

GATE

CRASH COURSE

Data Science & AI

Subject

**Python - For Data Science
Fundamentals of Python
Lec No. 02**

By – Satya Sir



Last Class

Quick Recap

- 1 Python History, Features
- 2 Python Datatypes
- 3 Input() and print() Functions
- 4 Examples



Topics to be covered

- 1 Types Of Operators
- 2 Logical Operators
- 3 Bitwise and Shift Operators
- 4 Arithmetic and Other Operators
- 5 Operator Precedence and Associativity





Python Operators



Operator: Operator Performs operation on Operands.

Ex: $x * y$

x, y operands
multiplication operation
* operator

Operators in Python

1) Arithmetic operators

2) Logical operators

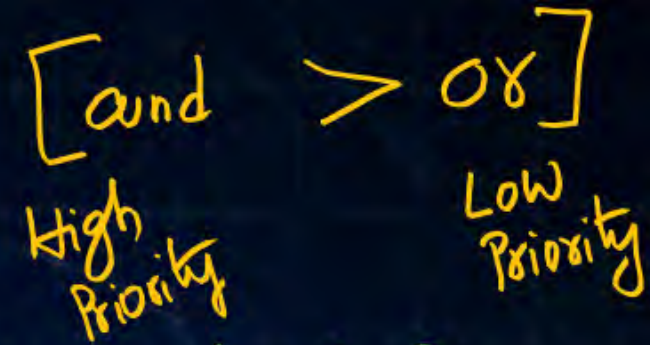
3) Bitwise (and) shift

4) Relational (or) Comparison

5) Assignment

6) Identity operators

7) Membership operators



- and, or, not

x	y	x and y	x or y	not y
<u>False</u>	False	<u>False</u>	False	True
<u>False</u>	True	<u>False</u>	True	False
<u>True</u>	False	<u>False</u>	<u>True</u>	True
<u>True</u>	True	True	True	False

Ex: $a = 15$
 $b = 12$
 $c = 10$

$b = 12$
 $c = 10$
 $15 < 12$ false
 $\text{result1} = (a < b) \text{ and } (b > c) \Rightarrow \text{false}$
 Need not to be Evaluated
 $(a > b)$ True
 $12 > 10$ True
 $\text{result2} = (b > c) \text{ or } (a > b)$
 Short-Circuit

```
Print (result 1, result 2)
```

Falge True



Logical Operators



Ex:

$$a = 5$$

$$b = 3$$

$$c = -2$$

Print ($a \leq b$ or $\boxed{b \geq c \text{ and } c \leq a}$) ^{True} ~~#~~ True

Print ($\boxed{b > a \text{ and } c < b}$ _{false} or $\frac{a > b}{\text{True}}$) ~~#~~ True

$$b \geq c \text{ and } c \leq a \Rightarrow \frac{3 \geq -2}{\text{True}} \text{ and } \frac{-2 \leq 5}{\text{True}} \Rightarrow \text{True}$$

$$a \leq b \text{ or True} \Rightarrow 5 \leq 3 \text{ or True} \Rightarrow \text{False or True} \Rightarrow \text{True}$$

o/p: True, True

Logical operators with non Conditional inputs : Decision Making Value will be result.

Ex:

