

文不對「題」—— 探討相同問題不同變 因對 ChatGPT 回覆內容的影響



研究領域：人文類

指導老師：曾敏雅 老師

研究學生：楊耀宇 撰

中華民國一百一十四年四月

摘要

因為我本人非常喜歡使用 ChatGPT，而且有一次我去上課時使用 ChatGPT 生成程式碼，老師生成的程式碼可以正常的運行，但我的不行。所以我想透過本實驗知道那些變因會影響 ChatGPT 給出不同的答案。

我在網路上收集資料後，選出了最適合提問 ChatGPT 的問題，並了解了提問 AI 的技巧和那些變因會影響 ChatGPT 給出不同答案後，我就針對不同時間、不同帳號、不同國家 IP、不同提問法和不同提問語言使用事實性問題、應用性問題和實作性問題三種提問類型，每個類型各三個問題，共九個問題進行提問。我想透過這個實驗更清楚的了解、驗證那些變因會影響 ChatGPT 給的回覆。

我在實驗中發現時間、國家 IP 和語言並不太會影響 ChatGPT 給的回覆，但國家 IP 和語言再查詢每個國家特有的文化、法規時，可能會因查詢的資料而所不同。帳號和提問法對 ChatGPT 給的回覆影響就很大了，使用愈多個人化資料的帳號給出的答案就會愈符合你個人的喜好，而使用愈精準的關鍵詞/指令 (Prompt) 可以讓 ChatGPT 給你更精準的回覆。

關鍵詞：ChatGPT、AI 提問

壹、緒論

一、研究動機

某一天我走在路上時碰巧聽到了一個很難的詞，但是當我回到家詢問爸媽時他們卻不知道那是什麼意思，剛好那時 ChatGPT 很紅，所以我就自己開了一個帳號去玩，想不到 ChatGPT 的回覆竟然淺顯易懂，所以自從那以後我就很喜歡使用 ChatGPT，遇到不會的問題就問他。

但是有一次我在外面上課時老師要我們請 ChatGPT 幫我們生成程式使 LED 閃爍，但我下的指令(prompt)跟老師大部分一樣，但我們的 ChatGPT 給出的指令卻不能運行，老師的卻可以，所以我就想來深入的探討一下到底有哪些變因會影響 ChatGPT 給不同的回覆。

二、研究目的

- (一)認識 ChatGPT
- (二)AI 提問技巧
- (三)針對不同變因向 ChatGPT 提問
- (四)自製檢核表檢視 ChatGPT 的回覆

貳、文獻探討

一、認識 ChatGPT

全名叫聊天生成預訓練轉換器(Chat Generative Pre-trained Transformer)是 openAI 在 2020 年推出的人工智慧聊天機器人程式，他可以用於自動生成文字、自動問答、自動摘要等多種任務，還有編寫和除錯電腦程式的能力，而且還可以透過灌輸一些資料來個人化(訓練)它。目前擁有 Chatgpt 4o、4o mini 等版本，各有不同優缺點，4o 是有限存取、ChatGPT 4o mini 則是免費使用，但因為 ChatGPT 版本更動頻繁，所以我的研究會使用實驗當下 ChatGPT 版本進行實驗，以下是他們的優缺點。

表 2-1 ChatGPT 4o、4o mini 的優缺點

	價格	差別
4o	免費、有限存取	一般
4o mini	免費、無限存取、便宜 60%	小型模型

註：研究者自行整理

(一)什麼原因會導致 ChatGPT 給出不同回覆

我還是查了一下有哪些原因會導致 ChatGPT 給出不同的回覆，我想看看別人發現了什麼原因會改變 ChatGPT 給出不同回覆，再加以測試、發掘新的原因。ChatGPT 可能會因以下幾個原因給出不同的回覆：

1. 上下文差異：如果你在不同的對話上下文中問同一個問題，模型可能會根據之前的對話內容給出不同的回應。
2. 溫度設置：生成模型的“溫度”參數會影響回覆的隨機性。較高的溫度會導致更具創造性但也更隨機的回覆，而較低的溫度則會產生更一致但可能較無趣的回應。
3. 模型更新：模型會定期更新，這可能導致它對相同問題的理解和回應方式發生變化。
4. 人化調整：根據你的使用情況和偏好，模型可能會逐漸調整回應的風格和內容，以更好地匹配你的需求。
5. 問題的複雜性：對於某些複雜或模糊的問題，模型可能會提供不同的解釋或觀點，因此回覆可能會有所不同。

(二)適合提問 ChatGPT 的問題

為了決定要提問 ChatGPT 什麼問題，我上網查了 ChatGPT 提問類型的分析。ChatGPT 的提問可以大致分成七種提問類型，但通常越沒有標準答案的問題得到的結果越佳，而且 ChatGPT 特別擅長「聯想連結」，像是把某個主題進行假設型問題「如果… 會怎樣」(What if)或著是把 A 領域的知識套用到 B 領域。以下是 8 種大致的 ChatGPT 提問類型：

1. 事實性問題：尋求客觀事實和已確定資訊的問題。例如：「水的化學式是什麼？」
2. 解釋性問題：需要提供深入解釋或分析的問題。例如：「為何經濟衰退會發生？」
3. 程序性問題：涉及具體操作或步驟的問題。例如：「如何進行科學實驗？」
4. 評價性問題：對某事進行價值判斷的問題。例如：「這部電影的藝術價值如何？」
5. 預測性問題：關於未來發展的問題。例如：「人工智能將如何影響就業市場？」
6. 概念性問題：旨在理解概念含義的問題。例如：「自由意志是什麼？」
7. 應用性問題：如何將理論應用於實際的問題。例如：「如何將教學理論應用於課程設計？」
8. 實作性問題：生成可以運行的程式碼並使用軟體運行。例如：「使用 python 令一個網頁版程式生成圖像。」

依據上述 ChatGPT 提問類型所收集到的資料，我將內容敘述的提問特點歸納成固定答案、正確結果、自由發揮、客觀、主觀五種類別，ChatGPT 提問類型與相對應符合的特點整理表格如下：

表 2-2 ChatGPT 提問類型的特點

ChatGPT 提問類型	固定答案	正確結果	自由發揮	客觀	主觀
事實性問題	V	X	X	V	X
解釋性問題	X	V	V	X	V
程序性問題	V	V	V	V	X
評價性問題	X	X	V	X	V
預測性問題	X	X	V	X	V
概念性問題	V	V	V	X	V
應用性問題	X	X	V	X	V
實作性問題	X	V	V	V	V

註：研究者自行整理

最後我決定使用事實性問題、應用性問題和實作性問題三種提問類型進行提問，因為事實性問題有正確答案且無容錯空間；應用性問題無正確答案可以自由發揮；實作性問題無正確答案但有正確結果，各自有它的特點。

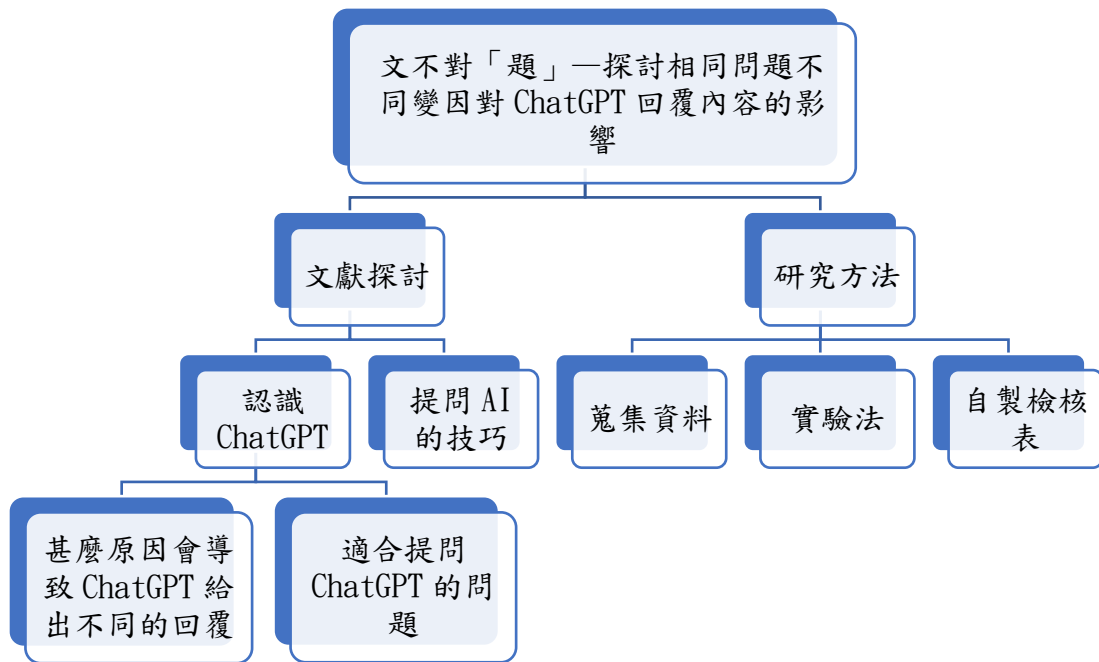
二、提問 AI 的技巧

在對 AI 進行提問時，有下列幾個提問的技巧，可以使 AI 更清楚明瞭提問者想提問的問題重點，同時收到的答覆也會比較符合問題的內容，提問技巧如下：

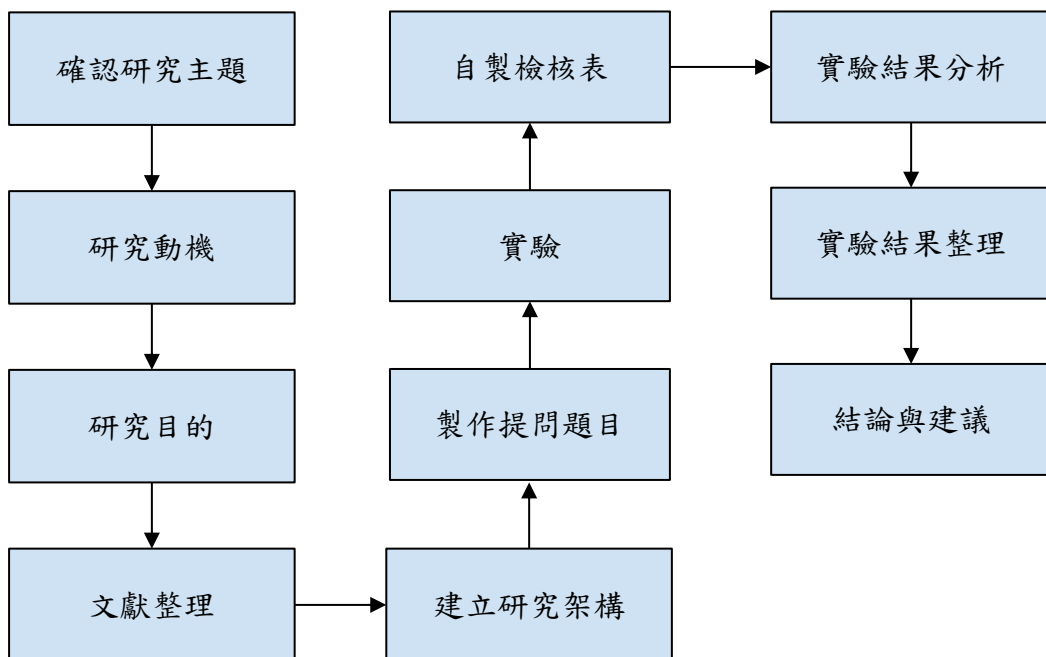
- (1) 清晰且具體的問題：提問時，確保問題簡單明了，避免過於複雜或模糊的表達。
- (2) 分解複雜問題：如果你有一個複雜的問題，將其分解為幾個小問題，這樣可以獲得更詳細和全面的回答。
- (3) 使用關鍵詞或提示：使用關鍵詞或給出明確的提示，幫助 AI 理解你需要的內容。
- (4) 要求範例或類比：當你不完全理解某個概念時，可以請求範例或類比來加深理解。
- (5) 讓 AI 解釋步驟或過程：如果你需要學習或理解某個過程，要求 AI 逐步解釋。

參、研究過程與方法

一、研究架構圖



二、研究步驟



三、研究方法

(一)實驗變因

根據研究目的，我的實驗變因內容說明如下：

1. 時間：我選擇我可以固定做研究的時間，有星期二下午 1 點至 2 點和星期五下午 3 點至 4 點。
2. 帳號(個人化資料)：我會以完整的個人資料、無資料以及造假資料去做實驗，因為 ChatGPT 會因為個人資料的完整度去個人化回覆。
3. 國家 IP：我選擇台灣、美國和日本的原因，是因為這三個國家是平均每人 IP 位置分配最多的國家。
4. 提問法：我會按照我說話的方式直接去敘述，從中了解 ChatGPT 給出的回覆是什麼。
5. 語言：我選擇中文、英文和西班牙文原因是因為這三個語言是世界最常用語言前三名。

表 3-1 實驗變因內容說明

實驗操縱變因	變因內容
時間	星期二下午 1 點至 2 點、星期五下午 2 點至 3 點
帳號 (個人化資料)	主帳號(最貼近我個人的資料)、實驗帳號(無資料和造假資料)
國家 IP	台灣、美國、日本
提問法	按照我說話的方式直接敘述、《ChatGPT 指令大全》的公式進行修改
語言	中文、英文、西班牙文

註：研究者自行整理

(二)評分表

根據我的實驗所問的問題，我需要去評斷 ChatGPT 的回覆是否符合我的預期，所以我針對文本內容設計了以下幾個評分項目，內容如下：

1. 相關聯性：回覆內容與問題的相關聯性，使用百分比做為評分標準。例如：回覆內容文不對題時給予一分，回覆內容與問題的相關性愈高分數愈高。（資料有誤在此扣分）
2. 段落鮮明度：回覆的段落鮮明度，使用百分比做為評分標準。例如：回覆內容沒有架構性無分明顯段落給予一分，如果段落鮮明且分為起、承、轉、合就可評為五分。
3. 用字遣詞：回覆用字遣詞的精準、適當與清楚程度。例如：回覆裡的用詞如果不適當或有更好的詞可代替時按比例給予一至四分。（用詞有誤在此扣分）
4. 內容豐富度：因為 C1~C3 問題著重於程式碼是否可執行於 CodePen，故此評分項目僅限 A1~B3 問題使用。回覆的豐富程度，使用百分比做為評分標準。例如：當回覆內容越豐富、越多補充，會給予越高的分數。
5. 執行結果：執行結果主要是檢視 ChatGPT 回覆的程式碼是否可執行並達到要求功能的程度，故此評分項目僅限 C1~C3 問題使用。此項目使用百分比做為評分標準。例如：回覆的程式碼如果可以執行且問題要求的功能都有達到就可評為五分；反之，如果沒有達到所有功能，會視情況給予一至四分。

表 3-2 實驗評分表

項目	1 分	2 分	3 分	4 分	5 分
相關聯性	問題與回覆內容相關性 0%	問題與回覆內容相關性 25%	問題與回覆內容相關性 50%	問題與回覆內容相關性 75%	問題與回覆內容相關性 100%
段落鮮明度	段落鮮明度百分比 0%	段落鮮明度百分比 25%	段落鮮明度百分比 50%	段落鮮明度百分比 75%	段落鮮明度百分比 100%
用字遣詞	用字遣詞不精準、不適當、不清楚	用字遣詞大部分不精準、不適當、不清楚	用字遣詞有一半精準、適當、清楚	用字遣詞大部分精準、適當、清楚	用字遣詞精準、適當、清楚

內容豐富度	回覆內容豐富程度 0%	回覆內容豐富程度 25%	回覆內容豐富程度 50%	回覆內容豐富程度 75%	回覆內容豐富程度 100%
執行結果	無法執行	可以執行，達到指令要求 25%	可以執行，達到指令要求 50%	可以執行，達到指令要求 75%	可以執行，達到指令要求 100%

註：研究者自行整理

四、提問問題

根據文獻探討的結果，我會使用事實性問題、應用性問題、實作性問題三種類型進行提問，而每種問題類型會問三種不同的問題，問題一是程式分類的相關問題，問題二是數學內容相關的問題，問題三是有關化學的問題，為使實驗方便記錄，我將提問的類型以分別以 A、B、C 作為代號，問題內容整理如下：

表 3-2 提問問題一覽表(按照說話的方法敘述)

提問問題類型	事實性問題 (A)	應用性問題 (B)	實作性問題 (C)
問題一	A1：C# 是一種程式語言嗎？	B1：如何教一個完全沒有程式基礎的人寫 python 程式？	C1：生成一個可以重複操作且有起始畫面的 python 射擊遊戲程式並可應用在 CodePen 網頁中。
問題二	A2：數學有幾年的歷史？	B2：如何將數學原理應用於生活之中？	C2：生成一個可以透過使用者的輸入來解讀一元一次方程式的 python 一元一次方程式計算機程式並可應用在 CodePen 網頁中。
問題三	A3：水的化學式是什麼？	B3：如何將科學原理應用於生活之中？	C3：生成一個元素週期表有元素的名字、代號與分類的 python 元素週期表程式，且點擊後會出現該元素的解釋、英文、分類、代號、名稱，同時此程式可應用在 CodePen 網頁中。

註：研究者自行整理

上方表格中所提出的問題是從我的學習內容中產生出的疑問，而我將這些疑問分為事實性問題、應用性問題和實作性問題，藉由我這次的獨立研究向

ChatGPT 提問來獲得解答。

依據 ChatGPT 的回覆，我會將每個回覆傳達一樣訊息的地方使用粗體標記，不一樣的地方就使用原本的字體，且如有用字或資料不當會使用紅色來標記。

肆、研究結果與討論

一、實驗一：實驗組(時間)

本實驗以實驗的時間作為我的操縱變因，有星期二下午 1~2 點和星期五下午 3~4 點兩個時段，控制變因會以主帳號、台灣 IP 以及我說話的方式以中文去敘述向 ChatGPT 提問，以下是我的提問問題以及 ChatGPT 的回覆：

表 4-1-1 時間組 A1

	A1：C#是一種程式語言嗎？	
實驗時間	星期二下午 1~2 點	星期五下午 3~4 點
回覆	是的，C#（發音為 "C-sharp"）是一種程式語言。它由微軟開發，主要用於構建各種應用程序，特別是 Windows 平台上的應用程式。C# 是基於 .NET 框架的一種現代、物件導向的語言，適用於桌面應用程式、網站、移動應用程式等開發。 它的語法類似於 C 和 C++，但在設計上更加簡單和安全。C# 支援許多現代程式語言的特性，例如垃圾回收、異常處理和多線程處理等。	是的，C#（發音為 "C-sharp"）是一種程式語言。它由微軟開發，主要用於 .NET 平台的應用程式開發。C# 是一種現代化、面向對象的語言，具有簡潔且強大的語法，並支援多種編程範式，如面向對象編程、事件驅動編程等。C# 廣泛應用於桌面應用程式、Web 應用程式和遊戲開發等領域。
相關聯性	5/5	5/5
段落鮮明度	4/5	3/5
用字遣詞	5/5	5/5
內容豐富度	5/5	4/5
總分	19/20	17/20

註：研究者自行整理

此問題總分只差兩分，其回復差別在於補充的數量與分段，主要回覆內容差距並不大。

表 4-1-2 時間組 A2

	A2：數學有幾年的歷史？	
實驗時間	星期二下午 1~2 點	星期五下午 3~4 點
回覆	數學的歷史可以追溯到古代文明，約有 4000 多年的歷史。最早的數學記錄來自古埃及和古巴比倫，這些文化在公元前 2000 年左右已經開始使用數字系統來解決日常問題，如測量土地和建造建築物。 隨著時間的推移，古希臘數學家如畢達哥拉斯和歐幾里得等人發展了更為抽象的數學理論，這些理論成為現代數學的基礎。數學經過幾千年的發展，形成了今天我們所知的各種分支，如算術、幾何學、代數、微積分等。 總的來說，數學的歷史非常悠久，跨越了多個世紀，並且隨著人類文明的進步不斷演變和發展。	數學的歷史可以追溯到大約公元前 3000 年左右的古埃及和古巴比倫時期。當時的人們已經開始發展數字系統、計算方法和幾何學。最早的數學紀錄包括對天文、建築、貿易等方面的應用。隨著時間的推移，古希臘數學家如歐幾里得、阿基米德等也對數學的發展做出了重要貢獻。因此，數學有大約五千多年的歷史。
相關聯性	5/5	5/5
段落鮮明度	4/5	2/5
用字遣詞	4/5	5/5
內容豐富度	5/5	4/5
總分	18/20	16/20

註：研究者自行整理

此問題 ChatGPT 的回覆要傳達的內容並沒有差很多，且多的部分只是總結而已，但問題一有些許不恰當的詞，所以分數沒有差很多。

表 4-1-3 時間組 A3

	A3：水的化學式是什麼？	
實驗時間	星期二下午 1~2 點	星期五下午 3~4 點
回覆	水的化學式是 H_2O 。這表示每個水分子由兩個氫原子 (H) 和一個氧原子 (O) 組成。	水的化學式是 H_2O 。這表示水分子由兩個氫原子 (H) 和一個氧原子 (O) 組成。
相關聯性	5/5	5/5
段落鮮明度	1/5	1/5
用字遣詞	5/5	5/5
內容豐富度	1/5	1/5
總分	12/20	12/20

註：研究者自行整理

回覆的內容有些許差別但在語句的意思上無差別。

表 4-1-4 時間組 B1

	B1：如何教一個完全沒有程式基礎的人寫 python 程式？	
實驗時間	星期二下午 1~2 點	星期五下午 3~4 點
回覆	教一個完全沒有程式基礎的人寫 Python 程式可以分成幾個階段，每個階段注重基礎概念、實踐練習，逐步建立理解。以下是一個簡單的教學架構：1. 介紹程式和 Python - 什麼是程式？：解釋程式是用來讓電腦執行指令的語言，程式可以解決問題、做計算，甚至建立應用程式。 - 為什麼選 Python？：Python 是簡單且易讀的語言，適合初學者，且有大量的資源和社群支持。 2. 基礎語法介紹 - 變數與資料類型： - 變數：如何儲存數字、文字等資料。 - 常見資料類型：整數 (int)、浮點數 (float)、字串 (str)、布林值 (bool)。 - 範例：age = 25, name = "Alice", is_active = True - 簡單運算： 加、減、乘、除、餘數等。 - 範例：x = 10 + 5, y = 7 * 2 - 輸入與輸出： - 使用 input() 來接受使用者輸入，使用 print() 來顯示輸出。 - 範例： <pre>name = input("What's your name? ") print("Hello, " + name)</pre> 3. 流程控制 - 條件語句 (if/else)： - 判斷某些條件是否為真，來決定要執行的操作。 - 範例： <pre>age = 18 if age >= 18: print("You are an adult.") else: print("You are a minor.")</pre>	教一個完全沒有程式基礎的人寫 Python 程式，可以依照以下步驟進行： - 介紹程式語言的基本概念 先從基本概念開始： - 什麼是程式語言，程式語言是用來與電腦溝通的工具。 - 解釋「指令」的概念，例如：程式就是一條條指令，讓電腦執行特定任務。 - 安裝 Python 環境 教學如何安裝 Python 及編輯器（如 VS Code 或 Jupyter Notebook），並解釋如何運行簡單的程式。 - 講解基本語法 從 Python 最基本的語法開始： - 輸出：print() 函數用來顯示訊息。 <pre>print("Hello, world!")</pre> - 變數：介紹變數和資料型別（例如，int, float, str）。 <pre>name = "Alice" age = 25 print(name, age)</pre> - 條件語句 教如何根據條件執行不同的指令： - 使用 if, else 等語句。 <pre>if age > 18: print("成年人") else: print("未成年")</pre> - 迴圈 介紹如何用迴圈重複執行某些指令： - 使用 for 迴圈。 <pre>for i in range(5): print(i)</pre> - 函數

	- 迴圈 (for 和 while) : - 用於重複執行某些操作。 - 範例： <pre>for i in range(5): print(i)</pre> 4. 簡單的實踐項目 - 計算器：讓學生寫一個簡單的加法、減法、乘法、除法計算器。 - 例子： <pre>num1 = float(input("Enter first number: ")) num2 = float(input("Enter second number: ")) operation = input("Choose operation (+, -, *, /): ") if operation == "+": print(num1 + num2) elif operation == "-": print(num1 - num2) elif operation == "*": print(num1 * num2) elif operation == "/": print(num1 / num2) else: print("Invalid operation")</pre> 5. 進一步的學習 函數：教學生如何將程式碼分成功能模塊，並使用函數來組織程式。 資料結構：介紹列表、字典等資料結構，讓學生能夠處理複雜的資料。 6. 持續練習與挑戰 鼓勵學生多寫程式，從簡單的問題開始，如計算某個數字的階乘、列出所有質數等。隨著技能進步，可以挑戰更複雜的項目。 7. 資源與學習工具 - 學習網站：如 Codecademy、LeetCode 等。 - 影片教學：YouTube 上有很多適合初學者的 Python 教學。 - 書籍推薦：《Python Crash Course》、《Automate the Boring Stuff with Python》。 學習程式是一個循序漸進的過程，鼓勵學生多動手寫程式，從錯誤中學習，並持續探索。	介紹函數的概念，教如何定義簡單的函數來重複使用程式碼： <pre>def greet(name): print("Hello, " + name) greet("Alice")</pre> 7. 簡單的專案 鼓勵學員做一些簡單的專案，例如： - 計算機：可以讓用戶輸入數字，然後進行加減乘除運算。 - 小遊戲：例如猜數字遊戲。 8. 提供練習和挑戰 透過練習來加強學習。挑戰學員編寫一些簡單的程式，逐漸增加難度。 在這過程中，重點是讓學員理解基礎概念，並且能夠逐步完成簡單的程式，對他們建立信心。
相關聯性	5/5	5/5
段落鮮明度	5/5	5/5
用字遣詞	5/5	5/5
內容豐富度	5/5	4/5
總分	20/20	19/20

