**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?")

**After printing this into repl.it, it will say “type your name”. U after type your name and then it will say “Hi, taranpreet, 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.
2. print("Enter a number from 1 to 9 : " )
3. name = input(1)

* 1. Calculates the square root of the number

**(10/5)**

* 1. Prints the result to the console output

**2**

* 1. Provides appropriate prompt and message strings to go with the input and output.

**I used “sqrt” to square root a number. I used 10 and 5. I printed sqrt = (10/5) print(sqrt) and got 2.**

* 1. Provide your complete program below.

**sqrt = (10/5)**

**print(sqrt)**

**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 .
3. 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?

**The python I used was python turtle. I used spaces and “tab” to complete the work.**

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

**I used “input”, “if”, “print”, to undergo the turns for each player. Each player has 1 to 8 spots to move too.**

* 1. How are free spaces recorded?

**The free spaces have nothing on them until they have a “X” or “O” on them. you can pick a number between 1 to 8 to put on the board.**

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)?

**It is easy to understand where to move for any row or column. This game is easy to understand and play.**

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

**Whenever a row or column is picked, a “X” or “O” is put on the row or column.**

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

**If you use a row or column that is already used, the game will skip the player who picked the already used row or column**

* 1. What does the program do if there is something wrong with the move?  
     **Nothing will happen and the next player is pick a spot.**

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?

**After someone picked a row or column, the next person will go and so on and so on**

* 1. How does the program keep asking for moves?

**it wont. It will skip the persons move and move on.**

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

**It will stop asking after someone wins and he other loses.**

theBoard = {'0': ' ','1': ' ','2': ' ',

'3': ' ','4': ' ','5': ' ',

'6': ' ','7': ' ','8': ' '}

def printBoard(board):

print(board['0'] + '|' + board['1'] + '|' + board['2'])

print('\_\_\_\_\_')

print(board['3'] + '|' + board['4'] + '|' + board['5'])

print('\_\_\_\_\_')

print(board['6'] + '|' + board['7'] + '|' + board['8'])

turn = 'X'

for i in range(8):

printBoard(theBoard)

print('Welcome player ' + turn + '. Move on which space?')

print('0, 1, 2, 3, 4, 5, 6, 7, 8')

move = input()

theBoard[move] = turn

if turn == 'X':

turn = 'O'

else:

turn = 'X'

theBoard = {'Top-Left': ' ','Top-Middle': ' ','Top-Right': ' ',

'Centre-Left': ' ','Centre-Middle': ' ','Centre-Right': ' ',

'Bottom-Left': ' ','Bottom-Middle': ' ','Bottom-Right': ' '}

def printBoard(board):

print(board['0'] + '|' + board['1'] + '|' + board['2'])

print('\_\_\_\_\_')

print(board['3'] + '|' + board['4'] + '|' + board['5'])

print('\_\_\_\_\_')

print(board['6'] + '|' + board['7'] + '|' + board['8'])

turn = 'O'

for i in range(8):

printBoard(theBoard)

print('Turn for ' + turn + '. Move on which space?')

print('0, 1, 2, 3, 4, 5, 6, 7, 8')

move = input()

theBoard[move] = turn

if turn == 'X':

turn = 'O'

else:

turn = 'X'

printBoard(theBoard)

**Level 3: Enhancements**

t.b.d.