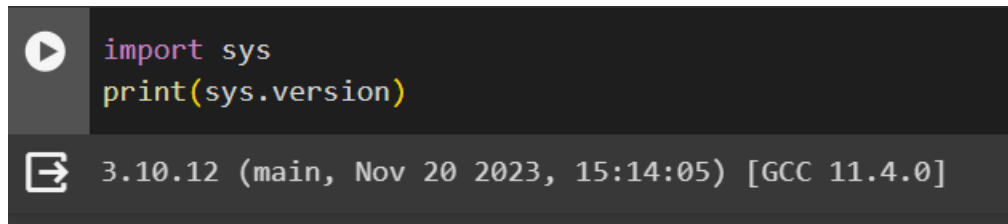


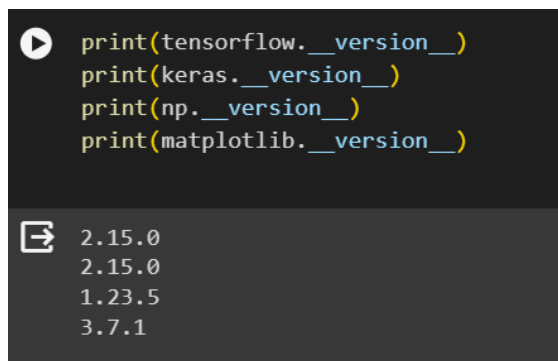
**Language : Python3**



```
import sys
print(sys.version)
```

3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0]

**List of Libraries used** : Numpy ,Tensorflow , keras.



```
print(tensorflow.__version__)
print(keras.__version__)
print(np.__version__)
print(matplotlib.__version__)
```

2.15.0  
2.15.0  
1.23.5  
3.7.1

**RL FRAMEWORK :**

### **DEEP Q NETWORK**

Learning rate = 0. 00025

Dense( 128, activation='relu', input\_shape=(9,)

Dense(128, activation='relu')

Dense(9, activation='linear')

loss=mean\_squared\_error

optimizer = RMS propogation

A similar style for target and train networks .

No of episodes – 10,000

Epsilon – 0.4

Discount Factor – 0.7

Minimum size of experience replay buffer – 4096

Maximum size of experience replay buffer – 1048576

**Evaluation on random player mode. 1000 times**

Wins : 546 Draws : 181 Loss : 273

## DETAILLS ABOUT ENVIRONMENT CLASS :

I have created the tic tac toe environment.

### 1.Initialization:

- The **Tictactoe\_v0** class is initialized with an empty game board, winning positions, and the current player's turn.
- The winning positions are defined for rows, columns, and diagonals.

### 2. Reset Method:

- The **reset** method resets the game board, player turn, and player marks based on whether the human or the agent should make the first move.
- If the agent starts, it calls the **env\_act** method to simulate the agent's move.

### 3. Winning Check:

- The **check\_win** method examines the board for winning positions or a tie.
- It returns:
  - **1** if the player wins.
  - **-1** if the opponent wins.
  - **0** in case of a tie.
  - **True** if the game is done, and **False** otherwise.

### 4. Environment Action:

- The **env\_act** method simulates the environment taking an action.
- It selects a random available move unless there is an opportunity for the player or opponent to win, in which case it makes a strategic move.
- It updates the board, checks for a win or tie, and switches the current player's turn.

### 5. Step Method:

- The **step** method is designed for the interaction between the agent and the environment during the training loop.
- It takes an action as input, updates the board, checks for a win or tie, and switches the current player's turn.
- It returns the current state of the board, the reward based on the game outcome, whether the game is done.