



3.2

進階提示技術：Zero-shot、One-shot、Few-shot Learning

📦 Zero-shot Learning（零範例學習）

- 概念：

在沒有提供範例的情況下，AI 完全依賴訓練資料生成回應。適用於基礎知識查詢或一般性問題。

- 範例：

提問：「什麼是人工智慧？」

AI 回應：「人工智慧是一種模擬人類智能的技術，應用於語音識別、自然語言處理等領域。」

📦 One-shot Learning（單範例學習）

- 概念：

提供一個範例，幫助 AI 學習格式或語氣。

- 範例：

提問：「請寫一個 50 字的產品描述，例如：『這款無線耳機提供高品質音效與長效電池續航。』」

AI 回應：「這款智能手錶具備全天候健康監測與長效續航，非常適合運動愛好者。」

🗑 Few-shot Learning (少範例學習)

- 概念：

提供多個範例，讓 AI 模仿範例模式生成更精確的回答。

- 範例：

提問：「請參考以下範例，寫一段科技產品評論。」

範例：

- 「這款手機擁有頂級相機性能，非常適合攝影師。」
- 「這款筆電搭載最新晶片，提供卓越續航力。」

AI回應：「這款耳機具備降噪功能與高解析音效，是音樂愛好者的理想選擇。」

請參考以下範例，寫一段科技產品評論。

範例：

- 「這款手機擁有頂級相機性能，非常適合攝影師。」
- 「這款筆電搭載最新晶片，提供卓越續航力。」

「這款智慧手錶配備先進健康監測功能，全天候追蹤心率與睡眠狀態，適合運動愛好者與健康管理。」





請參考以下範例，寫一段科技產品評論。

範例：

- 「這款手機擁有頂級相機性能，非常適合攝影師。」
- 「這款筆電搭載最新晶片，提供卓越續航力。」

2/2

「這款智慧手錶內建先進健康監測功能，能精準追蹤心率與睡眠狀況，適合健康管理人士。」



附註

再多重送二次問ChatGPT，三次結果，都有變化。

3.3 Chain-of-Thought (CoT) 推理與應用

📦 什麼是 Chain-of-Thought (CoT) 推理？

- CoT 的定義：

CoT 是指引導 AI 逐步推理，透過分步回答提升解決複雜問題的能力。

📦 CoT 的應用情境

1. 數學推理：例如多步運算題。
2. 邏輯推理：例如分析問題中的隱含關係。
3. 決策分析：幫助制定多步驟策略。

- CoT 實戰範例 提問：「如果一個農場有 5 隻雞，每隻雞每天生 2 顆蛋，10 天後總共會有多少顆蛋？」

AI 回應（逐步推理）：

- 「每隻雞每天生 2 顆蛋。」
- 「5 隻雞每天總共生 10 顆蛋。」
- 「10 天後總共會有 100 顆蛋。」

如果一個農場有 5 隻雞，每隻雞每天生 2 顆蛋，10 天後總共會有多少顆蛋？

讓我們一步步計算：

1. 每隻雞每天生的蛋數 = 2 顆
2. 農場有 5 隻雞，所以 每天的總產蛋數 = $5 \times 2 = 10$ 顆
3. 經過 10 天，總共生產的蛋數 = $10 \times 10 = 100$ 顆

答案：10 天後總共會有 100 顆蛋。

🔊 🗨️ 📄 📌 📎



3.4

ReAct 框架（Reasoning+Acting）提升 AI 問答能力

什麼是 ReAct？ReAct（推理 + 行動）框架讓 AI 結合邏輯思考與具體行動，以多輪對話逐步解決問題。

📦 ReAct 實戰範例

- 傳統提問：「請推薦一本適合學習 Python 的書籍。」
- ReAct 提問：
 - 「假設你是一位 Python 教師，請推薦一本適合初學者的 Python 書籍，並說明推薦理由。」
 - AI 回應：「我建議《Python Crash Course》。這本書內容基礎且實用，適合初學者快速掌握 Python。」

假設你是一位 Python 教師，請推薦一本適合初學者的 Python 書籍，並說明推薦理由。

推薦書籍：《Python 程式設計入門經典》

📖 推薦理由：

1. 內容循序漸進：從基礎語法到進階應用（如資料結構、物件導向），適合零基礎學習者。
2. 實作導向：書中包含大量範例與練習題，幫助讀者邊學邊做，提高實戰能力。
3. 清楚易懂：語言簡潔，概念解釋詳細，即使沒有程式背景也能輕鬆理解。

