1. What possible values can a Boolean expression have?

Arithmetic expressions evaluate to numeric values; a Boolean expression, sometimes called a predicate,

may have only one of two possible values: false or true.

2. Where does the term Boolean originate?

The term Boolean comes from the name of the British mathematician George Boole. A branch of discrete mathematics called Boolean algebra is dedicated to the study of the properties and the manipulation of logical expressions.

3. What is an integer equivalent to True in Python?

Python considers the integer value one to be true.

4. What is the integer equivalent to False in Python?

. Python considers the integer value zero to be false

5. Is the value -16 interpreted as True or False?

No

6. Given the following definitions:

x, y, z = 3, 5, 7

evaluate the following Boolean expressions:

(a) x == 3 🡺 print(x == 3 ) 🡺true

(b) x < y 🡺 true

(c) x >= y 🡺 false

(d) x <= y 🡺 true

(e) x != y - 2 🡺 false

(f) x < 10 🡺 true

(g) x >= 0 and x < 10 🡺 true

(h) x < 0 and x < 10 🡺 false

(i) x >= 0 and x < 2 🡺 false

(j) x < 0 or x < 10 🡺 true

(k) x > 0 or x < 10 🡺 true

(l) x < 0 or x > 10 🡺 false

7. Given the following definitions:

x, y = 3, 5 b1, b2, b3, b4 = True, False, x == 3, y < 3

evaluate the following Boolean expressions:

(a) b3 🡺 true

(b) b4 🡺false

(c) not b1🡺 false

(d) not b2 🡺true

(e) not b3 🡺false

(f) not b4 🡺true

(g) b1 and b2 🡺false

(h) b1 or b2 🡺true

(i) b1 and b3 🡺true

(j) b1 or b3 🡺true

(k) b1 and b4 🡺false

(l) b1 or b4 🡺true

(m) b2 and b3🡺 false

(n) b2 or b3 🡺false

(o) b1 and b2 or b3🡺true

(p) b1 or b2 and b3 🡺true

(q) b1 and b2 and b3 🡺false

(r) b1 or b2 or b3🡺 true

(s) not b1 and b2 and b3🡺false

(t) not b1 or b2 or b3 🡺true

(u) not (b1 and b2 and b3) 🡺false

(v) not (b1 or b2 or b3) 🡺true

(w) not b1 and not b2 and not b3🡺false

(x) not b1 or not b2 or not b3 🡺true

(y) not (not b1 and not b2 and not b3) 🡺true

(z) not (not b1 or not b2 or not b3) 🡺false

8. Express the following Boolean expressions in simpler form; that is, use fewer operators or fewer symbols. x is an integer.

(a) not (x == 2) 🡺 x!=0

(b) x < 2 or x == 2 🡺 x<=2

(c) not (x < y) 🡺 x>y

(d) not (x <= y) 🡺 x>=y

(e) x < 10 and x > 20 🡺 10<x<20

(f) x > 10 or x < 20 🡺

(g) x != 0 🡺 x!=0

(h) x == 0🡺 x==0

9. Express the following Boolean expressions in an equivalent form without the not operator. x and y are integers.

(a) not (x == y) 🡺 x!=y

(b) not (x > y) 🡺 x<y

(c) not (x < y) 🡺 x>y

(d) not (x >= y) 🡺 x<y

(e) not (x <= y) 🡺 x<y

(f) not (x != y) 🡺 x==y

(g) not (x != y) 🡺 x==y

(h) not (x == y and x < 2) 🡺 x=!y and x>2

(i) not (x == y or x < 2) 🡺 x!=y or x>2

(j) not (not (x == y)) 🡺 x==y

10. What is the simplest tautology?

f it's more about the outcome, a tautology simply means, it's always true. So **"True" (TRUE, true, 1 or whatever, depending on language or field) would be the simplest tautology value wise**.

11. What is the simplest contradiction?

"False" would be the simplest contradiction by the opposite line of reasoning.

