**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?")

It asks you to type your name in the black area, and then it says “Hi (your 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.
   2. Calculates the square root of the number
   3. Prints the result to the console output
   4. Provides appropriate prompt and message strings to go with the input and output.
   5. Provide your complete program below.

print("Type a number you want to square root between 1-10")

number=input()

if number==1:

print("1")

elif number==2:

print("1.41421356237")

elif number==3:

print("1.73205080757")

elif number==4:

print ("4")

elif number==5:

print("2.2360679775")

elif number==6:

print("2.44948974278")

elif number==7:

print("2.64575131106")

elif number==8:

print("2.82842712475")

elif number==9:

print("3")

elif number==10:

print("3.16227766017")

**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. pX="X"
4. pO="O"
5. pN=" "
6. board=[pN,pN,pN,
7. pN,pN,pN,
8. pN,pN,pN]
9. def printBoard():
10. print()
11. print(board[0],"|",board[1],"|", board[2])
12. print("---------")
13. print(board[3],"|",board[4],"|", board[5])
14. print("---------")
15. print(board[6],"|",board[7],"|", board[8])
16. print()
17. while True:
18. move=int(input("Player X, Make a move:"))
19. if move>8:
20. move=int(input("Choose another place between 0-8:"))
21. if move==0 and (board[0]==pX or board[0]==pO):
22. move=int(input("This spot is taken! Choose another spot:"))
23. if move==1 and (board[1]==pX or board[1]==pO):
24. move=int(input("This spot is taken! Choose another spot:"))
25. if move==2 and (board[2]==pX or board[2]==pO):
26. move=int(input("This spot is taken! Choose another spot:"))
27. if move==3 and (board[3]==pX or board[3]==pO):
28. move=int(input("This spot is taken! Choose another spot:"))
29. if move==4 and (board[4]==pX or board[4]==pO):
30. move=int(input("This spot is taken! Choose another spot:"))
31. if move==5 and (board[5]==pX or board[5]==pO):
32. move=int(input("This spot is taken! Choose another spot:"))
33. if move==6 and (board[6]==pX or board[6]==pO):
34. move=int(input("This spot is taken! Choose another spot:"))
35. if move==7 and (board[7]==pX or board[7]==pO):
36. move=int(input("This spot is taken! Choose another spot:"))
37. if move==8 and (board[8]==pX or board[8]==pO):
38. move=int(input("This spot is taken! Choose another spot:"))
39. board[move]=pX
40. printBoard()
41. if board[0]==pX and board[1]==pX and board[2]==pX:
42. print("X Wins")
43. break
44. if board[3]==pX and board[4]==pX and board[5]==pX:
45. print("X Wins")
46. break
47. if board[6]==pX and board[7]==pX and board[8]==pX:
48. print("X Wins")
49. break
50. if board[0]==pX and board[3]==pX and board[6]==pX:
51. print("X Wins")
52. break
53. if board[1]==pX and board[4]==pX and board[7]==pX:
54. print("X Wins")
55. break
56. if board[2]==pX and board[5]==pX and board[8]==pX:
57. print("X Wins")
58. break
59. if board[0]==pX and board[4]==pX and board[8]==pX:
60. print("X Wins")
61. break
62. if board[6]==pX and board[1]==pX and board[2]==pX:
63. print("X Wins")
64. break
65. if board[2]==pX and board[4]==pX and board[6]==pX:
66. print("X Wins")
67. break
68. if board[0]==pO and board[1]==pO and board[2]==pO:
69. print("O Wins")
70. break
71. if board[3]==pO and board[4]==pO and board[5]==pO:
72. print("O Wins")
73. break
74. if board[6]==pO and board[7]==pO and board[8]==pO:
75. print("O Wins")
76. break