這本書適合初學者快速掌握 Python 基礎，建立扎實的程式設計能力！📖📦

🔍 📄 📖 📑 🗑️

- 再增加資訊，直接指名書籍。

假設你是一位 Python 教師，要推薦適合初學者的 Python 書籍，書名“Python範例學習書 | 輕鬆、有趣學習Python程式設計”給學習者，並說明推薦理由。

■ 推薦書籍：《Python範例學習書 | 輕鬆、有趣學習Python程式設計》

◆ 推薦理由：

1. 範例導向，學習更直觀

- 本書透過 實際案例與情境 來教學，讓讀者在解決問題的過程中學習 Python，提升實戰能力。

2. 輕鬆有趣，降低學習門檻

- 內容淺顯易懂，搭配 趣味範例，即使沒有程式基礎，也能快速入門，避免枯燥的學習體驗。

3. 逐步引導，循序漸進學習

- 從基礎語法到進階應用，如資料處理、函式、物件導向，適合初學者打下扎實基礎。

★ 適合對象：完全沒有程式經驗的學習者、自學者，或希望透過實作掌握 Python 的讀者。

這本書能幫助你輕鬆踏入 Python 世界，讓學習變得更有興趣、更有效率！📖👍

🔊 📄 📌 📎 📧

🏠 ReAct 的應用領域

1. 技術支援：幫助診斷問題並提供解決方案。
2. 多輪對話：像人類專家一樣逐步深入分析。

8.2 AI 幫助寫 Python 自動化程式

生成 Python 自動化程式

● 範例 1：自動分類檔案。

提問：「請幫我撰寫python程式，將 'Downloads' 資料夾的文件按類型分類。」

AI 產出：

請幫我撰寫python程式，將 'Downloads' 資料夾的文件按類型分類。

這是一個 Python 程式，會掃描 Downloads 資料夾，並根據檔案類型（副檔名）將文件分類到對應的子資料夾中，例如 Documents、Images、Videos、Archives、Others 等。這可以幫助你整理下載的文件。

''' Sort Downloads

這個程式會：

1. 在 Downloads 資料夾內建立 Documents、Images、Videos 等分類資料夾。
2. 掃描 Downloads 內的檔案，根據副檔名移動到對應的資料夾。
3. 未知類型的檔案將移動到 Others 資料夾。

你可以將這個腳本設定為定期執行，例如使用 Windows 工作排程器或 macOS/Linux 的 `cron`。

🔊 📄 📁 🔄 🗑️



```
import os
import shutil

# 設定你的 Downloads 資料夾路徑
DOWNLOADS_FOLDER = "C:\\Users\\jack\\Downloads"

# 設定分類的目標資料夾
CATEGORIES = {
    "Documents": [".pdf", ".doc", ".docx", ".txt", ".xlsx",
                  ".csv", ".ppt", ".pptx"],
    "Images": [".jpg", ".jpeg", ".png", ".gif", ".bmp",
               ".tiff", ".svg"],
    "Videos": [".mp4", ".mkv", ".mov", ".avi", ".flv", ".wmv"],
    "Music": [".mp3", ".wav", ".aac", ".flac", ".ogg"],
    "Archives": [".zip", ".rar", ".7z", ".tar", ".gz"],
    "Executables": [".exe", ".msi", ".dmg", ".sh"],
    "Code": [".py", ".java", ".cpp", ".js", ".html", ".css",
             ".php", ".rb", ".go", ".swift"]
}

# 確保分類資料夾存在
def create_folders():
    for category in CATEGORIES.keys():
        folder_path = os.path.join(DOWNLOADS_FOLDER,
                                     category)
        if not os.path.exists(folder_path):
            os.makedirs(folder_path)

def classify_files():
```

```
for filename in os.listdir(DOWNLOADS_FOLDER):
    file_path = os.path.join(DOWNLOADS_FOLDER, filename)

    # 確保是檔案而不是資料夾
    if not os.path.isfile(file_path):
        continue

    file_ext = os.path.splitext(filename)[1].lower()
    moved = False

    for category, extensions in CATEGORIES.items():
        if file_ext in extensions:
            target_folder = os.path.join(DOWNLOADS_
FOLDER, category)
            shutil.move(file_path, os.path.join(target_
folder, filename))
            print(f"Moved: {filename} -> {category}")
            moved = True
            break

    # 未分類的檔案放到 Others
    if not moved:
        other_folder = os.path.join(DOWNLOADS_FOLDER,
"Others")
        if not os.path.exists(other_folder):
            os.makedirs(other_folder)
        shutil.move(file_path, os.path.join(other_folder,
filename))
    print(f"Moved: {filename} -> Others")
```



```
if __name__ == "__main__":  
    create_folders()  
    classify_files()  
    print("File classification completed!")
```