$$a = 5$$

$$b = 3$$

$$c = -1$$

$$d = 0$$

$$i = a \text{ or } b \text{ or } c \Rightarrow 5 \text{ or } 3 \text{ or } -1$$

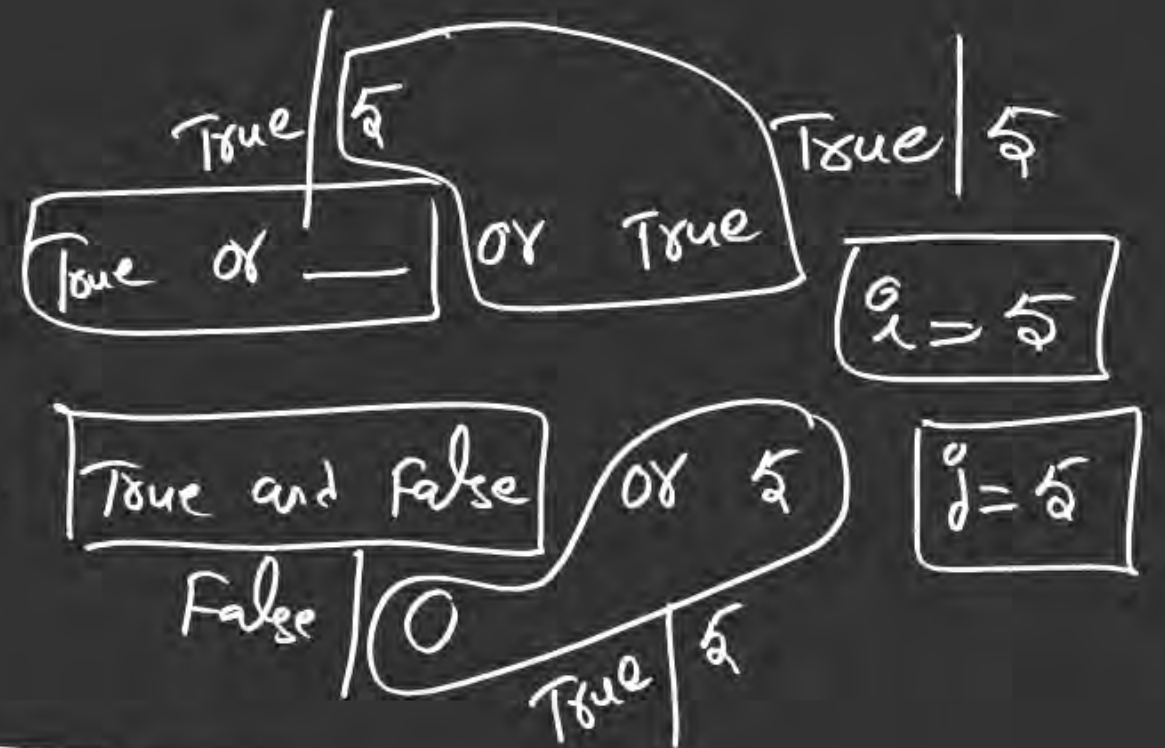
$$j = \boxed{b \text{ and } d} \text{ or } a \Rightarrow 3 \text{ and } 0 \text{ or } 5$$

$$k = c \text{ or } a \text{ and } b \Rightarrow -1 \text{ or } 5 \text{ and } 3$$

Print(i, j, k)

o/p: 5 5 -1

Any Non-Zero (\pm ve) Value == TRUE \Leftarrow Non-Empty string
Zero Value, None, False == FALSE \Leftarrow Empty string.



$$k = c \text{ or } a \text{ and } b \Rightarrow -1 \text{ or } 5 \text{ and } 3$$

Diagram illustrating the evaluation of the expression $-1 \text{ or } 5 \text{ and } 3$:

- -1 is True.
- $5 \text{ and } 3$ is True.
- The expression $-1 \text{ or } 5 \text{ and } 3$ evaluates to True.
- The result is $k = -1$.



Bitwise and Shift Operators



- ⇒ Perform operation on binary digits (bits = 1/0)
- ⇒ $\&$ (Bitwise AND), $|$ (Bitwise OR), \sim (Bitwise NOT), \wedge (Bitwise XOR)
- ⇒ $<<$ (Left shift), $>>$ (Right shift)

A	B	$A \& B$	$A B$	$A \wedge B$	$\sim A$
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0

Examples : 1)

$$i = 23 = 10111$$

$$j = 14 = 01110$$

$$k = i \oplus j$$

$$l = i \mid j$$

$$m = i \wedge j$$

$$n = \sim i$$

Print(k, l, m, n)

6 31 25 -8

$$2 \overline{) 23}$$

$$2 \overline{) 11} - 1$$

$$2 \overline{) 5} - 1$$

$$2 \overline{) 2} - 1$$

$$1 - 0 = 10111$$

$$k = \underline{2 \oplus 14}$$

$$i = 10111$$

$$j = 01110$$

$$\underline{00110} = 6$$

$$l = \underline{i \mid j}$$

$$i = 10111$$

$$j = 01110$$

$$\underline{11111} = 31 (1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0)$$

