Dylan

**Module B.2 Python Programming Introduction**

**Level 1: Basic Math & Strings**

Access the Tutorial and start at “Lesson 3: Math”.

Questions

1. Complete “Lesson 3: Math – Math Basics” by typing the sample commands in the black area of the IDE.
   1. Create your own expression using 5 “+” and “-“ operators.

20+2-4+6-1+9

* 1. List your expression and the result below.

32

1. Complete “Lesson 3: Math – More Operators” by typing the sample commands in the black area of the IDE.
   1. Create your own expression using 5 “\*” and “/” operators.

5\*9/2\*5\*6\*3

* 1. List your expression and the result below.

2025

1. Complete “Lesson 3: Math – More Division” by typing the sample commands in the black area of the IDE.
   1. Create one division expression that gives a whole number answer.

24/3= 8

* 1. And one division expression that gives a decimal number answer.

19/2= 9.5

* 1. List your expressions and the results below.
* 24/3= 8
* 19/2= 9.5

1. Complete “Lesson 3: Math – Floats” by typing the sample commands in the black area of the IDE.
   1. Use the “round()” function for the expressions you created in question #3 above.
   2. List your “round()” expressions and the results they return below.

Round(19/2)

= 10

1. Read through “Lesson 3: Math – Comparison Operators”.
   1. Why do you think Equals is “==” instead of “=”?

I think “==” is equal because the one equal sign would be used in other symbols. So, if the equal sign is used in other symbols then there should be a special sign for “equal is”

* 1. What does “=” mean?

The sign “=” would mean that you are appointing a variable on the left side of the equal sign a value for the variable to be when you would type the variable by itself again.

1. Complete “Lesson 3: Math – Practice” and “Lesson 3: Math – Practice Answers” by typing the sample commands in the black area of the IDE.
   1. Create an expression using 5 different operators that returns a “True” result
   2. And an expression using 5 different operators that returns a “False” result.
   3. List your expressions and the results returned below.

True Results

10\*4+6>20/2-6

* True

False Results

(5\*6)-2==(60-2)+5

* False

1. Complete “Lesson 4: Strings – Strings” and “Lesson 4: Strings – Examples” by typing the sample commands in the black area of the IDE.
   1. Explain why typing “apple” works and why typing apple without quotes gives an error.

If you want to type a string in Python you would have to write it in quotations because Python can only strings in quotations.

* 1. Also explain why “2 + 5” does not equal 7.

“2 +5” would not equal 7 because Python is reading this expression as a string due to the quotations around the “2 + 5”.

1. Complete “Lesson 4: Strings – Operators” by typing the sample commands in the black area of the IDE.
   1. Explain why typing “appl” + “e” works and why typing “apple” - “e” gives an error.

“appl” + “e” would work because you can put words together. While “apple” – “e” does not work because you cannot take words apart in Concatenation.

* 1. Also explain why “Hello” \* 10 works but why “Hello” / 10 does work.

“Hello” \* 10 would work because this would tell Python that you would want to repeat the word 10 times. While “Hello” / 10 is telling Python to divide the word by 10 and you cannot divide a word.

1. Complete “Lesson 4: Strings – Indexes” by typing the sample commands in the black area of the IDE.
   1. List the letters in your first name and the index for each letter in your first name.

My name letters and the index for each letter

* D 0
* y 1
* l 2
* a 3
* n 4

1. Complete “Lesson 4: Strings – Indexes Examples” by typing the sample commands in the black area of the IDE.
   1. Explain why print(“Hello!”[4]) does not print “l”.

(“Hello!”[4]) does not print “l because when you count the index for each letter, you would start with zero. So, “H” is 0, “e” is 1, “l” is 2, “l” is 3 and “o” is 4. Since “o” is 4, Python would “o” instead of “l” because “l” is 3 and not 4.

* 1. What does print(“Hay, Bob!”[4]) print? For a hint try print(“Hay, Bob!”[3]) and print(“Hay, Bob!”[5])

When you would write print(“Hay, Bob!”[4]) in Python it would print nothing because the index of 4 in “Hay, Bob!” is the space between the comma and “B”.

1. Complete “Lesson 4: Strings – Rules” by typing the sample commands in the black area of the IDE.
   1. Explain why print(“Hello!”[7]) gives an error.

When you would write print(“Hello!”[7]) in Python it would give you an error because “Hello!” there is no index of 7 in it. Therefore, Python would give you an error because you trying to find a letter in a string that is not there.

**Level 2: Booleans & Variables**

Access the Tutorial and start at “Lesson 5: Variables”

Questions

1. Complete “Lesson 5: Variables – Save a Value” by typing the sample commands in the black area of the IDE.
   1. What do you get if you type puppies / 3?

When you type puppies / 3 you would get 12.

* 1. Why doesn’t typing kittens / 3 work?

When you type kittens / 3 it would give an error because you did not an assign a value to kittens. So, when you would type kittens / 3 there would nothing to divide by 3 due to that “kittens” is a variable that has no value assigned to it.

