D1210799 王建葦 H03

**5.5.7.1 函式練習題 (一)：計算投手勝敗投**

題目：

編寫一個名為 determine\_pitcher\_result(innings\_pitched, earned\_runs, total\_runs, is\_winning\_team) 的 Python 函式。這個函式接收四個參數：

1. innings\_pitched (浮點數)：投手投球局數。
2. earned\_runs (整數)：投手自責分。
3. total\_runs (整數)：該隊總得分。
4. is\_winning\_team (布林值)：一個布林值，表示該投手所屬的球隊是否贏得比賽。

函式應該根據以下規則判斷投手的勝敗投，並回傳一個字串：

1. 如果投手投滿 5 局（含）以上，且其所屬球隊贏得比賽，則回傳 "勝投"。
2. 如果投手投滿 5 局（含）以上，且其所屬球隊輸掉比賽，則回傳 "敗投"。
3. 如果投手投球未滿 5 局，無論勝敗，都回傳 "中繼"。

設計方法概述：

**目標**

設計一個簡單的程式，根據投手的投球局數、失分、是否勝隊等資訊，判定並輸出該場比賽中投手的結果（如：勝投、敗投、中繼）。

**核心功能**

* **判定規則**：
  + **勝投**：投手投滿至少 5 局，且球隊獲勝。
  + **敗投**：投手投滿至少 5 局，但球隊落敗。
  + **中繼**：投球不足 5 局，不論球隊勝敗。
* 其他狀況則預設為空字串。

**函式設計**

* determine\_pitcher\_result(innings\_pitched, earned\_runs, total\_runs, is\_winning\_team) -> str
  + **輸入參數**：
    - innings\_pitched (float)：投球局數。
    - earned\_runs (int)：責任失分。
    - total\_runs (int)：總失分（目前未用到，但保留可擴充）。
    - is\_winning\_team (bool)：球隊是否獲勝。
  + **邏輯**：
    - 若投滿 5 局且球隊獲勝，回傳 "勝投"。
    - 若投滿 5 局但球隊落敗，回傳 "敗投"。
    - 若未投滿 5 局，回傳 "中繼"。
    - 其他狀況回傳空字串（理論上不會進入這個情況）。
* main() -> int
  + 呼叫 determine\_pitcher\_result 測試不同情境。
  + 使用 print() 將結果輸出到螢幕。
  + 最後回傳 0 作為程式正常結束的訊號。

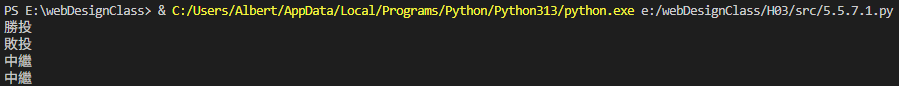
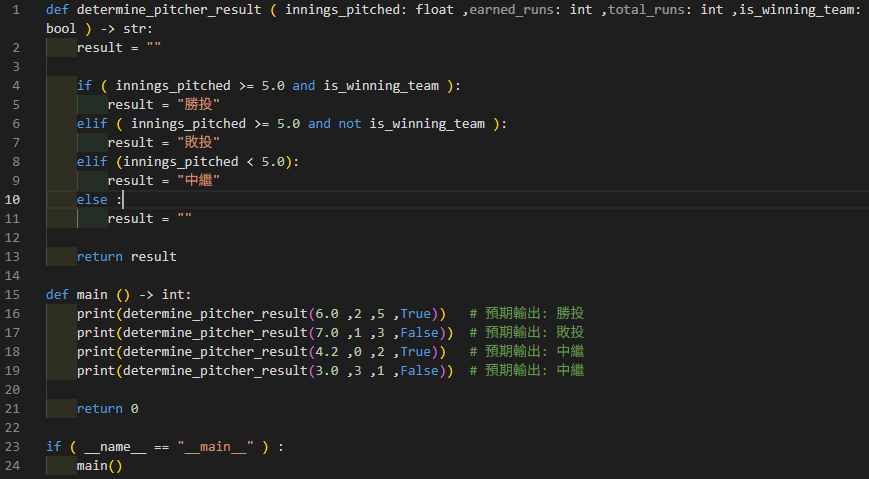
**程式流程**

