CS50 Python week 1

CS50 Python 課程的第一週主要探討了**條件語句** (if, elif, else) 和**布林表達式** (or, and, bool)。在解決問題時，我們學會了如何處理使用者輸入、進行字串操作、執行條件判斷以及數值計算。

以下是針對每個問題的關鍵概念和實作方法：

- Deep Thought (深度思考)
  - **目標**：程式會詢問使用者「生命、宇宙以及萬事萬物的終極答案是什麼？」，如果輸入是 42（數字、單字 "forty-two"、帶連字號的 "4-2" 或帶空格的 "4 2"），則不區分大小寫地輸出 "Yes"，否則輸出 "No"。
  - **核心概念**：
    - **input() 函式**：用於從使用者那裡獲取輸入，並將其儲存在變數中。例如，answer = input("What is the answer to the great question of life, the universe and everything? ")。
    - **if, elif, else 語句**：用於控制程式的執行流程。如果 if 條件為真，則執行其程式碼區塊；如果為假，則檢查 elif 條件；若所有條件都為假，則執行 else 區塊。
    - **字串比較**：input() 函式返回的是字串。因此，即使使用者輸入數字 42，程式也會將其視為字串 “42”。比較時應使用雙等號 == 來判斷相等性。
    - **lower() 方法**：用於將字串轉換為**小寫**，以實現**不區分大小寫**的比較。例如，answer.lower() 會將 "Forty-Two" 轉換為 "forty-two"。此方法只適用於字母，所以對於數字 42 不需使用。
    - **strip() 方法**：用於移除字串開頭和結尾的**空白字元**（預設為空格），確保即使使用者輸入了多餘的空格，程式也能正確識別答案。例如，answer.strip().lower()。
- File Extensions (檔案副檔名)
  - **目標**：提示使用者輸入檔案名稱，然後根據檔案的副檔名輸出其**媒體類型**。需支援 gif, jpeg, jpg, png, pdf, txt, zip 等副檔名，並且**不區分大小寫**。
  - **核心概念**：
    - **input() 函式**：獲取檔案名稱輸入。
    - **strip() 和 lower() 方法**：先移除空白字元，再將檔案名稱轉換為小寫，以便進行統一的比較。通常會將處理後

的字串儲存在一個新變數中，例如 new_file_name = file_name.lower()。
        - **if, elif, else 語句**：根據副檔名判斷並輸出對應的媒體類型。
        - **in 運算子**：用於判斷一個**子字串**是否存在於另一個字串中。例如，if ".gif" in new_file_name:。
        - **媒體類型對應**：需要根據特定的副檔名輸出對應的媒體類型字串，例如 image/gif、application/pdf、text/plain。如果沒有匹配的副檔名，則輸出預設的 application/octet-stream。值得注意的是，jpeg 和 jpg 都應該輸出為 image/jpeg。
- Home Federal Savings Bank (家園聯邦儲蓄銀行)
    - **目標**：提示使用者輸入問候語。如果問候語以 "hello" 開頭，輸出 $0；如果以 "h" 開頭（但不是 "hello"），輸出 $20；否則輸出 $100。需要忽略開頭的空白字元並**不區分大小寫**。
    - **核心概念**：
        - **input() 函式**：獲取問候語輸入。
        - **strip() 和 lower() 方法**：用於移除開頭/結尾的空白字元並將問候語轉換為小寫，以實現不區分大小寫和忽略空白字元的要求。處理後的字串可以儲存在如 new_answer 的變數中。
        - **in 運算子**：用於檢查字串中是否包含 "hello" 這個子字串。例如，if "hello" in new_answer:。
        - **字串索引 ``**：用於獲取字串的第一個字元。在 Python 中，字串索引是從 0 開始的。例如，new_answer 可以獲取問候語的第一個字元。
        - **if, elif, else 語句順序**：判斷條件的順序非常重要。應首先檢查最具體的條件（是否包含 "hello"），然後是較寬泛的條件（是否以 "h" 開頭），最後是 else 情況。
- Math Interpreter (數學解釋器)
    - **目標**：提示使用者輸入算術表達式（格式為 X Y Z，其中 X 和 Z 是整數，Y 是運算子），然後計算並以**一個小數位**輸出結果（浮點數）。
    - **核心概念**：
        - **input() 函式**：獲取表達式輸入。
        - **split() 方法**：用於將字串按**空白字元**（預設）分割成多個部分，並將其分配給多個變數。例如，x, y, z =

expression.split() 會將輸入 1 + 1 分割為 x="1", y="+", z="1"。

- **類型轉換 float()**：由於 split() 得到的 x 和 z 仍是字串，需要使用 float() 函式將它們轉換為**浮點數**，才能進行算術運算。例如，new_x = float(x) 和 new_z = float(z)。
- **if, elif 語句**：根據運算子 y 的值（+, -, *, /）執行對應的算術運算。題目中假設不會出現除以零的情況。
- **round() 函式**：用於將浮點數**四捨五入**到指定的小數位數。例如，round(result, 1) 會將結果四捨五入到一位小數。

- Meal Time (用餐時間)
  - **目標**：提示使用者輸入時間，並根據時間範圍輸出是**早餐時間**、**午餐時間**還是**晚餐時間**。
  - **核心概念**：
    - **main() 函式結構**：雖然在 Python 中不是強制性的，但使用 if __name__ == "__main__": 結構定義 main 函式是良好的程式碼組織習慣，它指定了程式的執行入口點。
    - **input() 函式**：獲取時間輸入，例如 what time is it?。
    - **輔助函式 convert()**：這是解決此問題的關鍵。它負責將使用者輸入的時間字串（例如 "7:30"）轉換為**浮點數**（例如 7.5）。
      - **split() 方法**：在 convert() 函式內部，使用 split(':') 方法將時間字串按冒號分割成小時和分鐘兩部分。例如，"7:30" 會被分割成 ["7", "30"]。
      - **分鐘轉換**：將分鐘部分（字串）轉換為浮點數，然後除以 60，將其轉換為小時的小數部分（例如 30 分鐘 -> 0.5 小時）。
      - **類型轉換 float()**：將小時和分鐘都轉換為浮點數以便計算。
      - **return 關鍵字**：convert() 函式會返回計算出的總小時數（浮點數）給呼叫它的主程式。
    - **if, elif, else 語句**：在 main 函式中，接收 convert() 函式返回的浮點數時間，然後使用 if/elif 語句和**邏輯運算子 and** 來判斷時間是否落在特定的範圍內：

- 早餐時間：7:00 到 8:00 (含)，即 time >= 7 and time <= 8。
- 午餐時間：12:00 到 13:00 (含)，即 time >= 12 and time <= 13。
- 晚餐時間：18:00 到 19:00 (含)，即 time >= 18 and time <= 19。

所有這些問題解決方案都強調了 Python 中基礎但強大的功能，例如使用者輸入、條件邏輯、字串操作和類型轉換，這些都是程式設計中的基本構件。