Cursor Composer User Guide

A complete guide on how to use Cursor Composer – Cursor IDE's most powerful AI multi-file editing feature in 2026.

Cursor Composer is the core feature of Cursor IDE – a code editor based on VS Code with deep AI integration . Launched and continuously upgraded, Composer allows you **to simply describe code in natural language** , and the AI ??will automatically write, edit, and refactor code across **multiple files simultaneously** throughout the entire project.

Unlike Chat, which only answers questions, or Inline Edit, which only modifies a single file, Composer acts like a true 'AI software engineer': it understands the entire codebase, creates new files, updates imports, runs the terminal, and suggests changes in clear diffs for you to review before applying them. If you're new to Cursor, learn about Composer through the guide below.

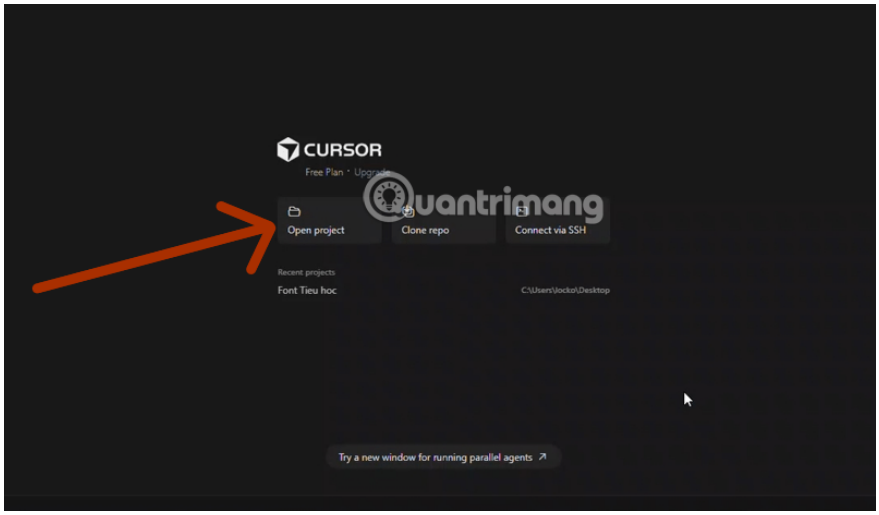
Instructions for using Cursor Composer

Step 1: Open the Cursor application on your desktop.

Download and install the latest version of Cursor at cursor.com (supports Windows , macOS , and Linux). Open the application as usual. The main interface is similar to VS Code but with an added AI bar on the side.

Step 2: Open the project/folder you are currently working on.

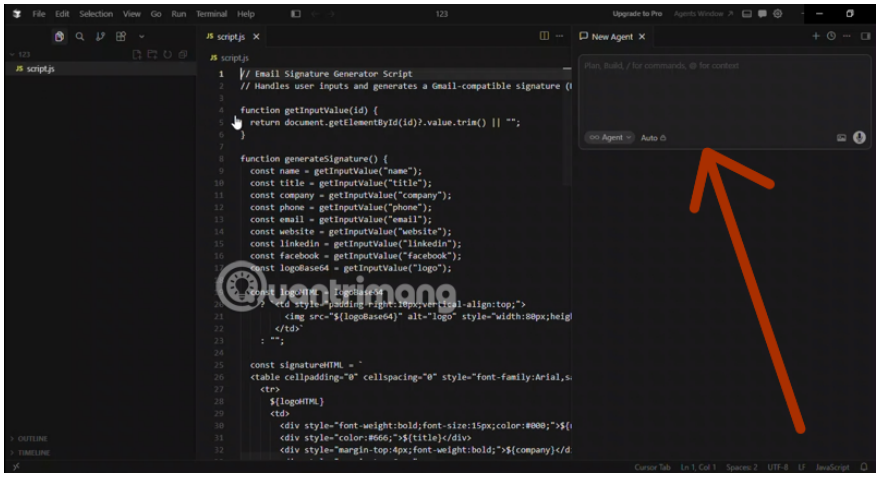
Select "Open Project" in the middle of the main screen to open the project. Cursor will automatically index the codebase so Composer understands the entire project structure.



Select the folder or file you want to upload to Cursor. When you select a folder containing files, it will index all code files, subfolders, package.json, .git, .cursorrules, etc., so that Composer, Chat, and Agent Mode understand the entire codebase.

Step 3: Open Composer (New Agent) using the shortcut Press **Cmd + I** (on Mac) or **Ctrl + I** (on Windows/Linux).

