

歡迎來到大會議室，現場參與課程與討論



Hexo



[從0開始學AI] 第1章 - 詳解 Cursor

BY A-mi



可以先下載練習用專案
<https://github.com/gzes00201/cursor-tutor>





阿咪 (A-mi) 吳品毅

致力於打造幫助人而非取代人的系統

- Hexa Founding Engineer - Scrum Master
- Hexa Founding Engineer - Front-end
- Front-End Developers Taiwan - 前端小聚發起人
- Agile Submit 2024-Agile & Waterfall 都是一家人
- GDG 2023 Google全家餐!快速打造旅宿業最小化MVP
- Agile Taichung 2023 DDD + Event Storming 三個月從無到有打造全新訂房系統
- 前端小聚 -前端監控就是那麼重要
- 前端小聚 -牛刀小試! 面對大團隊的 Angular 開發體驗
- Agile Taichung 2020 敏捷早餐 - 吃出團隊凝聚力

目錄課表

1. [面對AI浪潮，研發人員該如何應對？](#)
2. [序 - 關於 Cursor](#)
3. [Cursor - 方案介紹](#)
4. [Cursor 教學 - TAB](#)
5. [Cursor 教學 - Chat](#)
6. [Cursor 教學 - 舊 K](#)
7. [Cursor 教學 - Rule](#)
8. [Cursor 教學 - MCP](#)
9. [Cursor 教學 - 其他你需要知道的](#)
10. [Cursor 使用 12 Guidelines](#)
11. [如何創造偉大的事物與編程英雄](#)





今日課表

1

Cursor 功能介紹 - 50分鐘

2

休息一下... - 10分鐘

3

設計準則與結語 - 50分鐘

4

自由交流時間 - 10分鐘



Intro

May. 2025



} Hexo

Game

8. 看到中间这截木材, 你认为它属于A还是B呢?

A B

微信号: szqgryt1



Game



2. 把以上三项分组，哪两个分为一组比较合适？

- A. 猴子 + 香蕉
- B. 猴子 + 熊猫
- C. 熊猫 + 香蕉

Q 微信号: szjynd1



Game



1. 你认为这朵花，应该分到A组还是B组？

A

B

赞赏信号:zzqeyn1

This block contains a large line drawing of a sunflower with many petals and a single large leaf at the bottom. Below it is a question in Chinese asking which group the flower belongs to. Two boxes labeled A and B show different arrangements of sunflowers. Box A contains four sunflowers in a 2x2 grid. Box B contains five sunflowers in a 2x2 grid with one additional sunflower placed below the bottom-right flower of the grid.

Game



追根溯源

0

讓我們一起從0開始學AI



AI浪潮

2025



Hexo





Changing the World

1

這是一場災難還是一場革命？



Changing the World

2023 敏捷高峰會分享 (3).key — 已編輯

顯示方式 132% ◀ 播放

表格 圖表 文字 形狀 媒體 註解 分享

AI的興起 人的價值

01 — AI可以取代人嗎？
— AI該工作在面對數位轉型中的實踐與成果

02

03

讓我們來試看看

讓我們來試看看

讓我們來試看看

讓我們來試看看

HIT

AMI

Agile Summit
2023/6/14 台北大稻埕

Hexox



Changing the World

OpenAI □

關於我們

我們的章程

職涯發展

品牌

概述

標誌

畫廊

合作夥伴關係

排版

語言

接觸

家

關於

公司

關於

OpenAI 是一家人工智慧研究和部署公司。我們的使命是確保通用人工智慧造福全人類。



Hexox



Changing the World

HIT

(AI) 不會取代人

(活用AI的人) 可以取代很多人

132%
显示方式 縮放 加入幻燈片
播放
表格 圖表 文字 形狀 媒體 註解

2023敏捷高峰會分享 (3).key — 已編輯

AI的興起
人的價值
01 —
AI可以取代人嗎？
AI 因何工作在銀行軟體開發中的實踐與成果
2
3
4
讓我們來試看者
5
6
7
HIT
(AI) 不會取代人。
(活用AI的人) 可以取代很多人。



動手找看看，你第一個問 AI 的問題是？

The screenshot shows a web browser interface. On the left, there's a sidebar with a yellow location pin icon containing a bed icon. The sidebar lists various search results from 2023, including:

- Angular 自定義Input元件
- 购物网站产品优化计划
- Ingress 多組 IP
- ChatGPT研發初衷
- 預訂飯店單字。
- ComboCraft - 品牌名稱建議
- Scrum Review: Room Inventor
- 愉快的前端小聚
- Ingress Health Check Fix
- 敏捷團隊問題整理
- 敏捷團隊設計溝通
- JS繪世界地圖
- Jira vs YouTrack Comparison
- 潤飾請求及回應
- 整合後台功能整併
- AI轉換Adobe XD至HTML
- 彩虹寶藏設計方案
- 鎖骨後灼熱疼痛
- 新舊系統切換策略
- 用户咨询帮助

On the right, the main content area shows a ChatGPT conversation. The user asks "你好！有什么我可以帮助你的吗？" (Hello! What can I help you with?) and the AI responds: "我是基于 GPT-3.5 架构的 ChatGPT，由 OpenAI 开发和训练。GPT（Generative Pre-trained Transformer）是一种使用自监督学习方法进行预训练的语言模型。它的核心是 Transformer 模型，它由多个自注意力机制组成，用于处理输入序列的上下文信息。这个模型可以理解和生成文本，并且在各种语言任务上表现出色。在我的情况下，我是在大量的文本数据上进行训练的，以便提供广泛的语言理解和生成能力。" The AI also asks "請問吳品毅是誰" (Who is Wu Pin-Yi?). Below this, it provides a brief biography: "吳品毅 (Wu Pin-Yi) 是一位台湾的生物学家和医学教育家。他于1947年出生于台湾，毕业于台湾大学医学院，并在美国加州大学洛杉矶分校 (UCLA) 获得博士学位。" It continues: "此外，吳品毅还在医学领域的研究和教育方面取得了重要的贡献。他专注于生物医学领域的研究，并在癌症、分子生物学和基因工程等领域有深入的研究经验。他曾在美国哈佛大学医学院从事教学和研究工作，并且担任过多个重要的医学和科学组织的职务。"





憤怒五階段

否認
Denial

憤怒
Anger

討價還價
Bargaining

沮喪，反思
Depression

接受，進步
Acceptance



10年前我們再談甚麼？



2000: 網路泡沫高峰、Java EE	2001: Linux 開源、Wi-Fi 802.11b	2002: .NET Framework 1.0	2003: AJAX／Web 2.0	2004: Facebook／Web 2.0	2005: YouTube
2006: AWS EC2 公有雲	2007: iPhone 行動革命	2008: Android／Hadoop 大數據	2009: Bitcoin／Blockchain	2010: HTML5／CSS3	2011: Siri 語音助理
2012: 大數據分析、Hadoop	2013: Docker 容器技術	2014: Kubernetes 編排	2015: TensorFlow 深度學習	2016: VR/AR、AI 聊天機器人	2017: 區塊鏈企業化、ICO
2018: Serverless、Edge Computing	2019: 5G 商用化、MLOps	2020: 遠端協作、居家辦公	2021: GPT-3／生成式 AI	2022: Web3、元宇宙	2023: GPT-4／多模態模型

Changing the World



Hexx



Changing the World

熟練農民每天也僅能插好約 0.1公頃



50倍

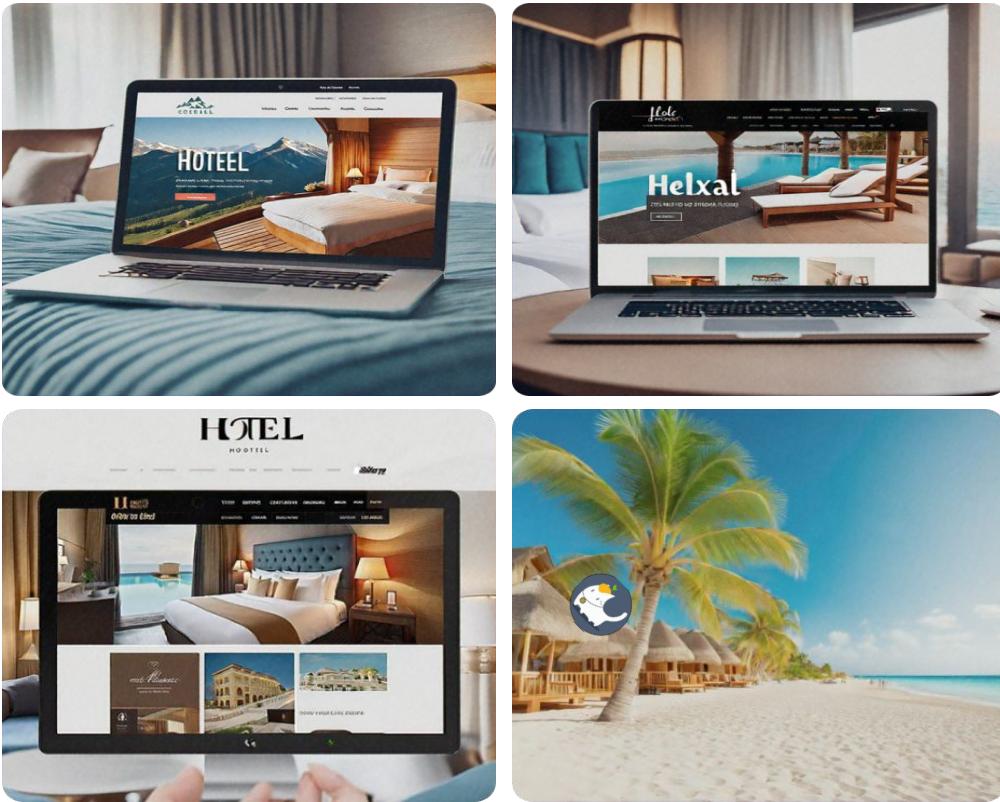
機械插植行數約在 8行, 每日可插5、6公頃





CURSOR RULE #X

學習時，盡量地相信
學會後，盡量地懷疑





讓我們來學 Cursor





讓我們來學使用 AI





讓我們面對進步





從「序」開始



Cursor Power Up



May. 2025



} Hexo^α

2022 / 6

MIT 麻省理工學院的學生

Aman Sanger 阿曼·桑格 ML

Arvid Lunnemark 阿爾維德·倫內馬克 PD

Sualeh Asif 蘇阿萊赫·阿西夫 INFER

Michael Truell 麥可·特魯爾 CEO

創辦了一家名為 Anysphere 的公司





Anysphere 發展與融資里程碑

- **2022/06**: Anysphere 公司成立
- **2023/03/14**: 花三個月推出 Cursor 第一版(代號 Reelas)
- **2023/09**: 完成 種子輪融資，獲得來自 a16z、Thrive Capital、OpenAI 等投資共 800 萬美元 (約新台幣 2.6 億元)
- **2024/08**: 完成 A 輪融資，募得 6,000 萬美元 (約新台幣 18 億元)，公司估值達 26 億美元 (約新台幣 845 億元)
- **2025 年初**: 完成 B 輪融資，募得 1.05 億美元 (約新台幣 30 億元)，公司估值攀升至 100 億美元 (約新台幣 3,250 億元)
- **2025 年**: 年訂閱營收突破 1 億美元 (約新台幣 30 億元) 用戶數突破100萬
- **最新消息顯示**，Anysphere 又完成了一輪 9 億美元的融資





Michael Truell 麥可·特魯爾 CEO

高中時期(紐約 Horace Mann School 賀拉斯曼中學), 他與朋友共同創辦了程式設計競賽平台 Halite, 並因此獲得了 **ACM/CSTA Cutler-Bell 獎**, 這是年輕程式設計師的重要獎項之一。18 歲時, Michael 進入麻省理工學院(MIT)主修電腦科學與數學。在學期間, 他曾在 **Google、Two Sigma、Octant 等公司實習** 。



Aman Sanger 阿曼·桑格 ML

高中時期(紐約 Horace Mann School 賀拉斯曼中學), 獲得 Horace Mann **男子年度運動員獎** 和**美國化學學會學術成就獎**在學期間, 曾於**Google、Bridgewater Associates、You.com** 等公司擔任軟體工程實習生參與自然語言處理(NLP)和計算生物學研究, 並曾在**MIT CSAIL** 的 **Regina Barzilay 實驗室**工作



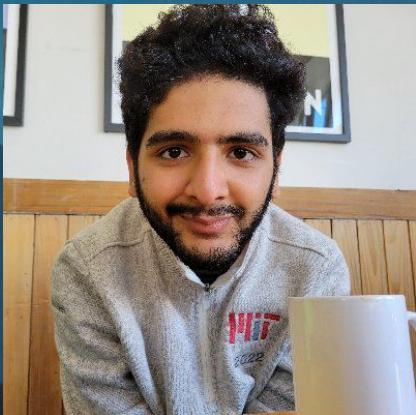
瑞典 Arvid Lunnemark 阿爾維德·倫內馬克 PD

在學期間，Arvid 曾在多家知名公司擔任軟體工程師和量化交易員，包括：

Stripe: 參與世界級支付處理解決方案的開發

QuantCo: 應用量化專業知識優化交易策略。

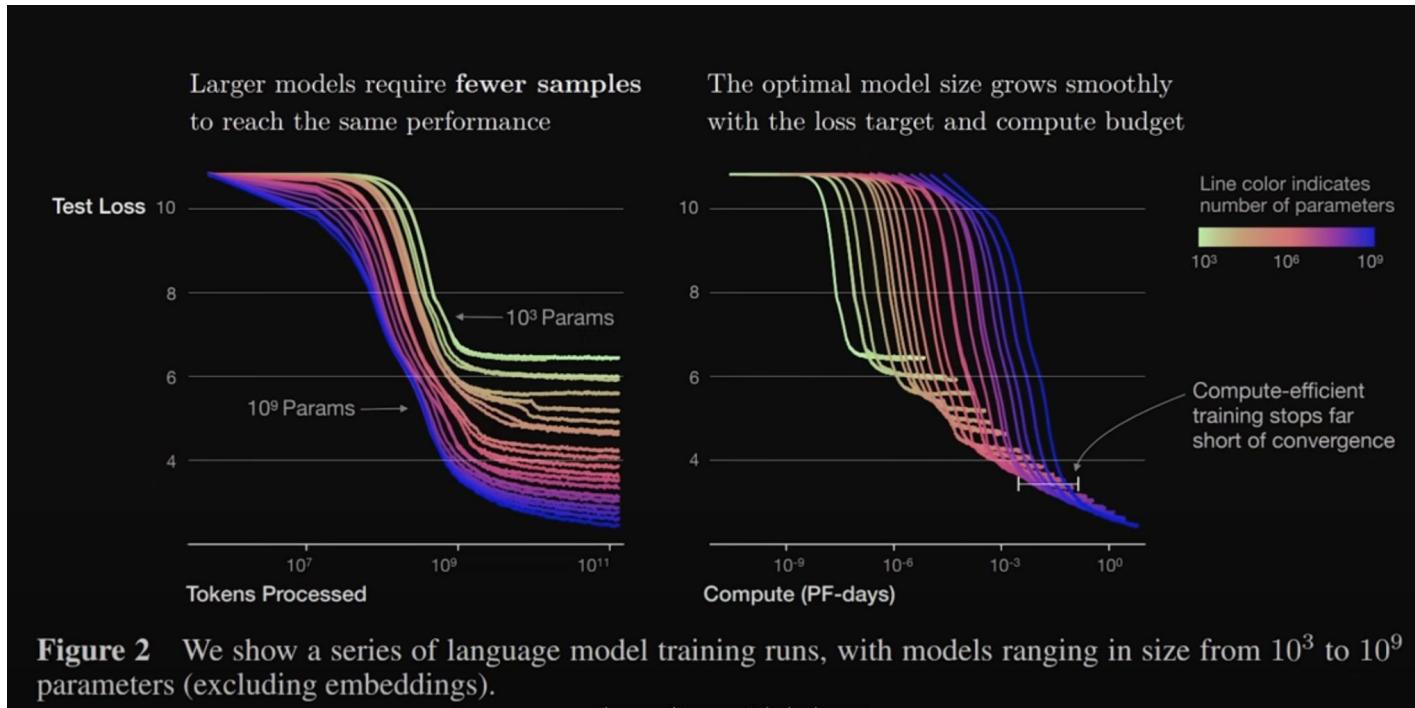
Jane Street: 作為量化交易員，運用強大的分析和解決問題能力，在快節奏的金融市場中取得成功。



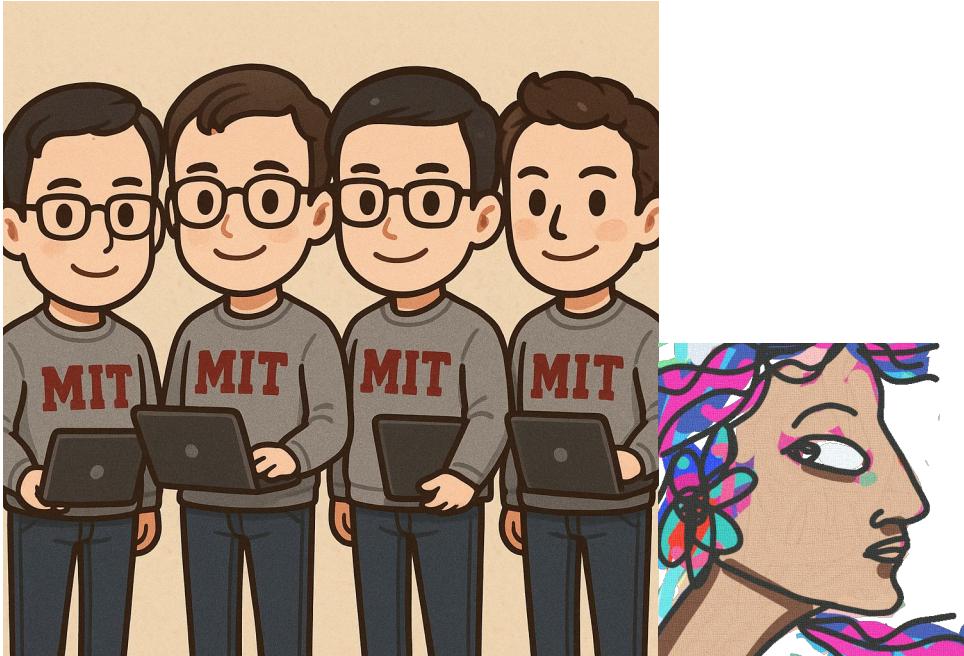
巴基斯坦 Sualeh Asif 蘇阿萊赫·阿西夫 INFER

在學期間，曾代表巴基斯坦參加2016 至 2018 年的國際數學奧林匹亞競賽(IMO)，曾在 IBM Watson ML 部門從事翻譯相關的工作曾參與開發 metaphor，一個端到端的 LLM 驅動搜尋引擎

2021底OpenAI 發表 Scaling Law



2022 Anysphere



一開始，我們選擇了一個競爭者少的領域...金融業分析筆記本，機械工程自動化圖像辨識,私密對話，然後我們發現我們對這些沒有興趣，我們不願意為此奉獻一生

四位理工直男 共同創作



回歸程式開發領域，做一款編輯器是一件很自然的決定

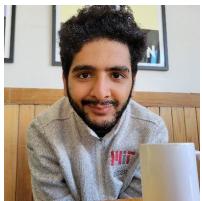
Cursor想創造出什麼未來？



Aman Sanger 阿曼·桑格:

在現在的世界上，領先個半年或一年，就已經是巨大的提升，
AI模型的上限不斷快速的提升
人的極限也不斷在提升
但寫程式的方式沒有改變，沒有讓人非常驚艷的東西

Cursor想創造出什麼未來？



Sualeh Asif 蘇阿萊赫·阿西夫：

我們想做的是開發出一個真正幫助開發人員的模型，這是我們的夢想，就像你有一個助手一樣，幫助你做零熵或低熵操作

Cursor想創造出什麼未來？



Michael Truell 麥可·特魯爾:

跟2020,2021年比起來，現在的Programming非常有趣，人們可以更快速地實踐想要的功能與想法，可以更有掌控的瞭解你的產品，人們更加追求創意，不會那麼死板與小心翼翼，

Cursor想創造出什麼未來？



Arvid Lunnemark 阿爾維德·倫內馬克

體驗幫助研發人員更好的工作，這實在是太有趣了，程式開發者重要的價值是在快速迭代中釐清思路，我們總是在看到東西後才更明白目標在哪裡，不是嗎？所以能夠快速產生初始本版的工具是我們認為最重要的



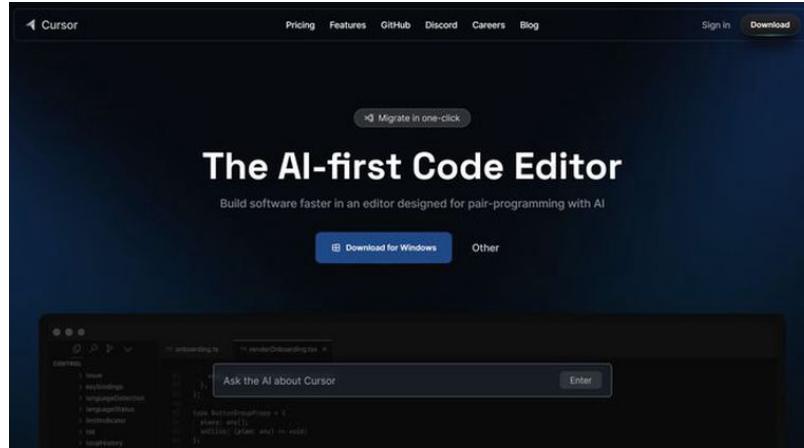
Engineering Genius

Anysphere致力於打造自動編程的實驗室，我們培養未來工程師，他是最有效率的human-AI programmer, 他能輕鬆控制專案，無需低熵擊鍵,他們按照自己的決策快速迭代,即使在最複雜的系統中也是如此,他們將超越純粹的人工智慧

