

PYTHON PROGRAMMING

GATE DA & DSA

Agenda :

→ Type Casting ✓

→ Operators

Type Casting : Type Casting is a method to convert the Python variable datatype into a certain data type in order to perform the operations -

$a=7$
 $b=3.0$
 $c=10.0$ $\text{int}(10.0)$ Type Casting Implicit Type Conversion
 Explicit Type Conversion.

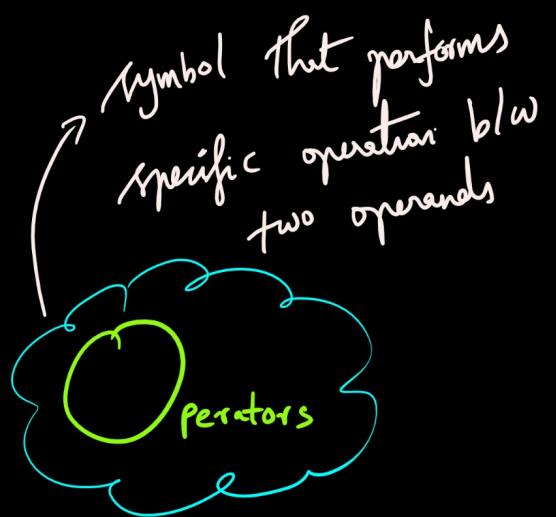
Implicit Type Conversion : In This method, Python converts the datatype into another datatype automatically. User is not involved in this process.

Explicit Type Conversion : Python needs user involvement to convert the datatype of variable into the derived datatype.

$\text{int}(\leftarrow)$ takes float or string as 'argument' and returns int

$\text{float}(\leftarrow)$ takes int or string as argument and returns float

$\text{str}(\leftarrow)$ takes float or int as argument and returns str.



- ## Supported Operators in Python
- Arithmetic Operators
 - Comparison Operators
 - Assignment Operators
 - Logical "
 - Bitwise "
 - Membership "
 - Identity "

Arithmetic Operators:

- + → addition
- → subtraction
- * → multiplication
- / → division
- % → remainder (modulus)
- **** → Exponent
- // → Floor Division

Comparison Operators

- Compare the two operands and returns

a true or false.

i) $= = \rightarrow$ check if a & b are equal.

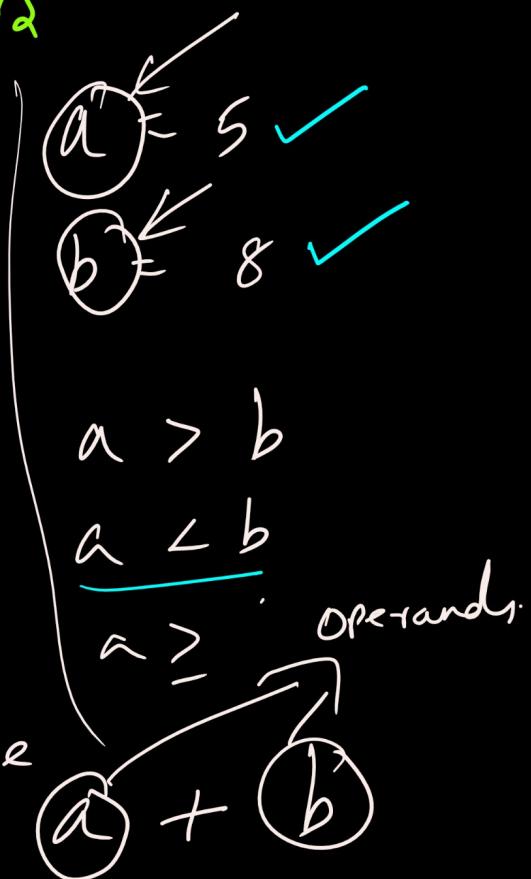
ii) $\neq \rightarrow$ check if a & b are not equal.

