Tic-Tac-Toe game without using OOP's

```
from termcolor import colored

# showing the game board after plotting any "X/0"
def game_board(x):
    try:
        print()
        print (f" {x[1]} | {x[2]} | {x[3]} ")
        print (f"---|---|---")
        print (f" {x[4]} | {x[5]} | {x[6]} ")
        print (f"---|---|---")
        print (f" {x[7]} | {x[8]} | {x[9]} ")
        print
    except Exception as e:
        pass

# Genereating possible outcome from user inputed options
def possibilities(x):
    poss_user = []
    for i in range(0,len(x)-2):                            # pos 1
        for j in range(1,len(x)-1):                        # pos 2
            for k in range(1,len(x)):                      # pos 3
                total=f"{x[i]}{x[j]}{x[k]}"                # joining
                poss_user.append(int(total))               # adding to
list of possiblities
    return poss_user


# Checking tthe final score if any user has made XXX/000 in a row
def check_score(x):
    fp = open("final_possible_scores.txt","r")
    data = eval(fp.read())                                       # All
possibilities of XXX/000 in a row on the board
    final_possible_scores = set(data)                            #
reading from a file


    user_score = set(possibilities(x))                            #
possiblities of users

    if (final_possible_scores & user_score):                   #
checking user inputs and possible win situations
        return True
    else:
        return False
```

```
# Places Reamining on the board for the players
dict = {1: '1', 2: '2', 3: '3', 4: '4', 5: '5', 6: '6', 7: '7', 8:
'8', 9: '9'}
choice = [1,2,3,4,5,6,7,8,9]

# User selected Locations
p1 = []
p2 = []

#Inital Board Situation
game_board(dict)

# will run until all spaces on board are occupied
while(choice!=[]):

    # Odd number means Player 1 will be playing
    if(len(choice)%2 != 0):
        try:
            pos1 = int(input(f"\nChoices Remaining : {len(choice)} ||
Player 1 Position : "))    # usr 1 I/P
        except Exception as e:
            print ("Please select an option from a range of 1-9")
        else:
            if pos1 in choice:              # Checking availability of
usr entered location on the board
                dict[pos1]= "X"             # updating Dictionary for
the player by putting "X" as value for the location on board
                p1.append(pos1)             # updating p1 ie. user
selced loations
                choice.remove(pos1)         # removing the choice so
other player cannot insert at that position
                game_board(dict)            # showing the O/P

            if(len(p1) >= 3):               # usr should have played
atleast 3 times to make a winning possibility
                if(check_score(p1)):        # checking score and
printing result
                    text = colored('PLAYER 1 IS WINNER', 'red',
attrs=['reverse', 'blink'])
                    print("\n|--------------------------|")
                    print(f"|--- {text} ---|")
                    print("|--------------------------|")
                    break

    # Even number means Player 2 will be playing
    else:
        try:
            pos2 = int(input(f"\nChoices Remaining : {len(choice)} ||
Player 2 Position : "))    # usr 2 I/P
```

```
        except Exception as e:
            print ("Please select an option from a range of 1-9")
        else:
            if pos2 in choice:
                dict[pos2]= "0"
                p2.append(pos2)
                choice.remove(pos2)
                game_board(dict)

            if(len(p2) >= 3):
                if(check_score(p2)):
                    text = colored('PLAYER 2 IS WINNER', 'red',
attrs=['reverse', 'blink'])
                    print("\n|--------------------------|")
                    print(f"|--- {text} ---|")
                    print("|--------------------------|")
                    break

# Match Draw
# If P1 & P2 both are not matching any of the winning possibilities
if( (check_score(p1)==False) and (check_score(p2)==False) ):
    text = colored('MATCH DRAW', 'red', attrs=['reverse', 'blink'])
    print("\n|--------------------------|")
    print(f"|------- {text} -------|")
    print("|--------------------------|")
```

Tic-Tac-Toe using OOP's

```
from termcolor import colored

# showing the game board after plotting any "X/0"
class Game():
    def __init__(self):
        self.player_list = []

    def game_board(x):
        try:
            print()
            print (f" {x[1]} | {x[2]} | {x[3]} ")
            print (f"---|---|---")
            print (f" {x[4]} | {x[5]} | {x[6]} ")
            print (f"---|---|---")
