**Outline**

Access the Python Development environment and continue the tutorial to gain an additional exposure to the Python programming language. Begin to develop an familiarity with intermediate programming concepts.

**Objectives**

* Use correct terminology to describe programming concepts;
* Describe the types of data that computers can process and store (e.g., numbers, text);
* Explain the difference between constants and variables used in programming;
* Use variables, expressions, and assignment statements to store and manipulate numbers and text in a program

**Materials**

* Python3 Development Environment at: //repl.it/
* Python Tutorial at: <http://www.letslearnpython.com/learn/>

**Accessing the Tutorial**

Accessing the Tutorial

* Go to: <http://www.letslearnpython.com/learn/>
* Read up to “Lesson 12: Input”

**Level 1: Input & Output**

1. Read through “Lesson 12: Input – What Is Input?” and “Lesson 12: Input – Example” and “Lesson 12: Input – Shortcut”.
2. Type the following code into the white area of the IDE and run the program. Explain what you see in the black area of the IDE.

print("Type your name:")

name = input()

print("Hi", name, "how are you?")

1. Create a short program that reads numerical input from the console and does the following:
   1. Uses the input() function to read a numerical value from the console.

print("Type your name:")

name = input()

print("Hi", name, "how are you?")

* 1. Calculates the square root of the number

num = 8

num\_sqrt = num \*\* 0.5

print('The square root of %0.3f is %0.3f'%(num ,num\_sqrt))

* 1. Prints the result to the console output

The square root of 8.00 is 2.828

* 1. Provides appropriate prompt and message strings to go with the input and output.
  2. Provide your complete program below.

**Level 2: Tic-Tac-Toe Game**

1. Write a Python program to play a game of Toc-Tac-Toe. (You may modify a program that you found on-line to meet the expectations of this module.)
   1. The program may be either player v. computer or player 1 v. player 2.
   2. The program does not need to determine a winner
   3. The program just needs to keep track of moves and spaces in the game board
2. Provide a complete listing of your program.
   1. Your listing **MUST** include line numbers .

import random

#GamePlay Board

Board = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

#3x3 board Positions 1-9

def showBoard():

print(Board[1], "│", Board[2], "│", Board[3])

print("━━━━━━━━━━")

print(Board[4], "│", Board[5], "│", Board[6])

print("━━━━━━━━━━")

print(Board[7], "│", Board[8], "│", Board[9])

#Checks each line for winning combination

def checkLine(player, spot1, spot2, spot3):

if Board[spot1] == player and Board[spot2] == player and Board[

spot3] == player:

return True

#Checks all posible combinations for winner

def checkWinner(player):

if checkLine(player, 1, 2, 3):

return True

if checkLine(player, 4, 5, 6):

return True

if checkLine(player, 7, 8, 9):

return True

if checkLine(player, 1, 4, 7):

return True

if checkLine(player, 2, 5, 8):

return True

if checkLine(player, 3, 6, 9):

return True

if checkLine(player, 1, 5, 9):

return True

if checkLine(player, 3, 5, 7):

return True

#Game function for moves and gameplay

while True:

showBoard()

#user input (1-9) User= X

X = input("Choose a spot (1-9):")

X = int(X)

if Board[X] != 'O' and Board[X] != 'X':

Board[X] = 'X'

else:

print("This spot is already taken")

#Checks if X won the game or not

if checkWinner('X') == True:

print("━━━━━━!!You beat the Computer, YOU WON!!━━━━━━")

break;

#Checks if game is DRAW

if checkWinner('O') == False and checkWinner('X') == False:

print('━━━━━━!!Nobody Wins, DRAW!!━━━━━━')

break;

#comuter Move Computer= O

while True:

O = random.randint(1, 9)

if Board[O] != 'O' and Board[O] != 'X':

Board[O] = 'O'

break;

#Checks if X won the game or not

if checkWinner('O') == True:

print('━━━━━━!!The Computer beat You, YOU LOST!!━━━━━━')

break;

1. Explain how your program keeps track of the game board.   
   (Provide specific code references by line number.)
   1. What python types and data structures are used?

It’s a List and Set

* 1. How are moves by player X and player O recorded?

They are recorded by putting either a X or a O on the number they entered

* 1. How are free spaces recorded?

Free space is recorded by the number of space they are on

1. Explain how moves and commands are input from the console.  
   (Provide specific code references by line number.)
   1. How does the player tell the program about the move location (row, column)?

Player must enter a number of the spot they want

* 1. How does the program verify that the move location is valid?

If it has not been chosen it is a free spot for the player

* 1. How does the program verify that the space is free?

If there is still a number on the spot it is free to use

* 1. What does the program do if there is something wrong with the move?  
     if someone chose your spot I would say spot taken but I it’s the number over 9 it would give an error and you would have to restart the game

1. Explain how the program keeps track of gameplay.  
   (Provide specific code references by line number.)
   1. How does the program switch between player X and player O moves?
2. move = int(input("player X, Make a move:"))
3. board[move] = pX
4. printBoard
5. move = int(input("player O, make a move:"))
6. board[move] = pO
7. printBoard()
   1. How does the program keep asking for moves?

Because of the code it asks if player one wants to make a move for example look at the code above.

* 1. How does the program decide when to stop asking for moves?

When a player wins or it say do you want to stop Y/N and if you want to stop you type “Y”.

**Level 3: Basic Enhancements**

1.    Explain, in plain words, a strategy for determining if player “x” or player “O” has won the game after a move is made.

If player “X” or player “O” has their icon in a full row, column or diagonal they win the game and it would say you beat the computer.

2.    Provide a function called “checkWinForX” that returns the Boolean value of “True” if player “x” won the game.

def checkWinner(player):

if checkLine(player, 1, 2, 3):

return True

if checkLine(player, 4, 5, 6):

return True

if checkLine(player, 7, 8, 9):

return True

if checkLine(player, 1, 4, 7):

return True

if checkLine(player, 2, 5, 8):

return True

if checkLine(player, 3, 6, 9):

return True

if checkLine(player, 1, 5, 9):

return True

if checkLine(player, 3, 5, 7):

return True

if the player has the Boolean value true the player wins the game

3.    Modify your program to check and print a message, and stop the game of player “x” or player “O” wins the game.

if checkWinner('X') == True:

print("━━━━━━!!You beat the Computer, YOU WON!!━━━━━━")

break;

this is the winning message for the winner

4.    Demonstrate your enhanced game to Mr. Nestor for credit for this level.

**Level 4: AI Enhancements**

1.    Explain, in plain words, a strategy for suggesting the best move for player “x” or player “O” to make when it is their turn to move.

To beat the AI most of the time just go to place 1 2 3 and you’ll win most of the time

2.    Create a function to implement your strategy for suggesting the best move.

3.    Modify your program to print a suggested move when it is each player’s turn to move.

4.    Demonstrate your AI enhanced game to Mr. Nestor for credit for this level.