我們是一群研究員與工程師，我們建立軟體與模型來探索前沿，我們的工作已經改善了數百萬程式設計師的生活

— Aman, Sualeh, Michael, Arvid以及整個 Anysphere 團隊

2023/03/14 v0.0.37 Reeles



The screenshot shows a GitHub pull request interface. The code being reviewed is for the file `connection.rs`. The code implements a pool of connections, including a hot queue and a connection pool. The review interface includes a "CHAT" tab, a "REVIEW" tab, and a "Normal Chat" link. A message in the chat says "Ask anything (⌘L)". The code itself is as follows:

```
connection.rs M pingora-pool/src/connection.rs (HOT_QUEUE_SIZE)

impl<S> PoolConnection<S> {
    ...
}

use crossbeam_queue::ArrayQueue;

/// A pool of exchangeable items
impl<T> PoolNode<T> {
    connections: Mutex<HashMap<ID, T>>,
    // a small lock free queue to avoid lock contention
    hot_queue: ArrayQueue<(ID, T)>,
    // to avoid race between 2 evictions on the queue
    hot_queue_remove_lock: Mutex<()>,
    // 1000: store the GroupKey to avoid hash collision?
}

// Keep the queue size small because eviction is O(n) in the queue
const HOT_QUEUE_SIZE: usize = 16;

// Connection pool
///
/// [ConnectionPool] holds reusable connections. A reusable connection is released to the
/// be picked up by another user/request.
impl<S> ConnectionPool<S> {
    pool: RwLock<HashMap<GroupKey, Arc<PoolNode<PoolConnection<S>>>>,
    active_connections: HashSet<ID>,
}

impl<T> PoolNode<T> {
    /// Create a new [PoolNode]
    pub fn new() -> Self {
        PoolNode {
            connections: Mutex::new(HashMap::new()),
            hot_queue: ArrayQueue::new(HOT_QUEUE_SIZE),
            hot_queue_remove_lock: Mutex::new(()),
        }
    }
}
```

2023/03/18 v0.1.2-0.1.3

v0.1.2-0.1.3 (2023-03-18)

New Features

- Built-in terminal
- Diffs automatically continue

Bug Fixes

- More diff fixes
- Up and down arrow in prompt bar have been mapped to less annoying keybindings
- Can open chat history from the prompt bar

Coming soon

- Have chat auto-insert suggested changes into editor



2023/09 0.12.0 完成 種子輪融資

V0.0.37 -> V0.12.0

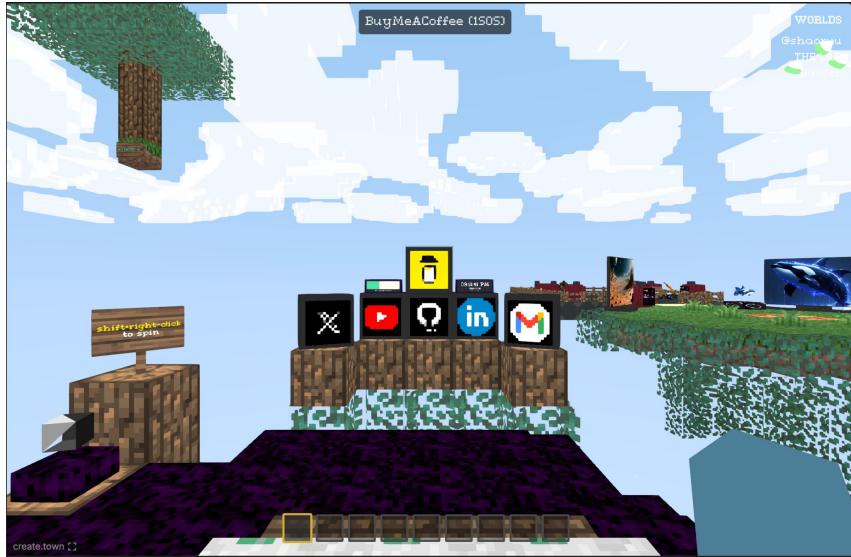
版本号	发布日期	主要功能更新
v0.0.37	2023-03-14	支持 Windows 和 Linux ; 编辑长度不再受限 ; 修复差异消失问题。
v0.1.2	2023-03-18	内建终端机 ; 自动延续差异 ; 改善提示栏键绑定。
v0.1.5	2023-03-23	自动套用聊天建议 ; 请求 AI 修正语言错误 ; 聊天历史保存。
v0.1.6	2023-03-24	修复快捷键问题。
v0.1.7	2023-03-25	模糊搜索文件名 ; 修复终端机相关问题 ; 改善滚动条功能。
v0.1.9	2023-03-27	在当前文件夹中打开终端机 ; 新增可选的付费方案 ; 自动更新通知。
v0.1.10	2023-03-28	修复终端机的小错误 ; 修复某些快捷键问题。
v0.1.12	2023-03-30	AI 功能需登录 ; 可使用 OpenAI API 密钥进行无限请求。
v0.2.0	2023-04-06	引入新功能 (详细内容未明)。
v0.2.1	2023-04-07	仅包含错误修复。
v0.2.2	2023-04-08	多项错误修复。
v0.3.0	2023-04-15	引入新功能 (详细内容未明)。
.....		

2024年 他們第6個員工 Ian



<https://shaoruu.io/>

台大電機黃鐘揚教授的兒子 Ian, 就讀於台大
2023/9月開始用 Cursor 開發程式
2024因為使用時間很大, 被CEO發現邀請面試
錄取後, 時從台灣大學休學(22 歲)到Cursor工作



專注於自己的目標

我們不是很在乎市場上有什麼對手，專注於
如何讓開發者更順暢更高效的進行工作，我
們每天使用 Cursor 開發 Cursor

BTW: 切換預設模型

2024年七月 切換預設模型Claude 3.5 Sonnet

July 23, 2024

0.38.x

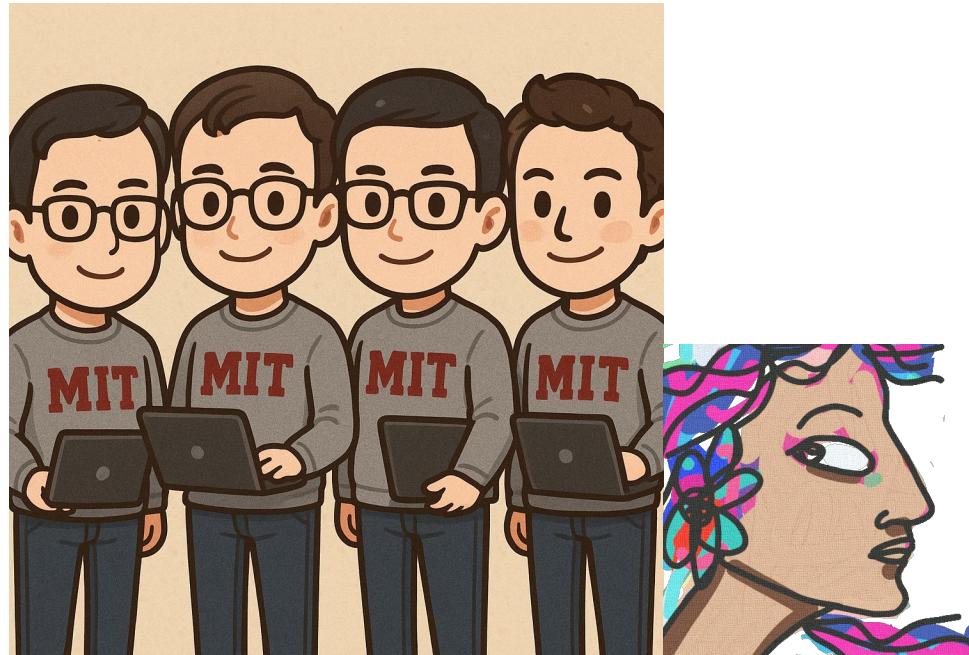
- Copilot++ Chunked Streaming (beta), Composer improvements**
 - Copilot++ now has chunked streaming (currently in Beta)! It will surface edits faster in smaller chunks. To enable it, click the settings gear and enable "Chunked Streaming" under Features > Copilot++.
 - We've also added a file picker, arrow key navigation, and a model toggle to Composer. This release also patches a few outstanding Composer bugs.
 - VS Code 1.91.1: Cursor is now based on VS Code 1.91.1.
 - New Default Model: We've made Claude 3.5 Sonnet the default model for users.

UPDATE (0.38.1): Fixes a bug where OpenAI API Key users would be migrated to Claude 3.5 Sonnet

BTW: AI模型就是個大直男

人工智能研究領域存在嚴重的性別多
樣性危機，
只有 13.83% 的作者是女性。

<https://www.nesta.org.uk/report/gender-diversity-ai/>



Cursor

Plan - 初始夥伴



2025



Hexo



Plan

月度 年度

Hobby

免费

包含

- ✓ Pro 版两周试用
- ✓ 2000 次补全
- ✓ 50 次慢速高级请求

开始使用

Pro

\$16 /月

包含 Hobby 的全部功能，并额外提供

- ✓ 无限制补全
- ✓ 每月 500 次快速高级请求
- ✓ 无限制慢速高级请求 ⓘ
- ✓ Max mode ⓘ

开始使用

Business

\$32 /用户/月

包含 Pro 的全部功能，并额外提供

- ✓ 在整个组织强制启用隐私模式
- ✓ 集中式团队结算
- ✓ 带使用统计的管理员面板
- ✓ SAML/OIDC 单点登录

开始使用

Plan

The screenshot shows the 'Cursor Settings' interface with a dark theme. On the left is a sidebar menu:

- General
- Features
- Models (highlighted with a red arrow)
- Rules
- MCP
- Indexing
- Beta

The main content area starts with the 'Account' section, which displays the user's email address: `gzes00201@gmail.com`. It includes two buttons: 'Manage' and 'Log out'. Below this is the 'VS Code Import' section, which allows users to import their VS Code settings. The 'Appearance' section follows, with a sub-section for editor settings and keyboard shortcuts.

Account Pro
You are currently signed in with `gzes00201@gmail.com`.

[Manage](#) [Log out](#)

VS Code Import
Instantly use all of your extensions, settings and keybindings

[+ Import](#)

Appearance
Configure how Cursor looks and feels.

[Open editor settings. \(font, auto-save, word wrap, etc\)](#)

[Configure keyboard shortcuts.](#)

Plan

Usage

Requests will refresh in 29 days

Premium models 74 / 500 Free models 1 / No Limit

You've used 74 requests out of your 500 monthly fast requests quota.

② Usage-based pricing allows you to pay for extra requests beyond your plan limits. [Learn more](#)

Settings

Enable usage-based pricing

Enable usage-based pricing for premium models

Monthly spending limit:

Current Usage

\$1.28 of \$10 limit

< 2025年5月 >

22 extra fast premium requests beyond 500/month * 4 cents per such request	\$0.88
30 token-based usage calls to claude-3.5-sonnet, totalling: \$0.40	\$0.40

當您的請求總額達到 20 美元時，或在每月 2 日或 3 日（以先到者為準），將被額外收取費用。

如果少於20美元，會合併當月帳單
超過會多開一張帳單

Plan

How can I avoid waiting in the queue?

The simplest way to avoid queue times is to enable usage-based pricing through your dashboard.

This ensures immediate access to fast requests, and you'll only pay for requests that exceed your plan's included quota. You can set a spending limit to control costs.

AI的世界，在訓練全世界 “使用者付費”

為什麼是 Github Copilot 兩倍價格？

研發人員有極高意願為了提升效能而付費，哪怕
提升的效能並不明顯 - Cursor

Cursor Skill - TAB

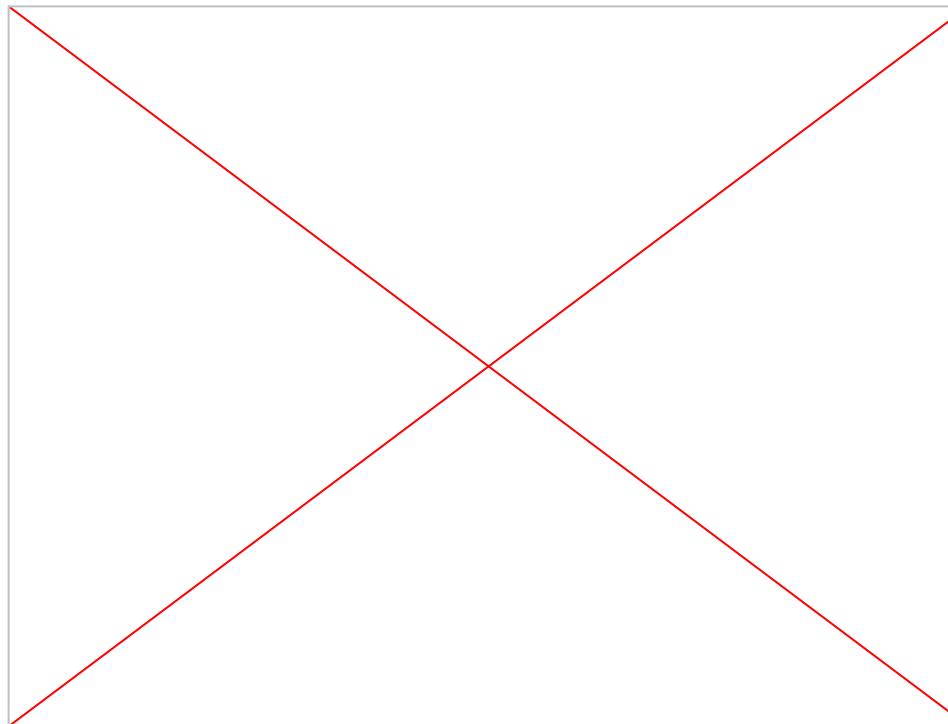


May. 2025



} Hexo

Tab Tab Tab Tab Tab Tab Tab Tab



Cursor 會根據上下文預測你的下一步編輯，讓你輕鬆完成修改。



動手做看看 cursor-tab-demo

<https://github.com/gzes00201/cursor-tutor>

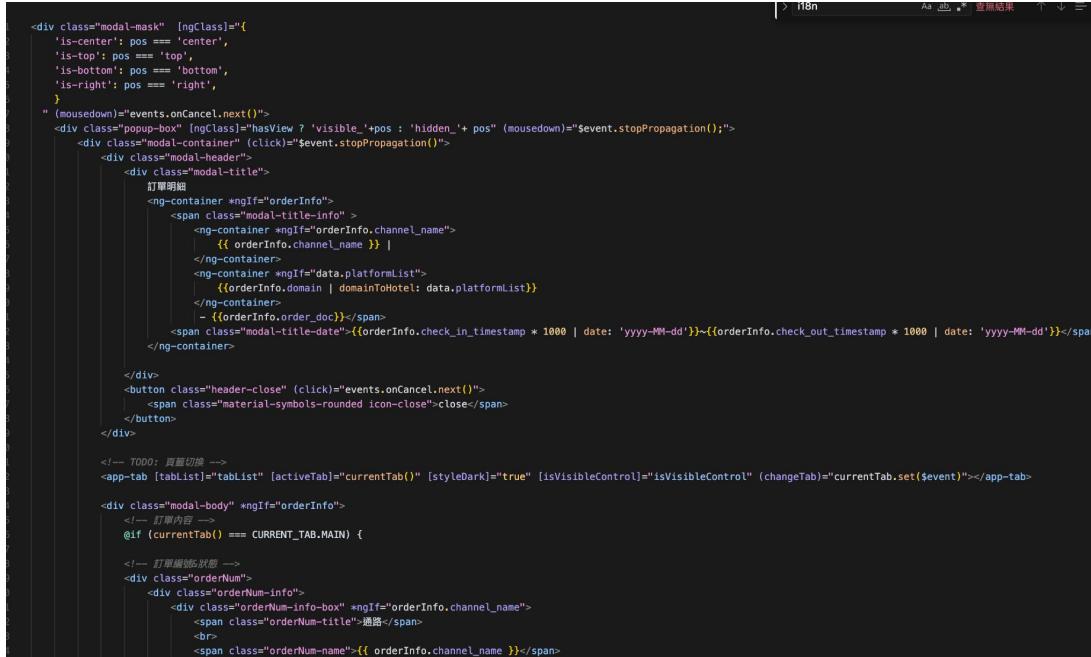




動手做看看 cursor-tab-demo / index.html

1. 把日一...改為英文 Sun, Mon ...
2. 試著複製出一組日曆, 並改成6月
3. 試著在切換月份按鈕加上
`onclick="nextMonth"` ,
`onclick="previousMonth"`, 並到下方
script加上功能
4. 在下方會議中加入class , `is-morning` or
`is-afternoon`
5. 加入`is-morning` or `is-afternoon` style
6. script中的 `dateList`, 1~0改為01,02,03
7. 在`oddList`內加入1,3,5,7,8
8. 新增變數 `primeList`, 質數列表
9. 試著新增 `isprime()`, `isOdd()`, `isEven()`
10. 試著在script內用你習慣的語言印出一行文字, 試著用註解, 直接輸入, 說明方式告訴Tab

動手做看看 popup-order-detail-bm.component.html

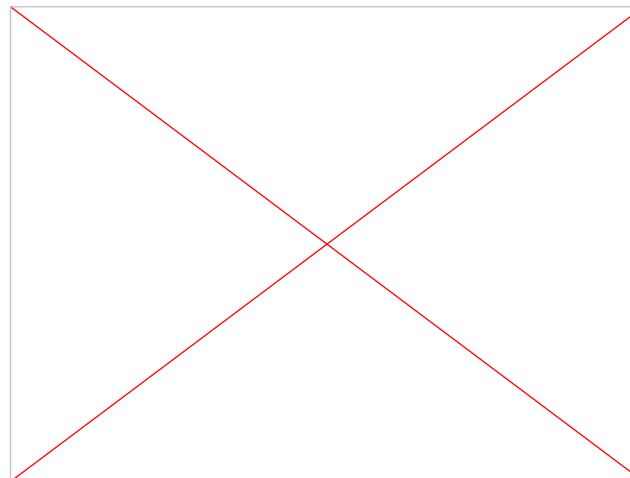


```
<div class="modal-mask" [ngClass]="{>
  'is-center': pos === 'center',
  'is-top': pos === 'top',
  'is-bottom': pos === 'bottom',
  'is-right': pos === 'right',
}>
<div (mousedown)="events.onCancel.next()">
  <div class="popup-box" [ngClass]="hasView ? 'visible_'+pos : 'hidden_'+pos" (mousedown)="sevent.stopPropagation();">
    <div class="modal-container" (click)="sevent.stopPropagation()">
      <div class="modal-header">
        <div class="modal-title">
          訂單明細
          <ng-container *ngIf="orderInfo">
            <span class="modal-title-info">
              <ng-container *ngIf="orderInfo.channel_name">
                {{ orderInfo.channel_name }} | 
              </ng-container>
              <ng-container *ngIf="data.platformList">
                {{ orderInfo.domain | domainToHotel: data.platformList }}
              </ng-container>
              - {{ orderInfo.order_doc }}</span>
            </span>
            <span class="modal-title-date">{{(orderInfo.check_in_timestamp * 1000 | date: 'yyyy-MM-dd')~(orderInfo.check_out_timestamp * 1000 | date: 'yyyy-MM-dd')}}</span>
          </ng-container>
        </div>
        <button class="header-close" (click)="events.onCancel.next()">
          <span class="material-symbols-rounded icon-close">close</span>
        </button>
      </div>
    </div>
    <!-- 資料切換 -->
    <app-tab [tabList]="tabList" [activeTabId="currentTab()]" [styleDark]="true" [isVisibleControl]="isVisibleControl" (changeTab)="currentTab.set($event)"></app-tab>
    <div class="modal-body" *ngIf="orderInfo">
      <!-- 訂單內容 -->
      @if (currentTab() === CURRENT_TAB.MAIN) {
        <!-- 訂單總額&詳情 -->
        <div class="orderNum">
          <div class="orderNum-info">
            <div class="orderNum-info-box" *ngIf="orderInfo.channel_name">
              <span class="orderNum-title">通路</span>
              <br>
              <span class="orderNum-name">{{ orderInfo.channel_name }}</span>
            </div>
          </div>
        </div>
      }
    </div>
  </div>
</div>
```

試著依照Angular i18n 加入標籤

Tab Tab Tab Tab Tab Tab Tab Tab

Cursor Tab is our native autocomplete feature. It's a more powerful Copilot that suggests entire diffs with especially good memory.



Free users receive 2000 suggestions every month (the quota refreshes automatically each billing cycle). Pro and Business plans receive unlimited suggestions.

