

為什麼把資料存進知識庫，不等於擁有知識？

庫裡的 Obsidian 入門筆記講的是「怎麼用」——卡片怎麼寫、[[]] 怎麼牽手；學習紀錄記的是「做了什麼」。但兩邊都沒回答一個更根本的問題：**什麼東西值得留下來，而且誰來決定？** 這張卡補上這個取捨標準，並把它跟 AI 協作的分工接起來。

這段心得的骨架是什麼？

原文的骨架是誤解、標準、分工、決定權：

層次	內容
誤解	以為知識庫就是把資料一直存進去，存得越多越好
標準	只留兩種內容：真正理解過的、以後還能再使用的
分工	AI 負責整理格式、建立連結、存檔——這些是「搬運與排版」的事
決定權	最後要留下什麼判斷，由自己決定，不外包

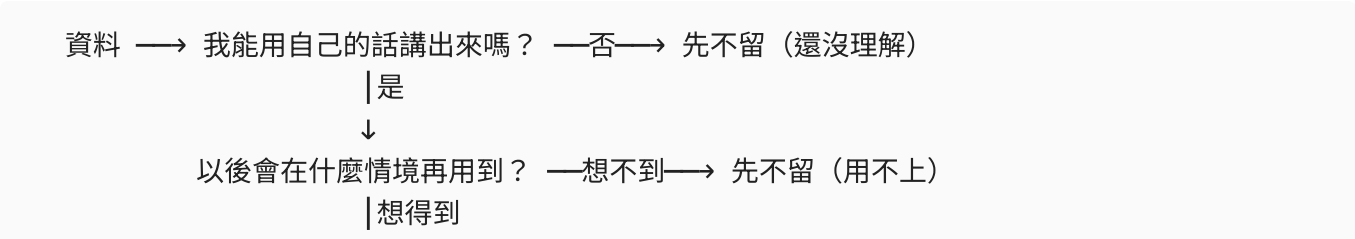
（這段心得沒有類比，所以沒有硬套「類比」這一層；四個角色是照原文的實際走向命名的。）

關鍵判斷：存進去只是動作，留下來才是結果。這段話把「知識庫」的定義從**容量**改成**篩選**——衡量它的不是裡面有多少東西，而是每一樣東西有沒有通過「我懂了」和「我還會用」這兩道關。

觀點一：知識庫的價值在篩選，不在容量

「知識庫不是把資料一直存進去，而是要把真正理解過、以後還能再使用的內容留下來。」

這句的鋒利處在於它給了兩個**可以檢驗**的門檻，而不是一句「要用心整理」的口號。「理解過」檢查的是過去：你能不能不看原文、用自己的話講出來？「以後還能再使用」檢查的是未來：想像三個月後的某個情境，這條筆記會被翻出來嗎？只通過其中一關的，前者是「懂了但用不上的知識」，後者是「還沒懂就先備著的資料」，都不必急著進庫。



↓
留下 → 知識庫

⚠ 這條反過來讀更有用

存得越多，越容易產生「這些我都學過了」的錯覺——資料躺在庫裡，人卻沒有真的理解它。所以一則筆記如果講不出自己的話、想不到會在哪裡再用，它就只是資料，不是知識，而且還會稀釋庫裡真正有用的部分。

觀點二：AI 代勞格式，判斷留給自己

「AI 可以幫我整理格式、建立連結和存檔，但最後要留下什麼判斷，應該由我自己決定。」

這句的鋒利處在於它把 AI 的工作**切成兩層**：格式、連結、存檔屬於「搬運與排版」，交出去沒有損失，還省下很多力氣；但「留下什麼判斷」屬於**取捨**——取捨的標準正是觀點一那兩道關（我理解了嗎、我會再用嗎），而這兩道關的答案只存在自己腦中，AI 看不到。所以能交給 AI 的，是把已經決定要留的內容整理好；不能交的，是「要不要留」這個決定本身。

資料 → [AI] 整理格式 → [AI] 建立連結 → [AI] 存檔
↓
[自己] 這條我理解嗎？還會用嗎？
↓
留下 / 丟掉

💡 這條反過來讀更有用

如果連「要留什麼」都交給 AI 決定，庫裡累積的就會是 AI 的判斷，不是你的。連 AI 自動建立的連結也一樣：你得說得出為什麼這兩張卡該連在一起，否則那條連結也只是被存下來的資料。

往外接到哪

你講的	學名	誰提的	它多講了什麼
只留「真正理解過」的內容	處理深度（Levels of Processing）；自我生成效應（Generation Effect...）	Fergus Craik 與 Robert Lockhart, 1972；Norman Slamecka 與 Peter Graf, 1978	記憶保留的好壞取決於處理的深度（意義層次勝過表面層次），而且自己生成的內容比單純讀過的記得更牢——這說明「用自己的話講出來」不只是檢查手段，本身就是讓知識留下來的機制

你講的	學名	誰提的	它多講了什麼
知識庫 不是把 資料一 直存進 去	收集者謬誤 (Collector's Fallacy)	Christian Tietze， zettelnkasten.de，約 2013； 後來被 Sönke Ahrens， 2017，《How to Take Smart Notes》引用推廣	指出收集資料會帶來「已經 在學習」的錯覺——收集本 身讓人感覺有進展，但沒有 經過理解與重組就沒有知識 產生，這正是「存越多越安 心」的心理來源
格式、 連結交 給 AI， 判斷自 己留	認知卸載 (Cognitive Offloading)	Evan Risko 與 Sam Gilbert，2016，《Trends in Cognitive Sciences》；Betsy Sparrow 等人，2011， 《Science》	卸載能提升當下任務的表 現，但被卸載的內容通常記 得比較差，人傾向只記「去 哪裡找」而不是內容本身 ——這給了「哪些可以卸 載」一個邊界：搬運與排版 可以，需要內化的理解與判 斷不行

邏輯連結

- 垂直：[Obsidian入門筆記-給小寶寶看的](#)（那張講 Obsidian 是「只放你自己的想法」的百寶箱，這張補上它沒說的取舍標準：什麼樣的想法才值得放進去）
- 橫向：[反向拆解知識卡樣板-用具體問題追出隱性規則](#)（那張示範「自己歸納出規則才算真懂」，正是觀點—「理解過」門檻的實作案例；同時也是 AI 只給素材、判斷由人做的分工範例）