註：研究者自行整理

兩個回覆的內容大致上沒有差很多，但對照組有較多補充，所以在內容豐富度上會有較高的分數。

表 4-1-5 時間組 B2

	B2：如何將數學原理應用於生活之中？	
實驗時間	星期二下午 1~2 點	星期五下午 3~4 點
回覆	將數學原理應用於日常生活中，其實是非常有用的。以下是一些例子，說明如何將數學概念與生活相結合： 1. 時間管理與計劃：利用數學中的比例、分數和比率來規劃時間。例如，假設你有三個任務，並且知道它們的完成時間，則可以計算每個任務的優先順序，並確保時間分配最有效。 2. 預算與財務規劃：在生活中，每月的收入和支出都需要計算。利用加法、減法、乘法和除法來管理你的預算，確保收支平衡。此外，了解利率和貸款計算也可以幫助你做出更聰明的財務決策。 3. 購物折扣：當你購物時，可以利用百分比來計算折扣。例如，如果一件衣服打七折，原價 1000 元，那麼你可以計算折後價格是 1000 元乘以 0.7，即 700 元。 4. 烹飪與食譜：做菜時，你常常需要調整食譜中的比例。這是應用比率和比例的方法。如果食譜是為 4 人設計的，而你是為 2 人做飯，你可以將所有材料的量減半。 5. 運動與健康：在運動時，數學可幫助你計算心率、卡路里消耗等。對於體重訓練，計算每組的重量和次數可以幫助你確保訓練達到目標。 6. 空間與物體擺放：當你整理房間或搬家時，數學中的幾何學能幫助你計算空間大小，並計算出最佳的擺放方式。 7. 交通與路程規劃：如果你想要從 A 地點到 B 地點，計算路程所需時間的速度和距離公式可以幫助你找到最佳路徑或了解交通狀況。 這些數學概念不僅能提高你的生活效率，還能讓你在解決日常問題時更加精確和有條理。	將數學原理應用於生活中可以幫助我們更有效地解決問題並作出決策。以下是一些例子： 1. 預算和財務管理：使用算術和比例來計算支出和收入，幫助你制定合理的預算，並確保不超支。 2. 時間管理：使用時間加減法來安排日程，或使用比例和百分比來確定各項活動的時間分配。 3. 購物折扣和比較價格：當購物時，使用百分比來計算折扣後的價格，或者利用簡單的數學計算來比較不同商店的價格，以獲得最佳的交易。 4. 建設與裝修：在家居設計或裝修時，利用幾何學原理來計算空間的面積、體積，甚至是選擇材料的數量。 5. 健康與飲食：使用比例來計算卡路里攝入量，或者通過數據分析來追蹤健康指標，例如體重、運動時間等。 數學無處不在，從基本的計算到更複雜的統計和概率，都能幫助我們提高生活效率和精確度。
相關聯性	5/5	5/5
段落鮮明度	5/5	5/5
用字遣詞	5/5	5/5
內容豐富度	5/5	4/5
總分	20/20	19/20

註：研究者自行整理

總分的一分差距在於內容豐富度，但對於我期望得到的答覆並無太大影響。

表 4-1-6 時間組 B3

	B3：如何將科學原理應用於生活之中？	
實驗時間	星期二下午 1~2 點	星期五下午 3~4 點
回覆	將科學原理應用於日常生活中，可以幫助我們更有效地解決問題，提升生活品質。以下是一些實際的例子： 1. 熱力學原理： ○ 冰箱和空調：這些家用電器利用熱力學中的熱交換原理來保持食物冷卻或創造舒適的室內環境。冰箱將熱量從內部轉移到外部，這是基於熱流動的基本原理。 2. 摩擦力：	將科學原理應用於日常生活中，可以幫助我們更有效地解決問題、提高效率，甚至提升生活質量。以下是一些例子： 1. 物理學原理： ○ 利用杠杆原理來減輕搬運重物的壓力。例如，使用撬棍來撬起重物或提升物品。 ○ 利用浮力原理讓船隻或潛水艇浮在水面上。這也能解釋為什麼物體在水中會浮起或沉入。 2. 化學原理：

	○ 穿鞋：選擇適合的鞋子可以減少摩擦，讓走路更加輕鬆。運動鞋的設計通常會利用摩擦原理來增加穩定性，防止滑倒。 3. 浮力原理： ○ 游泳：當你進入水中時，你會感受到一股向上的力量，這是由浮力原理引起的。根據阿基米德原理，物體在液體中所受的浮力等於它排開的液體的重力。 4. 光學原理： ○ 眼鏡和鏡子：眼鏡根據光的折射原理幫助我們改善視力，鏡子利用光的反射原理來形成影像。 5. 生物學原理： ○ 營養和健康：了解人體如何消化食物和吸收營養，可以幫助你制定更健康的飲食計劃，保持身體健康。例如，蛋白質對肌肉生長至關重要，而水則對體內的各項生理反應不可或缺。 6. 物理學原理： ○ 滑板：在滑板上滑行時，利用動能和慣性原理可以讓你輕鬆地保持平衡和移動。學會如何運用這些物理概念能夠讓你更有效地進行運動。 將科學原理融入生活中，不僅能讓你更好地理解世界，還能提升你的生活效率和舒適度。	○ 使用酸鹼中和反應來清潔。像是用醋或檸檬水清潔水垢，這些酸性物質可以溶解碳酸鹽結構。 ○ 烹飪中的化學反應，如高溫下食物的美拉德反應，能夠讓食物變得更加美味和有色澤。 3. 生物學原理： ○ 健康飲食基於營養學原理，能幫助維持體內平衡，改善免疫系統的功能。 ○ 了解人類的睡眠週期和生物鐘，改善睡眠質量，提升白天的精力和集中力。 4. 數學原理： ○ 利用概率來制定最佳的購物策略或賭博策略。了解不同情境下如何做出理性決策。 ○ 使用統計學來分析日常生活中的數據，從而做出更加明智的選擇，例如健康監測數據或花費預算。 科學原理能幫助我們在各種情境中做出更好的決策，使我們的生活更加高效和智慧。
相關聯性	5/5	5/5
段落鮮明度	5/5	5/5
用字遣詞	5/5	4/5
內容豐富度	5/5	4/5
總分	20/20	18/20

註：研究者自行整理

差的兩分在於內容豐富度與用字遣詞不當，但對於我期望得到的答覆並無太大影響。

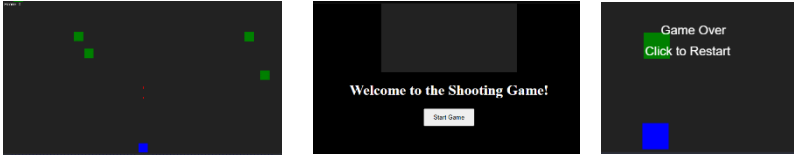
表 4-1-7 時間組 C1

	C1：生成一個可以重複操作且有起始畫面的 python 射擊遊戲程式並可應用在 CodePen 網頁中。	
實驗時間	星期二下午 1~2 點	星期五下午 3~4 點
回覆	要在 CodePen 中實現一個可以重複操作且有起始畫面的 Python 射擊遊戲，首先，您需要將 Python 程式碼轉換為 JavaScript，因為 CodePen 是基於 Web 技術（HTML, CSS, JavaScript）來運行的，無法直接執行 Python 程式碼。 我會提供一個簡單的 JavaScript 程式碼範例，這個範例包含了射擊遊戲的基本結構，包括起始畫面、遊戲邏輯，以及重複操作的機制。您可以將其放入 CodePen 的 JS 部分來測試。	要在 CodePen 中運行 Python 代碼並實現重複操作的射擊遊戲，我們需要將 Python 轉換為可以在瀏覽器中運行的語言（例如 JavaScript）。然而，CodePen 本身不支持 Python，僅支持 HTML, CSS 和 JavaScript。 所以，這裡我提供一個用 JavaScript 實現的簡單射擊遊戲的範例代碼。你可以將這段代碼粘貼到 CodePen 中運行。這個遊戲有一個起始畫面，並支持玩家控制一個小飛機進行射擊。 <pre> <!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>Shooting Game</title> <style> body { margin: 0; overflow: hidden; background-color: #1e1e1e; } canvas { display: block; margin: auto; background-color: #000; } .start-screen { position: absolute; top: 50%; left: 50%; transform </style> </head> <body> <div class="start-screen" id="start-screen"> <h1>Press Space to Start</h1> </div> </body> </pre> 如何使用：

	<pre><!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>Simple Shooting Game</title> <style> body { margin: 0; overflow: hidden; background-color: black; } canvas { display: block; margin: 0 auto; background-color: #222; } .hidden { display: none; } </style> </head> <body> <canvas id="gameCanvas"></canvas> <div id="startScreen" class="hidden"> <h1 style="color: white; text-align: center;">Welcome to the Shoot! <button style="display: block; margin: 0 auto; padding: 10px 20px;">Start Game </div> <script> const canvas = document.ge const ctx = canvas.getConte </script></pre> 這段代碼的功能： 起始畫面：當頁面加載完成後，顯示一個歡迎界面，包含一個按鈕可以開始遊戲。 遊戲邏輯： - 玩家可以使用左/右箭頭鍵控制移動，空格鍵發射子彈。 - 敵人會從畫面上方隨機生成並向下移動。 - 當子彈擊中敵人時，得分增加，敵人被摧毀。 - 如果敵人觸及畫面底部，遊戲結束，顯示 "Game Over" 和重啟提示。 重複操作：遊戲結束後，玩家可以點擊畫面來重新開始遊戲。 使用說明： - 將這段代碼粘貼到 CodePen 的 HTML 和 JS 部分即可運行。 - 點擊 "Start Game" 開始遊戲，使用箭頭鍵移動，按空格鍵發射子彈。 - 當遊戲結束後，點擊畫面可以重新開始。 這個遊戲是基本版本，您可以根據需要進行擴展，例如添加音效、更複雜的敵人行為和圖像等。	- 將此代碼粘貼到 CodePen：只需將代碼放到 HTML 部分中，並且該代碼將自動運行。 - 遊戲操作： - 使用左右箭頭鍵來控制飛機的移動。 - 按空格鍵發射子彈。 這個遊戲有一個簡單的起始畫面（按空格鍵開始遊戲），並且當遊戲開始後，玩家可以控制飛機並與敵人發生碰撞來進行擊敗。
相關聯性	5/5	4/5
段落鮮明度	5/5	5/5
用字遣詞	5/5	5/5
執行結果	5/5	5/5
總分	20/20	19/20

註：研究者自行整理

對照組(左)可以執行且我要求的功能都有做到。



時間組(右)也可以執行，只是功能方面對照組做得比較好，所以對照組在相關聯性裡會有比較高的分數。

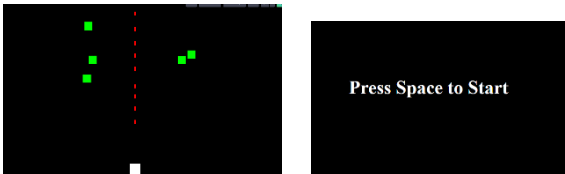


表 4-1-8 時間組 C2

	C2：生成一個可以透過使用者的輸入來解讀一元一次方程式的 python 一元一次方程式計算機程式並可應用在 CodePen 網頁中。	
實驗時間	星期二下午 1~2 點	星期五下午 3~4 點
回覆	要在 CodePen 中使用 Python 寫一個簡單的計算機來解一元一次方程式，通常會需要將 Python 程式轉換為 JavaScript，因為 CodePen 是基於 Web 的前端平台，它只支持 HTML、CSS、JavaScript，而不能直接運行 Python 代碼。 下面是一個用 JavaScript 寫的解一元一次方程式的計算機程式，可以應用於 CodePen 中。這個程式允許使用者輸入方程式的係數並計算出解。 <pre><!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>一元一次方程式計算機</title> </head> <body> <div> <h2>解一元一次方程式: ax + b = 0</h2> <form id="equation-form"> <label for="a"> (係數) </label> <input type="number" id="a" required</input> <label for="b"> (常數項) </label> <input type="number" id="b" required</input> <button type="submit">計算解</button> </form> <div>方程式解:</div> <div id="result"></div> </div> </body> </html></pre> 程式說明： - 用戶會被要求輸入方程式的兩個係數 a 和 b，這是來自方程式 ax + b = 0。 - 當用戶點擊 "計算解" 按鈕時，表單會觸發 submit 事件並進行計算。 - 程式會檢查 a 是否為零，如果是零，則顯示 "方程式無解" 的提示。 - 如果 a 不為零，計算並顯示方程式的解 x = -b/a。 你可以將這段代碼直接粘貼到 CodePen 的 HTML 區域，並運行它來計算一元一次方程式的解。	要在 CodePen 上運行這個一元一次方程式計算機，你需要使用 HTML、CSS 和 JavaScript。由於 CodePen 支持這些語言，我將提供一個簡單的程式碼範例，幫助你在網站上進行這項工作。 以下是簡單的計算機程式，它可以根據使用者輸入的方程式自動解出 x 的值。 HTML (在 CodePen 的 HTML 部分中) <pre><div> <h2>一元一次方程式計算機</h2> <p>請輸入一元一次方程式 (例如: 2x + 3 = 7):</p> <input type="text" id="equation" placeholder="例如: 2x + 3 = 7"> <button onclick="solveEquation()">解方程式</button> <p id="result"></p> </div></pre> CSS (在 CodePen 的 CSS 部分中) <pre>body { font-family: Arial, sans-serif; } function solveEquation() { // 獲取使用者輸入的方程式 const equation = document.getElementById('equation').value; const resultElement = document.getElementById('result'); // 正規表達式用於提取方程式的係數 const regex = /([+-]?[0-9]*)([+-]?[0-9]*)([+-]?[0-9]*)([+-]?[0-9]*)/; const match = equation.match(regex); if (match) { const coefficient = match[1] === '' ? 1 : match[1]; const constant = match[2] === '' ? 1 : match[2]; const result = -constant / coefficient; resultElement.textContent = `解: x = \${result}`; } else { resultElement.textContent = '方程式格式不正確'; } }</pre> JavaScript (在 CodePen 的 JavaScript 部分中) <pre>function solveEquation() { // 獲取使用者輸入的方程式 const equation = document.getElementById('equation').value; const resultElement = document.getElementById('result'); // 正規表達式用於提取方程式的係數 const regex = /([+-]?[0-9]*)([+-]?[0-9]*)([+-]?[0-9]*)([+-]?[0-9]*)/; const match = equation.match(regex); if (match) { const coefficient = match[1] === '' ? 1 : match[1]; const constant = match[2] === '' ? 1 : match[2]; const result = -constant / coefficient; resultElement.textContent = `解: x = \${result}`; } else { resultElement.textContent = '方程式格式不正確'; } }</pre> 如何使用： - 在 CodePen 創建一個新的 Pen。 - 將上面的 HTML 代碼粘貼到 HTML 部分。 - 將 CSS 代碼粘貼到 CSS 部分。 - 將 JavaScript 代碼粘貼到 JavaScript 部分。 這段代碼能夠讓使用者輸入一個形如 2x + 3 = 7 的方程式，並計算並顯示 x 的值。如果輸入格式不正確，它會提示使用者正確的格式。
相關聯性	4/5	5/5
段落鮮明度	5/5	5/5
用字遣詞	5/5	5/5
執行結果	5/5	5/5
總分	19/20	20/20

註：研究者自行整理

對照組(左)和時間組(右)都可以執行且都有達到要求的功能，只是時間組功能做的比較完善，所以在相關聯性會有比較高的分數。

解一元一次方程式: $ax + b = 0$

a (係數):

b (常數項):

方程式解:

方程式的解為 $x = -0.6666666666666666$

一元一次方程式計算機

請輸入一元一次方程式 (例如: $2x + 3 = 7$):

解: $x = 0.3333333333333333$

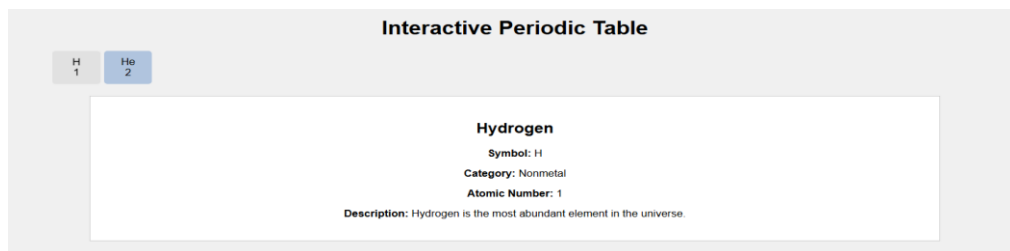
表 4-1-9 時間組 C3

	C3：生成一個元素週期表有元素的名字、代號與分類的python 元素週期表程式，且點擊後會出現該元素的解釋、英文、分類、代號、名稱，同時此程式可應用在 CodePen 網頁中。	
實驗時間	星期二下午 1~2 點	星期五下午 3~4 點
回覆	要在 CodePen 中使用 Python 並生成一個元素週期表，通常 Python 本身無法直接在瀏覽器中運行。你可以使用 HTML、CSS 和 JavaScript 來實現這個需求。這裡是如何用這三種語言創建一個動態元素週期表的範例： 1. HTML（結構） <pre><!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>Periodic Table</title> <link rel="stylesheet" href="styles.css"> </head> <body> <div id="periodic-table"> <!-- Elements will be inserted dynamically by JavaScript --> </div> <div id="element-details"> <h2 id="element-name">Element Name</h2> <p>Symbol: </p> <p>Category: </p> <p>Atomic Number: </p> <p>Description: </p> </div> </body> </html></pre> 2. CSS（樣式） <pre>body { font-family: Arial, sans-serif; background-color: #f0f0f0; } const elements = [{ name: "Hydrogen", symbol: "H", category: "Nonmetal", number: 1, de { name: "Helium", symbol: "He", category: "Noble Gas", number: 2, de // Add other elements here...]; function createElementButton(element) { const button = document.createElement('div'); button.className = 'element'; button.style.backgroundColor = getCategoryColor(element.category); }</pre> 3. JavaScript（功能） 詳解： <pre>const elements = [{ name: "Hydrogen", symbol: "H", category: "Nonmetal", number: 1, description: "Hydrogen is { name: "Helium", symbol: "He", category: "Noble Gas", number: 2, description: "Helium is // Add other elements here...]; function createElementButton(element) { // ... }</pre> ● HTML：設置了頁面結構，並創建了一個元素容器（.table）和一個顯示元素詳細信息的區域（.element-details），後者將顯示點擊的元素的詳細信息。 ● CSS：提供了基本的樣式，使元素以網格式形式顯示並進行了一些動效設置（如：元素縮放的懸停效果）。 ● JavaScript：創建了每個元素的按鈕，並將其顯示在表格中。當用戶點擊某個元素時，該元素的詳細信息會顯示在下方。 說明： ● 在 elements 陣列中，您可以根據需要添加更多的元素，每個元素包含名稱、符號、類別、原子數和描述。 ● getCategoryColor 函數根據元素的類別返回不同的顏色（可以擴展更多的類別和顏色）。 ● 當您點擊某個元素時，JavaScript 會動態顯示該元素的詳細信息。 如何在 CodePen 中使用： 1. 創建一個新的 Pen。	為了在 CodePen 中創建一個可以展示元素週期表的程序，我們需要結合 HTML、CSS 和 JavaScript 來實現這個功能。下面是如何實現此功能的步驟和代碼： 1. HTML 部分：創建元素週期表結構 <pre><!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>Interactive Periodic Table</title> <link rel="stylesheet" href="styles.css"> </head> <body> <div id="periodic-table"> <!-- Elements will be dynamically inserted here --> </div> <div id="element-details"> <h2 id="element-name"></h2> <p>Symbol: </p> <p>Category: </p> </div> </body> </html></pre> 2. CSS 部分：設計元素的顯示樣式 <pre>#periodic-table { // ... const elements = [{ name: 'Hydrogen', symbol: 'H', category: 'Nonmetal', color: '#ffcccb', { name: 'Helium', symbol: 'He', category: 'Noble gas', color: '#e0e0e0', { name: 'Lithium', symbol: 'Li', category: 'Alkali metal', color: '#ffcc // Add more elements here] } .element { // ... }</pre> 3. JavaScript 部分：元素資料、交互功能與顯示 <pre>const elements = [{ name: 'Hydrogen', symbol: 'H', category: 'Nonmetal', color: '#ffcccb', { name: 'Helium', symbol: 'He', category: 'Noble gas', color: '#e0e0e0', { name: 'Lithium', symbol: 'Li', category: 'Alkali metal', color: '#ffcc // Add more elements here]; function createElementButton(element) { const button = document.createElement('div'); }</pre> 4. 如何應用在 CodePen 1. HTML 部分：複製上面的 HTML 代碼並粘貼到 CodePen 的 HTML 區域。 2. CSS 部分：複製上面的 CSS 代碼並粘貼到 CodePen 的 CSS 區域。 3. JavaScript 部分：複製上面的 JavaScript 代碼並粘貼到 CodePen 的 JS 區域。 這樣，你就能在 CodePen 中看到一個可互動的元素週期表。每當你點擊一個元素，它會顯示該元素的詳細資訊，包括名稱、符號、分類和描述。

	2. 將 HTML 放入 HTML 編輯器中。 3. 將 CSS 放入 CSS 編輯器中。 4. 將 JavaScript 放入 JavaScript 編輯器中。 這樣，您就能在 CodePen 中看到互動式元素週期表，並能查看每個元素的詳細信息。	
相關聯性	5/5	5/5
段落鮮明度	5/5	5/5
用字遣詞	5/5	5/5
執行結果	2/5	2/5
總分	17/20	17/20

註：研究者自行整理

對照組可以執行但只有兩個元素，所以只能給兩分。



時間組雖然沒有給出原子序，但有多說明一個元素，兩者的重要程度互相抵消，所以執行結果的分數都一樣，總分數也都一樣，所以差距不大。

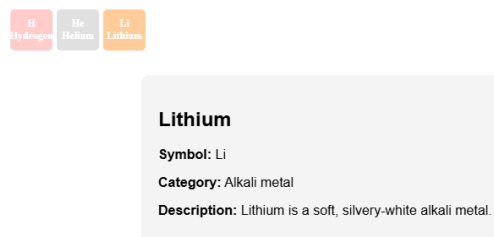


表 4-1-10 實驗組(時間)評分統整表

問題	總分	
	星期二下午 1~2 點	星期五下午 3~4 點
A1	19/20	17/20
A2	18/20	16/20
A3	12/20	12/20
B1	20/20	19/20
B2	20/20	19/20
B3	20/20	18/20
C1	20/20	19/20
C2	19/20	20/20
C3	17/20	17/20
總共	165/180	157/180

註：研究者自行整理

總分差距不大，差距在於資料內容的多寡，和分段。推測 ChatGPT 的回覆與提問時間並無太大關聯。

二、實驗二：實驗組(帳號)

本實驗以帳號作為我的操縱變因，有主帳號、無資料帳號和造假資料三個帳號，控制變因會以實驗時間為每星期二下午 1~2 點、台灣 IP 以及我說話的方式以中文去敘述向 ChatGPT 提問，以下是我的提問問題以及 ChatGPT 的回覆：

表 4-2-1 帳號組 A1

	A1：C#是一種程式語言嗎？		
實驗帳號	主帳號	無資料	造假資料
回覆	是的，C#（發音為 "C-sharp"）是一種程式語言。它由微軟開發，主要用於構建各種應用程序，特別是 Windows 平台上的應用程序。C# 是基於 .NET 框架的一種現代、物件導向的語言，適用於桌面應用程式、網站、移動應用程式等開發。 它的語法類似於 C 和 C++，但在設計上更加簡單和安全。C# 支援許多現代程式語言的特性，例如垃圾回收、異常處理和多線程處理等。	是的，C#（發音為「C-Sharp」）是一種程式語言。它是由微軟（Microsoft）於 2000 年所開發，並且是 .NET 框架的一部分。C# 是一種多用途、面向對象的程式語言，主要用於開發桌面應用程式、Web 應用程式、手機應用程式以及遊戲開發（特別是使用 Unity 引擎）。它的語法與 C++ 和 Java 有很多相似之處，並且支援現代編程特性，如垃圾回收、異常處理和事件驅動的編程模式。	是的，C#（發音為 "C-sharp"）是一種程式語言。它是由微軟開發的，並且通常與 .NET 框架一起使用。C# 是一種物件導向的語言，設計用來簡化程式設計並提供高效能的應用程式開發。C# 被廣泛用於桌面應用程式、網站、遊戲開發（特別是在 Unity 引擎中）以及企業級應用程式。
相關聯性	5/5	5/5	5/5
段落鮮明度	5/5	4/5	4/5
用字遣詞	5/5	5/5	5/5
內容豐富度	5/5	4/5	3/5
總分	20/20	18/20	17/20

註：研究者自行整理

三者的差異不大，但資料量由大到小分別是主帳號、無資料、造假資料，且主帳號段落最完整所以會有較高的分數。

表 4-2-2 帳號組 A2

	A2：數學有幾年的歷史？		
實驗帳號	主帳號	無資料	造假資料
回覆	數學的歷史可以追溯到古代文明，約有 4000 多年的歷史。最早的數學記錄來自古埃及和古巴比倫，這些文化在公元前 2000 年左右已經開始使用數字系統來解決日常問題，如測量土地和建造建築物。 隨著時間的推移，古希臘數學家如畢達哥拉斯和歐幾里得等人發展了更為抽象的數學理論，這些理論成為現代數學的基礎。數學經過幾千年的發展，形成了今天我們所知的各種分支，如算術、幾何學、代數、微積分等。 總的來說，數學的歷史非常悠久，跨越了多個世紀，並且隨著	數學的歷史可以追溯到公元前約 3000 年左右。早期的數學起源於古代文明，如古埃及、古巴比倫和古印度。這些文明使用數學來進行天文觀測、建築、貿易等活動。 最早的數學概念包括計數、測量、簡單的算術（加、減、乘、除）以及幾何。隨著時間的推移，數學逐漸發展出更複雜的理論和方法，尤其是在古希臘時期（大約公元前 500 年到公元前 300 年），當時數學家如歐幾里得、皮塔哥拉斯和阿基米德等人奠定了許多現代數學的基礎。	數學的歷史可以追溯到幾千年前。最早的數學證據出現在古埃及和古巴比倫，大約是公元前 3000 年到公元前 2000 年左右。當時的數學主要是用來進行測量、計算和天文觀察等基本應用。 數學的發展進入了新的階段，特別是古希臘時期（大約公元前 600 年到公元前 300 年），在這時，數學不僅是應用工具，也開始發展成為一門抽象學科。像是歐幾里得的《幾何原本》和畢達哥拉斯學派的數學思想，對後來的數學發展影響深遠。

