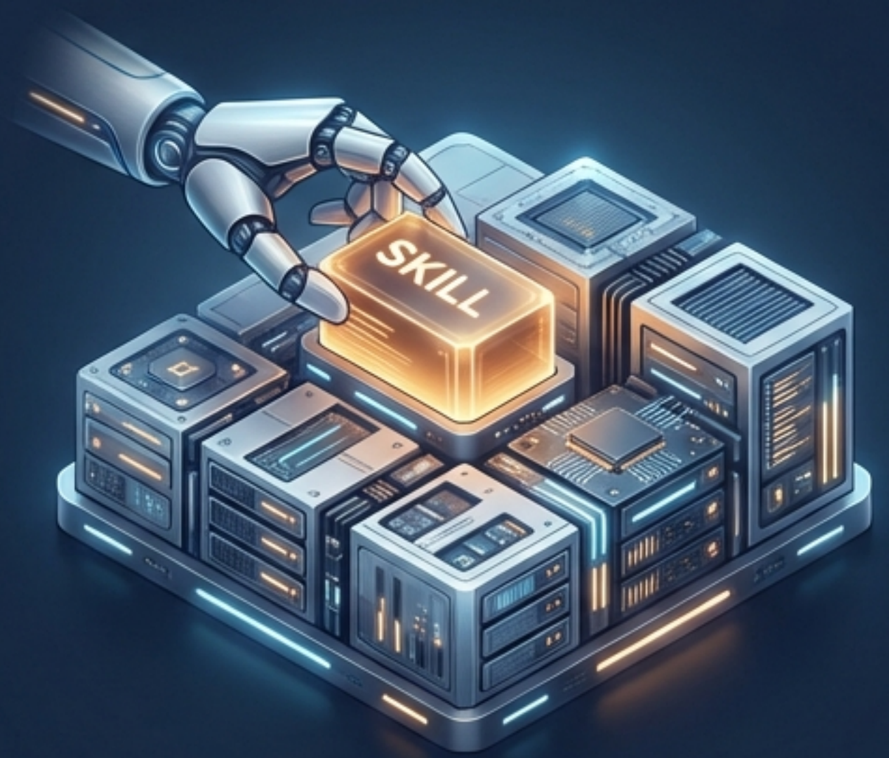
Agent Skills：打造模組化 AI 專業技能包

讓 AI 從「通用助理」進化為「專業工匠」



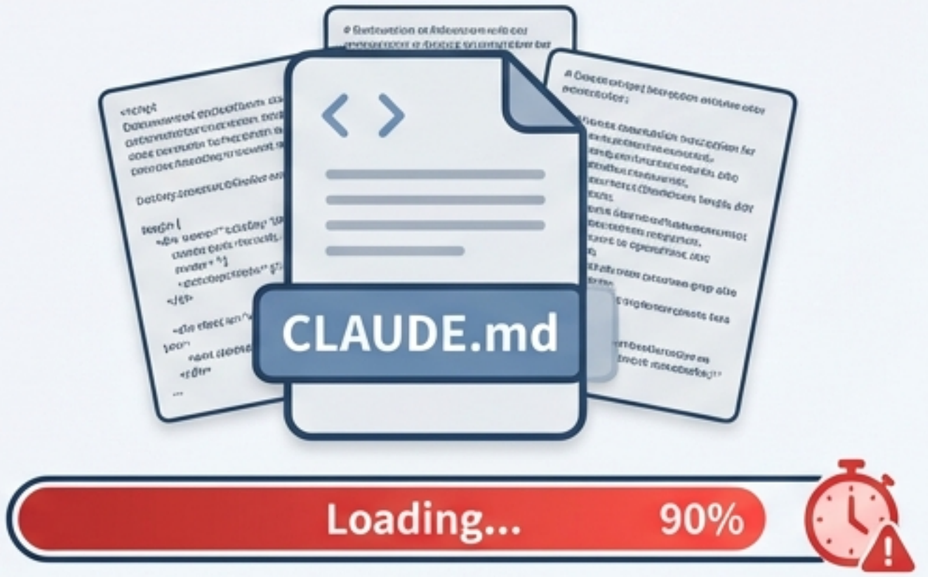
基於高見龍（Eddie Kao）文章與 Anthropic 官方規格 | 技術核心：Progressive Disclosure

痛點：每次開新專案，AI 都像剛入職的失憶新人？

The User Experience

高重複溝通成本

The System Reality



Context 肥大化與效能瓶頸

重複溝通：
每次都要重講 Coding Style 與流程。

Context 消耗：
一次性載入所有規則，浪費 Token。

維護困難：
單一檔案過長，難以管理。

解法：賦予 AI 專業領域的「武功秘笈」

什麼是 Agent Skills?