1. Complete “Lesson 5: Variables – Assign a New Value” by typing the sample commands in the black area of the IDE.
   1. Explain how the following sequence of commands works:
      * puppies = 36
      * puppies = puppies / 6
      * puppies

First you are assigning the variable puppies the value of 36. Then you are saying that the variable puppies is being changed to puppies with the old value of 36 to be divided by 6. Lastly when you would type puppies in Python it would give you the answer to 36 / 6 which is 6.

1. Read through “Lesson 5: Variables – Rules”.
2. Complete “Lesson 5: Variables – Math Operators” by typing the sample commands in the black area of the IDE.
   1. Explain what happens for following sequence of commands:
      * colour = “red”
      * puppies = 36
      * colour + puppies

First you are assigning the variable colour a value which is “red”. Then you are assigning the variable puppies a value which is 36. Lastly you are telling Python to do “red” + 36 because the variable for “red” is “colour” and the variable for “36” is “puppies”.

1. Complete “Lesson 5: Variables – String Operators” by typing the sample commands in the black area of the IDE.
   1. Explain why the following commands give different results:
      * Color + day \* fishes
      * ( Color + day ) \* fishes

Color + day \* fishes would give a different result than ( Color + day ) \* fishes because in Color + day \* fishes you are multiplying by fishes first due to order of operations and then it would do the addition. While in ( Color + day ) \* fishes it is doing the addition first because ( color + day )” has parentheses around them and in order of operations you would do the parentheses before multiplication.

1. Complete “Lesson 5: Variables – Indexes” by typing the sample commands in the black area of the IDE.
   1. What is the index of ‘r’ in “watermelon”?

The index of ‘r’ is 4 in “watermelon”.

* 1. Write an expression using mynumber to return ‘r’

fruit[mynumber +1]

* r

1. Complete “Lesson 5: Variables – Assignments or Comparisons” by typing the sample commands in the black area of the IDE.
   1. What is the difference between “=” and “==”?

The difference between “=” and “==” is that in “=” you are assigning a value to a variable and in “==” you trying to find out if a value is equal to another value.

* 1. Create your own mnemonic to remember this difference.

For “=” I made a sentence which is “assigning you a number” because you are assigning a variable a value in “=”.

For “==” I made a sentence which is “compare one to another” because in “==” you are comparing two values.

1. Complete “Lesson 6: Errors – Examples” by typing the sample commands in the black area of the IDE.
   1. What doesn’t “friend” + 5 work?

“friend” + 5 does not work because you cannot add 5 to “friend”. Also, Python can not add objects which are different types.

* 1. What is the difference between int and str?

The difference between int and str is that int is an integer and str is a string. Another thing is that an integer would only be whole numbers and a string would be words, letters, symbols and punctuation.

1. Read through “Lesson 6: Errors – Parts of an Error Message”.
   1. Is “friend” + 5 an example of:
      1. A Syntax Error?
      2. A Runtime Error?
      3. A Logic Error?

“friend” + 5 would be an example of a syntax error.

1. Read through “Lesson 6: Errors – Fixing Errors”.
   1. Use the ‘print’ command to print your first name and last name.

print(“Dylan, Kokilpersaud”)

1. Complete “Lesson 7: Booleans – Types of Data” by typing the sample commands in the black area of the IDE.
   1. What is the value of: type(“True”)

It is < class ‘str’ >

* 1. What is the value of: type( True )

It is < class ‘bool’ >.

* 1. Why is the result different?

In type(“True”) it had quotation marks around “true” and this would indicate that it is a string. However, in type( True ) there is no quotation marks around “true” so, Python will not classify it as a string.

1. Complete “Lesson 7: Booleans – What Is A Boolean” by typing the sample commands in the black area of the IDE.
   1. Why do you think that having a Boolean data type is important in computer programming?

It is important to have a Boolean data type in computer programming because in computer programming there is a lot of programming where there can be a true statement and a false statements.

1. Complete “Lesson 7: Booleans – Trying Out Booleans” by typing the sample commands in the black area of the IDE.
   1. Why do you think that there is no Maybe” Boolean data value in computer programming?

I think that there is no “maybe” in Boolean data because if there is a “maybe”, Python would have a hard time trying to determine if the data is true or false. Since there is no “maybe” data, it would make it easier for Python to determine if data is “true” or “false”. Also, most of the time data is either “true” or “false”.

**Level 3: Lists & Logic**

Access the Tutorial and start at “Lesson 7: Booleans”

Questions

1. Complete “Lesson 7: Booleans – AND Comparisons” by typing the sample commands in the black area of the IDE.
   1. Try the following Python statements and record the results.
      1. True and True

It is true.

* + 1. True and False

It is false.

* + 1. False and True

It is false.

* + 1. False and False

It is false.

* 1. Explain if there are any other combinations of True / False.

The other combinations of True / False would be if there is more than one true and more than one false. For example, one combination can be True and True and False, the answer to this would be false.

