

Fundamentals of **Python** **Programming**

Chapter 4

1. What possible values can a Boolean expression have? A boolean expression 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.
3. What is an integer equivalent to True in Python? 1
4. What is the integer equivalent to False in Python? 0
5. Is the value -16 interpreted as True or False? True
6. Given the following definitions:
x, y, z = 3, 5, 7
evaluate the following Boolean expressions:
 - (a) x == 3
 - (b) x < y
 - (c) x >= y
 - (d) x <= y
 - (e) x != y-2
 - (f) x < 10
 - (g) x >= 0 and x < 10
 - (h) x < 0 and x < 10
 - (i) x >= 0 and x < 2
 - (j) x < 0 or x < 10
 - (k) x > 0 or x < 10

(l) $x < 0$ or $x > 10$

- (a) True
- (b) True
- (c) False
- (d) True
- (e) False
- (f) True
- (g) True
- (h) False
- (i) False
- (j) True
- (k) True
- (l) False

7. Given the following definitions:

$x, y = 3, 5$

$b1, b2, b3, b4 = \text{True}, \text{False}, x == 3, y < 3$

evaluate the following Boolean expressions:

- (a) $b3$
- (b) $b4$
- (c) $\text{not } b1$
- (d) $\text{not } b2$
- (e) $\text{not } b3$
- (f) $\text{not } b4$
- (g) $b1 \text{ and } b2$
- (h) $b1 \text{ or } b2$
- (i) $b1 \text{ and } b3$
- (j) $b1 \text{ or } b3$
- (k) $b1 \text{ and } b4$
- (l) $b1 \text{ or } b4$
- (m) $b2 \text{ and } b3$
- (n) $b2 \text{ or } b3$
- (o) $b1 \text{ and } b2$
- (p) $b1 \text{ or } b2 \text{ and } b3$

- (q) $b_1 \text{ and } b_2 \text{ and } b_3$
- (r) $b_1 \text{ or } b_2 \text{ or } b_3$
- (s) $\text{not } b_1 \text{ and } b_2 \text{ and } b_3$
- (t) $\text{not } b_1 \text{ or } b_2 \text{ or } b_3$
- (u) $\text{not } (b_1 \text{ and } b_2 \text{ and } b_3)$
- (v) $\text{not } (b_1 \text{ or } b_2 \text{ or } b_3)$
- (w) $\text{not } b_1 \text{ and } \text{not } b_2 \text{ and } \text{not } b_3$
- (x) $\text{not } b_1 \text{ or } \text{not } b_2 \text{ or } \text{not } b_3$
- (y) $\text{not } (\text{not } b_1 \text{ and } \text{not } b_2 \text{ and } \text{not } b_3)$
- (z) $\text{not } (\text{not } b_1 \text{ or } \text{not } b_2 \text{ or } \text{not } b_3)$

- (a) True
- (b) False
- (c) False
- (d) True
- (e) False
- (f) True
- (g) False
- (h) True
- (i) True
- (j) True
- (k) False
- (l) True
- (m) False
- (n) True
- (o) False
- (p) True
- (q) False
- (r) True
- (s) False
- (t) True
- (u) True
- (v) False
- (w) False
- (x) True

- (y) True
- (z) False

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

- (a) $\text{not } (x == 2)$
- (b) $x < 2 \text{ or } x == 2$
- (c) $\text{not } (x < y)$
- (d) $\text{not } (x \leq y)$
- (e) $x < 10 \text{ and } x > 20$
- (f) $x > 10 \text{ or } x < 20$
- (g) $x \neq 0$
- (h) $x == 0$

- (a) $x \neq 2$
- (b) $x \leq 2$
- (c) $x \geq y$
- (d) $x > y$
- (e) $\text{!(}10 < x < 20\text{)}$
- (f) $10 < x < 20$
- (g) $x \neq 0$
- (h) $x == 0$

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

- (a) $\text{not } (x == y)$
- (b) $\text{not } (x > y)$
- (c) $\text{not } (x < y)$
- (d) $\text{not } (x \geq y)$
- (e) $\text{not } (x \leq y)$
- (f) $\text{not } (x \neq y)$
- (g) $\text{not } (x \neq y)$
- (h) $\text{not } (x == y \text{ and } x < 2)$

- (i) not (x == y or x < 2)
- (j) not (not (x == y))

- (a) x!=y
- (b) x<=y
- (c) y<=x
- (d) x<y
- (e) x>y
- (f) x==y
- (g) x==y
- (h) x!=y or x>=2
- (i) x!=y and x>=2
- (j) x==y

10. What is the simplest tautology?

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

11. What is the simplest contradiction? $x==2$ and $x<2$

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 number"))
if 1<=x<=100:
    print("\nok")
```

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 number"))
if 1<=x<=100:
    print("\nok")
```

```
else:  
    print("\nout 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("please enter one day of week in lowercase letters:")  
if x=='monday':  
    print ("lunes")  
elif x=='tuesday':  
    print("martes")  
elif x=='wednesday':  
    print("miércoles")  
elif x=='thursday':  
    print("jueves")  
elif x=='friday':  
    print("viernes")  
elif x=='saturday':  
    print("sábado")  
elif x=='sunday':  
    print("domingo")
```

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:  
        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
- (b) i is 3,j is 7, and k is 5
- (c) i is 5,j is 3, and k is 7
- (d) i is 5,j is 7, and k is 3
- (e) i is 7,j is 3, and k is 5
- (f) i is 7,j is 5, and k is 3

- (a) i is 5,j is 5, and k is 7
- (b) i is 3,j is 5, and k is 5
- (c) i is 7,j is 3, and k is 7
- (d) i is 5,j is 3, and k is 3
- (e) i is 5,j is 3, and k is 5
- (f) i is 7,j is 7, and k is 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
- (b) 21
- (c) 5
- (d) 17
- (e) -5

- (a) wow3
- (b) whoa 21
- (c) 6
- (d) 27
- (e) wow-5

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

<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
- (b) 1
- (c) 5
- (d) 50
- (e) 500
- (f) 5000

First program

- (a)****()
- (b)***()
- (c)***()
- (d)**()

```
(e)*()  
(f)()
```

Second program

```
(a)*()  
(b)*()  
(c)*()  
(d)*()  
(e)*()  
(f)*()
```

Why do the two programs behave as they do?

because multi-way conditional statements behave differently from sequential ifs.

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())  
b=int(input())  
c=int(input())  
d=int(input())  
e=int(input())  
max=a  
if b>max:  
    max=b  
if c>max:  
    max=c  
if d>max:
```

```

    max=d
if e>max:
    max=e
print("The maximum number entered was:", max)
min=a
if b<min:
    min=b
if c<min:
    min=c
if d<min:
    min=d
if e<min:
    min=e
print("the minimum number entered was:",min)

```

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())
b=int(input())
c=int(input())
d=int(input())
e=int(input())
if a==b or a==c or a==d or a==e:
    print("DUPLICATES")
elif b==c or b==d or b==e:
    print("DUPLICATES")
elif c==d or c==e:
    print("DUPLICATES")
elif d==e:
    print("DUPLICATES")
else:

```

```
print("ALL UNIQUE")
```