**Write a function that can print out a board. Set up your board as a list, where each index 1-9 corresponds with a number on a number pad, so you get a 3 by 3 board representation.**

**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(' | |')

2- Run your function on a test version of the board list, and make adjustments as necessary

test\_board **=** ['#','X','O','X','O','X','O','X','O','X']

display\_board(test\_board)

3-Write a function that can take in a player input and assign their marker as 'X' or 'O'. Think about using while loops to continually ask until you get a correct answer.

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')

player\_input()

def place\_marker(board, marker, position):

board[position] = marker

place\_marker(test\_board,'$',8)

display\_board(test\_board)

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

(board[1] == mark and board[2] == mark and board[3] == mark) or

(board[7] == mark and board[4] == mark and board[1] == mark) or

(board[8] == mark and board[5] == mark and board[2] == mark) or

(board[9] == mark and board[6] == mark and board[3] == mark) or

(board[7] == mark and board[5] == mark and board[3] == mark) or

(board[9] == mark and board[5] == mark and board[1] == mark))

win\_check(test\_board,'X')

import random

def choose\_first():

if random.randint(0, 1) == 0:

return 'Player 2'

else:

return 'Player 1'

def space\_check(board, position):

return board[position] == ' '

def full\_board\_check(board):

for i in range(1,10):

if space\_check(board, i):

return False

return True

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

def replay():

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

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

(this code only work in jupitor notebook)