

課程講義 / VIBE CODING

智慧計算與 AI 應用開發實戰

從模組化系統思維、GitHub 專案實作，到 Streamlit 雲端發布

Vibe Coding 雲端部署全攻略

定義

先寫輸入與輸出

實作

完成一個小功能

驗證

測試與修正結果

發布

部署並取得網址



想法 → 程式 → 儲存庫 → 公開網址

把系統切得清楚，
把功能連得可靠。

授課教師 | 陳宏銘 教授

東海大學 智慧計算暨應用數學系

圖文編排版 | 2026 年 10 月 22 日

依提供之課堂整理稿編製；新增教學圖解與技術校註。

這堂課要帶走的能力

把需求說清楚、把功能拆開、把錯誤讀懂，最後交出一個別人可以打開操作的作品。

導讀

學習路徑與內容索引

依「作業規範 → 系統思維 → 雲端實作 → 專案案例 → 繳交檢核」閱讀。

第一篇 | 課程與期末專題 03

三項繳交要素、課堂公告與評量提醒

第二篇 | 模組化開發思維 04

用輸入、處理、輸出拆解一個系統

第三篇 | 從零到一雲端部署 05-09

需求提示詞、GitHub、套件清單、部署與除錯

第四篇 | 實戰專案圖解 10-22

數學統計、賽局財務、醫療、教務與生活工具

第五篇 | 開發心得與作業範本 23-24

維護邊界、專題方向與交件檢核

附錄 | 名詞與校註 25-26

語音名詞還原、技術修正與官方參考資料

閱讀與使用方式

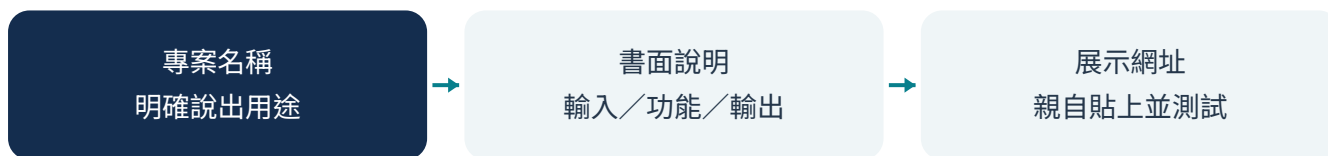
深藍區塊標示核心步驟；青綠區塊提示系統邏輯；橘色提醒標示需要判斷或修正的觀念。圖表為本講義重繪的教學示意，並非既有系統畫面或真實研究數據。

編輯說明：保留原稿的課堂故事、專案代號與作業要求，將重複口語整理為書面文字。專案功能及教師經驗依提供內容敘述，未逐一查驗原始碼或上線狀態；專案代號不代表已核實的公開網址。

第一篇 | 第一章

期末作業：三大必備要素

繳交平台：東海大學 iLearn。作業重點是讓讀者理解作品，並實際開啟操作。



一、開發程式的名稱

請為系統取一個明確名稱，例如「AI 賽事勝負預測紀錄系統」、「即時藝術品拍賣平台」或「3D 數學函數繪圖器」。名稱應讓人一眼看出作品要解決什麼問題。

二、作品的輸入、主要功能與輸出

- (一) 輸入：使用者輸入或上傳哪些資料？例如圖片、學號、方程式參數，或經去識別化的測試資料。
- (二) 主要功能：系統執行什麼處理？例如統計準確率、計算機率、比對答案、合成圖片。
- (三) 輸出：最後顯示或提供下載什麼？例如互動圖表、Word 報告、海報圖片或預測結果。

三、成果展示連結：必須親自貼上

Streamlit 公開展示網址為必填，GitHub 儲存庫網址依本課規範為選填。公開展示網址不保證能反查原始碼；若希望教師檢閱程式，建議一併附上 GitHub 網址。

課堂評量與公告

依原課堂說明，完成並繳交作業者，基本分數從 80 至 90 分起評；希望爭取更高分數，可主動展示作品細節。實際給分仍依教師評量。原稿提及「下週五」採自主學習與線上諮詢，日期與形式請以 iLearn 最新公告為準。

完整教材與教師作品說明依課堂公告於 iLearn 提供；遇到卡關可在課程群組提問或約教師討論。

第二篇 | 第二章

模組化：切得清楚，連得可靠

用小功能逐步組裝系統，讓錯誤可定位、更新可控制。

想像要建造一座樂高城堡：若一開始要求 AI 一次完成所有功能，任何小錯誤都可能牽動整個專案。比較可靠的做法，是先界定每個模組的責任與資料介面，再逐一完成、測試與整合。



圖 1 | 模組依輸入與輸出分工。實際專案的呼叫順序應依資料流程設計。

一、拆解功能，建立標準介面

登入只處理身分與權限；上傳只處理檔案驗證；計算只接收整理好的資料並回傳結果；報告模組再把結果轉成下載檔。每個模組都應定義欄位、型別、單位與失敗時的回應。

二、逐塊測試，再串起完整流程

先驗證單一功能可運作，再確認前一模組的輸出符合下一模組的輸入。遇到不支援的圖片格式時，優先修正上傳模組；仍須回歸檢查相關流程，因為模組間的資料契約也可能受到影響。

切得細，不等於切得越碎越好

以責任清楚、介面穩定為原則。過度拆分会增加串接成本；共享資料、版本與錯誤處理仍要有一致規則。

第三篇 | 第三章、第四章第一節

第一步：把想法變成開發規格

三項核心技能：請 AI 協作寫程式、用 GitHub 保存專案、用 Streamlit 發布展示。



一、先說清楚用途與使用者

課程示範使用 Google Gemini，也可使用其他 AI 工具。模型與介面名稱可能變更，無須拘泥於特定方案。需要瀏覽器介面時，明確指定 Python 與 Streamlit；若只做本機運算，則不一定需要網頁框架。