	人類文明的進步不斷演變和發展。	因此，數學有大約五千多年的歷史。	因此，數學的歷史已有超過四千年的時間。
相關聯性	5/5	5/5	5/5
段落鮮明度	5/5	5/5	5/5
用字遣詞	4/5	4/5	4/5
內容豐富度	3/5	4/5	5/5
額外加分	+1	—	—
總分	18/20	18/20	19/20

註：研究者自行整理

雖然「數學有幾年的歷史？」ChatGPT 給這個問題的回覆都不太一樣，但因為確切的年份眾說紛紜，且 ChatGPT 給的數字大約都是在 4000~5000 多年之間，並無太大差別，所以年份不予以計算至分數內。

在內容豐富度中主帳號的分數較低，因為是我的個人化灌輸的資訊，使他這麼做的，所以給予額外的加分。

整體來說總分差距並不大，但無資料帳號給的回覆較為全面，主帳號和造假資料較偏向個人化設定裡的要求。

表 4-2-3 帳號組 A3

	A3：水的化學式是什麼？		
實驗帳號	主帳號	無資料	造假資料
回覆	水的化學式是 H ₂ O。這表示每個水分子由兩個氫原子 (H) 和一個氧原子 (O) 組成。	水的化學式是 H ₂ O。這表示每個水分子由兩個氫原子 (H) 和一個氧原子 (O) 組成。	水的化學式是 H ₂ O。這表示水分子由兩個氫原子 (H) 和一個氧原子 (O) 組成。
相關聯性	5/5	5/5	5/5
段落鮮明度	1/5	1/5	1/5
用字遣詞	5/5	5/5	5/5
內容豐富度	5/5	5/5	5/5
總分	16/20	16/20	16/20

註：研究者自行整理

水的化學式這個問題對三個不同帳號提問，最後結果並無太大差別。

表 4-2-4 帳號組 B1

	B1：如何教一個完全沒有程式基礎的人寫 python 程式？		
實驗帳號	主帳號	無資料	造假資料
回覆	教一個完全沒有程式基礎的人寫 Python 程式可以分成幾個階段，每個階段注重基礎概念、實踐練習，逐步建立理解。以下是一個簡單的教學架構： 1. 介紹程式和 Python 什麼是程式？：解釋程式是用來讓電腦執行指令的語言，程式可以解決問題、做計算，甚至建立應用程式。	教一個完全沒有程式基礎的人寫 Python 程式，建議循序漸進，從基本概念開始，逐步提升。以下是一個教學的步驟： 1. 介紹程式語言和 Python - 程式語言的概念：簡單解釋什麼是程式語言，為何它是計算機與人類溝通的工具。 - Python 的特點：強調 Python 是一個簡單、	要教一個完全沒有程式基礎的人寫 Python 程式，可以從以下幾個步驟開始： 1. 介紹程式是什麼 先解釋程式語言的概念。程式是我們與電腦溝通的方式，告訴電腦該做什麼。可以舉一些生活中的例子，比如像是寫菜單給餐廳廚師。 2. 安裝 Python 和設定開發環境 介紹 Python 的安裝：可以下載並安裝

	- 為什麼選 Python？：Python 是簡單且易讀的語言，適合初學者，且有大量的資源和社群支持。 2. 基礎語法介紹 - 變數與資料類型： - 變數：如何儲存數字、文字等資料。 - 常見資料類型：整數 (int)、浮點數 (float)、字串 (str)、布林值 (bool)。 - 範例：age = 25, name = "Alice", is_active = True - 簡單運算： 加、減、乘、除、餘數等。 - 範例：x = 10 + 5, y = 7 * 2 - 輸入與輸出： - 使用 input() 來接受使用者輸入，使用 print() 來顯示輸出。 - 範例： <pre>name = input("What's your name") print("Hello, " + name)</pre> 3. 流程控制 - 條件語句 (if/else)： - 判斷某些條件是否為真，來決定要執行的操作。 - 範例： <pre>age = 18 if age >= 18: print("You are an adult.") else: print("You are a minor.")</pre> - 迴圈 (for 和 while)： - 用於重複執行某些操作。 - 範例： <pre>for i in range(5): print(i)</pre> 4. 簡單的實踐項目 - 計算器：讓學生寫一個簡單的加法、減法、乘法、除法計算器。	易讀的語言，適合初學者。 2. 安裝 Python - 引導他們如何安裝 Python 以及常用的編輯器（如 VS Code 或 PyCharm），或使用網頁平台（如 Google Colab 或 Jupyter Notebook）來寫程式。 3. 基本語法 - 印出文字：介紹 print() 函數來顯示文字。 <pre>print("Hello, World!")</pre> - 變數：解釋變數是用來儲存資料的容器，可以賦予不同的值。 <pre>greeting = "Hello" print(greeting + " " + "World")</pre> 4. 基本資料型態 - 數字 (int, float)：介紹整數和浮點數。 <pre>x = 5 # 整數 y = 3.14 # 浮點數</pre> - 字串 (str)：介紹字串的基本操作。 <pre>name = "Alice" age = 25 print(name, age)</pre> 5. 控制結構 - 條件語句 (if-else)：介紹如何根據條件執行不同的程式碼。 <pre>age = 18 if age >= 18: print("Adult") for i in range(5): print(i)</pre> - 迴圈 (for, while)：介紹如何重複執行一段程式碼。 6. 函數 - 介紹如何定義和使用函數來組織程式碼，提高可讀性和可重用性。 <pre>def greet(name): print("Hello, " + name)</pre> <pre>greet("Alice")</pre> 7. 練習題 - 提供簡單的練習題，幫助學生鞏固所學的基礎：	Python (https://www.python.org/downloads/)。 - 安裝一個簡單的編輯器，像是 VS Code 或者 Jupyter Notebook，這樣可以方便寫程式和測試。 3. 基本語法和概念 - 輸入與輸出：介紹 print() 函數，讓學習者能夠輸出文字。 <pre>print("Hello, World!")</pre> - 變數：讓學習者理解如何用變數存儲資料。比如： <pre>x = 5 y = 10 print(x + y)</pre> - 數字和字串：解釋數字和文字（字串）的不同。可以用簡單的例子來展示。 <pre>age = 24 name = "Alice" print(name + " is " + str(age) + " years old")</pre> 4. 條件判斷和邏輯 - 介紹 if 和 else 判斷式。 <pre>age = 20 if age >= 18: print("You are an adult.") else: print("You are a minor.")</pre> 5. 循環 - 介紹 for 和 while 循環，讓學習者能夠做重複的工作。 <pre>for i in range(5): print(i)</pre> 6. 函數 - 教學如何定義簡單的函數。 <pre>def greet(name): print("Hello, " + name)</pre> <pre>greet("Alice")</pre> 7. 實踐項目 - 提供一些簡單的實踐項目，讓學習者能夠實際動手操作。例如，寫一個簡單的計算器，或寫一個猜數字遊戲。 8. 循序漸進 - 確保每個概念都理解清楚再進一步學習。可以隨時用簡單的例
--	--	--	---

	○ 例子： <pre>num1 = float(input("Enter first number: ")) num2 = float(input("Enter second number: ")) operation = input("Choose operation (+, -, *, /): ") if operation == "+": print(num1 + num2) elif operation == "-": print(num1 - num2) elif operation == "*": print(num1 * num2) elif operation == "/": print(num1 / num2) else: print("Invalid operation")</pre> 5. 進一步的學習 ● 函數：教學生如何將程式碼分成功能模塊，並使用函數來組織程式。 ● 資料結構：介紹列表、字典等資料結構，讓學生能夠處理複雜的資料。 6. 持續練習與挑戰 鼓勵學生多寫程式，從簡單的問題開始，如計算某個數字的階乘、列出所有質數等。隨著技能進步，可以挑戰更複雜的項目。 7. 資源與學習工具 ● 學習網站：如 Codecademy、LeetCode 等。 ● 影片教學：YouTube 上有很多適合初學者的 Python 教學。 ● 書籍推薦：《Python Crash Course》、《Automate the Boring Stuff with Python》。 學習程式是一個循序漸進的過程，鼓勵學生多動手寫程式，從錯誤中學習，並持續探索。	1. 打印出從 1 到 10 的數字。 2. 寫一個函數，根據用戶的年齡判斷是否可以投票。 3. 計算 1 到 100 的總和。 8. 進階概念（可選） ● 資料結構：介紹列表 (list)、字典 (dict) 等資料結構。 ● 例外處理：介紹 try-except 來處理錯誤。 ● 模組與庫：介紹如何使用 Python 的標準庫，像是 math 模組來處理數學運算。 9. 反覆練習和改錯 ● 錯誤調試：幫助學生學習如何理解錯誤訊息並修正程式中的錯誤。 ● 練習：鼓勵學生自己寫小程序，逐步增加難度。 10. 資源推薦 ● 建議使用 Python 的線上教程和書籍，如《Python Crash Course》或 Codecademy 上的 Python 課程，來進一步學習。 逐步構建基礎，讓學生在理解每個概念後繼續進行下一步。最重要的是給他們足夠的練習機會，並讓他們在過程中解決問題，這樣能夠加深理解。	子來加深印象，並鼓勵多做練習。 通過這些步驟，會幫助學習者理解 Python 的基本概念，並且能夠開始寫一些簡單的程式。
相關聯性	5/5	5/5	5/5
段落鮮明度	5/5	5/5	5/5
用字遣詞	5/5	5/5	5/5
內容豐富度	4/5	4/5	3/5
總分	19/20	19/20	18/20

註：研究者自行整理

這三個問題裡，因為內容多寡的關係，所以分數有些微的差距，但結果與之前的實驗相同，無資料帳號的答案相對比較全面，故主帳號的答案會更符合我個人的要求。

表 4-2-5 帳號組 B2

B2：如何將數學原理應用於生活之中？			
實驗帳號	主帳號	無資料	造假資料
回覆	將數學原理應用於日常生活中，其實是非常有用的。以下是一些例子，說明如何將數學概念與生活相結合： 1. 時間管理與計劃：利用數學中的比例、分數和比率來規劃時間。例如，假設你有三個任務，並且知道它們的完成時間，則可以計算每個任務的優先順序，並確保時間分配最有效。 2. 預算與財務規劃：在生活中，每月的收入和支出都需要計算。利用加法、減法、乘法和除法來管理你的預算，確保收支平衡。此外，了解利率和貸款計算也可以幫助你做出更聰明的財務決策。 3. 購物折扣：當你購物時，可以利用百分比來計算折扣。例如，如果一件衣服打七折，原價 1000 元，那麼你可以計算折後價格是 1000 元乘以 0.7，即 700 元。 4. 烹飪與食譜：做菜時，你常常需要調整食譜中的比例。這是應用比率和比例的方法。如果食譜是為 4 人設計的，而你是為 2 人做飯，你可以將所有材料的量減半。 5. 運動與健康：在運動時，數學可幫助你計算心率、卡路里消耗等。對於體重訓練，計算每組的重量和次數可以幫助你確保訓練達到目標。 6. 空間與物體擺放：當你整理房間或搬家時，數學中的幾何學能幫助你計算空間大小，並計算出最佳的擺放方式。 7. 交通與路程規劃：如果你想要從 A 地點到 B 地點，計算路程所需時間的速度和距離公式可以幫助你找到最佳路徑或了解交通狀況。這些數學概念不僅能提高你的生活效率，還能讓你在解決日常問題時更加精確和有條理。	數學原理在日常生活中有廣泛的應用，幫助我們解決各種問題、做出更有效的決策。以下是一些具體的應用例子： 1. 財務規劃與預算管理： ○ 使用基本的數學運算來跟踪收入和支出，幫助管理個人或家庭的預算。 ○ 比如，使用加法、減法來計算日常支出，並利用百分比來計算儲蓄或投資的回報率。 2. 購物時的折扣計算： ○ 折扣、促銷和價格比較中，百分比（如“打七折”）是經常使用的數學概念。你可以快速計算出折後價格，選擇最佳優惠。 3. 時間管理與計劃： ○ 在時間規劃中，數學幫助我們分配時間和安排任務。例如，運用比例來確定每項工作所需的時間，或者將一天的時間劃分為不同的區塊，實現更高效的工作。 4. 料理中的測量： ○ 在烹飪中，我們經常使用比例、比例換算和測量。例如，當食譜需要修改份量時，你可以根據原始配方的比例調整每一項材料的數量。 5. 道路旅行的時間預估： ○ 在規劃旅行時，運用速率、時間和距離的公式來估算旅行所需的時間。這可以幫助你更好地安排休息和交通工具。 6. 統計與數據分析： ○ 在許多情境下，我們都在處理數據，比如分析工作表中的數字、消費記錄或測量結果。簡單的統計學原理，如平均數、標準差等，幫助我們理解數據的分布和趨勢。 7. 建設與裝修中的測量： ○ 在家居裝修或建設工程中，測量和計算所	數學原理可以在日常生活中應用在很多方面，這裡有一些例子： 1. 財務管理：你可以利用數學中的加法、減法、乘法和除法來計算日常開支、制定預算或管理儲蓄。例如，了解利息的計算方式幫助你選擇最合適的銀行帳戶或貸款。 2. 購物和折扣計算：當你購物時，數學可以幫助你計算折扣後的價格。假設一個商品原價 100 元，打 8 折，那麼折扣後的價格就是 $100 \times 0.8 = 80$ 元。 3. 時間管理：時間的分配和計算也可以用到數學。例如，你可以利用百分比來分配一天的時間，確保自己有足夠的時間完成工作、學習和休息。 4. 運動計算：在像籃球這樣的運動中，數學也可以幫助你分析表現。例如，計算射門命中率、比賽時間、得分差等，這些數字可以幫助你制定策略。 5. 健康與營養：數學幫助你計算卡路里攝入量、飲食比例等。了解如何計算 BMI（身體質量指數）來判斷自己的健康狀況也是一個實際應用。 6. 建設和裝修：在建造房屋或裝修時，數學中的幾何學幫助計算面積、體積、角度和比例，以確保材料的使用和設計的精確。 這些都是數學在現實生活中的應用，學會這些可以讓你更加有效地管理生活中的各種事物。

		需材料是必不可少的。數學幫助確保精確的計算，避免浪費。 這些例子只是冰山一角，數學的應用遍及生活的方方面面，能夠幫助我們提高效率、做出更好的選擇。	
相關聯性	5/5	5/5	5/5
段落鮮明度	5/5	5/5	5/5
用字遣詞	5/5	5/5	5/5
內容豐富度	4/5	5/5	4/5
總分	19/20	20/20	19/20

註：研究者自行整理

在無資料組裡面有一些錯字，但因為 ChatGPT 常有打錯字的問題，且錯字不會影響 ChatGPT 的回覆，所以不予以扣分。而在實驗結果一樣是無資料帳號的回覆內容較多、較全面，主帳號較符合我的要求教的是較為困難的東西。

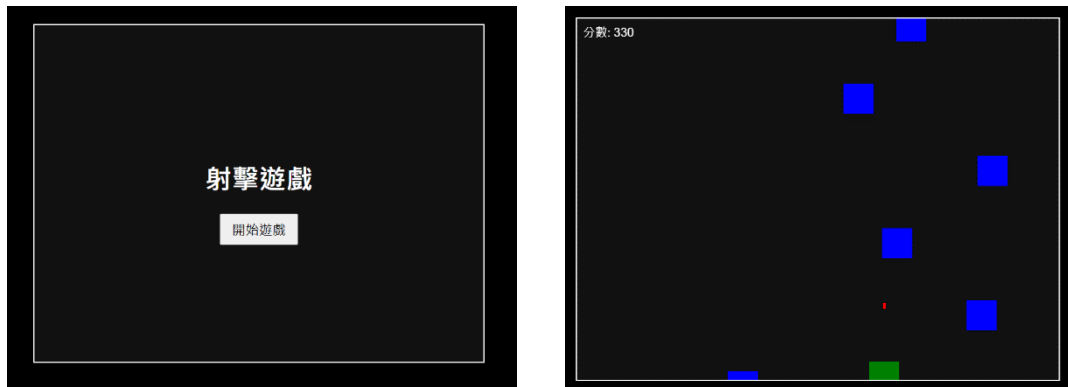
表 4-2-6 帳號組 B3

B3：如何將科學原理應用於生活之中？			
實驗帳號	主帳號	無資料	造假資料
回覆	將科學原理應用於日常生活中，可以幫助我們更有效地解決問題，提升生活品質。以下是一些實際的例子： 熱力學原理： - 冰箱和空調：這些家用電器利用熱力學中的熱交換原理來保持食物冷卻或創造舒適的室內環境。冰箱將熱量從內部轉移到外部，這是基於熱流動的基本原理。 摩擦力： - 穿鞋：選擇適合的鞋子可以減少摩擦，讓走路更加輕鬆。運動鞋的設計通常會利用摩擦原理來增加穩定性，防止滑倒。 浮力原理： - 游泳：當你進入水中時，你會感受到一股向上的力量，這是由浮力原理引起的。根據阿基米德原理，物體在液體中所受的浮力等於它排開的液體的重力。 光學原理： - 眼鏡和鏡子：眼鏡根據光的折射原理幫助我們改善視力，鏡子利用光的反射原理來形成影像。 生物學原理：	科學原理在日常生活中有許多應用，以下是一些常見的例子： 熱力學原理： 當你用鍋子加熱水時，熱力學原理發揮作用。熱能從爐火傳遞到鍋子，再由鍋子傳遞到水，水的溫度因此上升。這是熱的傳遞過程，並且這一原理也解釋了為何冷熱飲料會達到環境的溫度。 摩擦力： 摩擦力是你走路、開車、甚至使用滑板的基礎原理。摩擦力使你不會滑倒，或者使車輪能夠更好地與路面接觸。了解摩擦力可以幫助我們選擇適合的鞋子或輪胎，增強行走或駕駛的穩定性。 浮力原理： 當你進入游泳池時，你會發現身體部分浮起來。這是阿基米德原理（浮力原理）的應用，說明當物體被液體排開時，物體會受到向上的浮力。這解釋了為什麼船能浮在水面上。 光學： 當你使用眼鏡或相機時，光學原理發揮作用。鏡片會改變光線的路徑，幫助你看得更清楚，或者讓相機拍攝清晰的照片。了解光的折射和反射可以幫助我們設計更好的視覺設備。	將科學原理應用於生活中可以幫助我們更好地理解世界，並提高生活質量。以下是一些具體例子，展示了科學原理在日常生活中的應用： 物理學：物理學原理，如杠杆原理和重力，可以應用於日常活動。例如，使用撬棍搬運重物時，利用杠杆原理可以輕鬆撬起重物。重力也解釋了物體為什麼會向地面掉落，這對我們理解運動和安全有很大幫助。 化學：化學反應原理可以應用於烹飪。當食物加熱時，化學反應會改變食物的味道、顏色和結構。了解這些反應有助於我們更好地控制烹飪過程，讓食物更加美味。 生物學：生物學原理可以幫助我們保持健康。例如，了解人體如何消化食物，如何抵抗疾病，以及如何通過運動增強免疫系統，這些知識可以幫助我們做出更好的生活選擇。 數學：數學原理在日常生活中也有廣泛應用，像是計算預算、計劃旅行時間、或者計算購物折扣等，這些都依賴數學原理來幫助我們做出合理的決策。

	我會提供一個簡單的 JavaScript 程式碼範例，這個範例包含了射擊遊戲的基本結構，包括起始畫面、遊戲邏輯，以及重複操作的機制。您可以將其放入 CodePen 的 JS 部分來測試。  這段代碼的功能： 起始畫面：當頁面加載完成後，顯示一個歡迎界面，包含一個按鈕可以開始遊戲。 遊戲邏輯： - 玩家可以利用左/右箭頭鍵控制移動，空格鍵發射子彈。 - 敵人會從畫面上方隨機生成並向下移動。 - 當子彈擊中敵人時，得分增加，敵人被摧毀。 - 如果敵人觸及畫面底部，遊戲結束，顯示 "Game Over" 和重啟提示。 重複操作：遊戲結束後，玩家可以點擊畫面來重新開始遊戲。 使用說明： - 將這段代碼粘貼到 CodePen 的 HTML 和 JS 部分即可運行。 - 點擊 "Start Game" 開始遊戲，使用箭頭鍵移動，按空格鍵發射子彈。 - 當遊戲結束後，點擊畫面可以重新開始。 這個遊戲是基本版本，您可以根據需要進行擴展，例如添加音效、更複雜的敵人行為和圖像等。	JavaScript (game.js) <pre>// Canvas Setup const canvas = document.getElementById('gameCanvas'); const ctx = canvas.getContext('2d'); canvas.width = 800; canvas.height = 600; let isGameStarted = false;</pre> 說明： HTML：這部分包含了 canvas 元素，並引用了 game.js 檔案來控制遊戲邏輯。 CSS：使畫布居中並設置背景顏色。 JavaScript： - player 對象：定義了玩家的起始位置和移動速度。 - bullets：存儲所有子彈的位置。 - 按鍵事件處理器：用來檢測玩家是否按下左右方向鍵或空格鍵來進行移動和射擊。 - gameLoop：這是遊戲的主循環，負責更新畫面並持續進行遊戲。 功能： 開始畫面：顯示“按 Enter 鍵開始”字樣。 遊戲控制：玩家可以利用左右鍵來控制角色的移動，空格鍵來射擊子彈。 遊戲結束：當您想添加更多功能時，可以在適當的位置插入遊戲結束畫面和重啟邏輯。 這個範例實現了基本的射擊和移動功能。您可以根據需求擴展它，如加入敵人、得分系統或更多動畫效果。 將以上代碼放入 CodePen 中的 HTML、CSS 和 JavaScript 欄位，並進行測試。	HTML (index.html) <pre><!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1"> <title>射擊遊戲</title> <link rel="stylesheet" href="style.css"> </head></pre> CSS (style.css) <pre>body { margin: 0; padding: 0; background-color: black; display: flex;</pre> JavaScript (game.js) <pre>const canvas = document.getElementById('gameCanvas'); const ctx = canvas.getContext('2d'); const startScreen = document.getElementById('startScreen'); const startButton = document.getElementById('startButton'); let gameRunning = false; let player;</pre> 說明： HTML：這個文件包含了一個 <canvas> 元素來顯示遊戲畫面，以及一個起始畫面的 <div>，當用戶點擊開始按鈕時，遊戲將開始。 CSS：設定了畫布和起始畫面的樣式。 JavaScript： - 定義了三個主要的物件類別：Player（玩家）、Bullet（子彈）和 Enemy（敵人）。 - 遊戲邏輯包括移動玩家、發射子彈、生成敵人、碰撞檢測和顯示分數。 - 當按下空格鍵時，玩家會發射子彈。 - 點擊開始按鈕後，遊戲開始，並且可以在遊戲結束後重啟。 將這些代碼分別放進你的 CodePen HTML、CSS 和 JS 部分中，就可以創建一個簡單的射擊遊戲！
相關聯性	5/5	5/5	5/5
段落鮮明度	5/5	5/5	5/5
用字遣詞	5/5	5/5	5/5
執行結果	5/5	4/5	3/5
總分	20/20	19/20	18/20

註：研究者自行整理

造假資料(右)可以做到有起始畫面，射擊、得分，但可惜沒有死亡、遊戲結束的機制。



無資料(中)可以做到射擊且有起始畫面，但沒有敵人和分數，雖然我沒要求分數，但沒有敵人不就射擊遊戲了，所以在執行結果的地方扣分。



所以對照組(主帳號)可以做到我所要求的所有事。造假資料不會遊戲結束，我把它解釋為沒有辦法重複操作遊戲，所以也給予扣分。無資料完全沒有敵人和分數，所以扣更多的分。從這樣看下來，你灌輸越多資料給 ChatGPT 它給你的程式碼越完整，但目前還不知道原因。

表 4-2-8 帳號組 C2

C2：生成一個可以透過使用者的輸入來解讀一元一次方程式的 python 一元一次方程式計算機程式並可應用在 CodePen 網頁中。			
實驗帳號	主帳號	無資料	造假資料
回覆	要在 CodePen 中使用 Python 寫一個簡單的計算機來解一元一次方程式，通常會需要將 Python 程式轉換為 JavaScript，因為 CodePen 是基於 Web 的前端平台，它只支持 HTML、CSS、JavaScript，而不能直接運行 Python 代碼。 下面是一個用 JavaScript 寫的解一元一次方程式的計算機程式，可以應用於 CodePen 中。這個程式	這是你所要求的 Python 一元一次方程式計算機，並且能夠處理用戶的輸入。你可以將這段代碼應用到 CodePen 中，使用 HTML、CSS 和 JavaScript 結合 Python 來進行計算。雖然 CodePen 不能直接執行 Python，但你可以使用 Python 程式來處理方程式的運算，並用 JavaScript 作為前端與後端的橋樑。	這是一個簡單的一元一次方程式計算機程式，可以通過使用者輸入解讀並計算結果。這段程式碼是用 HTML 和 JavaScript 實現的，可以直接應用於 CodePen 網頁中。

