In [1]:

```
from IPython.display import clear_output

def display_board(board):
    clear_output()  # Remember, this only works in jupyter!

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

In [2]:

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

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

In [3]:

```
def player_input():
    marker = ''

    while not (marker == 'X' or marker == 'O'):
        marker = input('Player 1: Do you want to be X or O? ').upper()

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

In [4]:

```
player_input()
```

```
Player 1: Do you want to be X or O? x
```

Out[4]:

```
('X', 'O')
```

In [5]:

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

In [6]:

```python
place_marker(test_board,'$',8)
display_board(test_board)
```

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

In [7]:

```python
def win_check(board,mark):
    
    return ((board[7] == mark and board[8] == mark and board[9] == mark) or
    (board[4] == mark and board[5] == mark and board[6] == mark) or # across
    (board[1] == mark and board[2] == mark and board[3] == mark) or # across
    (board[7] == mark and board[4] == mark and board[1] == mark) or # down t
    (board[8] == mark and board[5] == mark and board[2] == mark) or # down t
    (board[9] ==win_check(test_board,'X') mark and board[6] == mark and boar
    (board[7] == mark and board[5] == mark and board[3] == mark) or # diagon
    (board[9] == mark and board[5] == mark and board[1] == mark)) # diagonal
```

In [8]:

```python
win_check(test_board,'X')
```

Out[8]:

```
True
```

In [9]:

```python
import random

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

In [10]:

```python
def space_check(board, position):
    
    return board[position] == ' '
```

In [11]:

```python
def full_board_check(board):
    for i in range(1,10):
        if space_check(board, i):
            return False
    return True
```

In [12]:

```python
def player_choice(board):
    position = 0

    while position not in [1,2,3,4,5,6,7,8,9] or not space_check(board, posi
        position = int(input('Choose your next position: (1-9) '))

    return position
```

In [13]:

```python
def replay():

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

In [14]:

```
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
   |   |
-----------
   |   |
 X | O | X
   |   |
Player 2 has won!
Do you want to play again? Enter Yes or No: 7
```

In [ ]:

```

```