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

Answer: True and False are the values of Boolean data type. This data type can be used to evaluate a statement to true or false

Eg: (10>5) returns True

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

Answer: AND, OR, 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 ).

Answer:

l1=[True, True, False, False ]

l2=[True, False, True, False]

AND operation:l1 and l2

output: [True, False, True, False]

AND operator Truth table

|  |  |  |
| --- | --- | --- |
| A | B | Result |
| True | True | True |
| True | False | False |
| False | True | False |
| False | False | False |

OR operation: l1 or l2

Output: [True, True, False, False]

OR operator truth table:

|  |  |  |
| --- | --- | --- |
| A | B | Result |
| True | True | True |
| True | False | True |
| False | True | True |
| False | False | False |

Not operation: l1=[True]

not l1

output: False

NOT operator Truth table:

|  |  |
| --- | --- |
| A | Result |
| True | False |
| False | True |

4. What are the values of the following expressions?

Answer:

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

not (5 > 4): True

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

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

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

(not False) or (not True): True

5. What are the six comparison operators?

Answer:

1. Greater than >
2. Greater than or equal to >=
3. Less than <
4. Less than or equal to <=
5. Equal to ==
6. Not equal to !=

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

Answer: == operator is used to compare two operands value and = is the assignment operator which is used to assign a value to an operand

Eg:

a=10

b=10

#assigned value 10 to a and b)

print(a==b) #comparing a and b values using == operator

output: True

7. Identify the three blocks in this code:

Answer:

spam = 0

#block 1, where we print eggs when spam==10

if spam == 10:

print('eggs')

#block 2, where we print bacon if spam >5

if spam > 5:

print('bacon')

#block 3, where we print ham, spam and spam if above two blocks are not satisfied and all other cases

else:

print('ham')

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.

Answer:

def spamcheck(spam):

if spam=1:

print(“Hello”)

if spam=2:

print(“Howdy”)

else:

print(“Greetings!”)

spamcheck(1)

output: Hello

spamcheck(2)

output: Howdy

spamcheck(5)

output: Greetings!

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

Answer: Ctrl+C

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

Answer:

**break** keyword can be used to break a loop at the current iteration by specifying a condition

eg:

a=1

while a<10:

print(“Hello”)

a=a+1

if a==3:

break

Above code prints Hello twice as we are breaking loop at 3rd iteration

**continue** keyword can skip the current iteration and move to the next iteration by putting an if condition of that iteration which we want to skip

a=1

while a<5:

print(“Hello”)

a=a+1

if a==3:

continue

Above code prints Hello 4 times as we skipped one iteration at a==3

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

Answer:

range(10): returns sequence of numbers starting from 0 till 10(10 is not included)

range(0,10): same as range(10) but the first param can be used to mention the start point and second param as the stop point

range(0, 10, 1): returns the sequence of numbers starting from 0 till 10(10 is not included) with a step size of 1

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.

Answer:

**For loop:**

for i in range(1, 11):

print(i)

**While loop:**

i=1

while(i<11):

print(i)

i=i+1

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

Answer:

import spam

My\_bacon=spam.bacon()