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

Solution:

True and False are the two values of the Boolean data type.

Example:

```
a = True

b = False

print(type(a))

print(type(b))
```

Output:

```
<class 'bool'>

<class 'bool'>
```

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

Solution:

AND, OR, NOT are the three different types of Boolean Operators.

Example:

```
a = 100

b = 300

print(a<b and b>a)

print(a<b or b>a)

print(not(b>a))
```

Output:

```
True

True

False
```

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).

Solution

Truth table for “AND” Operator

True and True is True

True and False is False

False and True is False

False and False is False

Truth table for “OR” Operator

True or True is True

True or False is True

False or True is True

False or False is True

Truth table for “NOT” Operator

not(True) is False

not(False) is 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)

Solution:

1.(5 > 4) and (3 == 5)

Output: False

2. not (5 > 4)

Output: False

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

Output: True

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

Output: False

5. (True and True) and (True == False)

Output: False

6. (not False) or (not True)

Output: True

5. What are the six comparison operators?

Solution:

==, !=, <, >, <=, >= this are the six comparison operators.

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

Solution:

Assignment Operator(=)

If we want to assign value to a variable at that time we use Assignment Operator (=)

Example:

```
a = 5   # Assignment Operator

print(a)
```

output: 5

Relational Operator(==)

A relational operator is used to compare two variables or constants (==)

Example:

```
a = 5

b = 5

print(a==b) #Relational Operator
```

Output: 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')
```

Solution:

```
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')
```

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.

Input:

```
def spam_function(spam):
    if (spam==1):
        print("Hello")
    elif (spam==2):
        print("Howdy")
    else:
        print('Greetings')
spam_function(1)
spam_function(2)
spam_function(3)
```

Output:

Hello

Howdy

Greetings

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

Solution:

Press ctrl - C if our program is stuck in endless loop.

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

Solution:

Break

break is used to exit while loop and for loop when certain conditions get satisfied.

Continue

continue statement will just bypass the current iteration and continue with the next iteration.

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

Solution:

range(10): call range from 0 to 9 (but not include 10)

range(0,10): explicitly tells the loop to start at 0

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.

Using For Loop

```
for i in range(1,11):
    print(i)
```

Output:

1

2

3

4

5

6

7

8

9

10

Using While Loop

```
i=1
while(i<11):
  print(i)
  i=i+1
```

Output:

1

2

3

4

5

6

7

8

9

10

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

Solution:

This function can be called with spam.bacon()