



TeacherOS × 專案三層記憶系統指引手冊

交付物清單與目標受眾

目標受眾

以專案式/程式式教學為主、希望用多套 AI 工具（如 ChatGPT¹、Gemini²）長期累積備課資產的教師（尤其偏向敘事、跨領域、差異化教學者）。

本手冊交付物

- 一份可執行的操作手冊（繁中），含：執行摘要、背景與目標
- 三層記憶模型（TeacherOS / Project / Working）詳細說明與視覺圖（ASCII + mermaid）
- 每日工作流程（載入→協作→更新→存檔）步驟清單 + mermaid 流程圖
- 檔案與資料夾建議結構（含範例路徑）
- 比較表格：YAML vs Markdown、Layer2 vs Layer3、網站/PDF/Word 輸出來源
- 附錄：可直接複製的範本檔案（TeacherOS.yaml、ProjectTemplate.yaml、ProjectMinimal.yaml、子 YAML、lesson.md、manifest、publish/build.yaml）
- 實務建議：何時建立 lesson.yaml、何時只更新 Markdown、請 AI 更新 YAML 的 Prompt 範例

執行摘要

你要建立的不是「把所有資料塞進同一個筆記本」的系統，而是一套可移植、可重複載入、可出版的教師工作流：

- **YAML 管結構**：讓 AI 每次「秒懂你是誰、專案在哪、今天要推進哪條工作線」。YAML 是資料序列化格式，適合存「狀態與邏輯骨架」；語法核心是縮排、清單、映射（key/value）。³
- **Markdown 管內容**：真正的備課輸出（教案文字、故事稿、活動流程、講義）。
- **三層記憶**：TeacherOS（不常變）→ Project（專案生命線）→ Working（工作線分流）。
- **打包輸出**：用 manifest 指定「要輸出的檔案清單與章節順序」，再用網站生成器/轉檔工具輸出成網站或文件；例如 MkDocs⁴ 以 Markdown 為來源生成網站，配置檔用 YAML。⁵；Pandoc⁶ 可把 Markdown 轉成 PDF/Word（docx）。⁷

背景與目標

你在對話中已釐清兩個關鍵現實：

- 1) 多個 AI 工具之間通常不會自動共享同一份背景；要靠你用「可再載入的檔案」把上下文帶進去。
- 2) 若把所有內容都塞進單一資料容器，會造成 Context 爆炸、重點模糊、推理品質下降。

因此你的目標應是：

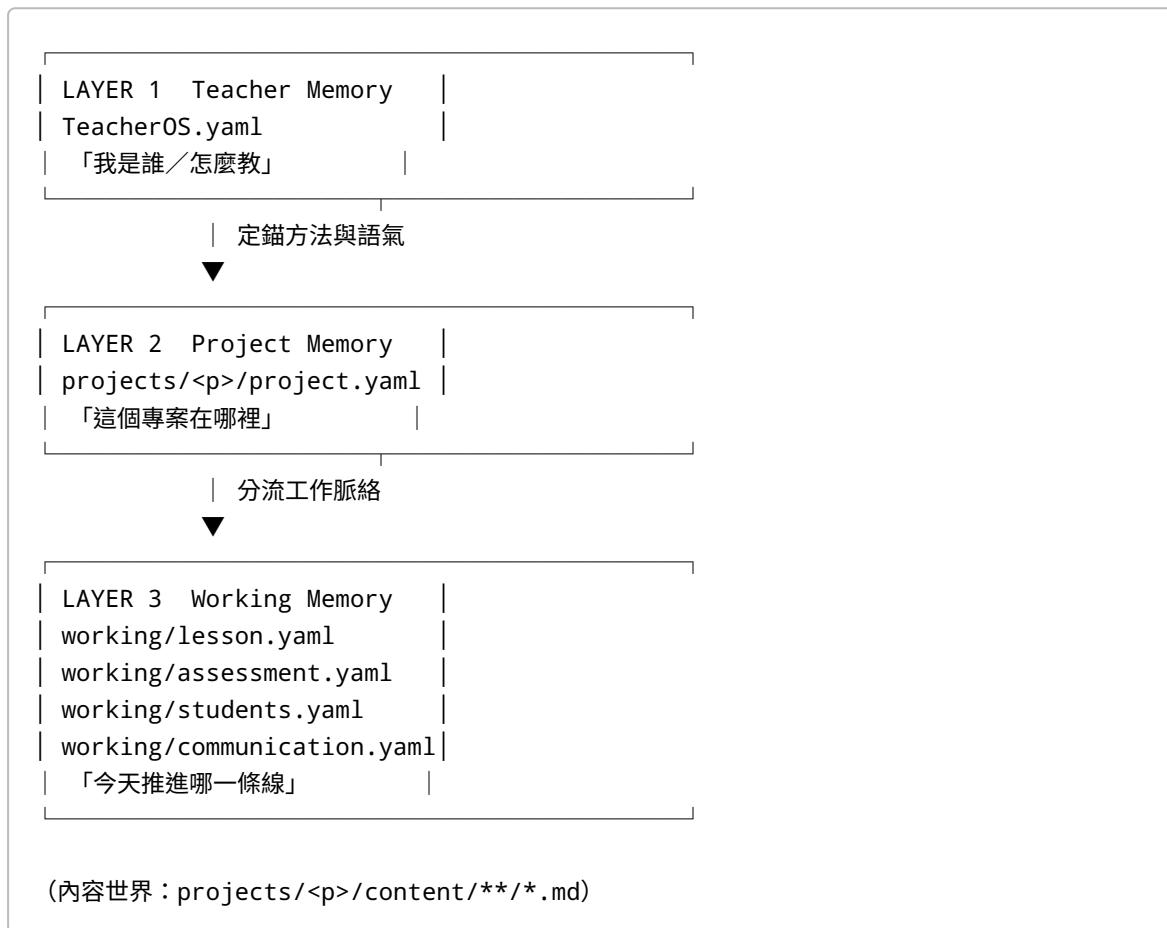
- 用 TeacherOS.yaml 固定「你作為教師的教學哲學與偏好」
- 用 project.yaml 固定「該專案的意義、現況、下一步」
- 用 working/*.yaml 分流「課程線、評量線、學生觀察線、家長溝通線」
- 用 content//.md 存「真正備課內容」
- 用 manifests/.yaml + publish/build.yaml** 讓專案能被「一鍵打包輸出」成網站/PDF/Word

