

In the name of god

(HW_2)

Kian nazari

.....

1. What possible values can a Boolean expression have? True & False

.....

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

.....

7. Given the following definitions:

B1, b2, b3, b4 = True, False, x == 3 (True) , y < 3 (True)

- (a) b3 \rightarrow True
- (b) b4 \rightarrow True
- (c) not b1 \rightarrow False
- (d) not b2 \rightarrow True
- (e) not b3 \rightarrow False
- (f) not b4 \rightarrow False

(g) b1 and b2	-> False
(h) b1 or b2	-> True
(i) b1 and b3	-> True
(j) b1 or b3	-> True
(k) b1 and b4	-> True
(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) not (x == 2) | -> x != 2 |
| (b) x < 2 or x == 2 | -> x <= 2 |

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

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

Are integers.

- (a) $\text{not } (x == y)$ $\rightarrow !(x == y)$ $\Rightarrow x \neq y$ / $x \neq y$
 (b) $\text{not } (x > y)$ $\rightarrow !(x > y)$ $\Rightarrow x \leq y$ / $x \leq y$
 (c) $\text{not } (x < y)$ $\rightarrow !(x < y)$ $\Rightarrow x \geq y$ / $x \geq y$
 (d) $\text{not } (x \geq y)$ $\rightarrow !(x \geq y)$ $\Rightarrow x < y$ / $x < y$
 (e) $\text{not } (x \leq y)$ $\rightarrow !(x \leq y)$ $\Rightarrow x > y$ / $x > y$
 (f) $\text{not } (x \neq y)$ $\rightarrow x == y$
 (g) $\text{not } (x == y)$ $\rightarrow x \neq y$
 (h) $\text{not } (x == y \text{ and } x < 2)$ $\rightarrow x \neq y \text{ or } !(x < 2) \Rightarrow (x \neq y) \text{ or } (x \geq 2)$ / $x \neq y \text{ or } x \geq 2$
 (i) $\text{not } (x == y \text{ or } x < 2)$ $\rightarrow x \neq y \text{ and } !(x < 2) \Rightarrow x \neq y \text{ and } x \geq 2$
 (j) $\text{not } (\text{not } (x == y))$ $\rightarrow x == y$

10. What is the simplest tautology? The simplest tautology is "A statement is true if and only if it is true."

11. What is the simplest contradiction? $A = \text{not } a$

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.

```
#Start and input section

Num = int(input("Enter a buetiful num between 1 , 100 → "))

# Check number blocks

If 1 < num < 100 :

    Print("OK !!")

#else :

    # pass
```

.....

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."

```
#Start and input section

Num = int(input("Enter a buetiful num between 1 , 100 → "))

# Check number blocks

If 1 < num < 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.

```
Print("\t\t\t<< Do you like to learn days in Spanish? >> ")
```

```
Print("\n\n* [ Type end for stop ] * \n\n")
```

```
While (True):
```

```
    Day = input("\nEnter a day [completely] → ")
```

```
    Day = day.lower()
```

```
# Saturady
```

```
    If day == "Saturday" or 'sat' or 'satur' :
```

```
        Print("\n\t\t\t<< In Spanish it is ' Sábado ' >>\n")
```

```
# Sunday
```

```
    Elif day == 'Sunday' or 'sun' or 'sund' :
```

```
        Print("\n\t\t\t<< In Spanish it is ' Domingo ' \n")
```

```
# Monday
```

```
    Elif day == 'Monday' or 'mon' or 'mond' :
```

```
        Print("\n\t\t\t<< In Spanish it is ' Lunes ' \n")
```

```
# Tuesday
```

```
    Elif day == "Tuesday" or 'tues' or 'tu' :
```

```
        Print("\n\t\t\t<< In Spanish it is ' Martes ' \n")
```

```
# Wednesday
```

```
    Elif day == 'Wednesday' or 'wedn' or 'wednes' :
```

```
        Print("\n\t\t\t<< In Spanish it is ' Miércoles ' \n")
```

```
# Thursday
```

```
    Elif day == "Thursday" or 'thurs' or 'thur' :
```

```
        Print("\n\t\t\t<< In Spanish it is ' Jueves ' \n")
```

```
# Friday
```

```
    Elif day == 'Friday' or 'fri' or 'f' :
```

```
        Print("\n\t\t\t<< In Spanish it is ' Viernes ' \n")
```

```
# End cheking
```

```

Elif day == 'end' or 'e' or 'en' :

    Print("Have a nice time")

    Break

Else :

    Print("\n\t\t\t Incorrect input I don't undrestand \n")

```

.....

15. Consider the following Python code fragment:

```

# l, j, and k are numbers

If l < j:

    If j < k:

        l = j

    Else:

        J = k

Else:

    If j > k:

        J = i

    Else:

        l = k

Print("l =", l, " j =", j, " k =", k)

```

_What will the code print if the variables l, j, and k have the following values?

- | | |
|--------------------------------|--------------------------|
| (a) l is 3, j is 5, and k is 7 | -> l = 5 , j = 5 , k = 7 |
| (b) l is 3, j is 7, and k is 5 | -> l = 3 , j = 5 , k = 5 |
| (c) l is 5, j is 3, and k is 7 | -> l = 7 , j = 3 , k = 7 |
| (d) l is 5, j is 7, and k is 3 | -> l = 5 , j = 3 , k = 3 |
| (e) l is 7, j is 3, and k is 5 | -> l = 5 , j = 3 , k = 5 |

(f) l is 7, j is 5, and k is 3

-> l = 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:

```
_1:
    n = int(input())
    if n < 1000:
        print('*', end='')
    if n < 100:
        print('*', end='')
```



```

if n < 10:
    print('*', end='')
if n < 1:
    print('*', end='')
    print()

```

```

_2:
    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	(1) -> *****	&&	(2) -> *
(b) 1	(1) -> ***	&&	(2) -> *
(c) 5	(1) -> ***	&&	(2) -> *
(d) 50	(1) -> **	&&	(2) -> *
(e) 500	(1) -> *	&&	(2) -> *
(f) 5000	(1) ->	&&	(2) ->

_ Why do the two programs behave as they do?

1. In program (1) first start to check if n < 1000 (if this is true execute block of the if else pass to the next if)

2. But in program (2) After we check if n < 1000 (if this is true must execute the if n < 1000 block and down but if it

Was not true check the elif)

.....

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 $n5 > \text{max}$:

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".

```
n1 = int(input("num 1 → "))
```

```
n2 = int(input("num 2 → "))
```

```
n3 = int(input("num 3 → "))
```

```
n4 = int(input("num 4 → "))
```

```
n5 = int(input("num 5 → "))
```

```
Duplicated = 0
```

```
If n1 == n2 :
```

```
Duplicated += 1
```

```
If n1 == n3 :
```

```
Duplicated += 1
```

```
If n1 == n4 :
```

```
Duplicated += 1
```

```
If n1 == n5 :
```

```
Duplicated += 1
```

```
If n2 == n3 :
```

```
Duplicated += 1
```

```
If n2 == n4 :
```

```
Duplicated += 1
```

```
If n2 == n5 :  
    Dublicated += 1  
If n3 == n4 :  
    Dublicated += 1  
If n3 == n5 :  
    Dublicated += 1  
If n4 == n5 :  
    Dublicated += 1  
If dublicated != 0 :  
    Print(" * DUPLICATES *")  
Else :  
    Print("ALL UNIQUE")
```

.....