This one is a Milestone Project from the Python Bootcamp course on Udemy. Here I made a tic-tac-toe game in Python.

In [1]:
```
# Defining what is displayed to users
def display(board):
    for i in range(6,8):
        print(board[i], end='')
        print('|', end='')
    print(board[8])
    for i in range(3,5):
        print(board[i], end='')
        print('|', end='')
    print(board[5])
    for i in range(2):
        print(board[i], end='')
        print('|', end='')
    print(board[2])
```

In [2]:
```
# Defining a prompt for players to choose sides
def side():
    accept = ['x','o']
    choice = 'a'
    while choice not in accept:
        choice = input('Player 1 chooses side (x or o) ')
    if choice == 'x':
        print('Player 1 will go first')
    else:
        print('Player 2 will go first')
    return choice
```

In [3]:
```
# Asking if players are ready to go for another game
def ready():
    accept = ['yes','no']
    choice = 'a'
    while choice not in accept:
        try:
            choice = input('Are you ready to play another one? yes/no ')
            if choice not in accept:
                print('Please enter a valid answer')
        except:
            print('Please enter a valid answer')
    if choice == 'yes':
        return True
    return False
```

In [10]:
```
# Prompt for players to choose where to put a mark
def position(board):
    pos = 'a'
    accept = range(1,10)
    while pos not in accept:
        try:
            pos = int(input('Choose position from 1 to 9 '))
        except:
```

```
            print('Please enter a number')
        if pos in accept:
            if board[pos-1] not in 'xo':
                return pos
            else:
                print('The position has already been used')
                pos = 'a'
        else:
            print('Enter a suitable position')
```

In [5]:
```
# Function that checks whether player 1 chose the side 'x'
def p1_x():
    choice = side()
    if choice == 'x':
        return True
    else:
        return False
```

In [6]:
```
# Function that puts a mark on the board
def put_pos(board, pos, t1):
    if t1: #t1 here refers to turn 1
        board[pos-1] = 'x'
    else:
        board[pos-1] = 'o'
    return board
```

In [7]:
```
# Function to check if the game is still on
# Checks various winning conditions
def game_on(board):
    for i in range(len(board)):
        if i == 0 or i == 3 or i ==6:
            if board[i]==board[i+1]==board[i+2]:
                return False
        if i == 0 or i == 1 or i == 2:
            if board[i]==board[i+3]==board[i+6]:
                return False
        if i == 0 and board[i]==board[i+4]==board[i+8]:
            return False
        if i == 2 and board[i]==board[i+2]==board[i+4]:
            return False
    else:
        return True
```

In [8]:
```
# Function to check if all the positions on the board have been used and there is a
def is_draw(board):
    a = []
    for item in board:
        if not item.isdigit():
            a.append(item)
    if len(a) == 9:
        return True
    return False
```

In [11]:
```
# The game logic here
from IPython.display import clear_output
```

```
while True:
    clear_output()
    test_board = ['1','2','3','4','5','6','7','8','9']
    t1 = True # Turn 1 is set true
    p1 = p1_x() # Player 1 makes a choice
    while game_on(test_board) and not is_draw(test_board):
        clear_output()
        display(test_board)
        if p1:
            print('Player 1 turn')
        else:
            print('Player 2 turn')
        pos = position(test_board)
        test_board = put_pos(test_board, pos, t1)
        if not game_on(test_board):
            clear_output()
            display(test_board)
            if p1:
                print('Congrats! Player 1 won')
            else:
                print('Congrats! Player 2 won')
        elif is_draw(test_board):
            clear_output()
            display(test_board)
            print('Draw')
        else: # The game is on and we have to change turns
            p1 = not p1
            t1 = not t1
    if not ready():
        break
```

```
o|o|x
x|x|o
o|x|x
Draw
Are you ready to play another one? yes/no no
```