1. 主程式 main() 執行。
2. 依序測試不同投手數據組合。
3. 每次呼叫 determine\_pitcher\_result 判定並輸出結果。

**擴充性**

* earned\_runs 和 total\_runs 目前未參與判定，日後可加入更細緻的判斷（如資格敗投、中繼失敗等情境）。
* 可進一步考慮「救援成功」、「救援失敗」等情況，讓系統更完整。

程式、執行畫面：



參考資料與使用工具及比例：

ChatGPT: 5% 🡪 設計方法概述內容修飾

**5.5.7.2 函式練習題 (二)：格式化球員打擊數據**

題目：

編寫一個名為 format\_batting\_stats(name, at\_bats, hits, home\_runs) 的 Python 函式。這個函式接收四個參數：

1. name (字串)：球員的姓名。
2. at\_bats (整數)：球員的打數。
3. hits (整數)：球員的安打數。
4. home\_runs (整數)：球員的全壘打數。

函式應該計算球員的打擊率（安打數除以打數，保留三位小數），並回傳一個格式化的字串，包含球員的姓名、打數、安打數、全壘打數和打擊率，格式如下：

[球員姓名] 的打擊數據：

打數: [打數]

安打: [安打]

全壘打: [全壘打]

打擊率: [打擊率]

設計方法概述：

**目標**

設計一個簡單的函式，計算打者的打擊率，並將打者的基本打擊數據（打數、安打數、全壘打數、打擊率）以格式化的字串方式輸出。

**核心功能**

* + 計算打擊率：
    - 打擊率 = 安打數 / 打數。
    - 小數點後保留三位數顯示。
  + 格式化輸出：
    - 按照指定格式，逐行輸出打數、安打數、全壘打數、打擊率。

**函式設計**

format\_batting\_stats(name, at\_bats, hits, home\_runs) -> str

* + - 輸入參數：
      * name (str)：球員姓名。
      * at\_bats (int)：打數（出場打擊的次數）。
      * hits (int)：安打數。
      * home\_runs (int)：全壘打數。
    - 邏輯：
      * 計算打擊率（hits / at\_bats）。
      * 組合成格式化的多行文字字串。
      * 回傳格式化好的字串。

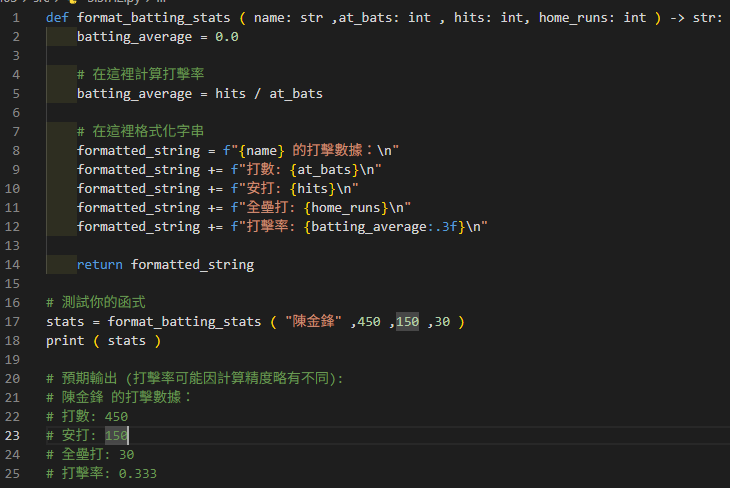
**程式流程**

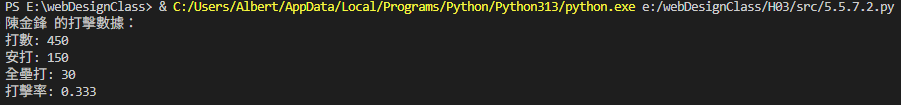
1. 呼叫 format\_batting\_stats，傳入球員的打數、安打數、全壘打數。
2. 函式內計算打擊率。
3. 將各項數據組成整齊的文字。
4. 回傳並輸出結果。

