' panel is visible, showing a tree view of UI components and a 'Design' panel with CSS and appearance settings.

**Cursor** Features Enterprise Pricing Resources Sign In Download

## Modern engineering teams are powered by Cursor.

**stripe**  
Cursor quickly grew from hundreds to thousands of extremely enthusiastic Stripe employees. We spend more on R&D and software creation than any other undertaking, and there's significant economic outcomes when making that process more efficient and productive.  
Patrick Collison  
Co-Founder & CEO, Stripe

**NVIDIA**  
My favorite enterprise AI service is Cursor. Every one of our engineers, some 40,000, are now assisted by AI and our productivity has gone up incredibly.  
Jensen Huang  
President & CEO, NVIDIA

**coinbase**  
By February 2025, every Coinbase engineer had utilized Cursor, which has become the preferred IDE for most of our developers. Single engineers are now refactoring, upgrading, or building new codebases in days instead of months.  
Brian Armstrong  
CEO, Coinbase

**RIPLLING**  
Cursor has transformed the way our engineering teams write and ship code, with adoption growing from 150 to over 500 engineers (~60% of our org!) in just a few weeks.  
Albert Strasheim  
CTO, Rippling

**SENTRY**  
Watching a dozen agent branches merge every day has become normal, and that freed-up velocity shows up everywhere from release cadence to bug-backlog burn-down. Cursor isn't a convenience add-on; it's a scale-multiplier for the whole org.  
Cody De Arkland  
Senior Director, Sentry

**Optiver**  
At Optiver, we've recently expanded our Cursor usage to firm-wide deployment. Our teams have consistently found Cursor to be robust, context-aware, and flexible enough to support the scale and complexity of a global trading firm.  
Scott McKenzie  
Head of Engineering, Optiver

Bring Cursor to your team →

**Components**

- section.action
- section.action bg-theme-bg text-theme-text
- div.task-lg container
- section.enterprise-security
- section.action bg-theme-bg text-theme-text
- div.container
- div.text-center max-auto mb-v2.5 max-w-p...
- div.gap-g1 grid grid-col
- div.mt-v2.5 container

**Design** CSS

Border box

**Appearance**

Opacity 100 % Corner Radius 0 px

**Text**

Font CursorGothic

400 16 px

Color Solid #26251E 100 %

Line Height Letter Spacing A 24px 2px normal

Alignment

Background

# (A few!) Ways of Working

Code-Centric

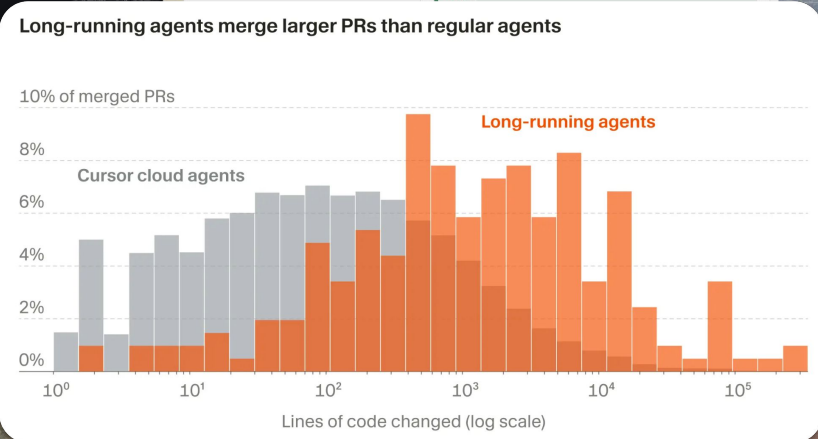
Agent-First

Multi-Agent  
Terminals

Frontend Design  
& Testing

Long-Running  
Async Agents

The screenshot shows a development environment setup interface for Grafana. It includes a sidebar with navigation options like 'New Agent', 'Automations', and 'Dashboard'. The main content area has a 'Walkthrough' section with a video player showing a 'Hello World Dashboard' in Grafana. Below the video, there are instructions to 'Save this environment so new agents start' and 'Add follow up for setup agent'. On the right, there's a diff view for 'AGENTS.md' showing code changes for build tags, config defaults, and database migrations.



