# **AI 工具使用培訓手冊**

## **目錄**

1. 培訓簡介

2. AI 工具介紹TraeCursor

• Trae

• Cursor

3. 安裝與登錄Trae 安裝與登錄Cursor 安裝與登錄

• Trae 安裝與登錄

• Cursor 安裝與登錄

4. 核心功能與使用Trae 使用指南Cursor 使用指南

• Trae 使用指南

• Cursor 使用指南

5. Trae 與 Cursor 對比

6. 常見問題

## **培訓簡介**

本培訓手冊旨在幫助大家熟悉並掌握 Trae 和 Cursor 兩款 AI 輔助工具的使用方法，從而提高處理文字、文案等工作的效率。

### **學習目標**

• 瞭解 Trae 和 Cursor 的基本功能與應用場景。

• 掌握 Trae 和 Cursor 的安裝、登錄及基本操作流程。

• 理解兩款工具的主要區別，並能根據需求選擇合適的工具。

• 能夠運用所學工具輔助完成日常的文字處理和文案撰寫工作。

## **AI 工具介紹**

### **Trae**

Trae 是由字節跳動開發的新一代 AI 驅動的代碼編輯器，基於 Microsoft Visual Studio Code 構建。它提供了強大的 AI 輔助功能，包括免費訪問 Claude 3.5 Sonnet 和 GPT-4o 模型，幫助開發者更高效地編寫、調試和優化代碼。

**主要特點：**

• 基於 Visual Studio Code，提供熟悉的開發環境

• 內置強大的 AI 輔助功能

• 支持中英雙語界面

• 完全免費使用

• 支持從 VS Code 或 Cursor 遷移配置

**適用場景：**

• 代碼開發與調試

• AI 輔助編程

• 項目構建與管理

• 團隊協作開發

### **Cursor**

Cursor 是一款專爲提高編程效率而設計的 **AI 驅動的代碼編輯器**。它基於流行的 VS Code 構建，因此界面和操作邏輯對許多開發者來說非常熟悉，並且可以輕鬆導入現有的 VS Code 設置（如擴展、主題、快捷鍵）。Cursor 的核心目標是讓 AI 深度融入編碼流程，成爲開發者的智能"結對程序員"。

**主要功能與特點：**

• **強大的代碼庫理解:** Cursor 能夠理解整個項目的代碼結構和內容。通過 @Codebase 或 Ctrl+Enter 快捷鍵，你可以直接向 AI 提問關於項目的問題（例如，"這個函數在哪裏被調用了？"），AI 會智能地查找相關代碼作爲上下文進行回答。

• **自然語言驅動的開發:** 使用 Ctrl+K 快捷鍵，你可以用自然語言描述需求來生成新代碼，或者選中現有代碼塊讓 AI 進行修改、重構或解釋。

• **智能預測與補全:** Cursor 的自動補全功能 (Tab 鍵) 非常強大，它不僅能補全單行代碼，還能根據上下文和你的編輯歷史預測並建議多行代碼的修改，極大地提高了編碼速度。

• **集成 AI 聊天:** 內置的聊天窗口允許你隨時與 AI 交流。AI 能看到你當前打開的文件和光標位置，方便你進行提問、討論代碼邏輯、請求代碼示例等。聊天中生成的代碼可以一鍵應用到編輯器中。

• **上下文引用:** 在聊天或使用 Ctrl+K 時，可以通過 @ 符號輕鬆引用項目中的其他文件、代碼片段、文檔（@Docs）、甚至進行網絡搜索（@Web）來爲 AI 提供更豐富的上下文信息。

• **錯誤檢測與修復:** AI 可以自動檢測代碼中的 lint 錯誤，並嘗試提供修復建議。

• **終端智能:** 在集成的終端裏，也可以使用 Ctrl+K，用自然語言描述你想要執行的操作，AI 會幫你生成對應的終端命令。

• **隱私與安全:** 提供"隱私模式"，啓用後代碼不會被髮送到雲端進行處理，並通過了 SOC 2 安全認證。

**適用場景：**

Cursor 特別適合需要處理複雜代碼庫、希望通過 AI 加速編碼、重構、調試過程，或者想要更自然地與 AI 交互來輔助編程的開發者。對於文字工作者來說，雖然它主要面向代碼，但其強大的自然語言理解和生成能力，以及與文檔交互的功能，也可能在處理結構化文本、生成模板化內容等方面提供幫助。