補充：YAML 內容可用中文（含 key 也可），因 YAML 規格支援 Unicode；雙引號字串亦可用 Unicode 跳脫序列。⁸ 但為了跨工具解析穩定，你已採用「Key 英文、Value 中文」的策略。

三層記憶模型

三層模型視覺圖

ASCII 版（三層記憶）



mermaid 版（三層 + 內容世界）

```
graph TD  
    A[Layer1: TeacherOS.yaml<br/>教師人格/理念/輸出偏好] --> B[Layer2:  
    project.yaml<br/>專案目的/狀態/next action]  
    B --> C[Layer3: working/*.yaml<br/>lesson/assessment/students/  
    communication]  
    C --> D[content/**/*.md<br/>教案/故事/講義/反思（真正輸出）]  
    B --> D
```

Layer 1 : TeacherOS.yaml

用途：定義「你是怎樣的老師」與「AI 該如何協助」。它像作業系統設定檔，**少改、長用**。
(YAML 作為資料序列化格式，適合持久保存結構化設定。⑨)

放什麼：教學哲學、年級範圍、常用輸出格式、差異化矩陣、回應風格。

Layer 2 : Project (projects/<p>/project.yaml)

用途：專案生命線：目的、背景限制、當前焦點、下一步。它是 AI 下次接續工作的「起點」。

關鍵原則：Project.yaml 是「狀態」，不是「完整報告」。先用 Minimal 版啟動，再滾動式增補。

Layer 3 : Working (projects/<p>/working/*.yaml)

用途：把同一個專案內的不同工作線拆開，避免混線。

- lesson.yaml : 教學階段理解（不是每堂課一份）
- assessment.yaml : 證據來源、rubric 維度、評量敘述規則
- students.yaml : 班級能量、困難點、成長跡象（供回饋/評語）
- communication.yaml : 家長訊息主軸、疑慮、溝通排程

只在需要時才出生：當你開始反覆做某類任務（例如一直在寫家長信），就開一個對應的 sub-yaml。

YAML 與 Markdown 的分工表

類型	放什麼	典型檔案	何時更新
YAML (結構/狀態)	核心意圖、進度、下一步、規則、分類	TeacherOS.yaml、project.yaml、working/*.yaml	每次工作結束 「更新狀態」
Markdown (內容/輸出)	教案文字、故事全文、活動細節、講義、板書、反思	content/*.md	每次產出可直接使用的備課內容

