SECTION 7: MILESTONE PROJECT - 1, 1 hour 40 minutes, 9 parts

- **6/9 First Python Milestone Project Overview**
  - Programming a ttt board from the functoins in the previous lectures
  - **about the game**
    - two players, same computer
    - board printed out every time a move is made
    - accepting the input of the player position and placing a symbol on the board

  - **the code**
    - numpad
    - a larger keyboard
    - a board of 9 numbers -> two lists placed on top of each other
    - -> there is a ttt board with an x at one location
    - -> there is a walk through notebook

  - **in the project .ipynb file**
    - -> she is showing the solution JN (the output)
    - -> an input is being asked for -> x or 0
    - -> are you ready to play yes or no
    - -> a board is printed out -> with {}'s in each of the sections on the board
    - -> one user onputs a position
    - -> then an 0 is put there on the board
    - -> the process repeats
    - -> until someone wins
    - -> then there is a message to show that they've won
    - -> and it asks if they want to play again

  - **on GitHub**
    - -> milestone project repo
    - -> one of then is the JN for the current video lecture
    - -> the main one is called the walkthrough steps workbook -> this is to walk through the steps in the problem
      - this contains hints
      - **-> clear_output() <- this can be used to clear the screen**
    - -> the project is divided into steps -> and each of then has tests (for the code)
      - **-> testing the code after each time it's been ran**
      - -> write a function which will print out the board
      - -> you can call the function later in the code to integrate in the board

  - **in the project .ipynb file**
    - **-> printing out the board**
      - there is a test board list -> an empty list with 9 parts
      - to print out the board -> you can print out three lists, stacked on top of each other
      - **-> the function takes a list -> and returns the elements in the array stacked on top of each other**
      - **-> clear_output() -> this gets rid of what is on the screen and replaces it with something else**

    - **-> write a function which takes input**
      - player_input()
      - -> marker = ''
      - -> then while marker != 'x' and marker != 'o':

- -> in other words, it's doing nothing if the input from the user is wrong
- -> the return is either as a list or a tuple -> so tuple unpacking can be used later
- -> **the idea is to build a while loop in case the input of the user is in the wrong format**
- -> you can also do, in this example player1_marker, mayer2_marker = player_input()