Skills 是模組化、獨立封裝的「**專業技能包**」。它包含**專業知識** (Knowledge)、**工作流程** (Workflows) 與**工具** (Tools)。

Application Scenarios

- Code Review 標準流程
- 特定檔案處理 (e.g., PDF)
- 團隊內部的商業邏輯

核心機制：Progressive Disclosure（漸進式揭露）

打破 Context Window 限制：用多少，拿多少

Metadata (Pre-load)
name, description

~100 tokens

Instructions (On Demand)
Rules, Workflows

< 5000 tokens

Resources (Runtime)
Python scripts, API docs

Unlimited

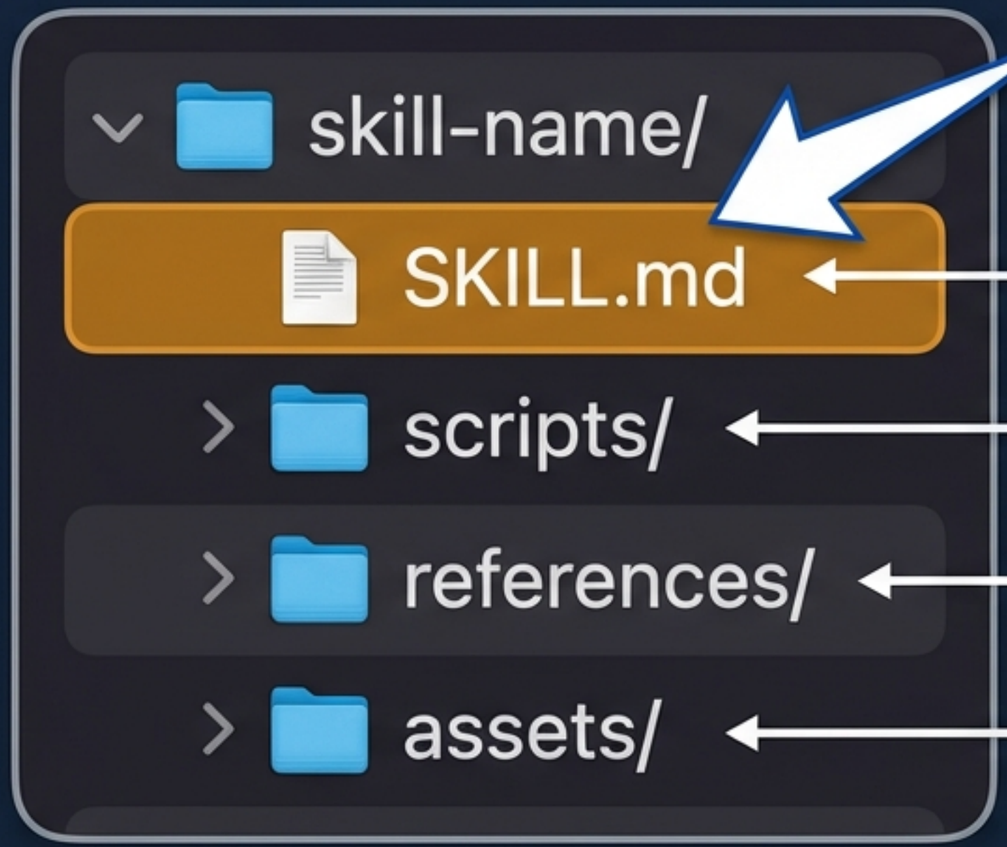
理論上可打包無限量的知識，但只佔用極少的 Context Window。

超級比一比：Skills vs. MCP vs. Commands

Skills (大腦/知識)	MCP (手腳/工具)
 **Model-invoked (自動判斷)** 提供「知識」與「流程」指導行為。跨專案通用。	 **Tool Calling (按需呼叫)** 連接外部 API、資料庫、操作系統。
Custom Commands (巨集/SOP)	**Subagents (分身)**
 **User-invoked (手動 /xxx)** 固定 Prompt 範本，適合重複性操作。	 **Autonomous (獨立)** 擁有獨立 Context，處理複雜平行任務。