# (A few!) Ways of Working

Code-Centric

Agent-First

Multi-Agent  
Terminals

Frontend Design  
& Testing

Long-Running  
Async Agents

Automations  
across the SDLC

Automations > Triage linear tickets Create Agent

### Triage linear tickets

Triggers

Issue Created Code Quality any project

in everyisphere on main

+ Add Trigger

Tools

- Open Pull Request
- Linear Login as Service Account All team automations will share this credential
- Datadog Login as Service Account All team automations will share this credential
- + Add Tool or MCP

GitHub GitLab slack Linear PagerDuty

1Password snyk Custom Webhooks Custom Crons

ryokun6 7 minutes ago

just one more thing!

ryokun6 committed

All checks have passed  
3 successful

- Cursor / Integration
- Cursor / Smoke Tests
- Cursor Bugbot

Merge pull request

Agents Automations Dashboard DR

Ask Cursor to build, fix bugs, explore

Codex 5.3 High Fast

fieldsphere/edu - main - Set up environment

Use private worker

Q

Yesterday

- 5000 token example page +182 edu 16h
- Document review edu Codex 5.3 High Fast 18h
- Opus frontier model references +26 -26 benchmarking-app 19h
- GPT-5.3-Codex cache costs everyisphere Composer 1 • 22h

This Week

- Cookbook catalog hook +40 cookbook • 2d

# (A few!) Ways of Working

Code-Centric

Agent-First

Multi-Agent  
Terminals

Frontend Design  
& Testing

Long-Running  
Async Agents

Automations  
across the SDLC

Inside your  
IDEs of choice

 CURSOR +  JETBRAINS



**IntelliJ IDEA**

IDE for Java and Kotlin  
developers

Java Kotlin Spring



**PyCharm**

IDE for Python developers and  
data scientists

Python Django Jupyter



**PhpStorm**

IDE for PHP developers

PHP Laravel Symfony



**GoLand**

IDE for Go developers

Go (Golang) JavaScript TypeScript



**Rider**

IDE for .NET and game  
developers

C# .NET ASP.NET



**CLion**

IDE for C and C++ developers

C C++ CMake



**RustRover**

IDE for Rust developers

Rust SQL JavaScript



**WebStorm**

IDE for JavaScript and  
TypeScript developers

JavaScript TypeScript React



**RubyMine**

IDE for Ruby and Rails  
developers

Ruby on Rails (RoR) Hotwire RuboCop



**DataGrip**

Tool for multiple databases

Databases SQL NoSQL



**DataSpell**

Your AI-powered tool for  
analyzing data and sharing  
insights

Python Jupyter SQL cells



**ReSharper**

Extension that makes Visual  
Studio a much better IDE

C# .NET ASP.NET

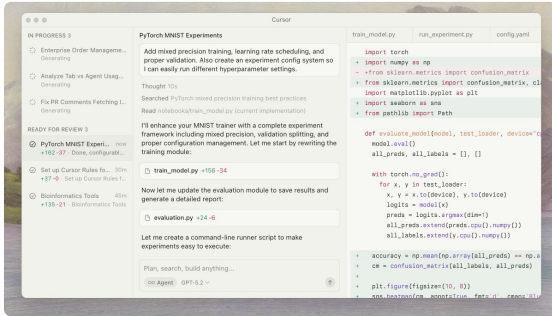
Integrated via ACP with your Cursor subscription. No extra charge.

# Where can I use Cursor?

## Cursor Desktop

The agentic development platform made for professional SWE teams.

Adaptable to every level of agent autonomy and every style of work.