iii) $\leq \rightarrow$ check if a is less than or equal to b

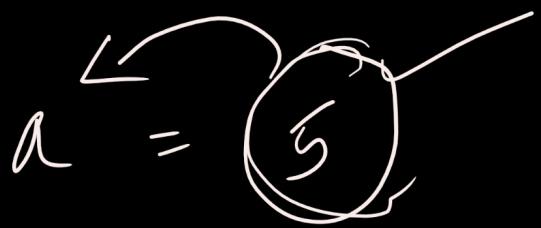
iv) $\geq \rightarrow a$ is greater than $= b$

v) $<$

vi) $>$



Assignment Operators :-



right hand side expression value
is assigned to left hand side
operand

- $=$
- $+ =$
- $- =$
- $* =$
- $/ \cdot =$
- $** =$
- $// =$

Bitwise Operators :-

Bitwise OR (|)

Bitwise AND (&)

Bitwise XOR (^)

Negation (~)

Left Shift (<<)

Right Shift (>>)

$$\begin{array}{l} a = 2 \rightarrow 0010 \\ b = 5 \rightarrow 0101 \end{array}$$

AND

OR

0000

0111

Logical Operators :-

a, b
and → if both are true, then it returns true
or → if either one of them is true, it returns true.
not

↳ $\text{not } (\text{a})$ a is true
 false.

Membership Operators :-

The membership of a value inside a python data structure.

→ `` in ''

→ `` not in ''

$L = [1, 2, 3, 4, 5]$

$\Rightarrow 3 \text{ in } L \rightarrow \text{True}$

$\Rightarrow 8 \text{ not in } L \rightarrow \text{True}$

$\Rightarrow 8 \text{ in } L \rightarrow \text{False}$

Identity Operators:

'is': If the reference on both operands points to same object, then it is true.

'is not': If the reference on both sides do not point at same object, then it is false.

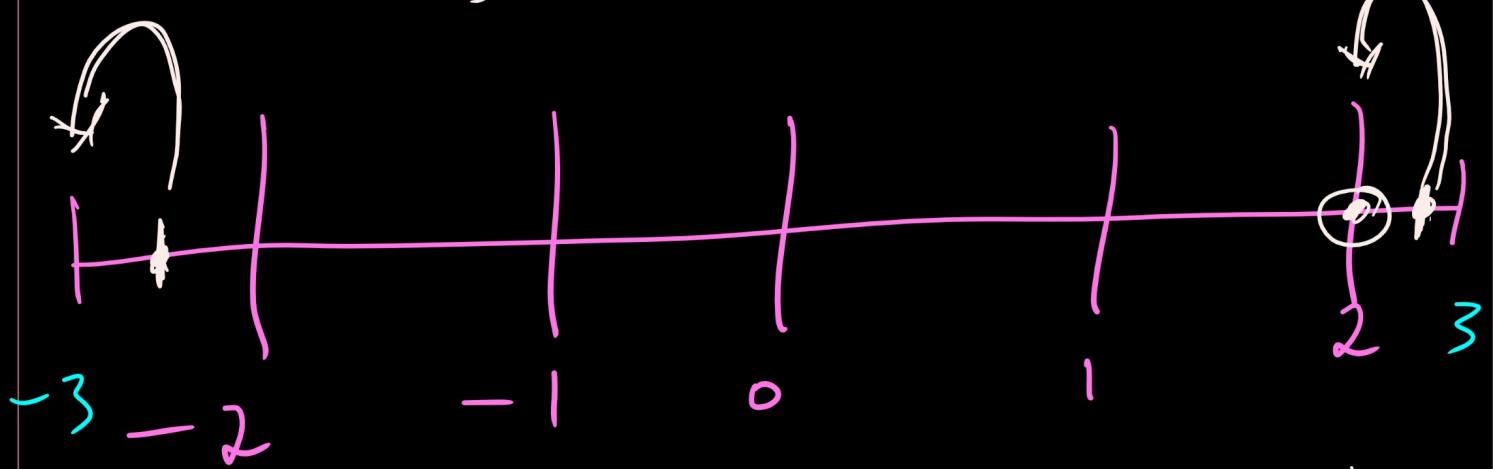
Rough:

$$2^5 = 35$$

$$\lceil 2 ** 5 \rceil \leftarrow \text{In Python.}$$

$a ** b = a^b$

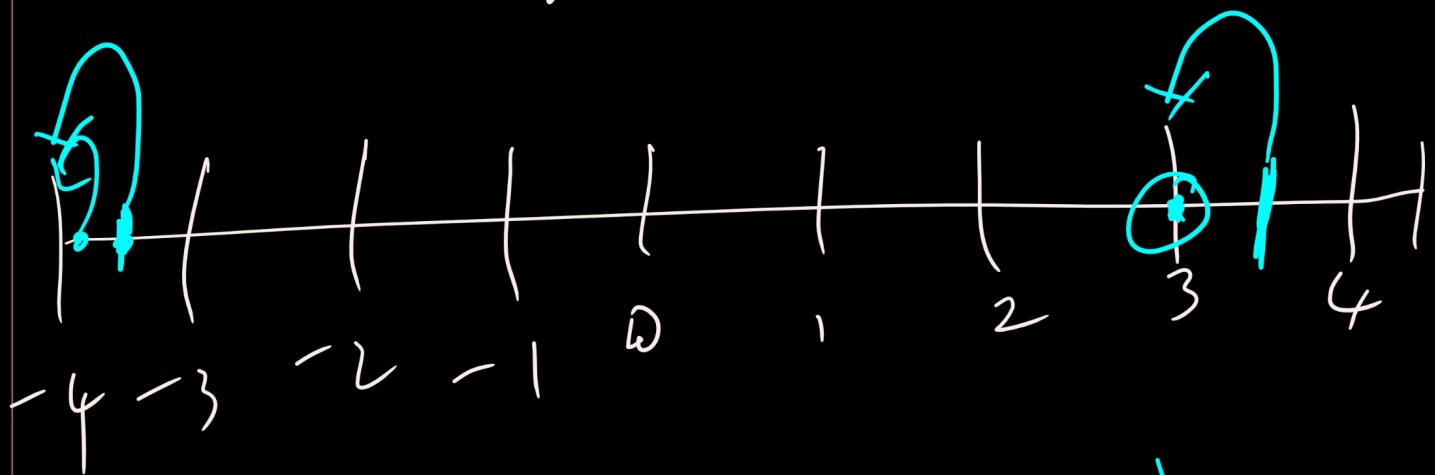
floor division



$$(5 // 2) = \lfloor 2^5 \rfloor = 2$$

$$(-5 // 2) = \lfloor -2^5 \rfloor = -3$$

$$7//2 = \lfloor 3.5 \rfloor = 3$$



$$-7//2 = \lfloor -3.5 \rfloor = -4$$

$$-7 \cdot 9 // 2 = \lfloor -3 \cdot 95 \rfloor = -4$$

A division diagram showing $8 \div 5$. The quotient is 1 and the remainder is 3. The remainder 3 is circled and labeled R , while the quotient 1 is circled and labeled Q .

\rightarrow modulus operator

$$11 \cdot 1 \cdot 2 \rightarrow \textcircled{1} - R$$

$$17 \cdot 1 \cdot 5 \rightarrow \textcircled{2}$$

$$\begin{array}{r} 12 \cdot 5 \\ 8 \cdot 7 \\ \hline 12 \end{array} \quad (8 \times 1) + 4$$

if a number
is less than
the number you
want to take
modulus

$$\textcircled{2}^8 \cdot 4^{-1} \cdot 8 \Rightarrow 4$$

$$x \cdot 1 \cdot 8 = x$$
$$17 \cdot 1 \cdot 35 = \textcircled{17}$$

$$12 \cdot 5$$
$$7 \cdot 5 \quad (7 \cdot 5 * 1) + (5 \cdot 0)$$

$$20 \cdot 5 \cdot 7 \cdot 5 \Rightarrow 7 \cdot 5 + 20 \cdot 5 - 15 \rightarrow 5 \cdot 5$$

-7 / -5

Modulus \rightarrow will return +ve values only.

$$(-7 \cdot 1) \cdot 5$$

↓

$$(-1) \cdot 5$$

$$-5 + 2 = -3$$

$$(-5 + 2) \bmod 5$$

$$(-5 \bmod 5) + (2 \bmod 5)$$

↑

$$0 + 2 = 2$$

$$-7 \bmod 5$$

$$-5 - 2$$

$$(-10 + 3) \bmod 5$$

$$(-10 \cdot 1 \cdot \underline{\text{mod } 5}) + \underline{3 \text{ mod } 5}.$$