表 4-2-9 帳號組 C3

	C3：生成一個元素週期表有元素的名字、代號與分類的 python 元素週期表程式，且點擊後會出現該元素的解釋、英文、分類、代號、名稱，同時此程式可應用在 CodePen 網頁中。		
實驗帳號	主帳號	無資料	造假資料
回覆	要在 CodePen 中使用 Python 並生成一個元素週期表，通常 Python 本身無法直接在瀏覽器中運行。你可以使用 HTML、CSS 和 JavaScript 來實現這個需求。這裡是如何用這三種語言創建一個動態元素週期表的範例： 1. HTML（結構） <pre></head> <body> <div>Interactive Periodic Table</div> <div class="table"> <!-- Elements will be injected dynamically by JavaScript --> </div> </body></pre> 2. CSS（樣式） <pre>@keyframes color { 0% {background-color: #f0f0f0;} 50% {background-color: #e0e0e0;} 100% {background-color: #f0f0f0;} }</pre> 3. JavaScript（功能） <pre>const periodicTable = { // Define the data for the periodic table elements: [{name: "Hydrogen", "symbol": "H", "atomic_number": 1, "category": "Gases"}, {name: "Helium", "symbol": "He", "atomic_number": 2, "category": "Gases"}, // Add more elements as needed] }; // Function to generate the periodic table structure function generateTable() { // Create the table structure const table = document.querySelector(".table"); // Loop through the elements and create the table rows for (let i = 0; i < periodicTable.elements.length; i++) { const element = periodicTable.elements[i]; // Create the element details const details = `       <div>         <div>\${element.name}</div>         <div>\${element.symbol}</div>         <div>\${element.atomic_number}</div>         <div>\${element.category}</div>       </div>     `; // Append the details to the table table.innerHTML += details; } }</pre> 說明： ● HTML： 設置了頁面結構，並創建了一個元素容器 (.table) 和一個顯示元素詳情的區域 (.element-details)，後者將顯示點擊的元素的詳細信息。 ● CSS： 提供了基本的樣式，使元素以表格形式顯示並進行了一些動效設置（如：元素縮放的懸停效果）。 ● JavaScript： 創建了每個元素的按鈕，並將其顯示在表格中。當用戶點擊某個元素時，該元素的詳細信息會顯示在下方。 說明： ● 在 elements 陣列中，您可以根據需要添加更多的元素，每個元素包含名稱、符號、類別、原子數和描述。 ● getCategoryColor 函數根據元素的類別返回不同的顏色（可以擴展更多的類別和顏色）。 ● 當您點擊某個元素時，JavaScript 會動態顯示該元素的詳細信息。 如何在 CodePen 中使用： 1. 創建一個新的 Pen。		
	以下是一個使用 Python、HTML 和 JavaScript 來實現的元素週期表範例程式。它會顯示元素名稱、代號、顏色和分類，並且點擊元素時會顯示該元素的更多資訊。這個程式可以嵌入在 CodePen 網頁中。 HTML + CSS + JavaScript 結合 Python 的元素週期表範例： 解釋： 1. HTML 部分： ○ table-container 是顯示元素的區域，每個元素都是一個 <div> 元素，點擊後會顯示該元素的詳細資訊。 ○ element-info 用來顯示元素的詳細信息，包括名稱、符號、分類和描述。 2. CSS 部分： ○ element 是每個元素的樣式，顯示為彩色的方塊，並且當用戶點擊時會顯示詳細信息。 ○ element-info 用來顯示元素的詳細資訊。 3. JavaScript 部分： ○ elements 數組儲存了每個元素的基本資料，包括名稱、符號、分類、描述和顏色。 ○ createElementButton 函數用來創建每個元素的顯示按鈕，並且為每個按鈕加上點擊事件處理器，當按鈕被點擊時顯示相應元素的資訊。 ○ window.onclick 監聽點擊事件，當用戶點擊頁面上的其他區域時，隱藏元素的詳細資訊。		
	以下是一个完整的 Python 代码示例，使用 plotly 库生成一个带有元素名称、代号、颜色和分类的交互式元素周期表，点击元素时显示其详细信息。该代码还可以适用于 Web 端的代码，特别是在 CodePen 上。 为了在网页上显示这个周期表，我们可以利用 HTML 和 JavaScript 的 plotly.js 库生成可交互的图表。在 CodePen 中，你可以使用以下 HTML 和 JavaScript 代码嵌入周期表。 Python 程式 首先，我们需要使用 plotly 库生成一个交互式周期表。以下是 Python 代码： <pre>import plotly.graph_objects as go # Define the data for the periodic table elements = [{"name": "Hydrogen", "symbol": "H", "atomic_number": 1, "category": "Gases"}, {"name": "Helium", "symbol": "He", "atomic_number": 2, "category": "Gases"}, # Add more elements as needed]</pre> 在 CodePen 中的 HTML 和 JavaScript 部分： 你可以将以下 HTML 和 JavaScript 代码嵌入到 CodePen 中，以生成周期表并处理点击事件。 说明： <pre><!-- DOCTYPE HTML --> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1"> </head> <body></pre> 1. Python 部分： 使用 plotly 生成一个交互式周期表图，并可以通过点击元素查看详细信息。 2. HTML & JavaScript 部分： 用于在 CodePen 中显示周期表。点击元素会弹出该元素的详细信息。 你可以复制并粘贴这些代码到你自己的开发环境或 CodePen 页面中，然后调整元素的颜色、布局和详细信息，适应你的需求。		

	2. 將 HTML 放入 HTML 編輯器中。 3. 將 CSS 放入 CSS 編輯器中。 4. 將 JavaScript 放入 JavaScript 編輯器中。 這樣，您就能在 CodePen 中看到互動式元素週期表，並能查看每個元素的詳細信息。	1. 打開 CodePen 網站 (https://codepen.io/pen/)。 2. 在 HTML、CSS 和 JavaScript 欄位中分別粘貼上述代碼。 3. 點擊“Run”按鈕來查看元素週期表。 這個簡單的元素週期表可以被擴展，添加更多的元素、顏色或樣式來使其更加詳細和吸引人。	
相關聯性	4/5	5/5	4/5
段落鮮明度	5/5	5/5	5/5
用字遣詞	5/5	5/5	5/5
執行結果	2/5	1/5	1/5
總分	16/20	16/20	15/20

註：研究者自行整理

在這個問題裡，無資料缺少了代號和顏色，且顯示有點奇怪，造假資料缺少代號和解釋。但三個組別都沒有列出所有元素，所以主帳號的功能最完善，且不會有 bug，所以分數會有些微的差距。

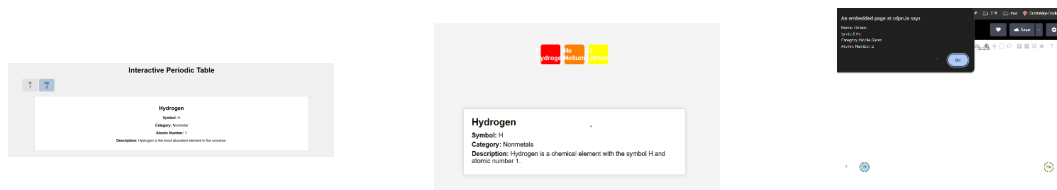


表 4-2-10 實驗組(帳號)評分統整表

問題	總分		
	主帳號	無資料	造假資料
A1	20/20	18/20	17/20
A2	18/20	18/20	19/20
A3	16/20	16/20	16/20
B1	19/20	19/20	18/20
B2	19/20	20/20	19/20
B3	18/20	19/20	18/20
C1	20/20	19/20	18/20
C2	19/20	20/20	13/20
C3	16/20	16/20	15/20
總共	165/180	165/180	153/180

註：研究者自行整理

根據結果，主帳號、無資料的總分最高，因為主帳號有我灌輸的個人化資料，所以給的回覆都是我比較喜歡的答案，所以分數較高。無資料因為沒有給他任何資料，所以回覆較為全面，內容較多較全面。

三、實驗三：實驗組(國家 IP)

本實驗以國家 IP 作為我的操縱變因，有台灣、美國和日本三個國家的 IP，控制變因會以主帳號、實驗時間為每星期二下午 1~2 點以及我說話的方式以中文去敘述向 ChatGPT 提問，以下是我的提問問題以及 ChatGPT 的回覆：

(一)操縱變因：台灣

(二)操縱變因：美國

(三)操縱變因：日本

表 4-3-1 國家組 A1

	A1：C#是一種程式語言嗎？		
實驗 IP	台灣	美國	日本
回覆	是的，C#（發音為 "C-sharp"）是一種程式語言。它由微軟開發，主要用於構建各種應用程序，特別是 Windows 平台上的應用程序。C# 是基於 .NET 框架的一種現代、物件導向的語言，適用於桌面應用程序、網站、移動應用程序等開發。 它的語法類似於 C 和 C++，但在設計上更加簡單和安全。C# 支援許多現代程式語言的特性，例如垃圾回收、異常處理和多線程處理等。	是的，C#（發音為 "C-sharp"）是一種程式語言。它由微軟（Microsoft）開發，主要用於開發 Windows 應用程式、Web 應用程式、以及遊戲開發（特別是在 Unity 引擎中）。C# 是一種面向對象的語言，擁有簡單的語法和強大的功能，適合各種開發需求。	是的，C#（發音為 "C-sharp"）是一種程式語言。它由微軟（Microsoft）開發，主要用於開發桌面應用程式、網頁應用程式以及遊戲等。C# 是一種面向對象的程式語言，語法簡單易學，並且與 .NET 平台密切集成，使開發者能夠在不同的環境中創建高效的應用程式。
相關聯性	5/5	5/5	5/5
段落鮮明度	5/5	3/5	3/5
用字遣詞	5/5	5/5	5/5
內容豐富度	5/5	4/5	4/5
總分	20/20	17/20	17/20

註：研究者自行整理

本實驗差距的三分在於內容豐富度和段落鮮明度，但整體回覆內容並無太大差距。

表 4-3-2 國家組 A2

	A2：數學有幾年的歷史？		
實驗 IP	台灣	美國	日本
回覆	數學的歷史可以追溯到古代文明，約有 4000 多年的歷史。最早的數學記錄來自古埃及和古巴比倫，這些文化在公元前 2000 年左右已經開始使用數字系統來解決日常問題，如測量土地和建造建築物。 隨著時間的推移，古希臘數學家如畢達哥拉斯和歐幾里得等人發展了更為抽象的數學理論，這些理論成為現代數學的基礎。數學經過幾千年的發展，形成了今天	數學的歷史可以追溯到大約公元前 3000 年左右。最早的數學形式出現在古代的美索不達米亞和古埃及，當時的數學主要用於計算土地面積、記錄貿易和建立日曆等。古埃及的數學家使用基本的算術和幾何方法來解決實際問題，而古代的巴比倫人和埃及人也發展了初步的數字系統。 隨著時間的推移，數學逐漸演變成一門更抽象的學科，尤其在古希臘時期，數學開始有了更為系	數學的歷史可以追溯到大約公元前 3000 年左右。最早的數學記錄來自古埃及和美索不達米亞（今天的伊拉克地區），那時的數學主要用於天文學、建築和貿易等實際應用。 例如，古埃及人發展了幾何學，幫助他們進行金字塔的建設；而古巴比倫人則使用了一些基本的數學概念，如加法、減法、乘法、除法以及某些數學方程的求解方法。

	我們所知的各種分支，如算術、幾何學、代數、微積分等。 總的來說，數學的歷史非常悠久，跨越了多個世紀，並且隨著人類文明的進步不斷演變和發展。	統化的發展。例如，畢達哥拉斯和歐幾里得等人對數學理論進行了深入研究，並建立了數學證明的基礎。 因此，數學大約有超過 5000 年的歷史。	古希臘時期的數學家，如畢達哥拉斯、歐幾里得和阿基米德，將數學理論進一步發展，並創造了數學的嚴謹證明方法。這些基礎的發展對後來數學的發展影響深遠。 因此，數學歷史可以說有超過 5000 年的發展歷程。
相關聯性	5/5	5/5	5/5
段落鮮明度	4/5	4/5	5/5
用字遣詞	4/5	4/5	4/5
內容豐富度	4/5	4/5	5/5
總分	17/20	17/20	19/20

註：研究者自行整理

這三個回覆其實並無太大差別，差距在於內容量，但內容量也沒差到很多，推測不同國家 IP 可能會影響找到的資料，進而影響內容與資料量。

表 4-3-3 國家組 A3

	A3：水的化學式是什麼？		
實驗 IP	台灣	美國	日本
回覆	水的化學式是 H ₂ O。這表示每個水分子由兩個氫原子（H）和一個氧原子（O）組成。	水的化學式是 H ₂ O。這表示水由兩個氫原子（H）和一個氧原子（O）組成。	水的化學式是 H ₂ O。這表示每個水分子由兩個氫原子（H）和一個氧原子（O）組成。
相關聯性	5/5	5/5	5/5
段落鮮明度	5/5	5/5	5/5
用字遣詞	5/5	5/5	5/5
內容豐富度	1/5	1/5	1/5
總分	16/20	16/20	16/20

註：研究者自行整理

回覆的內容有些許差別但在語句的意思上無差別。

表 4-3-4 國家組 B1

	B1：如何教一個完全沒有程式基礎的人寫 python 程式？		
實驗 IP	台灣	美國	日本
回覆	教一個完全沒有程式基礎的人寫 Python 程式可以分成幾個階段，每個階段注重基礎概念、實踐練習，逐步建立理解。以下是一個簡單的教學架構： 1. 介紹程式和 Python - 什麼是程式？：解釋程式是用來讓電腦執行指令的語言，程式可以解決問題、做計算，甚至建立應用程式。 - 為什麼選 Python？：Python 是簡單且易讀的語言，適合初學者，且	教一個完全沒有程式基礎的人寫 Python 程式可以從以下步驟開始： 1. 介紹程式設計的基本概念 - 什麼是程式語言？讓他們理解程式語言是與電腦溝通的工具，透過它來告訴電腦該做什麼。 - 什麼是 Python？說明 Python 是一種簡單且強大的程式語言，適合初學者。 - 基本概念：變數、運算符、資料型態（整	教導完全沒有程式基礎的人寫 Python 程式時，可以從以下幾個步驟開始： 1. 介紹程式設計的基本概念 - 什麼是程式設計：解釋程式設計是如何讓電腦按照我們的指示執行工作的，類似給生活中的指示，例如如何教朋友做某件事情。 - 語言的作用：說明 Python 就是與電腦溝通的一種語言，類似於我們使用的語言，

	有大量的資源和社群支持。 2. 基礎語法介紹 - 變數與資料類型： - 變數：如何儲存數字、文字等資料。 - 常見資料類型：整數 (int)、浮點數 (float)、字串 (str)、布林值 (bool)。 - 範例：age = 25, name = "Alice", is_active = True - 簡單運算： 加、減、乘、除、餘數等。 - 範例：x = 10 + 5, y = 7 * 2 - 輸入與輸出： - 使用 input() 來接受使用者輸入，使用 print() 來顯示輸出。 - 範例： <pre>name = input("What's your name") print("Hello, " + name)</pre> 3. 流程控制 - 條件語句 (if/else)： - 判斷某些條件是否為真，來決定要執行的操作。 - 範例： <pre>age = 18 if age >= 18: print("You are an adult.") else: print("You are a minor.")</pre> - 迴圈 (for 和 while)： - 用於重複執行某些操作。 - 範例： <pre>for i in range(5): print(i)</pre> 4. 簡單的實踐項目 - 計算器：讓學生寫一個簡單的加法、減法、乘法、除法計算器。 - 例子： <pre>num1 = float(input("Enter first number: ")) num2 = float(input("Enter second number: ")) operation = input("Choose operation (+, -, *, /): ") if operation == "+": print(num1 + num2) elif operation == "-": print(num1 - num2) elif operation == "*": print(num1 * num2) elif operation == "/": print(num1 / num2) else: print("Invalid operation")</pre>	數、浮點數、字串、布林值)、條件語句、循環等。 2. 安裝 Python 開發環境 - 指導他們下載並安裝 Python，並介紹 Python 的執行環境 (如 IDLE、Jupyter Notebook 或 Visual Studio Code)。 - 教他們如何執行簡單的 Python 程式，像是 print("Hello, World!")，來確認他們的環境設置正確。 3. 學習 Python 基本語法 - 變數與資料型別： <pre>x = 5 # 整數 y = 3.14 # 浮點數 name = "Alice" # 字串 is_active = True # 布林</pre> - 輸入與輸出： <pre>name = input("請輸入你的名字: ") # 輸入 print("你好, ", name) # 輸出</pre> - 運算符：+, -, *, /, % <pre>a = 10 b = 2 print(a + b) # 加法</pre> 等 4. 條件語句與循環 - if 條件語句： <pre>age = int(input("請輸入你的年齡: ")) if age >= 18: print("你是成年人") else: print("你是未成年人")</pre> - while 循環： <pre>count = 0 while count < 5: print(count) count += 1</pre> - for 循環： <pre>for i in range(5): print(i)</pre> 5. 簡單的專案 讓學習者實際動手做一些簡單的專案，比如： - 計算機程式：讓用戶輸入兩個數字並進行加減乘除。 - 猜數字遊戲：電腦隨機選擇一個數字，用戶猜測直到猜對為止。	但需要用特定的格式來表達。 2. 安裝 Python 和編輯器 - 讓學生安裝 Python (可以下載 Python 官網的最新版本)。 - 建議使用簡單易用的編輯器，如 VSCode 或 Thonny，讓學生能夠輕鬆編寫和運行程式。 3. 從簡單的程式開始 - Hello World 範例：介紹最基本的程式，讓學生了解如何用 print 輸出訊息。 <pre>print("Hello, World!")</pre> - 解釋程式的結構，如何執行程式，並向學生展示他們編寫的程式如何執行。 4. 介紹基本語法和變數 - 變數：解釋什麼是變數，如何存儲資料。例如： <pre>name = "Alice" age = 25 print(name) print(age)</pre> - 資料型別：介紹基本的資料型別 (整數、浮點數、字串) 及其用法。 5. 控制流程：條件判斷與迴圈 - 條件判斷 (if 語句)：介紹如何根據條件執行不同的程式碼。 <pre>age = 18 if age >= 18: print("You are an adult.") else: print("You are a minor.")</pre> - 迴圈 (for、while)：介紹如何重複執行某些操作。 <pre>for i in range(5): print(i)</pre> 6. 函數的使用 - 解釋如何寫函數來重複使用某些程式碼，減少重複性。 <pre>def greet(name): print(f"Hello, {name}!") greet("Alice") greet("Bob")</pre> 7. 逐步挑戰問題
--	--	---	--

	5. 進一步的學習 ● 函數：教學生如何將程式碼分成功能模塊，並使用函數來組織程式。 ● 資料結構：介紹列表、字典等資料結構，讓學生能夠處理複雜的資料。 6. 持續練習與挑戰 鼓勵學生多寫程式，從簡單的問題開始，如計算某個數字的階乘、列出所有質數等。隨著技能進步，可以挑戰更複雜的項目。 7. 資源與學習工具 ● 學習網站：如 Codecademy、LeetCode 等。 ● 影片教學：YouTube 上有很多適合初學者的 Python 教學。 ● 書籍推薦：《Python Crash Course》、《Automate the Boring Stuff with Python》。 學習程式是一個循序漸進的過程，鼓勵學生多動手寫程式，從錯誤中學習，並持續探索。	6. 錯誤處理與除錯 ● 教學如何識別錯誤並進行除錯。包括語法錯誤、邏輯錯誤等。 ● 使用 try-except 來處理可能的錯誤： 7. 持續練習與進階學習 <pre>try: num = int(input("請輸入一個數字:")) except ValueError: print("那不是一個有效的數字!")</pre> ● 鼓勵學員通過解決小問題來加深理解，可以使用網站如 LeetCode、HackerRank 進行練習。 ● 隨著基礎掌握，介紹更高階的概念，比如函數、模組、資料結構（列表、字典、元組等）。 通過循序漸進的方式，從簡單到複雜，逐步增加挑戰，學習者將能夠建立信心並開始編寫自己的 Python 程式。	● 在學生理解基礎語法後，可以通過實際的小範例來練習和鞏固所學，例如： ○ 輸入姓名並輸出問候語。 ○ 計算從 1 加到 10 的總和。 ○ 撰寫簡單的猜數字遊戲。 8. 理解錯誤與除錯 ● 引導學生了解常見錯誤和如何閱讀錯誤訊息。教他們如何查詢和解決問題。 9. 學習資料結構 ● 學習列表（List）和字典（Dictionary），這些是 Python 中常用的資料結構。 <pre>fruits = ["apple", "banana", "cherry"] print(fruits[0]) # apple</pre> 10. 鼓勵多練習和學習資源 ● 鼓勵學生做一些簡單的專案，如建立一個計算器、簡單的計時器或小遊戲。 ● 推薦一些線上學習資源或教程（如 Codecademy、LeetCode 或 Python 官方文檔），讓學生持續進步。 通過這樣循序漸進的方法，學生能夠逐步理解和掌握 Python 程式設計的基本概念，並逐漸增加編寫程式的信心和能力。
相關聯性	5/5	5/5	5/5
段落鮮明度	5/5	5/5	5/5
用字遣詞	5/5	5/5	5/5
內容豐富度	4/5	3/5	5/5
總分	19/20	18/20	20/20

註：研究者自行整理

本題的分數差距沒有到很大，只在於回覆內容的豐富度，整體上並無太大影響。

表 4-3-5 國家組 B2

	B2：如何將數學原理應用於生活之中？		
實驗 IP	台灣	美國	日本
回覆	將數學原理應用於日常生活中，其實是非常有用的。以下是一些例子，說明如何將數學概念與生活相結合： 1 時間管理與計劃：利用數學中的比例、分數和比率來規劃時	數學原理在生活中有許多實際應用，以下是一些例子： 1. 財務管理： ○ 預算規劃：利用加法、減法來管理收入與支出，並使用比例和百分	數學原理可以應用於生活中的很多方面，這裡有一些例子： 1. 預算管理：數學中的加法、減法、乘法和除法在日常的財務管理中非常有用。通過計算收

	間。例如，假設你有三個任務，並且知道它們的完成時間，則可以計算每個任務的優先順序，並確保時間分配最有效。 2. 預算與財務規劃：在生活中，每月的收入和支出都需要計算。利用加法、減法、乘法和除法來管理你的預算，確保收支平衡。此外，了解利率和貸款計算也可以幫助你做出更聰明的財務決策。 3. 購物折扣：當你購物時，可以利用百分比來計算折扣。例如，如果一件衣服打七折，原價 1000 元，那麼你可以計算折後價格是 1000 元乘以 0.7，即 700 元。 4. 烹飪與食譜：做菜時，你常常需要調整食譜中的比例。這是應用比率和比例的方法。如果食譜是為 4 人設計的，而你是為 2 人做飯，你可以將所有材料的量減半。 5. 運動與健康：在運動時，數學可幫助你計算心率、卡路里消耗等。對於體重訓練，計算每組的重量和次數可以幫助你確保訓練達到目標。 6. 空間與物體擺放：當你整理房間或搬家時，數學中的幾何學能幫助你計算空間大小，並計算出最佳的擺放方式。 7. 交通與路程規劃：如果你想要從 A 地點到 B 地點，計算路程所需時間的速度和距離公式可以幫助你找到最佳路徑或了解交通狀況。 這些數學概念不僅能提高你的生活效率，還能讓你在解決日常問題時更加精確和有條理。	比來計算儲蓄或貸款的利率。 ○ 投資回報：用複利公式計算投資的增值，幫助做出更好的理財決策。 2. 時間管理： ○ 使用時間除法來計劃每項任務所需時間，並確保每項活動都能有效分配時間。 3. 購物比較： ○ 當購買商品時，計算折扣或比價，這涉及到百分比和比例。例如，計算 20% 折扣後的價格，或計算每單位價格（如每公斤的成本）來決定哪個商品更划算。 4. 烹飪： ○ 在烹飪時，可以利用比例來調整食譜。例如，將食譜中的份量乘以 2 或減少一半，以滿足家庭人數或食材限制。 5. 建築與設計： ○ 在房屋建設或裝修中，幾何學原理被廣泛應用。你會用到面積、體積、角度等計算來設計空間或安排家具。 6. 健身與健康： ○ 當你計算卡路里消耗、身體質量指數（BMI）或運動的強度時，都在運用數學原理。例如，計算你的最大心率或運動效果，涉及到代數和統計。 7. 行車與導航： ○ 開車時，會用到速率、時間和距離的公式來估算到達目的地的時間，或計算油費和行駛距離等。 數學原理無處不在，學會將它們運用在日常生活中，可以幫助我們更加高效和精確地處理各種問題。	支、儲蓄和投資回報，可以幫助你更好地規劃自己的財務狀況。 2. 購物與折扣：在購物時，折扣和優惠的計算是基於百分比的數學運算。例如，計算折扣後的價格，你可以用原價乘以折扣率，再從原價中減去這個值。 3. 時間管理：當你安排日程時，數學中的時間加法減法可以幫助你分配時間，確保所有活動都能夠順利進行。比如計算一個活動的持續時間，或者安排在特定時間內完成多項任務。 4. 烹飪：食譜中的比例和份量也涉及到數學。當你需要調整食譜的份量時，可以利用比例來進行縮放。例如，將一個四人的食譜調整為兩人份或六人份。 5. 工程與建設：日常生活中的建設和修繕常常需要運用幾何學原理。例如，計算牆面面積來確定需要多少油漆，或者計算房間的體積來確定適合的空調尺寸。 6. 健康與健身：數學也可以用來跟踪你的健身進度，例如計算卡路里消耗、運動時間和心率區間等，來幫助你達成健身目標。 7. 交通與導航：無論是計算最短路線還是估算旅行所需時間，數學中的比例、比率和速度公式都是必須的。 這些例子展示了數學是如何滲透到我們日常生活中的，幫助我們做出更明智的決策。
相關聯性	5/5	5/5	5/5
段落鮮明度	5/5	5/5	5/5
用字遣詞	5/5	5/5	5/5
內容豐富度	3/5	5/5	4/5
總分	18/20	20/20	19/20

