

```
In [14]: from IPython.display import clear_output

def display_board(board):

    clear_output()
    print(board[7] + '|' + board[8] + '|' + board[9])
    print(board[4] + '|' + board[5] + '|' + board[6])
    print(board[1] + '|' + board[2] + '|' + board[3])
```

```
In [15]: test_board = ['#', 'X', 'O', 'X', 'O', 'X', 'O', 'X', 'O', 'X']
display_board(test_board)
display_board(test_board)
```

```
X|O|X
O|X|O
X|O|X
```

Function that can take in a player input and assign their marker as "X" or "O". Using while loops to continually ask until we get a correct answer .

```
In [17]: def player_input():

    marker = ' '
    while marker != 'X' and marker != 'O':
        marker= input("Player1 : Choose X or O:").upper()

    if marker== 'X':
        return('X','O')
    else:
        return('O','X')

player1_marker, player2_marker = player_input()

Player1 : Choose X or O:O
```

A function that takes in the board list object, a marker ("X" or "O"), and a desired position(number 1-9) and assigns it to the board

```
In [18]: def place_marker(board, marker, position):

    board[position]= marker
```

```
In [19]: test_board

Out[19]: ['#', 'X', 'O', 'X', 'O', 'X', 'O', 'X', 'O', 'X']
```

```
In [21]: place_marker(test_board, '$', 8)
display_board(test_board)

X|$|X
O|X|O
X|O|X
```

A function that takes in a board and a mark(X or O) and then checks to see if that mark has won

```
In [22]: def win_check(board, mark):

    #WIN TIC TAC TOE?
    #ALL ROWS and check to see if all share the same marker
    #ALL COLUMNS and check to see if marker matches
    #2 diagonals and check to see match

    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

Run the win_check function against test_board-It should return true
```

```
In [23]: display_board(test_board)
win_check(test_board, 'X')

X|$|X
O|X|O
X|O|X

Out[23]: True
```

A function that uses the random module to randomly decide which player goes first. Use random.randint() and return a string of which player went first

```
In [24]: import random

def choose_first():
    if random.randint(0, 1) == 0:
        return 'Player 2'
    else:
        return 'Player 1'
```

A function that returns a boolean indicating whether a space on the board is freely available

```
In [25]: def space_check(board, position):

    return board[position] == ' '
```

Write a function that checks if the board is full and returns a boolean value. True if full, False otherwise.

```
In [26]: def full_board_check(board):
    for i in range(1,10):
        if space_check(board, i):
            return False
    return True
```

A function that asks for a player's next position (as a number 1-9) and then uses the function from step 6 to check if its a free position. If it is, then return the position for later use.

```
In [27]: def player_choice(board):
    position = 0

    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
```

A function that asks the player if they want to play again and returns a boolean True if they do want to play again.

```
In [29]: def replay():

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

Use while loops and the functions you've made to run the game

```
In [ ]: print('Welcome to Tic Tac Toe!')

while True:
    # Reset the board
    theBoard = [' '] * 10
    player1_marker, player2_marker = player_input()
    turn = choose_first()
    print(turn + ' will go first.')

    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

    while game_on:
        if turn == 'Player 1':
            # Player1's turn.

            display_board(theBoard)
            position = player_choice(theBoard)
            place_marker(theBoard, player1_marker, position)

            if win_check(theBoard, player1_marker):
                display_board(theBoard)
                print('Congratulations! You have won the game!')
                game_on = False
            else:
                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

| |
X| |X
O|O|O
Player 2 has won!
```