**測試案例**

* + 測試資料：
    - 姓名：陳金鋒
    - 打數：450
    - 安打：150
    - 全壘打：30
  + 預期輸出格式：
* 陳金鋒 的打擊數據：
* 打數: 450
* 安打: 150
* 全壘打: 30
* 打擊率: 0.333

程式、執行畫面：





參考資料與使用工具及比例：

ChatGPT: 5% 🡪 設計方法概述內容修飾

**5.6.8.1 棒球員**

題目：

請設計一個名為 BaseballPlayer 的類別，用於表示一位棒球員。這個類別應該包含以下屬性：

* name (字串)：球員的姓名。
* number (整數)：球員的背號。
* position (字串)：球員守備位置（例如："投手", "捕手", "一壘手" 等）。
* batting\_average (浮點數)：球員的打擊率（初始值為 0.0）。
* home\_runs (整數)：球員的全壘打數（初始值為 0）。

這個類別應該包含以下方法：

* \_\_init\_\_(self, name, number, position)：建構子，用於初始化球員的姓名、背號和守備位置。打擊率和全壘打數應初始化為預設值。
* display\_info(self)：顯示球員的所有資訊，包括姓名、背號、守備位置、打擊率和全壘打數。
* increase\_batting\_record(self, hits, at\_bats)：更新球員的打擊紀錄。這個方法接收本次打擊的安打數 (hits) 和打數 (at\_bats)，並根據這些資訊更新球員的 batting\_average。請注意，打擊率的計算方式是總安打數除以總打數。
* hit\_home\_run(self)：當球員擊出全壘打時呼叫此方法，將球員的 home\_runs 增加 1。

操作要求：

1. 創建至少三位 BaseballPlayer 物件，並賦予他們不同的姓名、背號和守備位置。
2. 針對其中一位球員，呼叫 display\_info() 方法顯示其初始資訊。
3. 讓其中一位球員進行兩次打擊紀錄更新：
   * 第一次打擊：2 安打，3 打數。
   * 第二次打擊：1 安打，4 打數。

每次更新後，都呼叫 display\_info() 方法查看該球員的最新資訊。

1. 讓其中一位球員擊出一支全壘打，並呼叫 display\_info() 方法查看其全壘打數是否已更新。

進階挑戰：

* 為 BaseballPlayer 類別添加一個 \_\_str\_\_(self) 方法，使其在使用 print() 函數列印球員物件時，能以更友好的格式顯示球員資訊。
* 創建一個球隊 (Team) 類別，該類別可以儲存多個 BaseballPlayer 物件，並提供方法來新增球員、顯示球隊所有球員資訊，以及計算球隊的平均打擊率。

設計方法概述：

**目標**

建立一套基礎的棒球球員與球隊管理程式，能夠紀錄個別球員的打擊表現、全壘打數，以及統整球隊的成績統計（如球隊總打擊率）。

**系統設計概覽**

* **球員類別（BaseballPlayer）**：
  + 記錄個別球員的基本資料與打擊成績。
  + 支援更新打擊紀錄、擊出全壘打的行為。
  + 可以顯示球員資訊，或以自訂格式列印球員摘要。
* **球隊類別（Team）**：
  + 管理一群球員成員。
  + 提供新增球員、列出所有球員、計算球隊總打擊率等功能。
* **主程式流程（main()）**：
  + 建立球員 → 建立球隊 → 加入球員 → 操作球員記錄 → 輸出結果。

**類別設計詳述**

**BaseballPlayer 類**

* **屬性**：
  + name：球員姓名
  + number：球員背號
  + position：守備位置（如：內野手 IF、外野手 OF）
  + batting\_average：打擊率（自動計算）
  + home\_runs：全壘打數
  + hits：安打數
  + at\_bats：打數
  + **方法**：
    - display\_info()：列印球員詳細資料。
    - increase\_batting\_record(hits, at\_bats)：增加安打數與打數，並更新打擊率。
    - hit\_home\_run()：全壘打數加 1。
    - \_\_str\_\_()：定義球員物件的簡潔列印格式。