* 1. Explain how the AND operator is similar to a math operator and how it is different.

It way it is similar to a math operator is that both operators would compare values. The difference is that an AND operator would use the word “and” to compare and a math operator would use a symbol.

1. Complete “Lesson 7: Booleans – OR Comparisons” by typing the sample commands in the black area of the IDE.
   1. Try the following Python statements and record the results.
      1. True or True

It is true.

* + 1. True or False

It is true.

* + 1. False or True

It is true.

* + 1. False or False

It is false.

* 1. Explain how the OR operator is similar to the AND operator and how it is different.

Both operators would compare “true” and “false” and the difference between them is that in a OR operator if there is one “true”, the whole statement is true. While in an AND operator if there is one “false”, the whole statement is false.

1. Complete “Lesson 7: Booleans – NOT Comparisons” by typing the sample commands in the black area of the IDE.
   1. Try the following Python statements and record the results.
      1. not (True or True)

It is false.

* + 1. not (True or False)

it is false.

* + 1. not (False or True)

It is false.

* + 1. not (False or False)

It is true.

* 1. Explain how the combination of the NOT & OR operators is similar to the AND operator by itself and how it is different.

The NOT and OR operator combination would be similar to an AND operator because is that if there is one false in the combination, the answer would be false. Also, the difference would be that all the answer is the opposite of what they are supposed to be.

1. Complete “Lesson 7: Booleans – Expressions” by typing the sample commands in the black area of the IDE.
   1. Explain why the following two Python statements give different results.
      1. not (True or True)
      2. not True or True

The parentheses in the first statement around “True and True” would make both “trues” a false and it would give an answer of false. While in the second statement the “not” would make one of the “trues” false because there are no parentheses. So, the answer for the second statement would be true.

* 1. Explain why the following two Python statements give the same results.
     1. not (True and True)
     2. not True and True

It gave the same results for both statements because in AND operators if there is just one false, the answer will be false. Since the NOT operator made a false in each statement, the answers would be false.

1. Complete “Lesson 7: Booleans – Practice” by typing the sample commands in the black area of the IDE.
   1. Create three more practice expressions similar to those in the tutorial.
   2. Provide the results for your practice expressions

* 9 == 9 and 9 == 6+4

=>False

* "fish" == "fish"

=> True

* False or 2 == 2

=> True

1. Complete “Lesson 8: Lists – A Collection of Objects” by typing the sample commands in the black area of the IDE.
   1. Create a list of your favorite sports teams.

[“Blue Jays”, “Toronto Raptors”, Maple leaf’s”]

* 1. Assign your list to a variable.

Sports = [“Blue Jays”, “Toronto Raptors”, Maple leafs”]

* 1. Confirm that your variable and your list are the same.

Sports

=> [“Blue Jays”, “Toronto Raptors”, Maple leafs”]

1. Complete “Lesson 8: Lists – List Indexes” by typing the sample commands in the black area of the IDE.
   1. What is the list index of the last team in your list of favorite sports teams.

The list index of the last item in my favourite sports team list is 2.

* 1. In the tutorial, the error produced by typing “fruit[3]” is an example of:
     1. A Syntax Error?
     2. A Runtime Error?
     3. A Logic Error?

This is an example of a syntax error.

1. Complete “Lesson 8: Lists – Practice” and “Lesson 8: Lists – Practice Answers” by typing the sample commands in the black area of the IDE.

* colours = ["orange", "blue", "green"]
* print(colours[0])

=> orange

* print(colours[1])

=> blue

* print(colours[2])

=> green

NOTE: Starting with Lesson 9 you should use the WHITE area of the IDE for entering example code with multiple statements.

1. Complete “Lesson 9: Logic – Making Decisions” by typing the sample commands in the white area of the IDE.
   1. Modify the tutorial code to print “Hi Alfred!” based on a decision using numbers

This is my modified code

myname = ["Hi Alfred!", "Hi Steve!", "Hi Jeff!"]

if myname == ["Hi Alfred!", "Hi Steve!", "Hi Jeff!"]:

print(myname[0])

1. Complete “Lesson 9: Logic – Adding A Choice” by typing the sample commands in the white area of the IDE.
   1. Modify the tutorial code to print your first name or your last name based on a choice (using “else”).

This is the code that I modified

myname = "Dylan"

if myname == "Ginger":

print("Dylan")

else:

print("Kokilepersaud")

1. Complete “Lesson 9: Logic – Adding Many Choices” and “Lesson 9: Logic – Practice” by typing the sample commands in the white area of the IDE.
   1. Modify the tutorial code and “elif” statements to make a choice using at least 4 of your friends names.

This is the code that I modified

myname = "Harjot"

if myname == "Harjot":

print("Hi Harjot!")

elif myname == "Jordon":

print("Hi Jordon!")

elif myname == "Amrit":

print("Hi Amrit!")

elif myname == "Kevin":

print("Hi Kevin!")

else:

print("Nice to meet you")