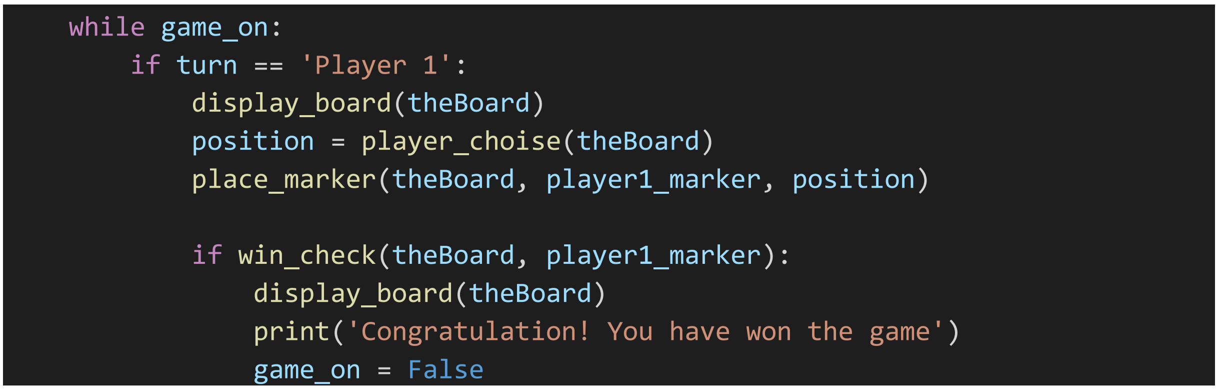
Assignment

Practice the project taught in today's lecture.

```
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('   |   |')
test_board = ['#','X','0','X','0', 'X', '0', 'X', '0','X']
display_board(test_board)
def player_input():
    marker = ''

    while not (marker == 'X' or marker == '0'):
        marker = input('Player 1: Do you want to be X or O? ').upper()

        if marker == 'X':
            return ('X', '0')

        else:
            return ('0', '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 #
accross the top(board[4] == mark and board[5] == mark and board[6] == mark) or
# across the middle
    (board[4] == mark and board[5] ==mark and board[6] == mark ) or
    (board[1] == mark and board[2] == mark and board[3] == mark) or # across
the bottom
    (board[7] == mark and board[4] == mark and board[1] == mark) or # down the
middle
    (board[8] == mark and board[5] == mark and board[2] == mark) or #  down
the middle
```

```
    (board[9] == mark and board[6] == mark and board[3] == mark) or # down the
right side
    (board[7] == mark and board[5] == mark and board[3] == mark) or # diagonal
    (board[9] == mark and board[5] == mark and board[1] == mark))  # diagonal

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_choise(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 reply():
    return input('Do you want to play again ? Enter Yes or No:
').lower().startswith('y')
print('Welcome to LetsUpgrade Tic Tac Toe Game')

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':
            display_board(theBoard)
            position = player_choise(theBoard)
            place_marker(theBoard, player1_marker, position)

            if win_check(theBoard, player1_marker):
                display_board(theBoard)
                print('Congratulation! You have won the game')
                game_on = False
```