註：研究者自行整理

總分差的幾分在於內容豐富度，整體回覆內容差距不大。

表 4-3-6 國家組 B3

	B3：如何將科學原理應用於生活之中？		
實驗 IP	台灣	美國	日本
回覆	將科學原理應用於日常生活中，可以幫助我們更有效地解決問題，提升生活品質。以下是一些實際的例子： 1. 熱力學原理： ○ 冰箱和空調：這些家用電器利用熱力學中的熱交換原理來保持食物冷卻或創造舒適的室內環境。冰箱將熱量從內部轉移到外部，這是基於熱流動的基本原理。 2. 摩擦力： ○ 穿鞋：選擇適合的鞋子可以減少摩擦，讓走路更加輕鬆。運動鞋的設計通常會利用摩擦原理來增加穩定性，防止滑倒。 3. 浮力原理： ○ 游泳：當你進入水中時，你會感受到一股向上的力量，這是由浮力原理引起的。根據阿基米德原理，物體在液體中所受的浮力等於它排開的液體的重力。 4. 光學原理： ○ 眼鏡和鏡子：眼鏡根據光的折射原理幫助我們改善視力，鏡子利用光的反射原理來形成影像。 5. 生物學原理： ○ 營養和健康：了解人體如何消化食物和吸收營養，可以幫助你制定更健康的飲食計劃，保持身體健康。例如，蛋白質對肌肉生長至關重要，而水則對體內的各項生理反應不可或缺。 6. 物理學原理： ○ 滑板：在滑板上滑行之時，利用動能和慣性原理可以讓你輕鬆地保持平衡和移動。學會如何運用這些物理概念能夠讓你更有效地進行運動。 將科學原理融入生活中，不僅能讓你更好地理解世界，還能提升你的生活效率和舒適度。	將科學原理應用於生活中，可以讓我們更高效、更智慧地解決問題。以下是一些常見的應用範例： 1. 物理學原理： ○ 槓桿原理：當你需要搬動重物時，使用槓桿可以讓你用較小的力量達到較大的效果。例如，用長棍撬起石頭或是推門時，使用槓桿原理可以讓你的操作更輕鬆。 ○ 摩擦力：選擇適合的鞋子或車胎可以減少不必要的摩擦力，提高效率和安全性。例如，冬季使用防滑鞋墊來防止滑倒。 2. 化學反應： ○ 酸鹼中和：當你遇到胃酸過多時，可以使用碳酸氫鈉（小蘇打）來中和胃酸，達到舒緩作用。這是化學中酸鹼中和反應的實際應用。 ○ 氧化反應：食物保存中，使用抗氧化劑（如維生素 C、維生素 E）可以減緩食物的氧化過程，延長保鮮期。 3. 生物學原理： ○ 生物適應性：人們在不同的環境中可以根據生物學原理調整生活方式，比如高山環境中，人體會產生更多紅血球來攜帶更多氧氣。 ○ 免疫系統：保持良好的衛生習慣和飲食習慣可以增強免疫系統，幫助我們對抗病菌和病毒。 4. 數學原理： ○ 機率和統計：在投資或是規劃行程時，了解機率和統計的基本概念能幫助你做出更明智的決策。 ○ 比例與比例問題：當你要做食物分配或改變食譜的份量時，掌握比例關係可以精確計算需要的食材量。 5. 心理學原理： ○ 行為心理學：知道如何利用正向強化（如獎勵或鼓勵）來改變習慣或提升工作效率，可以在日常生活中產生積極的變化。	將科學原理應用於日常生活中，可以幫助我們更好地理解周圍的世界，並提高生活質量。以下是一些常見的例子： 1. 節能減碳：了解熱力學原理，尤其是熱量轉換和能量守恒，能幫助我們在家裡節約能源。例如，關閉不使用的電器，選擇節能燈泡，利用保溫材料減少熱損失。 2. 清潔和消毒：科學研究告訴我們，水的酸鹼度（pH 值）和溫度對清潔效果有很大影響。例如，熱水比冷水更有效地清潔油脂，而酸性或鹼性的清潔劑對不同的污漬有不同的效果。 3. 健康飲食：生物學和營養學的原理可以幫助我們選擇更健康的食物。了解食物中的營養成分，如蛋白質、碳水化合物和脂肪，能幫助我們搭配均衡飲食，保持良好的身體狀態。 4. 運動和身體健康：運動科學告訴我們，如何設計有效的運動計劃來提高體能，減少受傷的風險。了解人體肌肉運作原理，可以幫助我們進行正確的運動訓練，達到增強肌力或提高耐力的效果。 5. 心理學與行為：了解心理學原理能幫助我們更好地管理情緒、處理壓力和建立正向習慣。例如，透過認知行為療法的原理，我們可以改善消極思維模式，提升心理健康。 6. 天氣和氣候：了解氣象學的基礎原理可以幫助我們應對不同的天氣情況，選擇適當的衣物，或預測天氣變化，以便安排日常活動。 將這些科學原理實際應用到生活中，不僅能提高效率，還能促進健康、環保和可持續發展。

	○ 敵人會從畫面上方隨機生成並向下移動。 ○ 當子彈擊中敵人時，得分增加，敵人被摧毀。 ○ 如果敵人觸及畫面底部，遊戲結束，顯示 "Game Over" 和重啟提示。 6. 重複操作：遊戲結束後，玩家可以點擊畫面來重新開始遊戲。 使用說明： ● 將這段代碼粘貼到 CodePen 的 HTML 和 JS 部分即可運行。 ● 點擊 "Start Game" 開始遊戲，使用箭頭鍵移動，按空格鍵發射子彈。 ● 當遊戲結束後，點擊畫面可以重新開始。 這個遊戲是基本版本，您可以根據需要進行擴展，例如添加音效、更複雜的敵人行為和圖像等。	2. CSS：設置畫布尺寸和基本佈局。 3. JavaScript：包含了遊戲的邏輯，主要是玩家控制、子彈發射、敵人生成和碰撞檢測等功能。 如何使用： 1. 啟動遊戲：按下任意鍵開始遊戲。 2. 遊戲操作： ○ 左右箭頭控制玩家移動。 ○ 空格鍵發射子彈。 3. 遊戲結束：當敵人碰到玩家或超過畫面底部，遊戲結束，按空格鍵重新開始。 你可以將這段代碼分別放到 CodePen 的 HTML、CSS 和 JavaScript 區域中來運行它。	並提示玩家按 "R" 鍵重啟遊戲。 如何使用： 1. 你可以將這段程式碼直接粘貼到 中，並運行它。 2. 當你按下 "Start Game" 按鈕時，遊戲會啟動。使用左右箭頭鍵或 'A'、'D' 鍵來移動，空格鍵發射子彈。 這樣，你就可以在 CodePen 中運行並測試這個簡單的射擊遊戲了！
相關聯性	5/5	4/5	5/5
段落鮮明度	5/5	5/5	5/5
用字遣詞	5/5	5/5	5/5
執行結果	5/5	5/5	4/5
總分	20/20	19/20	19/20

註：研究者自行整理

在本問題裡，美國組沒有分數，但其他都有做到，所以在相關聯性扣分。日本組沒辦法結束，所以視為沒有重新開始的畫面，所以在執行結果的地方扣分。

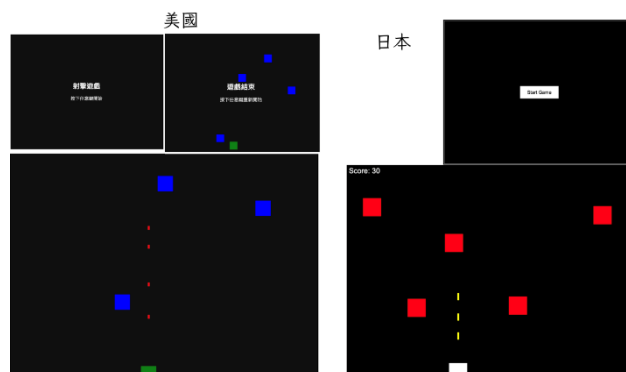


表 4-3-8 國家組 C2

	C2：生成一個可以透過使用者的輸入來解讀一元一次方程式的 python 一元一次方程式計算機程式並可應用在 CodePen 網頁中。		
實驗 IP	台灣	美國	日本
回覆	要在 CodePen 中使用 Python 寫一個簡單的計算機來解一元一次方程式，通常會需要將 Python 程式轉換為 JavaScript，因為 CodePen 是基於 Web 的前端平台，它只支持 HTML、CSS、JavaScript，而不能直接運行 Python 代碼。 下面是一個用 JavaScript 寫的解一元一次方程式的計算機程式，可以應用於 CodePen 中。這個程式允許使用者輸入方程式的係數並計算出解。 <pre><DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>一元一次方程式計算機</title> </head> <body> <div>解一元一次方程式 $ax + b = 0$</div> <div class="equation-form"> <label for="a">a (係數)</label> <input type="number" id="a" required/> <label for="b">b (常數)</label> <input type="number" id="b" required/> <button type="submit">計算</button> </div> <div>方程式解:</div> <p id="result"></p> </body></pre> 程式說明： - 用戶會被要求輸入方程式的兩個係數 a 和 b，這是來自方程式 $ax + b = 0$。 - 當用戶點擊 "計算解" 按鈕時，表單會觸發 submit 事件並進行計算。 - 程式會檢查 a 是否為零，如果是零，則顯示 "方程式無解" 的提示。 - 如果 a 不為零，計算並顯示方程式的解 $x = -b/a$。 你可以將這段代碼直接粘貼到 CodePen 的 HTML 區域，並運行它來計算一元一次方程式的解。	以下是一个可以解读一元一次方程式的 Python 程式，并可以应用在 CodePen 网页中的解决方案。为了在网页上运行，你可以使用 Python 的 Web 框架（如 Flask 或 Django），但为了适应 CodePen 环境，通常会用 JavaScript 来实现该功能，而不是 Python。以下是一个基于 JavaScript 的版本，你可以在 CodePen 上运行它。 JavaScript 代码（适用于 CodePen）： 功能说明： - 用户输入方程式（如 $2x+5=11$）。 - 代码会通过正则表达式解析并解出方程式中的 x 值。 - 如果方程式格式正确，程序将返回 x 的值；如果输入不符合预期格式，则提示用户输入有效的方程式。 如何使用： - 复制并粘贴这段代码到 CodePen 中。 - 在网页上输入方程式并点击 "解算方程式" 按钮，查看解答。 这是一个简单的解方程式的实现，你可以根据需要扩展功能。如果你有其他需求，欢迎告诉我！	以下是可以解析並解決一元一次方程式的 Python 程式碼，並且可以嵌入到 CodePen 網頁中的簡單應用。這個程式會透過用戶輸入一個標準的一元一次方程式（例如 $2x+3=7$），並自動解算出 x 的值。 由於 CodePen 是一個基於 JavaScript 的平台，因此 Python 程式無法直接在 CodePen 上運行。你可以使用 JavaScript 來實現相同的功能。下面是如何用 JavaScript 實現一個一元一次方程式解算器的範例，並能嵌入到 CodePen 中： 如何使用： <pre><DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>一元一次方程式計算機</title> <style> body { font-family: Arial, sans-serif; text-align: center; margin-top: 50px; } input { width: 100px; } button { padding: 5px; font-size: 16px; } </style> </head> <body> <div>解一元一次方程式</div> <div class="equation-form"> <input type="text" id="equation" placeholder="例如: 2x+5=11" /> <button onclick="solveEquation()">解算</button> </div> <div>結果:</div> <p id="result"></p> </body></pre> - 將上述 HTML 代碼複製並粘貼到 CodePen 中的 HTML 部分。 - 用戶可以在輸入框中輸入一個標準格式的一元一次方程式，並點擊「解方程式」按鈕來查看解答。 這段 JavaScript 代碼會解析方程式，並根據一元一次方程式的解法 $ax+b=cax + b = cax+b=c$，計算出 xxx 的值。如果輸入的方程式格式不正確，它會提示用戶重新輸入。
相關聯性	4/5	5/5	5/5
段落鮮明度	5/5	5/5	5/5
用字遣詞	5/5	5/5	5/5
執行結果	5/5	5/5	5/5
總分	19/20	20/20	20/20

註：研究者自行整理

在本實驗裡，日本和美國可以做到的功能是相同的，而對照組(台灣)的較為麻煩，需要先將等號後面的式子變為零，所以在相關聯性扣分。

美國 日本
一元一次方程式解算器 解一元一次方程式

請輸入一元一次方程式：例如： $2x+5=11$

$$3x+4=13$$
$$x=3$$

請輸入一個方程式，例如: $2x + 3 = 7$

$$3x+4=13$$

表 4-3-9 國家組 C3

	C3：生成一個元素週期表有元素的名字、代號、與分類的python 元素週期表程式，且點擊後會出現該元素的解釋、英文、分類、代號、名稱，同時此程式可應用在 CodePen 網頁中。		
實驗 IP	台灣	美國	日本
回覆	要在 CodePen 中使用 Python 並生成一個元素週期表，通常 Python 本身無法直接在瀏覽器中運行。你可以使用 HTML、CSS 和 JavaScript 來實現這個需求。這裡是如何用這三種語言創建一個動態元素週期表的範例： 1. HTML（結構） <pre></head> <body> <div interactive-periodic-table> <div class="table"> <!-- Elements will be injected dynamically by JavaScript --> </div> </div></pre> 2. CSS（樣式） <pre>table { width: 100%; border-collapse: collapse; text-align: center; font-family: sans-serif; font-size: 0.9em; color: #333; } tbody tr { height: 40px; } tbody tr td { padding: 5px; width: 33.33%; }</pre> 3. JavaScript（功能） <pre>const periodicTable = { elements: [{ name: "Hydrogen", symbol: "H", atomicNumber: 1, category: "Nonmetals" }, { name: "Helium", symbol: "He", atomicNumber: 2, category: "Noble gases" }, { name: "Lithium", symbol: "Li", atomicNumber: 3, category: "Alkali metals" }, // ... more elements ...] }; function renderTable() { const tbody = document.querySelector("tbody"); tbody.innerHTML = ""; periodicTable.elements.forEach(element => { const tr = document.createElement("tr"); tr.innerHTML = `       <td>\${element.name}</td>       <td>\${element.symbol}</td>       <td>\${element.atomicNumber}</td>       <td>\${element.category}</td>     `; tbody.appendChild(tr); }); }</pre> 代碼解釋： - HTML 部分：我們創建了一個簡單的網頁結構，包含一個元素週期表和一個顯示元素詳情的區域。 - CSS 部分：我們使用 grid 來建立元素週期表的佈局， 說明： - HTML：這是網頁的基本結構，#periodic-table 是元素週期表的容器，#element-info 是用來顯示選中元素詳情的容器。		

	- 在 elements 陣列中，您可以根據需要添加更多的元素，每個元素包含名稱、符號、類別、原子數和描述。 - getCategoryColor 函數根據元素的類別返回不同的顏色（可以擴展更多的類別和顏色）。 - 當您點擊某個元素時，JavaScript 會動態顯示該元素的詳細信息。 如何在 CodePen 中使用： - 創建一個新的 Pen。 - 將 HTML 放入 HTML 編輯器中。 - 將 CSS 放入 CSS 編輯器中。 - 將 JavaScript 放入 JavaScript 編輯器中。 這樣，您就能在 CodePen 中看到互動式元素週期表，並能查看每個元素的詳細信息。	並且對每個元素應用了 hover 效果，讓其在鼠標懸停時放大。 - 另外，legend 是用來顯示元素詳細資訊的區域，並在點擊元素後顯示。 JavaScript 部分： - showElementInfo 是當用戶點擊某個元素時觸發的函數，會顯示該元素的名稱、符號、英文名稱和分類。 - closeLegend 用來關閉詳細資訊視窗。 如何使用： - 在 CodePen 或其他 HTML 編輯器中，將上面的 HTML 代碼粘貼到 HTML 區域。 - 你可以繼續添加更多的元素，並根據元素的屬性設置不同的顏色和分類。 - 當用戶點擊元素時，會顯示該元素的詳細資料。 這個簡單的範例展示了如何創建一個互動式的元素週期表，並能夠提供點擊元素後顯示的更多信息。你可以根據需要擴展元素，或者調整顏色和設計。	CSS： 設定了元素週期表的佈局和樣式，並且加入了滑鼠懸停效果，當滑鼠懸停於元素上時會改變背景顏色。 JavaScript： 定義了一些元素資料（你可以根據需要添加更多元素），並設置了點擊元素時顯示該元素的詳細資訊。 如何使用： - 將這些代碼放到 CodePen 中的對應區域（HTML、CSS 和 JavaScript）即可。 - 點擊元素週期表中的任何元素，會顯示該元素的名稱、符號、原子數、分類和英文名稱。 你可以根據需要增加更多的元素資料或調整樣式。
相關聯性	4/5	5/5	5/5
段落鮮明度	5/5	5/5	5/5
用字遣詞	5/5	5/5	5/5
執行結果	2/5	1/5	1/5
總分	16/20	16/20	16/20

註：研究者自行整理

在本實驗裡，因為美國組和日本組都沒有元素的解釋，所以會在執行結果的地方扣分，而對照組(台灣)因為只有兩種元素而日本和美國組有三種，所以也在相關聯性的地方多扣分，所以總分會是一樣的。

美國

日本



表 4-3-10 實驗組(國家 IP)評分統整表

問題	總分		
	台灣	美國	日本
A1	20/20	17/20	17/20
A2	17/20	17/20	19/20
A3	16/20	16/20	16/20
B1	19/20	18/20	20/20
B2	18/20	20/20	19/20
B3	19/20	19.5/20	20/20
C1	20/20	19/20	19/20
C2	19/20	20/20	20/20
C3	16/20	16/20	16/20
總共	164/180	162.5/180	166/180

註：研究者自行整理

在本實驗裡，分數差距並不大，推測 ChatGPT 給的回覆和國家 IP 並無太大關聯。

四、實驗四：實驗組(提問法)

本實驗以提問法作為我的操縱變因，會以我說話的方式直接敘述和《ChatGPT 指令大全》的公式進行修改，控制變因會以主帳號、台灣 IP 以、實驗時間為每星期二下午 1~2 點並使用中文向 ChatGPT 提問，以下是我的提問問題以及 ChatGPT 的回覆：

(一)操縱變因：我說話的方式直接敘述

(二)操縱變因：《ChatGPT 指令大全》的公式進行修改

表 4-4-1 提問法組 A1

	A1：C#是一種程式語言嗎？	A1：你扮演 程式老師的角色，我需要理解 C#是一種程式語言嗎？請用淺顯易懂方式描述。
提問法	直接敘述	《ChatGPT 指令大全》的公式進行修改
回覆	是的，C#（發音為 "C-sharp"）是一種程式語言。它由微軟開發，主要用於構建各種應用程序，特別是 Windows 平台上的應用程式。C# 是基於 .NET 框架的一種現代、物件導向的語言，適用於桌面應用程式、網站、移動應用程式等開發。 它的語法類似於 C 和 C++，但在設計上更加簡單和安全。C# 支援許多現代程式語言的特性，例如垃圾回收、異常處理和多線程處理等。	是的，C# 是一種程式語言。它是一種由微軟（Microsoft）開發的語言，通常用來開發各種應用程式，比如桌面應用程式、網站、甚至是手機應用程式。C# 是一種高階語言，這意味著它比較接近人類的語言，容易理解和使用。 簡單來說，C# 就像是你和電腦溝通的一種方式，你可以用 C# 編寫指令，告訴電腦該做什麼。這些指令組成了程式，電腦會按照這些指令來執行任務。舉個例子，你可以用 C# 撰寫一個程式來計算數字，顯示在螢幕上，或是創建一個遊戲角色的動作

		等等。C# 是非常強大且靈活的，適用於許多不同的開發需求。
相關聯性	5/5	5/5
段落鮮明度	4/5	5/5
用字遣詞	5/5	5/5
內容豐富度	5/5	3/5
總分	19/20	18/20

註：研究者自行整理

此問題雖然在分數上看來沒有太大差距，實則差很多。指令大全組的內容使用了很高明的手法比喻，讓人一聽就懂，但缺點就是資料內容會較少，較多的會是解釋。

表 4-4-2 提問法組 A2

	A2：數學有幾年的歷史？	A2：你扮演數學老師的角色，我需要理解數學有幾年的歷史。請用淺顯易懂方式描述。
提問法	直接敘述	《ChatGPT 指令大全》的公式進行修改
回覆	數學的歷史可以追溯到古代文明，約有 4000 多年的歷史。最早的數學記錄來自古埃及和古巴比倫，這些文化在公元前 2000 年左右已經開始使用數字系統來解決日常問題，如測量土地和建造建築物。 隨著時間的推移，古希臘數學家如畢達哥拉斯和歐幾里得等人發展了更為抽象的數學理論，這些理論成為現代數學的基礎。數學經過幾千年的發展，形成了今天我們所知的各種分支，如算術、幾何學、代數、微積分等。 總的來說，數學的歷史非常悠久，跨越了多個世紀，並且隨著人類文明的進步不斷演變和發展。	數學的歷史已經有幾千年的時間了。它最早起源於古代文明，最早的數學形式主要與計算、測量和記錄有關。以下是數學發展的簡單歷程： 古代埃及和美索不達米亞（大約公元前 3000 年）：最早的數學應用出現在古代的工程和貿易上。埃及人使用數學來建造金字塔，並計算土地的面積。而美索不達米亞人（今天的伊拉克地區）則發明了早期的數字系統，用於天文觀察和計算。 古希臘時期（公元前 500 年左右）：古希臘的數學家，如畢達哥拉斯和歐幾里得，開始進行抽象數學研究。他們不僅僅關注計算，還開始思考數學的理論和原則，這是數學發展的重要一步。 印度和阿拉伯數學（公元 500 年到 1200 年）：在印度，數學家發明了“零”的概念，並創造了十進位數字系統。這些發明對後來的數學發展至關重要。在中世紀時期，阿拉伯學者對數學進行了大量的研究，他們翻譯了許多希臘的數學著作，並且進一步發展了代數。 歐洲文藝復興時期（公元 1500 年到 1600 年）：這個時期數學逐漸發展成為一門獨立的學科。數學家開始探索更多抽象的數學結構，並且發展了微積分、代數和幾何等分支。 現代數學（公元 1700 年以後）：隨著時間的推移，數學越來越專業化，涵蓋了更廣泛的領域，包括數論、拓撲學、數學邏輯、概率論等。數學的應用範圍也越來越大，從科學到工程，再到計算機科學等。

		總結來說，數學有超過 5000 年的歷史，它從早期的計算工具和實用技術發展成為今天的抽象科學，影響了我們的生活和各種科技領域。
相關聯性	5/5	5/5
段落鮮明度	3/5	5/5
用字遣詞	5/5	5/5
內容豐富度	3/5	5/5
總分	16/20	20/20

註：研究者自行整理

不同於上一個問題，這個問題的差距實在是非常的大，指令大全組使用了分享列點和類似時間線的方式讓人可以清楚地明白先後順序。

表 4-4-3 提問法組 A3

	A3：水的化學式是什麼？	A3：你扮演化學老師的角色，我需要理解水的化學式。請用淺顯易懂方式描述。
提問法	直接敘述	《ChatGPT 指令大全》的公式進行修改
回覆	水的化學式是 H_2O 。這表示每個水分子由兩個氫原子（H）和一個氧原子（O）組成。	水的化學式是 H_2O 。這表示水由兩個氫原子（H）和一個氧原子（O）組成。這個化學式告訴我們，水分子裡有兩個氫原子和一個氧原子，它們通過化學鍵連接在一起。 想像氫原子像小球一樣，氧原子則像一個大球。氫原子和氧原子之間有一種叫做共價鍵的聯繫，就像兩個小球用彈性繩緊緊拉在一起。這樣組成的水分子在生活中非常常見，無論是海洋、河流還是我們日常使用的水。 所以，水就是由這些氫和氧原子組成的一個小小分子。
相關聯性	5/5	5/5
段落鮮明度	1/5	5/5
用字遣詞	5/5	5/5
內容豐富度	1/5	5/5
總分	12/20	20/20

註：研究者自行整理

在這個問題裡，就可以清楚地發現兩個不同變因對回覆的影響非常的巨大，指令大全組的回覆使用了比喻使人更能了解，和直接敘述組比起來，差了十萬八千里。

表 4-4-4 提問法組 B1

	B1：如何教一個完全沒有程式基礎的人寫 python 程式？	B1：詳細的說明如何教一個完全沒有程式基礎的人寫 python 程式？
提問法	直接敘述	《ChatGPT 指令大全》的公式進行修改
回覆	1. 介紹程式和 Python - 什麼是程式？：解釋程式是用來讓電腦執行指令的語言，程式可以解決問題、做計算，甚至建立應用程式。 - 為什麼選 Python？：Python 是簡單且易讀的語言，適合初學者，且有大量的資源和社群支持。 2. 基礎語法介紹 - 變數與資料類型： - 變數：如何儲存數字、文字等資料。 - 常見資料類型：整數 (int)、浮點數 (float)、字串 (str)、布林值 (bool)。 - 範例：age = 25, name = "Alice", is_active = True - 簡單運算： 加、減、乘、除、餘數等。 - 範例：x = 10 + 5, y = 7 * 2 - 輸入與輸出： - 使用 input() 來接受使用者輸入，使用 print() 來顯示輸出。 - 範例： <pre>name = input("What's your name? ") print("Hello, " + name)</pre> 3. 流程控制 - 條件語句 (if/else)： - 判斷某些條件是否為真，來決定要執行的操作。 - 範例： <pre>age = 18 if age >= 18: print("You are an adult.") else: print("You are a minor.")</pre> - 迴圈 (for 和 while)： - 用於重複執行某些操作。 - 範例： <pre>for i in range(5): print(i)</pre> 4. 簡單的實踐項目 - 計算器：讓學生寫一個簡單的加法、減法、乘法、除法計算器。	要教一個完全沒有程式基礎的人寫 Python 程式，最好的方式是從基礎概念開始，逐步建立信心，並讓他們通過實際編程來學習。以下是具體的步驟： 1. 介紹程式語言的基本概念 首先，介紹程式語言的基本概念，例如： - 什麼是程式語言？：程式語言是人們用來與電腦溝通的一種方式，類似於我們用語言交流。 - 程式的結構：解釋程式通常由命令（語句）組成，這些命令告訴電腦該做什麼。 2. 安裝和設定 Python 開發環境 - 安裝 Python，並確保學生知道如何在本地計算機上設置 Python 編輯器或 IDE（例如 PyCharm 或 Visual Studio Code）。 - 教授如何使用 Python 交互式命令行，這是學習初期的一個重要工具。 3. 從簡單的「Hello, World!」開始 - 向學生展示如何在 Python 中打印訊息。這是程式設計中經典的第一步 <pre>print("Hello, World!")</pre> 4. 學習基本語法 引導學生了解 Python 的基本語法，這是寫程式的基礎。 - 變數：介紹變數和數據類型（如整數、浮點數、字串）。 <pre>x = 5 name = "Alice"</pre> - 數學運算符：加法、減法、乘法、除法等。 <pre>sum = 3 + 4 product = 2 * 3</pre> - 輸入：如何從用戶處獲取輸入。 <pre>age = input("What is your age? ")</pre> 5. 流程控制 讓學生理解程式的流程控制結構。 - 條件語句 (if, else)： <pre>if x > 10: for i in range(5):10") else: print(i) han 10")</pre> - 迴圈 (for, while)： 6. 函數的概念 介紹如何使用函數來組織程式碼。 - 定義函數： 教授學生函數的結構，如何調用函數及傳遞參數。 <pre>def greet(name): print(f"Hello, {name}!") greet("Alice")</pre> 7. 列表和字典