0

3

$$-21 \cdot 1 \cdot 8 \Rightarrow 3$$

$$(-24 + 3) \cdot -8$$

-16 5

$$(-24 \cdot 1 \cdot 8) + 3 \text{ mod } 5$$

0

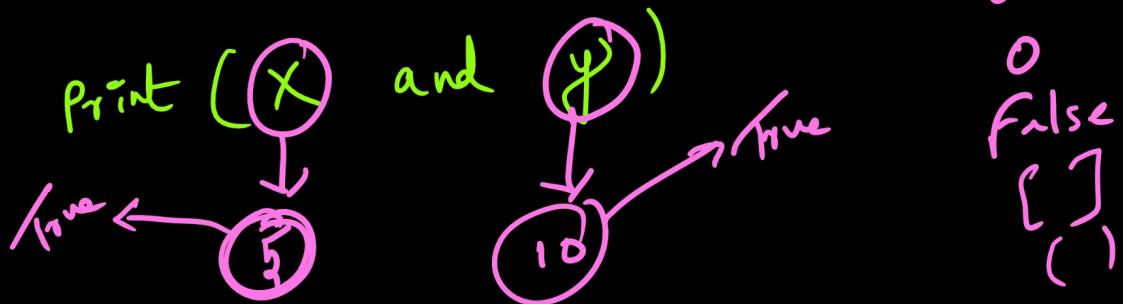
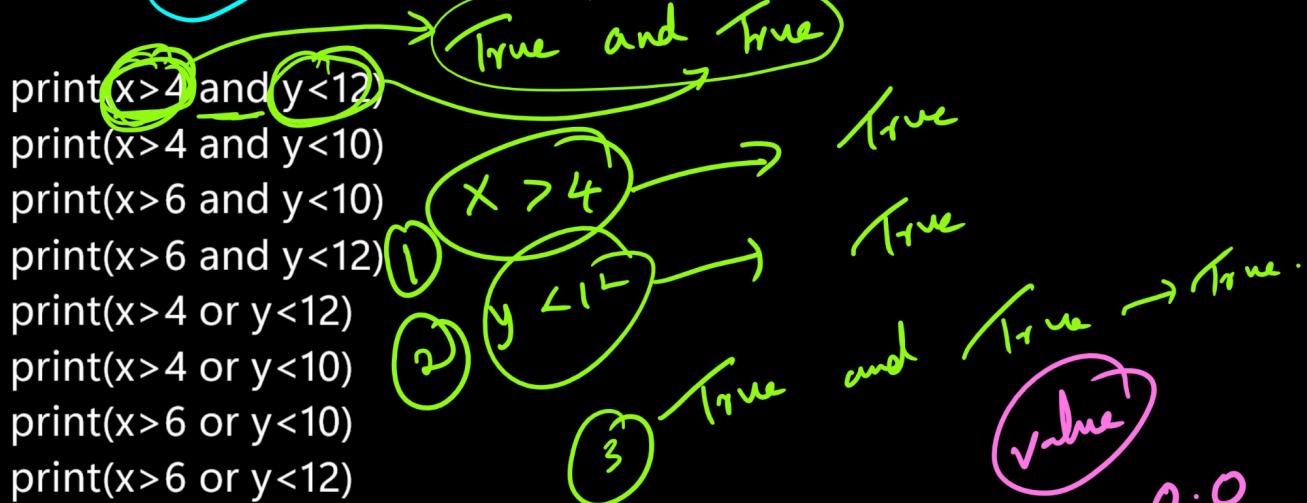
3