77. if board[0]==pO and board[3]==pO and board[6]==pO:
78. print("O Wins")
79. break
80. if board[1]==pO and board[4]==pO and board[7]==pO:
81. print("O Wins")
82. break
83. if board[2]==pO and board[5]==pO and board[8]==pO:
84. print("O Wins")
85. break
86. if board[0]==pO and board[4]==pO and board[8]==pO:
87. print("O Wins")
88. break
89. if board[6]==pO and board[1]==pO and board[2]==pO:
90. print("O Wins")
91. break
92. if board[2]==pO and board[4]==pO and board[6]==pO:
93. print("O Wins")
94. break
95. move=int(input("Player O, Make a move:"))
96. if move>8:
97. move=int(input("Choose another place between 0-8:"))
98. if move==0 and (board[0]==pX or board[0]==pO):
99. move=int(input("This spot is taken! Choose another spot:"))
100. if move==1 and (board[1]==pX or board[1]==pO):
101. move=int(input("This spot is taken! Choose another spot:"))
102. if move==2 and (board[2]==pX or board[2]==pO):
103. move=int(input("This spot is taken! Choose another spot:"))
104. if move==3 and (board[3]==pX or board[3]==pO):
105. move=int(input("This spot is taken! Choose another spot:"))
106. if move==4 and (board[4]==pX or board[4]==pO):
107. move=int(input("This spot is taken! Choose another spot:"))
108. if move==5 and (board[5]==pX or board[5]==pO):
109. move=int(input("This spot is taken! Choose another spot:"))
110. if move==6 and (board[6]==pX or board[6]==pO):
111. move=int(input("This spot is taken! Choose another spot:"))
112. if move==7 and (board[7]==pX or board[7]==pO):
113. move=int(input("This spot is taken! Choose another spot:"))
114. if move==8 and (board[8]==pX or board[8]==pO):
115. move=int(input("This spot is taken! Choose another spot:"))
116. board[move]=pO
117. printBoard()
118. if board[0]==pX and board[1]==pX and board[2]==pX:
119. print("X Wins")
120. break
121. if board[3]==pX and board[4]==pX and board[5]==pX:
122. print("X Wins")
123. break
124. if board[6]==pX and board[7]==pX and board[8]==pX:
125. print("X Wins")
126. break
127. if board[0]==pX and board[3]==pX and board[6]==pX:
128. print("X Wins")
129. break
130. if board[1]==pX and board[4]==pX and board[7]==pX:
131. print("X Wins")
132. break
133. if board[2]==pX and board[5]==pX and board[8]==pX:
134. print("X Wins")
135. break
136. if board[0]==pX and board[4]==pX and board[8]==pX:
137. print("X Wins")
138. break
139. if board[6]==pX and board[1]==pX and board[2]==pX:
140. print("X Wins")
141. break
142. if board[2]==pX and board[4]==pX and board[6]==pX:
143. print("X Wins")
144. break
145. if board[0]==pO and board[1]==pO and board[2]==pO:
146. print("O Wins")
147. break
148. if board[3]==pO and board[4]==pO and board[5]==pO:
149. print("O Wins")
150. break
151. if board[6]==pO and board[7]==pO and board[8]==pO:
152. print("O Wins")
153. break
154. if board[0]==pO and board[3]==pO and board[6]==pO:
155. print("O Wins")
156. break
157. if board[1]==pO and board[4]==pO and board[7]==pO:
158. print("O Wins")
159. break
160. if board[2]==pO and board[5]==pO and board[8]==pO:
161. print("O Wins")
162. break
163. if board[0]==pO and board[4]==pO and board[8]==pO:
164. print("O Wins")
165. break
166. if board[6]==pO and board[1]==pO and board[2]==pO:
167. print("O Wins")
168. break
169. if board[2]==pO and board[4]==pO and board[6]==pO:
170. print("O Wins")
171. break
172. 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 data structure that is used is print and define. Lines 1-3 and 9-16 are an example.

