**Mudule b.2**

**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.
   2. List your expression and the result below.

1+2+3+4+5+6

=> 21

10-9-8-7-6-5

=> -25

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.
   2. List your expression and the result below.

1\*2\*3\*4\*5\*6

=> 720

1/2/3/4/5/6

=> 0.001388888888888889

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
   2. And one division expression that gives a decimal number answer.
   3. List your expressions and the results below.

5+7

=> 12

3.4+2

=> 5.4

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(3.4+2)

=> 5

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

Two equal signs (==) are used when comparing two symbols or values to see if they are equal. i.e. a == c

* 1. What does “=” mean?

In the python = means equality as a command or assignment of a value. i.e. a = 5

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.

a. 10>4+3

=> True

11<10\*2

=> True

10>1\*8

=> True

20<10\*10

=> True

15<30-10

=> True

b.

10>15\*2

=> False

30>50

=> False

4>30-20

=> False

30==10

=> False

10==30+6

=> 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.

Typing apple without the quotes does not work because the rule of python is that the program can only understand a command if it is in quotations.

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

“2+5” does not equal 7 because it is in quotations. When doing a mathematical equation quotations are not needed but when dealing with characters such as symbol and letters quotations have to be used.

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.
   2. Also explain why “Hello” \* 10 works but why “Hello” / 10 does not work.

When adding the “e” to “appl” it works because the program is just adding a letter (e) to (appl) which the computer recognizes as a random string of letters. The reason why it does not work when subtracting the “e” from “apple” is because “apple” is seen as a random string of letters and the python program cant show a letter being subtracted from a word.

“hello” \*10 works because the program is just writing the word hello out ten times because that’s what the command was. “hello” / 10 does not work because the program cant divide the word hello into ten parts.

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.

Zakariya

Z=0, a=1, k=2, a=3, r=4, i=5, y=6, a=7

In python we start counting from zero not one.

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 print “I” because in python we start counting at zero and not number one, so the letter h is zero, and so on. The letter o has an index of 4.

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

When print(“Hay, Bob!”[4]) was put into the program it printed a space, so it basically printed nothing. This is because the fourth index is a space.

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.

print(“Hello!”[7]) gives an error because

**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 I typed puppies I got an error but when I typed puppies = 6\*6 I got nothing and after that I typed puppies and then it printed 36.

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

When I typed kittens I got an error but when I typed kittens = 6\*6 I got nothing and after that I typed kittens and then it printed 36.

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

When typing all of these values in python and after hitting enter it gives you a result of the number 6. Puppies = 36 means that I’m telling the program that there are 36 puppies. Puppies divided by 6 means 36 divided by 6. That is why it gives a result of 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
3. 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

These two commands will give you different answers because in the first command the program is doing the multiplication first and then adding and in the second command the program is adding first and then multiplying. This is because of the brackets around the color and day. When doing equations the rule is to always solve what’s in the brackets first.

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 the letter “r” in watermelon is 4.

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

fruit [mynumber-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 “==”?

One equal sign is used when you are assigning something. two equal signs are used when you are comparing values.

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

cats = "dogs"

cats == "dogs"

=> True

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 work does not work because Python cannot concatenate objects of different types. Friend is a string and 5 is a integer and python doesn’t know how to use them together because they are two different types of data.

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

Int stands for integer and str stands for string. In this case the number 5 is the integer and the word friend is the string.

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

print ("zak")

zak

print ("kalair")

kalair

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”)

The value of (“True”) is ‘true’

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

The value of (True) is True

* 1. Why is the result different?

The results are different because the first command has quotations around it and the second one does not.

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?

Having Boolean data type is important in computer programing because it determines weather something is true or false. Without Boolean data types codes would be inaccurate and wouldn’t make sense at all.

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?

There is no maybe in Boolean data value in computer programing because something has to be either true or false. For example in a game a move or a answer cant be a maybe its either right, wrong, 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
      2. True and False
      3. False and True
      4. False and False

i. True and True

=> True

ii. True and False

=> False

iii. False and True

=> False

iv. False and False

=> False

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

There are other combinations of true and false. This would happen if there is more then one true and false. For example, one combination can be two trues and one false. This would result in a false answer.

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

The AND operator is similar to a math operator because the end result is telling you if the statement including numbers is true or false. They are different because in the AND operator its only dealing with the two equal signs or weather it is true or false, and in a math operator it deals with all type of equations.

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
      2. True or False
      3. False or True
      4. False or False

True or True

=> True

True or False

=> True

False or True

=> True

False or False

=> False

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

The OR operator is similar to the AND operator because they both compare “true” and “false”. However there is a difference between these to operators. If there is one “true” in the OR operator then the whole statement becomes true. In the AND operator, if there is at least one “false” then 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)
      2. not (True or False)
      3. not (False or True)
      4. not (False or False)

not(True or True)

=> False

not (True or False)

=> False

not (False or True)

=> False

not (False or False)

=> 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 the OR operators is similar to the AND operator because if there is one false in the combination then the final statement would be false. Next, the difference is that all the answers is the opposite of what they are actually 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 first command gives a result of false and the second command gives a result of true. These two commands give different results because one of them has a set of brackets around them.

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

These two statements give the same result because in the AND operator if there is just one false then the whole statement would be false, but in this case they are both true which gives a result of true for both of them.

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

True and True

=> True

1 == 1 or 1 == 1

=> True

False and False

=> False

"Shark" == "Shark"

=> 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.

["Toronto Maple Leafs", "Golden State Warriors", "Toronto Blue Jays"]

* 1. Assign your list to a variable.

Favourite sports teams = ["Toronto Maple Leafs", "Golden State Warriors", "Toronto Blue Jays"]

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

Favourite sports teams

=> ["Toronto Maple Leafs", "Golden State Warriors", "Toronto Blue Jays"]

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 team in my list of favourite teams is 2. This is because python starts counting items at zero. Therefor Toronto Maple Leafs was zero, Golden State Warriors was one and the Toronto Blue Jays was number 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?

In this tutorial, the error produced by typing “fruit[3]” 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.

colors = ["blue", "red", "green"]

print(colors[0])

blue

print(colors[1])

red

print(colors[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

myname = ["Hi Alfred", "Hi Zak", "Hello Anikat"]

if myname == ["Hi Alfred", "Hi Zak", "Hello Anikat"]:

print(myname[0])

The result of this code in the black area is Hi Alfred

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”).

myname = "Zakariya"

if myname == "Ginger":

print("Zakariya")

else:

print("Zakariya")

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.

myname = "Dylan"

if myname == "Dylan"

print ("Hi Dylan")

elif myname == "Anikat"

print("Hello Anikat")

elif myname == "Nabeel"

print ("Hello Nabeel")

elif myname == "Kareem"

print("Hello Kareem")

else:

print("How are you my friends?")