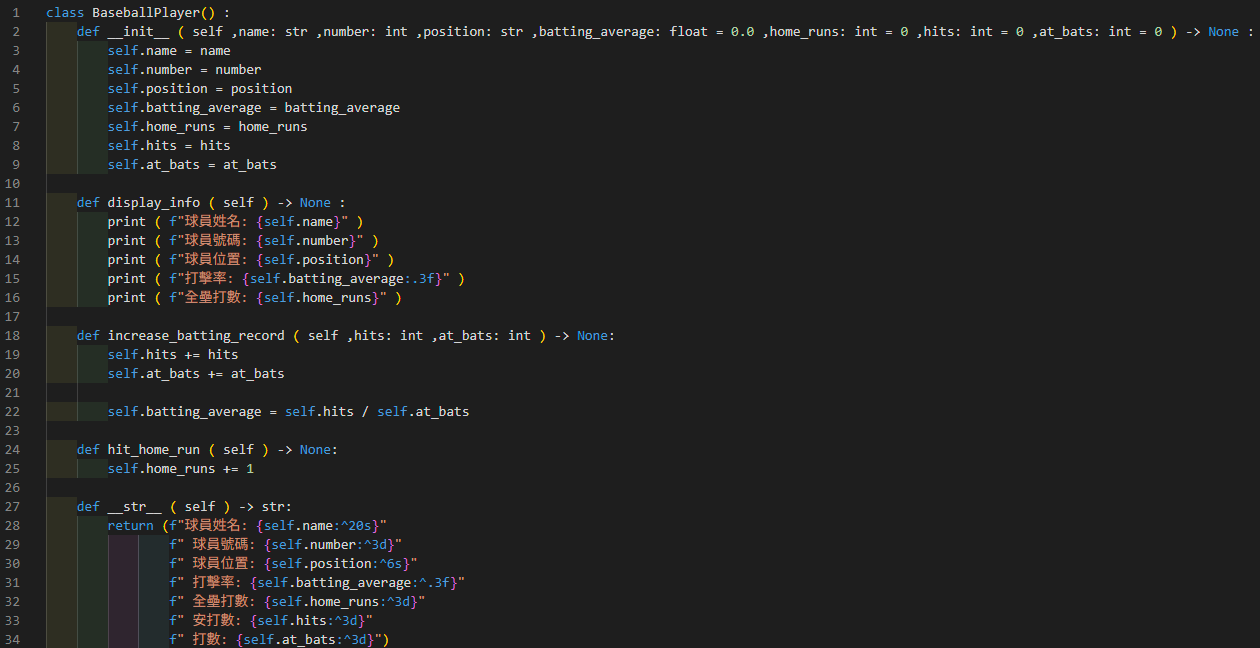
**Team 類**

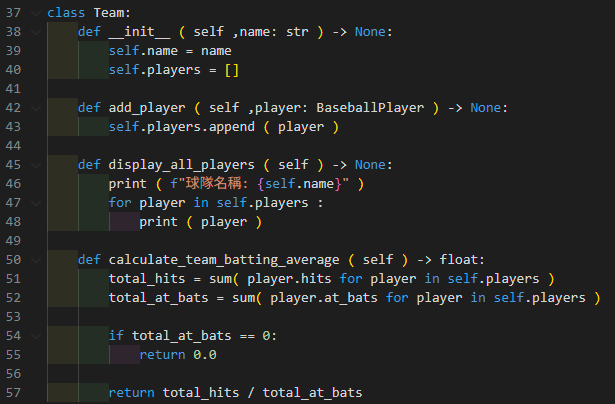
* **屬性**：
  + name：球隊名稱
  + players：球員列表
* **方法**：
  + add\_player(player)：將球員加入球隊。
  + display\_all\_players()：列印所有球員的資訊。
  + calculate\_team\_batting\_average()：計算整支球隊的平均打擊率。

**程式運行流程（main()）**

1. **建立球員物件**（player1, player2, player3）。
2. **建立球隊物件**（team）。
3. **將球員加入球隊**。
4. **顯示單一球員的初始資訊**。
5. **模擬球員打擊與全壘打表現**，更新打擊紀錄。
6. **列印全隊所有球員的資訊**。
7. **計算並顯示球隊總打擊率**。

程式、執行畫面：







參考資料與使用工具及比例：

ChatGPT: 5% 🡪 設計方法概述內容修飾

ChatGPT: 5% 🡪 加分題諮詢