## **安裝與登錄**

### **Trae 安裝與登錄**

1. **系統要求**

2. 操作系統：macOS（Windows 版本即將推出）

3. 網絡連接：需要穩定的互聯網連接

4. **安裝步驟**

5. 前往 Trae 官網，下載安裝包

6. 運行安裝程序

7. 首次啓動 Trae，點擊"開始"按鈕

8. 選擇主題（暗色、亮色、深藍）和顯示語言（簡體中文和 English）

9. **配置遷移**

10. 若已安裝 VS Code 或 Cursor，可點擊"從 VS Code 導入"或"從 Cursor 導入"按鈕

11. 一鍵導入插件、IDE 設置、快捷鍵設置等信息

12. **登錄授權**

13. 點擊"安裝 trae 命令"按鈕

14. 完成授權流程

15. 使用手機號或稀土掘金賬號登錄，以便使用 AI 服務

### **Cursor 安裝與登錄**

*在此處添加 Cursor 的詳細安裝步驟和登錄流程說明，可以配上截圖（資源放在 assets 目錄下）。*

## **核心功能與使用**

### **Trae 使用指南**

#### **1. Builder 模式**

Builder 模式是 Trae 的核心功能之一，用於從頭開始構建項目或進行大規模代碼生成。

**Windows 系統配置：**  
- 需安裝 PowerShell 6 或更高版本  
- 在 Trae 中打開"終端"面板  
- 點擊右上方區域的"+">"選擇默認配置文件"  
- 選擇帶有"\PowerShell{版本號}\xxx"字樣的配置文件

**使用步驟：**  
1. **啓用 Builder 模式**  
 - 快捷鍵：macOS（Command + U）或 Windows（Ctrl + U）  
 - 打開側邊對話框，點擊左上角"Builder"  
 - 點擊"啓用 Builder"按鈕，選擇想使用的大語言模型

1. **代碼變更處理**

2. 批量處理項目變更：點擊輸入框上方待審查文件列表中的"全部接受"或"全部拒絕"

3. 處理單個文件：在編輯器內打開文件，點擊上方提示條中的"接受"或"拒絕"

4. 處理單處變更：將鼠標懸浮至代碼變更處，點擊右上角的"^Y"或"^N"按鈕

5. **預覽功能**

6. 點擊"預覽"按鈕，在 Webview 窗口查看開發成果

7. 點擊 Webview 窗口右上角的"在瀏覽器中打開"按鈕，可在瀏覽器中預覽

8. **版本回退**

9. 找到目標對話氣泡，點擊左側的"回退"按鈕

10. 注意：回退操作不可撤銷

11. 僅支持在最近活躍的 Builder 窗口中回退

12. 僅支持回退至最近 10 輪會話內的版本

#### **2. AI 聊天功能**

Trae 提供兩種 AI 交互模式：

**側邊對話（Chat 模式）：**  
- 快捷鍵：macOS（Command + U）或 Windows（Ctrl + U）  
- 打開側邊對話框，點擊左上角"Chat"  
- 默認使用"Doubao-1.5-pro"模型  
- 可切換至 DeepSeek - Reasoner (R1) 或 DeepSeek - Chat (V3) 模型  
- AI 助手會自動創建或編輯文件並保存代碼

**內嵌對話（Inline Chat）：**  
- 快捷鍵：macOS（Command + I）或 Windows（Ctrl + I）  
- 使用方式：  
 - 在光標處使用快捷鍵  
 - 選中代碼後使用快捷鍵或點擊懸浮菜單中的"編輯"按鈕  
- 如需重新生成，可點擊對話框左下方的"重試"按鈕

