#Board

#Display Board

#Play game

#Handle turn [x or o]

#Check win

#Check Rows

#Check Columns

#Check Diagonals

#Check Tie

#Flip player

#Global variable

#game board

#Global variable

#game board

board=["-", "-", "-",

"-", "-", "-",

"-", "-", "-" ]

#if game is still going

game\_still\_going=True

#who won or tie

winner=None

#whose turn is it

current\_player="x"

def display\_board():

print(board[0]+"|"+ board[1]+"|"+ board[2])

print(board[3]+"|"+ board[4]+"|"+ board[5])

print(board[6]+"|"+ board[7]+"|"+ board[8])

def play\_game():

display\_board() #display the initial board

while game\_still\_going:

#handle a single player of an arbitary playerl

handle\_turn(current\_player)

check\_if\_game\_over()

#Flip to the other player

flip\_player()

#the Game has ended

if winner=="x"or winner=="o":

print(winner + "won")

elif winner==None:

print("tie.")

def handle\_turn(player):

print(player +"'s turn")

position=input("chose b/w 1-9:")

valid=False

while not valid:

while position not in ["1","2","3","4","5","6","7","8","9"]:

position=input("invalid Input->choose b/w 1-9")

position=int(position)-1

if board[position]=="-":

valid=True

else:

print("you can't go there. Enter again")

board[position]=player ##???????

display\_board()

def check\_if\_game\_over():

check\_for\_win()

check\_if\_tie()

def check\_for\_win():

global winner

#check rows

row\_winner=check\_rows()

#check compile

column\_winner=check\_columns()

#check diagonals

diagonal\_winner=check\_diagonals()

if row\_winner:

#there was a win

winner=row\_winner

elif column\_winner:

#there was a win

winner=column\_winner

elif diagonal\_winner:

winner=diagonal\_winner

else:

winner=None

return

def check\_rows():

global game\_still\_going

row\_1=board[0]==board[1]==board[2] !="-"

row\_2=board[3]==board[4]==board[5] !="-"

row\_3=board[6]==board[7]==board[8] !="-"

if row\_1 or row\_2 or row\_3:

game\_still\_going=False

if row\_1:

return board[0]

elif row\_1:

return board[3]

elif row\_3:

return board[6]

return

def check\_columns():

global game\_still\_going

column\_1=board[0]==board[3]==board[6] !="-"

column\_2=board[1]==board[4]==board[7] !="-"

column\_3=board[2]==board[5]==board[8] !="-"

if column\_1 or column\_2 or column\_3:

game\_still\_going=False

if column\_1:

return board[0]

elif column\_2:

return board[1]

elif column\_3:

return board[2]

return

def check\_diagonals():

global game\_still\_going

diagonals\_1=board[0]==board[4]==board[8] !="-"

diagonals\_2=board[2]==board[4]==board[6] !="-"

if diagonals\_1 or diagonals\_2:

game\_still\_going=False

if diagonals\_1:

return board[0]

elif diagonals\_2:

return board[6]

return

def check\_if\_tie():

global game\_still\_going

if "-" not in board:

game\_still\_going=False

return

def flip\_player():

global current\_player

if current\_player=="x":

current\_player="o"

elif current\_player=="o":

current\_player="x"

return

play\_game()