#CodeYork

Assignment 4

Summer 2017

Task 1 - The Game Board

Our game game board representation will be 2D arrays. Consider the following:

```
board = [['O', 'X', None], [None, 'O', 'X'], ['X', None, 'O']]
```

Here, the Os have won by placing zeros across the leading diagonal. You can access the top row, middle row, and bottom row, respectively with:

```
board[0]        board[1]        board[2]
```

From each row, we can get to the elements of it just by using the index notation again. Here we will get the middle right symbol, which should be an X:

```
board[1][2]
```

Try all this out in the IDLE interpreter, and ask if this is not making any sense.

Task 2 - Tic-Tac-Toe Code

Take the code off the website, save it on your machine, and then **open it up in IDLE**. Have a quick **read through it**, and make sure you understand roughly how it works, and how it's structured.

Note: The code won't work yet. It is your job in the later tasks to change this.

Task 3 - Implementing Draws

There are two areas in the code that need finishing in order for the game to work. You will be implementing these in this task and the next.

A game is drawn if neither play has won, and the board is full. **You will be writing the *is_full* function.** If the board is full, you need to return true.

Hint: Think about what it means for a board to be full in our representation. There must be no None entries.

Task 4 - Implementing Winners

The second thing that needs implementing is the *has_won* function. You need to **modify the code to return true if the given player has won**.

Hint: A player can win by taking any row, any column, and either diagonal. You will need to check all of these. Think how you can use for loops to reduce code duplication.

Task 5 - Testing the Code

You can run the code from idle with the *F5* key. **Check it works correctly.** If not, go back, and check your answers to tasks 3 and 4.

Task 6 - Pretty Printing

The current code simply prints the 2D array out to the screen to represent the board. Now you will **improve this by modifying the *print_board* function** to print something that actually looks like a real game board.