#### **3. AI 代碼補全**

Trae 內置原生 AI 代碼補全功能：

• **自動補全觸發：**

• 敲擊回車鍵換行時，AI 自動分析並補全後續代碼

• 添加註釋後換行，AI 根據註釋生成相關代碼

• **接受補全建議：**

• 按下 Tab 鍵：接受所有自動補全的代碼

• 使用 Ctrl + → 快捷鍵：逐字接受代碼

#### **4. 插件管理**

**安裝插件：**  
1. **從 Trae 插件市場安裝：**  
 - 點擊左側導航欄"插件市場"圖標  
 - 搜索並選擇插件  
 - 點擊"安裝"

1. **從 VS Code 插件市場安裝：**

2. 前往 VS Code 插件市場

3. 獲取插件詳情頁信息

4. 修改特定 URL 並下載 .vsix 文件

5. 將文件拖拽至 Trae 插件市場面板

**禁用插件：**  
- 在"已安裝"列表中找到目標插件  
- 方式一：鼠標懸浮，點擊"設置">"禁用"  
- 方式二：打開詳情窗口，點擊"禁用"

**卸載插件：**  
- 在"已安裝"列表中找到目標插件  
- 方式一：鼠標懸浮，點擊"卸載"  
- 方式二：打開詳情窗口，點擊"卸載"

### **Cursor 使用指南**

以下是 Cursor 主要 AI 功能的詳細使用方法和技巧：

**1. AI 聊天 (Chat Panel -** Cmd+L**)**

這是與 AI 交互的核心區域。默認情況下，聊天面板會顯示在編輯器的側邊。

• **提問關於當前文件:** 直接在聊天輸入框中提問，AI 會基於你當前打開的文件和光標位置進行回答。例如："解釋一下這個函數的作用" 或 "這附近代碼有 bug 嗎？"。

• **提問關於整個項目 (@Codebase 或 Cmd+Enter):**

• 在聊天輸入框輸入 @Codebase 或直接按 Cmd+Enter，然後輸入你的問題，例如："項目中哪裏定義了用戶認證邏輯？" AI 會搜索整個代碼庫尋找答案。

• Cursor 會自動分析並引用相關的代碼文件作爲上下文，並在回答中展示。

• **引用特定代碼 (Cmd+Shift+L):**

• 在編輯器中選中你想要討論的代碼塊。

• 按下 Cmd+Shift+L，選中的代碼會自動添加到聊天輸入框中作爲上下文，然後你可以輸入你的問題。

• **引用文件/符號 (@):**

• 在聊天輸入框中輸入 @ 符號，會彈出一個列表，顯示項目中的文件和代碼符號（如函數、類名）。選擇一個或多個，它們將被用作 AI 回答的上下文。

• **引用文檔 (@Docs):**

• 輸入 @ 後選擇 Docs，可以搜索並引用已添加的外部文檔（例如庫的 API 文檔），或者添加新的文檔鏈接。

• **網絡搜索 (@Web):**

• 輸入 @Web 後跟上你的搜索查詢，AI 會進行網絡搜索並將結果作爲信息來源來回答你的問題，適合獲取最新的信息或外部知識。

• **使用圖片:** 點擊聊天輸入框下方的圖片按鈕或直接拖拽圖片到輸入框，可以將圖片作爲視覺上下文提供給 AI。

• **應用代碼建議:** AI 回答中的代碼塊上方通常會有一個"播放"按鈕或類似選項，點擊即可將該代碼片段應用（插入或替換）到你當前光標位置或選區。

• **新建/切換聊天:** 使用 Cmd+N 或 Cmd+R 創建新的聊天會話。使用 Cmd+[ 和 Cmd+] 在不同的聊天會話間切換。

**2. AI 編輯與生成 (**Cmd+K**)**

這是快速調用 AI 進行代碼修改或生成的主要方式。

• **編輯現有代碼:**

• 在編輯器中選中你想要修改的代碼。

• 按下 Cmd+K，會彈出一個輸入框。

• 在輸入框中用自然語言描述你的修改要求，例如："將這段代碼重構爲一個可複用的函數"，"給這段代碼添加 JSDoc 註釋"，"修復這裏的潛在錯誤"。

• AI 會生成修改建議，通常以差異對比（diff）的形式展示。你可以按 Cmd+Y 接受修改，或 Cmd+N 拒絕。

• **生成新代碼:**

• 將光標置於你想要插入新代碼的位置。

• 按下 Cmd+K。

• 在輸入框中描述你想要生成的代碼，例如："生成一個發送 HTTP POST 請求的函數，接受 URL 和 data 參數"，"創建一個 React 組件，包含一個輸入框和一個按鈕"。