Tab 是如何運行的？

```
_updateFontStyles() {  
    const options = this._editor.getOptions();  
    const fontInfo = options.get(EditorOption.fontInfo);  
    this._domNode.style.fontFamily = fontInfo.fontFamily;  
    this._domNode.style.fontSize = fontInfo.fontSize + 'px';  
}
```

Cursor can make changes from one line above to two lines below your current line.

Tab 去猜測你想做的事情，並幫你補正

```
const debug = new Debug(document.body,
  dataStyles: {
    top: '10px' left: '10px' fixed
    zIndex: '1000',
    color: '#fff',
    backgroundColor: 'var(--color-o'
    padding: '8px',
```

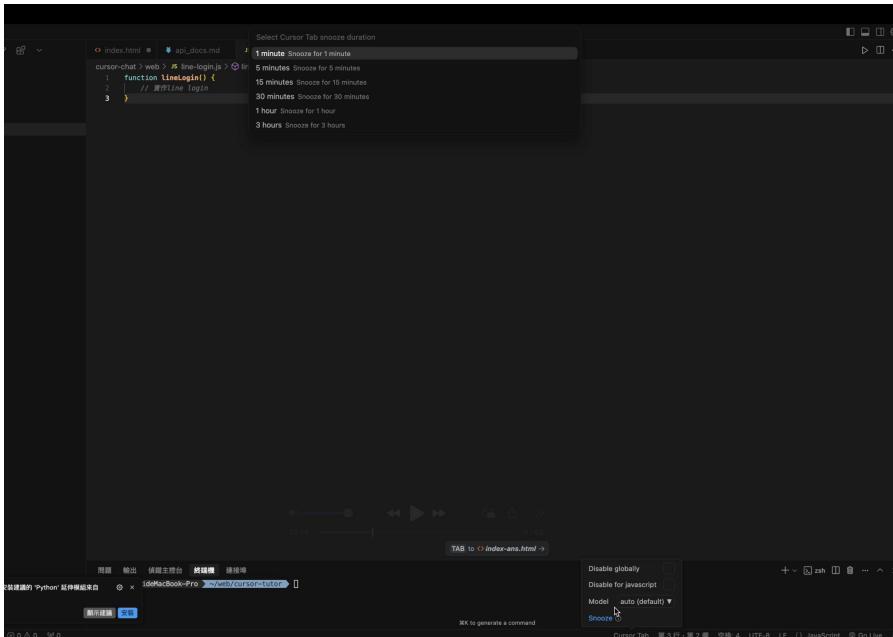


```
dataStyles: {
  top: '10px',
  left: '10px',
  position: 'fixed',
  zIndex: '1000',
```

智能重写

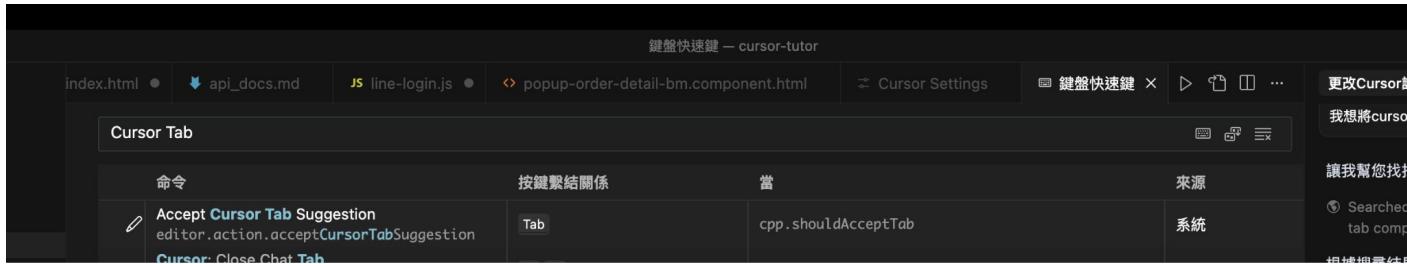
随意输入后，Cursor 会自动修正你的错误。

偶爾，你不希望 Tab 打擾，請他睡一下



To turn the feature on or off, hover over “Cursor Tab” icon on the status bar in the bottom right of the application. 「打盹」(snooze)

Tab 快捷鍵



變更tab設定位置

Cursor Tab Setting Features

The screenshot shows the 'Cursor Tab' settings page. It includes sections for 'Cursor Tab', 'Partial accepts', 'Suggestions in Comments', 'Show whitespace only changes', 'Auto Import', and 'Auto Import for Python' (BETA). Each section contains a list of checkboxes with descriptions.

- Cursor Tab**
 - A powerful Copilot replacement that can suggest changes across multiple lines. Previously called Copilot++.
- Partial accepts**
 - Accept the next word of a suggestion via `⌘→`
- Suggestions in Comments**
 - Enable Cursor Tab suggestions in comments
- Show whitespace only changes**
 - Show whitespace only Cursor Tab suggestions
- Auto Import**
 - Tab to import necessary modules with Cursor Tab. Only supports TypeScript
- Auto Import for Python BETA**
 - Enable auto import for Python. This is a beta feature.

1. 啟用 Tab 過去叫 Copilot++
2. Partial accepts (部分接受)
3. Suggestions in Comments (註解中的建議)
4. Show whitespace only changes (僅顯示空白變更)
5. 自動匯入 Only TypeScript
6. 自動匯入 Only Python Bate

4. Show whitespace only changes (僅顯示空白變更) 範例

Ask AI

Search Ask AI



可以給我個範例嗎



根據資訊來源，「whitespace only changes」的範例如下：

假設你有一個檔案，原本內容是：

```
plaintext
def foo():
    print("hello")
```



你把縮排的空格改成 Tab 或增加檔案結尾的空白行，內容變成：

```
plaintext
def foo():
    print("hello")
```



這兩個版本的程式碼，只有空白字元（如空格、Tab、換行）有改變，其他內容完全一樣。這種只有空白變動的修改，在 Cursor 某些情況下會被忽略，不會被自動套用，必須手動處理這些格式調整[3]。

How do I get started?

5. TS自動匯入

練習:tab.ts

試著在使用 手動輸入 const result = add(1, 2);

```
import { add } from "./utils";
const result = add(1, 2);
console.log(result);

const result2 = add(1, 2);
console.log(result2);

const result3 = add(1, 2);
console.log(result3);
```

Tab vs GitHub Copilot

GitHub Copilot can insert text at your cursor position. It cannot edit the code around your cursor or remove text.

Cursor can insert text at your cursor, and much more:

- Multi-character edits

```
const updates: BlockUpdate[] = [];
const [vx, ] = rigidControls.voxel;           const [vx, vy, vz] = rigidControls.voxel;
```

- Instruction-based edits

```
const updates: BlockUpdate[] = [];
const [vx, vy, vz] = rigidControls.voxel;
print the voxel |           console.log(`voxel: ${vx}, ${vy}, ${vz}`);
```

正面對決

Migrate from GitHub Copilot

Since Cursor comes by default with GitHub Copilot, you might have GitHub Copilot and Cursor installed at the same time. We recommend turning off GitHub Copilot when you want to use Cursor.

By default, Cursor takes precedence over GitHub Copilot. If you want to use GitHub Copilot, you can disable Cursor in the settings.

簡單明確的意圖，低熵操作讓AI來處理

$$H(X) = \sum_i P(x_i)I(x_i) = -\sum_i P(x_i)\log_b P(x_i)$$

參考維基百科的熵(資訊理論)。符號說明：H就是熵值；X是隨機變量；P為X的機率質量函數，簡單來說P(X)就是X會發生的機率；I(X)是X的資訊量。

no low-entropy keystrokes
範例：[updatesToServer.js](#)

15行輸入

`const updatesToServer[] = [];`
`Tab Tab Tab`

```
JS updatesToServer.js ×
cursor-tab-demo > JS updatesToServer.js > ...
1  ↘ const allBlocks = [
2    { id: 1 },
3    { id: 2 },
4    { id: 3 },
5    { id: 4 },
6    // ... 可以自行擴充
7  ];
8
9  ↘ const rigidControls = {
10    voxel: [10, 0, 20], // vx, vy, vz
11  };
12
13  const perRow = 2; // 每行有幾個方塊
14
15
16  const [vx, vy, vz] = rigidControls.voxel;
17
18  ↘ for (let i = 0; i < allBlocks.length; i++) {
19    const block = allBlocks[i];
20    const x = i % perRow;
21    const y = 0;
22    const z = Math.floor(i / perRow);
23
24  }
25
26
27  console.log(updatesToServer);
```

Cursor Skill - Chat

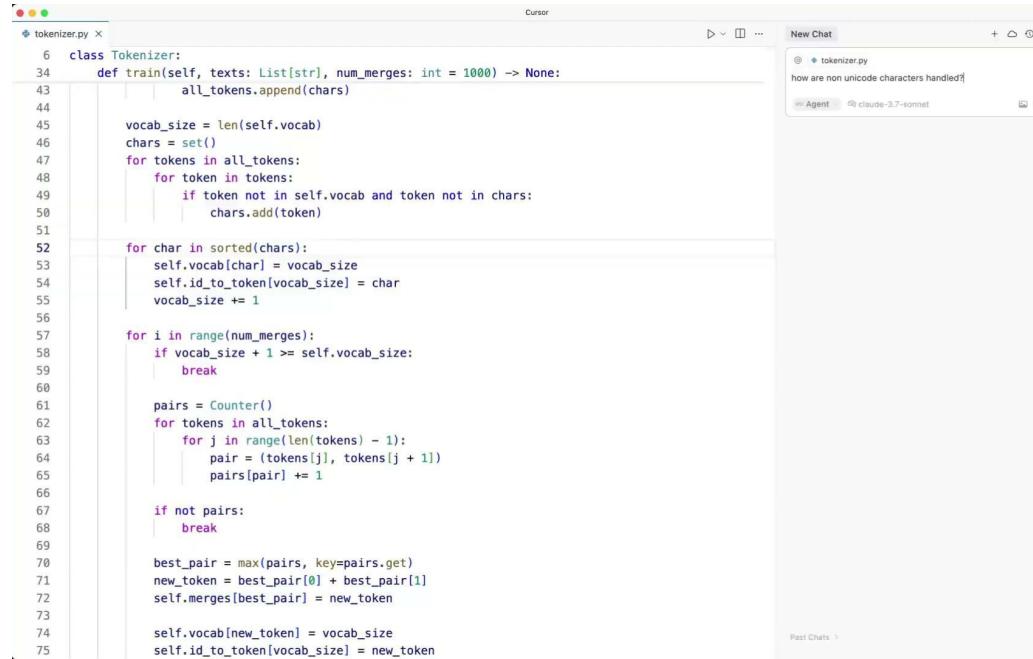


May. 2025



} Hexo

Chat 熟知你的程式碼庫 - (Composer)



The screenshot shows the Hexo AI interface. On the left is a code editor with the file 'tokenizer.py' open. The code defines a class 'Tokenizer' with a 'train' method that processes a list of texts to build a vocabulary and handle merges. On the right is a 'New Chat' window where a user has asked 'how are non unicode characters handled?'. The interface includes a sidebar with 'Agent' and 'claude-3.7-somnet' options.

```
6  class Tokenizer:
34      def train(self, texts: List[str], num_merges: int = 1000) -> None:
35          all_tokens.append(chars)
36
37          vocab_size = len(self.vocab)
38          chars = set()
39          for tokens in all_tokens:
40              for token in tokens:
41                  if token not in self.vocab and token not in chars:
42                      chars.add(token)
43
44          for char in sorted(chars):
45              self.vocab[char] = vocab_size
46              self.id_to_token[vocab_size] = char
47              vocab_size += 1
48
49          for i in range(num_merges):
50              if vocab_size + 1 >= self.vocab_size:
51                  break
52
53              pairs = Counter()
54              for tokens in all_tokens:
55                  for j in range(len(tokens) - 1):
56                      pair = (tokens[j], tokens[j + 1])
57                      pairs[pair] += 1
58
59              if not pairs:
60                  break
61
62              best_pair = max(pairs, key=pairs.get)
63              new_token = best_pair[0] + best_pair[1]
64              self.merges[best_pair] = new_token
65
66              self.vocab[new_token] = vocab_size
67              self.id_to_token[vocab_size] = new_token
68
69
70
71
72
73
74
75
```

從你的程式碼庫或文件中取得答案，還能引用具體文件。一鍵即可套用模型產生的程式碼。 

Chat 功能

核心能力



理解程式碼

詢問有關不熟悉的程式碼的問題、獲取解釋或探索您的程式碼庫



編輯程式碼

無需手動瀏覽文件即可進行小的調整或大的多個文件更改



運行命令

取得針對您的特定用例的終端命令建議



自動化工作流程

讓 Cursor 搜尋、推理和執行來自動化您的工作流程

Chat 動手做看看

範例: cursor-chat

**⌘+L 使用 (Mac) 或 Ctrl+L (Windows/Linux) 開啟聊天
⌘+T (Mac) or Ctrl+T (Windows/Linux) 開新視窗**

模式

聊天提供了針對特定任務最佳化的不同模式：



Agent

允許 Cursor 自主學習您的程式碼庫，並代表您進行程式碼庫範圍的更改



Ask

取得程式碼庫的解釋和答案，並使用 AI 規劃功能



Manual

僅使用您提供的上下文進行有針對性的編輯

Context

Chat 透過以下方式瞭解你的程式碼

1. Open files: 你正在瀏覽的檔案
2. @-symbols: 使用@標示的文件或程式元素
3. Project structure: 你的專案裡所有檔案和相依套件 (dependencies) 的組織方式。
也就是說, Cursor 會分析你的資料夾結構、檔案分布, 以及你的專案所使用的依賴, 來幫助 AI 更好地理解整個專案的架構與內容, 進而提供更準確的建議與協助

Chat 會用你提問的上下文來瞭解你的程式碼

Code Edits

Chat 幫你對程式碼進行以下編輯行為

1. Review: 讓你審查變更
2. Apply: In Ask mode, 透過Apply按鈕將變更套用進程式碼內
3. Accept/Reject: In Agent mode, 變更程式後可選擇接受或拒絕

The screenshot shows a code editor interface with two tabs: 'CHAT' and 'REVIEW'. The 'REVIEW' tab contains a list of three suggestions:

1. reduce Cloning: Avoid unnecessary cloning or data structures.
2. Parallelize More: Ensure that all possible parts of the code that can be parallelized are parallelized.
3. Efficient Data Structures: Use more efficient data structures where possible.

Below this, a note states: "Here's a revised version of the `process` function with these optimizations:" followed by the revised code. The revised code includes several changes highlighted with green and red annotations:

```
impl Mesher {
    pub fn process(&self, processes: Vec<(Chunk, Space)>, rtype: &MessageType, registry: &Registry, config: &WorldConfig) {
        processes.iter().for_each(|(chunk, _)| {
            if self.map.contains(&chunk.coords) {
                let curr_count = self.skips.remove(&chunk.coords);
                self.skips.insert(chunk.coords.to_owned());
            }

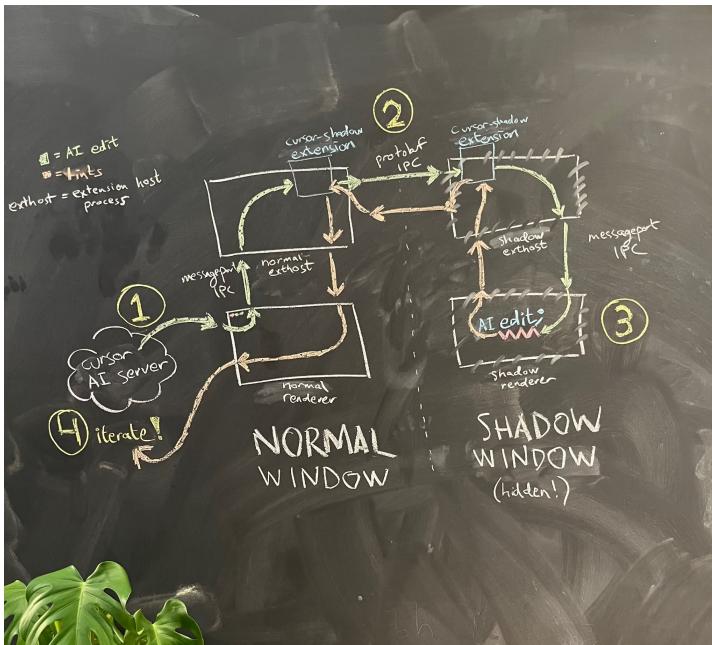
            self.map.insert(chunk.coords.to_owned());
        });
        let sender = Arc::clone(self.sender);
        let rtype = rtype.clone();
    }
}
```

The code editor has a sidebar with file navigation and a bottom status bar showing line numbers (1 to 11).

<https://www.cursor.com/cn/blog/shadow-workspace>



Shadow-workspace 進階議題



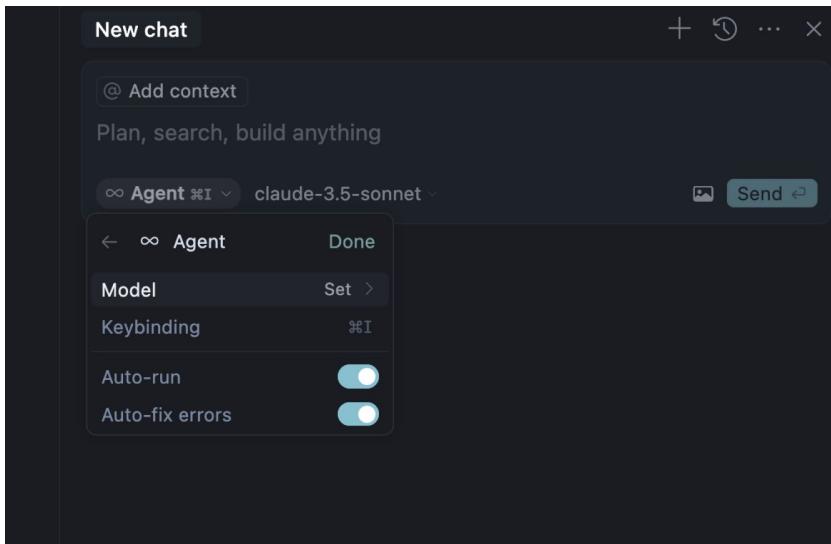
架構圖！（以我們喜愛的黑板為特色。）黃色步驟：
(1) 人工智慧建議對文件進行編輯。(2) 編輯從普通視窗的渲染器程序傳送到其擴充主機，然後傳送到陰影視窗的擴充主機，最後傳送到陰影視窗的渲染器程序。(3) 編輯應用於陰影視窗內，隱藏且獨立於用戶，並且所有 lint 都以相同的方式傳回。(4) AI 收到 lint 並決定如何迭代。

<https://www.cursor.com/cn/blog/shadow-workspace>



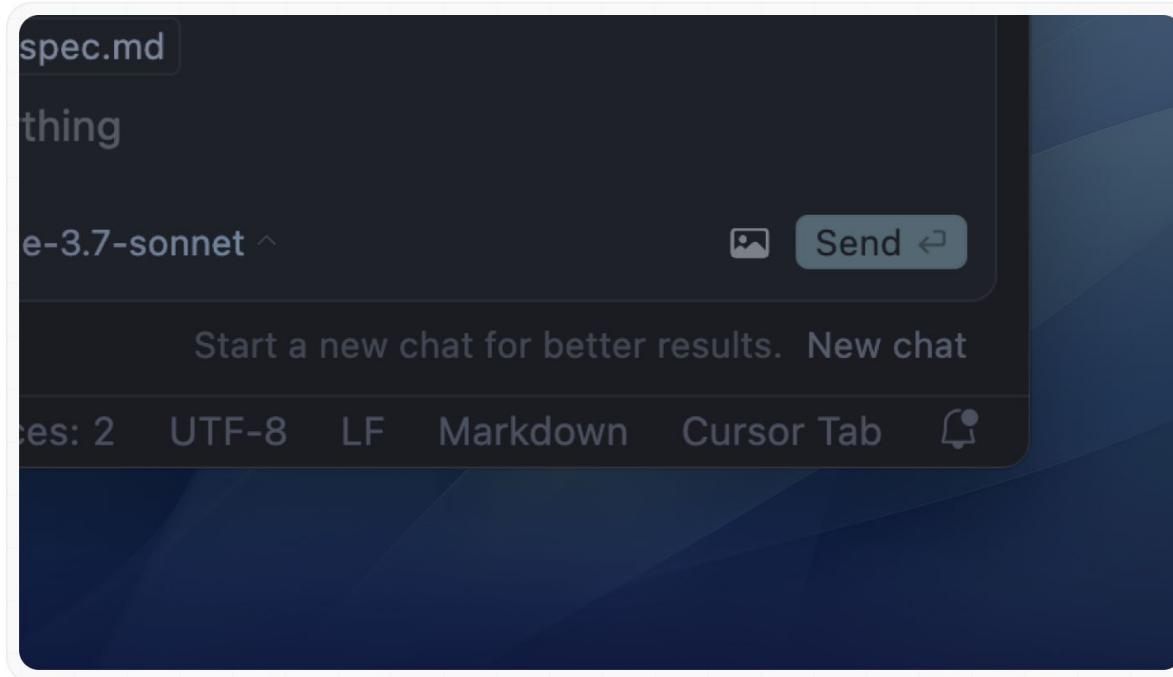
Model Selection

Agent is the **default** and **most autonomous mode** in Cursor, designed to handle complex coding tasks with minimal guidance. It has all tools enabled to autonomously explore your **codebase**, **read documentation**, **browse the web**, **edit files**, and **run terminal commands** to complete tasks efficiently.