(MkDocs 以 Markdown 作為來源文件、用 YAML 設定整體配置，是「YAML 管結構、Markdown 管內容」的典型工具鏈。⑤)

每日工作流程

標準步驟清單

- 1) 載入：丟 TeacherOS.yaml + project.yaml (以及今天要推進的 working/.yaml)
- 2) 協作：自由發散設計課程/評量/家長信，產出 Markdown 內容
- 3) 更新：收工前請 AI 「只更新變動區塊」到 YAML (不要重寫全部)
- 4) 存檔*：把更新後 YAML 與新增/修改的 Markdown 存回資料夾 (建議版本控管)

mermaid 流程圖

```
graph TD; S[開始工作] --> L[載入<br/>TeacherOS.yaml + project.yaml (+所需working.yaml)]; L --> W[協作產出<br/>content/*.md]; W --> U[請AI更新YAML<br/>僅更新變動區塊]; U --> A[存檔回Repo<br/>project.yaml/working.yaml + md]; A --> N[下次從同狀態續推]
```

檔案與資料夾建議結構

推薦專案 Repo 架構

```
TeacherOS-Repo/
├── ai-core/
│   └── teacheros.yaml
├── projects/
│   └── farm-internship/
│       ├── project.yaml
│       ├── working/
│       │   ├── lesson_01.yaml
│       │   ├── assessment.yaml
│       │   ├── students.yaml
│       │   └── communication.yaml
│       ├── content/
│       │   ├── overview.md
│       │   ├── lessons/day01.md
│       │   └── stories/story_01.md
│       ├── assets/
│       └── exports/
└── manifests/
    └── farm-internship.yaml
└── publish/
    ├── build.yaml
    └── templates/
```

打包輸出邏輯

- **manifests/<project>.yaml**：定義「輸出菜單」 = 章節順序 + 要包含哪些 md/yaml
- **publish/build.yaml**：定義「出版規則」 = 網站導覽、PDF 封面、Word 模板等
- **工具層**：
 - 網站：MkDocs（或同類靜態站）讀 Markdown 生成 HTML。⑩
 - 文件：Pandoc 把 Markdown 轉 PDF/Word（docx），也支援參考模板。⑪

輸出格式對應來源表

目標輸出	主要來源	控制檔	常見工具
網站 (HTML)	content/*.md + assets	publish/build.yaml、 manifests/*.yaml	MkDocs (Markdown→HTML) ⑫
PDF	content/*.md	publish/build.yaml (PDF 參 數)	Pandoc (可輸出 PDF) ⑬
Word (docx)	content/*.md	publish/build.yaml (reference docx)	Pandoc (Markdown→docx) ⑭

補充：若你在 Markdown 檔案頂部加「front matter」（YAML metadata），很多文件/網站系統都能讀取並注入頁面屬性；例如 Jekyll、Docusaurus、MkDocs 均支援以 `---` 包住的 YAML front matter。⑮

實務建議與常用 Prompt

何時建立 lesson.yaml

lesson.yaml 代表「教學階段」，不是「單堂課」。建立新 lesson.yaml 的三個訊號：

- 教學意圖改變：從「適應勞動」轉到「理解責任」
- 教學策略轉向：從故事導入改成學生自主/實作主導
- 班級狀態質變：能量、抵抗、合作氛圍出現明顯轉折

若只是換活動、換故事、內容細節調整：只更新 Markdown。

何時只更新 Markdown

- 你今天多寫了一段故事全文、增加了講義題目、調整了活動流程細節
- 但「核心目標/階段/下一步教學動作」沒有變

這些都屬內容層，放 `content/**/*.md`。

請 AI 更新 YAML 的 Prompt 範例

Prompt A：收工時更新 Project (Minimal)

請根據我們今天的討論，更新 `project.yaml`，只更新以下區塊：`current_focus`、`next_action`。不要改動其他欄位。輸出為完整 YAML（可直接覆蓋檔案）。

Prompt B：蒸餾 lesson.yaml (只留結構理解)

請把今天的課程設計內容「蒸餾」成 `lesson_01.yaml`：

- 1) 不要包含故事全文或活動逐字稿
- 2) 只保留：`theme`、`developmental_focus`、`current_phase`、`core_pedagogical_move`、`student_response`、`next_teaching_move`
- 3) 用中文 value、英文 key

Prompt C：從對話生成 lesson.md

請把我們今天定案的教案輸出成 `content/lessons/day01.md`，包含：課程目標、材料、流程（分段+時間）、教師講述重點、學生任務、收束反思提問。最後加一段「備忘」給我（教師用）。

