進階版 FrozenLake Q-Learning 規格說明（根據你這份完整進階版程式）

1. 匯入必要的函式庫

- `gym`：建立與操作 FrozenLake 環境。
- `numpy`：進行矩陣與亂數運算。
- `tabulate`：美化表格輸出，將 Q-Table 視覺化成漂亮的文字表格。
- `matplotlib.pyplot`：動態繪製迷宮與 Q-Table 熱力圖。
- `time`：控制顯示更新的間隔，使動畫流暢。

2. 環境初始化

- 建立 `FrozenLake-v1` 環境，設 `is_slippery=False`（不滑動）。
- 取得：
  - `n_states`：格子總數（狀態數）。
  - `n_actions`：可選的行動總數。

3. Q-table 初始化

- 建立形狀為 `(n_states, n_actions)` 的全零矩陣，作為初始 Q-table。

4. 超參數設定

- 學習率 `learning_rate = 0.8`。
- 折扣因子 `gamma = 0.95`。
- 總訓練回合數 `episodes = 2000`。

5. 探索率與衰減設定

- 初始探索率 `initial_epsilon = 0.3`。
- 最小探索率 `min_epsilon = 0.01`。
- 每回合探索率衰減係數 `epsilon_decay = 0.995`。
- 設定變動中的 `epsilon = initial_epsilon`。

6. 定義輔助繪圖函式：`plot_maze`

- 顯示當前迷宮狀態：
  - `H`（洞穴）為黑色。
  - `G`（終點）為金色。
  - `S`（起點）為淺藍色。
  - Agent 位置標為紅色。
- 每個格子中顯示該狀態下各行動的 Q-value。
- 使用 `plt.pause(0.01)` 動態更新畫面。

---

7. 訓練過程

每回合流程

- 重置環境，取得初始狀態。
- 初始化：
  - `done = False`。
  - `step = 0`。
  - `total_reward = 0`。
  - `qtable_before`：儲存更新前的 Q-table（用來比較變化量）。
  - `reached_goal = False`：標記是否成功到達終點。

回合內互動

- 直到 `done == True`：
  - 以 ε-greedy 策略選擇行動：
    - 以 `epsilon` 機率隨機探索。
    - 否則選擇目前狀態下 Q-value 最大的行動。
  - 執行行動，取得 `next_state`、`env_reward`、`done`、`info`。
  - **Reward shaping（調整回饋）**：
    - 若成功到達終點，reward = +100。
    - 若掉進洞或失敗結束，reward = -10。
    - 若撞牆（移動後座標未改變），reward = -1。
    - 否則，一般移動 reward = -0.1。
  - **Q-Learning 更新**（TD(0)）：
    - 計算 `td_target`：
      - 若撞牆，直接用當下 reward。
      - 否則用 `reward + gamma × best Q-value in next_state`。
    - 計算 `td_error = td_target - Q[state, action]`。

        - 更新 `Q[state, action] += learning_rate × td_error`。
    - 用 `plot_maze` 函式動態顯示當前迷宮與 Q-value。
    - 更新 `state = next_state`。
    - `step += 1`。

回合結束後

- 每100回合：
    - 印出當前回合數。
    - 用 `tabulate` 格式化並印出 Q-Table。
    - 繪製 Q-Table 熱力圖（橫軸為行動，縱軸為狀態）。
- 記錄並印出當回合的「探索學習心得」：
    - 若成功到達終點，顯示激勵訊息與 Q-table 總變化量。
    - 若沒到達但有學習（Q-table有變化），顯示努力進步的訊息。
    - 若 Q-table 沒有變化，建議多做探索。
- 更新探索率 `epsilon`：
    - `epsilon = max(min_epsilon, epsilon × epsilon_decay)`。

---

8. 訓練完成後

- 印出最終的 Q-table（使用 `tabulate`）。
- 繪製最終 Q-Table 熱力圖（`viridis` 顏色地圖）。

---

9. 測試訓練結果

- 重置環境，取得初始狀態。
- 進入動態測試：
    - 每步選擇 Q-table 中對當前狀態最佳的行動（不再探索）。
    - 執行行動後，更新迷宮畫面（agent移動動畫）。
    - 控制每次更新間隔（使用 `time.sleep(0.05)`）。
- 測試結束後關閉動態模式並完成展示。