Notes

Tic Tac Toe Game

Introduction
This is a Tic Tac Toe game project created using the REPL.it platform within Python language. The board is set up as the empty string list then calls upon the respective functions (shown in the code) to start the game.

Game Logic
After the program has decided which player goes first, the game progresses as follows:

Show the board
A player chooses their position
Place the marker on the position based on user input
Check whether a player has won by either 3-in-a-row
Check to see if the board is full and call a draw
How to Play
Clone or download the project files.
Open the terminal and navigate to the project directory.
Run the following command: python tic_tac_toe.py
Follow the instructions in the terminal to play the game.

References
Tic Tac Toe Game

Credits
This script was created by Anthony Constant (AC). If you have any questions or suggestions, you can contact him at anthonyconstant.co.uk/

License

Tic Tac Toe Project	
REPL.IT Share Link	https://replit.com/@Ant94x/tic-tac-toe

Tic Tac Toe Project	
GitHub Share Link	https://github.com/Anthony-Constant/Tic-Tac-Toe

```
import random

def display_board(board):


    print('   |   |')
    print(' ' + board[1] + ' | ' + board[2] + ' | ' + board[3]) # Use index position 1,2,3
    print('   |   |')
    print('------------')
    print('   |   |')
    print(' ' + board[4] + ' | ' + board[5] + ' | ' + board[6]) # Use index position 4,5,6
    print('   |   |')
    print('------------')
    print('   |   |')
    print(' ' + board[7] + ' | ' + board[8] + ' | ' + board[9]) # Use index position 7,8,9
    print('   |   |')


def player_input(): # Take the player input and assign their marker as either 'X' or 'O'

    marker = '' # Create an empty string

    while not (marker == 'X' or marker == 'O'): # While the marker is not equal to either X or O do this:
        marker = input('Player 1: Do you want to be X or O? ').upper()

    if marker == 'X': # If player1 chooses X then return X to player1 and O to Player2
        return ('X', 'O')
    else:
        return ('O', 'X') # If player1 chooses O then return O to player1 and X to player2

def place_marker(board, marker, position):# Takes in the board list object, a marker ('X' or 'O' and a desired position (numbers 1-9)
and assigns it to the board)
    board[position] = marker

def win_check(board,mark):

    return ((board[7] == mark and board[8] == mark and board[9] == mark) or # across the top
    (board[4] == mark and board[5] == mark and board[6] == mark) or # across the middle
```

```
    (board[1] == mark and board[2] == mark and board[3] == mark) or # across the bottom
    (board[7] == mark and board[4] == mark and board[1] == mark) or # down the middle
    (board[8] == mark and board[5] == mark and board[2] == mark) or # down the middle
    (board[9] == mark and board[6] == mark and board[3] == mark) or # down the right side
    (board[7] == mark and board[5] == mark and board[3] == mark) or # diagonal
    (board[9] == mark and board[5] == mark and board[1] == mark)) # diagonal


def choose_first(): # Choose_first function to determine which player goes first using the random module.
    if random.randint(0, 1) == 0: # Choose a random integer between 0 or 1
        return 'Player 2'
    else:
        return 'Player 1'

def space_check(board, position): # Space_check function to check if the board is full and returns a boolean value. True if full, False
otherwise.

    return board[position] == ' '

def full_board_check(board): # full_board_check function to check whether the board is full and returns a boolean value. True if full,
False otherwise.
    for i in range(1,10): # Run a for loop for the 9 spaces on the board starting from 1 and not including 10.
        if space_check(board, i): # If space is available it means board is false otherwise True.
            return False
            # BOARD IS FULL IF WE RETURN TRUE
    return True


def player_choice(board):  # player_choice function to ask for the player's next position (as numer 1-9) and then uses the space_check
function to check if it's a free position.
    position = 0 # Position starts here at 0
    # Check to see if the user input number is valid on the board or check if space is still available.
    while position not in [1,2,3,4,5,6,7,8,9] or not space_check(board, position):
        position = int(input('Choose your next position: (1-9) '))

    return position

# Replay function to check whether the player wants to play again
```

```
def replay():

    return input('Do you want to play again? Enter Yes or No: ').lower().startswith('y')

print('Welcome to Tic Tac Toe!')

while True:

   # SET EVERTHING UP (BOARD, WHO GOES FIRST, CHOOSE MARKERS X,O)
    # Reset the board
    theBoard = [' '] * 10 # Set up the board as empty string list
    player1_marker, player2_marker = player_input() # Call the player_input function
    turn = choose_first() # Call the choose_first function
    print(turn + ' will go first.') # Return which player goes first as string; Concatenation;
   # check if player is ready to play using play_game local variable.
    play_game = input('Are you ready to play? Enter Yes or No.')

    if play_game.lower()[0] == 'y':
        game_on = True
    else:
        game_on = False

    ## GAME PLAY

    while game_on:
        if turn == 'Player 1':
            # Player1's turn.
             # show the board
            display_board(theBoard)
            # choose the position
            position = player_choice(theBoard)
            # place the marker on the position
            place_marker(theBoard, player1_marker, position)
            # check if they won
            if win_check(theBoard, player1_marker):
                display_board(theBoard)
                  # break out of while loop end game.
                print('Congratulations! You have won the game!')
```

```
            game_on = False
          # if player 1 has not won the game, execute this block of code
        else:
          # checks if the board is full
            if full_board_check(theBoard):
                display_board(theBoard)
                print('The game is a draw!')
                break
            else:
                turn = 'Player 2'

    else:
        # Player2's turn.

        display_board(theBoard)
        position = player_choice(theBoard)
        place_marker(theBoard, player2_marker, position)

        if win_check(theBoard, player2_marker):
            display_board(theBoard)
            print('Player 2 has won!')
            game_on = False
        else:
            if full_board_check(theBoard):
                display_board(theBoard)
                print('The game is a draw!')
                break
            else:
                turn = 'Player 1'

if not replay():
    break
```

PYTHON COPY & PASTER | COPY SOURCE CODE