二、課堂提示詞：AI 預測與實際結果配對

可以直接使用的需求範本

請以 Python 與 Streamlit 開發「AI 賽事預測紀錄網站」。使用者可新增比賽、填寫各 AI 模型的預測、上傳相關圖片，並在賽後輸入實際結果。系統依模型計算正確次數、有效預測筆數與準確率，顯示排行榜並提供資料匯出。請先列出模組、資料欄位、儲存方式及驗收條件，確認後再產生程式。

三、先補齊四個規格問題

1. 一場比賽能否有多次預測？是否保留賽前時間戳記？
2. 平手、取消或尚未結束的比賽要如何計分？
3. 使用者填寫 AI 結果，還是系統真的呼叫模型 API？
4. 資料要保存多久？是否需要帳號、管理權限與備份？

教學提醒：紀錄各模型的預測不等於網站已自動串接模型。若要自動呼叫 API，需另外處理金鑰、費用、失敗重試與回應格式。

第三篇 | 第四章第二節

第二步：建立 GitHub 專案

先準備兩個關鍵檔案，再逐步擴充模組。

一、建立儲存庫

登入 GitHub，選擇 New repository，輸入專案名稱，例如 aicode。課堂展示可採 Public，讓他人查看程式；涉及私人程式時可採 Private，並依部署平台的存取授權設定。完成後建立儲存庫。



二、建立 app.py 並提交版本

選擇 Create new file，將檔名設為 app.py，貼上經檢查的 Python 程式碼，再以 Commit changes 儲存。提交訊息應說明改動，例如「新增預測紀錄表單」，方便日後追蹤與回復。

三、建立 requirements.txt

此檔案告訴部署環境需要安裝哪些第三方套件。以下為示意清單；真正使用時只列出專案所需套件，並於測試後設定相容版本。[1]

```
streamlit
pandas
plotly
Pillow
python-docx
```

不要把所有 import 名稱照抄進清單

os、time、math、sqlite3、hashlib 屬於 Python 標準函式庫，通常不列入 pip 安裝清單。另有套件的安裝名稱不同於匯入名稱，例如安裝 Pillow，程式使用 PIL；安裝 python-docx，程式使用 docx。
[1][4]

建議補上 README.md，說明系統用途、啟動方式、輸入欄位與已知限制。不要將 API 金鑰或真實個資提交至公開儲存庫。[3]

第三篇 | 第四章第三節

第三步：發布到 Streamlit

部署平台正式名稱為 Streamlit Community Cloud；畫面文字以實際版本為準。[2]



一、連結 GitHub 並建立應用程式

前往 share.streamlit.io，登入並授權所需的 GitHub 存取權。選擇 Create app，指定儲存庫、分支與主程式路徑。若 `app.py` 放在子資料夾，必須填入正確的相對路徑。

二、設定網址與部署環境

依平台畫面設定可用網址，確認 Python 版本、相依套件與資料檔位置。若需 API 金鑰或服務憑證，將其放入平台的 Secrets 設定，並由程式讀取；不要硬寫在公開程式碼內。[1][2][3]

三、按下 Deploy，先讀紀錄再判斷

部署完成所需時間取決於套件安裝與程式載入，可能需數分鐘。成功後親自開啟網址，測試表單、上傳、圖表與下載。若作品需任何人可使用，還須確認分享與存取設定符合課程要求。

上線後要再測一次

用未登入開發帳號的瀏覽器或無痕視窗開啟展示網址；確認使用者不必取得你的 GitHub 帳號權限即可操作預定公開的功能。測試結束後再貼進作業。

四、更新程式與重新部署

對部署分支提交變更後，平台會依變更處理更新或重建環境。若未成功更新，查看管理介面與執行紀錄，必要時重新啟動。不要把「幾秒內完成」當作每次都成立的保證。

第三篇 | 第四章第三節補充

除錯：從紅字回到可重現的問題

課堂案例：套件清單混入內建模組，導致部署環境安裝失敗。



一、先區分是哪一層出錯

- (一) 安裝階段：套件名稱錯誤、版本不相容，或系統相依套件未備妥。
- (二) 啟動階段：語法錯誤、匯入失敗、檔案路徑不正確。
- (三) 執行階段：資料欄位缺失、圖片解碼失敗、API 金鑰或網路連線問題。

二、課堂修正案例

若 requirements.txt 誤列 os、math 或 sqlite3，應移除這些標準函式庫名稱，保留真正需要安裝的第三方套件後提交。這是本案例的可能原因，不能將所有部署紅字都歸因於相同問題；先閱讀完整紀錄再修正。[1][4]

交給 AI 的除錯提示詞

以下是我的執行環境、重現步驟、完整錯誤訊息與相關程式片段。請先判斷根因，指出需要修改的檔案與最小修改，說明原因，再提供驗證步驟。請保留已完成的功能，並提醒我是否需要同步調整套件清單。

三、修正後的驗證順序

先重做原本失敗的操作，確認錯誤消失；再測正常輸入、空白輸入與錯誤格式；最後檢查其他相關功能。每次提交盡量聚焦一個問題，降低回復與追查的難度。

分享錯誤紀錄前，遮蔽金鑰、密碼、個資與資料庫連線資訊。若金鑰已公開，應撤銷並更換，而非只刪除畫面上的文字。[3]

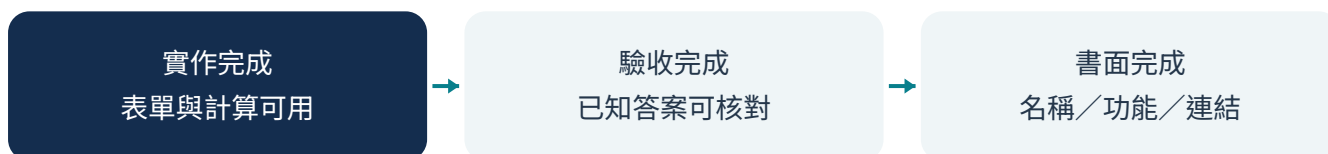
第三篇 | 第四章第四節

完成閉環：驗收作品並產出說明

「可以啟動」只是起點；「別人能照說明完成操作」才是可交付成果。

一、先用小資料確認統計正確

例如某模型有 10 筆已結束且可評分的預測，其中 7 筆正確，準確率應為 70%。未結束、取消或缺少預測的比賽，應按事前規則處理，不得任意混入分母。排行榜也應顯示樣本數，避免把 1 次全對誤判為穩定優勢。



二、請 AI 整理說明，但由本人核對

作業說明提示詞

請依據以下實際程式功能，整理一份作業說明，包含：一、開發程式名稱；二、使用對象與用途；三、輸入資料；四、主要功能；五、輸出結果；六、成果展示網址；七、已知限制。請勿加入程式尚未完成的功能。展示網址先留欄位，由我親自填入。

三、匯出文件與完成交件

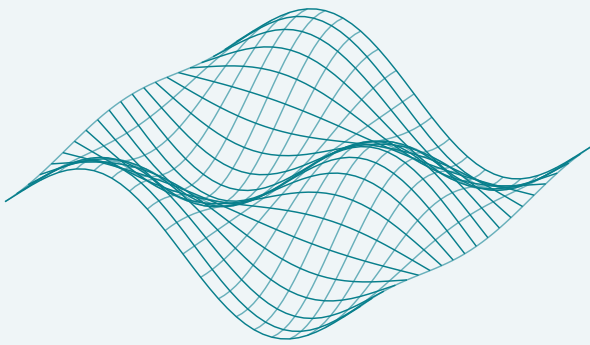
將說明複製到 Word 或其他文件工具，調整字體、段落與標題。若 AI 工具支援文件匯出，可依其功能下載。最後貼上真實展示網址，確認作業附件能開啟，再依 iLearn 作業區要求提交。

本講義第 24 頁提供可直接套用的作業結構與交件檢核。

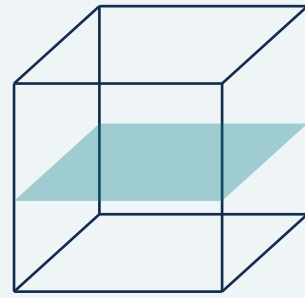
第四篇 | 第五章第一、二節

數學視覺化：從方程式到立體截面

專案代號：3dmath；cubecut、cubecut2、cubecut3



曲面示意： $z = \sin(x) \cos(y)$



立方體平面截面示意

圖 2 | 左：函數曲面網格；右：平面切割立方體。均為概念重繪。

一、3D 數學函數繪圖器

微積分與工程數學常遇到難以想像的曲面。輸入方程式、座標範圍與解析度後，系統建立網格，再繪製可旋轉的 3D 圖形。若需畫圓環或其他非單值曲面，通常須採參數式或隱式曲面表示。

範圍決定顯示區域；解析度決定採樣密度。若兩個方向各採 n 個點，網格點數約為 n 的平方，過高會增加計算與繪圖負擔。參數紀錄可另存至表單、試算表或資料庫，以便重現圖形。

二、立方體平面切割模擬器

課堂故事源於教師的幾何教學需求：斜切、垂直切或沿不同位置切入，截面如何變化？可用平面上一點與法向量定義切平面，計算與 12 條稜的交點，再去除重複點、依平面內順序排序，形成截面多邊形。