• 同樣可以通過 @ 引用文件或文檔來提供上下文。

• **快速提問 (Option+Enter):** 在 Cmd+K 輸入框中，輸入問題後按 Option+Enter (macOS) 可以快速就選中的代碼或當前上下文提問，答案會直接顯示，而不是進行代碼修改。

**3. 智能自動補全 (**Tab**)**

Cursor 的 Tab 補全功能是默認開啓的，並且會持續學習你的編碼習慣。

• **接受建議:** 當編輯器中出現灰色字體的 AI 代碼建議時（可能是單行或多行），按下 Tab 鍵即可接受建議。

• **逐詞接受 (Cmd+→):** 如果你只想接受建議中的一部分，可以按 Cmd+→ (右箭頭) 來逐個單詞地接受建議。

• **預測性編輯:** Cursor 會嘗試預測你接下來可能要做的修改，並主動提供建議。例如，如果你在一個地方修改了某個變量名，它可能會自動建議在其他相關地方也進行修改。

**4. 終端中的 AI (**Cmd+K **in Terminal)**

• 在 Cursor 內置的終端中，按下 Cmd+K。

• 在彈出的輸入框中用自然語言描述你想要執行的命令，例如："查找當前目錄下所有大於 1MB 的 .log 文件"，"將遠程 main 分支的最新代碼拉取下來"。

• AI 會生成對應的終端命令。

• 你可以按 Enter 鍵運行生成的命令，或按 Esc 鍵將其接受並填充到終端輸入行，以便在運行前進行修改。

**5. AI Agent 模式**

這是一個更高級的功能，旨在讓 AI 端到端地完成更復雜的任務。

• 通常通過聊天面板或特定的命令觸發（具體觸發方式可能隨版本更新）。

• 向 AI 描述一個完整的任務目標，例如："爲這個項目添加單元測試框架，併爲 user.js 文件編寫基礎測試用例"。

• AI Agent 會嘗試分解任務、查找信息、編寫代碼、運行命令、甚至根據錯誤進行迭代。它會在執行關鍵步驟前徵求你的確認。

**使用技巧:**

• **提供清晰的上下文:** 無論是聊天還是使用 Cmd+K，提供儘可能清晰和相關的上下文（通過 @ 引用、選中代碼等）能顯著提高 AI 生成結果的質量。

• **明確指令:** 使用簡潔、明確的語言描述你的需求。避免模糊不清的指令。

• **迭代優化:** AI 可能不會一次就生成完美的結果。利用聊天進行追問，或使用 Cmd+K 對生成結果進行逐步修改。

• **善用快捷鍵:** 熟練掌握上述快捷鍵能極大提升你使用 Cursor 的效率。

• **檢查與測試:** AI 生成的代碼務必仔細檢查和測試，確保其正確性、安全性和性能符合要求。

## **Trae 與 Cursor 對比**

### **功能側重點**

**Trae：**  
- 更注重中文開發者體驗，提供完整的中文界面  
- 內置多箇中文大語言模型（如 Doubao-1.5-pro）  
- Builder 模式強調項目級代碼生成  
- 支持從 VS Code 和 Cursor 無縫遷移