32

24

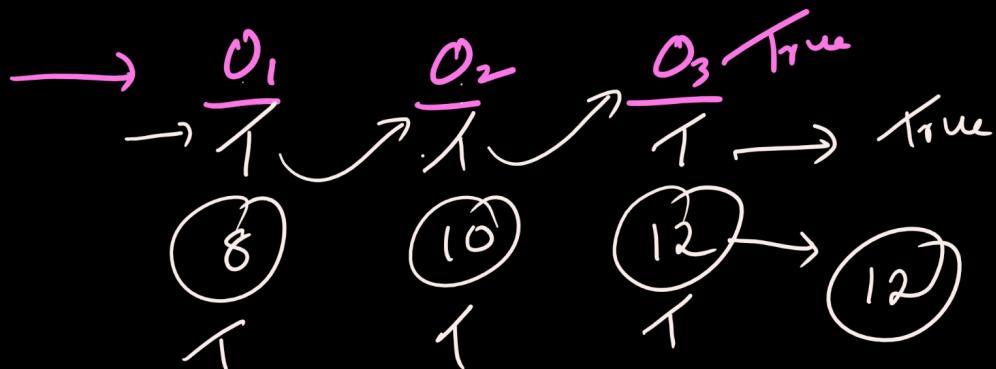
-21

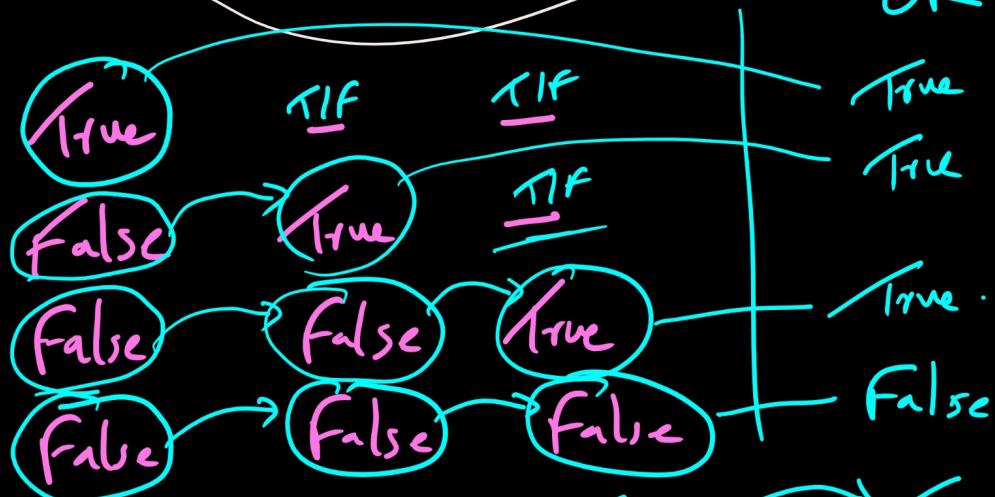
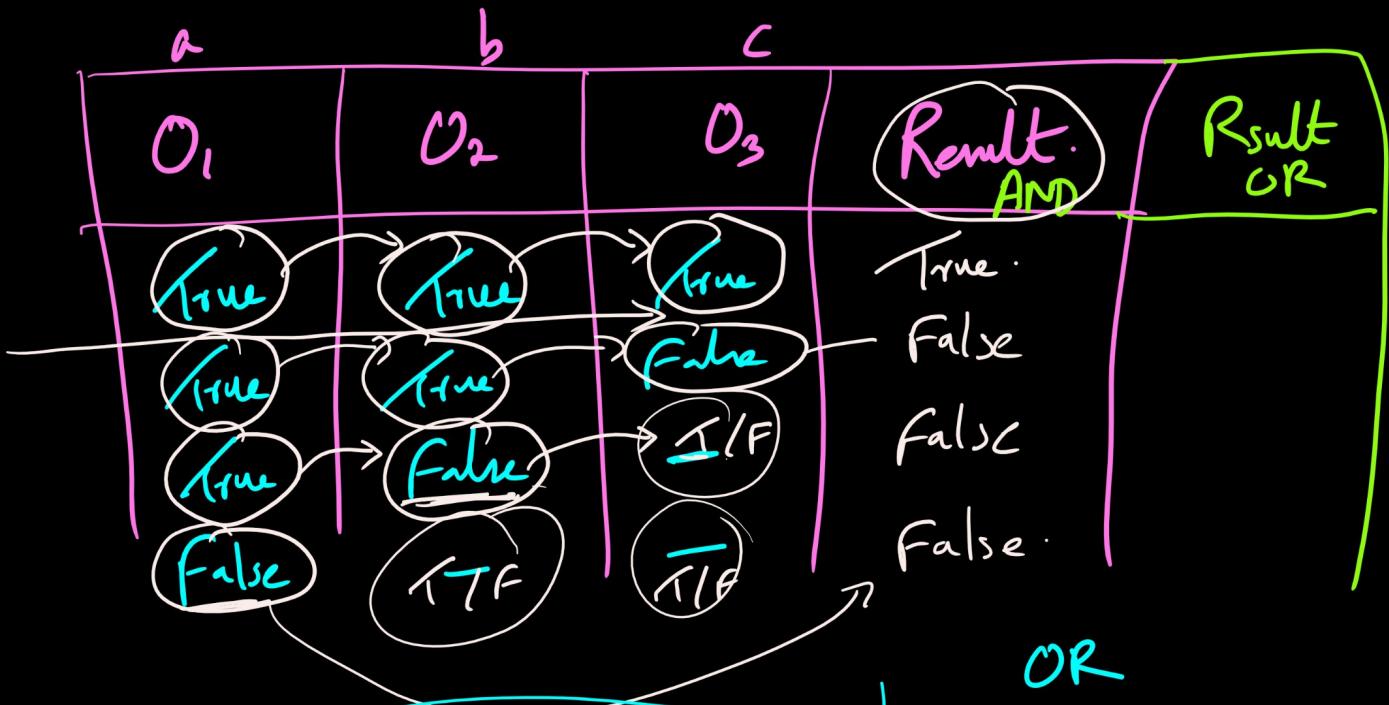
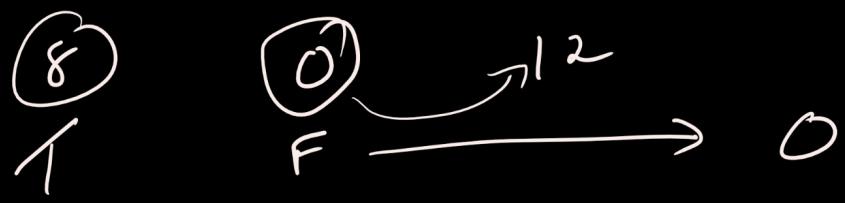
| $x = 5$; $y = 10$ | | |
|-------------------------------|------------------------------|-------|
| print($x > 4$ and $y < 12$) | True and True \rightarrow | True |
| print($x > 4$ and $y < 10$) | True and False \rightarrow | False |
| print($x > 6$ and $y < 10$) | False \rightarrow | False |
| print($x > 6$ and $y < 12$) | False \rightarrow | False |
| print($x > 4$ or $y < 12$) | True or \rightarrow | True |
| print($x > 4$ or $y < 10$) | True \rightarrow | True |
| print($x > 6$ or $y < 10$) | False or False \rightarrow | False |
| print($x > 6$ or $y < 12$) | False or True \rightarrow | True |



5, 10, 12

and operators.







a = 10
b = 5
c = 12
d = 0

print(a and c and b) # q1
print(d or a or c) # q2
print(not(a and d and b)) # q3
print(not(d or d or a)) # q4

return either (True or False)

q1: 10 and 12 and 5 → T and T and T → 5

q2: 0 or 10 or 12 → F or T or T → T

q3: not(10 and 0 and 5)
not(0) → False

q4: not(d or 0 or a)
not(0) → True
not(10) → False

Reference:-
 1. Head First Python
 2. Let Us Python - Yashwant Kanetkar.

Colab Notebooks Link:

Class - 1: <https://colab.research.google.com/drive/1FWWZdAPJoUMDZZUd2g6IPj6MW3B63AkW?usp=sharing>

Class - 2: https://colab.research.google.com/drive/1yzZzBM7HBIMhWIxNDO_5nNqc1MgYrJaQ?usp=sharing

Class - 3: <https://colab.research.google.com/drive/1g0ZmthPl1b-GLOZ4krWzBXy37xmDI3R-?usp=sharing>

Class - 4: <https://colab.research.google.com/drive/1YWTYiEHtg8W1pqbtgMixePWW8GGrNQLL?usp=sharing>