幾何邊界也要測試

平面可能不與立方體相交，也可能通過頂點、稜或與某一面重合。應處理重複交點與浮點誤差；若連續切割多刀，還需更新剩餘多面體的邊界。

第四篇 | 第五章第三節

高爾頓板：隨機如何形成分配

專案代號：galtonboard、galtonboard2

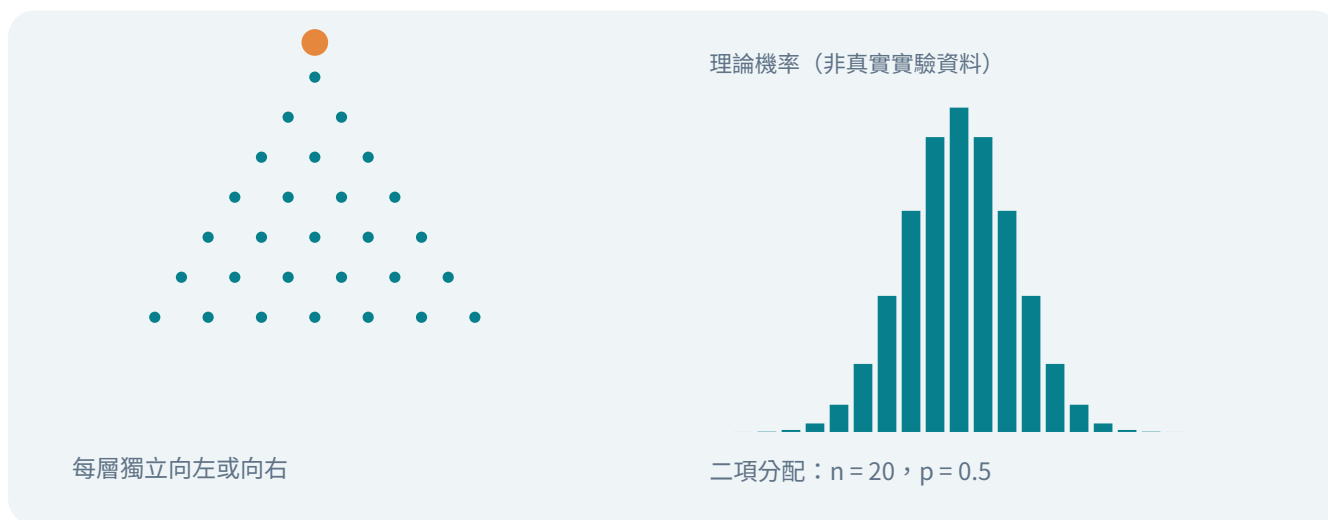


圖 3 | 理想化釘板與二項機率柱形。重點是區分「球數」與「層數」。

一、先理解二項分配

在理想模型中，每顆球經過 n 層，各層獨立地以機率 p 向右，以機率 $1-p$ 向左。向右的次數決定落點，因此落點服從二項分配。對稱釘板常取 $p = 0.5$ 。

二、為什麼看起來像鐘形曲線？

固定釘板層數時，增加球數會使觀察到的相對頻率更接近該二項分配；不會只因球數增加就自動變成常態分配。當層數夠大，且向左、向右的期望次數都不太小時，二項分配才可用常態分配近似。

三、動畫工具要配合呈現需求

課堂比較 Streamlit 網頁與 Python 本機動畫。若以逐格重跑網頁呈現，可能受通訊與重繪負擔影響；Pygame 或其他動畫方案可更直接控制畫面更新。網頁也能採前端動畫元件，流暢度取決於具體實作。

課堂故事與數學解讀

每顆珠子的方向看似不可預測，整體仍可能呈現穩定統計規律。這是機率分配的觀察，不代表個別生命結果已被固定決定。

第四篇 | 第六章第一、二節

拍賣與博弈：互動背後的機制

專案代號：auction、auctiontw；bidgame

一、即時拍賣：價格只是其中一個欄位

課堂以貓咪圖片、起標價 50 元與買家代號示範競標：賣家設定底價，買家出價，系統更新最高出價並於結標後顯示得標者。完整系統還需定義最低加價、截止時間、同時出價的先後與管理權限。

現場 WebP 圖片無法上傳，改用 JPG/PNG 後成功。這表示當時程式或環境的支援範圍有限，並非 WebP 格式必然不能使用。應檢查副檔名、實際內容與圖片解碼結果。

單一路徑示意：初始本金 100、每次下注 10、勝率 0.46

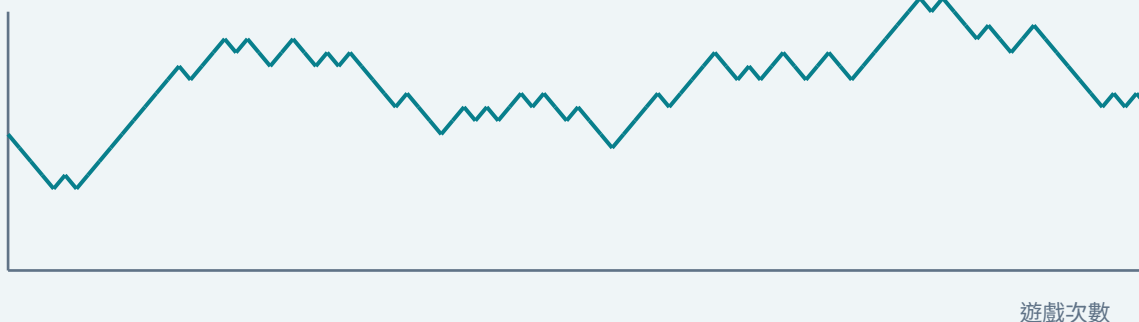


圖 4 | 負期望遊戲的一條人工模擬路徑，單次波動不能證明所有玩家的結局。

二、比大小：先問規則，再算期望值

若每次贏 1 元、輸 1 元，勝率為 p ，則每次期望淨收益為 $2p-1$ 元。當 p 小於 0.5，長期反覆下注具有負期望；有限本金也可能因波動先行耗盡。破產機率仍取決於勝率、賠率、下注方式與停止條件。

「本多終勝」是口語比喻，不是普遍定理

本金較大可能較能承受短期波動，但不會把負期望變為正期望，也不保證獲利。「十賭九輸」是課堂勸誡，不能直接解讀為所有遊戲都有 90% 輸錢機率。

第四篇 | 第六章第三節

財務報告：從計算走到文件交付

專案代號：cash_flow_report、cashreportfull、stock 系列

現金流示意：單位可為萬元；不是任何公司的實際投資案



圖 5 | 先明確定義現金流時間，再計算淨現值與產出圖表。

一、情境：算完之後，還要交報告

資本預算、企業估價或投資損益分析，常需要將結果交給不使用程式的人。課堂系統讓使用者在網頁輸入數據，完成計算與繪圖，再一鍵匯出含標題、圖表與說明的 Word 文件。

二、財務計算模組要先固定假設

輸入至少應說明初始投資、各期現金流、期間、折現率與幣別。淨現值是「未來各期現金流折現值的總和，再扣除初始支出」。股票損益則另須處理買賣價格、股數、股息與交易成本，不能與專案現金流混用。

三、圖表與文字共用同一份結果

建議由計算模組產出結構化結果，再同時交給網頁與報告模組。如此可避免畫面上的數值與下載檔不同。報告應列出資料日期、來源、假設、單位與四捨五入規則，並檢查中文字型與分頁。