**Cursor：**  
- 專注於英文開發環境  
- 深度集成 GPT-4 等英文模型  
- 強調代碼庫理解和智能補全  
- 提供更豐富的代碼分析功能

### **交互方式**

**Trae：**  
- 側邊對話（Side Chat）和內聯對話（Inline Chat）雙模式  
- Builder 模式支持項目級代碼生成  
- 支持多模態交互（圖片上傳、錯誤截圖分析）  
- 代碼變更批量處理機制

**Cursor：**  
- 聊天面板（Chat Panel）和內聯編輯  
- 強大的代碼庫搜索和理解能力  
- 支持網絡搜索和文檔引用  
- 智能終端命令生成

### **適用人羣**

**Trae：**  
- 中文開發者  
- 需要中文 AI 助手的團隊  
- 偏好項目級代碼生成的開發者  
- VS Code 用戶尋求 AI 增強

**Cursor：**  
- 英文開發環境用戶  
- 需要深度代碼理解的開發者  
- 注重代碼質量和重構的團隊  
- 需要強大文檔集成的項目

### **優缺點分析**

**Trae 優點：**  
- 完全免費，包括高級 AI 模型訪問  
- 優秀的中文支持  
- 強大的項目生成能力  
- 完善的配置遷移支持

**Trae 缺點：**  
- 目前僅支持 macOS  
- 部分功能仍在開發中  
- 插件生態相對較新

**Cursor 優點：**  
- 成熟的代碼分析能力  
- 豐富的 AI 輔助功能  
- 完善的插件生態  
- 跨平臺支持

**Cursor 缺點：**  
- 部分高級功能需付費  
- 中文支持相對較弱  
- 網絡要求較高

### **選擇建議**

1. **如果您是：**

2. 中文開發者

3. 需要免費使用高級 AI 功能

4. 主要使用 macOS

5. 偏好項目級代碼生成  
 → 推薦選擇 **Trae**

6. **如果您是：**

7. 使用英文開發環境

8. 需要深度代碼分析

9. 跨平臺開發

10. 注重生態系統完整性  
 → 推薦選擇 **Cursor**

11. **混合使用場景：**

12. 兩者可以並存使用

13. Trae 適合中文項目和快速原型開發

14. Cursor 適合深度開發和代碼維護

## **常見問題**

### **Trae 相關問題**

1. **Trae 是否完全免費？**

2. 是的，Trae 目前完全免費，包括訪問 Claude 3.5 Sonnet 和 GPT-4o 模型的功能。

3. **Windows 版本何時發佈？**

4. Windows 版本正在開發中，具體發佈時間待官方通知。

5. **如何提高 AI 響應質量？**

6. 提供清晰的上下文

7. 使用具體的示例

8. 分步驟提出需求

9. 及時提供反饋

10. **如何處理代碼安全問題？**

11. 不要上傳敏感代碼

12. 使用環境變量管理密鑰

13. 定期檢查權限設置

14. 遵循安全最佳實踐

15. **如何處理 AI 生成的代碼？**

16. 仔細審查生成的代碼

17. 進行必要的測試

18. 確保符合項目規範

19. 適當修改和優化

### **Cursor 相關問題**

1. **Cursor 是否完全免費？**

2. 是的，Cursor 目前完全免費，包括訪問 GPT-4 等英文模型。

3. **如何提高 AI 響應質量？**

4. 提供清晰的上下文

5. 使用具體的示例

6. 分步驟提出需求

7. 及時提供反饋

8. **如何處理代碼安全問題？**

9. 不要上傳敏感代碼

10. 使用環境變量管理密鑰

11. 定期檢查權限設置

12. 遵循安全最佳實踐

13. **如何處理 AI 生成的代碼？**

14. 仔細審查生成的代碼

15. 進行必要的測試

16. 確保符合項目規範

17. 適當修改和優化

18. **如何提高智能補全質量？**

19. 提供清晰的上下文

20. 使用具體的示例

21. 分步驟提出需求

22. 及時提供反饋

23. **如何處理終端中的 AI 命令？**

24. 仔細審查生成的命令

25. 進行必要的測試

26. 確保符合項目規範

27. 適當修改和優化

28. **如何提高 AI Agent 模式的效果？**

29. 提供清晰的上下文

30. 使用具體的示例

31. 分步驟提出需求

32. 及時提供反饋

33. **如何處理 AI 生成的代碼？**

34. 仔細審查生成的代碼

35. 進行必要的測試

36. 確保符合項目規範

37. 適當修改和優化

38. **如何提高智能補全質量？**

39. 提供清晰的上下文

40. 使用具體的示例

41. 分步驟提出需求

42. 及時提供反饋

43. **如何處理終端中的 AI 命令？**

• 仔細審查生成的命令

• 進行必要的測試

• 確保符合項目規範

• 適當修改和優化