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

At the beginning, the players are defined. pX is player X, pO is player O. Whenever the person makes a move, move is the input and board[move]=pX is when it is recorded in the board. Lines 19-51 are an example.

* 1. How are free spaces recorded?

Free spaces already exist at the beginning. They were created by blank spaces. Lines 3-7 are an example.

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)?

The player tells the program a number from 0-9. Then, the program does the following program if the input is between 0-8. Lines 19-22 are an example for player X.

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

If the person prints anything else than a number between 0 and 8, it will give the message “Choose another place between 0-8”. Lines 21 and 22 are an example for player X.

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

There is a program which checks to see if it is free or not. Lines 24-49 are an example.

* 1. What does the program do if there is something wrong with the move?

If the number is more than 8, it will ask the person to choose somewhere to go using the numbers 0-8. Lines 21-22 check that. Words are obviously not going to be inputted, so it will do nothing if someone inputs a letter or word or sentence.

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?

When player X has gone, it checks for any win. Then it goes onto player O. Lines 53 to 123 is an example of how the program takes in the input and checks for any win.

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

It is set on repeat. Line 18 is an example. The program goes as following: It first asks for Player X for a move, then if the input is greater than 8, it asks for something from 0-8. After that, it checks if the spot is taken, and then checks for any win. The program does the same thing for Player O(the program is written out). And then, it starts again from the beginning.

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

It stops after someone has won. Lines 53 to 123 are an example.

**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.

To determine if player “x” or player “o” has won, you could check if player X has gone in three spots in a row. This would require the program to check the board spots.

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

The following program does a check for both X and O.

def checkWinForX():

if board[0]==pX and board[1]==pX and board[2]==pX:

print("X Wins")

elif board[3]==pX and board[4]==pX and board[5]==pX:

print("X Wins")

elif board[6]==pX and board[7]==pX and board[8]==pX:

print("X Wins")

elif board[0]==pX and board[3]==pX and board[6]==pX:

print("X Wins")

elif board[1]==pX and board[4]==pX and board[7]==pX:

print("X Wins")

elif board[2]==pX and board[5]==pX and board[8]==pX:

print("X Wins")

elif board[0]==pX and board[4]==pX and board[8]==pX:

print("X Wins")

elif board[6]==pX and board[1]==pX and board[2]==pX:

print("X Wins")

def checkWinForO():

if board[0]==pO and board[1]==pO and board[2]==pO:

print("O Wins")

elif board[3]==pO and board[4]==pO and board[5]==pO:

print("O Wins")

elif board[6]==pO and board[7]==pO and board[8]==pO:

print("O Wins")

elif board[0]==pO and board[3]==pO and board[6]==pO:

print("O Wins")

elif board[1]==pO and board[4]==pO and board[7]==pO:

print("O Wins")

elif board[2]==pO and board[5]==pO and board[8]==pO:

print("O Wins")

elif board[0]==pO and board[4]==pO and board[8]==pO:

print("O Wins")

elif board[6]==pO and board[1]==pO and board[2]==pO:

print("O Wins")

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

def checkWinForX():

if board[0]==pX and board[1]==pX and board[2]==pX:

print("X Wins")

break

elif board[3]==pX and board[4]==pX and board[5]==pX:

print("X Wins")

break

elif board[6]==pX and board[7]==pX and board[8]==pX:

print("X Wins")

break

elif board[0]==pX and board[3]==pX and board[6]==pX:

print("X Wins")

break

elif board[1]==pX and board[4]==pX and board[7]==pX:

print("X Wins")

break

elif board[2]==pX and board[5]==pX and board[8]==pX:

print("X Wins")

break

elif board[0]==pX and board[4]==pX and board[8]==pX:

print("X Wins")

break

elif board[6]==pX and board[1]==pX and board[2]==pX:

print("X Wins")

break

def checkWinForO():

if board[0]==pO and board[1]==pO and board[2]==pO:

print("O Wins")

break

elif board[3]==pO and board[4]==pO and board[5]==pO:

print("O Wins")

break

elif board[6]==pO and board[7]==pO and board[8]==pO:

print("O Wins")

break

elif board[0]==pO and board[3]==pO and board[6]==pO:

print("O Wins")

break

elif board[1]==pO and board[4]==pO and board[7]==pO:

print("O Wins")

break

elif board[2]==pO and board[5]==pO and board[8]==pO:

print("O Wins")

break

elif board[0]==pO and board[4]==pO and board[8]==pO:

print("O Wins")

break

elif board[6]==pO and board[1]==pO and board[2]==pO:

print("O Wins")

break

1. 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.

For the human playing, I would suggest anywhere where it would block the AI. For AI, I would set up different spots to go. For example, I would make the AI bock the player first, and then make the AI make a smart move.

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

If board[0] and board[1]==pX:

Board[2]==pO

The program above would be repeated anywhere someone would be trying to win.

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

If board[0] and board[1]==pX:

Print(“Recommendation: Go for place 3”)

The above program would be repeated anywhere someone would be trying to win.

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