



## More on Booleans

Boolean is a type of datatype that contains only True and False values. When we convert or typecast any value into boolean then we use bool() function. This bool() function either return True or False after passing any value in the argument of bool() function. If the bool() function returns **True** then the value which we pass in the argument of bool() function is **.truthy** value. If the bool() function returns **False** then the value which we pass in the argument is **falsy** value.

```
In [4]: #Truthy and Falsy Code
print(bool("Amrit Keshari"))
print(bool(68))
print(bool(0))
print(bool(" "))
print(bool(None))
print(bool([]))
```

```
True
True
False
True
False
False
```

All the integer values, float values, string, list, tuple, dictionary, set values are truthy values as when we pass these things in bool() function then it returns **True** as an output. If we pass empty string, None, zero, empty list, empty tuple, empty dictionary, empty set then it returns **False** as an output.

## Operator & Operand

Operators are the symbols which is mainly used to perform some mathematical and logical operation in our program during computation. To perform we need operators and operands. Now operands are the data in which we perform mathematical or logical operation. Just like we add two numbers we see we have two numbers as an operand and + sign as an additional operator.

## Types of Operators

1. Arithmetic Operator is mainly used to perform mathematical operation. Just like addition(+), subtraction (-), multiplication (\*) and division(/ %) of numbers. It is mainly used to compute or calculate the resultant value of any given equation along with values for each variable present in the equation. "%" is remainder operator that returns

remainder and “/” division operator only it return quotient only.

In [49]: *#Arithmetic Operator*

```
k = 48
e = 24
print(k+e)
print(k-e)
print(k*e)
print(k/e)
print(k%e)
```

```
72
24
1152
2.0
0
```

2. Logical Operator is mainly used to perform logical operation on operands. (**and**) AND operator is used in checking conditions. If we have to check two condition is true or not then we use (**and**) operator and we put and operator between two condition or expression. If both the condition returns true then it returns true otherwise it returns false. (**or**) OR operator is used to check whether any one condition or expression from both condition or expression is true or not. If any one condition or expression is true then it returns true. (**not**) NOT operator is used to change the true into false and false into true.

In [45]: *#Logical Operator*

```
k = 48
e = 24
s = 33
print(k>e and e<s)
print(k<e or e<s)
print(not (k>e))
```

```
True
True
False
```

3. Relational Operator is mainly use to perform relational operation in any expression. It compares two operand thats why it is also known as comparision operator. It returns boolean value as output. (**>=**) greater than equal to operator is used to check whether left operand is greater than and equal to right operand or not. (**<=**) less than equal to operator is used to check whether left operand is lesser than and equal to right operand or not. (**==**) equals to operator is used to check

whether both the operands are equals or not. (>) greater than operator is used to check whether left operand is greater than right operand or not. (<) less than operator is used to check whether left operand is less than right operand or not.

```
In [47]: #Relational Operator
k = 48
e = 24
print(k>=e)
print(k<e)
print(k==e)
print(k>e)
print(k<=e)
```

```
True
False
False
True
False
```

4. Bitwise Operator is mainly used to perform binary operations on any numbers. It perform operation in bitwise level. After performing operation, it gives number as an output. (&) bitwise AND operator set bit as 1 if both the bit is 1. (|) bitwise OR operator set bit as 1 if one of the both bit or both the bit is 1. (^) bitwise XOR operator set bit as 1 if any one of the bit from both the bit is 1. (<<) left shift operator is just shift the left bit by adding zeroes in rightest side of the binary. (>>) right shift operator shift the leftmost bit to the right side of the binary.

```
In [43]: #Bitwise Operator
k = 4
e = 2
print(k & e)
print(k | e)
print(k ^ e)
print(k << 2)
print(k >> 2)
```

```
0
6
6
16
1
```

5. Identity Operator (**is**) operator is mainly used to check whether the left operand is pertaining to object to the right operand. It just check two operand if the object of both operand is same then it returns true

otherwise it returns false. (**is not**) operator is mainly used to check whether the left operand is not pertaining to the right one then it returns true. Simply means if both the values are equal then (**is**) operator returns true.

```
In [59]: #Identity Operator  
k = 2  
e = 2  
print(k is e)
```

True

6. Membership Operator (**in**) operator is mainly used to check whether the left operand is present in the sequence of right operand it means left operand is just a variable and right one is sequential type variable just like list. To check whether the particular value of variable present in the list or not. If it present in the list, it returns true otherwise false.

```
In [64]: #Membership Operator  
k = [1,3,5,7]  
e = 3  
print(e in k)
```

True