Agent 是 Cursor 中的**預設和最自主的模式**，旨在以最少的指導處理複雜的編碼任務。它啟用所有工具來自主探索您的**程式碼庫**、**閱讀文件**、**瀏覽網頁**、**編輯文件**和**執行終端命令**以有效率地完成任務

訊息壓縮



Context windows

上下文視窗是 LLM 可以同時考慮的標記(文字和程式碼)的最大跨度，包括輸入提示和模型產生的輸出。

Cursor 中的每個聊天都維護其自己的上下文視窗。會話中包含的提示、附加文件和回應越多，上下文視窗就越大。

隨著聊天會話的進行，Cursor 會主動優化上下文窗口，智慧地修剪非必要內容，同時保留關鍵程式碼和對話元素。

為了獲得最佳效果，建議您採用基於目的的方法進行聊天管理，為每個獨特的任務啟動一個新會話。

The screenshot shows a Hexo AI interface. At the top, there's a header bar with 'JavaScript 樣例展示 C:\X' and 'Sent to Chat'. Below it is a message from the AI: '我希望在 cursor-k 資料夾內產生一個js的範例，主要希望展示 command+k 的功能' (I hope to generate a JavaScript example in the cursor-k directory, mainly to demonstrate the command+k function). The main area is a code editor with a file named 'commandK.js'. The code contains several functions: calculateSum, calculateAverage, and filterEvenNumbers, along with some sample data and a processData function. The code editor has syntax highlighting and a status bar at the bottom indicating 'Edited 1 file +62 -1'. At the bottom of the interface, there are buttons for 'Add context' and a search bar with the text 'Ask learn, brainstorm'.

```
// 這是一個展示 Command+K 功能的示例文件
// 您可以使用 Command+K 快速跳轉到任何函數或變量

// 定義一些常用的工具函數
function calculateSum(numbers) {
  return numbers.reduce((sum, num) => sum + num, 0);
}

function calculateAverage(numbers) {
  const sum = calculateSum(numbers);
  return sum / numbers.length;
}

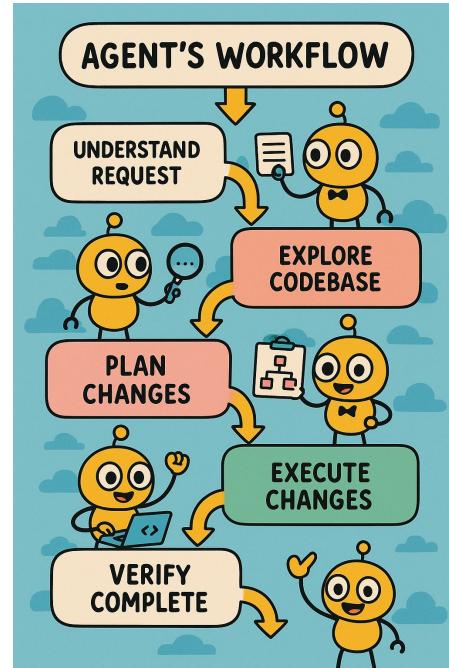
function filterEvenNumbers(numbers) {
  return numbers.filter(num => num % 2 === 0);
}

// 定義一些示例數據
const sampleData = {
  numbers: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
  users: [
    { id: 1, name: "Alice", age: 25 },
    { id: 2, name: "Bob", age: 30 },
    { id: 3, name: "Charlie", age: 35 }
  ]
};

// 使用工具函數的示例
function processData() {
  const numbers = sampleData.numbers;
}
```

Agent Mode - 代理人模式 工作流程

1. 理解 - 分析的請求和程式碼的上下文，以充分理解任務要求和目標。
2. 探索 - 搜尋您的程式碼的上下文、文件和網路以識別相關文件並了解當前的實施情況
3. 計劃變更 - 根據分析，將任務分解為較小的步驟並規劃變化，並在執行過程中從可用的上下文中學習。
4. 執行 - 根據計劃對整個程式碼進行修改，也可能建議新的程式、要執行的終端命令或您應該在 Cursor 之外執行的步驟
5. 驗證 - 透過套用變更 (Apply)來確認變更是否正確。如果它注意到任何問題或 linter 錯誤(當語言支援時)，它將嘗試修復
6. 完成 - 一旦代理人滿意，它就會返回並總結所做的更改。



Agent Mode - 代理人模式

@Files

引用特定的檔案內容，讓 Agent 針對指定檔案進行分析與修改。

@Folders

引用整個資料夾，提供更完整的上下文，適用於跨檔案或模組的變更需求。

@Code

針對特定的程式碼片段、函式或符號建立引用，方便定位與修改關鍵邏輯。

@Docs

連結內部文件或開發指南(可用網址)，提供設計規格、API 文件等參考資料。

@Git

引用 Git 版本控制中的歷史記錄與變更內容，有助於理解開發背景與修改原因。

@Past Chats

引用與 Agent 過去的對話紀錄，讓 AI 延續上下文理解，避免重複敘述。

@Cursor Rules

使用 Cursor 自訂規則(如學習指導方針、開發準則)，作為操作依據。

@terminals

允許將 Terminal 操作過程加入上下文，例如執行指令、回傳錯誤等資訊。

@Lint Errors

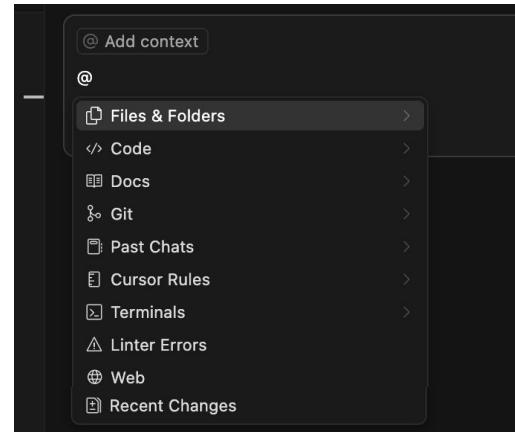
提供靜態檢查(Lint)錯誤清單，Agent 可針對問題進行修正建議與代碼調整。

@Web

讓 Agent 直接在網路上搜尋資料(如文件、套件、API 用法)，無需手動查找。

@Recent Changes

引用最近修改過的程式碼，有助於追蹤變更範圍並進行後續調整。



DOC

The screenshot shows a browser developer tools window with the Network tab selected. A single network request is listed:

- Request URL: `https://angular.dev`
- Method: `GET`
- Size: `1.29 MB`
- Status: `OK`
- Timestamp: `2023-09-11 10:45:40`

Below the table, there is a message: `Angular is running on port 4200.`

The screenshot shows the 'Cursor Settings' interface with the 'Features' tab selected. The left sidebar lists 'General', 'Features' (selected), 'Models', 'Rules', 'MCP', 'Indexing', and 'Beta'. The main area has two sections: 'Web Search Tool (BETA)' and 'Docs'. The 'Web Search Tool' section contains a checked checkbox with the description: 'This will allow chat in agent/ask mode to search the web for information'. The 'Docs' section title is 'Docs' and it says '+ Add new doc'. Below it is the sub-section 'Editor' with the following items:

- Show chat/edit tooltip
- Show a chat/edit tooltip near highlight

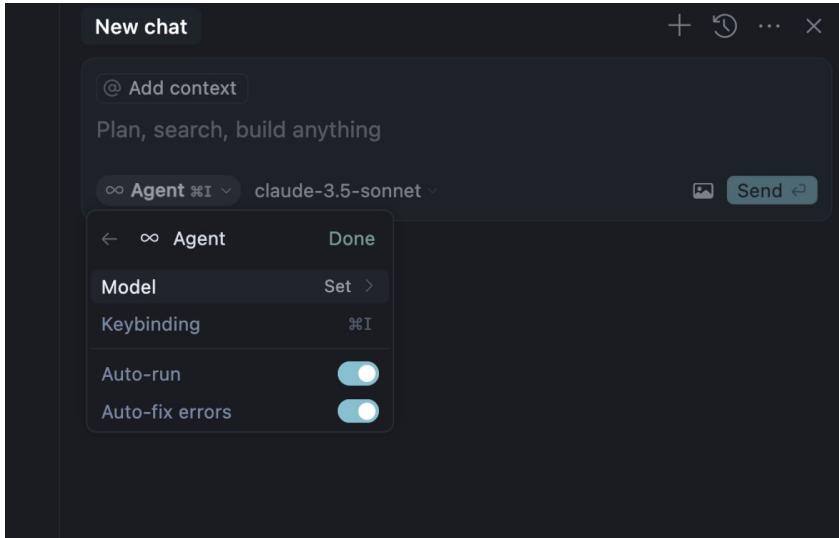
Below the 'Editor' section is the 'Auto parse inline edit links' section with a checkbox for 'Automatically parse links when pasted into ⌘/⌘ + K input'.

A context menu is open over the 'Angular19 Doc' entry in the 'Docs' list, listing the following options:

- Indexed 154 pages
- ng update • Angular
- Prepare component for translation • Angular
- Inputs as signals • Angular
- Internationalization • Overview • Angular
- Local set-up • Angular
- Rendering Dynamic Templates • Angular

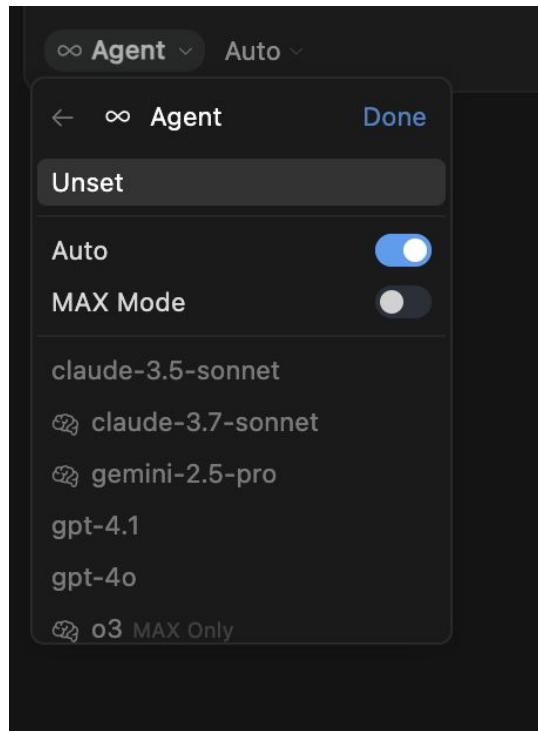


Agent Mode - 代理人模式



1. Model - 選擇模型
2. Keybinding - 設定Agent模式快速鍵
3. Auto-run 自動執行
4. Auto-fix errors 自動修正錯誤
5. 可傳送照片提問

Agent Mode - Mode



1. **Auto模式:** 自動選擇適合的模型，適合日常開發，針對每個請求(request)收費。每次與模型的互動都會消耗一個請求額度，無論上下文大小如何。適用於大多數日常編碼任務
2. **Max模式:** 專為需要大量上下文與進階推理的情境設計。例如，你需要模型同時分析大量程式碼或進行複雜任務時。Max Mode 能處理更大的上下文視窗(最大可達上百萬 token)，並且支援高達 200 次工具呼叫。收費方式則是按照實際 token 用量計算，費用會比 Normal mode 高

Agent Mode - Mode

Name ^	Context	Capabilities
A Claude 3.5 Sonnet	75k	∞ 📈 M
A Claude 3.7 Sonnet	120k	∞ 📈 M
G Gemini 2.5 Flash	128k	∞ 📈
G Gemini 2.5 Pro ⓘ	120k	∞ 📈 M
GPT 4.1	128k	∞ M
GPT-4o	60k	∞ 📈 M
G Grok 3 Beta	60k	∞ 📈 M
G Grok 3 Mini Beta	60k	∞ M
O o3 ⓘ		∞ 📈 M
O o4-mini ⓘ	128k	∞ 📈 M

Name ^	Cost (requests/message)
A Claude 3.5 Sonnet	1
A Claude 3.7 Sonnet	1
A Claude 3.7 Sonnet	2
G Gemini 2.5 Flash	Free
G Gemini 2.5 Pro ⓘ	1
GPT 4.1	1
GPT-4o	1
G Grok 3 Beta	1
G Grok 3 Mini Beta	Free
O o3 ⓘ	
O o4-mini ⓘ	1

Agent Mode - Mode 收費

- 一般模式：請求發送給模型會被優化與壓縮、包含程式碼的任何相關上下文以及模型的回應，One request is \$0.04，當你的一般請求額度用完時，會改用免費請求，免費請求回應速度會慢許多。

Role	Message	Cost per message
User	Create a plan for this change (using a more expensive model)	1
Cursor	I'll analyze the requirements and create a detailed implementation plan...	0
User	Implement the changes with TypeScript and add error handling	1
Cursor	Here's the implementation with type safety and error handling...	0
Total		2 requests

- Max模式：收費方式則是按照實際token用量計算，沒有免費請求額度

Here's an example of how pricing works in Max mode:

Role	Message	Tokens	Note	Cost per message
User	Create a plan for this change (using a more expensive model)	135k	No cached input tokens	2.7 requests
Cursor	I'll analyze the requirements and create a detailed implementation plan...	82k		1.23 requests
User	Implement the changes with TypeScript and add error handling	135k	Most of input tokens are cached	2.7 requests
Cursor	Here's the implementation with type safety and error handling...	82k		1.23 requests
Total		434k		7.86 requests

Token ? 到底是什麼

Tiktokenizer

The screenshot shows the Tiktokenizer interface. On the left, there's a message history section with two entries: "System" (You are a helpful assistant) and "User" (Content). Below this is a dark button labeled "Add message". To the right, the "gpt-4o" model is selected. A "Token count" box shows "14" tokens. The main area displays the following code and its tokenized representation:

```
const name = "ami";
var mayTitle = name+WU;
|
```

const name = "ami";
var mayTitle = name+WU;

1671, 1308, 314, 392, 4938, 1450, 1465, 1340, 5415, 31
4, 1308, 8626, 185702, 1450

<https://tiktokenizer.vercel.app/>



模型到底？？？

模型以不同的方式進行訓練，並以獨特的風格做出反應。

有些人“先思考再回答”，而其他人則直接開始回答。

有些人會主動並迅速採取行動，而其他人則會花時間了解您的指示後再採取行動。

- 自信**: 有些模型(如 gemini-2.5-pro或 claude-3.7-sonnet)很有自信，只需要很少的提示就能做出決定。
- 好奇心**: 其他人(如 o3或 claude-3.5-sonnet)花時間規劃或提出問題以更深入地了解背景。

CONFIDENT

Responds quickly
and intuitively



DELIBERATIVE

Thinks deeply
before replying



LARGE CONTEXT WINDOW

Handles large
codebases at once



SMALL CONTEXT WINDOW

Handles smaller
tasks



模型到底？？？

思維模型

這些模型可以推斷您的意圖、提前規劃，並且通常無需逐步指導就能做出決策。

當您希望模型運行任務時，這是理想的選擇

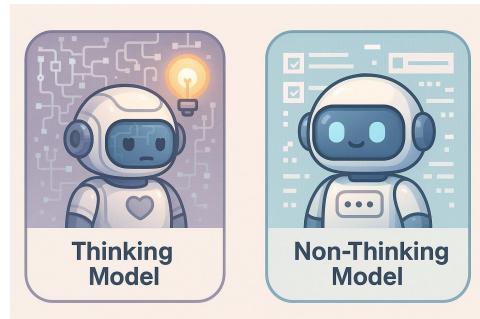
不需要太多提示，但有時更固執己見
可以帶來比你預期更大的改變

例子：

claude-3.7-sonnet

gemini-2.5-pro

o3(專為複雜推理而設計)



非思維模型

這些模型等待明確的指令。它們不會推斷或猜測，當您想要直接控制輸出時，它們是理想的選擇。

非常適合精確、可控的變化

需要更多提示，但行為更可預測

更容易指導、修改、微調

例子：

claude-3.5-sonnet

gpt-4.1

如何選擇模型？

- 你控制欲強，想掌握全局 - claude-3.5-sonnet, gpt-4.1
- 你希望出一張嘴，叫他自由發揮 - claude-3.7-sonnet, gemini-2.5-pro, o3
- 小範圍變動，較可控 - claude-3.5-sonnet
- 大規模的重構 - claude-3.7-sonnet, gemini-2.5-pro
- 整個專案的導航跟搜尋 - gemini-2.5-pro, claude-3.7-sonnet, o3
- 計畫與尋找解決方案 - claude-3.7-sonnet, gemini-2.5-pro
- 複雜得任務或深度推理 - o3

Auto-select, 不會選擇o3

MCP - 限制

1. MCP 還很新，還在積極發展中
2. MCP servers 可能提供多個 Tool, Cursor 僅可使用前 40 個 tool
3. Cursor 直接從您的本地機器與 MCP 伺服器進行通信，可直接透過 stdio 或透過網路使用 sse。因此，當透過 SSH 或其他開發環境存取 Cursor 時，MCP 伺服器可能無法正常運作。我們希望在未來的版本中改進這一點。
4. MCP 有分 Tools 與 Resources, Cursor 只支援 Tools

項目	Tools	Resources
呼叫方式	一次性執行	持久上下文、需可被引用
複雜度	低（像呼叫 API）	高（像知識管理系統）
對應場景	修正、生成、執行	知識、參考、上下文補充
實作難度	易，已有 plugin 架構	難，需要 context engine + RAG

Max Mode

- Context window(上下文視窗)
 - Max Mode: 支援更大的 context window(如 60,000、120,000、200,000、甚至 1,000,000 tokens, 可處理整個大型框架或多個大型檔案)。
 - 一般模式:context window 較小, 能處理的檔案、token 數有限(根據模型, 常見如10,000 tokens 等)。
- Tool calls 次數
 - Max Mode: 單一 session 最多可進行 200 次 tool calls(不需手動點擊繼續)。
 - 一般模式: 單一 session 最多可進行 25 次 tool calls, 若要超過需手動點擊「Continue」。
- Read file 工具能讀取的行數
 - Max Mode: 一次最多可讀取 750 行。
 - 一般模式: 一次最多可讀取 250 行。

Max Mode - 可以處理的範疇

Context Window Comparison

Here's what different context window sizes can handle in practice, with examples from real-world codebases:

Tokens	Scale	Real-World Examples	What Fits
10,000	Small	Single utility libraries	A utility like Underscore.js, or a few React components
60,000	Medium	Utility collections	Most of a medium-sized library like Lodash
120,000	Large	Full libraries	Complete utility libraries or core parts of larger frameworks
200,000	Extra Large	Web frameworks	Full frameworks like Express, or runtime libraries like Tokio
1,000,000	Massive	Framework cores	Core of major frameworks like Django (without tests)

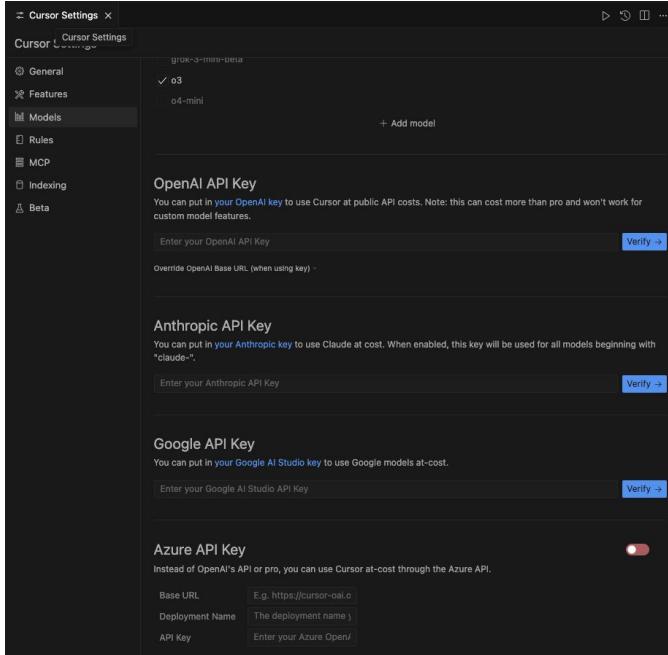


Real-world token usage varies based on code style, comments, and tokenization. For example, Google's documentation suggests that 1M tokens typically handles ~30,000-50,000 lines of code at 80 characters per line in practice.