	○ 例子： <pre>num1 = float(input("Enter first number: ")) num2 = float(input("Enter second number: ")) operation = input("Choose operation (+, -, *, /): ") if operation == "+": print(num1 + num2) elif operation == "-": print(num1 - num2) elif operation == "*": print(num1 * num2) elif operation == "/": print(num1 / num2) else: print("Invalid operation")</pre> 5. 進一步的學習 函數：教學生如何將程式碼分成功能模塊，並使用函數來組織程式。 資料結構：介紹列表、字典等資料結構，讓學生能夠處理複雜的資料。 6. 持續練習與挑戰 鼓勵學生多寫程式，從簡單的問題開始，如計算某個數字的階乘、列出所有質數等。隨著技能進步，可以挑戰更複雜的項目。 7. 資源與學習工具 ● 學習網站：如 Codecademy、LeetCode 等。 ● 影片教學：YouTube 上有很多適合初學者的 Python 教學。 ● 書籍推薦：《Python Crash Course》、《Automate the Boring Stuff with Python》。 學習程式是一個循序漸進的過程，鼓勵學生多動手寫程式，從錯誤中學習，並持續探索。	介紹資料結構，如列表和字典，並教學生如何操作它們。 ● 列表： <pre>fruits = ["apple", "banana", "cherry"] print(fruits[0]) # 輸出 'apple'</pre> ● 字典： <pre>student = {"name": "John", "age": 21} print(student["name"]) # 輸出 'John'</pre> 8. 錯誤處理 介紹如何捕捉和處理錯誤，並教學生如何調試程式。 ● 使用 try 和 except 來處理例外情況： <pre>try: number = int(input("Enter a number: ")) except ValueError: print("That's not a valid number!")</pre> 9. 實踐與小挑戰 鼓勵學生寫一些簡單的程式來鞏固學到的知識。例如： ● 輸入一個數字，計算它的平方。 ● 設計一個簡單的計算器。 ● 創建一個程式來檢查給定的數字是否為質數。 10. 逐步增加難度 一旦學生掌握了基礎，可以進一步介紹： ● 模塊和庫：介紹如何使用 Python 標準庫或第三方庫來增強功能。 ● 面向對象編程：簡要介紹類和物件的概念，讓學生理解如何在程式中創建和使用物件。 11. 持續練習與反饋 程式設計的學習需要不斷地實踐。建議學生參加線上的編程挑戰或課程，並提供反饋。 總結： ● 用簡單的語言解釋每一個概念，並提供具體的例子。 ● 讓學生有足夠的練習機會，這樣他們才能夠熟悉和理解 Python 的基本語法。 ● 透過不斷的小挑戰來增強他們的學習動機。 這樣的步驟可以幫助完全沒有程式基礎的人逐步掌握 Python。
相關聯性	5/5	5/5
段落鮮明度	5/5	5/5
用字遣詞	4/5	5/5
內容豐富度	4/5	5/5
總分	18/20	20/20

註：研究者自行整理

在此問題中 ChatGPT 的回覆其實也沒有太大差別，但因為指令大全組的用字遣詞讓人更容易理解，所以在直接敘述組的用字遣詞扣分。

表 4-4-5 提問法組 B2

	B2：如何將數學原理應用於生活之中？	B2：詳細的說明如何將數學原理應用於生活之中？
提問法	直接敘述	《ChatGPT 指令大全》的公式進行修改
回覆	將數學原理應用於日常生活中，其實是非常有用的。以下是一些例子，說明如何將數學概念與生活相結合： 時間管理與計劃： 利用數學中的比例、分數和比率來規劃時間。例如，假設你有三個任務，並且知道它們的完成時間，則可以計算每個任務的優先順序，並確保時間分配最有效。 預算與財務規劃： 在生活中，每月的收入和支出都需要計算。利用加法、減法、乘法和除法來管理你的預算，確保收支平衡。此外，了解利率和貸款計算也可以幫助你做出更聰明的財務決策。 購物折扣： 當你購物時，可以利用百分比來計算折扣。例如，如果一件衣服打七折，原價 1000 元，那麼你可以計算折後價格是 1000 元乘以 0.7，即 700 元。 烹飪與食譜： 做菜時，你常常需要調整食譜中的比例。這是應用比率和比例的方法。如果食譜是為 4 人設計的，而你是為 2 人做飯，你可以將所有材料的量減半。 運動與健康： 在運動時，數學可幫助你計算心率、卡路里消耗等。對於體重訓練，計算每組的重量和次數可以幫助你確保訓練達到目標。 空間與物體擺放： 當你整理房間或搬家時，數學中的幾何學能幫助你計算空間大小，並計算出最佳的擺放方式。 交通與路程規劃： 如果你想要從 A 地點到 B 地點，計算路程所需時間的速度和距離公式可以幫助你找到最佳路徑或了解交通狀況。 這些數學概念不僅能提高你的生活效率，還能讓你在解決日常問題時更加精確和有條理。	數學原理在生活中的應用非常廣泛，以下是一些具體的例子，幫助理解如何將數學原理應用於日常生活： 財務管理： 預算規劃與理財： 數學中的加法、減法、乘法和除法幫助我們計算日常開支、收入和儲蓄目標。比如，若希望每月儲蓄一定金額，可以利用簡單的乘法和除法來計算每月儲蓄的金額，或使用利息公式來計算長期儲蓄的增長。 複利計算： 複利計算公式用於計算儲蓄或貸款的利息，這對於了解如何最好地管理長期投資至關重要。 時間管理： 優化時間使用： 使用數學中的分數和百分比來計算如何將時間有效地分配到不同任務。例如，如果一天有 8 小時工作，並且計劃花 2 小時閱讀，可以計算出閱讀所占時間的百分比（2 小時 / 8 小時 = 25%）。 計算工作效率： 利用數學的速率概念，可以估算完成一項任務所需的時間，或計算完成工作的效率。 購物和折扣： 價格比較： 在購物時，利用百分比來計算折扣價格，幫助比較不同商店的優惠。例如，某商品原價 \$100，打 20% 的折扣後價格為 \$80，這樣可以比較不同商店的折扣。 單位價格計算： 用數學方法計算每個單位的價格（如每公升、每克），有助於做出更聰明的購物選擇。 健康和飲食： 卡路里計算： 了解食物的營養成分，可以使用加法和乘法來計算每餐的總卡路里，或通過計算每月的總卡路里攝入量來保持健康飲食。 BMI（身體質量指數）計算： 使用數學公式來計算自己的 BMI（體重公斤數除以身高的平方），這有助於了解自己的體重是否處於健康範圍內。 旅行和路程規劃： 速率與時間： 在旅行中，使用數學公式來計算旅行時間。例如，假設你要開車 200 公里，車速為 50 公里/小時，那麼預計需要 $200 \div 50 = 4$ 小時。 距離、速度與時間的關係： 使用這些數學概念來計算交通工具的最佳出發時間或到達時間，從而幫助計劃行程。 建築和裝修： 計算面積和體積： 在裝修家居時，需要計算牆壁的面積（長 $\times$ 寬）來確定需要多少油漆或壁紙；計算房間的體積（長 $\times$ 寬 $\times$ 高）來選擇適當大小的家具。

		● 角度和比例：使用三角學中的角度和比例，來確保家具、裝飾物等能夠合理地安置在空間中。 7. 問題解決： ● 數學模型：可以用數學模型來解決許多現實生活中的問題。例如，在設計交通系統或物流安排時，使用線性方程式來計算最佳路線或最佳運輸方式。 ● 概率和統計：在決策過程中，使用概率來預測某些事件的發生機率，或者通過統計分析來理解趨勢和模式，從而做出更好的選擇。 總的來說，數學是一種有力的工具，能夠幫助我們做出更明智的決策和解決各種生活中的問題。數學不僅僅是一門學科，它在我們的日常生活中無處不在，從簡單的計算到複雜的問題解決，數學的應用無處不在。
相關聯性	5/5	5/5
段落鮮明度	5/5	5/5
用字遣詞	5/5	5/5
內容豐富度	3/5	5/5
總分	18/20	20/20

註：研究者自行整理

在這個問題裡差距就比較沒有那麼大了，差的兩分在於內容量和版面配置，雖然內容量並沒有太大差距，但指令大全組的使用淺顯易懂的版面配置，讓回覆更好理解。

表 4-4-6 提問法組 B3

	B3：如何將科學原理應用於生活之中？	B3：詳細的說明如何將科學原理應用於生活之中？
提問法	直接敘述	《ChatGPT 指令大全》的公式進行修改
回覆	將科學原理應用於日常生活中，可以幫助我們更有效地解決問題，提升生活品質。以下是一些實際的例子： 1. 熱力學原理： ○ 冰箱和空調：這些家用電器利用熱力學中的熱交換原理來保持食物冷卻或創造舒適的室內環境。冰箱將熱量從內部轉移到外部，這是基於熱流動的基本原理。 2. 摩擦力： ○ 穿鞋：選擇適合的鞋子可以減少摩擦，讓走路更加輕鬆。運動鞋的設計通常會利用摩擦原理來增加穩定性，防止滑倒。 3. 浮力原理： ○ 游泳：當你進入水中時，你會感受到一股向上的力量，這是由浮力原理引起的。根據阿基米德原理，物體在液體中所受的浮力等於它排開的液體的重力。 4. 光學原理： ○ 眼鏡和鏡子：眼鏡根據光的折射原理幫助我們改善視力，鏡子利用光的反射原理來形成影像。	科學原理可以應用於我們的日常生活中，從改善健康、節省能源到提高工作效率，科學原理不僅能讓我們更好地理解世界，還能提供實用的解決方案。以下是一些具體的例子，展示如何將科學原理應用於生活中： 1. 物理學：節能與熱效率 物理學中的熱傳導、對流和輻射原理可以幫助我們更有效地使用能源。例如，選擇適合的絕緣材料來減少熱損失，或者選擇節能家電來降低能源消耗。在冬天使用厚窗簾來減少室內熱量流失，或者使用保溫杯來保持飲品溫度，這些都可以運用熱學原理來提高效率。 2. 生物學：健康與飲食 生物學的知識幫助我們了解如何保持健康。例如，了解人體如何消化食物以及如何吸收營養，能幫助我們制定合理的飲食計劃。植物如何利用光合作用生長，也能啟發我們選擇富含維生素和礦物質的食物，提升免疫系統的功能。

	5. 生物學原理： ○ 營養和健康：了解人體如何消化食物和吸收營養，可以幫助你制定更健康的飲食計劃，保持身體健康。例如，蛋白質對肌肉生長至關重要，而水則對體內的各項生理反應不可或缺。 6. 物理學原理： ○ 滑板：在滑板上滑行時，利用動能和慣性原理可以讓你輕鬆地保持平衡和移動。學會如何運用這些物理概念能夠讓你更有效地進行運動。 將科學原理融入生活中，不僅能讓你更好地理解世界，還能提升你的生活效率和舒適度。	3. 化學：清潔與環保 化學反應原理可以應用於家務清潔和環保。例如，酸鹼中和反應可用於清除水垢或廚房油污。氧化還原反應則是金屬生鏽或食物氧化變質的原因，了解這些過程有助於預防物品腐壞或延長其使用壽命。 4. 心理學：情緒與行為管理 心理學幫助我們理解行為和情緒的運作。例如，情緒調節理論讓我們學會如何應對壓力或情緒波動，並通過冥想、深呼吸等技巧改善心理健康。認知行為療法（CBT）則提供了一種有效的方式來改變負面思維模式，促進積極行為。 5. 數學：計劃與管理 數學原理可以幫助我們在日常生活中進行計劃和資源管理。例如，時間管理技巧（如番茄工作法）能夠提高工作效率，數學中的概率原理可以幫助我們做出更有信心的決策，精確計算預算與開支。 6. 地理學：氣候與天氣預測 知道地球的氣候變化和天氣預測原理，有助於我們規劃日常活動。例如，在了解氣象模式後，我們可以更好地選擇穿著和攜帶物品，如雨具、保暖衣物等，也能提前做好防範自然災害的準備。 透過這些例子，我們可以看到科學原理如何幫助我們在日常生活中做出更明智的選擇，並解決各種問題。
相關聯性	5/5	5/5
段落鮮明度	4/5	5/5
用字遣詞	5/5	5/5
內容豐富度	4/5	5/5
總分	18/20	20/20

註：研究者自行整理

在這個問題裡，總分差距其實也沒有到很大。但因為指令大全組使用粗體和一些格式使文字更好閱讀，且有更多補充，所以總分會更高一點。

表 4-4-7 提問法組 C1

	C1：生成一個可以重複操作且有起始畫面的 python 射擊遊戲程式並可應用在 CodePen 網頁中。	C1：你現在是一個 Python 專家，請幫我用 Python 寫一個函式，它需要做到可以重複操作且有起始畫面的 Python 射擊遊戲程式並可應用在 CodePen 網頁中。
提問法	直接敘述	《ChatGPT 指令大全》的公式進行修改
回覆	要在 CodePen 中實現一個可以重複操作且有起始畫面的 Python 射擊遊戲，首先，您需要將 Python 程式碼轉換為 JavaScript，因為 CodePen 是基於 Web 技術（HTML, CSS, JavaScript）來運行的，無法直接執行 Python 程式碼。	要在 CodePen 上運行一個 Python 射擊遊戲並非直接可行，因為 CodePen 主要支援 HTML、CSS 和 JavaScript，而不支援 Python。但你可以使用 Python 的 pygame 庫來創建一個射擊遊戲，並將遊戲部署在網站上。如果你希望使用 JavaScript 來實現這個

	我會提供一個簡單的 JavaScript 程式碼範例，這個範例包含了射擊遊戲的基本結構，包括起始畫面、遊戲邏輯，以及重複操作的機制。您可以將其放入 CodePen 的 JS 部分來測試。 這段代碼的功能： 起始畫面：當頁面加載完成後，顯示一個歡迎界面，包含一個按鈕可以開始遊戲。 遊戲邏輯： ○ 玩家可以使用左/右箭頭鍵控制移動，空格鍵發射子彈。 ○ 敵人會從畫面上方隨機生成並向下移動。 ○ 當子彈擊中敵人時，得分增加，敵人被摧毀。 ○ 如果敵人觸及畫面底部，遊戲結束，顯示 "Game Over" 和重啟提示。 重複操作：遊戲結束後，玩家可以點擊畫面來重新開始遊戲。 使用說明： ● 將這段代碼粘貼到 CodePen 的 HTML 和 JS 部分即可運行。 ● 點擊 "Start Game" 開始遊戲，使用箭頭鍵移動，按空格鍵發射子彈。 ● 當遊戲結束後，點擊畫面可以重新開始。 這個遊戲是基本版本，您可以根據需要進行擴展，例如添加音效、更複雜的敵人行為和圖像等。	遊戲，則可以用它來創建類似的遊戲並將其嵌入在 CodePen 中。 如果你堅持要用 Python 來創建遊戲並進行部署，我建議使用 Flask 或 Django 來建立一個 Web 應用，再使用 JavaScript 來處理前端與 Python 後端的交互。若你想繼續使用 Python 實現射擊遊戲並準備將它轉換為可以在 CodePen 中運行的格式，這裡有一個簡單的 Python 範例來創建一個射擊遊戲： 使用 Pygame 創建射擊遊戲 這是用 Pygame 的 Python 範例，創建一個基礎的射擊遊戲： 如何將其轉換為可在 Web 頁面中運行的格式： <pre>import pygame import random import time # 初始化 Pygame pygame.init()</pre> 1. 使用 Flask 或 Django 創建一個簡單的後端來提供遊戲。 2. 使用 pygame 遊戲引擎處理遊戲邏輯和渲染。 3. 前端用 JavaScript 來呈現遊戲，並與後端進行交互。 不過，實際要將此轉換到 CodePen 並不是一個直接的步驟。如果你想知道如何將其進行 Web 化，請告訴我，我可以提供相關的步驟與技術參考！
相關聯性	5/5	5/5
段落鮮明度	5/5	5/5
用字遣詞	5/5	5/5
執行結果	5/5	4/5
總分	20/20	19/20

註：研究者自行整理

在本實驗裡，因為指令大全組沒有起始畫面而對照組有，所以指令大全組在執行結果的地方扣了分。

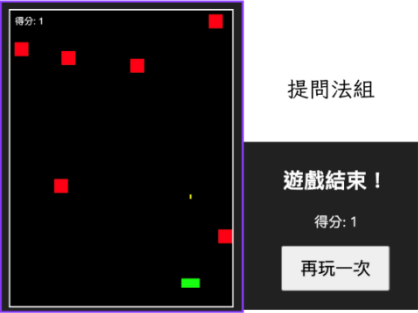


表 4-4-8 提問法組 C2

	C2：生成一個可以透過使用者的輸入來解讀一元一次方程式的 python 一元一次方程式計算機程式並可應用在 CodePen 網頁中。	C2：你現在是一個 Python 專家，請幫我用 Python 寫一個函式，它需要做到可以透過使用者的輸入來解讀一元一次方程式的 Python 一元一次方程式計算機，並可應用在 CodePen 網頁中。
提問法	直接敘述	《ChatGPT 指令大全》的公式進行修改
回覆	要在 CodePen 中使用 Python 寫一個簡單的計算機來解一元一次方程式，通常會需要將 Python 程式轉換為 JavaScript，因為 CodePen 是基於 Web 的前端平台，它只支持 HTML、CSS、JavaScript，而不能直接運行 Python 代碼。 下面是一個用 JavaScript 寫的解一元一次方程式的計算機程式，可以應用於 CodePen 中。這個程式允許使用者輸入方程式的係數並計算出解。 程式說明： <pre><DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>一元一次方程式計算機</title> </head> <body> <p>解一元一次方程式: $ax + b = 0$ (a ≠ 0)</p> <div id="equation-form"> <label for="a">a (係數)</label> <input type="number" id="a" required></div> <label for="b">b (常數項)</label> <input type="number" id="b" required></div> <button type="submit">計算解</button> </div> <div id="result"></div> <p id="result"></p> </body></pre> - 用戶會被要求輸入方程式的兩個係數 a 和 b，這是來自方程式 $ax + b = 0$。 - 當用戶點擊 "計算解" 按鈕時，表單會觸發 submit 事件並進行計算。 - 程式會檢查 a 是否為零，如果是零，則顯示 "方程式無解" 的提示。 - 如果 a 不為零，計算並顯示方程式的解 $x = -b/a$。 你可以將這段代碼直接粘貼到 CodePen 的 HTML 區域，並運行它來計算一元一次方程式的解。	好的，我可以幫你寫一個簡單的 Python 函式，用來解一元一次方程式。這個函式會接受使用者輸入的方程式並計算其解。 以下是這個 Python 函式的範例： <pre>import sympy as sp def solve_linear_equation(): # 提示使用者輸入一元一次方程式 equation_input = input("請輸入一元一次方程式，例如: 2*x + 3 = 7: ") # 解析方程式 x = sp.symbols('x')</pre> 這個函式會： - 請使用者輸入一個方程式，例如 $2*x + 3 = 7$。 - 利用 sympy 這個符號計算庫解析方程式並計算解。 - 返回方程式的解。 這樣的函式可以在任何支援 Python 的環境中運行，例如 Jupyter Notebook 或本地開發環境。至於在 CodePen 上直接使用 Python，CodePen 主要支援前端語言（如 HTML、CSS 和 JavaScript），因此你無法直接在 CodePen 上運行 Python 代碼。不過，你可以使用像是 Brython 這樣的工具來在 CodePen 上運行 Python 代碼。Brython 是一個將 Python 轉換為 JavaScript 的工具，可以讓你在瀏覽器端執行 Python 程式碼。 如果你有更多問題，或需要進一步的幫助，隨時告訴我！
相關聯性	5/5	5/5
段落鮮明度	5/5	5/5
用字遣詞	5/5	5/5
執行結果	5/5	0/5
總分	20/20	15/20

註：研究者自行整理

在本實驗裡，因為指令大全組的計算機沒辦法算出答案，所以在執行結果的地方是零分。

提問法組
一元一次方程式計算機 (使用 Python)

請輸入一元一次方程式 (例如： $2*x + 3 = 7$)：

解方程式

表 4-4-9 提問法組 C3

	C3：生成一個元素週期表有元素的名字、代號與分類的 python 元素週期表程式，且點擊後會出現該元素的解釋、英文、分類、代號、名稱，同時此程式可應用在 CodePen 網頁中。	C3：你現在是一個 Python 專家，請幫我用 Python 寫一個函式，它需要做到可以生成一個元素週期表有全部元素的名字、代號與分類，且點擊後會出現該元素的解釋、英文、分類、代號、名稱，同時此程式可應用在 CodePen 網頁中。
提問法	直接敘述	《ChatGPT 指令大全》的公式進行修改
回覆	要在 CodePen 中使用 Python 並生成一個元素週期表，通常 Python 本身無法直接在瀏覽器中運行。你可以使用 HTML、CSS 和 JavaScript 來實現這個需求。這裡是如何用這三種語言創建一個動態元素週期表的範例： - HTML（結構） <pre> <!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>Periodic Table</title> <link rel="stylesheet" href="styles.css"> </head> <body> <div Interactive Periodic Table</div> <div class="table"> <!-- Elements will be inserted dynamically by JavaScript --> </div> <div class="element-details" id="element-details"> <div id="element-name">Element Name</div> <div id="element-symbol">Element Symbol</div> <div id="element-category">Element Category</div> <div id="element-number">Element Number</div> <div id="element-description">Element Description</div> </div> <script src="script.js"></script> </body> </pre> - CSS（樣式） <pre> body { font-family: Arial, sans-serif; background-color: #f0f0f0; } const elements = [{ name: "Hydrogen", symbol: "H", category: "Nonmetal", number: 1, description: "Hydrogen is a colorless, odorless, and tasteless gas that is the most abundant element in the universe." }, { name: "Helium", symbol: "He", category: "Noble Gas", number: 2, description: "Helium is a colorless, odorless, and tasteless gas that is the second most abundant element in the universe." } // Add other elements here...]; function createElementButton(element) { const button = document.createElement('div'); button.className = 'element'; button.style.backgroundColor = getCategoryColor(element.category); } </pre> - JavaScript（功能） 詳解： - HTML：設置了頁面結構，並創建了一個元素容器（.table）和一個顯示元素詳細信息的區域（.element-details），後者將顯示點擊的元素的詳細信息。 - CSS：提供了基本的樣式，使元素以表格形式顯示並進行了一些動效設置（如：元素縮放的懸停效果）。 - JavaScript：創建了每個元素的按鈕，並將其顯示在表格中。當用戶點擊某個元素時，該元素的詳細信息會顯示在下方。 說明： - 在 elements 陣列中，您可以根據需要添加更多的元素，每個元素包含名稱、符號、類別、原子數和描述。	為了完成這個需求，我們需要先設計一個包含元素週期表所有元素的資料結構。然後，再設計一個前端展示頁面，當用戶點擊元素時，會顯示該元素的詳細資訊。我將使用 HTML、CSS 和 JavaScript 來構建此頁面，並利用 Python 來處理資料。以下是完整的方案設計步驟和程式碼： - Python 處理資料 我們需要建立一個元素資料的 Python 函數，輸出元素的基本資訊（如名稱、代號、顏色和分類）。由於元素週期表比較大，我們將資料儲存在一個字典中，並且每個元素擁有名稱、代號、顏色、分類等屬性。 - HTML 結構 在 CodePen 中，我們可以使用 HTML 來結構化頁面，並且為每個元素創建一個點擊區域。這些元素會使用 CSS 來調整顏色和樣式。 - CSS 樣式 我們會為元素提供顏色，並設置一個簡單的網格布局來顯示元素週期表。 - JavaScript 控制邏輯 <pre> def generate_periodic_table(): elements = [{"symbol": "H", "name": "Hydrogen", "color": "#FFFFFF", "category": "Nonmetal"}, {"symbol": "He", "name": "Helium", "color": "#000000", "category": "Noble Gas"}] <!-- DOCTYPE html --> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>Periodic Table</title> </head> <body> <div class="periodic-table"> <div class="element"> <div class="element-header"> <div class="element-symbol">H</div> <div class="element-name">Hydrogen</div> </div> <div class="element-body"> <div class="element-category">Nonmetal</div> <div class="element-description">Hydrogen is a colorless, odorless, and tasteless gas that is the most abundant element in the universe.</div> </div> </div> <div class="element"> <div class="element-header"> <div class="element-symbol">He</div> <div class="element-name">Helium</div> </div> <div class="element-body"> <div class="element-category">Noble Gas</div> <div class="element-description">Helium is a colorless, odorless, and tasteless gas that is the second most abundant element in the universe.</div> </div> </div> </div> </body> </pre> 在 JavaScript 中，我們會動態生成元素週期表，並在用戶點擊元素時顯示詳細資訊。 完整說明： - HTML：用來呈現元素週期表的結構，並且顯示元素詳細資料的區塊。 - CSS：用來設計元素的顏色、大小以及布局，讓元素週期表更具有視覺吸引力。 - JavaScript：負責動態生成元素區塊，並處理用戶點擊後顯示相應元素詳細資訊的邏輯。 如何使用： - 將上述代碼分別放入 HTML、CSS 和 JavaScript 文件中。 - 在 CodePen 中，將 HTML 部分放在

	● getCategoryColor 函數根據元素的類別返回不同的顏色（可以擴展更多的類別和顏色）。 ● 當您點擊某個元素時，JavaScript 會動態顯示該元素的詳細信息。 如何在 CodePen 中使用： 5. 創建一個新的 Pen。 6. 將 HTML 放入 HTML 編輯器中。 7. 將 CSS 放入 CSS 編輯器中。 8. 將 JavaScript 放入 JavaScript 編輯器中。 這樣，您就能在 CodePen 中看到互動式元素週期表，並能查看每個元素的詳細信息。	HTML 窗口，CSS 放在 CSS 窗口，JavaScript 放在 JS 窗口中。 3. 點擊元素後，右側會顯示該元素的詳細資訊。 這樣，你就可以在 CodePen 中展示元素週期表並顯示每個元素的詳細資料了！
相關聯性	5/5	5/5
段落鮮明度	5/5	5/5
用字遣詞	5/5	5/5
執行結果	2/5	1/5
總分	17/20	16/20

註：研究者自行整理

在本實驗裡，因為指令大全組只有一個元素，且沒有原子序，所以在執行結果的地方分數會較對照組低。

He

提問法組

Helium

Symbol: He

Category: Noble Gas

Description: Helium is a colorless, odorless, tasteless, non-toxic, inert, monatomic gas.

