

Tic-Tac-Toe

Board positions format is similar to numpad

7|8|9

4|5|6

1|2|3

```
In [33]: #Printing the board
from IPython.display import clear_output
def display_board(board):
    """
    Function to display board
    """
    clear_output()
    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 [34]: #Player input
#Choosing the input X or O
def player_input():
    """
    Function to chose player input
    """
    marker=''
    #Keep asking for an input till X or O is entered
    while not (marker == 'X' or marker == 'O'):
        marker=input('Player 1 choose X or O: ')

    # Assign the opposite value to player 2
    if marker=='X':
        return('X','O')
    else:
        return('O','X')

In [35]: def place_marker(board,marker,position):
board[position]=marker

In [36]: def win_check(board,mark):
    """
    Function defining win conditions
    """
    #Win conditions
    # all rows have same marker
    return ((board[1]==board[2]==board[3]==mark) or #across the bottom
(board[4]==board[5]==board[6]==mark) or #accross the middle
(board[7]==board[8]==board[9]==mark) or #across the top
    #all coloumns have same marker
    (board[1]==board[4]==board[7]==mark) or #down the left
    (board[2]==board[5]==board[8]==mark) or #down the middle
    (board[3]==board[6]==board[9]==mark) or #down the right
    #all coloumns have same marker
    (board[1]==board[5]==board[9]==mark) or #Left Diagonal
    (board[3]==board[5]==board[7]==mark)) #right diagonal
    #all coloumns have same marker
    #both diagonals have same marker

In [37]: import random
def choose_first():
    """
    Randomizing player turns
    """
    flip= random.randint(0,1)
    if flip==0:
        return 'Player 1'
    else:
        return 'Player 2'

In [38]: def space_check(board,position):
    """
    Function to check if position on board is empty
    """
    return board[position]!=' '

In [39]: def full_board(board):
    """
    Funtion to check if the board is full
    """
    for i in range(1,10):
        if space_check(board,i):
            return False
    #Board is fll if we return True
    return True

In [40]: def player_choice(board):
    """
    Function to take player's chosen position
    """
    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 a position (1-9): '))
    return position

In [41]: def replay():
    """
    Function to ask if the user wants to play again
    """
    choice= input('Want to play again Yes or No: ')
    return choice=='Yes'

In [43]: #While loop tp keep running the game
print("Welcome to Tic-Tac-Toe")

while True:
    #play the game
    #SET THE BOARD(WHO IS FIRST, SET THE MARKERS)
    game_board=[" "]*10
    player1_marker,player2_marker= player_input()

    turn= choose_first()
    print(turn+ "will go first")
    play_game= input('Ready to play? y or n?: ')
    if play_game=='y':
        game_on=True
    else:
        game_on=False
    #Gameplay
    while game_on:
        if turn=='Player 1':
            #show the board
            display_board(game_board)
            #choose a position
            position=player_choice(game_board)
            #place marker on the position
            place_marker(game_board,player1_marker,position)
            #check if they won
            if win_check(game_board,player1_marker):
                display_board(game_board)
                print('Player 1 has won!!')
                game_on=False
            else:
                if full_board(game_board):
                    display_board(game_board)
                    print("Game Tied")
                    break
                else:
                    turn='Player 2'
            #check if there is a tie

            #no tie and no win? The next players turn

        else:
            #show the board
            display_board(game_board)
            #choose a position
            position=player_choice(game_board)
            #place marker on the position
            place_marker(game_board,player2_marker,position)
            #check if they won
            if win_check(game_board,player2_marker):
                display_board(game_board)
                print('Player 2 has won!!')
                game_on=False
            else:
                if full_board(game_board):
                    display_board(game_board)
                    print("Game Tied")
                    break
                else:
                    turn='Player 1'

    if not replay():
        break
#Break out of the while loop on replay()

    | | |
    x | o | x
    | | |
    -----
    | | |
    x | o | o
    | | |
    -----
    | | |
    o | x | o
    | | |
    Game Tied
    Want to play again Yes or No: No

In [ ]: 
```