$$m = \underline{i \wedge j}$$

$$i = 10111$$

$$j = 01110$$

$$\underline{11001} = 25$$

$$n = \sim i = \sim (10111) = 01000 = 8$$

Ex: 2

Home-work

Post Answer/Solution in TG: Satya Sir PW

$$x = 37$$

$$y = 43$$

$$z = 69$$

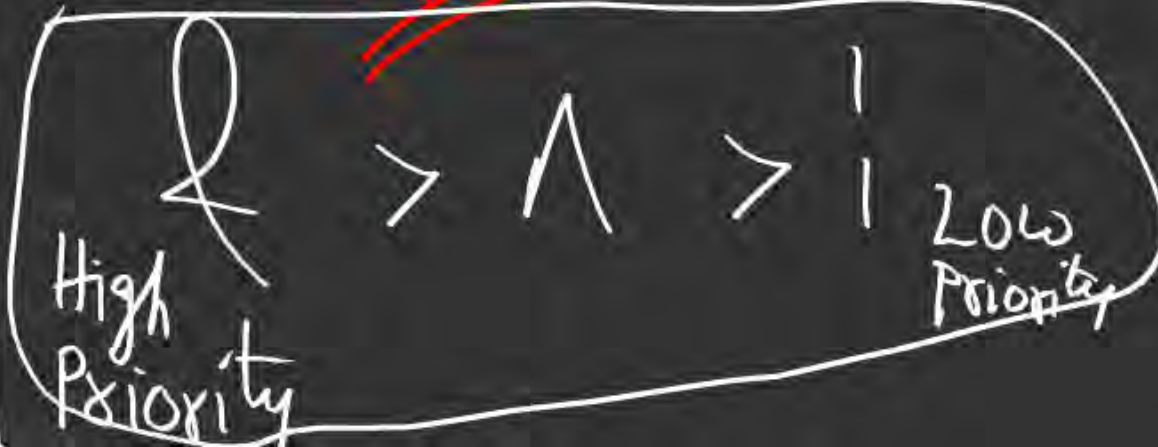
$$a = x \oplus y \wedge z$$

$$b = x \wedge y \mid z$$

$$c = z \mid x \wedge y$$

Print(a, b, c)

H/W



Shift operators (<<, >>)

$$\text{Variable} \ll \text{count} \equiv \text{Variable} * 2^{\text{count}}$$

$$\text{Variable} \gg \text{count} \equiv \text{Variable} // 2^{\text{count}}$$

* Decimal Integer

Examples

1) $a = 17$

$$b = 9$$

$$c = a \ll 2 \Rightarrow 17 \ll 2 \Rightarrow 17 * 2^2 = 68$$

$$d = b \gg 3 \Rightarrow 9 \gg 3 \Rightarrow 9 // 2^3 = 9 // 8 = 1$$

$$\text{Print}(c - d) \# 68 - 1 = \underline{\underline{67}}$$

$$\text{o/p} : 67$$

2)

O/p: 1526

$$x = 0034$$

$$y = 0x2A5$$

$$z = 0b10110110$$

$$\text{Print}((x \ll 2) - (y \gg 4) + (z \ll 3))$$

O/p =

$$\underline{\underline{(34 * 2^2) - (2A5 // 2^4) + (10110110 * 2^3)}}$$

$(34)_8 = (28)_{10}$ $3 \times 8^1 + 4 \times 8^0$ $= 24 + 4 = 28$	$(2A5)_{16} = (677)_{10}$ $2 \times 16^2 + 4 \times 16^1 + 5 \times 16^0$ $512 + 160 + 5$ $= 677$	$(10110110)_2$ $= 1 \times 2^7 + 1 \times 2^5 + 1 \times 2^4$ $+ 1 \times 2^2 + 1 \times 2^1$ $= 128 + 32 + 16 + 4 + 2 = 182$
--	--	--

$$(28 * 2^2) - (677 // 2^4) + (182 * 2^3) \quad (182)_{10}$$
$$= 112 - 42 + 1456 = 1456 + 70$$
$$= 1526$$



Arithmetic Operators



$$14 + 3 - 9 = 8$$

$$14.25 + (3 + 4j) - 9 = 8 + 4j$$

./(Modulus)

⇒ + (addition), - (subtraction), * (multiplication), / (Floating Point Division (or) Fractional Division), // (True Division), ** (Exponentiation), % (Modulus), & (Concatenation with strings)

EX:

1) $a = 7$

$b = 3$

$c = a + b$ ✓

2) $a = 7$

$b = \text{"GATE"}$

$c = a + b$ ✗

Invalid

3) $a = 14.25$
 $b = 3 + 4j$
 $c = 9$

$d = a + b - c$
 $= 8 + 4j$

4) $a = 14.25$

$b = 9$

$c = a - b$

$d = a + b$

$5.25, 23.25$

5) $a = 14.25$

$b = 3 + 5j$

$c = \text{"GATE"}$

$i = a + c$ # Invalid

$j = b + c$ # Invalid

6) $a = \text{"GATE"}$

$b = \text{"EXAM"}$

$c = \text{"2025"}$

$i = a + b + c$

$j = b - c$ # Error, Invalid

→ o/p:

GATEEXAM2025

7) $\text{str1} = \text{"GATE EXAM"} \neq \text{len}(\text{str1}) = 9$

$\text{str2} = \text{str1} * 3$

$\text{str2} = \text{GATE EXAMGATE EXAMGATE EXAM} \neq \text{len}(\text{str2}) = 27$

