

What is Plan Mode in Cursor? How to use Plan Mode in Cursor?

This guide provides detailed instructions on how to use Plan Mode in Cursor 2026 to program intelligently, safely, and optimize AI workflows.

In the era of AI agents, allowing artificial intelligence to directly interfere with source code can sometimes be risky if not carefully controlled. **Plan Mode** in Cursor is the "think before you act" solution that every modern programmer needs to master.

This article will guide you from A to Z on how to use Plan Mode, and also update you on the latest features of Cursor as of May 2026.

1. What is Plan Mode in Cursor?

Unlike conventional AI tools that only provide fragmented code, **Plan Mode** allows AI to analyze the entire project and build a step-by-step plan before making any changes. This minimizes errors, saves tokens, and ensures consistent logic across files in the system.

As of **Cursor** version 3.1 , released in April 2026, Plan Mode has been deeply integrated into the **Agents Window** system , enabling multitasking and handling of complex repositories (multi-root workspaces).

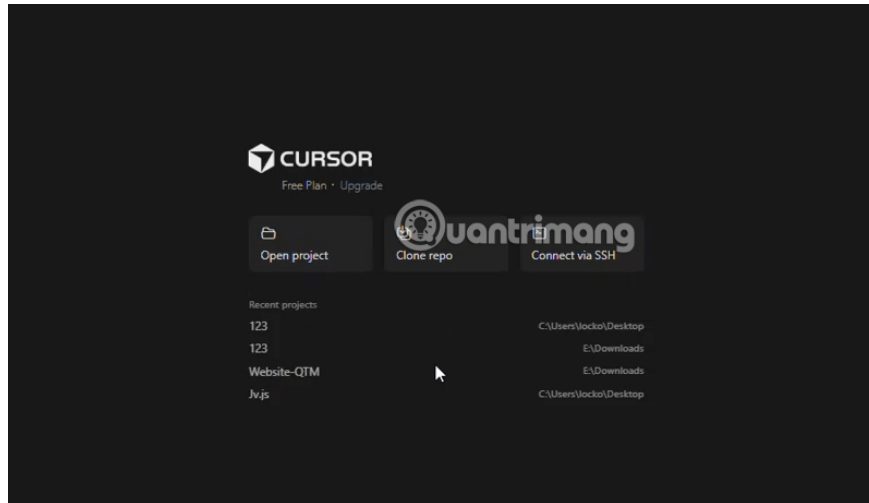
2. Detailed instructions on how to use Plan Mode in Cursor

To get started with Plan Mode, you need to follow these standardized steps (verified with the latest version):

Step 1: Start and Log In

Open the Cursor editor and make sure you are logged into your account (especially the Pro or Ultra plan to use the most powerful language models like Claude 3.5 Sonnet or GPT-5 High MAX).

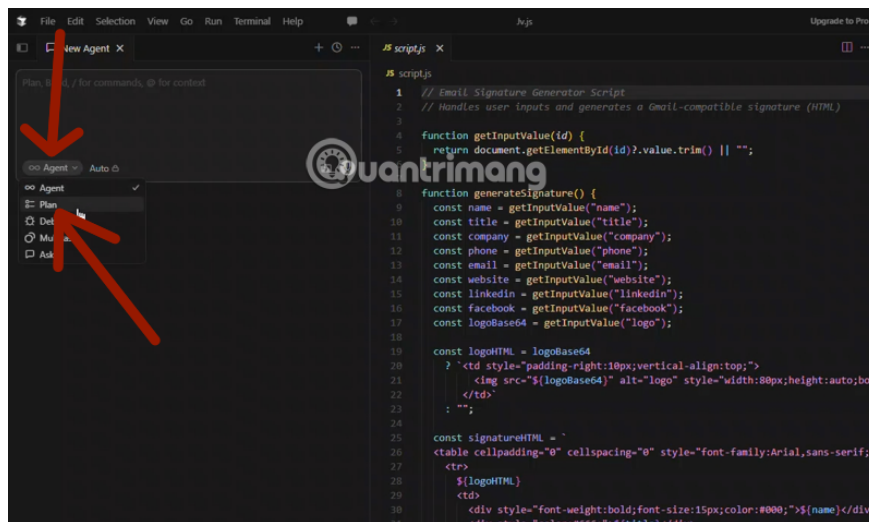
Go to the File menu at the top and select **Open Folder** to open the project folder you want to work with. If you already have a project, simply select it from the main screen.



