**Python Basic Assignment 2 - Submitted by Shankar Eaga / Full Stack Data Science course**

**Answers captured in red color**

1. What are the Boolean data type's two values? How do you go about writing them?

Answer: The Boolean data types : True and False. There multiple ways of writing Boolean outputs ,

1. If you need to know expression is returning True or False
2. If you need to compare two values and return True or False.
3. If you run a condition in an if statement, Python returns True or False.

Examples :

Print(10>9) Returns True

Print(10<9) Returns False

2. What are the three different types of Boolean operators?

Answer : There are three types of Boolean or Logical operators in Python program,

1. And
2. Or
3. Not

Operators are used to perform operations with variables and values.

3. Make a list of each Boolean operator's truth tables (i.e. every possible combination of Boolean values for the operator and what it evaluate ).

Answer :

**Operator - Description Example**

and Returns True if both condition statements are True x < 5 and x < 10

and Returns False if anyone condition statement is False x < 5 and x < 10

or Returns True if one or both condition statements are True x < 5 or x < 10

or Returns False if both condition statements are False x < 5 or x < 10

not Reverse the result, returns True if condition statement is False not (True)

not Reverse the result, returns False if condition statement is True not(False)

4. What are the values of the following expressions?

(5 > 4) and (3 == 5) Answer : False

not (5 > 4) Answer : False

(5 > 4) or (3 == 5) Answer : True

not ((5 > 4) or (3 == 5)) Answer : False

(True and True) and (True == False) Answer : False

(not False) or (not True) Answer : True

5. What are the six different types of reference operators?

Answer : Apology I have not come across any reference operators in the Python but generally available operators in Python are as follows,

1. Arithmetic operators
2. Assignment operators
3. Comparison operators
4. Logical operators
5. Identity operators
6. Membership operators
7. Bitwise operators

6. How do you tell the difference between the equal to and assignment operators?

Answer : For example

x = 5 and

x = x+5,

1. In the first statement, it is true in both mathematics and computer programming language. x value is equal 5 which is correct and value 5 is assigned to variable x which is also correct as per python programming language.
2. In the second statement, it is not correct as per mathematics but in computer programming, it is a correct statement (expression will be evaluated first and then result will be assigned to x).

7. Describe a condition and when you would use one.

Answer : The below logical conditions are used in Python program,

* Equals : a ==b
* Not Equals : a != b
* Less than : a < b
* Less than or equal : a <= b
* Greater than : a > b
* Greater than or equal : a > = b

These conditions can be used in many ways but most commonly used in “if statements ” and loops

* If statement is written by using “if” keyword

a = 10,

b = 20

if a < b:

print(“The value of” + str(a)+” is less than”+ str(b)”)

else:

print(“The value of” + str(a)+” is greater than”+ str(b)”)

In the above example, a and b are two variables , which are used as part of the if statement to test whether b is greater than a. As a is 10, and b is 20, we know that 20 is greater than 10, and so controller will print to screen that "The value of a is less than b".

8. Recognize the following three blocks in this code:

Answer : See below for correct code

spam = 0

if spam == 10:

print('eggs')

if spam > 5:

print('bacon')

else:

print('ham')

print('spam')

print('spam')

The output from above code : print spam last statement as if condition outcome is False.

9. Create a programme that prints. If 1 is stored in spam, prints Hello; if 2 is stored in spam, prints Howdy; and if 3 is stored in spam, prints Salutations! if there's something else in spam.

Answer : See below code,

spam = int(input("Enter the integer number : "))

if spam == 1:

print("Hello")

elif spam == 2:

print("Howdy")

elif spam == 3:

print("Salutations!")

else:

print("Spam")

10.If your programme is stuck in an endless loop, what keys can you press?

Answer: Press Control + C on the keyboard, the loop will terminate.

11. How can you tell the difference between break and continue?

Answer : In Python break and continue statements can alter the flow of normal loop. Loops will iterate a block of code until the expression result is false. But sometime if we wish to terminate the current iteration or even whole loop without checking the test expression. The break and continue statements are used in these cases. A simple illustration to show how break and continue statement can be used in the

program,

Code starts here,

a = 10

b = 20

if a > b:

print(“False”)

break

statement1… # the below statements would not executed by Python interpreter as

there is a break statement

statement2…

else:

print(“True”)

continue

statement3… # the below statements would be executed by Python interpreter as

there is a continue statement

statement4..

12. In a for loop, what is the difference between range(10), range(0, 10), and range(0, 10, 1)?

Answer : A typical for loop statement with range function

for i in range(10): (or) for i in range(0,10): (or) for i in range(0,10,1):

print(i) print(i) print(i)

from the above three “if” statements, the output is same such as 0 to 9. The main difference between them is that in the second statement, starting and ending values in the range defined and in third statement starting, ending and increment value are defined. Otherwise starting value and increment values in the first and increment value in the second statements are implicitly mentioned.

13. Using a for loop, write a short programme that prints the numbers 1 to 10 Then, using a while loop, create an identical programme that prints the numbers 1 to 10.

Answer :

For loop:

for i in range(1,11):

print(i)

while loop:

i = 1

while i <11:

print(i)

i+=1

14. If you had a bacon() function within a spam module, what would you call it after importing spam?

Answer :

A function is a block of organized, reusable code that is used to perform a single, related action. Functions provide better modularity for your application and a high degree of code reusing.

Python gives you many built-in functions like print(), etc. but you can also create your own functions. These functions are called user-defined functions.

# user defined function

def bacon\_user(str):

print(str)

return:

# Call user defined function

bacon\_user(“this is user defined function 1”)

bacon\_user(“this is user defined function 2”)

Python Module :

A module allows you to logically organize your Python code. Grouping related code into a module makes the code easier to understand and use. A module is a Python object with arbitrarily named attributes that you can bind and reference.

Simply, a module is a file consisting of Python code. A module can define functions, classes and variables. A module can also include runnable code, example example.py

# example.py module

def print\_function(str):

print(“this is part of module”, str)

return:

# Need to import example module in the code at the top of the script,

# import example module

import example

# now you can call that defined function that module as follows

example.print\_function(“Demonstration of importing user defined module and function”)