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

Answer:

The Python Boolean type has only two possible values:

* True.
* False.

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

Answer:

The three different types of Boolean operators are **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:

AND truth table:

|  |  |  |  |
| --- | --- | --- | --- |
| X | and | Y | Returns |
| True | and | True | True |
| True | and | False | False |
| False | and | True | False |
| False | and | False | False |

OR truth table:

|  |  |  |  |
| --- | --- | --- | --- |
| X | or | Y | Returns |
| True | or | True | True |
| True | or | False | True |
| False | or | True | True |
| False | or | False | False |

NOT truth table:

|  |  |  |
| --- | --- | --- |
| not | X | Returns |
| not | True | False |
| not | 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)

Answer:

**(5 > 4) and (3 == 5)** 🡪 False

**not (5 > 4)** 🡪 False

**(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:

The six comparison operators are **>, <,>=, <=, ==,!=**.

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

Answer:

The **“=”** is an assignment operator isused to assign the value on the right to the variable on the left , where has ,the '**=='** operator checks whether the two given operands are equal or not , it returns Boolean value.

a=10 # assigning value 10 to variable a

if a==10: #checking whether a value is 10 or not if it is 10 returns True and prints ‘a’ value

print(a)

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')

Answer:

spam = 0

if spam == 10:

print('eggs') if block

if spam > 5:

print('bacon') if block

else:

print('ham')

print('spam') else block

print('spam')

It contains 2 if blocks and one else block , as the value of spam is 0 else block gets executed.

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:

spam=int(input(“Enter spam value:”))

if spam==1:

print(“Hello”)

elif spam==2:

print(“Howdy”)

else:

print(“Greetings”)

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

Answer:

Press **CTRL+C** to force exit the endless(infinite) loop.

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

Answer:

The break statement terminates whole iteration of a loop whereas continue skips the current iteration.

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

Answer:

There is no difference between range(10), range(0, 10), and range(0, 10, 1) in for loop.

l1=[]

l2=[]

l3=[]

for i in range(10):

   l1.append(i)

for i in range(0,10):

   l2.append(i)

for i in range(0,10,1):

   l3.append(i)

print(l1,l2,l3)

output:

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9] [0, 1, 2, 3, 4, 5, 6, 7, 8, 9] [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.

Answer:

l1=[]

l2=[]

for i in range(1,11):

   l1.append(i)

k=1

while(k<=10):

   l2.append(k)

   k=k+1

print(l1,l2)

output:

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10] [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?

Answer:

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