1. What are the two values of the Boolean data type? How do you write them?

ANS: TRUE and FALSE

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

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

ANS : 1) AND 2) OR 3) NOT

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

1. 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: There are true and false, based on this values we can create truth table for boolean operators as follow:

True ~ T

False ~ F

Truth table of **AND** operator:

|  |  |  |
| --- | --- | --- |
| Input 1 | Input 2 | Input 1 **AND** Input 2 |
| T | T | T |
| T | F | F |
| F | T | F |
| F | F | F |

Truth table of **OR** operator:

|  |  |  |
| --- | --- | --- |
| Input 1 | Input 2 | Input 1 **OR** Input 2 |
| T | T | T |
| T | F | T |
| F | T | T |
| F | F | F |

Truth table of **NOT** operator:

|  |  |
| --- | --- |
| **Input** | **Output** |
| **T** | **F** |
| **F** | **T** |

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

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 :

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

not (5 > 4) = False

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

not ((5 > 4) or (3 == 5)) =Not(True) = False

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

(not False) or (not True) = True OR False = True

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

5.What are the six comparison operators?

ANS :  
the six comparison operators are

**less than ( < )**

**greater than ( > )**

**less than or equal to ( < = )**

**greater than or equal to ( > =)**

**equal to ( == )**

**and not equal to ( != )**

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

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

ANS: The assignment operator “=” is used to assign the value on the right to the variable on the left.

Eg.

A=10

B=”Python”  
c = 33

The “==” operator checks whether the given two operands are equal or not , if so it returns True , otherwise it ill return False.

Eg.

A= 5

B= 5

A==B

It will return True.

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

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:

**The 1st block is:**

if spam == 10:

print('eggs')

**The 2nd block is:**

if spam > 5:

print('bacon')

**The 3rd block is:**

else:

print('ham')

print('spam')

print('spam')

And it will return the output as:

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:

spam =”2”

if spam ==”1”:

print(“Hello”)

elif spam ==”2”:

print(“Howdy ”)

else:

print(“Greetings!”)

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

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

ANS: Ctrl + c

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

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

ANS :

Both “break” and “continue” are the ****‘jump’ statements****, that transfer control of the program to another part of the program.

The main difference between break and continue is that break is used for immediate termination of loop.

The break statement terminates the loop containing it. Control of the program flows to the statement immediately after the body of the loop.

If the break statement is inside a nested loop (loop inside another loop), the break statement will terminate the innermost loop.

On the other hand, ‘continue’ terminate the current iteration and resumes the control to the next iteration of the loop.

The continue statement is used to skip the rest of the code inside a loop for the current iteration only. Loop does not terminate but continues on with the next iteration.

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

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

ANS:

There is no difference in range(10), range(0, 10), and range(0, 10, 1)

If we execute

for I in range(10):

Print(i)

OR

for I in range(0,10):

Print(i)

OR

for I in range(0,10,1):

Print(i)

The output will be same.

i.e.

0

1

2

3

4

5

6

7

8

9

Because range(10): means it will start the range from(default start point) 0 and it will stop at 9 and the default step size is 1 , hence it will it be 0 1 2 3 4 5 6 7 8 9

And range(0,10) will also be same because it has defined its starting range whcih is 0 and again the default step size is 1.

And (0,10,1) here it has mentioned its starting number, ending range and the default step size which is similar as range (10) and range(0,10)

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

1. 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 :

for i in range(1,11):

Print(i)

USING WHILE LOOP:

i =1

while i < 11:

print(i)

i+=1

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