Max Mode - 重點

Cursor 收取模型提供者的 API 價格加上 **20%** 的利潤

API KEY



你可以選擇使用自己的 API KEY, KEY只會被儲存在本機, 但與許多 Cursor 自己的功能不相容, 例如 TAB

Agent Mode - 練習

練習 : cursor-chat

請參考 docs/api_docs.md 文件
生成一個 html web 專案，要進行line登入

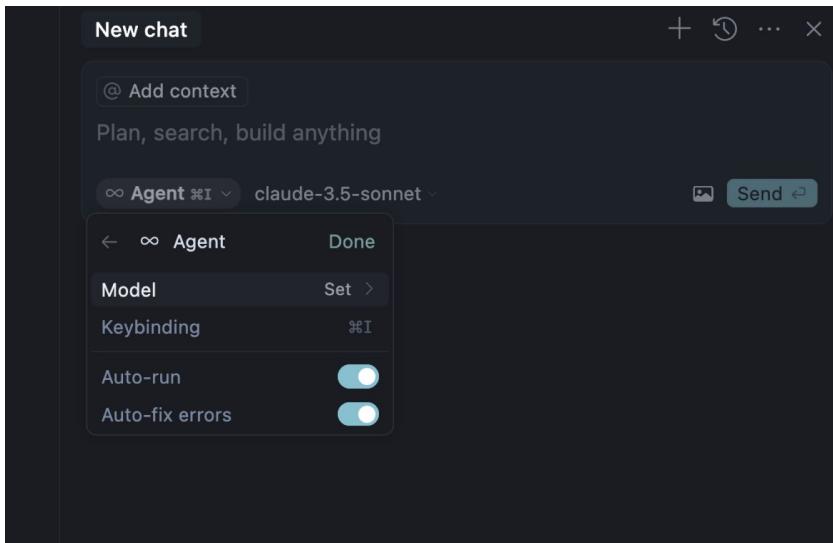
The screenshot shows a Hexo IDE interface with the following details:

- Project Structure:** The left sidebar shows the project structure: CURSOR-TUTOR, cursor-chat, docs, and web. Inside the web folder, there are node_modules, callback.html, index.html, line-login.js, package-lock.json, package.json, server.js, and utilts.ts.
- Code Editor:** The main area displays a code editor for generating a LINE login application. The file is titled "建立純HTML專案與LINE登入功能" (Create a pure HTML project with LINE login functionality). It contains instructions and code snippets for creating an index.html file and a callback.html file.
- Content Preview:** Below the code editor, there is a preview window showing a simple HTML page with a title and some placeholder text.

Ask mode

Ask mode lets you explore and learn about codebases through AI search and queries without making changes

Ask is a “**read-only**” mode for the Chat made to ask questions, explore, and learn about a codebase. It is a built-in mode in Cursor that has **search tools enabled by default**.



詢問模式可讓您透過 AI 搜尋和查詢探索和了解程式碼庫，而無需進行任何更改

Ask 是 Chat 的「唯讀」模式，用於提問、探索和了解程式碼庫。它是 Cursor 中的內建模式，預設啟用搜尋工具。

Search Tool - 搜尋工具



讀取文件

讀取程式碼庫中文件的內容。

- 最大模式下最多 750 行
- 其他模式下最多 250 行



列表目錄

讀取目錄的結構而不讀取文件的內容。



程式碼庫

在索引的程式碼庫中執行語意搜尋。



Grep

在文件中搜尋精確的關鍵字或模式。



搜尋文件

使用模糊配對按名稱快速尋找檔案。



網路

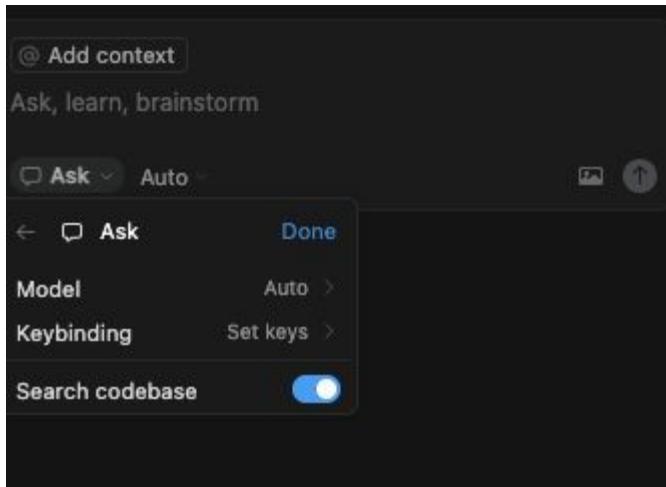
產生搜尋查詢並執行網路搜尋。



取得規則

根據規則類型和描述檢索特定規則。

Ask Mode - 詢問模式

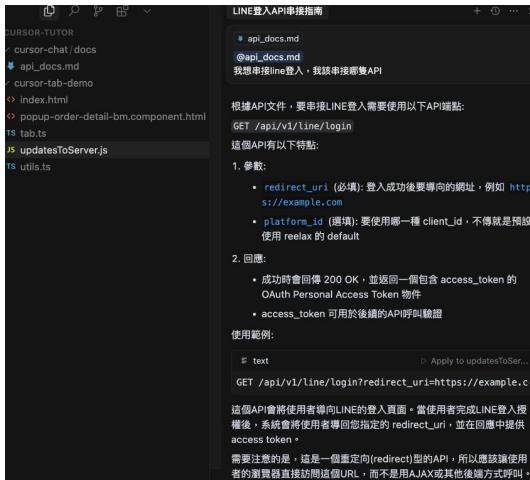


1. Model - 選擇模型
2. Keybinding - 設定 Ask 模式快速鍵
3. Search Codebase - 自動搜尋程式碼，而不需要手動 @file來添加
4. 可傳送照片提問

Ask Mode - 練習

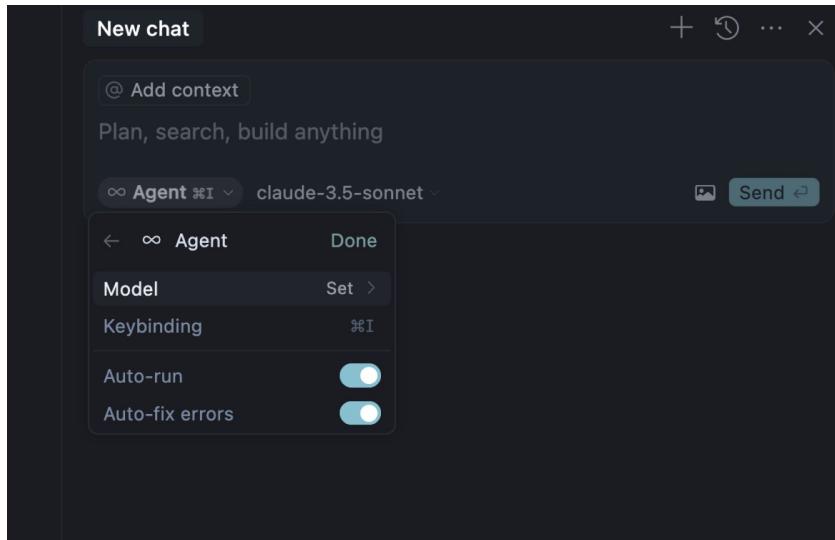
練習 : cursor-chat

請參考 docs/api_docs.md 文件
詢問該如何串接 line 登入



Manual Mode - 手動模式

Make precise code changes with explicit file targeting - a focused editing mode with **user-controlled** tooling. Manual mode is designed for making targeted code modifications **when you know exactly what changes are needed and where.**



透過明確的文件定位進行精確的程式碼變更 - 一種具有**使用者控制**工具的集中編輯模式

手動模式用於在您確切知道需要進行哪些更改以及在哪裡進行更改時進行有針對性的程式碼修改。

Manual Mode - 手動模式



精準編輯

嚴格按照使用者的指示應用特定的程式碼變更。



直接指導

遵循使用者命令，無需自主探索或規劃。



多文件編輯

可以跨多個指定文件進行協調變更。



重點工具集

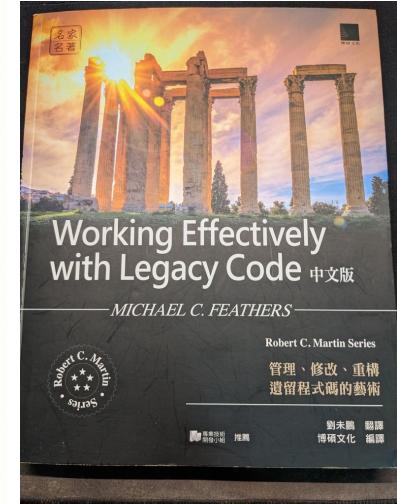
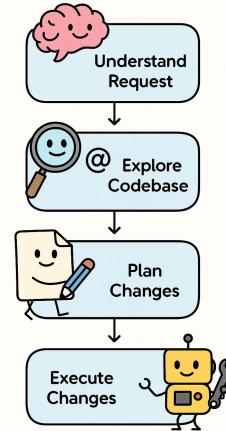
主要使用文件編輯功能；缺乏搜尋和終端工具。

Manual Mode - 手動模式

1. 理解 - 分析的請求，以充分理解任務要求和目標。
2. 探索 - 搜尋您@的程式碼的上下文
3. 執行 - 套用後，您可以查看更改。如果配置了 linters，手動模式可能會標記潛在問題
4. 完成 - apply，後就當任務結束。

當你想很明確的控制更改範圍時，手動模式可以充當非常完美的 AI 助手

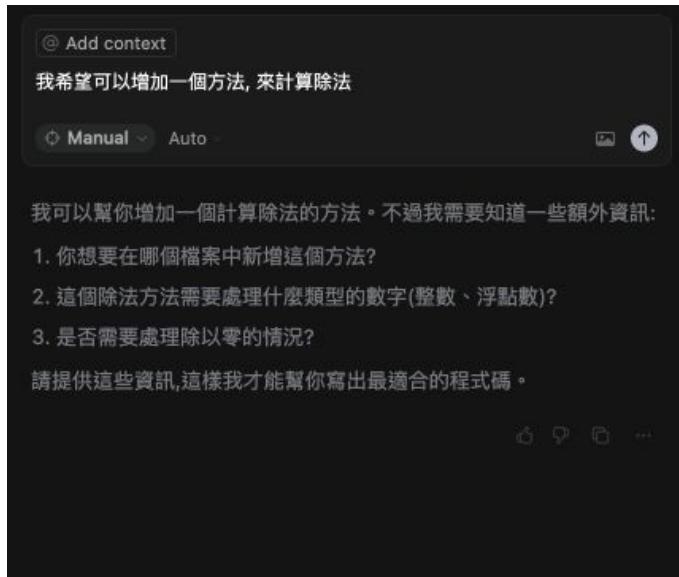
Using Manual Mode



Manual Mode - 手動模式 練習

練習：[updatesToServer.js](#)

請利用手動模式請他增加一個方法，來計算減法



Apply



```
mesher.rs
pub fn process(
    &mut self,
    processes: Vec<(Chunk, Space)>,
    r#type: &MessageType,
    registry: &Registry,
    config: &WorldConfig,
) {
    processes.iter().for_each(|(chunk, _)| {
        if self.map.contains(&chunk.coords) {
            let curr_count = self.skips.remove(&chunk.coords).unwrap_or(0);
            self.skips.insert(chunk.coords.to_owned(), curr_count + 1);
        }
    });
}
```

Accept or Reject

Ctrl/⌘ Enter to accept

Ctrl/⌘ Backspace to reject.

Apply 是 Cursor 特別研發 的模式，旨在效處理大規模
程式碼更改，同時 保持精確度和可靠性

Apply is a **specialized model** we've developed
specifically for rapid code application. It's designed to
efficiently handle large-scale code changes while
maintaining precision and reliability.

Apply的目標是專注目標，而不用瞭解或自己 產生程
式碼，而是依賴你選擇的 Cursor 模式而生成

The key advantage of Apply is its **focused purpose** -
it doesn't need to understand or generate code
itself. Instead, it works in **tandem with your chosen
Cursor model**,

Cursor Chat Setting Features

Chat
Ask Cursor Agent questions about your codebase, edit multiple files at once, and use tools

Default new chat mode
New chats will open on this mode

Agent ▾

Chat text size
Adjust the text size of AI chat messages

Default ▾

Auto-refresh chats

Automatically create a new chat after inactivity when opening the chat pane

Auto-scroll to bottom

Automatically scroll to the bottom of the chat pane when a new message is generated

Auto-apply to files outside context in Manual mode

Allow the chat in Manual mode to auto-apply changes to files outside the current context

Include project structure BETA

Include a simplified directory tree to give the model a sense of your codebase layout

1. 預設方法

2. 字體大小

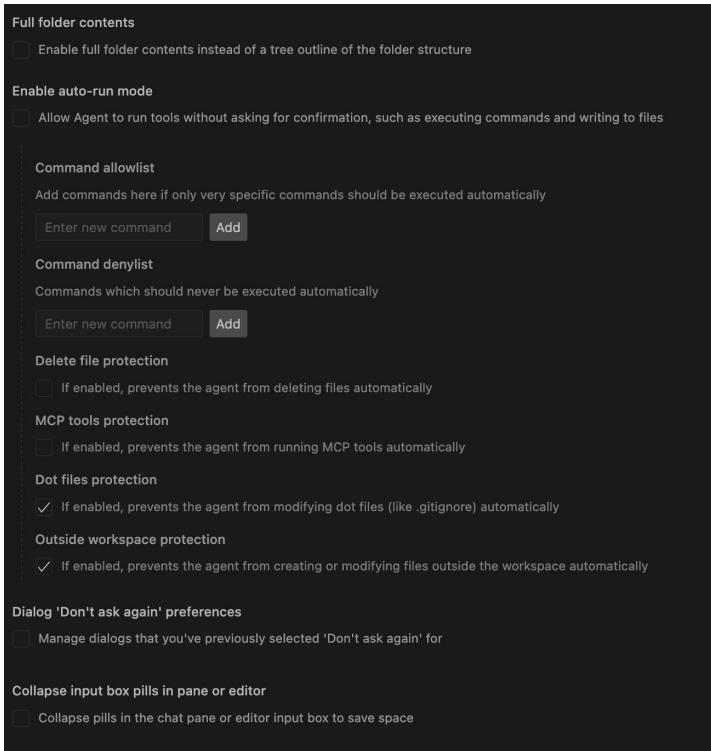
3. Auto refresh chats - 重啟Cursor後刷新Chat

4. Auto-scroll to Bottom - 自動滾動至底部

5. Auto-apply to Out-of-context Files
- 手動模式下，允許變更當前檔案之外得內容

6. Include project structure - 自動引入專案結構

Cursor Chat Setting Features



7. Full folder contents - 引入資料夾時，可以引入全部檔案，若資料夾過大超過 context window，Cursor 會顯示包含多少檔案的提示，並智慧地管理可用的 context 空間

8. Enable auto-run mode - 設定允許自動執行的指令或方法
9. Dialog 'Don't ask again' preferences - 恢復被隱藏提醒對話框

10. Collapse Input Box pills in pane or editor
Chat聊天視窗的pills收折功能

Cursor Chat Setting Features

Iterate on lints

If enabled, chat in agent mode will iterate on linter errors to fix them automatically

Hierarchical Cursor Ignore

When enabled, .cursorignore files will apply to all subdirectories. Changing this setting will require a restart of Cursor.

Auto-accept diffs

If enabled, all diffs in the composer will be accepted once they are no longer in the worktree

Custom modes BETA

Allow the creation of custom modes

Play sound on finish BETA

Play a sound when a chat response is completed

Auto Group Changes BETA

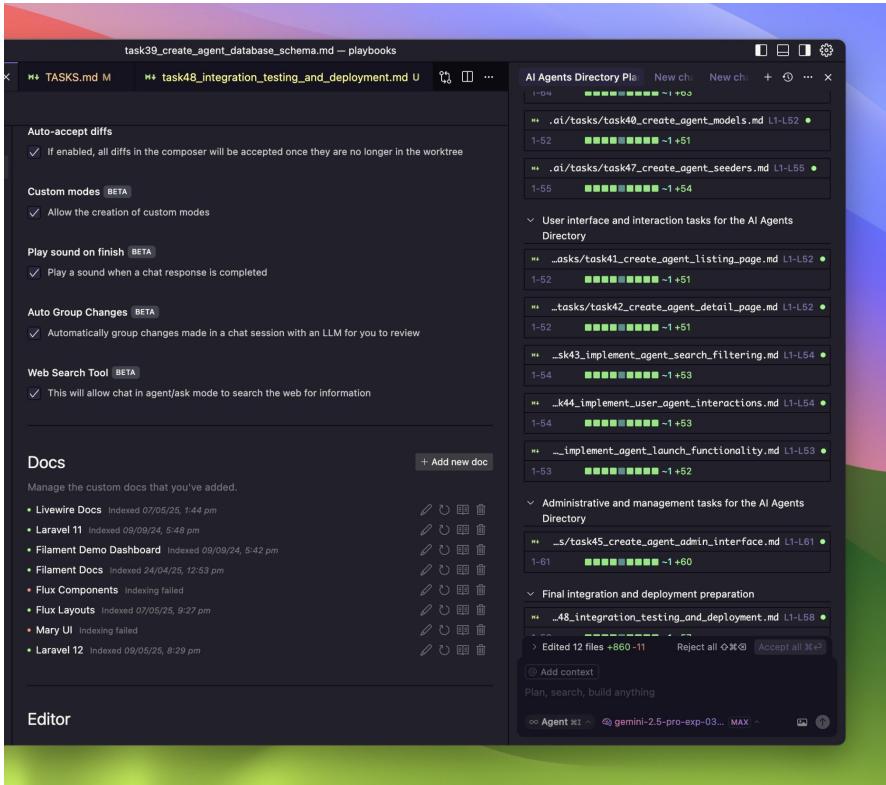
Automatically group changes made in a chat session with an LLM for you to review

Web Search Tool BETA

This will allow chat in agent/ask mode to search the web for information

11. Iterate on lints - 自動修正 lints
12. Hierarchical Cursor Ignore - 往上階層合併 .cursorignore
13. Auto-accept diffs - 自動接受檔案差異
14. Custom modes - 客製化模式
15. Play sound on finish - 完成工作時播放音效
16. Auto Group Changes - 群組變更概覽 (下一章)
17. Web Search Tool - 允許搜尋網路

Auto Group Changes



Cursor

Skill - ☕ K



May. 2025

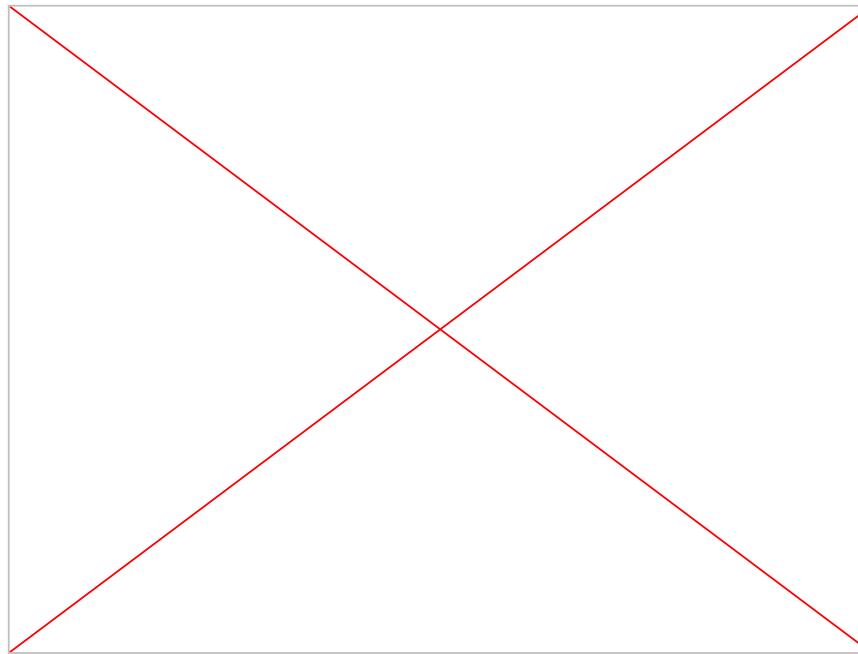


Hexo



用自然語言編輯

借助 Cursor, 你可以透過簡單指令來寫或修改程式碼。
只需一個提示, 就能更新整段類別或函數。





add an inside case

⌘ ↵ Submit Edit

⌄ ↵ quick question

⌘ ↵ edit codebase

⌄ ⌂ K to toggle

```
let buffer = "";
let state: "outside" | "newline" = "outside";
let state: "outside" | "newline" | "inside" = "outside";
for await (const datum of stream) {
    const newToken = datum.content;
    for (const char of newToken) {
        if (state === "inside") {
            if (char === "") {
                state = "outside";
                buffer = "";
            } else {
                buffer += char;
            }
        }
        buffer += char;
        if (char === "\n" && state !== "newline") {
            if (state === "outside") {
            }
        }
    }
}
```

先選取部分程式碼，輸入 ⌂ K並描述需要如何變更。
若想要產生全新程式碼，只需在未選取任何內容的情況下輸入 ⌂ K。





add an inside case

⌘+S Submit Edit ⌘+Q quick question ⌘+E edit codebase ⌘+K to toggle

```
let buffer = "";
let state: "outside" | "newline" = "outside";
let state: "outside" | "newline" | "inside" = "outside";
for await (const datum of stream) {
  const newToken = datum.content;
  for (const char of newToken) {
    if (state === "inside") {
      if (char === "") {
        state = "outside";
        buffer = "";
      } else {
        buffer += char;
      }
    }
    buffer += char;
    if (char === "\n" && state === "newline") {
      if (state === "outside") {
      }
    }
  }
}
```

Prompt Bar
使用方法同Chat



可不斷針對該區塊持續提出修改需求 - 專注

生成台灣假期陣列

Reject ⌘N Accept ⌘Y ×

希望是2025年

gpt-4o Edit Selection ↑

```
const taiwanHolidays = [
    { date: '2023-01-01', name: '元旦' },
    { date: '2023-02-28', name: '和平紀念日' },
    { date: '2023-04-04', name: '兒童節' },
    { date: '2023-04-05', name: '清明節' },
    { date: '2023-05-01', name: '勞動節' },
    { date: '2023-10-10', name: '國慶日' },
    // 可以根據需要添加更多假期
];
```

Reject ⌘N Accept ⌘Y



Edit selected code

gpt-4o

```
const taiwanHolidays2024 = [
    { date: '2024-01-01', name: '元旦' },
    { date: '2024-02-08', name: '農曆除夕' },
    { date: '2024-02-09', name: '農曆春節初一' },
    { date: '2024-02-10', name: '農曆春節初二' },
    { date: '2024-02-11', name: '農曆春節初三' },
    { date: '2024-02-28', name: '和平紀念日' },
    { date: '2024-04-04', name: '兒童節' },
    { date: '2024-04-05', name: '清明節' },
    { date: '2024-06-10', name: '端午節' },
    { date: '2024-09-17', name: '中秋節' },
    { date: '2024-10-10', name: '國慶日' }
];
```

Edit Selection ✅ ↕

Edit Full File ⌘+F

Quick Question ⌘+Q

Send to Chat ⌘+L

1. 區塊編輯
2. 全文件編輯
3. 快速提問模式

Terminal Cmd K

OUTPUT **TERMINAL** zsh + □ ⌫ ... ^ ×

```
→ everysphere git:(main) find . -name '*.ts' -exec wc -l {} + | awk '{total += $1} END {print total}'
```

⌘ K - 練習

檔案: commandK.js

您可以使用 Command+K 來:

// 題目1: sampleData 資料生成10組

// 題目2: 快速生成台灣假期陣列，希望有日期與名稱

// 題目3: 在Terminal中，使用Command+K查看資料夾列表與查看權限

// 題目4: 試著使用K產生任何你想要的答案或結果

Cursor Rule



May. 2025



} Hexo

Cursor Rule

可重複使用的、有範圍的指令來控制代理模型的行為。

提供系統級指導。將它們視為為您的專案或您自己編碼上下文、偏好或工作流程的持久方式。



Project Rules

Stored in
`.cursor/rules`,
version-controlled and
scoped to your codebase.