教學重點

把重複工作自動化的前提，是先確認公式與輸入正確。AI 可協助排版與文字說明，但結論仍須能追溯至計算結果。本圖為教學案例，未提供任何投資標的建議。

第四篇 | 第六章第四節

FRED：把資料來源放回圖表裡

專案代號：gdp-dashboard、gdp-dashboard-1

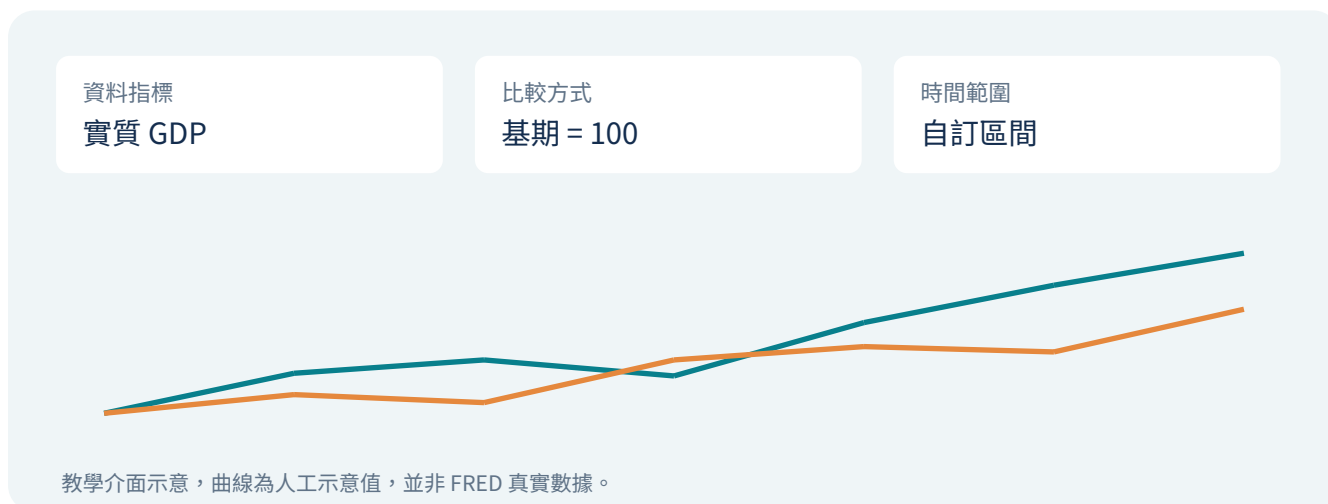


圖 6 | GDP 儀表板介面示意。真正比較前，須確認指標、單位、頻率與基期。

一、資料來源：聖路易斯聯準銀行 FRED

FRED 是 Federal Reserve Economic Data，由聖路易斯聯準銀行維護，彙整許多國內外機構的經濟時間序列。可依指標與序列代碼查詢，並透過下載或 API 串接；它並不涵蓋所有國家、所有指標。[5]

二、GDP 比較要先對齊定義

名目 GDP 受價格變動影響；實質 GDP 用於觀察經價格調整的產出變化。跨國比較還需考慮匯率、購買力平價、人口與基期。不同頻率或缺值的資料，也應先清理再畫圖。

三、課堂的投資觀察如何嚴謹表述？

原課堂用長期 GDP 成長鼓勵同學關注投資。教材應補充：GDP 可能衰退，不保證持續上升；總體經濟成長也不等於任一股票必然獲利。此案例重點是資料閱讀與圖表分析能力。

每張資料圖至少保留四項資訊

指標全名與序列代碼、資料提供者、單位與頻率、擷取日期。使用 FRED 時也要查看原始來源及使用條件，不能把可下載視為所有用途皆無限制。[5]

第四篇 | 第七章第一節

認知風險預測：模型與臨床判斷

專案代號：dementiapro、dementiaapp



圖 7 | 風險模型是輔助工具，量表結果仍應與病史及其他臨床資訊共同評估。[6]

一、課堂展示的輸入與輸出

原稿以年齡、性別與認知量表分數示範模型推論，並描述批次上傳 Excel 後產生機率與分布圖的功能。開發時須記錄量表版本、分數範圍、缺值規則及模型適用族群。

二、不能把示範數值當成醫學事實

原稿提及「99 分、正常約 80%、MCI 約 11.9%」等畫面數字，未提供模型版本、完整輸入或驗證資料，因此本講義不將其列為可驗證的風險估計。年齡較高可能影響風險，但不能據此推導特定個人的診斷。

三、量表不同，分數意義也不同

CASI 與 MMSE 的計分尺度不同，不能把所有量表都視為滿分 100 分。低分應進一步評估，但不能僅以單一分數直接判為重度失智。認知測試可發現問題，單靠結果不能確認失智症及其原因。[6]

系統驗證需要什麼？

確認訓練與外部測試族群、區別能力、校準情形及不同族群表現。批次上傳只代表自動處理資料，不代表結果已完成醫師診斷。教學展示請使用合成或適當去識別化資料。

第四篇 | 第七章第二節

醫療共享決策：從文獻到討論

專案代號：Sdmpoc；PoC＝概念驗證原型

課堂故事來自醫療專家提出的需求：希望結合文獻與 AI 分析，協助整理醫療資訊並支持醫病共享決策。原型可驗證技術是否可行，但距離正式臨床應用仍有資料、驗證、工作流程與維護等問題要解決。