            print (f" {x[7]} | {x[8]} | {x[9]} ")
            print
        except Exception as e:
            pass

    # Genereating possible outcome from user inputed options
    def possibilities(x):
        poss_user = []
        for i in range(0,len(x)-2):                            # pos 1
            for j in range(1,len(x)-1):                        # pos 2
                for k in range(1,len(x)):                      # pos 3
                    total=f"{x[i]}{x[j]}{x[k]}"                #
joining
                    poss_user.append(int(total))               #
adding to list of possiblities
        return poss_user


    # Checking tthe final score if any user has made XXX/000 in a row
    def check_score(x):
        fp = open("final_possible_scores.txt","r")
        data = eval(fp.read())                                       #
All possibilities of XXX/000 in a row on the board
        final_possible_scores = set(data)                            #
reading from a file


        user_score = set(Game.possibilities(x))
# possiblities of users

        if (final_possible_scores & user_score):                   #
checking user inputs and possible win situations
            return True
        else:
            return False



# Places Reamining on the board for the players
dict = {1: '1', 2: '2', 3: '3', 4: '4', 5: '5', 6: '6', 7: '7', 8:
'8', 9: '9'}
choice = [1,2,3,4,5,6,7,8,9]

# User selected Locations
p1 = Game()
p2 = Game()

#Inital Board Situation
Game.game_board(dict)

# will run until all spaces on board are occupied
while(choice!=[]):

    # Odd number means Player 1 will be playing
    if(len(choice)%2 != 0):
        try:
            pos1 = int(input(f"\nChoices Remaining : {len(choice)} ||
Player 1 Position : "))    # usr 1 I/P
        except Exception as e:
            print ("Please select an option from a range of 1-9")
        else:
            if pos1 in choice:              # Checking availability of
usr entered location on the board
                dict[pos1]= "X"             # updating Dictionary for
the player by putting "X" as value for the location on board
                p1.player_list.append(pos1)             # updating p1
ie. user selced loations
                choice.remove(pos1)         # removing the choice so
other player cannot insert at that position
                Game.game_board(dict)            # showing the O/P

            if(len(p1.player_list) >= 3):               # usr should
have played atleast 3 times to make a winning possibility
                if(Game.check_score(p1.player_list)):        #
checking score and printing result
                    text = colored('PLAYER 1 IS WINNER', 'red',
attrs=['reverse', 'blink'])
                    print("\n|--------------------------|")
                    print(f"|--- {text} ---|")
                    print("|--------------------------|")
                    break

    # Even number means Player 2 will be playing
    else:
        try:
            pos2 = int(input(f"\nChoices Remaining : {len(choice)} ||
Player 2 Position : "))    # usr 2 I/P
        except Exception as e:
            print ("Please select an option from a range of 1-9")
        else:
            if pos2 in choice:
                dict[pos2]= "0"
                p2.player_list.append(pos2)
                choice.remove(pos2)
                Game.game_board(dict)

            if(len(p2.player_list) >= 3):
                if(Game.check_score(p2.player_list)):
                    text = colored('PLAYER 2 IS WINNER', 'red',
attrs=['reverse', 'blink'])
                    print("\n|--------------------------|")
                    print(f"|--- {text} ---|")
                    print("|--------------------------|")
                    break

# Match Draw
# If P1 & P2 both are not matching any of the winning possibilities
if( (Game.check_score(p1.player_list)==False) and
(Game.check_score(p2.player_list)==False) ):
    text = colored('MATCH DRAW', 'red', attrs=['reverse', 'blink'])
    print("\n|--------------------------|")
    print(f"|------- {text} -------|")
    print("|--------------------------|")

#     final_possible_scores = { 123,132,147,174,159,195,
#                              213,231,258,285,
#                              312,321,369,396,357,375,
#                              417,471,456,465,
# All possibilities of XXX/000 in a row on the board
#                              519,591,537,573,528,582,546,564,
#                              639,693,654,645,
#                              714,741,789,798,753,735,
#                              879,897,852,825,
#                              915,951,936,963,978,987 }
```