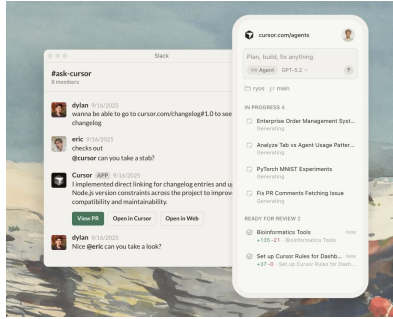
OpenVSX  
Extension  
Registry

[cursor.com/download](https://cursor.com/download)

## Cloud Agent Platform

Web, Mobile, Slack, API

Trigger & manage sync + async agents and long-running tasks from anywhere.



[cursor.com/agents](https://cursor.com/agents)

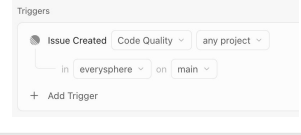
## Bugbot Code Review

GitHub GitLab

Automatic bug identification, and PR autofix, at the org & repo level.

## Automations

Always-on Agents augmenting the entire SDLC.

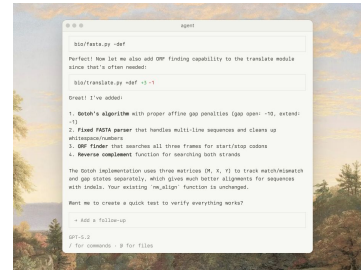


## Cursor CLI

Plug Cursor into any IDE or CI/CD process.



`curl https://cursor.com/install -fss | bash`



## JetBrains IDEs



Available across IntelliJ, PyCharm, and the entirety of the JetBrains family. A frontier AI coding experience for devs in Java, Kotlin, Python, .NET, C#, and more.



# 10-Second Quickstart

Download Cursor Desktop: [cursor.com/download](https://cursor.com/download)

- ✓ VS Code settings, extensions, keybinds can be auto-imported on startup
- ✓ Immediately use any existing third-party Rules, Skills, & Subagents  
(don't worry if this is new to you - we'll teach you about these!)



You can access your Cursor Settings via  $\text{⌘}+\text{Shift}+\text{J}$  /  $\text{Ctrl}+\text{Shift}+\text{J}$   
Or use the Command Palette ( $\text{⌘}+\text{Shift}+\text{P}$ ) and type in what you need.

# Demo: Getting Oriented

The screenshot displays the Cursor IDE interface during a code review session. The main workspace is divided into three panes:

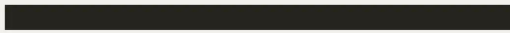
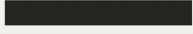
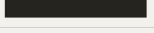
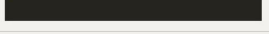

- Left Pane:** A sidebar with a search bar labeled "Search Agents" and a "New Agent" button. Below it, a "Today 4" section lists agents like "Edge proxy caching bug" and "Model selector dropdown polish". A "This week 56" section lists more agents like "Refactor file upload logic" and "Improve search index latency".
- Middle Pane:** A code review window for "Sonnet 4.5" (2 Files, +137-59) with a title "Fix: Introduce copyStrokeProps function to safely copy stroke-related properties". It shows a diff for "icons.ts" and "ModelSwitcher.tsx". The review text discusses the safety of cloning stroke-related props and the introduction of a dedicated helper function. A list of tasks to be completed is shown at the bottom, including adding a dropdown component, embedding a header, and closing popovers.
- Right Pane:** A "Review Sonnet 4.5" window showing the code diff for "icons.ts" and "ModelSwitcher.tsx". The diff highlights changes to the "copyStrokeProps" function and the addition of a tooltip in "ModelSwitcher.tsx".

On the far right, a floating chat window titled "Agent Dashboard" is visible, containing a search bar, a "Composer 1" input field, and a list of recent agent interactions, including "Edge proxy caching bug" and "Model selector dropdown polish".