一、輸入：文獻與個案情境

先確認文獻來源、研究對象、介入方式、比較方案與結果指標。病患狀態須由適當授權的人員整理，避免將不必要的可識別資料送入外部服務。

二、處理：摘要仍要能追溯證據

AI 可以協助抽取與比較資訊，但生成內容要附可核對的來源，並保留不確定性。查無資料時應明確說明，不得以看似完整的語句填補證據空缺。

三、輸出：支援討論的摘要

適當輸出是供專業人員核對的方案整理、可能效益、風險、證據限制與待確認事項。共享決策還需要病患理解與偏好，不能只讓模型輸出單一「最佳答案」就宣告完成。

課堂的實務提醒

低成本、短時間與大量功能往往互相牽制。接案前應釐清原型要驗證什麼、由誰確認內容、後續由誰維護，並將未完成的部分寫進交付說明。

第四篇 | 第八章第一、二節

教務自動化：把規則寫成可追蹤流程

案例：學測落點與網站分析；grading-system

一、學測落點與校系資源比較

依原稿，系統輸入成績與相關指標後，提供落點分析及各校系網站資源比較。完整實作應註明招生年度、管道、採計方式與資料來源；歷年成績不能直接保證未來錄取，應將結果定位為資料比較與參考。



二、雲端簽到、作品上傳與加分

課堂以測試學號 8899 與教師姓名示範註冊、簽到及上傳作品。原示範中，簽到與上傳分別觸發加分，分數達到 90 分；再抽到加 1 分，總分為 91 分。這是系統展示情境，不應直接取代正式課程的評量規則。

三、抽獎機制需要可說明的機率

教師幽默提到自己總是第一個抽、只抽到 1 分。開發時應確認抽獎分布、是否獨立、是否有限額，以及每人能抽幾次；不能只憑少數測試就認定「第一個抽比較差」。

教務系統的資料設計

學生識別碼、時間戳記、作業檔案位置、評分項目與人工修改原因，建議分開記錄。限制重複簽到與重複加分，並讓管理者能查核每筆分數來源。

第四篇 | 第八章第三、四節

題庫與證書：驗證身分，也驗證結果

專案代號：ipasexam、login2026；inmc、certification、ParticipateIncertification

一、考生登入與線上題庫

教師示範以密碼登入、選擇考科、作答並提交考卷。複製登入資料時，可能混入不預期的空白或大小寫差異；應先檢查輸入規則。密碼中的空白也可能是有效字元，不應未告知就自動刪除。

課堂盲猜得到 0 分是有趣的展示情節；實作時則要確認題目版本、正確答案、計分方式與交卷後顯示權限。應保存作答紀錄，避免事後更動題目造成成績無法核對。

修正原稿中的密碼處理方式

正式系統不應讓開發者直接查回使用者原密碼。密碼應採合適的加鹽密碼雜湊機制，例如 Argon2id；忘記密碼時走身分驗證與重設流程。單純使用 hashlib 並不代表已安全儲存密碼。[7]

二、I-NMC 頒獎展示與證明驗證

原稿提及已辦三屆 I-NMC，並展示得獎名單、音樂、錄音播報與獎項動畫。年份與屆次以活動公告為準。證書系統則依授權資訊查詢、下載與驗證參賽證明或獎狀。

建議採唯一證書編號或驗證碼查驗，公開端只揭露必要資訊。姓名、學號或身分證字號不宜單獨作為充分驗證因素；管理端需另有權限與操作紀錄。

自由接案提醒：功能開發往往比後續修改容易，應在交付前約定維護範圍、次數、期間與費用。

第四篇 | 第八章第五節

圖片字幕：功能完成之後，還有設計

專案代號：imagetext、speechposter



圖 8 | 原創貓咪與字幕版面示意，呈現「日期、事件、可調參數」的設計關係。

一、使用者真正想完成什麼？

學長希望在公務照片上固定加入日期、地點與事件標題，快速產出可下載的紀錄圖片。課堂用貓咪照片與「10 月 22 日、颱風來襲、開心的貓」作為操作例子；這是示範文案，不代表真實天氣紀錄。

二、把「嫌醜」轉成可調整的規格

與其反覆修改程式中的固定座標，不如先提供字級、文字位置、對齊、顏色、背景與透明度等選項。再準備幾個預設樣式，讓使用者在穩定範圍內自行調整。

三、輸入與輸出一起驗證

確認圖片格式、方向、尺寸與檔案大小；輸出前檢查文字是否超出邊界，字體是否支援繁體中文。下載圖片的尺寸、品質與命名也應明確，避免網頁預覽正常、存檔後卻模糊或裁切。

軟體工程的實際成本

AI 能快速產生功能，但美感、易用性與維護仍需反覆驗證。先談好可調範圍與驗收範例，能減少無止境的版面修改。

第四篇 | 第八章第六節

生活資料：發票與十年油耗紀錄

專案代號：invoicelist；oil、oilprice

一、發票管理：從記錄走向核對

發票工具可以保存日期、店家、金額與號碼，協助分類、查找與核對。若增加對獎功能，須明確標示期別、獎號來源、更新時間與核對結果；不能只憑不完整資料宣稱已中獎。

課堂以「有人兩次中千萬元」的新聞故事鼓勵同學保持希望；本稿未提供新聞來源，因此保留其勉勵意義，不將該故事列作已核實事實。中獎具隨機性，也不應當作可靠收益來源。



二、十年油耗：長期記錄的價值

依教師分享，自購車後持續記錄加油里程與油耗，累積約十年資料。這類自己真正在用的題材，適合從簡單表單逐步擴充為趨勢分析、費用統計與保養提醒。

三、先統一油耗定義

若比較每公升公里數，以行駛公里除以加油公升數；若比較每百公里耗油量，則以公升數除以公里數再乘 100。兩種指標的好壞方向相反，圖表必須註明單位。滿油法還要排除漏記或未加滿等情況。

