Data Science & Artificial Intelligence

Python For Data Science

Basics Of Python

Lecture No.- 05

By- Satya sir

Recap of Previous Lecture

- Logical operators : and, or, not
- Bitwise operators : &, |, ^, ~
- Shift operators : <<, >>

Topics to be Covered

- H/W Q's Solving
- Arithmetic operators
- == Vs =
- Operator Precedence and Associativity
- Control statements in Python

H/w Question - 1

a = 'True'

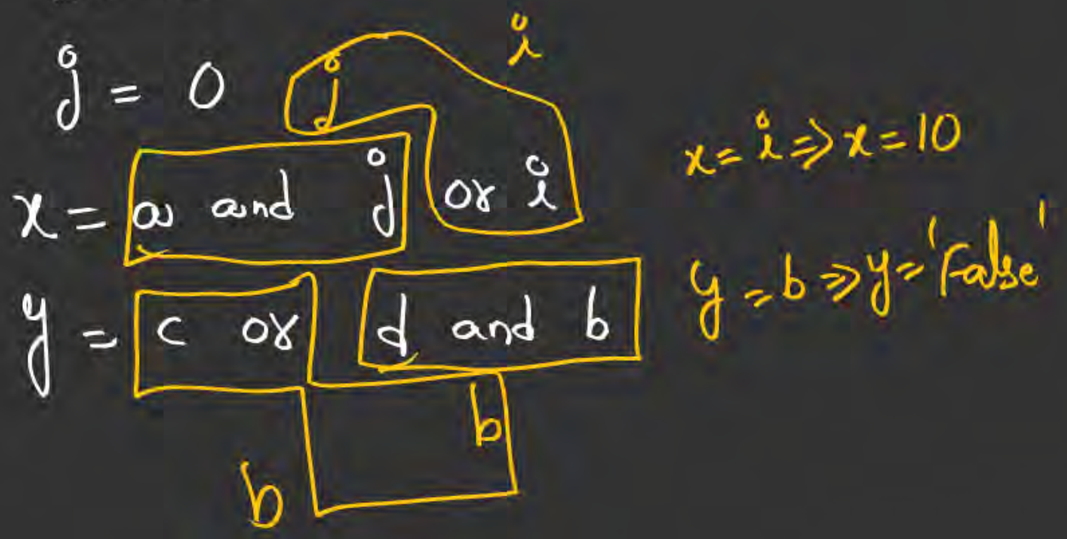
b = 'False'

c = False

d = True

i = 10

j = 0



x = i ⇒ x = 10

y = b ⇒ y = 'False'

H/W Question-2

$a = 13 = 0001101$

$b = 27 = 0011011$

$c = 34 = 0100010$

$d = 71 = 1000111$

$x = a \wedge b \,\&\, d$

$y = b \mid c \wedge a$

$z = c \mid d \,\&\, a$

Print(x, y, z)

14 63 39

$x =$ —

$b \& d = 0011011$
$\quad\quad\quad\; 1000111$
$\quad\quad\quad\; 0000011$

$\wedge a = 0001101$
$\quad\quad\quad\; 0001110$

$= 14$

$y =$ —

$c = 0100010$

$\wedge a = 0001101$
$\quad\quad\quad\; 0101111$

$\mid b = 0011011$
$\quad\quad\quad\; 0111111$

$= 63$

$z =$ —

$d = 1000111$

$\& a = 0001101$
$\quad\quad\quad\; 0000101$

$\mid c = 0100010$
$\quad\quad\quad\; 0100111$

$= 39$

Topic : Operators in Python

Arithmetic operators

Operators	Meaning
+	addition
−	Subtraction
*	Multiplication
/	Classic Division
//	True Division (or) floor Division
%	Modulus
**	Exponentiation

/ and //

divisor | Quotient / Dividend / Remainder

'//', '/' ⇒ Performs division among inputs and return Quotient as result.

'/' ⇒ Produce result always in fractional form.

'//' ⇒ Result type depends upon inputs type.

Ex: 1

```
a = 14
b = 3
c = a/b
d = a//b
print(c,d)
```

4.666666666666667 4

a/b ⇒

```
3)14(4.666
 12
 --
 20
 18
 --
 20
 18
 --
 20
 18
 --
 20
  :
```

a//b ⇒

```
3 | 14 | 4
    12
    --
     2
```

Topic : Operators in Python

Ex: 2

a = 14.4

b = 3

c = a / b # 4.8

d = a // b # 4.0

Print (c, d)

4.8 4.0

3 | 14.4 | 4.8
14.4
0

3 | 14.4 | 4.0
12.0
2.4

Ex: 3

a = 14

b = -3

c = a / b ⇒ -4.666 666 666 666 667

d = a // b ⇒ -4 ⇒ -5 (lower bound)

Print (c, d)

-4.666 666 666 666 667 -5

A	B	/ or //
+ve	+ve	+ve
+ve	-ve	-ve
-ve	+ve	-ve
-ve	-ve	+ve

Rounded off to it's Lower bound

Ex: 4

i = -29

j = 11

k = i / j # -2.6363..7

l = i // j # -3

Print (k, l)

