

## Identity Operators

**Similar**

**Check Both are Same or Not**

**LHS and RHS**

**Same - True**

**NSame - False**

**a=10**

**b=20**

**a==b False**

**is**

**Same -> True**

**OUTPUT**

**is not**

**Different -> True**

**BOOLEAN**

**a is not b**

a=10

b=20

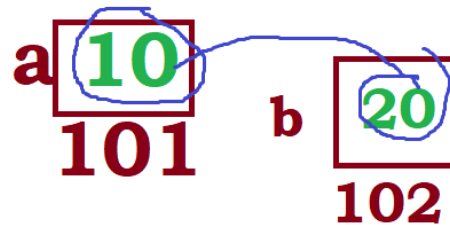
print(a is b)#False

print(a is not b)#True

print(a==b)#False

# is and == YES

10 is 20 - False



a=[10,20,30]

b=[10,20,30]

print(a==b)#True

print(a is not b)#True

print(a is b)#False



**Membership Operators** It Returns **TRUE** when the value is present in a Collection or not

**in**

**not in**

**a=10**

**b=20**

**a in a**

**a** **10** **10** **.**

**b = [ 10 , 20 ,30 ]**

**b in b**

**10 20 30**

**20 30 10**

**print(b in b)#False**

**print(a in b is c)#False**

**# True is 20 # False**

**print(b is not a)#True**

**print(b not in**

**a)#TypeError**

```
a=10
c=20
b=[10,20,30]
print(a in b)#True
print(a in a)#TypeError
print(b in b)#False
print(a in b is c)#False
# True is 20 # False
print(b is not a)#True
print(b not in a)#TypeError
```

**OUTPUT**

**BOOLEAN**

## Bitwise Operators ( Dangrouis Operators - They Works on Only Bits )

<b>&amp;</b> <b> </b> <b>not</b> <b>^</b> <b>~</b> <b>&lt;&lt;</b> <b>&gt;&gt;</b>	<b>Bitwise and</b> <b>Bitwise or</b> <b>Bitwise not</b> <b>Bitwise XoR</b> <b>Bitwise Tiled</b> <b>Bitwise Left Shift</b> <b>Bitwise Right Shift</b>	<b>a=20</b> <b>b=30</b>	<div> <div>16</div> <div>8</div> <div>4</div> <div>2</div> <div>1</div> </div> <div> <div>1</div> <div>0</div> <div>1</div> <div>0</div> <div>0</div> </div> <div> <div>1</div> <div>1</div> <div>1</div> <div>1</div> <div>0</div> </div> <div> <div>1</div> <div>0</div> <div>1</div> <div>0</div> <div>0</div> </div> <div> <div>1</div> <div>1</div> <div>1</div> <div>1</div> <div>0</div> </div> <div> <div>0</div> <div>1</div> <div>0</div> <div>1</div> <div>0</div> </div>	<div> <div>- 20</div> <div>- 30</div> <div>&amp; 20</div> <div>  30</div> <div>^ 10</div> </div>	<div> <div>^</div> <div>Ex OR - XoR</div> <div>Same - 0</div> </div> <div> <div>R</div> <div>A</div> <div>B</div> </div> <div> <div>0</div> <div>1</div> <div>1</div> <div>0</div> </div> <div> <div>0</div> <div>0</div> <div>1</div> <div>1</div> </div> <div> <div>0</div> <div>1</div> <div>0</div> <div>1</div> </div>
--	--	----------------------------	--	--	---

## Tiled Operators ~ Only for Single Var

$\sim 2 = ?$  -3

1. Consider Value
2. Increment by 1
3. Assign Opposite Sign

2	-100
$2+1 = 3$	$-100+1 = -99$
-3	-99
	99

.

20	Most Significant Bit (Sign)					
MSB	16	8	4	2	1	~20 => -21
0	1	0	1	0	0	
+1				+	1	
1	1	0	1	0	1	
-	16 + 4 + 1 = 21					

```

a=20
b=30
print(a & b)#20
print(a | b)#30
print(a ^ b)#10
print(~a)#-21
print(~-100)#99
print(b<<5)#960
print(b>>5)#0

```

a = 13

Left Shift ( << )

(a) << Count a << 2

64	32	16	8	4	2	1
			1	1	0	1
		1	1	0	1	
	1	1	0	1		
1	1	0	1			

= 13

= 26 (1)

= 52 (2)

= 104 (3)

Val is Doubled by  
Original Value for Every  
Shift



count >> values **a = 20**

Half - Part of Original value for  
Each and Every Shift

3 >> a

Right Shift

	16	8	4	2	1				
20	1	0	1	0	0				
(1) 10		1	0	1	0	0			
(2) 5			1	0	1	0	0		
(3) 2				1	0	1	0	0	