$\text{Print}(\text{len}(\text{str2}) - \text{len}(\text{str1})) \neq 27 - 9 = 18$

18



Arithmetic Operators



$$17/4 \Rightarrow 4 \overline{) 17} \begin{array}{r} 4.25 \\ 16 \\ \hline 10 \\ 8 \\ \hline 20 \\ 20 \\ \hline 0 \end{array}$$

$$4 \overline{) 17} \begin{array}{r} 4 \\ 16 \\ \hline 1 \end{array} \begin{array}{l} = Q \\ \\ = R \end{array}$$

A Sign	B Sign	A/B (or) A/B Sign	A/B Sign
-ve	+ve	-ve	+ve
-ve	-ve	+ve	-ve
+ve	-ve	-ve	-ve
+ve	+ve	+ve	+ve

Dividend divisor divisor sign

$/$, $//$, \div \rightarrow $/$ Performs division & return Quotient in float form always.
 \rightarrow $//$ " " " " " " as Per input type
 \rightarrow \div " " " " " " remainder as result.
 (type depends on input type)

Ex:1

$a = 17$
 $b = 4$
 $c = a/b \# 4.25$
 $d = a//b \# 4$
 $e = a\div b \# 1$
 $\text{Print}(c, d, e)$
 $\text{o/p} : 4.25 \quad 4 \quad 1$

Ex:2

$a = 17.6$ $4.0 \overline{) 17.6} \begin{array}{r} 4.4 \\ 16.0 \\ \hline 1.6 \\ 1.6 \\ \hline 0 \end{array}$
 $b = 4.0$
 $c = a/b \# 4.4$
 $d = a//b \# 4.4$
 $\rightarrow 4.0$
 $\text{Print}(c, d)$
 $\# 4.4 \quad 4.0$

Ex:3

$a = -17$
 $b = 4$
 $c = a/b - 4.25$
 $d = a//b - 4.4 \Rightarrow \text{Lower bound} = -5.0$
 $e = a\div b \quad 3$
 $\text{Print}(c, d, e)$
 $\# -4.25 \quad -5 \quad 3$

$a \div b$
 \Rightarrow For different Signs $\Rightarrow b - (a \div b)$
 $-17 \div 4$
 $\Rightarrow 4 - (17 \div 4)$
 $\Rightarrow 4 - (+1) \rightarrow \text{rem}$
 $\Rightarrow 4 - 1 = 3 \rightarrow \text{Divisor sign}$



== Vs =

- **==** is Relational Equal to
- **=** is Assignment Equal to
- **==** Compare Left and right Expression results.
return True if they are Equal,
False otherwise.
- **=** assign (or) copy Right Expression result
to Left Expression.
- **== > =**
High Priority Less Priority



Compound Assignment (or) Short-hand Assignment

$$\text{Exp1 Operator} = \text{Exp2} \\ \Rightarrow \text{Exp1} = \text{Exp Operator}(\text{Exp2})$$

Ex: $a += b \Rightarrow a = a + b$

$$a /= b \Rightarrow a = a / b$$

$$x \&= y \Rightarrow x = x \& y$$

$$i -= j - k \Rightarrow i = i - (j - k) \\ = i - j + k$$



Membership Operators



Membership Operators : `in`, `not in` \Rightarrow Use with Collection Types

Ex: `a = [12, 15, 10, 19]`
`b = 10`

`x = 15 in a`

`y = b not in a`

`Print(x, y)`

True False



Identity Operators



→ is, is not

a = [11, 22, 33] # ordered

b = [22, 11, 33]

c = {1, 3, 5, 7} # unordered

d = {3, 5, 7, 1}

x = a is b # false

y = c is not b # false

Print(x, y)

o/p: false, false

Operator Precedence and Associativity

Left to Right

{(,), [,], unary sign(+), unary sign(-)} > {*, *} > {/, //, %, *}> (+, -) >

~ , not L TO R

(<<, >>) > (!) > (^) > (!) > {<, <=, >, >=} > {==, !=, is, is not, in, not in}

L TO R L TO R L TO R L TO R L TO R

> and > or > {=, +=, /=, &= ...}

L TO R L TO R R TO L



Summary

— Operators

The word 'Thank' is written in a large, bold, yellow, cursive-style font. A thick yellow line starts from the top left, crosses the 'T', and extends horizontally across the top of the word, ending in an arrowhead pointing to the right.

Thank
THANK



Keep Hustling!