11 | 29 | 2.6363
22
70
66
40
33
70
66
40
33

11 | 29 | 2
22
7

k = -2.636 363 636 363 63637

l = -2 ⇒ -3

Topic : Operators in Python

Ex: 5

```
x = -17.5
y = -3.5
i = x/y
j = x//y
Print(i, j)
# 5.0   5.0
```

$$3.5 \overline{)\,17.5\,} 4.0$$
$$14.0$$
$$\overline{3.5}$$

$\Rightarrow$ 4.0 $\Rightarrow$ rounded off to Greatest Integer Function

$\Rightarrow$ 5.0

Ex:6

```
i = 23.7
j = 4.3
k = i/j  => 5.5116...
l = i//j
```

$$4.3 \overline{)\,23.7\,} 5$$
$$21.5$$
$$\overline{2.2}$$

$\Rightarrow$ 5.0 $\xrightarrow{\text{gif}}$ 4.0

l = 5.0

Ex: 7

```
i = 16.5
j = 5.5
k = i/j  => 3.0
l = i//j => 3.0
```

%. (Modulus) operator

- When both input signs are same, It Perform division and return remainder with divisor Sign as result.

Ex:

a = 17	a = -17
b = 3	b = -3
c = a %. b	c = a %. b
c = 2	c = -2

3 | 17 | 5
15
2 remainder

When i/p's are of different sign

a %. b [a sign +ve, b sign -ve (or) a sign -ve, b sign +ve]

If 'a' is +ve, keep adding 'b' until result is -ve or Zero

If 'a' is -ve, keep adding 'b' until result is +ve or Zero

Ex: 17 %. -3 => 17+(-3)+(-3)+(-3)+(-3)+(-3)+(-3) = -1

-17 %. 3 => -17+3+3+3+3+3+3 = 1

$$a \,//\, b == b - (a \,//\, b)$$

$c = 17 \,//\, -3 \Rightarrow$ (a = 17, b = -3)

$\Rightarrow -3 - (17 \,//\, 3)$

$\Rightarrow -3 - (-2)$ (2 → divisor sign)

$\Rightarrow -3 - (-2) \Rightarrow -3 + 2 = -1$

$c = -17 \,//\, 3 \Rightarrow 3 - (17 \,//\, 3)$

$\Rightarrow 3 - (+2)$ (2 → divisor sign)

$\Rightarrow 3 - 2$

$\Rightarrow 1$

** (Exponentiation) : Right to Left Associative

Ex:1

a = 2

b = 3

c = a**b

Print(c) # $2^3 = 8$

Ex:2

a = 2

b = 3

c = 4

d = a**b**c

Print(d)

option-1 : (a**b)**c

$(a^b)^c$

$= (2^3)^4 = 2^{12}$

option-2 : a**(b**c)

$(a)^{b^c}$

$= (2)^{2^4} = 2^{81}$

(Comparison) == Vs = (Assignment)

a == b	a = b
Compares a and b Values and return True if same False - Not Same	Assign (or) copies 'b' value into 'a'.

Short-hand (or) Compound Assignment

$$X\ op = Y \Rightarrow X = X\ op\ (Y)$$

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

$i /= 2 \Rightarrow i = i / 2$

$j \& = k \Rightarrow d = j \& k$

Ex: $a = 5$, $b = 3$, $c = 2$, $d = 7$

d -= b-c-a

Print(d)

b-c-a = 3-2-5 = -4

d -= -4 ⇒ d = d-(-4)

= d+4

= 7+4 = 11

d = d - b-c-a

d = d-(b-c-a)

= d-b+c+a

= 7-3+2+5 = 11

Topic : Operators in Python

Precedence and Associativity

Operators	Associativity
() Highest precedence	Left - Right
**	Right - Left
+x , -x, ~x	Left - Right
*, /, //, %	Left - Right
+, -	Left - Right
<<, >>	Left - Right
&	Left - Right
^	Left - Right
|	Left - Right
Is, is not, in, not in, <, <=, >, >=, ==, !=	Left - Right
Not x	Left - Right
And	Left - Right
Or	Left - Right
If else	Left - Right
Lambda	Left - Right
=, +=, -=, *=, /= Lowest Precedence	Right - Left

Topic : Control Statements in Python - 1

#Q. What is the output of below Python code ?

```
i = 17 * 2 // 4 & 15 << 3 - 1 | 13 >> 2
print(i)
```

$17 * 2 = 34$

$34 // 4 = 8$

$3 - 1 = 2$

$15 << 2 = 15 * 2^2 = 60$

$13 >> 2 = 13 // 2^2 = 3$

$8 \& 60$

$8 = 00\,1000$

$60 = 11\,1100$

$00\,1000$

$= 8$

$8 | 3 = 1000 \mid 0011 = 1011$

$= 11$

$i = 11$

Topic : Control Statements in Python - 1

Control Statement ?

- The statement that can change (control), the order of execution of statements in a Program is called control statement

- 3 Types of control statements in Python :

1) Selection statements : if, if-else, if-elif, if-elif-else, match-case
2) Iterative statements : while, for
3) Jumping statements : break, continue, Pass, return

To be contd ...

2 mins Summary

→ Arithmetic operators

→ == Vs =

→ Compound Assignment

→ Precedence & Associativity

→ Control Stmts.

THANK - YOU