選題建議

先從自己願意長期記錄的問題開始。穩定、乾淨且持續累積的資料，通常比一次加入大量華麗功能更有分析價值。

第四篇 | 第八章第七節

儲存架構：前端之外，資料放哪裡？

專案代號：cloudinarytw、googleconn、my-survey-app

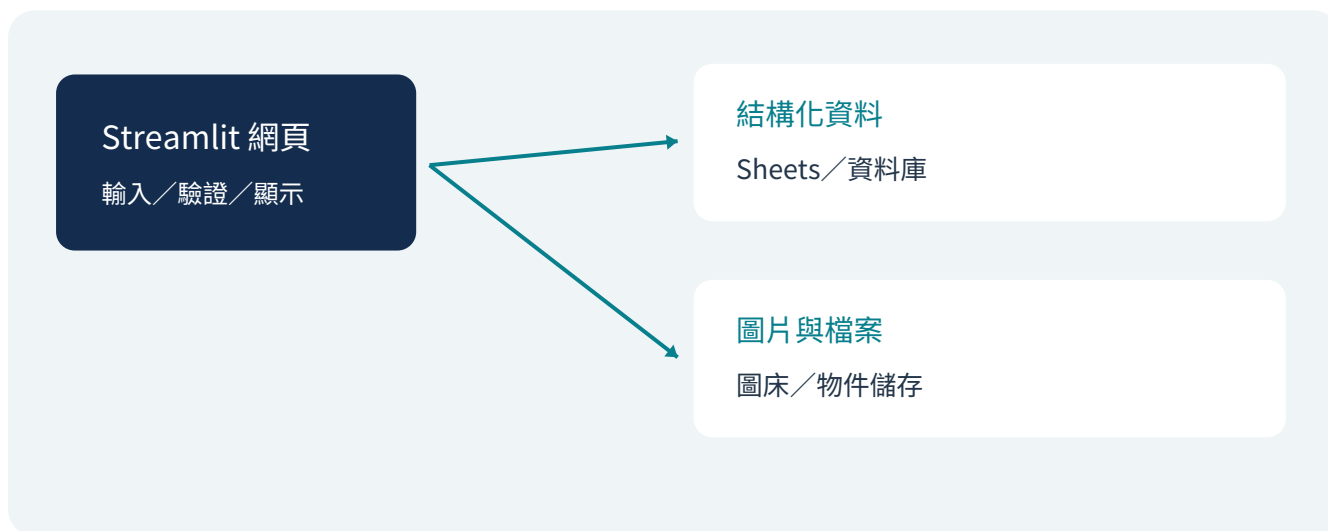


圖 9 | 表格欄位與圖片檔案可分開保存，再以識別碼或檔案位置建立關聯。

一、先區分資料種類

姓名代號、日期、分數與表單欄位，適合結構化表格或資料庫；圖片與文件通常放在檔案或物件儲存服務。Cloudinary 可處理圖片等媒體資產，Google Sheets 可支援部分小型表單情境，但兩者用途不同。

二、公開網址不等於資料永久保存

Streamlit 的工作階段狀態、程式執行環境與外部資料庫不是同一件事。需要跨工作階段、重新部署後仍保存的資料，應另規劃持久儲存、備份與權限，不能只依賴記憶體或部署目錄。

三、服務條件與多人使用

免費服務可能有容量、流量、請求數或方案限制，額度與計費方式以當期官方條款為準，不宜固定寫成「每天免費」。多人同時寫入試算表時，還要處理重複資料、競爭更新與錯誤重試。

串接時的最低設計要求

將金鑰放入 Secrets，限制可寫入範圍，設定檔案大小與格式，保存提交時間與唯一識別碼，並定義刪除及備份方式。[3]

第四篇 | 第八章第八節

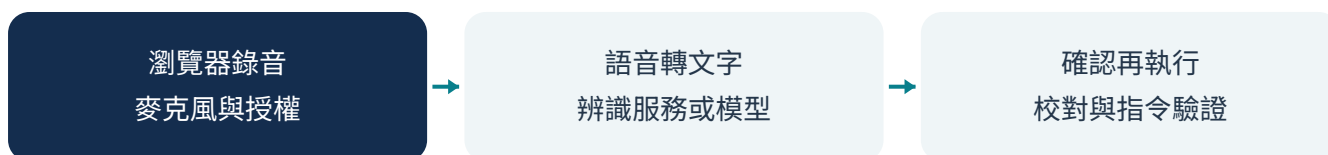
密碼估算與語音輸入：看清模型假設

專案代號：pwhash、pwsecrets；voice2text、voicetotext

一、密碼破解時間估算器

若每個位置可從 A 種符號獨立且等機率選取，長度為 L ，所有組合數是 A 的 L 次方。以每秒 r 次嘗試估算，最壞遍歷時間為組合數除以 r ，平均約為其一半。這是簡化模型，不是實際密碼強度保證。

真實攻擊還會利用常見密碼、字典、外洩資料與可預測模式；線上登入另有速率限制，離線破解則受雜湊演算法與硬體影響。展示時請使用虛構密碼，不要把正在使用的密碼輸入不明工具。[7]



二、網頁語音轉文字

課堂因教室桌機未接麥克風，僅展示介面；可改用具備麥克風的筆電或手機測試。除了硬體，還需確認瀏覽器授權、錄音格式、檔案大小及辨識服務是否可連線。

三、把語音接回 Vibe Coding

語音可以降低輸入需求的門檻，但辨識出的文字仍可能出錯。先顯示逐字結果，再讓使用者確認，才轉成程式需求或操作指令。涉及刪除、付款或公開發布等動作，更需明確的人為確認步驟。