表 4-4-10 實驗組(提問法)評分統整表

問題	總分	
	星期二下午 1~2 點	星期五下午 3~4 點
A1	19/20	18/20
A2	16/20	20/20
A3	12/20	20/20
B1	18/20	20/20
B2	18/20	20/20
B3	18/20	20/20
C1	20/20	19/20
C2	20/20	15/20
C3	17/20	16/20
總共	158/180	168/180

註：研究者自行整理

在本實驗裡，指令大全組和對照組的差距實在是很大，推測是因為要求 (Prompt) 的長短的關係。

五、實驗五：實驗組(語言)

本實驗以語言作為我的操縱變因，會以中文、英文以及西班牙文進行提問，控制變因會以實驗時間為每星期二下午 1~2 點、主帳號、台灣 IP 以及我說話的方式去向 ChatGPT 提問，以下是我的提問問題以及 ChatGPT 的回覆：

(一)操縱變因：中文

(二)操縱變因：英文

(三)操縱變因：西班牙文

表 4-5-1 語言組 A1

	A1：C#是一種程式語言嗎？		
實驗 IP	中文	英文	西班牙文
回覆	是的，C#（發音為 "C-sharp"）是一種程式語言。它由微軟開發，主要用於構建各種應用程序，特別是 Windows 平台上的應用程序。C# 是基於 .NET 框架的一種現代、物件導向的語言，適用於桌面應用程式、網站、移動應用程式等開發。 它的語法類似於 C 和 C++，但在設計上更加簡單和安全。C# 支援許多現代程式語言的特性，例如垃圾回收、異常處理和多線程處理等。	Yes, C# (pronounced "C-sharp") is a programming language developed by Microsoft. It's part of the .NET framework and is used for building a wide range of applications, from web and desktop software to mobile and game development. C# is known for its simplicity, efficiency, and object-oriented features, making it a popular choice for developers.	Sí, C# (se pronuncia "C-sharp") es un lenguaje de programación desarrollado por Microsoft. Es un lenguaje moderno, orientado a objetos y se utiliza principalmente para el desarrollo de aplicaciones en la plataforma .NET. C# es muy popular para crear aplicaciones de escritorio, web y videojuegos, especialmente con el motor Unity para desarrollo de juegos.
相關聯性	5/5	5/5	5/5
段落鮮明度	5/5	4/5	4/5
用字遣詞	5/5	5/5	5/5
內容豐富度	5/5	4/5	4/5
總分	20/20	18/20	18/20

註：研究者自行整理

在本問題裡，分數其實並無太大差距，差距在於，內容量與分段，推測原因為用不同語言提問時，會查到不同的資料、研究，根據 ChatGPT 本人所說，當你使用中文進行提問時，查找的資料會有約 70%是中文資料，約 30%是其他語言，且通常英語會佔較大比重。

表 4-5-2 語言組 A2

	A2：數學有幾年的歷史？		
實驗 IP	中文	英文	西班牙文
回覆	數學的歷史可以追溯到古代文明，約有 4000 多年的歷史。最早的數學記錄來自古埃及和古巴比倫，這些文化在公元前 2000 年左右已經開始使用數字系統來解決日常問題，如測量土地和建造建築物。	Mathematics, as a formal field of study, has been around for thousands of years. The earliest records of mathematical concepts date back to ancient civilizations such as the Sumerians around 3000 BCE, who developed early counting systems.	Las matemáticas, como campo formal de estudio, tienen miles de años de historia. Se pueden rastrear sus orígenes a las antiguas civilizaciones de Mesopotamia, Egipto y Grecia, donde comenzaron a desarrollarse

	隨著時間的推移，古希臘數學家如畢達哥拉斯和歐幾里得等人發展了更為抽象的數學理論，這些理論成為現代數學的基礎。數學經過幾千年的發展，形成了今天我們所知的各種分支，如算術、幾何學、代數、微積分等。總的來說，數學的歷史非常悠久，跨越了多個世紀，並且隨著人類文明的進步不斷演變和發展。	The ancient Egyptians and Babylonians also contributed to the development of mathematics around 2000 BCE, particularly in geometry, algebra, and arithmetic. In short, mathematics is at least 5,000 years old, and many of the fundamental principles have evolved and been refined over the centuries.	conceptos como la aritmética y la geometría. Si consideramos que las matemáticas comenzaron a ser sistematizadas y estructuradas, podríamos decir que tienen al menos 4,000 a 5,000 años de antigüedad. Sin embargo, algunos de los principios fundamentales de las matemáticas, como la noción de contar y medir, se han usado desde mucho antes.
相關聯性	5/5	5/5	5/5
段落鮮明度	5/5	4/5	4/5
用字遣詞	5/5	5/5	5/5
內容豐富度	5/5	3/5	4/5
總分	20/20	17/20	18/20

註：研究者自行整理

本問題分數差距在於內容量，只是在英文組終有一個較為特別例子，是前面幾個實驗都不曾提到的字詞蘇美人，但其實廣義來說就是古美索不達米亞人，所以前面其實也都有提到，推測是因為蘇美這個字詞在台灣較少提到，所以在英文組才會出現。

表 4-5-3 語言組 A3

	A3：水的化學式是什麼？		
實驗 IP	中文	英文	西班牙文
回覆	水的化學式是 H ₂ O。這表示每個水分子由兩個氫原子（H）和一個氧原子（O）組成。	The chemical formula of water is H ₂ O. This means that each molecule of water consists of two hydrogen (H) atoms bonded to one oxygen (O) atom.	La fórmula química del agua es H ₂ O. Esto significa que una molécula de agua está compuesta por dos átomos de hidrógeno (H) y uno de oxígeno (O).
相關聯性	5/5	5/5	5/5
段落鮮明度	1/5	1/5	1/5
用字遣詞	5/5	5/5	5/5
內容豐富度	1/5	1/5	1/5
總分	12/20	12/20	12/20

註：研究者自行整理

回覆的內容有些許差別但在語句的意思上無差別。

表 4-5-4 語言組 B1

	B1：如何教一個完全沒有程式基礎的人寫 python 程式？		
實驗 IP	中文	英文	西班牙文
回覆	教一個完全沒有程式基礎的人寫 Python 程式可以分成幾個階段，每個階段注重基礎概念、實踐練習，逐步建立理解。以下是一個簡單的教學架構：	Teaching someone with no programming foundation to write a Python program requires a step-by-step approach. Here's a simple guide to help them get started：	Enseñar a alguien sin experiencia previa en programación a escribir un programa en Python puede ser un desafío emocionante. Aquí tienes

	1. 介紹程式和 Python ● 什麼是程式？：解釋程式是用來讓電腦執行指令的語言，程式可以解決問題、做計算，甚至建立應用程式。 ● 為什麼選 Python？：Python 是簡單且易讀的語言，適合初學者，且有大量的資源和社群支持。 2. 基礎語法介紹 ● 變數與資料類型： ○ 變數：如何儲存數字、文字等資料。 ○ 常見資料類型：整數 (int)、浮點數 (float)、字串 (str)、布林值 (bool)。 ○ 範例：age = 25, name = "Alice", is_active = True ● 簡單運算： ○ 加、減、乘、除、餘數等。 ○ 範例：x = 10 + 5, y = 7 * 2 ● 輸入與輸出： ○ 使用 input() 來接受使用者輸入，使用 print() 來顯示輸出。 ○ 範例： <pre>name = input("What's your name") print("Hello, " + name)</pre> 3. 流程控制 ● 條件語句 (if/else)： ○ 判斷某些條件是否為真，來決定要執行的操作。 ○ 範例： <pre>age = 18 if age >= 18: print("You are an adult.") else: print("You are a minor.")</pre> ● 迴圈 (for 和 while)： ○ 用於重複執行某些操作。 ○ 範例： <pre>for i in range(5): print(i)</pre> 4. 簡單的實踐項目 ● 計算器：讓學生寫一個簡單的加法、減法、乘法、除法計算器。	1. Start with the Basics Begin with explaining what programming is and why Python is a great language to start with : ● What is programming? Programming is giving instructions to a computer to perform tasks. ● Why Python? It's easy to read and write, with a simple syntax, and is used in many fields like web development, data science, and automation. 2. Introduce Python Syntax Show how Python looks. Here's the general flow : ● Print Statements : Introduce basic printing, which outputs information. <pre>print("Hello, World!")</pre> ● Explain the concept : "This command tells the computer to display the text on the screen." 3. Variables and Data Types Teach the concept of variables and the different data types in Python : ● Strings, Integers, and Floats : <pre>name = "Alice" # String age = 25 # Integer height = 5.7 # Float</pre> ● Explain variables : "A variable stores information, like a box where you can keep data." ● Explain types : "The type of data tells us what kind of value it is (e.g., a number or text)." 4. Basic Operations Introduce basic arithmetic operations and how to use them in Python : <pre>number1 = 5 number2 = 3 print(number1 + number2) # Addition print(number1 - number2) # Subtraction</pre> 5. Control Flow (If-Else Statements) Teach them how to make decisions in a program with if statements. 6. Loops (For and While Loops) <pre>age = 20 if age >= 18: print("You are an adult.") else: print("You are a minor.")</pre>	algunos pasos que puedes seguir para facilitar el proceso : 1. Explicar la importancia de la programación ● Qué es un programa : Explica qué es un programa de manera sencilla, como una serie de instrucciones que la computadora sigue para realizar una tarea. ● Usos prácticos : Da ejemplos de cómo la programación puede ser útil, como automatizar tareas repetitivas o desarrollar aplicaciones interesantes. 2. Introducción a Python ● Lenguaje fácil de aprender : Destaca que Python es un lenguaje con sintaxis sencilla y es muy popular para principiantes. ● Instalar Python : Enséñales a instalar Python en su computadora (desde python.org). ● Usar un editor de texto : Puedes recomendar usar un entorno de desarrollo como Visual Studio Code o simplemente el IDLE que viene con Python. 3. Conceptos básicos de programación ● Variables : Explica qué son las variables (espacios donde se guardan datos) y cómo usarlas. Ejemplo : <pre>x = 5 nombre = "Juan"</pre> ● Tipos de datos : Habla sobre los tipos básicos, como números, cadenas de texto (strings), y booleanos (True/False). ● Operadores : Enséñales cómo realizar operaciones matemáticas básicas. <pre>a = 10 b = 5 suma = a + b print(suma) # Salida: 15</pre> 4. Estructuras de control
--	---	--	--

	○ 例子： <pre>num1 = float(input("Enter first number: ")) num2 = float(input("Enter second number: ")) operation = input("Choose operation (+, -, *, /): ") if operation == "+": print(num1 + num2) elif operation == "-": print(num1 - num2) elif operation == "*": print(num1 * num2) elif operation == "/": print(num1 / num2) else: print("Invalid operation")</pre> 5. 進一步的學習 ● 函數：教學生如何將程式碼分成功能模塊，並使用函數來組織程式。 ● 資料結構：介紹列表、字典等資料結構，讓學生能夠處理複雜的資料。 6. 持續練習與挑戰 鼓勵學生多寫程式，從簡單的問題開始，如計算某個數字的階乘、列出所有質數等。隨著技能進步，可以挑戰更複雜的項目。 7. 資源與學習工具 ● 學習網站：如 Codecademy、LeetCode 等。 ● 影片教學：YouTube 上有很多適合初學者的 Python 教學。 ● 書籍推薦：《Python Crash Course》、《Automate the Boring Stuff with Python》。 學習程式是一個循序漸進的過程，鼓勵學生多動手寫程式，從錯誤中學習，並持續探索。	Teach them to repeat tasks using loops. ● For loop：For a set number of times. <pre>for i in range(5): print(i)</pre> ● While loop：Until a condition is met. <pre>i = 0 while i < 5: print(i) i += 1</pre> 7. Functions Teach the importance of functions in making code reusable： <pre>def greet(name): print(f"Hello, {name}!") greet("Alice")</pre> 8. Project-based Learning Once they've grasped the basics, give them simple projects to practice： ● A program that asks for the user's name and age, then prints a message. ● A calculator that performs basic operations. ● A simple number guessing game. 9. Debugging and Practice Teach them how to troubleshoot their code： ● Common errors：Syntax errors, runtime errors, logical errors. ● The importance of testing and iterating："Try to run your program often to catch mistakes early!" 10. Encourage Exploration Let them experiment with small changes to the code and explore online resources like the Python documentation or coding platforms like Codecademy and W3Schools. Starting small and ensuring plenty of practice will help them gradually develop the confidence and skills they need to write Python programs.	● Condicionales：Explica cómo tomar decisiones en un programa con if, else, y elif. Ejemplo： <pre>edad = 18 if edad >= 18: print("Eres adulto")</pre> ● Bucles：Introduce los bucles for y while para repetir acciones. Ejemplo： <pre>for i in range(5): print(i) # Imprime Los números del 0 a 4</pre> 5. Crear programas simples ● Proyecto básico：Invítalos a escribir su primer programa simple, como un "Hola Mundo" o un programa que haga cálculos simples, como la suma de dos números ingresados por el usuario： <pre>num1 = float(input("Introduce un número: ")) num2 = float(input("Introduce otro número: ")) print(num1 + num2)</pre> 6. Fomentar la práctica ● Desafíos：Propón pequeños retos, como crear un programa que calcule el área de un círculo o un programa que convierta unidades de medida. ● Errores como oportunidad：Enséñales que los errores son normales y parte del proceso de aprendizaje. Anímalos a leer los mensajes de error y entender lo que está fallando. 7. Explorar recursos adicionales ● Tutoriales online：Recomienda tutoriales gratuitos, como los de Python.org. ● Documentación：Motívalos a consultar la documentación de Python cuando tengan dudas. 8. Avanzar poco a poco ● A medida que se sientan cómodos, pueden empezar a explorar conceptos más avanzados como funciones, listas, diccionarios, y clases.
--	--	---	--

			Es importante mantener un enfoque paciente, alentador y práctico, utilizando ejemplos que sean cercanos a sus intereses. La práctica constante les ayudará a entender mejor cómo funcionan los programas y cómo escribir los suyos propios.
相關聯性	5/5	5/5	5/5
段落鮮明度	5/5	5/5	5/5
用字遣詞	4/5	5/5	5/5
內容豐富度	4/5	5/5	4/5
總分	18/20	20/20	19/20

註：研究者自行整理

在這個問題裡內容由多到少分別是英文、西班牙文、中文，且西班牙文和英文的內容較為口語化，所以給予的分數較高。

表 4-5-5 語言組 B2

	B2：如何將數學原理應用於生活之中？		
實驗 IP	中文	英文	西班牙文
回覆	將數學原理應用於日常生活中，其實是非常有用的。以下是一些例子，說明如何將數學概念與生活相結合： 1. 時間管理與計劃：利用數學中的比例、分數和比率來規劃時間。例如，假設你有三個任務，並且知道它們的完成時間，則可以計算每個任務的優先順序，並確保時間分配最有效。 2. 預算與財務規劃：在生活中，每月的收入和支出都需要計算。利用加法、減法、乘法和除法來管理你的預算，確保收支平衡。此外，了解利率和貸款計算也可以幫助你做出更聰明的財務決策。 3. 購物折扣：當你購物時，可以利用百分比來計算折扣。例如，如果一件衣服打七折，原價 1000 元，那麼你可以計算折後價格是 1000 元乘以 0.7，即 700 元。 4. 烹飪與食譜：做菜時，你常常需要調整食譜中的比例。這是應用比率和比例的方法。如果食譜是為 4 人設計的，而你是為 2 人做飯，你可以將所有材料的量減半。 5. 運動與健康：在運動時，數學可幫助你計算心率、卡路里消耗等。對於體重訓練，計算每組的重量和次數可以幫助你確保訓練達到目標。 6. 空間與物體擺放：當你整理房間或搬家時，數學中的	Mathematics can be applied in many aspects of daily life, even beyond the classroom. Here are some practical examples： 1. Budgeting and Finance：Understanding addition, subtraction, multiplication, and percentages helps you manage your money, track expenses, and save for the future. 2. Time Management：Using concepts like scheduling, estimating durations, and prioritizing tasks involves basic arithmetic and organization, helping you make efficient use of your time. 3. Problem-Solving：Math encourages logical thinking, helping you break down complex problems into smaller, manageable parts—whether you're dealing with a work issue or everyday challenges. 4. Cooking：Recipes often require measurements and proportions, where fractions, conversions,	Aplicar los principios de las matemáticas en la vida diaria puede ser muy útil, ya que muchos conceptos matemáticos tienen aplicaciones prácticas que pueden facilitar la toma de decisiones y mejorar diversas áreas de la vida. Aquí te dejo algunos ejemplos de cómo hacerlo： 1. Presupuesto y Finanzas Personales：Las matemáticas son esenciales para llevar un control de tus finanzas. Por ejemplo, puedes usar el cálculo de porcentajes para entender descuentos, impuestos o intereses en préstamos. También puedes aplicar la aritmética para planificar tu presupuesto mensual. 2. Resolución de Problemas：Las matemáticas desarrollan habilidades de resolución de problemas que pueden aplicarse en cualquier situación de la vida, desde arreglar algo en casa hasta tomar decisiones sobre inversiones o carreras. 3. Planificación y Organización：La

	幾何學能幫助你計算空間大小，並計算出最佳的擺放方式。 7. 交通與路程規劃：如果你想要從 A 地點到 B 地點，計算路程所需時間的速度和距離公式可以幫助你找到最佳路徑或了解交通狀況。 這些數學概念不僅能提高你的生活效率，還能讓你在解決日常問題時更加精確和有條理。	and multiplication/division come in handy. 5. Travel Planning : Understanding distances, speeds, and travel times helps you estimate how long trips will take and when you'll arrive, applying basic geometry and arithmetic. 6. Home Improvement : Whether you're measuring for furniture, calculating area for paint or flooring, or determining material quantities, math ensures you get the correct amounts and measurements. 7. Fitness : Tracking your workout progress, monitoring heart rate zones, and understanding calorie intake all involve applying math concepts like averages, percentages, and ratios. 8. Shopping : Math helps you compare prices, calculate discounts, and understand unit prices to get the best deals. By integrating math into daily life, it becomes a useful tool for better decision-making, efficiency, and problem-solving!	geometría, el álgebra y otros campos matemáticos pueden ayudarte a organizar tu tiempo y tus recursos de manera más eficiente. Por ejemplo, usar la optimización para encontrar la forma más eficiente de distribuir el tiempo entre tus actividades diarias. 4. Cálculo de Distancias y Velocidades : Si conduces o viajas, las matemáticas te permiten calcular distancias, tiempos de viaje y velocidades, lo que te ayuda a planificar tus trayectos de manera eficiente. 5. Cocina : Al preparar recetas, las matemáticas son útiles para ajustar las proporciones de los ingredientes según la cantidad de personas, o para convertir unidades de medida. 6. Estadísticas y Probabilidades : En la vida cotidiana, puedes usar estos conceptos para entender las probabilidades de ciertos eventos, como en juegos o decisiones basadas en datos, y evaluar riesgos en diversas situaciones. En resumen, las matemáticas son mucho más que solo números y fórmulas; están presentes en casi todo lo que hacemos. Al comprender y aplicar estos principios, puedes mejorar tu capacidad para tomar decisiones y gestionar tus actividades diarias de manera más efectiva.
相關聯性	5/5	5/5	5/5
段落鮮明度	5/5	5/5	5/5
用字遣詞	5/5	5/5	5/5
內容豐富度	4/5	4/5	5/5
總分	19/20	19/20	20/20

註：研究者自行整理

在本問題中，分數差距其實並沒有很大，但在西班牙文組裡有一些我們平常使用中文不會用到的詞，推測是因為在西班牙文裡可能沒有那些我們會使用的專有名詞所導致。總分差距在於內容量，影響不大。

表 4-5-6 語言組 B3

B3：如何將科學原理應用於生活之中？			
實驗 IP	中文	英文	西班牙文
回覆	將科學原理應用於日常生活中，可以幫助我們更有效地解決問題，提升生活品質。以下是一些實際的例子： 7. 熱力學原理： ○ 冰箱和空調：這些家用電器利用熱力學中的熱交換原理來保持食物冷卻或創造舒適的室內環境。冰箱將熱量從內部轉移到外部，這是基於熱流動的基本原理。 8. 摩擦力： ○ 穿鞋：選擇適合的鞋子可以減少摩擦，讓走路更加輕鬆。運動鞋的設計通常會利用摩擦原理來增加穩定性，防止滑倒。 9. 浮力原理： ○ 游泳：當你進入水中時，你會感受到一股向上的力量，這是由浮力原理引起的。根據阿基米德原理，物體在液體中所受的浮力等於它排開的液體的重力。 10. 光學原理： ○ 眼鏡和鏡子：眼鏡根據光的折射原理幫助我們改善視力，鏡子利用光的反射原理來形成影像。 11. 生物學原理： ○ 營養和健康：了解人體如何消化食物和吸收營養，可以幫助你制定更健康的飲食計劃，保持身體健康。例如，蛋白質對肌肉生長至關重要，而水則對體內的各項生理反應不可或缺。 12. 物理學原理： ○ 滑板：在滑板上滑行之時，利用動能和慣性原理可以讓你輕鬆地保持平衡和移動。學會如何運用這些物理概念能夠讓你更有效地進行運動。 將科學原理融入生活中，不僅能讓你更好地理解世界，還能提升你的生活效率和舒適度。	Scientific principles can be applied to life in many ways to enhance understanding, improve decision-making, and solve problems. Here are some examples： 1. Critical Thinking and Problem Solving： Science teaches us how to approach problems logically and systematically. By using observation, experimentation, and reasoning, we can make informed decisions and solve everyday problems. 2. Health and Nutrition： The principles of biology, chemistry, and physics can help us understand how the body works, how nutrition affects health, and how diseases spread. This knowledge can guide us in making healthier lifestyle choices, from what we eat to how we exercise. 3. Environmental Awareness： Understanding ecological principles can lead to better environmental conservation. Concepts like sustainability, the impact of human activities on ecosystems, and the importance of renewable resources can shape habits that contribute to the well-being of the planet. 4. Technology and Innovation： The application of scientific principles drives technological advancements. Whether in communication, transportation, or entertainment, science continuously improves our quality of life by creating new tools and devices.	Los principios científicos pueden aplicarse a la vida cotidiana de muchas maneras, ayudando a mejorar la toma de decisiones, la resolución de problemas y la comprensión del mundo que nos rodea. Aquí te dejo algunas formas de hacerlo： 1. Pensamiento crítico： La ciencia nos enseña a analizar y cuestionar las cosas en lugar de aceptar afirmaciones sin pruebas. Aplicar este principio nos ayuda a tomar decisiones más informadas en la vida diaria, como en el ámbito de la salud, las finanzas o la educación. 2. Observación y experimentación： La ciencia se basa en observar fenómenos y experimentar para entender cómo funcionan. Puedes aplicar esto en tu vida personal al probar diferentes métodos para resolver problemas y observar cuáles son los más efectivos. 3. Resolución de problemas： La ciencia se enfoca en resolver problemas complejos de manera lógica y ordenada. Al enfrentar desafíos, puedes utilizar el método científico：identificar el problema, formular hipótesis, probar soluciones y evaluar los resultados. 4. Curiosidad y aprendizaje continuo： La ciencia fomenta una mentalidad curiosa, siempre buscando nuevas respuestas. Puedes aplicar esto al seguir aprendiendo sobre temas nuevos, ya sea mediante libros, cursos o experimentos personales. 5. Adaptación a cambios： Los avances científicos nos muestran cómo la adaptación es

		5. Psychology and Behavior : Psychological principles, such as understanding how people think, feel, and behave, can be applied to improve relationships, enhance mental well-being, and foster better communication skills. 6. Time Management and Efficiency : The scientific method, which involves testing and refining hypotheses, can be applied to personal efficiency. By experimenting with different time-management strategies, people can find what works best for them. In essence, applying scientific principles can lead to a more informed, rational, and efficient way of living.	clave para sobrevivir y prosperar. En la vida, esto se traduce en aprender a adaptarse a los cambios y estar dispuesto a modificar nuestras acciones cuando las circunstancias lo requieren. 6. Salud y bienestar : Los principios científicos están presentes en la medicina y la biología. Aplicar estos principios a la salud, como seguir una dieta balanceada, hacer ejercicio y dormir bien, puede mejorar nuestra calidad de vida. 7. Tecnología y sostenibilidad : La ciencia aplicada a la tecnología puede mejorar nuestra vida cotidiana, desde el uso de dispositivos electrónicos hasta el aprovechamiento de fuentes de energía renovable. Adoptar tecnologías sostenibles y prácticas respetuosas con el medio ambiente también es un principio científico que ayuda a cuidar el planeta. Al integrar estos principios en la vida diaria, no solo mejoramos nuestro entendimiento del mundo, sino que también tomamos decisiones más informadas y efectivas.
相關聯性	5/5	4/5	4/5
段落鮮明度	5/5	5/5	5/5
用字遣詞	5/5	5/5	5/5
內容豐富度	4/5	3/5	5/5
總分	19/20	17/20	19/20