# 3 Need-to-Know Tips

- Models in Cursor *understand* your codebase, through **Semantic Search** & Cursor's custom embedding model. This makes models smarter, faster, and more conformant with existing patterns - especially on large codebases. Be ambitious!

## All models improve with semantic search

Model (alphabetical)	Relative improvement (Cursor Context Bench)
Composer	23.5% 
Gemini 2.5 Pro	8.7% 
GPT-5	6.5% 
Grok Code	11.9% 
Sonnet 4.5	14.7% 

## Large Codebase Exploration Test

Popular Terminal UI / CLI Agents (no indexing, no semsearch)	~2 minutes 20 seconds ✓ All issues identified
Cursor with Semantic Search	~41 seconds ( <b>3.4x faster</b> ) ✓ All issues identified

# 3 Need-to-Know Tips

- Models in Cursor *understand* your codebase, through **Semantic Search** & Cursor's custom embedding model. This makes models smarter, faster, and more conformant with existing patterns - especially on large codebases. Be ambitious!
- **Plan Mode** is a superpower for more complex tasks.
  - 💡 Press shift+tab in chat to quickly transition into Plan Mode.
- Build Skills and Subagents progressively, not all upfront.
  - 💡 Use /create-skill or /create-subagent to interactively adapt work into reusable agent abilities, for you and your team.

# Model Choice 101




i Cursor is a model-neutral platform, with best-in-class harnesses for every frontier model.

With some basic familiarity with Model Selection, you can use the best model for each task, and make efficient use of tokens.



# Top Frontier Models

The most intelligent & capable frontier models.  
Excellent for long-horizon tasks and asynchronous work, and suitable for  
Cursor's specialized [\*\*\*Long-Running Agent Harness\*\*\*](#).

Model	Spend Efficiency (input tokens per \$100)	Terminal- Bench 2.0 Score
 <a href="#">GPT-5.5</a> OpenAI	20 million	82.7%
 <a href="#">Claude 4.7 Opus</a> Anthropic	20 million	69.4%
 <a href="#">Composer-2</a> Cursor	200 million	61.7%

# Multi-Model Agent Flows

## Research & Plan

Ask Mode  
Plan Mode

“Add 429 handling to all APIs and build clear frontend user feedback for backoff/retry.”

Main agent spins off parallel Subagents with clear context windows and focused goals

**Subagent 1**  
Investigate current patterns for API error handling

**Subagent 2**  
Identify surfaces that require 429 handling

**Subagent 3**  
Identify surfaces for 429 handling across HTTP APIs, frontend data fetching, and plugin proxies.

## Execute

Agent Mode

High-reasoning main agent

Work is orchestrated across multiple Subagents, Assigning the best models for each task

The most capable models are now able to invoke *other* models as “Subagents”. Cursor lets your main agent choose from a full range of fast, intelligent, and capable models.

### High-Reasoning Frontier Models

The smartest, most capable models. Great for planning, deep codebase exploration, and coordinating Subagents.

GPT-5.5 Opus 4.7

Composer 2

### Fast, Cost-Effective, Focused Models

Gemini 3 Flash

Composer 2 (Dual-Purpose)

# Token Efficiency Pointers

Rest assured, Cursor handles all *major* measures automatically.

- ✓ Caching Prompts & Context  
(Most “token usage” on your usage report just Cache Reads: ~90% more cost-effective than direct inputs)
- ✓ Dynamic Context Discovery  
(reduces total agent tokens by ~46.9%)
- ✓ Semantic Search  
(improves agent accuracy and speed in navigating large codebases)

As a user, consider these concepts for even more efficiency:

**Choose the right model for the job:**

Model prices and relative strengths can vary widely! Take a moment to understand relative pricing, or use Auto to keep it simple.

**Use Plan Mode before Executing:**

Align on an approach upfront, so that every token goes toward desirable outcomes.

Thank You