🗑 AI 協助除錯與優化

1. 錯誤排查：

提問：「這段程式碼出錯，幫我找出問題。」

AI 提供詳細分析與修正建議。

2. 性能優化：

- 減少多餘迴圈。
- 使用有效率工具如 pandas 進行資料處理。

8.3 VBA + AI 提升 Excel 自動化能力

🗑 VBA 的核心概念

● 應用情境：

1. 自動生成報表。
2. 批量資料處理與格式化。
3. 自動化資料驗證。

讓 AI 幫助撰寫 VBA 程式碼

● 範例 1：排序 Excel 資料。

提問：「Excel 如附檔，請給用價格排序 vba 碼。」

AI 產出：

編號	品名	價格	數量	金額
A001	巧克力	100	2	
A002	牛奶糖	50	3	
A003	口香糖	30	5	
A004	巧克力	100	2	
A005	牛奶糖	50	3	
A006	口香糖	30	5	

Excel 如附檔，請給用價格排序 vba 碼

以下是 VBA 程式碼，可根據 價格 欄 (C 欄) 對表格數據進行排序 (由小到大或由大到小)：

VBA 程式碼

```
vba
Sub SortByPrice()
    Dim ws As Worksheet
    Dim lastRow As Long

    ' 設定當前工作表
    Set ws = ActiveSheet

    ' 找到最後一行數據 (基於 A 欄)
    lastRow = ws.Cells(ws.Rows.Count, 1).End(xlUp).Row

    ' 檢查是否有足夠的數據可供排序
    If lastRow < 2 Then
        MsgBox "沒有足夠的數據可供排序！", vbExclamation, "排序失敗"
        Exit Sub
    End If
```



```
' 執行排序，根據價格 (C欄) 進行升序排列  
ws.Range("A1:E" & lastRow).Sort Key1:=ws.Range("C2"), Order1:=xlAscending, Header:=xlYes  
  
' 提示完成  
MsgBox "數據已根據價格排序完成！", vbInformation, "排序成功"  
End Sub
```

如何使用

1. 開啟 Excel，按 **Alt + F11** 進入 VBA 編輯器。
2. 插入模組：點選 插入 > 模組。
3. 貼上程式碼並關閉 VBA 編輯器。
4. 在 Excel 按 **Alt + F8**，選擇 **SortByPrice**，然後點選 執行。

功能說明

- 這段程式碼會對 **A1:E** 最後一行的數據進行排序。
- 以 **C 欄 (價格)** 作為排序依據，**xlAscending** 代表 由小到大。
- 如果要改為由大到小，請將：

```
vb Order1:=xlAscending
```

改為：

```
vba Order1:=xlDescending
```

這樣就能快速根據價格對表格進行排序了！✍

產生以下結果：

編號	品名	價格	數量	金額
A003	口香糖	30	5	
A006	口香糖	30	5	
A002	牛奶糖	50	3	
A005	牛奶糖	50	3	
A001	巧克力	100	2	
A004	巧克力	100	2	

再要求價格由大至小：

價格由大至小

以下是 根據價格 (C 欄) 由大到小 (降序) 排序的 VBA 程式碼：

VBA 程式碼

```
vba
Sub SortByPriceDescending()
    Dim ws As Worksheet
    Dim lastRow As Long

    ' 設定當前工作表
    Set ws = ActiveSheet

    ' 找到最後一行數據 (基於 A 欄)
    lastRow = ws.Cells(ws.Rows.Count, 1).End(xlUp).Row

    ' 檢查是否有足夠的數據可供排序
    If lastRow < 2 Then
        MsgBox "沒有足夠的數據可供排序！", vbExclamation, "排序失敗"
        Exit Sub
    End If

    ' 執行排序，根據價格 (C 欄) 進行降序排列
    ws.Range("A1:E" & lastRow).Sort Key1:=ws.Range("C2"), Order1:=xlDescending, Header:=xlYes

    ' 提示完成
    MsgBox "數據已根據價格 (由大至小) 排序完成！", vbInformation, "排序成功"
End Sub
```



結果如下：

編號	品名	價格	數量	金額
A001	巧克力	100	2	
A004	巧克力	100	2	
A002	牛奶糖	50	3	
A005	牛奶糖	50	3	
A003	口香糖	30	5	
A006	口香糖	30	5	

- 範例 2：自動化 Excel 報表製作。

提問：「請用 VBA 自動生成樞紐分析表。」

AI 產出：提供完整的 VBA 程式碼範例，包括資料提取、篩選與格式化。