Skills 給 AI 腦袋，MCP 給 AI 手腳。兩者是互補而非取代。

解剖 Skill：標準化目錄結構



目錄名稱必須與 name 一致

核心：必要的指令與 Frontmatter

選用：Python, Bash, JS 執行腳本

選用：額外的參考文件

選用：範本、圖片

靈魂核心：SKILL.md 與 Metadata

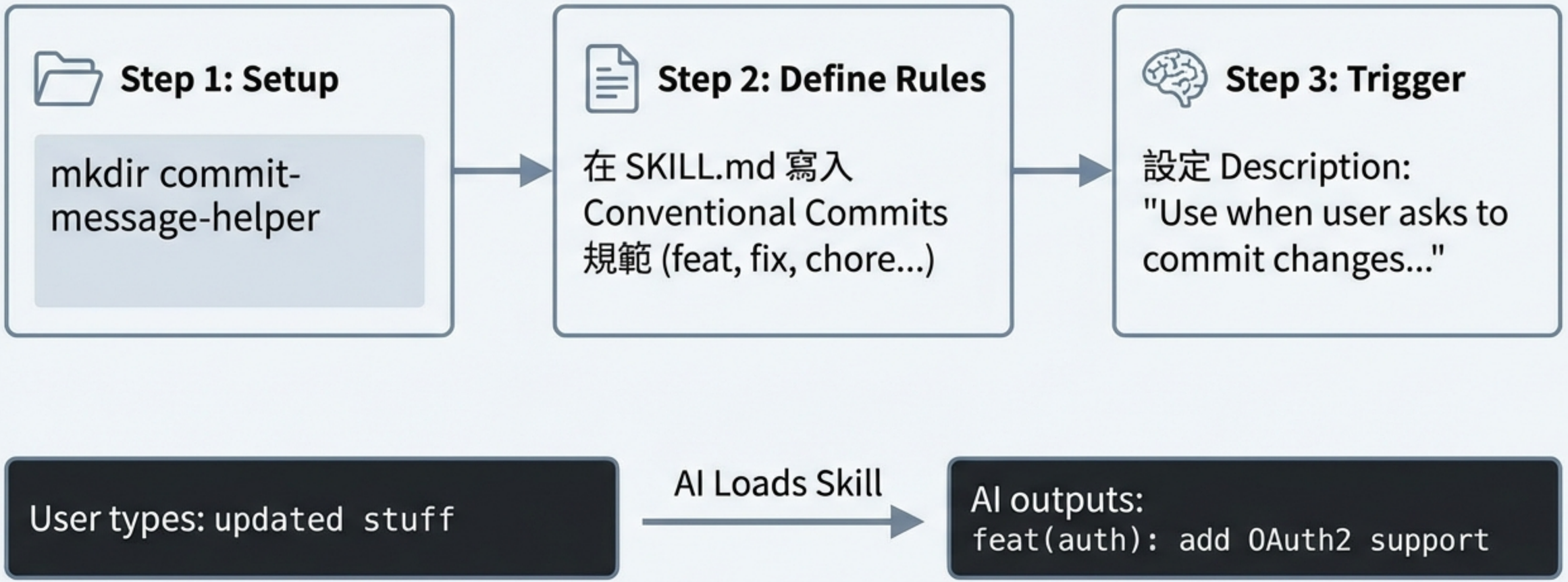
```
---
name: commit-message-helper
description: Helps write Git commit
messages following Conventional
Commits specification. Use this skill
when user asks to commit changes...
---
# Instructions start here...
```

1-64 字元。僅限小寫、數字、短橫線。

關鍵！
1-1024 字元。這是 AI 決定「何時喚醒」此技能的唯一依據。

Tip: Description 必須包含「功能」與「使用時機」。

實戰演練 I：打造 Commit Message 助手 (知識篇)



實戰演練 II：加入驗證腳本（能力篇）

從「靜態建議」升級為「動態執行」

SKILL.md

```
...
---
Run validation: `python
scripts/validate_commit.py`
---
...
```

scripts/validate_commit.py

```
pattern = r'^(feat|fix|docs|..): .{1,50}'
if re.match(pattern, message): ...
```

Tip: AI 不只幫你寫，還會自動跑測試驗證格式正確性。

最佳實踐：如何寫出 AI 看得懂的 Description

Description 不是給人看的行銷文案，是給 AI 的技術指引

❌ Bad (太模糊)	✅ Good (明確觸發條件)
`description: Helps with PDFs.`	`description: Extracts text from PDFs.` `Use when user mentions PDF processing or document extraction.`
AI 不知道何時該用，或隨便亂用。	精準命中任務需求。

💡 **Pro Tip**：不知道怎麼寫？用魔法對付魔法——請 AI 幫你寫 Description。

最佳實踐：結構、範例與邊界情況

Few-Shot Prompting

提供範例：Input/Output 對照，讓 AI 模仿。

```
Ex: `Input: "fixed bug" -> Output: "fix: resolve NPE"`
```

References

善用 References：主檔保持 < 500 行，細節放入 references/。

Edge Cases

定義模糊指令的處理方式 (e.g., Ask for clarification)。

安全優先：防禦惡意技能

⚠ Skills 可以執行本機程式碼，請保持警覺。

- **來源審查**：只安裝受信任來源 (Trusted Source) 的 Skills。
- **代碼檢查**：務必檢查 scripts/ 內的每一行程式碼。
- **Human in the Loop**：Claude 載入 Skill 前會詢問使用者許可，不要盲目按 Yes。

生態系：Agentskills.io 開放標準

Phase 1:
Chatbot
(一問一答)

Phase 2 :
Tool Calling
(使用工具)

Phase 3 : Skills
數位工匠 -
知識 + 流程 + 工具

ANTHROPIC

VS Code

Community Callout

GitHub Copilot
(支援 Agent Mode)

Kevin Tseng
(.NET Testing Skills) - 2025 iThome 鐵人賽冠軍

總結：我該使用哪種工具？

Skills

...AI 自動判斷何時使用知識？

Custom Commands

...手動觸發固定 SOP？

你需要...

MCP

...連接外部 API/資料庫？

Subagents

...複雜平行任務？

Power Combo: Command 觸發 Skill + MCP 抓取資料！

結語：為你自己學，打造數位分身

- Tools 再強，核心還是你腦中的知識 (Domain Knowledge)。
- Skills 讓你將知識「模組化」，讓團隊共享資深工程師的 Context。

整理你的 CLAUDE.md　　建立你的第一個 Skill　　成為真正的數位工匠