何時需要 front matter

當你準備輸出成網站/文件並需要頁面 metadata（標題、標籤、年級、受眾）時，在 `.md` 檔案頂部加 YAML front matter。這是多個文件系統的共同慣例。[14](#)

圖表與流程圖語法提醒

本手冊用的 `mermaid` 流程圖屬於文字定義圖，適合在文件系統中維護並避免「圖表腐爛 (doc-rot)」；Mermaid 官方提供 flowchart 語法與範例。[15](#)

附錄：可複製範本檔案

以下區塊可直接複製成檔案使用。提醒：YAML 以縮排表達層級；序列用 - 。 9

TeacherOS.yaml

```
teacher_identity:  
  name: David  
  role: 華德福教師  
  location: 台灣  
  school_type: 體制外華德福實驗學校
```

```
teaching_scope:  
  grade_focus:  
    - 七年級  
    - 八年級  
    - 九年級  
  subjects:  
    - 英文  
    - 人文與社會  
    - 歷史  
    - 跨領域主題課程
```

```
pedagogical_framework:  
  philosophy: 華德福教育  
  developmental_focus:  
    - 思考  
    - 情感  
    - 意志  
  core_values:  
    - 經驗式學習  
    - 藝術融入教學  
    - 人格與道德發展  
    - 長期生命成長
```

```
teacher_working_style:  
  input_method: 以語音輸入為主  
  thinking_style: 系統建構型思考  
  design_preference:  
    - 專案式學習  
    - 敘事導向課程  
    - 跨學科整合  
    - 長期主課程設計
```

```
ai_collaboration_goal:  
  purpose:  
    - 減少教師行政與備課負擔  
    - 建立差異化教學系統  
    - 生成可重複使用教材  
    - 建立長期可持續工作流
```

```

differentiated_learning_model:
student_matrix:
  A: { ability: 高, motivation: 高 }
  B: { ability: 低, motivation: 高 }
  C: { ability: 高, motivation: 低 }
  D: { ability: 低, motivation: 低 }

communication_preferences:
language:
  - 繁體中文
  - 英文
  - 日文學習中
response_style:
  - 結構清晰
  - 簡潔
  - 可直接操作

```

ProjectTemplate.yaml

```

project_meta:
  project_id: "YYYY-TERM-SEQ"
  name: "專案名稱"
  type: "課程/主課程/專題/旅行/活動/戲劇"
  status: "規劃中/進行中/收尾中/已完成"
  owner: "David"
  created_date: "YYYY-MM-DD"
  last_updated: "YYYY-MM-DD"

context:
  grade:
    - "九年級"
  term: "春季/夏季/秋季/冬季"
  duration:
    start_date: "YYYY-MM-DD"
    end_date: "YYYY-MM-DD"
  weekly_rhythm: "每週__次／每次__分鐘"
constraints:
  - "時間限制："
  - "場地限制："
  - "資源限制："

purpose:
  why_now: "為什麼此刻要做這個？"
growth_meaning:
  - "成長意義1"
learning_meaning:
  - "學習意義1"
success_criteria:
  - "完成標準1"

```

```
plan:  
phases: []  
  
assessment:  
approach: "形成性/總結性/混合"  
evidence_sources:  
- "觀察紀錄"  
- "作品/文本"  
- "口頭分享"  
  
communication:  
audiences:  
- "學生"  
- "家長"  
channels:  
- "口頭"  
- "紙本"  
- "LINE"  
  
ai_integration:  
primary_use_cases:  
- "教案生成"  
- "評量工具"  
- "家長通知"  
inputs_to_provide_each_session:  
- "TeacherOS.yaml"  
- "本專案project.yaml"
```

ProjectMinimal.yaml

```
project:  
name: 專案名稱  
type: 主課程/活動/戲劇/實習  
status: planning  
  
context:  
grade:  
- 九年級  
time_frame: 春季學期  
  
purpose:  
growth_meaning:  
- 成長意義  
learning_meaning:  
- 學習經驗  
  
current_focus:  
this_week_goal: 本週正在推進什麼  
current_problem: 現在卡住的地方
```

```
next_action:  
  - 下一步1  
  - 下一步2
```

