

1.What possible values can a Boolean expression have?

True and false

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.

* اصطلاح بولی از نام جورج بول ریاضیدان بریتانیایی گرفته شده است. شاخه ای از ریاضیات گسسته به نام جبر بولی به مطالعه خواص و دستکاری عبارات منطقی اختصاص دارد .

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

It's number one . (true == 1) .

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

It's number zero . (false ==0).

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

#Part 1

```
# Get number from the user
value = int(input("Please enter an integer value in
the range 0...1023: "))
# Initial binary string is empty
binary_string = ""
# Integer must be less than 1024
if 0 <= value < 1024:
    binary_string += str(value//512)
    value %= 512
    binary_string += str(value//256)
    value %= 256
    binary_string += str(value//128)
    value %= 128
    binary_string += str(value//64)
    value %= 64
    binary_string += str(value//32)
    value %= 32
    binary_string += str(value//16)
    value %= 16
```

#Part 2

```
binary_string += str(value//8)
value %= 8
binary_string += str(value//4)
value %= 4
binary_string += str(value//2)
value %= 2
binary_string += str(value)
# Report results
if binary_string != "":
    print(binary_string)
else:
    print('Unable to convert')
```

RESULT :

```
Please enter an integer value in the range 0...1023: 16
0000010000
VALUE OF 16 : 0000010000
```

6. Given the following definitions:

$x, y, z = 3, 5, 7$

evaluate the following Boolean expressions:

(a) $x == 3$ true

(b) $x < y$ true

(c) $x >= y$ false

(d) $x <= y$ false

(e) $x != y - 2$ false

(f) $x < 10$ true

(g) $x >= 0$ and $x < 10$ false

(h) $x < 0$ and $x < 10$ true

(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: `true` `false`

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

(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)` `true`

(v) `not (b1 or b2 or b3)` `false`

(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) $\text{not } (x == 2)$ $x != 2$

(b) $x < 2 \text{ or } x == 2$ $x <= 2$

(c) $\text{not } (x < y)$ $x > y$

(d) $\text{not } (x <= y)$ $x >= y$

(e) $x < 10 \text{ and } x > 20$ false

(f) $x > 10 \text{ or } x < 20$ true

(g) $x != 0$ true

(h) $x == 0$ false

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

(a) $\text{not } (x == y)$ $x != y$

(b) $\text{not } (x > y)$ $x < y$

(c) $\text{not } (x < y)$ $x > y$

(d) $\text{not } (x >= y)$ $x <= y$

(e) $\text{not } (x <= y)$ $x >= y$

(f) $\text{not } (x != y)$ $x == y$

(g) $\text{not } (x != y)$ $x == y$

(h) $\text{not } (x == y \text{ and } x < 2)$ $y > 2$

(i) $\text{not } (x == y \text{ or } x < 2)$ $y != x > 2$

(j) $\text{not } (\text{not } (x == y))$ $x == y$

10. What is the simplest tautology?

A Boolean expression that is always true is known as a tautology.

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

* یک عبارت بولی که همیشه درست است به عنوان یک توتولوژی شناخته میشود.

اگر بیشتر در مورد نتیجه باشد، یک توتولوژی به سادگی به این معنی است که همیشه درست است. بنابراین «درست» (درست، درست، 1 یا هر چیز دیگری، بسته به زبان یا رشته) سادهترین ارزش توتولوژی خواهد بود، در حالی که «نادرست» سادهترین تناقض در خط استدلال مخالف است.

11. What is the simplest contradiction?

A statement that is always false is known as a contradiction.

* جمله ای که همیشه نادرست است به عنوان تناقض شناخته می شود.

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.

```
a = int(input("enter a number :"))
```

```
if 0 <= a <= 100 :
```

```
    print("ok")
```

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."

```
a = int(input("enter a number :"))
```

```
if 0 <= a <= 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.

```
a = str(input("Enter your desired day : "))
```

```
if a == "Saturday" :
```

```
    print(" Sábado ")
```

```
elif a == "Sunday " :
```

```
    print(" domingo ")
```

```
elif a == "Monday " :
```

```
    print(" lunes ")
```

```
elif a == "Tuesday " :
```

```
    print(" martes ")
```

```
elif a == "Wednesday " :
```

```
    print(" miércoles ")
```

```
elif a == "Thursday " :
```

```
    print(" Jueves ")
```

```
elif a == "Friday " :
```

```
    print(a("Viernes "))
```

```
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

(c) 5 line1 : 5 / line2 : 6

(d) 17 line1 : 17 / line2 : 27

(e) -5 wow 5

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

#Part 1

#Part 2

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

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

(a) 0 → 1) * / 2) ****

(b) 1 → 1) * / 2) ***

(c) 5 → 1) *** / 2) *

(d) 50 → 1) ** / 2) *

(e) 500 → 1) * / 2) *

(f) 5000 → 1) -nothing- / 2) -nothing-

Why do the two programs behave as they do?

* In the first part, all items are written with if, if the condition is true, it will print, but in the second part, we have elif, in such a situation, if if is not correct, we try elif, and it finds the correct item among elif and acts on it.

* در قسمت اول همه ی موارد با <<ایف>> نوشته شده است که اگر شرط درست باشد چاپ میشود اما در قسمت دوم <<ایف>> داریم، در چنین شرایطی اگر <ایف> صحیح نبود <ایف> را امتحان میکنیم و مورد درست را از میان <ایف> ها پیدا میکند و به آن عمل میکند .

If = ایف / elif = الیف

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.

```
n1 = int(input("num 1 --> "))
```

```
max = n1
```

```
min = n1
```

```
n2 = int(input("num 2 --> "))
```

```
n3 = int(input("num 3 --> "))
```

```
n4 = int(input("num 4 --> "))
```

```
n5 = int(input("num 5 --> "))
```

```
# Check minimum num
```

```
if n2 < min :
```

```
    min = n2
```

```
if n3 < min :
```

```
    min = n3
```

```
if n4 < min :
```

```
    min = n4
```

```
if n5 < min :
```

```
    min = n5
```

```
print("Min = ",min)
```

```
# Maximum check
```

```
if n2 > max :
```

```
    max = n2
```

```
if n3 > max :
```

```
    max = n3
```

```
if n4 > max :
```

```
    max = n4
```

```
if n5 > max :
```

```
max = n5  
print("Max = ",max)
```

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".

```
num1 = int(input("Please enter a number: "))  
num2 = int(input("Please enter a number: "))  
num3 = int(input("Please enter a number: "))  
num4 = int(input("Please enter a number: "))  
num5 = int(input("Please enter a number: "))  
  
if num1 == num2 or num1 == num3 or num1 == num4 or num1 == num5 or num2 == num3 or /  
num2 == num4 or num2 == num5 or num3 == num4 or num3 == num5 or num4 == num5:  
    print("DUPLICATES")  
else:  
    print("ALL UNIQUE")
```