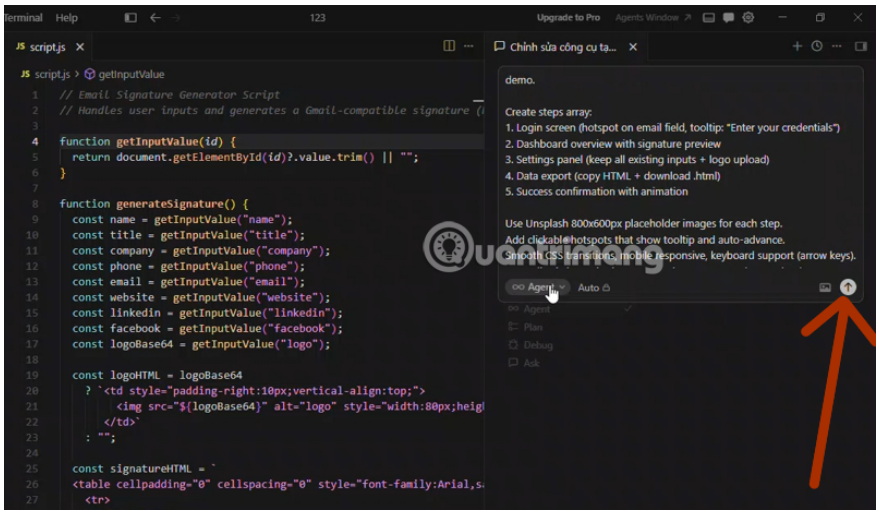
Composer will open as a floating window or side panel (depending on your settings). This is the main interface where you enter multi-file commands.



Step 4: Enter your coding requirements into the input field.

Type a detailed description in English or Vietnamese (Cursor 2026 supports both).

For example: 'Add Google OAuth login functionality to your Next.js application: create a login page, authentication middleware, update the navbar, protect the dashboard route, and add a sample environment file.' You can @mention specific files or let Composer find them automatically. After entering your requirements, select Enter or the Submit icon as shown in the image below.



Step 5: Review the changes in the diff view and apply them.

Composer will analyze, plan, and display a detailed **diff view** for each affected file (green = add, red = delete).

You can:

1. 1. Preview each file.
   2. Accept/Reject each change individually.
   3. Press **Accept All** (Cmd + Enter) to apply everything. In **Agent** Mode (turn on the toggle in Composer), the AI ??can automatically run the terminal, install packages, and iterate without you having to go through each step.

After applying, you just need to test the code and continue working.

The outstanding advantages of Cursor Composer compared to other AI coding tools.

Composer not only saves time but will also completely change the way we program in 2026. A single prompt can accomplish complex functionality that previously took hours or even days. It understands dependencies between files, avoids import errors, and keeps code clean according to best practices if you provide guidance via the "cursorrules" file.

Compared to GitHub Copilot or Claude Projects, Composer is stronger in its multi-file agentic and **parallel editing** capabilities (it can run multiple agents simultaneously). Developers report a 3-5x increase in productivity when building large features or refactoring old codebases.

Tips for using Composer 2.0 most effectively in 2026

1. **Write a clear, structured prompt** : Start with the goal, then list the specific requirements in bullet points. Add 'following Next.js 15 best practices' or 'use TypeScript strict mode'.
2. **Use .cursorrules** : Create this file in the project root to define permanent coding rules (style, framework, naming convention).
3. **Combine Agent Mode** : For complex tasks, enable Agent Mode so that Composer automatically runs commands, tests, and fixes errors without your intervention.
4. **Commit Git regularly** : Before running a large Composer run, commit the current code to make it easier to roll back if needed.
5. **Choose the right model** : Composer 2.0 is the most powerful for multi-file tasks; Claude 4.6 or GPT-5 are suitable for tasks requiring high creativity.
6. **Start small** : With a new project, try the prompt 'Create skeleton Next.js + Tailwind + TypeScript according to standard folder structure' before working on larger features.

Important notes when using

1. Composer works best when the project is already open and indexed (this can take a few seconds with large projects).
2. Token limits still exist, but Composer 2.0 handles context very intelligently thanks to self-summarization.
3. Always carefully review differences before accepting all requests, especially with production codebases.

Cursor Composer is currently the most powerful AI coding tool on the market in 2026, helping programmers focus on ideas instead of typing every line of code. With just the 5 simple steps above, you can start building complex features in minutes.