註：研究者自行整理

在本實驗中在英文和西班牙文中不知道為什麼，有非常多『間接』有科學原理的東西，所以在相關聯性幫他們扣了分，除此之外並無太大差別。

表 4-5-7 語言組 C1

C1：生成一個可以重複操作且有起始畫面的 python 射擊遊戲程式並可應用在 CodePen 網頁中。			
實驗 IP	中文	英文	西班牙文
回覆	要在 CodePen 中實現一個可以重複操作且有起始畫面的 Python 射擊遊戲，首先，您需要將 Python 程式碼轉換為 JavaScript，因為 CodePen 是基於 Web 技術（HTML, CSS, JavaScript）來運行的，無法直接執行 Python 程式碼。 我會提供一個簡單的 JavaScript 程式碼範例，這個範例包含了射擊遊戲的基本結構，包括起始畫面、遊戲邏輯，以及重複操作的機制。您可以將其放入 CodePen 的 JS 部分來測試。 這段代碼的功能： <pre><!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>Simple Shooting Game</title> <style> body { margin: 0; overflow: hidden; background-color: black; } canvas { display: block; margin: 0 auto; background-color: black; } </style> </head> <body> <canvas id="gameCanvas"></canvas> <div id="startScreen" class="hidden"> <h1 style="color: white; text-align: center;">Welcome to the Game! <button style="display: block; margin: 0 auto; padding: 5px;">Start Game </div> </body> </html></pre> 7. 起始畫面：當頁面加載完成後，顯示一個歡迎界面，包含一個按鈕可以開始遊戲。 8. 遊戲邏輯： ○ 玩家可以使左/右箭頭鍵控制移動，空格鍵發射子彈。 ○ 敵人會從畫面上方隨機生成並向下移動。 ○ 當子彈擊中敵人時，得分增加，敵人被摧毀。 ○ 如果敵人觸及畫面底部，遊戲結束，顯示 "Game Over" 和重啟提示。 9. 重複操作：遊戲結束後，玩家可以點擊畫面來重新開始遊戲。 使用說明： ● 將這段代碼粘貼到 CodePen 的 HTML 和 JS 部分即可運行。 ● 點擊 "Start Game" 開始遊戲，使用箭頭鍵移動，按空格鍵發射子彈。 ● 當遊戲結束後，點擊畫面可以重新開始。	To create a simple, repeatable Python-based shooter program with a start screen that can be applied to CodePen, it's important to note that CodePen only supports web technologies (HTML, CSS, and JavaScript). Python is not directly supported there, but you can create a similar game using JavaScript and HTML5 for the front-end. Below is a basic structure for a shooter game with a start screen using HTML5 Canvas, CSS for styling, and JavaScript for game logic. You can paste this code into the HTML, CSS, and JS panels of CodePen. HTML (index.html) <pre><!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>Simple Shooting Game</title> <style> body { margin: 0; overflow: hidden; background-color: black; } canvas { display: block; margin: 0 auto; background-color: black; } </style> </head> <body> <canvas id="gameCanvas"></canvas> <div id="startScreen" class="hidden"> <h1 style="color: white; text-align: center;">Welcome to the Game! <button style="display: block; margin: 0 auto; padding: 5px;">Start Game </div> </body> </html></pre> CSS (style.css) <pre>body { margin: 0; overflow: hidden; background-color: black; }</pre> JavaScript (script.js) <pre>const canvas = document.getElementById('gameCanvas'); const ctx = canvas.getContext('2d'); const startScreen = document.getElementById('start-screen');</pre> How the Game Works : 1. Start Screen : Displays a welcome screen with a button to start the game. Once the button is pressed, it hides the start screen and starts the game. 2. Ship Movement : The player controls the ship using the left and right arrow keys. The spacebar is used to shoot bullets. 3. Enemy Spawn : Enemies spawn every second and move downward. The game ends when an enemy passes the bottom of the screen. Instructions to Use : 1. Paste the HTML code into the HTML section of your CodePen project. 2. Paste the CSS code into the CSS section.	A continuación te doy un ejemplo básico de un programa de disparo en Python usando Pygame. Este código no puede ejecutarse directamente en CodePen, ya que CodePen no soporta Python ni Pygame. Sin embargo, podrías adaptarlo a un entorno web con HTML, JavaScript y CSS para lograr algo similar. En este caso, para CodePen, la mejor opción sería utilizar JavaScript en lugar de Python. Código en Python (Pygame) : Este código crea un juego básico donde puedes disparar a un objetivo con el teclado. <pre>import pygame import sys</pre> Adaptación a JavaScript para CodePen Para adaptarlo a CodePen, te recomiendo usar HTML, CSS y JavaScript. Aquí tienes un ejemplo simple con JavaScript que crea un juego de disparos básico : <pre><!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> </head> <body> <div id="startScreen" class="hidden"> <h1 style="color: white; text-align: center;">Welcome to the Game! <button style="display: block; margin: 0 auto; padding: 5px;">Start Game </div> </body> </html></pre> Resumen : ● Python (Pygame) : Ideal para aplicaciones locales, no en un entorno web como CodePen. ● JavaScript (HTML5 Canvas) : Este es el código adecuado para integrarse en una página web como CodePen, donde el canvas permite gráficos y animaciones. Espero que esto te ayude a crear tu juego de disparos. Si necesitas más detalles, ¡avísame!

	這個遊戲是基本版本，您可以根據需要進行擴展，例如添加音效、更複雜的敵人行為和圖像等。	3. Paste the JavaScript code into the JS section. 4. Click the "Start Game" button to begin playing! This simple shooter game can be expanded with additional features like collision detection, scoring, power-ups, and improved graphics.	
相關聯性	5/5	4/5	4/5
段落鮮明度	5/5	5/5	5/5
用字遣詞	5/5	5/5	5/5
執行結果	5/5	5/5	1/5
總分	20/20	19/20	15/20

註：研究者自行整理

在本實驗裡英文組沒有分數，所以分數會比對照組低一些。西班牙文組沒有敵人、分數，也沒有結束，所以視為沒有重新開始的畫面，所以在執行結果的地方只能給一分。

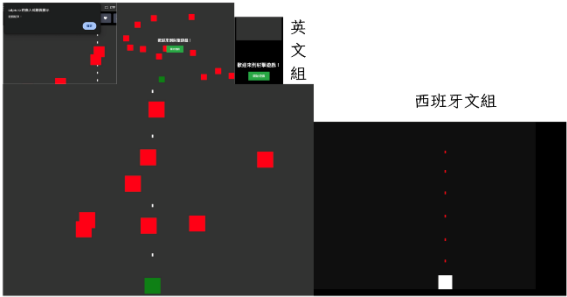


表 4-5-8 語言組 C2

	C2：生成一個可以透過使用者的輸入來解讀一元一次方程式的 python 一元一次方程式計算機程式並可應用在 CodePen 網頁中。		
實驗 IP	中文	英文	西班牙文
回覆	要在 CodePen 中使用 Python 寫一個簡單的計算機來解一元一次方程式，通常會需要將 Python 程式轉換為 JavaScript，因為 CodePen 是基於 Web 的前端平台，它只支持 HTML、CSS、JavaScript，而不能直接運行 Python 代碼。 下面是一個用 JavaScript 寫的解一元一次方程式的計算機程式，可以應用於 CodePen 中。這個程式允許使用者輸入方程式的係數並計算出解。 <pre> <DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>一元一次方程式計算機</title> </head> <body> <div>一元一次方程式: ax + b = 0 (a ≠ 0)</div> <div id="equation-form"> <label for="a">a (係數)</label> <input type="number" id="a" required/> <label for="b">b (常數)</label> <input type="number" id="b" required/> <button type="submit">計算</button> </div> <div>方程式的解:</div> <div id="result"></div> </pre>	Here is a Python program that interprets unary equations from user input and evaluates them. You can use this in a web environment like CodePen, but CodePen only supports JavaScript for client-side scripting. Below is an equivalent JavaScript implementation that can be used in CodePen : Python code (for the logic) : <pre> def evaluate_unary_equation(equation): # Remove any unwanted characters (such as spaces) eqpy = equation.replace(" ", "") # Evaluate the equation (assuming the input is safe and is a valid unary form) </pre> JavaScript code (for use in CodePen) : To make this work in a web environment like CodePen, here's the	Para crear un programa en Python que pueda interpretar ecuaciones unarias (como las funciones matemáticas que implican un solo operando) y que funcione en una página web de CodePen, podemos usar JavaScript en lugar de Python, ya que CodePen está diseñado para trabajar con tecnologías web como HTML, CSS y JavaScript. Sin embargo, te proporcionaré un ejemplo de cómo podrías escribir una función en JavaScript para este propósito. Este código interpretará ecuaciones unarias como funciones matemáticas, por ejemplo, sqrt(x) (raíz cuadrada) o abs(x) (valor absoluto), dependiendo de lo que el usuario ingrese.

	程式說明： - 用戶會被要求輸入方程式的兩個係數 a 和 b，這是來自方程式 $ax + b = 0$。 - 當用戶點擊 "計算解" 按鈕時，表單會觸發 <code>submit</code> 事件並進行計算。 - 程式會檢查 a 是否為零，如果是零，則顯示 "方程式無解" 的提示。 - 如果 a 不為零，計算並顯示方程式的解 $x = -b/a$。 你可以將這段代碼直接粘貼到 CodePen 的 HTML 區域，並運行它來計算一元一次方程式的解。	JavaScript code that evaluates unary equations provided by the user : How it works : HTML Form : The user enters the unary equation into an input field. JavaScript Evaluation : When the user clicks the "Evaluate" button, the equation is evaluated using JavaScript's <code>eval()</code> function. Display Result : The result is shown below the input field. You can copy and paste the JavaScript version into a CodePen project to try it out! The Python code is for a backend implementation but can't be directly used in CodePen, which only supports JavaScript for client-side code.	Aquí tienes un ejemplo en JavaScript para usar en CodePen : Cómo funciona : <pre><!DOCTYPE html> <html lang="es"> <head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1"></pre> - El usuario ingresa una ecuación unaria en el campo de texto, como <code>sqrt(16)</code> o <code>abs(-5)</code>. - El programa toma esa entrada y convierte las funciones <code>sqrt()</code> y <code>abs()</code> a las funciones JavaScript <code>Math.sqrt()</code> y <code>Math.abs()</code>, respectivamente. - Usa <code>eval()</code> para evaluar la ecuación y mostrar el resultado. Instrucciones para usar en CodePen : - Crea un nuevo Pen en CodePen. - Copia y pega el código HTML, CSS y JavaScript en los paneles correspondientes. - Guarda y ejecuta el código. Este programa es simple y debería funcionar bien para ecuaciones unarias que solo involucran funciones matemáticas básicas.
相關聯性	5/5	5/5	5/5
段落鮮明度	5/5	5/5	5/5
用字遣詞	5/5	5/5	5/5
執行結果	5/5	0/5	0/5
總分	20/20	15/20	15/20

註：研究者自行整理

在本實驗裡，英文組沒辦法算出答案，而西班牙文組只能接收開根號和平方值，所以視為無法算出答案，所以兩個實驗組在執行結果都只能給予零分。

英文組

Unary Equation Evaluator

Enter Unary Equation:

Error: Invalid or unexpected token

西班牙文組

Interprete de Ecuaciones Unarias

$\sqrt{16} \cdot |-5|$

$\text{sqrt}(16), \text{abs}(-5) =$

Ingrese una ecuación unaria (ejemplo: sqrt(16), abs(-5)):

Resultado: 5

表 4-5-9 語言組 C3

C3：生成一個元素週期表有元素的名字、代號與分類的 python 元素週期表程式，且點擊後會出現該元素的解釋、英文、分類、代號、名稱，同時此程式可應用在 CodePen 網頁中。			
實驗 IP	中文	英文	西班牙文
回覆	要在 CodePen 中使用 Python 並生成一個元素週期表，通常 Python 本身無法直接在瀏覽器中運行。你可以使用 HTML、CSS 和 JavaScript 來實現這個需求。這裡是如何用這三種語言創建一個動態元素週期表的範例： 1. HTML (結構) <pre><div> <div> </div> </div></pre> 2. CSS (樣式) <pre>body { font-family: Arial, sans-serif; }</pre> 3. JavaScript (功能) <pre>const periodicTable = { elements: [{ name: 'Hydrogen', symbol: 'H', atomicNumber: 1, color: 'red', classification: 'non-metal', description: 'A colorless, odorless, tasteless gas that is the lightest and most abundant element in the universe. It is essential for life and is used in a variety of industrial processes.' }, { name: 'Helium', symbol: 'He', atomicNumber: 2, color: 'blue', classification: 'noble gas', description: 'A colorless, odorless, tasteless gas that is the second lightest and most abundant element in the universe. It is used in a variety of applications, including balloons, airships, and cryogenics.' }, // ... more elements ...] };</pre> 詳解： HTML： 設置了頁面結構，並創建了一個元素容器 (.table) 和一個顯示元素詳情的區域 (.element-details)，後者將顯示點擊的元素的詳細信息。 CSS： 提供了基本的樣式，使元素以網格形式顯示並進行了一些動效設置（如：元素縮放的懸停效果）。 JavaScript： 創建了每個元素的按鈕，並將其顯示在表格中。當用戶點擊某個元素時，該元素的詳細信息會顯示在下方。 說明： - 在 elements 陣列中，您可以根據需要添加更多的元素，每個元素包含名稱、符號、類別、原子數和描述。 - getColorByCategory 函數根據元素的類別返回不同的顏色（可以擴展更多的類別和顏色）。 - 當您點擊某個元素時，JavaScript 會動態顯示該元素的詳細信息。 如何在 CodePen 中使用： 9. 創建一個新的 Pen。	To create an interactive periodic table in Python that displays an element's information (name, code, color, classification), which can be used on a CodePen webpage, we would typically use HTML, CSS, and JavaScript for the front end. While Python is not directly supported in CodePen for front-end applications, I can provide a combination of HTML, CSS, and JavaScript code that you can integrate into a CodePen project. Here's how you can set up the interactive periodic table: HTML Structure The HTML defines the structure of the periodic table. CSS (styles.css) This CSS handles the design of the <pre>body { font-family: Arial, sans-serif; }</pre> periodic table and modal popup. JavaScript (script.js) This script creates the periodic table, adds event listeners for each element, and displays a modal with the element's information. How to Use This on CodePen : - Copy the HTML code into the HTML section of your CodePen. - Copy the CSS code into the CSS section of your CodePen. - Copy the JavaScript code into the JS section of your CodePen. - Save and run your CodePen. Now, you'll have a functional periodic table where each element is clickable, and a modal will appear with detailed information about the element. Feel free to customize it further by adding more elements, improving the styling, or modifying the behavior!	Aquí tienes un ejemplo de cómo podrías generar una tabla periódica interactiva utilizando Python (con bibliotecas como Dash o Plotly) y luego llevarla a una página web como CodePen. Aunque CodePen no puede ejecutar Python directamente, te proporcionaré el enfoque usando HTML, CSS, y JavaScript para crear la tabla interactiva. Este código incluirá los siguientes aspectos : - Tabla periódica con nombres, códigos, colores y clasificaciones. - Interacción para que al hacer clic en un elemento, se muestre más información como su clasificación, nombre Explicación del código : HTML : Estructura básica de la página web que contiene el título, la tabla periódica (usando un div con una cuadrícula), y un área de información donde se mostrará la descripción del elemento cuando se haga clic en él. CSS : Estilos para la tabla periódica. Cada elemento tiene un color de fondo, un borde, y una transición visual cuando se pasa el cursor. JavaScript : Lógica para crear dinámicamente los elementos de la tabla periódica y asignarles comportamientos interactivos. Al hacer clic en cualquier elemento, se muestra su nombre, código, clasificación y una breve explicación.

	10. 將 HTML 放入 HTML 編輯器中。 11. 將 CSS 放入 CSS 編輯器中。 12. 將 JavaScript 放入 JavaScript 編輯器中。 這樣，您就能在 CodePen 中看到互動式元素週期表，並能查看每個元素的詳細信息。		Este código puede ser copiado y pegado directamente en CodePen, y podrás verlo funcionando como una tabla periódica interactiva en la página web. Si necesitas más detalles o quieres agregar más elementos, no dudes en decirme.
相關聯性	5/5	5/5	5/5
段落鮮明度	5/5	5/5	5/5
用字遣詞	5/5	5/5	5/5
執行結果	2/5	1/5	3/5
總分	17/20	16/20	18/20

註：研究者自行整理

在本實驗裡，英文組沒有代號，所以分數較對照組低。西班牙文組都有且多一個元素，所以分數會較對照組高一點。

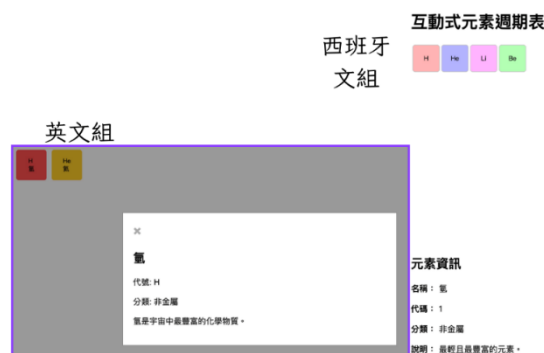


表 4-5-10 實驗組(語言)評分統整表

問題	總分		
	中文	英文	西班牙文
A1	20/20	18/20	18/20
A2	20/20	17/20	18/20
A3	12/20	12/20	12/20
B1	18/20	20/20	19/20
B2	19/20	19/20	20/20
B3	19/20	17/20	19/20
C1	20/20	19/20	15/20
C2	20/20	15/20	15/20
C3	17/20	16/20	18/20
總共	165/180	153/180	154/180

註：研究者自行整理

在本實驗裡，中文組的分數遠高於其他兩組，推測是因為英文和西班牙文組搜尋到的資料並不是我們常用的，在文化和語言上有些微的差異，所以不合我的胃口。

六、研究討論

(一) 不同時間對 ChatGPT 提問是否會獲得不同結果？

根據我的實驗結果，提問時間對 ChatGPT 的回覆並無太大影響。

(二) 不同帳號對 ChatGPT 提問是否會獲得不同結果？

不同帳號差別在於帳號中的個人資料是否完整，愈完整的個人化資料，可以讓 ChatGPT 給你期望的回覆，因為 ChatGPT 會根據你的個人化資料給你偏好或和你有相關的答案，所以帳號個人資料的完整度會影響回覆結果且結果的差距很大。

(三) 不同國家 IP 對 ChatGPT 提問是否會獲得不同結果？

獲得回覆的結果會取決於你提問的內容是否是一個國家特有的特色或法規，如果是的話可能會有些許影響。推測不同國家 IP 可能會影響找到的資料，進而影響結果，但影響不大，所以大可不必在意。

(四) 不同提問法對 ChatGPT 提問是否會獲得不同結果？

經過本研究實驗結果發現，提問時給予愈精準、愈多的關鍵詞與要求指令，會使 ChatGPT 給你的回覆更接近你期望的答案，比起簡單的指令獲得的結果會有較大的差距與影響。

(五) 不同語言對 ChatGPT 提問是否會獲得不同結果？

ChatGPT 所提供的資訊，有約 70%是用提問語言進行資料搜尋，剩下約 30%會搜尋其他語言相關資料，且英語佔的比重最大，故提問的內容會因語言造成資料搜尋時有所限制而有所不同的結果。

(六) 為什麼在語言組的實驗英文和西班牙文的分數會這麼低？

推測是因為在英文和西班牙文組的實驗結果翻譯時(請 ChatGPT 幫忙翻譯)，我們常用的文法、語詞不一樣，導致他們兩組給的答案不是我們喜歡、常用的答案，也有可能是因為文化、相關法規、規定或語言的差別，導致給的答案和我們認知的不一樣，還有另外一種可能，是因為在 C1~C3 問題時，因為 ChatGPT 給的答案大部分無法執行，所以會把總分拉低。

(七) 評分表有必要這樣子評分嗎?

評分過程中，我發現使用「比較」的方式進行評分可以更客觀。透過每個提問利用實驗組與對照組進行比較，才可以更為客觀的比較出二到三個組別哪個組別回覆更符合期望，所以最後我把結果最好的一個實驗組當成基準量，而其他的實驗組則當成比較量，依照我做的評分表評分。

(八) C1~C3 為什麼有時候程式問題會在相關聯性的地方評分?

因為有時候有些組別的功能會做得較好，且不是我要求一定要有的東西，如有這種狀況我就會將沒有額外功能的實驗組在相關連性的地方扣分。

伍、研究結論與建議

一、結論

- (一)根據我的實驗結果，提問時間對 ChatGPT 的回覆並無太大影響，所以在任何時間都可以向 ChatGPT 提問。
- (二)根據我的實驗結果，提問帳號對 ChatGPT 的回覆影響很大，所以提問時要使用有自己個人化資料的主帳號，得到的回覆會較貼近個人期望。
- (三)根據我的實驗結果，提問時使用的國家 IP 對 ChatGPT 的影響不大，所以不用特別使用 VPN 改變國家 IP 來進行提問。
- (四)根據我的實驗結果，提問的方法對 ChatGPT 回覆的影響很大，所以提問時要精準使用關鍵詞，並且給予較多的關鍵詞指令，才能獲得期望的回覆。
- (五)根據我的實驗結果，提問的語言對 ChatGPT 的回覆會因為地區、文化、用語造成影響，所以提問時要使用符合當地的用語來進行提問。
- (六)我評分的方式是將各個評分項目中表現最好的一個實驗組挑出來當作基準量，和基準量相比較差的根據較差的程度進行扣分。

二、建議

- (一) 因為我的實驗在評分的地方很多都是使用我的主觀意見，所以我比較喜歡、給予比較高分數的實驗組不一定是每個人都喜歡的，所以當作參考就好，實際上每個人使用 ChatGPT 的方法、喜歡的變因，和我實驗出來的結果可能不一定一樣。未來如果還有人想做類似的實驗的話，評分方式要用較為客觀的方式評分。
- (二) 使用 ChatGPT 時，任何時間都可以提問，而帳號建議使用自己常用的帳號，並且常常和 ChatGPT 聊天灌輸自己的個人化資料，且愈精確、愈多的關鍵詞可以讓 ChatGPT 給你你最想要的答案。如果你是要查詢某一國特有的物品、文化或法規時，建議使用那個國家的 IP，且提問語言使用該國的語言，不使用不會影響，使用了會使答案更好。

陸、參考文獻

- ChatGPT*. (2024, November 7). 維基百科，自由的百科全書. <https://zh.wikipedia.org/zh-tw/ChatGPT>
- 曾子軒. (2024, July 24). *ChatGPT 與 GPT-4o Mini 終極使用教學*. 遠見雜誌. <https://www.gvm.com.tw/article/114545#GPT4o%20mini%E6%98%AF%E4%B%80%E9%BA%BC%EF%BC%9>
- Circleghost li. (2023, March 8). *人工智慧也不完美！揭秘 ChatGPT 8 大注意事項 x 免費/付費版差異全攻略*. Awoo. <https://www.awoo.ai/zh-hant/blog/ai-chatgpt-8-notes/>
- Mr.Roo. (2024, October 17). *ChatGPT-4o 功能有哪些？OpenAI O1 模型再升級？GPT-4o 價格、詳細應用一次看！*. 袋鼠金融. <https://roo.cash/blog/chatgpt-4o-functions/>
- Chuck chiang. (2024, February 16). *ChatGPT 提問類型分析*. Medium. <http://chuckchiang.medium.com/chatgpt-%E6%8F%90%E5%95%8F%E9%A1%9E%E5%9E%8B%E5%88%86%E6%9E%90-15159da3e10d>
- 橙果杏. (2025, March 19). **【入門級】—C#介紹**. Vocus. https://vocus.cc/article/6416f174fd89780001dca7ca?fbclid=IwY2xjawJVk5J1eHRuA2F1bQIxMAABHSpaf1h2fuTYom9ssZV9mGfHjLzAwedWoH6apG8rT7i9D166MNdin0Mdvw_aem_Zv71kIS7jb8TUyglAFn7xA
- C#. (2025, May 6). 維基百科，自由的百科全書. <https://zh.wikipedia.org/zh-tw/C%E2%99%AF>
- 桃熊. (2021). *Day20 類別與物件介紹*. IT 邦幫忙. <https://ithelp.ithome.com.tw/m/articles/10277825>

柒、研究心得

在我一開始做研究的時候，我原本以為，做關於 AI 的題目不會有太多的困難，因為現在 ChatGPT 非常有名，可以參考的資料、研究應該不少，且應該不會有太大的難點，網路上應該都找得到。但事實上，並非如此。

在本實驗中，我遇到了非常多次的瓶頸、困難，其中最讓我抓狂的，就是版面配置了。因為 ChatGPT 給的回覆內容都是非常多的，所以如果要把回覆都放到報告裡的話，需要經過很多調整，而有一次 word 還跑版，讓我花了非常多時間調整。除此之外，其實幾乎沒有什麼其他的獨立研究是在做關於 ChatGPT 的，而且現在網路上很多資料，但很多都是重複的，且都沒有提到很深度的內容，所以對我的研究幫助不大。

在本次的研究中，我學到了很多事，其中我最印象深刻的一點是絕對要做好時間規劃，要盡量能做就做避免最後時間來不及而趕著做完，且在研究的途中就要好好善用筆記軟體，把該記的東西記下來讓研究更順利。