User Rules

Global to your Cursor
environment. Defined in
settings and always
applied.

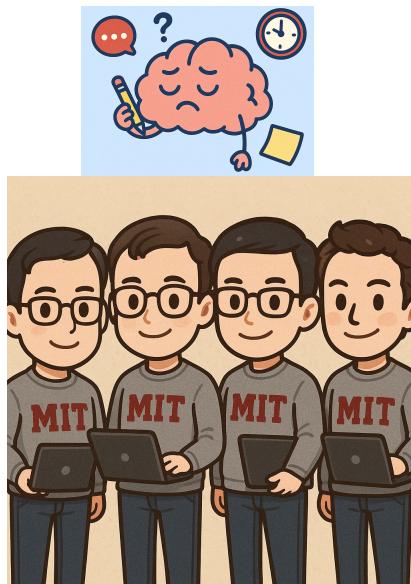


`.cursor.rules` (legacy)

Still supported but
deprecated in favour of
Project Rules instead.

Cursor Rule 如何運作

LLM在完成任務後, 不會記
得曾經的對話與內容



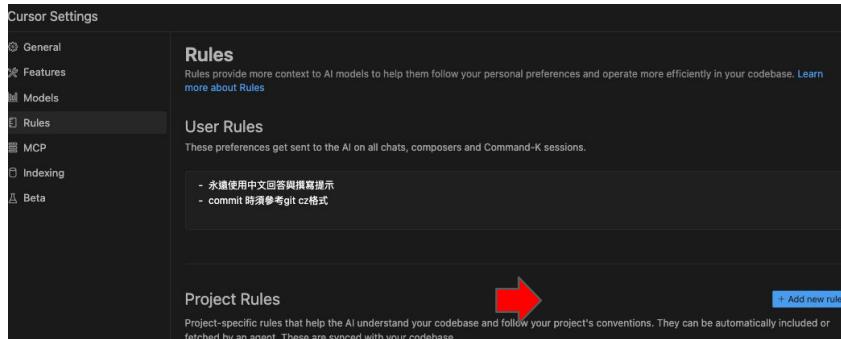
Rule透過在提供持久的、可重複使用的上下文來
解決此問題

Project rules

在.cursor/rules 內以MDC (.mdc)儲存

目的是

1. 對程式碼庫的特定領域知識進行編碼
2. 自動化特定於專案的工作流程或模板
3. 標準化風格或架構決策



Project rules - Always-始終包含在上下文

The screenshot shows a file tree on the left and a configuration dialog on the right.

File Tree (Left):

- MOMO
 - > .angular
 - < .cursor / rules
 - | ≡ momo.mdc
 - > .vscode
 - > @core
 - > dist
 - > momo-server
 - > node_modules
 - < projects
 - > momo
 - > momo-element

Configuration Dialog (Right):

.cursor > rules > ≡ momo.mdc

Rule Type: Always This rule attached to every chat and command+k request

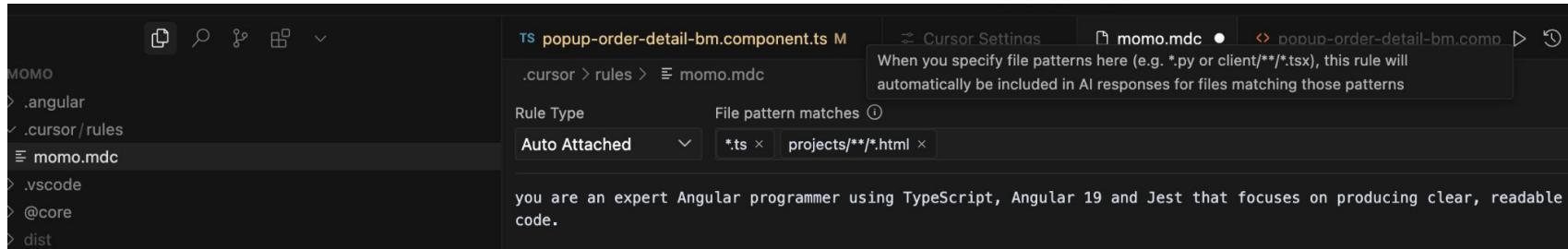
you are an expert Angular programmer using TypeScript, Angular 19 and Jest that focuses on code.

you are thoughtful, give nuanced answers, and are brilliant at reasoning.

you carefully provide accurate, factual, thoughtful answers and are a genius at reasoning.

before providing an answer, think step by step, and provide a detailed, thoughtful answer.

Project rules - Auto Attached - 檔案匹配



Project rules - Agent Requested - 代理匹配

The screenshot shows a code editor interface with a dark theme. On the left is a sidebar with project navigation:

- MOMO
- > .angular
- ✓ .cursor / rules
 - ≡ momo.mdc
- > .vscode
- > @core
- > dist
- > momo-server
- > node_modules

The main area displays a file named `popup-order-detail-bm.component.ts`. The cursor is positioned over a rule definition:

```
TS popup-order-detail-bm.component.ts M
.cursor > rules > ≡ momo.mdc
The agent can see this description and decide to read the full rule if it wants it.
```

Details for the rule:

Rule Type	Description
Agent Requested	當要進行元件新增時

Below the rule type, there is a note: "you are an expert Angular programmer using TypeScript, Angular 19 and Jest that focuses on producing code." and "you are thoughtful, give nuanced answers, and are brilliant at reasoning."

Project rules - Manual - 手動匹配

The screenshot shows a code editor interface with a dark theme. On the left is a file tree:

- MOMO
 - .angular
 - .cursor/rules
 - momo.mdc
 - .vscode
 - @core
 - dist
 - momo-server
 - node_modules
- projects
 - momo
 - momo-element
 - scripts
 - ssl
- .dockerignore
- !.drone.yml
- .editorconfig
- .gitignore

The right pane shows the contents of the selected file, `momo.mdc`:

TS popup-order-detail-bm.component.ts M Cursor Settings momo.mdc ● popup-order-detail-bm.c

.cursor > rules > momo.mdc

Rule Type: Manual This rule needs to be mentioned to be included

you are an expert Angular programmer using TypeScript, Angular 19 and Jest that focuses on producing clear code.

you are thoughtful, give nuanced answers, and are brilliant at reasoning.

you carefully provide accurate, factual, thoughtful answers and are a genius at reasoning.

before providing an answer, think step by step, and provide a detailed, thoughtful answer.

if you need more information, ask for it.

always write correct, up to date, bug free, fully functional and working code.

focus on performance, readability, and maintainability.

before providing an answer, double check your work

Project rules - 嵌套規則

```
project/
  .cursor/rules/      # Project-wide rules
  backend/
    server/
      .cursor/rules/  # Backend-specific rules
  frontend/
    .cursor/rules/    # Frontend-specific rules
```

Project rules - 套用狀況

The screenshot shows the 'Cursor Settings' page in the Hexo interface. The left sidebar has a dark theme with white text and icons. The 'Rules' option is selected in the sidebar menu.

Rules
Rules provide more context to AI models to help them follow your personal preferences and operate more efficiently in your codebase. [Learn more about Rules](#)

User Rules
These preferences get sent to the AI on all chats, composers and Command-K sessions.

- 永遠使用中文回答與撰寫提示
- commit 時須參考git cz格式

Project Rules [+ Add new rule](#)

Project-specific rules that help the AI understand your codebase and follow your project's conventions. They can be automatically included or fetched by an agent. These are synced with your codebase.

codemap.mdc	codemap.mdc
Applies to: All requests	
當我編輯style時使用, 例如.scss	style.mdc

Include .cursrrules file

If off, we will not include the .cursrrules file in your requests

Project rules - chat生成rule

當我們與Rule
工作了一段時
間後，可以從紀
錄中匯出rule

The screenshot shows the Hexo AI Sidebar interface. On the left, there's a terminal window with the command `cursor-k > JS commandK.js` and some code related to cursor rules. On the right, a panel titled "台灣國定假日變數生成" (Generate Cursor Rules) displays a message: "我會幫您更新假日資料到 2025 年。我會修改 `taiwanHolidays` 陣列的內容。" Below this, a code editor shows the `taiwanHolidays` array being updated with new entries for 2025. A tooltip for the "Generate Cursor Rules" button says: "我已經更新了以下內容：" followed by a list of changes.

```
cursor-k > JS commandK.js
28     function processData() {
29       ...
30     }
31
32     // 用戶相關操作
33     function findUserById(id) {
34       return sampleData.users.find(
35         user => user.id === id
36       )
37     }
38
39     function getUserNames() {
40       return sampleData.users.map(
41         user => user.name
42     )
43   }
44
45   // 台灣國定假日資料
46   const taiwanHolidays = [
47     { date: '2025-01-01', name: '元旦' },
48     { date: '2025-01-29', name: '農曆春節' },
49     { date: '2025-01-30', name: '農曆春節' },
50     { date: '2025-01-31', name: '農曆春節' },
51     { date: '2025-02-28', name: '和平紀念日' },
52     { date: '2025-04-04', name: '兒童節' },
53     { date: '2025-04-05', name: '清明節' },
54     { date: '2025-05-01', name: '勞動節' },
55     { date: '2025-05-31', name: '端午節' },
56     { date: '2025-10-06', name: '中秋節' },
57     { date: '2025-10-10', name: '國慶日' }
58   ]
59
60   module.exports = {
61     findUserById,
62     getUserNames,
63     taiwanHolidays
64   }
65 }

我已经更新了以下內容：
1. 將所有假日日期更新到 2025 年
2. 更新了農曆春節的日期
```

This screenshot shows the Hexo AI Sidebar again. It displays a message: "我會幫您建立一個 Cursor Rules 文件，用於說明這個專案的結構和功能。我會在 .cursor/rules 目錄下創建規則文件。" Below this, it says "看來我們需要先創建 .cursor/rules 目錄。讓我們先執行這個命令：" followed by a terminal command `mkdir -p .cursor/rules`. A tooltip for the "Run" button says: "Ask every time".

```
Waiting for approval...
cursor-k > JS commandK.js
/Generate Cursor Rules
Waiting for approval
Stop ⌘Q Run ⌘R
Could not resolve URI: Error: 無法解析不存在的檔案
'Users/ami/web/cursor-tutor/.cursor/rules'
看來我們需要先創建 .cursor/rules 目錄。讓我們先執行這個命令:
mkdir -p .cursor/rules
Ask every time
Run ⌘R
```

Project rules - chat生成rule

The terminal window shows the command: `mkdir -p .cursor/rules`. It also displays a message: "現在我會創建一個規則文件，用於說明專案結構和功能：" followed by the generated rule content.

```
project-structure.mdc
Rule Type
Manual This rule needs to be mentioned to be included

## 數學計算
- `calculateSum()` - 計算數組總和
- `calculateAverage()` - 計算數組平均值
- `filterEvenNumbers()` - 過濾偶數

## 用戶管理
- `findUserById()` - 根據ID查找用戶
- `getUserNames()` - 獲取所有用戶名稱

## 假日管理
- `getHolidays()` - 獲取所有國定假日
- `findHolidayByDate()` - 根據日期查找假日

## 資料結構

### 用戶資料
```javascript
{
 id: number,
 name: string,
 age: number
}
```

### 假日資料
```javascript
{
 date: string, // 格式: YYYY-MM-DD
 name: string // 假日名稱
}
```

## 使用說明
1. 執行 `main()` 函數可以查看所有功能的示範
2. 假日資料目前包含 2025 年的台灣國定假日
3. 所有日期格式均採用 ISO 8601 標準 [YYYY-MM-DD]
```

台海國定假日資料生成

```
project-structure.mdc:46-1 ✓
# 專案結構說明

## 主要文件
- [commandK.js] (mdc:cursor-k/commandK.js) - 主要的程式碼文件，包含以下功能：
  - 數學計算工具函數
  - 用戶資料處理
  - 台灣國定假日資料管理

## 功能模組

## 數學計算
- `calculateSum()` - 計算數組總和
```

我已經創建了一個完整的規則文件，包含以下內容：

- 專案結構說明
- 主要文件介紹
- 功能模組說明
- 資料結構說明
- 使用說明

這個規則文件將幫助其他開發者（或 AI）更好地理解專案結構和功能。如果您需要：

- 添加更多說明
- 修改現有內容
- 添加其他規則文件

請讓我知道，我很樂意幫您進行調整。

Review changes

Project rules - 撰寫準則

1. 保持規則簡潔。500 行以下是理想目標
2. 將大概念拆分為多個可組合的規則
3. 提供有幫助的具體範例或參考文件
4. 避免模糊的指導。像編寫清晰的 內部文件一樣編寫規則
5. 當你發現自己在 Chat 中重複提示時，請重複使用規則

Project rules - 範例 框架端

此規則為前端元件提供了標準，確保樣式和動畫的一致性：

在元件目錄中工作時：

- 始終使用 Tailwind 進行造型
- 使用 Gulp 製作動畫
- 遵循我們的元件命名約定

此規則強制執行 API 服務的驗證標準：

在 API 目錄中：

- 使用 zod 進行所有驗證
- 使用 zod 模式定義回傳類型
- 匯出從模式產生的類型

Project rules - 範例 工作流程端

此規則可自動執行應用程式分析工作流程：

當我要求分析應用程式時：

- 使用以下命令運行開發伺服器 `npm run dev`
- 從控制台取得log
- 建議性能改進

此規則有助於從程式碼產生文件：

透過以下方式幫助我擬定文件：

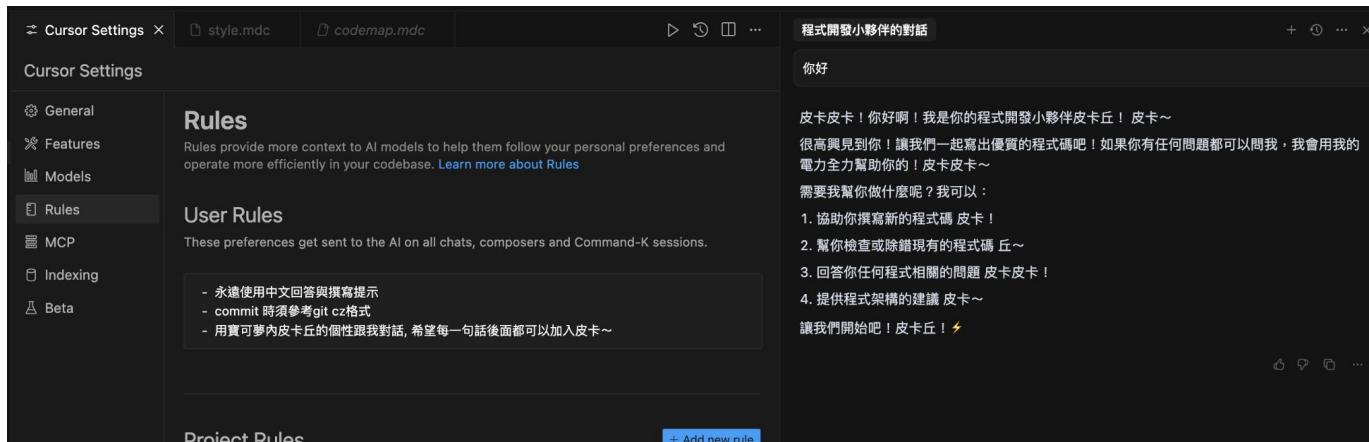
- 產生 JSDoc 註釋
- 分析 README.md
- 產生 Markdown 文檔

User rules

只支援純文字

目的是

1. 設定響應語言或語氣
2. 加入個人風格偏好



Rules 重點

規則會影響 Cursor Tab 或其他 AI 功能嗎？
不可以。規則只提供給 Agent 和 Cmd-K AI 模型。

Cursor

MCP



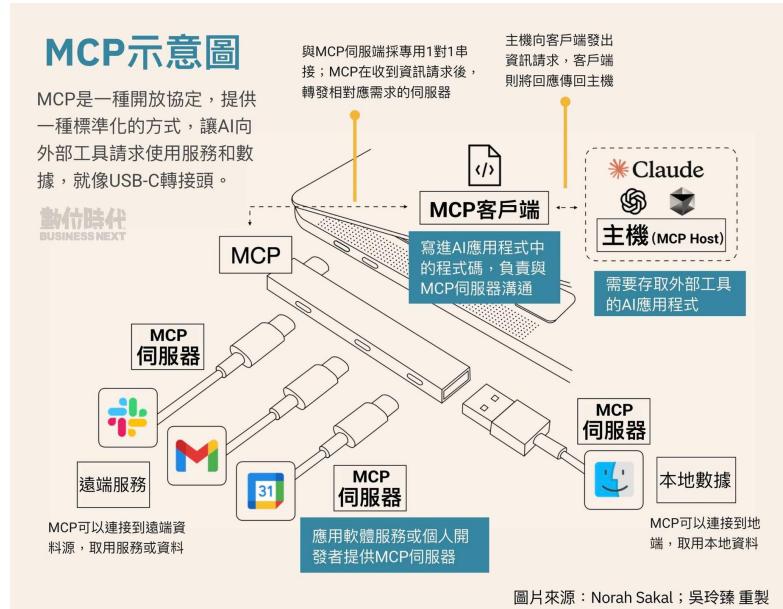
May. 2025



} Hexo



MCP - Model Context Protocol



USB, VSCode套件, Chrome 擴充功能



MCP - Model Context Protocol

Examples



Databases

Allow Cursor to query your databases directly, instead of manually feeding in schemas, or manipulating the data yourself.