12. Write a Python program that requests an integer value from the user. If the value is between 1 and 100 inclusive, print ”OK;” otherwise, do not print anything.

x=int(input("please enter a int number"))

if 1<=x and x<=100 :

print("oK")

else :

print("")

13. Write a Python program that requests an integer value from the user. If the value is between 1 and 100 inclusive, print ”OK;” otherwise, print ”Out of range.”

x=int(input("please enter a int number"))

if 1<=x and x<=100 :

print("oK")

else :

print("out of range")

14. Write a Python program that allows a user to type in an English day of the week (Sunday, Monday, etc.). The program should print the Spanish equivalent, if possible.

x=input("pleas enter a weekday")

if x=="sunday":

print("domingo")

else :

if x=="monday":

print("lunes")

else :

if x=="tuesday":

print("martes")

else :

if x=="wednesday":

print("miércoles")

else :

if x=="thursday":

print("jueves")

else :

if x=="friday":

print("viernes")

else :

if x=="saturday":

print("sábado")

else :

print("out of range")

15. Consider the following Python code fragment:

# i, j, and k are numbers

if i < j:

if j < k

i = j

else: j = k

else:

if j > k:

j = i

else: i = k

print("i =", i, " j =", j, " k =", k)

What will the code print if the variables i, j, and k have the following values?

(a) i is 3, j is 5, and k is 7 🡺 i = 5 j = 5 k = 7

(b) i is 3, j is 7, and k is 5 🡺 i = 3 j = 5 k = 5

(c) i is 5, j is 3, and k is 7 🡺i = 7 j = 3 k = 7

(d) i is 5, j is 7, and k is 3 🡺i = 5 j = 3 k = 3

(e) i is 7, j is 3, and k is 5🡺i = 5 j = 3 k = 5

(f) i is 7, j is 5, and k is 3🡺i = 7 j = 7 k = 3

16. Consider the following Python program that prints one line of text:

val = int(input())

if val < 10:

if val != 5:

print("wow ", end='')

else:

val += 1

else:

if val == 17:

val += 10

else:

print("whoa ", end='')

print(val)

What will the program print if the user provides the following input?

(a) 3 🡺 wow 3

(b) 21 🡺 whoa 21

(c) 5 🡺6

(d) 17🡺 27

(e) -5🡺 wow -5

17. Consider the following two Python programs that appear very similar:

|  |  |
| --- | --- |
| n = int(input())  if n < 1000:  print('\*', end='')  if n < 100:  print('\*', end='')  if n < 10:  print('\*', end='')  if n < 1:  print('\*', end='')  print() | n = int(input())  if n < 1000:  print('\*', end='')  elif n < 100:  print('\*', end='')  elif n < 10:  print('\*', end='')  elif n < 1:  print('\*', end='')  print() |

How do the two programs react when the user provides the following inputs?

(a) 0 🡺 \*\*\*\* \*

(b) 1 🡺\*\*\* \*

(c) 5 🡺 \*\*\* \*

(d) 50 🡺 \*\* \*

(e) 500 🡺 \* \*

(f) 5000 🡺 nothing nothing

Why do the two programs behave as they do? The first program use if condition which is use when you have two condition and one of them is true and the other one is wrong but we use elif for print the third condition when the other condition is wrong.

18. Write a Python program that requests five integer values from the user. It then prints the maximum and minimum values entered. If the user enters the values 3, 2, 5, 0, and 1, the program would indicate that 5 is the maximum and 0 is the minimum. Your program should handle ties properly; for example, if the user enters 2, 4, 2, 3, and 3, the program should report 2 as the minimum and 4 as maximum.

a=int(input("please enter a number"))

b=int(input("please enter a number"))

c=int(input("please enter a number"))

d=int(input("please enter a number"))

e=int(input("please enter a number"))

print(max(a,b,c,d,e))

print(min(a,b,c,d,e))

19. Write a Python program that requests five integer values from the user. It then prints one of two things: if any of the values entered are duplicates, it prints "DUPLICATES"; otherwise, it prints "ALL UNIQUE".

a=int(input("please enter a number"))

b=int(input("please enter a number"))

c=int(input("please enter a number"))

d=int(input("please enter a number"))

e=int(input("please enter a number"))

if a==b or a==c or a==d or a==e:

print ("DUPLICATES")

else :

if b==c or b==d or b==e:

print("DUPLICATES")

else:

if c==d or c==e:

print("DUPLICATES")

else :

if d==e:

print("DUPLICATES")

else :

print("ALL UNIQUE")