def main():

# The main function

    introduction = intro()

    board = create\_grid()

    pretty = printPretty(board)

    symbol\_1, symbol\_2 = sym()

    full = isFull(board, symbol\_1, symbol\_2) # The function that starts the game is also in here.

def intro():

# This function introduces the rules of the game Tic Tac Toe

    print("Hello! Welcome to Pam's Tic Tac Toe game!")

    print("\n")

    print("Rules: Player 1 and player 2, represented by X and O, take turns "

          "marking the spaces in a 3\*3 grid. The player who succeeds in placing "

          "three of their marks in a horizontal, vertical, or diagonal row wins.")

    print("\n")

    input("Press enter to continue.")

    print("\n")

def create\_grid():

# This function creates a blank playboard

    print("Here is the playboard: ")

    board = [[" ", " ", " "],

             [" ", " ", " "],

             [" ", " ", " "]]

    return board

def sym():

# This function decides the players' symbols

    symbol\_1 = input("Player 1, do you want to be X or O? ")

    if symbol\_1 == "X":

        symbol\_2 = "O"

        print("Player 2, you are O. ")

    else:

        symbol\_2 = "X"

        print("Player 2, you are X. ")

    input("Press enter to continue.")

    print("\n")

    return (symbol\_1, symbol\_2)

def startGamming(board, symbol\_1, symbol\_2, count):

# This function starts the game.

    # Decides the turn

    if count % 2 == 0:

        player = symbol\_1

    elif count % 2 == 1:

        player = symbol\_2

    print("Player "+ player + ", it is your turn. ")

    row = int(input("Pick a row:"

                    "[upper row: enter 0, middle row: enter 1, bottom row: enter 2]:"))

    column = int(input("Pick a column:"

                       "[left column: enter 0, middle column: enter 1, right column enter 2]"))

    # Check if players' selection is out of range

    while (row > 2 or row < 0) or (column > 2 or column < 0):

        outOfBoard(row, column)

        row = int(input("Pick a row[upper row:"

                        "[enter 0, middle row: enter 1, bottom row: enter 2]:"))

        column = int(input("Pick a column:"

                           "[left column: enter 0, middle column: enter 1, right column enter 2]"))

        # Check if the square is already filled

    while (board[row][column] == symbol\_1)or (board[row][column] == symbol\_2):

        filled = illegal(board, symbol\_1, symbol\_2, row, column)

        row = int(input("Pick a row[upper row:"

                        "[enter 0, middle row: enter 1, bottom row: enter 2]:"))

        column = int(input("Pick a column:"

                            "[left column: enter 0, middle column: enter 1, right column enter 2]"))

    # Locates player's symbol on the board

    if player == symbol\_1:

        board[row][column] = symbol\_1

    else:

        board[row][column] = symbol\_2

    return (board)

def isFull(board, symbol\_1, symbol\_2):

    count = 1

    winner = True

# This function check if the board is full

    while count < 10 and winner == True:

        gaming = startGamming(board, symbol\_1, symbol\_2, count)

        pretty = printPretty(board)

        if count == 9:

            print("The board is full. Game over.")

            if winner == True:

                print("There is a tie. ")

        # Check if here is a winner

        winner = isWinner(board, symbol\_1, symbol\_2, count)

        count += 1

    if winner == False:

        print("Game over.")

    # This is function gives a report

    report(count, winner, symbol\_1, symbol\_2)

def outOfBoard(row, column):

# This function tells the players that their selection is out of range

    print("Out of boarder. Pick another one. ")

def printPretty(board):

# This function prints the board nice!

    rows = len(board)

    cols = len(board)

    print("---+---+---")

    for r in range(rows):

        print(board[r][0], " |", board[r][1], "|", board[r][2])

        print("---+---+---")

    return board

def isWinner(board, symbol\_1, symbol\_2, count):

# This function checks if any winner is winning

    winner = True

    # Check the rows

    for row in range (0, 3):

        if (board[row][0] == board[row][1] == board[row][2] == symbol\_1):

            winner = False

            print("Player " + symbol\_1 + ", you won!")

        elif (board[row][0] == board[row][1] == board[row][2] == symbol\_2):

            winner = False

            print("Player " + symbol\_2 + ", you won!")

    # Check the columns

    for col in range (0, 3):

        if (board[0][col] == board[1][col] == board[2][col] == symbol\_1):

            winner = False

            print("Player " + symbol\_1 + ", you won!")

        elif (board[0][col] == board[1][col] == board[2][col] == symbol\_2):

            winner = False

            print("Player " + symbol\_2 + ", you won!")

    # Check the diagnoals

    if board[0][0] == board[1][1] == board[2][2] == symbol\_1:

        winner = False

        print("Player " + symbol\_1 + ", you won!")

    elif board[0][0] == board[1][1] == board[2][2] == symbol\_2:

        winner = False

        print("Player " + symbol\_2 + ", you won!")

    elif board[0][2] == board[1][1] == board[2][0] == symbol\_1:

        winner = False

        print("Player " + symbol\_1 + ", you won!")

    elif board[0][2] == board[1][1] == board[2][0] == symbol\_2:

        winner = False

        print("Player " + symbol\_2 + ", you won!")

    return winner

def illegal(board, symbol\_1, symbol\_2, row, column):

    print("The square you picked is already filled. Pick another one.")

def report(count, winner, symbol\_1, symbol\_2):

    print("\n")

    input("Press enter to see the game summary. ")

    if (winner == False) and (count % 2 == 1 ):

        print("Winner : Player " + symbol\_1 + ".")

    elif (winner == False) and (count % 2 == 0 ):

        print("Winner : Player " + symbol\_2 + ".")

    else:

        print("There is a tie. ")

# Call Main

main()

Output:

Hello! Welcome to Pam's Tic Tac Toe game!

Rules: Player 1 and player 2, represented by X and O, take turns marking the spaces in a 3\*3 grid. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row wins.

Press enter to continue.

Here is the playboard:

---+---+---

   |   |

---+---+---

   |   |

---+---+---

   |   |

---+---+---

Player 1, do you want to be X or O? 0

Player 2, you are X.

Press enter to continue.0

Player X, it is your turn.

Pick a row:[upper row: enter 0, middle row: enter 1, bottom row: enter 2]: