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

🡪 A variable of the primitive data type boolean can have two values: true and false.

The **bool()** function allows you to evaluate any value, and give you True or False in return

print(bool("Hello"))

print(bool(15))

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

🡪 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 ).

|  |  |  |
| --- | --- | --- |
| X | Y | X AND Y |
| FALSE | FALSE | FALSE |
| TRUE | FALSE | FALSE |
| FALSE | TRUE | FALSE |
| TRUE | TRUE | TRUE |

|  |  |  |
| --- | --- | --- |
| X | Y | X OR Y |
| FALSE | FALSE | FALSE |
| TRUE | FALSE | TRUE |
| FALSE | TRUE | TRUE |
| TRUE | TRUE | TRUE |

|  |  |
| --- | --- |
| X | NOT X |
| FALSE | TRUE |
| TRUE | FALSE |

4. What are the values of the following expressions?

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

* print((5 > 4) and (3 == 5)) 🡪 False

not (5 > 4)

* print(not (5 > 4)) 🡪 False

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

🡪print((5 > 4) or (3 == 5)) 🡪 True

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

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

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

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

(not False) or (not True)

🡪 print((not False) or (not True)) 🡪 True

5. What are the six comparison operators?

| Operator | Title |
| --- | --- |
| == | Equal to |
| != | Not equal to |
| < | Less than |
| > | Greater than |
| <= | Less than or equal to |
| >= | Greater than or 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.

🡪 In mathematics “ = “ is an equal to operator. In programming “ = “ is an assignment operator, which means that it assigns a value to a variable.

The following code will store the value,

X=5

And x = x + 4; 🡪 in that code 🡪 right hand side code will execute and store the value in x

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')
* if spam > 5:
* print('bacon')
* 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.

-🡪

spam=input()

if spam =='1':

print("Hellow")

elif spam =='2':

print("Howdy")

else:

print("Greetings!")

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

🡪 CTRL+C

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

🡪 A **break** statement is used to terminate the loop whenever a particular condition is satisfied.

The **continue** statement skips the remaining lines of code, for the current iteration of the loop. In this case, the loop does not end, it continues with the next iteration.

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

🡪

for i in range(10):

print(i, end=" ")

print()

Ans - 0 1 2 3 4 5 6 7 8 9

🡪

for i in range(0,10):

print(i, end=" ")

print()

Ans - 0 1 2 3 4 5 6 7 8 9

🡪

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

print(i, end=" ")

print()

Ans - 0 1 2 3 4 5 6 7 8 9

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.

🡪

for i in range(1,11):

print(i)

🡪 i=1

while (i<=10):

print(i)

i+=1

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

🡪 This function can be called with spam. bacon().