ASSIGNMENT - 2

1. What are the two values of the boolean data types? how do you write them ?

Ans: True and False are two values of the boolen data types.

a = True

b = False

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Ans: *The three different types of Boolean operators in python are: or and not.*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

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) ?

Ans: *The Truth tables for the boolean tables are as follows:*

*Truth Table for AND operator*

| *Input(A)* | *Input(B)* | *Output(A.B)* |
| --- | --- | --- |
| *True* | *True* | *True* |
| *True* | *False* | *False* |
| *False* | *True* | *False* |
| *False* | *False* | *False* |

*Truth Table for OR operator*

| *Input(A)* | *Input(B)* | *Output(A+B)* |
| --- | --- | --- |
| *True* | *True* | *True* |
| *True* | *False* | *True* |
| *False* | *True* | *True* |
| *False* | *False* | *False* |

*Truth Table for NOT operator*

| *Input(A)* | *Output (A not)* |
| --- | --- |
| *True* | *False* |
| *False* | *True* |

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

4. What are the values of the following expressions ?

(5 > 4) and (3 == 5)

not (5 > 4)

(5 > 4) or (3 == 5)

not ((5 > 4) or (3 == 5))

(True and True) and (True == False)

(not False) or (not True)

Ans: *print((5>4)and(3==5)) # False*

*print(not(5>4)) # False*

*print((5>4)or(3==5)) # True*

*print(not((5>4)or(3==5))) # False*

*print((True and True)and(True==False)) # False*

*print((not False)or(not True)) # True*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

5. What are the six comparison operators?

Ans: *The Six comparison operators available in python are:*

*== , != , < , > , <= , =>*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

6. How do you tell the difference between the equal to and assignment operators?Describe a condition and when you would use one ?

Ans: *== is the equal to operator that compares two values and evaluates to a Boolean, while = is that assignment operator that stores a value in a variable.*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

7. Identify the three blocks in this code:

spam = 0

if spam == 10:

print('eggs')

if spam > 5:

print('bacon')

else:

print('ham')

print('spam')

print('spam')

Ans: spam = 0

if spam == 10:

print('eggs') # block #1

if spam > 5:

print('bacon') # block #2

else:

print('ham') # block #3

print('spam')

print('spam')

Output :-

ham

spam

spam

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

8. Write code that prints Hello if 1 is stored in spam, prints Howdy if 2 is stored in spam, and prints Greetings! if anything else is stored in spam.

Ans:

*def spamCode(spam):*

*if spam==1:*

*print('Hello')*

*elif spam==2:*

*print('Howdy')*

*else:*

*print('Greetings')*

*spamCode(1)*

*spamCode(2)*

*spamCode(3)*

*Output:*

*Hello*

*Howdy*

*Greetings*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

9.If your programme is stuck in an endless loop, what keys you’ll press?

Ans: *Press Ctrl-c to stop a program stuck in an infinite loop*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

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

Ans: *The break statement will move the execution outside the loop if break condtion is satisfied. Whereas the continue statement will move the execution to the start of the loop.*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

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

Ans: *The Differences are as follows:*

* *The range(10) call range from 0 to 9 (but not include 10)*
* *The range (0,10) explicitly tells the loop to start at 0*
* *The range(0,10,1) explicitly tells the loop to increase the variable by 1 on each iteration.*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

12. Write a short program that prints the numbers 1 to 10 using a for loop. Then write an equivalent program that prints the numbers 1 to 10 using a while loop ?

Ans:

*#Using for loop*

*print('-'\*10,'Using For Loop','-'\*10)*

*for i in range(1,11):*

*print(i, end=" ")*

*print('\n')*

*#Using while loop*

*print('-'\*10,'Using While Loop','-'\*10)*

*i=1*

*while i<=10:*

*print(i, end=" ")*

*i+=1*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

13. If you had a function named bacon() inside a module named spam, how would you call it after importing spam ?

Ans: *This function can be called with spam.bacon()*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*