When you open a folder, press **Ctrl + I** to open the Cursor command input dialog box if you don't see it.

Step 2: Turn on Plan Mode

In the chat interface, you'll see the Plan Mode option . Click on it to activate planning mode. Enter the task or feature you want to deploy in the chat box. For example: *"Add a user authentication system using JWT and storage to Redis"* .



Step 3: Review the plan

Cursor won't write code immediately. Instead, it lists the steps it intends to take: which files to modify, which libraries to add, and what the logic will be. Take the time to carefully review these steps.

```

1 // Email Signature Generator Script
2 // Handles user inputs and generates a Gmail-compatible signature
3
4 function getInputValue(id) {
5   return document.getElementById(id)?.value.trim() || "";
6 }
7
8 function generateSignature() {
9   const name = getInputValue("name");
10  const title = getInputValue("title");
11  const company = getInputValue("company");
12  const phone = getInputValue("phone");
13  const email = getInputValue("email");
14  const website = getInputValue("website");
15  const linkedIn = getInputValue("linkedIn");
16  const facebook = getInputValue("facebook");
17  const logoBase64 = getInputValue("logo");
18
19  const logoHTML = logoBase64
20    ? `<td style="padding-right:10px;vertical-align:top;"`
21      ``
23      : "";
24
25  const signatureHTML = `
26  <table cellpadding="0" cellspacing="0" style="font-family:Arial,
27  <tr>

```

Step 4: Build

Once you're satisfied with the plan, click the **"Build"** button . Cursor will then automatically and safely apply the changes to your project.

Module script.js hiện tại quá dài. Hãy lập kế hoạch tách nó thành các file nhỏ hơn theo quy tắc Single Responsibility. Đừng viết code ngay, hãy cho tôi xem sơ đồ cấu trúc mới

input, dùng HTML, cập nhật UI, copy clipboard, upload logo, và wiring sự kiện). Mình sẽ tạo kế hoạch tách thành các module riêng kèm sơ đồ luồng/module để bạn duyệt trước khi triển khai.

tach-script-srp_59eea336.plan.md

tach-script-srp

Tách 'script.js' thành các module nhỏ theo Single Responsibility để dễ bảo trì, test và mở rộng. Giữ nguyên hành vi hiện tại của trình tạo email signature nhưng có lập rõ phần DOM, business logic, rendering và event wiring.

5 To-dos

- Xác nhận cấu trúc thư mục và tên module mới theo SRP.
- Tách phần đọc dữ liệu form và dùng HTML sang các module độc lập.
- Tách clipboard và image base64 thành service riêng.
- ... 2 more

View Plan Build Ctrl+B

Plans > tach-script-srp_59eea336.plan.md Auth: Build Ctrl+B

Kế hoạch tách 'script.js' theo Single Responsibility

Mục tiêu

- Giảm độ phức tạp của module hiện tại bằng cách tách theo trách nhiệm rõ ràng.
- Duy trì nguyên hành vi: generate preview/output, copy signature, upload logo base64.
- Tạo cấu trúc dễ mở rộng (thêm mạng xã hội, đổi template, thêm validation) mà không sửa nhiều file.

Hiện trạng trách nhiệm đang trộn trong một file

- Đọc dữ liệu form (getInputValue + lấy nhiều field).
- Dùng HTML chữ ký (template string lớn).
- Cập nhật UI (preview, output).
- Tương tác clipboard (copySignature).
- Chuyển ảnh sang base64 (handleLogoUpload).
- Khởi tạo event listeners (DOMContentLoaded).

Cấu trúc file đề xuất

- c:\Users\lock\Desktop\src\js\src\index.js
 - Entry point: bootstrap ứng dụng và đăng ký sự kiện.
- c:\Users\lock\Desktop\src\js\src\dom\selectors.js
 - Tập trung truy cập phần tử DOM theo id (single source of truth cho selectors).

Build Kế hoạch tách 'script.js' theo Single Responsibility

5 of 5 To-dos Completed

Mình sẽ triển khai đúng kế hoạch đã chốt: cập nhật trạng thái to-do theo thứ tự, tạo cấu trúc module mới, tách logic, rồi kiểm tra nhanh để đảm bảo hành vi không đổi. Bắt đầu bằng to-do đầu tiên về layout module.

Started to-do Xác nhận cấu trúc thư mục và tên module mới theo SRP.