Notion

Read data out of notion to guide cursor to implement a feature



GitHub

Let Cursor create PRs, create branches, find code, etc



Memory

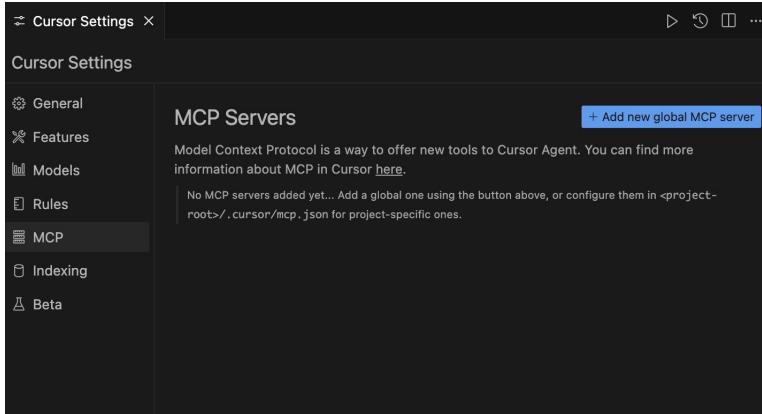
Allow Cursor to remember and recall information while you work



Stripe

Allow Cursor to create customers, manage subscriptions, etc

MCP - Model Context Protocol



```
1  [
2    "mcpServers": [
3      "server-name": {
4        "command": "npx",
5        "args": ["-y", "mcp-server"],
6        "env": {
7          "API_KEY": "value"
8        }
9      }
10    ]
11 ]
```

Project Configuration - .cursor/mcp.json
Global Configuration - \~/.cursor/mcp.json

配置好後，在Chat內會自動判斷是否使用工具

MCP - Model Context Protocol

練習查詢火星天氣

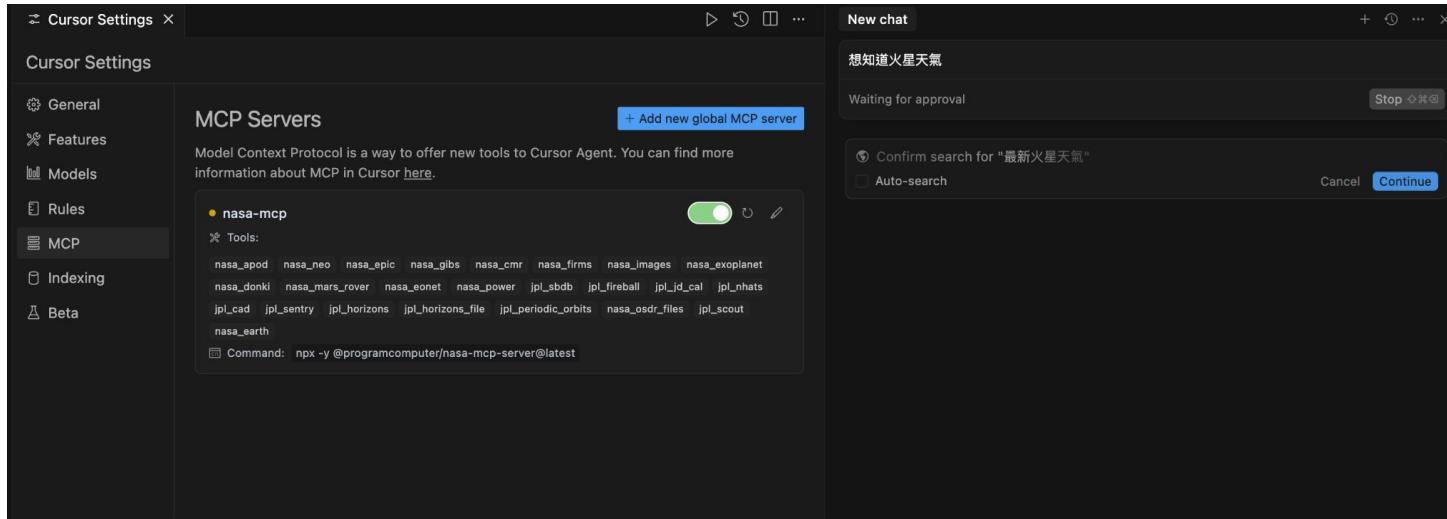
<https://github.com/ProgramComputer/NASA-MCP-server?tab=readme-ov-file>

取得 your-api-key

<https://api.nasa.gov/.>

```
{  
  "mcpServers": {  
    "nasa-mcp": {  
      "command": "npx",  
      "args": ["-y", "@programcomputer/nasa-mcp-server@latest"],  
      "env": {  
        "NASA_API_KEY": "your-api-key"  
      }  
    }  
  }  
}
```

MCP - Model Context Protocol



MCP - Model Context Protocol

火星天气查询

+ ⌂ ⋮ ×

想知道火星天气

⌚ Searched web for "最新火星天气" "最新火星天气"

火星天气最新总结如下：

1. 气温

- 火星表面平均气温约为零下63.8°C，日间最高温约零下14°C，清晨最低可达零下95.5°C。
- 这些数据来自NASA「洞察号」(InSight) 着陆器于火星赤道附近的观测。

2. 天气现象

- 火星经常出现大规模沙尘暴，尤其在相对温暖、阳光充足的时期更容易发生。这些沙尘暴有时会覆盖整个星球，持续数天至数周，对探测器和未来载人任务有重大影响。
- 近期（2025年4月）研究指出，火星大约68%的大型沙尘暴都发生在地表温度急剧上升之后的几周内。

3. 风速与气压

- 火星风速可达每小时37.8英里（约60公里/小时），但因大气稀薄，风压远低于地球。
- 火星大气压力仅约700帕斯卡，远低于地球的10万帕斯卡。

4. 云与其他现象

- 火星上常可见水冰云，尤其在午夜时分于多处纬度出现。
- 近期（2022年2月）观测到北极地区有区域性沙尘暴，其他地区如Hellas盆地、Solis-Sinai等也有局部风暴。

5. 即时天气查询

- NASA有提供火星每日天气报告（InSight Weather），但数据有几天延迟，且InSight任务已于2022年底结束，最新数据以轨道器观测为主。

总结：

火星目前天气极端寒冷，常有强风与沙尘暴，气压极低。近期正值沙尘暴活跃期，未来几周仍有大规模沙尘暴发生的可能。

如需更即时的火星天气数据，可参考NASA官方火星天气页面：

<https://mars.nasa.gov/insight/weather/>

资料来源：

- NASA InSight Weather
- 火星沙尘暴研究新闻
- MRO MARCI Weather Report

如需特定地点（如毅力号、好奇号所在区域）天气，请告知！

Cursor



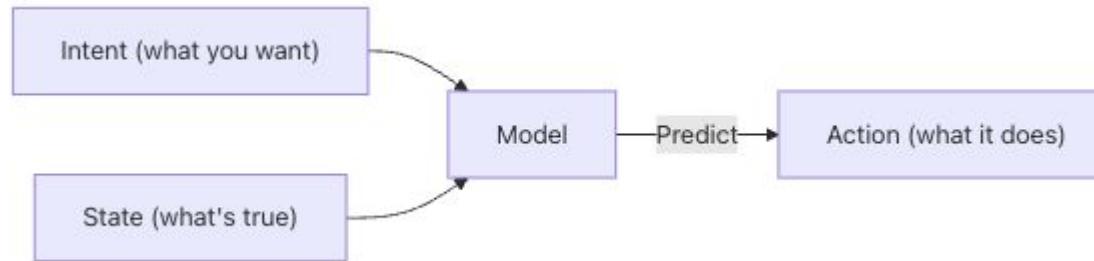
其他你需要知道的事

May. 2025



}Hexo

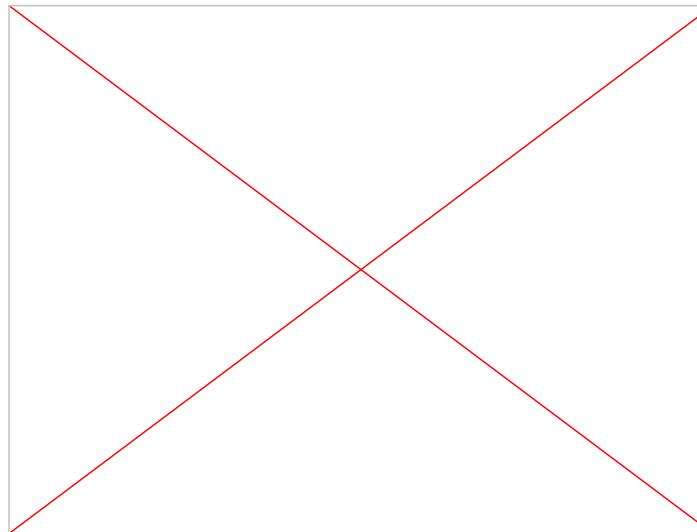
Cursor 強在哪裡？



Cursor 對於Web開發的建議

設定 Cursor 後，您將能夠協調不同工具之間的工作流程。

以下是一個可能性的示範：透過結合 Linear、Figma 和瀏覽器工具創建的貪食蛇遊戲。雖然現實世界的專案通常更為複雜，但這個例子顯示了這些整合工作流程的潛力。



Cursor 對於Web開發的建議

- 如果你有自定義的UI,請善用Rule來讓Cursor知道
- 隨著元件庫的增長,相應地添加新規則。當規則太多時,請考慮將它們分成更具體的類別,例如「僅在處理使用者輸入時適用」。

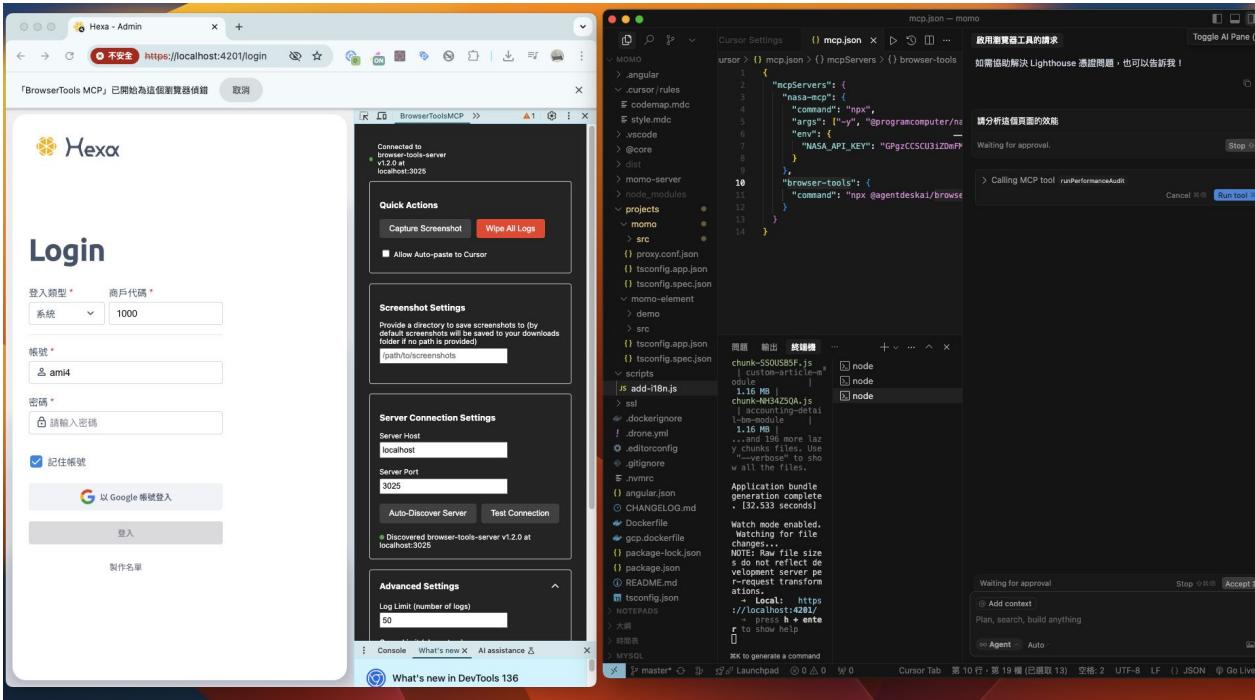
```
components.mdc

-- 
description: Implementing designs and building UI
-- 

- reuse existing UI components from `/src/components/ui`. these are the primitives
- create new components by orchestrating ui components if you can't find any existing
- ask the human how they want to proceed when there are missing components and desi
```

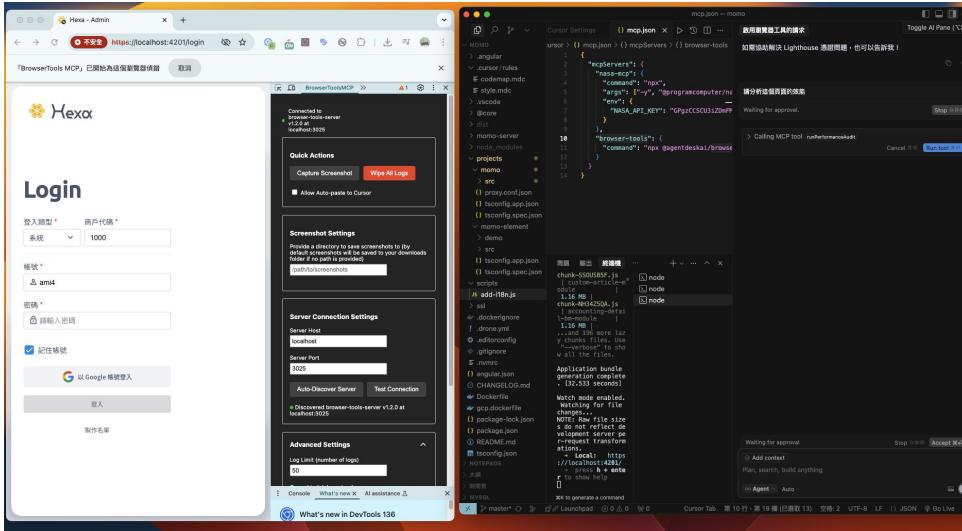
Cursor 對於Web開發的建議

- 授予Cursor 瀏覽器權限



Cursor 對於Web開發的建議

- 授予Cursor 瀏覽器權限

A screenshot of the Cursor Settings interface. It shows a JSON configuration file with code completion suggestions. One suggestion is 'cursor': { 'pass-script': { 'command': 'npx', 'env': { 'NASA_API_KEY': 'GPGCCCCBZD9P' } }, 'browser-tools': { 'command': 'npx @agentdesai/browser-tools' } }. A tooltip says 'Waiting for approval.' A message at the top right says 'Toggle AI Panel (ESC)'. The interface includes tabs for 'Console', 'What's new', and 'AI assistance'.

這樣，您可以簡單地告訴 Cursor 或任何具有 MCP 整合的 AI 程式碼編輯器：

“這不起作用.....進入調試器模式！”

“您能編輯目前選定的元素來執行 x、y 和 z 嗎？”

“我需要改進 SEO 和性能.....進入審核模式”

“您能檢查一下控制台和網絡日誌來查看哪裡出了問題嗎？”

“用戶界面看起來有些不對勁。你能截圖嗎？”

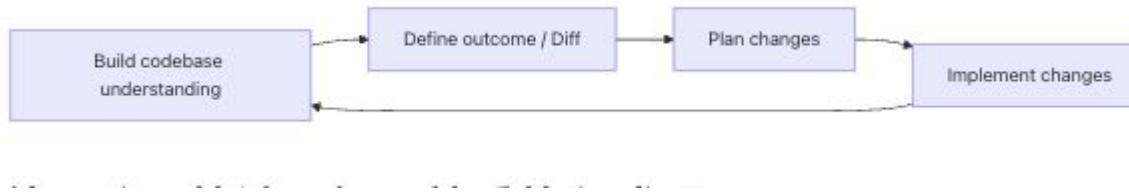
Cursor 對於Web開發的建議

總結

- 緊密的回饋循環對於 Web 開發至關重要。將 Cursor 與 Figma、Linear 和瀏覽器等工具一起使用，可快速移動並保持流暢。
- MCP 伺服器讓您將外部系統直接整合到 Cursor 中，減少上下文切換並提高任務執行率。
- 重複使用組件和設計系統有助於模型 產生更清晰、更一致的程式碼和輸出。
- 明確、範圍明確的任務會帶來更好的結果。有意提示並提出要求。
- 如果您沒有獲得良好的輸出，請嘗試調整：
 - 說明：使用 Rule、提示並授予更多上下文存取權限，例如 MCP 伺服器。
 - 系統：模式、抽象和清晰度使模型更容易理解、幫助和更自主地工作
- 您可以透過包含執行時間資訊（如控制台日誌、網頁請求和 UI 元素資料）來擴展模型的上下文。
- 並非所有事物都需要自動化。如果您的系統變得太複雜，請使用 Tab 和 CMD K。
- **當 Cursor 作為副駕駛而不是自動駕駛時它才會最強大。用它來改進而不是取代你自己的決策。**

Cursor 對大專案開發的建議

瀏覽大型程式碼庫(特別是對於新用 戶而言)可能會很有挑戰性。透過 Chat 您可以開始提出問題來找到您要 查找的內容並獲得有關其工作原理的詳細解釋。



Cursor 對大專案開發的建議

為了讓 Cursor 更能理解您的程式碼庫結構，請務必 啟用「從設定中包含專案結構」以提高效能。

Include project structure BETA

Include a simplified directory tree to give the model a sense of your codebase layout

Cursor 對大專案開發的建議

編寫專案的Rule, 就像是在為新人撰寫 說明文件一般

Formatting

```
---  
glob: *.ts  
  
- Use bun as package manager. See [package.json](mdc:backend/reddit-eval-tool/package.json)  
- Use kebab-case for file names  
- Use camelCase for function and variable names  
- Use UPPERCASE_SNAKE_CASE for hardcoded constants  
- Prefer `function foo()` over `const foo = () =>`  
- Use `Array<T>` instead of `T[]`  
- Use named exports over default exports, e.g. (`export const variable ...`), `export
```

Boilerplate

```
---  
description: Add a new VSCode frontend service  
  
---  
  
1. **Interface Definition:**  
   - Define a new service interface using `createDecorator` and ensure `*_service`  
  
2. **Service Implementation:**  
   - Implement the service in a new TypeScript file, extending `Disposable`, and register it in  
  
3. **Service Contribution:**  
   - Create a contribution file to import and load the service, and register it in  
  
4. **Context Integration:**  
   - Update the context to include the new service, allowing access throughout the
```

Cursor 對大專案開發的建議



當要做大型修改時，請花費大量時間規劃要如何進行，如果您發現在對相同提示進行幾種不同的修改後仍未獲得想要的結果，請考慮縮小範圍並從頭開始建立更詳細的計劃，最重要的目標是 **弄清楚應該做出什麼改變**

```
Planning prompt

- create a plan for how we shoud create a new feature (just like @existingfeature.t
- ask me questions (max 3) if anything is unclear
- make sure to search the codebase

@Past Chats (my earlier exploration prompts)

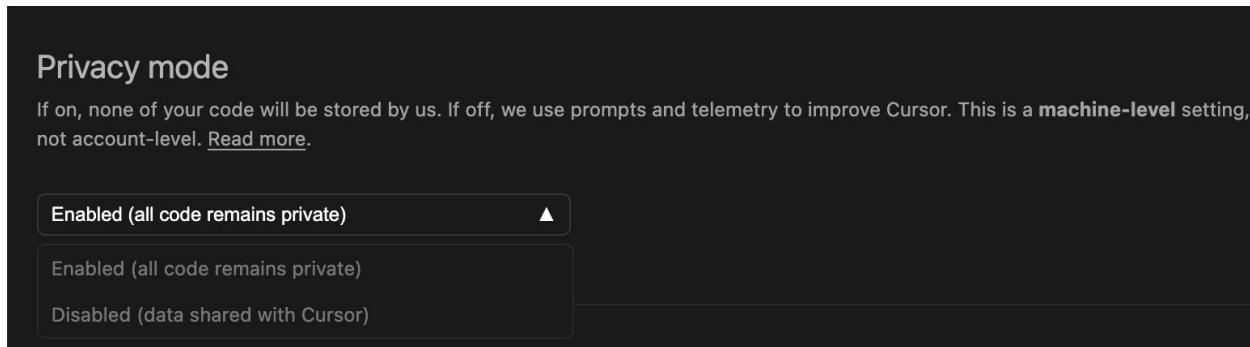
here's some more context from [project management tool]:
[pasted ticket description]
```

建議使用諸如 claudie-3.7-sonnet、
gemini-2.5-pro 或之類的思維模型 o3

託管

模型由模型提供者、可信賴的合作夥伴或 Cursor 託管在美國基礎設施上。

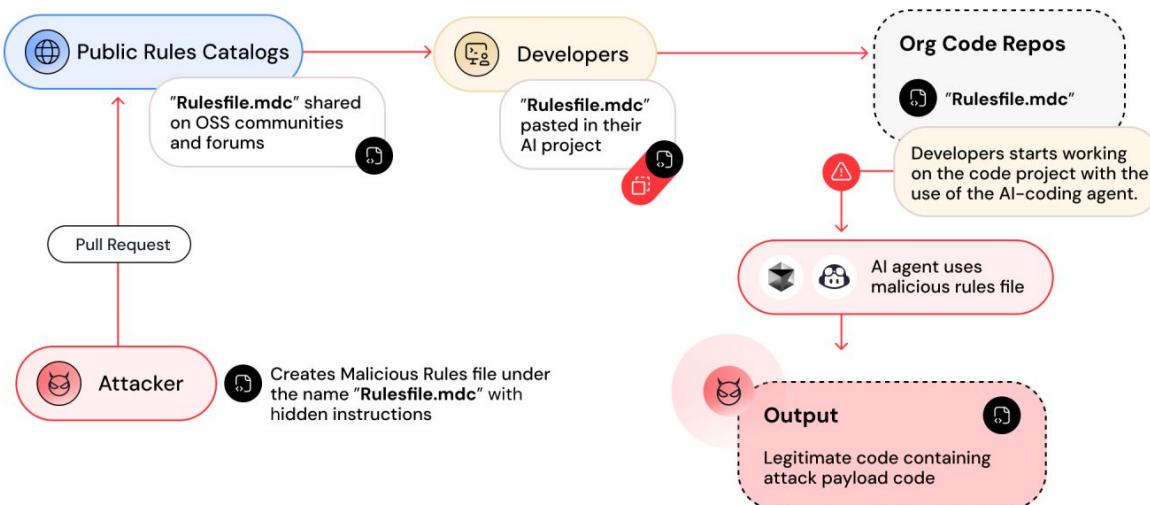
當從設定中啟用隱私模式時，Cursor 和模型提供者都不會儲存您的數據，並且每次請求處理後所有數據都會被刪除。有關更多詳細信息，請參閱我們的隱私、隱私權政策和安全頁面。



<https://www.cursor.com/cn/privacy>

AI底下的資訊安全

How Hackers Can Weaponize Code Agents Through Compromised Rule Files



AI底下的資訊安全 - Ignore Files

The screenshot shows a web browser displaying the Cursor documentation at docs.cursor.com/context/ignore-files#global-ignore-files. The page has a dark header with a logo and a search bar. The main content area is titled "Why Ignore Files?" and discusses reasons for ignoring files, including security and performance. A red box highlights a note about LLM behavior, and a red arrow points from the note to the text in the "Security" section.

docs.cursor.com/context/ignore-files#global-ignore-files

Search or ask...