教學練習

用一段 10 秒錄音描述需求，逐字檢查專有名詞，再改寫成包含「輸入、處理、輸出、驗收條件」的開發提示詞。

第五篇 | 教師心得

先做自己用得到的工具，再談擴張

作品的價值來自解決問題；長期可維護，才能持續使用。

一、想像力要接上具體需求

教師展示大量小工具，是希望同學發現程式可以處理生活、教學、研究與行政中的重複工作。起步時不必追求包山包海，先完成自己真正需要、願意反覆使用的一個小功能。

二、功能完成後，才進入維護考驗

免費提供系統也可能伴隨排版調整、需求變更與使用者抱怨。接案或協助開發前，應說明交付內容、支援期間、修改次數、資料責任與額外費用；需要維護的人力與時間，也應納入規劃。

三、團隊開發與 SOHO 接案

大型團隊需共同遵守架構、版本與協作規則；獨立接案則常由一人承擔溝通、開發、部署與維護。兩者都不只是在寫程式。課堂提醒不要低估個人小案子的持續溝通成本。

給同學的課堂勉勵

學會這套方法，完成自己的作品，並記得交作業。中間遇到問題，就回來問、回來學。AI 時代帶來許多實作機會，願大家都能做出有用途、能說明、可展示的成果。

四、選題時，先問自己三個問題

1. 誰會使用？目前哪一步最花時間或最容易出錯？
2. 我能先做出哪一個最小、可驗證的功能？
3. 資料、成本與後續維護，由誰負責？

第五篇 | 交件範本

期末專題說明與自我檢核

依此結構整理 Word 附件或 iLearn 作業區文字，並填入自己的真實成果。

一、專案名稱

填入作品名稱、姓名、學號與版本日期。

二、使用情境

說明使用對象、目前遇到的問題，以及本作品要改善的流程。

三、輸入資料

列出欄位、資料型別、單位、範例、檔案格式及必要限制。

四、主要功能

依實際操作順序說明資料處理、計算方式與模組分工。

五、輸出結果

說明畫面、圖表、數值或下載檔，並附至少一個完整操作例子。

六、成果連結

必填：Streamlit 展示網址。選填：GitHub 儲存庫網址。

七、驗證與已知限制

列出已測試情境、尚未完成部分、資料來源與必要的使用限制。

提交前逐項確認

- ☐ 展示網址可以開啟，分享權限正確。
- ☐ 作品說明與實際功能一致。
- ☐ 輸入、處理、輸出三者清楚，至少核對一組已知答案。
- ☐ 空白、錯誤格式與重複提交等情況已處理。
- ☐ 公開程式及畫面未包含真實密碼、金鑰或不必要個資。
- ☐ 附件能開啟，且已於 iLearn 完成提交。

附錄一

語音專有名詞勘誤與複習

依原稿脈絡整理；無原始錄音可再次核對者，以課堂整理稿為準。

Jimmy/Timmy/雙子座

Google Gemini：生成式 AI 工具與模型家族。

streaming/strangthen/Pishon

依語境還原為 Streamlit 或 Python，兩者不可互換。

嘻哈 b/Dhup

GitHub：程式碼託管與版本協作平台。

Newrepostitoria

New repository：建立新的儲存庫。

APP 點 PY

app.py：本課採用的主程式檔名，並非唯一可用名稱。

retiemment/RQUIREMENTSTST

requirements.txt：本課使用的 Python 相依套件清單。

四川/pre App/prearpp

依語境為 Streamlit Community Cloud 或 Create app。

VBCoding

Vibe Coding：以自然語言與 AI 協作開發的方式。

Ilan/idan/iland/ilane

東海大學 iLearn 教學平台。

全世界 Fade/GTP/AP

依語境為 FRED、GDP 或 API，屬不同概念。

WEPP

WebP：圖片格式；是否可用取決於程式與環境支援。

搜狐族

SOHO：小型或居家工作型態，本課語境指獨立接案。

補充名詞：PoC＝概念驗證；SDM＝醫病共享決策；MCI＝輕度認知功能障礙；API＝應用程式介面；FRED＝Federal Reserve Economic Data。

附錄二

技術校註與延伸閱讀

核對日期：2026 年 10 月 22 日。官方介面與服務條件仍可能更新。

一、本版主要校正

1. 增加球數使相對頻率接近二項分配；常態近似另有層數與機率條件。
2. 「本多終勝」不列作數學定理，GDP 成長不作為股市保證獲利的推論。
3. 密碼改採安全雜湊與重設流程，不教導查回原密碼。
4. 認知量表與模型機率不等於臨床診斷；未核實的示範機率不列為醫療依據。
5. app.py 為本課檔名慣例；部署時間、免費額度與格式支援均依環境確認。

二、官方參考資料

[1] Streamlit | App dependencies

<https://docs.streamlit.io/deploy/streamlit-community-cloud/deploy-your-app/app-dependencies>

[2] Streamlit | Deploy your app

<https://docs.streamlit.io/deploy/streamlit-community-cloud/deploy-your-app/deploy>

[3] Streamlit | Secrets management

<https://docs.streamlit.io/deploy/streamlit-community-cloud/deploy-your-app/secrets-management>

[4] Python Packaging User Guide | pip and venv

<https://packaging.python.org/en/latest/guides/installing-using-pip-and-virtual-environments/>

[5] St. Louis Fed | What is FRED?

<https://fred.stlouisfed.org/help/about/about-fred/what-is-fred/>

[6] U.S. National Library of Medicine | Cognitive Testing

<https://medlineplus.gov/lab-tests/cognitive-testing/>

[7] OWASP | Password Storage Cheat Sheet

https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html

來源界線：課堂公告、教師經驗、各專案用途與代號來自使用者提供文本；圖解及編輯校註為本版新增。官方資料用於核對一般技術與醫療解讀原則，並不驗證個別課堂專案的功能、評量或研究效能。