Explored 2 searches, 1 command

Mình đã kiểm tra workspace: hiện chỉ có 'script.js'. Giờ mình sẽ tạo cấu trúc 'src/...' và chuyển từng phần logic sang module riêng đúng SRP, sau đó rút gọn 'script.js' thành entry script.

```

js selectors.js +18
export const FIELD_IDS = [
  "name",
  "title",
  "company",

```

```

js signatureForm.js +7
import { FIELD_IDS, getTrimmedInputValue } from "../dom/selecto

```

Plan, Build, / for commands, @ for context

Sơ đồ cấu trúc mới

```

graph TD
  UserAction --> indexBootstrap
  indexBootstrap --> signatureController
  signatureController --> signatureForm
  signatureController --> signatureTemplate
  signatureController --> previewRenderer
  signatureForm --> dumSelectors
  signatureTemplate --> dumSelectors
  previewRenderer --> dumSelectors

```

Luồng xử lý sau khi tách

- index.js khởi tạo và bind events.
- Khi click Generate: controller lấy signatureData từ signatureForm HTML, rồi preview để render.
- Khi click Copy: controller gọi clipboard service để copy từ output.
- Khi upload logo: controller gọi imageToBase64 rồi cập nhật input.

3. Explain the core features of Plan Mode.

Plan Mode is more than just a chat window. Its power lies in three main pillars:

Think before you act (Reasoning first)

1. Instead of jumping straight into writing code and easily falling into "hallucination" (AI illusion), Plan Mode forces the language model to interpret the logic. This is extremely useful for major architectural changes, where a small error could break the entire system.

Supports multiple files and a wide range of contexts.

1. In 2026, Cursor upgraded its contextual understanding capabilities to the maximum level. Plan Mode can scan hundreds of files, understand the relationships between the frontend, backend, and database to provide the most accurate modification path without requiring you to specify each file individually.

Integrating Agents Window and Canvas

1. The latest version introduces **Canvas** – a visual interactive interface. In Plan Mode, you can view the planned logic diagrams or UI in Canvas before the AI ??writes the actual code.

4. Advantages and Disadvantages

Advantage

1. **High accuracy:** Reduces logic errors by up to 80% compared to letting AI automatically write code without a plan.
2. **Complete control:** Programmers act as "Architects," approving the blueprint before the AI ?? "builders" execute it.
3. **Cost savings:** By previewing the plan, you can immediately correct the request if the AI ??misinterprets it, avoiding wasting tokens on unusable code.
4. **Multitasking support:** With Cursor 3, you can run multiple Plan Modes simultaneously for different branches.

Disadvantages

1. **Speed:** The two-step process (Planning -> Build) will be slightly slower than generating code instantly.
2. **Model-dependent:** The effectiveness of Plan Mode largely depends on which model you choose (Claude, GPT, or Gemini).
3. **Requires Review skills:** If you lack the foundational knowledge to review the plan, this feature loses 50% of its value.

5. Benefits for users

Using Plan Mode provides practical value for each specific object:

1. **For Junior Developers:** This is a great learning tool. Watching how AI plans helps you understand problem-solving thinking and standard project structures.

2. **For Senior/Tech Leads:** To help accelerate the deployment of complex features while maintaining tight control over the architecture.
3. **For Startups/Solo Developers:** Allows one person to work with the power of a small team, reducing the burden of managing overlapping changes in code.

6. Latest update on Cursor (May 2026)

If you haven't updated Cursor in the past few weeks, here's what you've been missing:

1. **Cursor 3 & Agents Window:** Agent mode can now automatically run background tasks such as unit tests or security checks while you're viewing the plan.
2. **Async Subordinates:** Plan Mode can now break down a large task into subtasks and process them in parallel, significantly shortening build time.
3. **Design Mode (Cmd + Shift + D):** Integrates direct browser feedback into Plan Mode to precisely fine-tune the UI/UX interface.
4. **Spend Management:** Business administrators can now set "soft" spending limits and receive smart alerts when the team exceeds the allowed limits in Plan Mode.

Conclude

Plan Mode in Cursor is more than just a feature; it's a new programming mindset in the age of AI. By combining the immense computing power of AI with the finesse of human control, we are entering a phase of programming that is safer and more efficient than ever before.

Open Cursor today, turn on Plan Mode, and experience how your project is built in the most systematic way!

You finished reading the article "**What is Plan Mode in Cursor? How to use Plan Mode in Cursor?**" 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.