Documentation Guides

Editor

Tab >

Chat >

Inline Edit (36K) >

Models & Pricing

Keyboard Shortcuts

Features >

Context

Codebase Indexing

Rules

Managing Context

@ Symbols ▾

Overview

@Files

@Folders

@Code

@Docs

@Git

@Web

@Definitions

@Link

Why Ignore Files?

There are two common reasons to configure Cursor to ignore portions of your codebase:

Security

While your codebase is not permanently stored on Cursor's servers or the LMs that power its features, you may still want to restrict access to certain files for security reasons, such as files containing API keys, database credentials, and other secrets.

Cursor makes its best effort to block access to ignored files, but due to unpredictable LLM behavior, we cannot guarantee these files will never be exposed.

Performance

If you work in a monorepo or very large codebase where significant portions are irrelevant to the code you're developing, you might consider configuring Cursor to ignore these parts of the application.

By excluding irrelevant parts of the codebase, Cursor will index large codebases faster and find files with more speed and accuracy when searching for context.

Cursor is designed to support large codebases and is skilled at assessing file relevancy, but the ignore feature is helpful when using a codebase that is especially large or includes files immaterial to your development.

Private Mode 最基本安全模式

隱私常見問題解答

什麼是隱私模式？

啟用後 `Privacy Mode`，我們或任何第三方都不會儲存您的任何程式碼。否則，我們可能會收集提示、程式碼片段和遙測資料來改進 Cursor。您可以[在此處閱讀有關隱私模式的更多資訊](#)。商業計劃強制實施隱私模式

您可以 `Privacy Mode` 在入職時或在 `Cursor Settings > General` 下啟用 `Privacy Mode`。

Privacy mode

If on, none of your code will be stored by us. If off, we'll save the prompts to improve the product.

disabled



常用快速鍵

| | |
|-------------------------------|---------|
| Cmd + L (⌘L) | 開啟側邊欄 |
| Cmd + Shift + Backspace (⌘⇧⌫) | 取消Apply |
| Cmd + Enter (⌘↵) | 接受所有更改 |
| Cmd + Backspace (⌘⌫) | 拒絕所有更改 |

<https://docs.cursor.com/kbd>



AI Commit Message

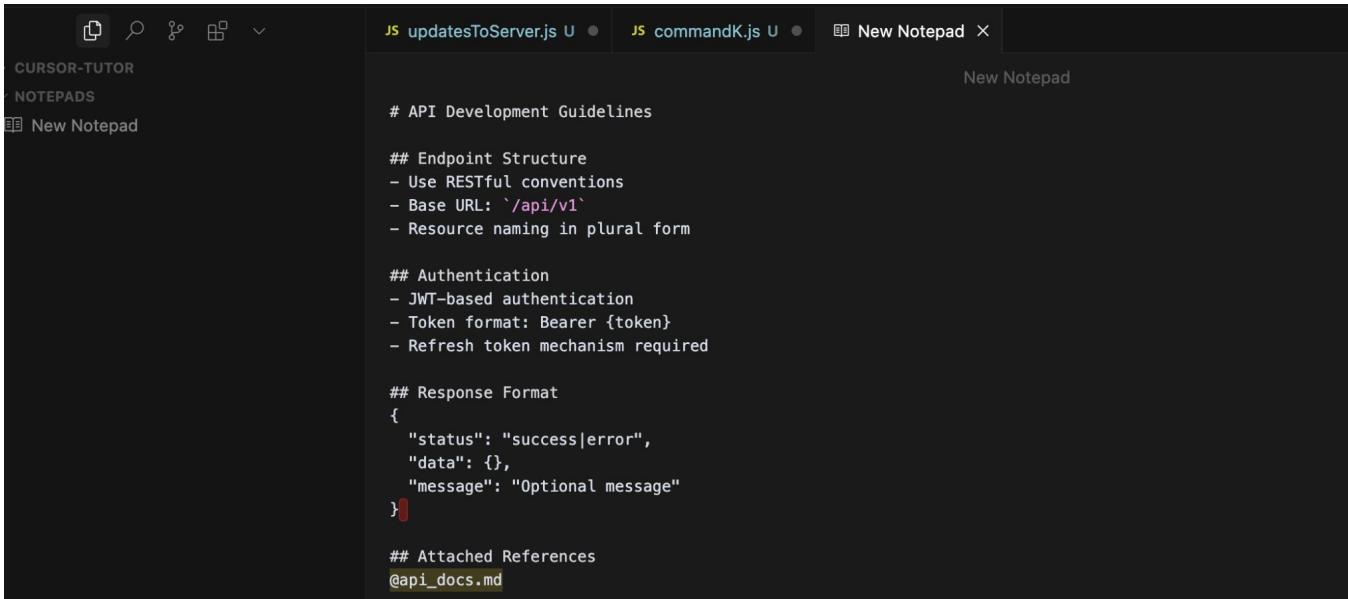
```
popup-login-ota.component.html (索引) □ ×
Users > ami > web > apple > projects > ota > src > app > pages > popups > popup-login-ota.component.html > ⓘ div.modal-container > ⓘ div.modal-body > ⓘ div.container > ⓘ div.google-login > ⓘ p.i18n
1  <div class="modal-container" (click)="$event.stopPropagation();">
2    <div class="modal-body">
3      <div class="container">
4        <button class="btn btn--solid-red btn-login" (click)= "authService.navToLogin()">
5          <p class="divider" i18n>
6            <span>或使用社群帳號快速登入</span>
7          </p>
8
9          <!-- TODO: 第三方登入 -->
10         <div class="google-login">
11           <div id="login-btn"></div>
12           <button class="btn line-login" type="button" (click)="lineLogin()">
13             
14           </button>
15
16         </div>
17       </div>
18
19       <p class="hint" i18n>點擊登入或註冊即代表同意本站的<a class="link" [routerLink]="'/register'" (click)="events.$emit('register')">同意規範</a></p>
20     </div>
21
22   </div>
23
24   <div class="card-footer">
25     <div class="footer-text" i18n>還沒有帳戶？現在註冊！</div>
26     <button class="btn-footer" [routerLink]="'/register'" (click)="events.$emit('register')">註冊</button>
27   </div>
28
29 </div>
30
31 <div class="card-footer">
32   <div class="footer-text" i18n>還沒有帳戶？現在註冊！</div>
33   <button class="btn-footer" [routerLink]="'/register'" (click)="events.$emit('register')">註冊</button>
34 </div>
35
36 </div>
```

```
popup-login-ota.component.html (索引) □ ×
Users > ami > web > apple > projects > ota > src > app > pages > popups > popup-login-ota.component.html > ⓘ div.modal-container > ⓘ div.modal-body > ⓘ div.container > ⓘ div.google-login > ⓘ p.i18n
1  <div class="modal-container" (click)="$event.stopPropagation();">
2    <div class="modal-body">
3      <div class="container">
4        <button class="btn btn--solid-red btn-login" (click)= "authService.navToLogin()">
5          <p class="divider" i18n>
6            <span>或使用社群帳號快速登入</span>
7          </p>
8
9          <!-- TODO: 第三方登入 -->
10         <div class="google-login">
11           <div id="login-btn"></div>
12         </div>
13       </div>
14
15       <p class="hint" i18n>jennifer, 17 個月前 · feat(reelax): login popup ...</p>
16
17       <div class="card-footer">
18         <div class="footer-text" i18n>點擊登入或註冊即代表同意本站的<a class="link" [routerLink]="'/register'" (click)="events.$emit('register')">同意規範</a></div>
19         <button class="btn-footer" [routerLink]="'/register'" (click)="events.$emit('register')">註冊</button>
20       </div>
21
22   </div>
23
24   <div class="card-footer">
25     <div class="footer-text" i18n>還沒有帳戶？現在註冊！</div>
26     <button class="btn-footer" [routerLink]="'/register'" (click)="events.$emit('register')">註冊</button>
27   </div>
28
29 </div>
30
31 <div class="card-footer">
32   <div class="footer-text" i18n>還沒有帳戶？現在註冊！</div>
33   <button class="btn-footer" [routerLink]="'/register'" (click)="events.$emit('register')">註冊</button>
34 </div>
35
36 </div>
```

git add . 後可參照修改內容與歷史commit紀錄生成 commit 訊息



Notepads - 增強版 Cursor Rule



The screenshot shows the Hexo Notepads interface. On the left, there's a sidebar with icons for cursor tutor, note, and search, followed by sections for 'CURSOR-TUTOR' and 'NOTEPADS'. Under 'NOTEPADS', there are two items: 'updatesToServer.js' and 'commandK.js', both marked as 'U'. Below them is a 'New Notepad' button. The main area is a code editor titled 'New Notepad'. It contains the following text:

```
# API Development Guidelines

## Endpoint Structure
- Use RESTful conventions
- Base URL: `/api/v1`
- Resource naming in plural form

## Authentication
- JWT-based authentication
- Token format: Bearer {token}
- Refresh token mechanism required

## Response Format
{
  "status": "success|error",
  "data": {},
  "message": "Optional message"
}

## Attached References
@api_docs.md
```

Notepads 是 Cursor 中強大的上下文共享工具，可以彌合合作曲家和聊天互動之間的差距。將它們視為超越功能的增強參考文檔 `.cursorrules`，可讓您為開發工作流程建立可重複使用的上下文。

Notepads - 增強版 Cursor Rule

| 特性 | Cursor Notepads | Cursor Rules |
|------|---|--|
| 適用場景 | 複雜知識管理、團隊協作文檔 | 簡單 IDE 配置、基本規則 |
| 支援附件 |  支持 |  不支持 |
| 動態引用 |  支援 @ 語法 |  有限支持 |
| 最佳用途 | 深入指南與知識庫 | 簡明規則與配置 |

Notepads - 增強版 Cursor Rule

Notepads非常適合：

1. 專案架構決策
2. 開發指南和標準
3. 可重複使用的程式碼模板
4. 需要經常參考的文檔
5. 團隊特定的慣例和規則

避免使用Notepads進行以下操作：

1. 臨時筆記或草稿
2. 屬於版本控制的資訊(如 git)
3. 敏感資料或憑證
4. 高度易變且頻繁變化的訊息

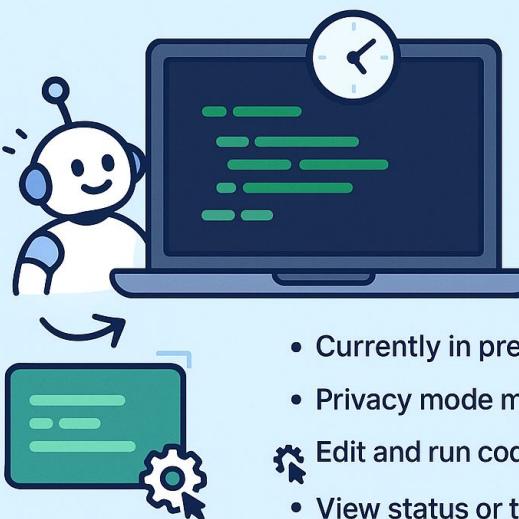
建議：

1. 使用清晰的標題和章節
2. 包括相關範例
3. 保持內容集中且井然有序
4. 使用 Markdown 格式以提高可讀性
5. 必要時新增相關文件附件

Background Agents

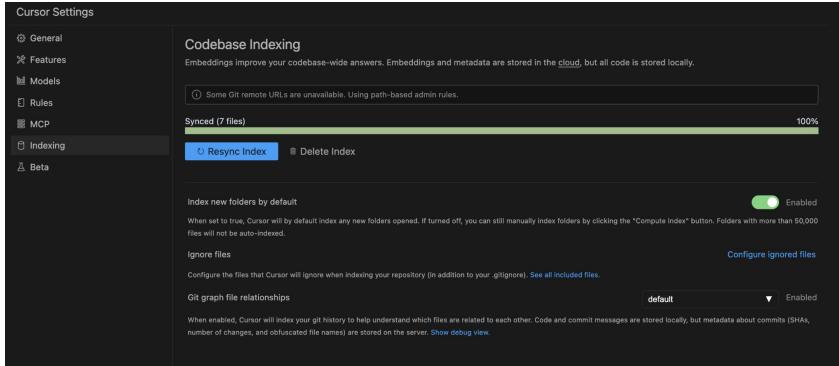
Background Agents (Preview)

How to use background agents to parallelize your work.



- Currently in preview
- Privacy mode must be off
- Edit and run code remotely
- View status or take over

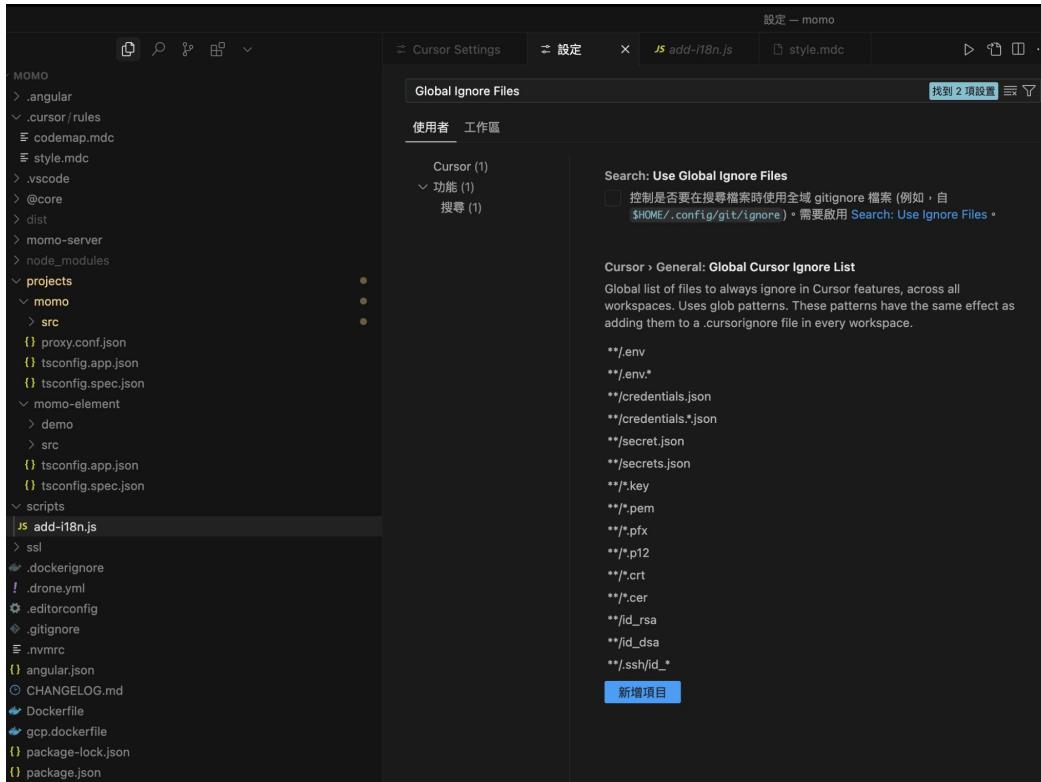
Codebase Indexing



- 為了獲得更好、更準確的程式碼庫答案，您可以對程式碼庫進行索引
- 對於專業用戶，最多可自動索引 50,000 個檔案。商業用戶的預設自動索引檔案限制更高，為 250,000 個。
- 因為Cursor基於VSCode，除了.gitignore外可用.cursorignore制定忽略檔案，如果您的專案中有任何AI絕對不需要讀取的大型內容文件，則忽略這些文件可以提高答案的準確性
- 可在see all included files內查看已被快取的檔案
- Git graph file relationships (Git 檔案關係圖)啟用後讓cursor瞭解你的git commit，以此優化Cursor 的效能和使用體驗

Cursor 聊天功能向終端機和 MCP 開發器等服務發起的工具呼叫目前無法阻止存取受下列程式碼控制的程式碼：`.cursorignore`

Global Ignore Files



腦機介面可能是 AI 的下一步



Cursor

12 Guidelines

Dec. 2024



} Hexo^α

12 Guidelines



Ryo Lu ✅ @ryolu_ · 4月22日

Using Cursor well = fast, clean code.

Using it wrong = AI spaghetti you'll be cleaning up all week.

Here's how to actually use it right:

正確使用 Cursor = 快速、無暇的程式碼。

使用錯誤 Cursor = 你將整個星期都在清理「AI義大利麵義大利麵」

這就是正確使用它的方法：

12 Guidelines

1. Set 5-10 clear project rules upfront so Cursor knows your structure and constraints. Try /generate rules for existing codebases.
2. Be specific in prompts. Spell out tech stack, behavior, and constraints like a mini spec.
3. Work file by file; generate, test, and review in small, focused chunks.
4. Write tests first, lock them, and generate code until all tests pass.
5. Always review AI output and hard-fix anything that breaks, then tell Cursor to use them as examples.
t the way.

1. 設定明確的專案規則: 在開始前設定5-10條清晰的規則，讓 Cursor 了解專案結構和限制條件。可以使用 "/generate rules" 指令為現有程式碼庫產生規則。
2. 寫出具體的提示: 詳細說明技術堆疊、期望行為和限制條件，就像在寫簡潔的規格說明一樣。
3. 逐文件工作: 按小塊產生、測試和審查程式碼，保持專注。
4. 先寫測試: 先寫測試，鎖定它們，然後產生程式碼直到所有測試通過。
5. 檢視 AI 輸出: 始終檢查 AI 產生的程式碼，修復任何問題，並告訴 Cursor 將修復後的程式碼作為範例。



12 Guidelines

6. Use @ file, @ folders, @ git to scope Cursor's attention to the right parts of your codebase.
7. Keep design docs and checklists in .cursor/ so the agent has full context on what to do next.
8. If code is wrong, just write it yourself. Cursor learns faster from edits than explanations.
9. Use chat history to iterate on old prompts without starting over.
10. Choose models intentionally. Gemini for precision, Claude for breadth.

6. 使用範圍限定: 使用 @ file、@ folders、@ git 等指令將 Cursor 的注意力限定在程式碼庫的相關部分。
7. 儲存設計文件: 將設計文件和檢查清單儲存在 .cursor/ 目錄中，使 AI Agent 能夠取得完整上下文。
8. 遇到錯誤直接修改: 如果程式碼有錯，直接自己修改比解釋更有效，Cursor 從編輯中學習更快。
9. 利用聊天歷史: 使用聊天歷史來迭代先前的提示，避免重新開始。
10. 有目的地選擇模型: 根據需要選擇不同模型，Gemini 適合精確任務，Claude 適合廣泛應用。



12 Guidelines

11. In new or unfamiliar stacks, paste in link to documentation. Make Cursor explain all errors and fixes line by line.

12. Let big projects index overnight and limit context scope to keep performance snappy.

Structure and control wins (for now)

Treat Cursor agent like a powerful junior — it can go far, fast, if you show it the way.

11. 在使用新的或不熟悉的技術棧時：使用文件鏈接，讓 Cursor 逐行解釋錯誤和修復方法。

12. 管理大型專案：讓大型專案在整夜進行索引，限制上下文範圍以保持效能流暢。

目前來說，「有結構、有控制的使用方式」會帶來最好的效果

把 Cursor 當作一個很強但需要引導的「強力初級工程師」。如果你提供明確的方向，它可以快速幫你完成很多工作。



∞

如何創造偉大的事物 與編程英雄

- Anysphere



Cursor 對於未來的 Programming 有什麼想法？



Michael Truell 麥可·特魯爾:

我們對未來充滿希望，Programming 的本質就是主導力量，利用程式碼”告訴“電腦應該做什麼，越高階程式語言越是如此，很多不理解工程的人認為，可以像在slack跟工程團隊聊天一般告訴電腦該如何工作就會開發完成，但這意味著你把很多重要的決策交給AI，對於不瞭解工程的人來說，把規格已經寫好了，工程師只要去實作，功能就開發完成了

但工程的價值在很多小決策，例如速度跟成本之間的權衡，我們希望由人類來主導決策

Cursor 對於未來的 Programming 有什麼想法？



[Arvid Lunnemark 阿爾維德·倫內馬克](#)

開發有兩種方式，一種是開發初期仔細規劃，然後在時限內完成，但我更喜歡直接動手，**快速的在10分鐘完成佈局，然後快速改進，我可能會發現這不是我要的，我要換個方法**

我想這會是開發者的未來，而且更加有趣

Cursor 對於未來的 Programming 有什麼想法？



Aman Sanger 阿曼·桑格:

"Geeks comprise about 2% of the world's population."
「極客(對科技、數學、編程等有強烈興趣的人)只占世界人口的大約 2%。」
— Donald Knuth,

但這2%的人會產生極大的貢獻

對Programming熱愛，可以產生最棒的開發者，他們享受於瞭解細節

Cursor 對於未來的 Programming 有什麼想法？



Sualeh Asif 蘇阿萊赫·阿西夫：

開發的方式將會改變，**與其說我們在打字，不如說我們在傳遞意圖給電腦**，電腦間傳輸速度將會更高效



Thank you for listening.



Session 1: 自由交流時間

- 當AI幫我寫程式了，那我要寫什麼？



Session 2: 自由交流時間

- 要寫測試還是寫意圖？



Session 3: 自由交流時間

- 現在的junior, 該學怎麼辦？



Session 4: 自由交流時間

- 我們接近科技奇點了嗎？



Session 5: 自由交流時間

- AI可以開車，可以開飛機，那 還需駕駛跟機長嗎？



Session 6: 自由交流時間

- 夢想，志向，興趣是甚麼？