子 YAML 範例

working/lesson_01.yaml

```
lesson_state:  
  theme: 勞動與責任  
  developmental_focus:  
    - 意志  
    - 身體經驗  
  current_phase: 導入階段  
  core_pedagogical_move:  
    - 先以故事建立內在動機  
    - 再進入實際勞動  
  student_response:  
    - 初期抗拒  
    - 開始產生投入  
  next_teaching_move:  
    - 加入觀察任務  
    - 引導反思書寫
```

working/assessment.yaml

```
assessment:  
  observation_focus:  
    - 工作持續度  
    - 團隊合作  
  evidence:  
    - 教師觀察  
    - 學生反思  
  rubric_dimensions:  
    - 投入度  
    - 完成品質  
    - 合作承擔  
    - 反思深度
```

working/students.yaml

```
class_status:  
  energy: 疲累但投入  
  challenges:  
    - 部分學生逃避勞動
```

```
growth_signs:  
- 開始互相協助
```

working/communication.yaml

```
parent_communication:  
current_message: 說明實習的教育意義與安全措施  
concerns:  
- 家長擔心安全  
- 家長擔心影響學科進度  
next_update_date: "YYYY-MM-DD"
```

lesson.md 範例

```
---
```

title: Day 1 | 進入農場：勞動與責任
type: lesson
audience: teacher
grade: 9
tags: [農場實習, 意志]

```
---
```

課程目標
- 讓學生在身體勞動中「進入情境」，形成初步責任感。
- 建立觀察任務：我如何工作？我如何與他人合作？

材料
- 工作手套、筆記本、簡易工具（依場域）
- 反思提問卡（教師自製）

流程（90分鐘）
1. 開場與定錨（10'）：今日主題一句話 + 安全提醒
2. 故事導入（15'）：短故事聚焦「承擔」與「選擇」
3. 現地勞動（45'）：分工 + 教師巡視觀察
4. 收束反思（20'）：三題反思書寫 + 一輪口頭分享

教師講述重點
- 勞動不是懲罰，而是讓我看見自己的能力與限制。
- 你承擔的那一小塊，會支撐整個共同體。

學生任務
- 完成觀察紀錄：我今天最想逃避的是什麼？我怎麼跨過去？

備忘（教師用）
- 特別留意：逃避勞動的學生是否需要更明確的「可完成小步驟」。

(front matter 的 YAML 區塊以 `---` 包住並置於檔案最前方，是多個文件系統採用的慣例。 13)

manifests/farm-internship.yaml 範例

```
project: farm-internship
title: 農場實習（九年級）
outputs:
  site: true
  pdf: true
  docx: false

include_context:
  - ai-core/teacheros.yaml
  - projects/farm-internship/project.yaml
  - projects/farm-internship/working/lesson_01.yaml

chapters:
  - title: 專案總覽
    files:
      - projects/farm-internship/content/overview.md
  - title: 課程設計
    files:
      - projects/farm-internship/content/lessons/day01.md
  - title: 故事
    files:
      - projects/farm-internship/content/stories/story_01.md
```

publish/build.yaml 範例

```
site:
  generator: mkdocs
  nav_from_manifest: true
base_sections:
  - 專案總覽
  - 課程設計
  - 故事
  - 講義

pdf:
  engine: pandoc
  page_size: A4
  cover_template: publish/templates/cover.md
  pdf_engine_hint: xelatex

docx:
  engine: pandoc
  reference_docx: publish/templates/reference.docx
```

(Pandoc 可在多種標記/文件格式間轉換，包含 Markdown、docx，也可產出 PDF；並提供示例指令與模板參考方式。[7](#))

1 15 Flowcharts Syntax

https://mermaid.ai/open-source/syntax/flowchart.html?utm_source=chatgpt.com

2 8 YAML Ain't Markup Language (YAML™) Version 1.2

https://yaml.org/spec/1.2.1/?utm_source=chatgpt.com

3 YAML Ain't Markup Language (YAML™) revision 1.2.2

https://yaml.org/spec/1.2.2/?utm_source=chatgpt.com

4 5 10 MkDocs

https://www.mkdocs.org/?utm_source=chatgpt.com

6 12 Demos

https://pandoc.org/demos.html?utm_source=chatgpt.com

7 11 Pandoc User's Guide

https://pandoc.org/MANUAL.html?utm_source=chatgpt.com

9 YAML Ain't Markup Language (YAML)

https://www.loc.gov/preservation/digital/formats/fdd/fdd000645.shtml?utm_source=chatgpt.com

13 14 Front Matter | Jekyll • Simple, blog-aware, static sites

https://jekyllrb.com/docs/front-matter/?utm_source=chatgpt.com