GATE

CRASH COURSE

Data Science & AI

Subject

Python - For Data Science Control Statements in Python

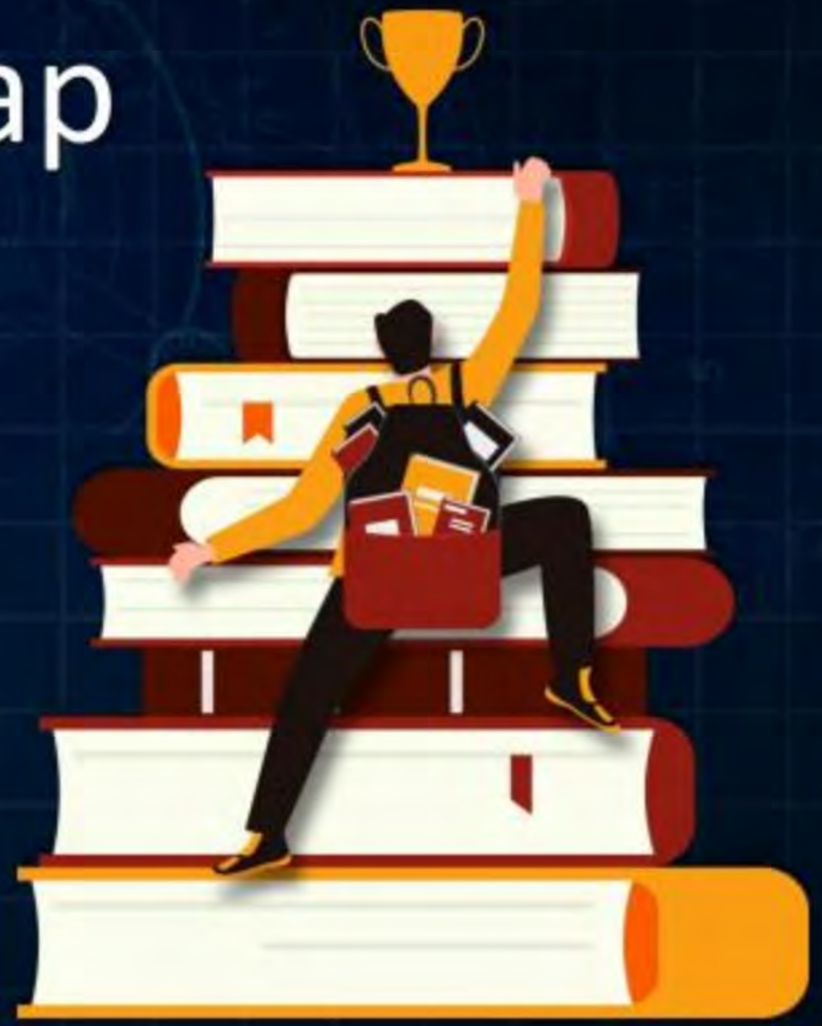
Lec No. 03

By – Satya Sir

Last Class

Quick Recap

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



Topics to be covered

1. Homework Question Solution
2. Types Of Control Statements
3. Selection Statements
4. Iterative Statements
5. Examples

Homework Question

x=37 =0100101
y=43 =0101011
z=69 =1000101
a=x & y ^ z
b=x ^ y | z
c= z | x ^ y
print (a,b,c)

x & y

```
  0100101
  0101011
  -------
  0100001
^ 1000101
  -------
  1100100
```

=100

```
x = 0100101
y = 0101011
  -------
^ = 0001110
z = 1000101
  -------
  1001111
```

=79

100 79 79

Operator 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

Operator Precedence and Associativity

$a = 17 \,\&\, 14 << 3 * 5 // 2 - 7 \;|\; 19 \wedge 13 ** 2 \,\&\, 11$

Print(a)

o/p = 0

$13 ** 2 = 13^2 = 169$

$3 * 5 = 15$

$15 // 2 = 7$

$7 - 7 = 0$

$14 << 0 = 14 * 2^0 = 14 * 1 = 14$

17 & 14

17 = 10001
01110
00000 = 0

169 = 1010 1001
& 11 = 0000 1011
0000 1001
19 0000 1001
0000 0000

0 | 0 = 0

Control Statements

- The Statement, that can Change (Control) the Order (Sequence) of Execution of other statements.

- Types of Control statements :

① Conditional (or) Selection (or) Decision making statements : if, if-else, if-elif, if-elif-else, match-case

② Iterative (or) Looping statements : while, for, while-else, for-else

③ Jumping (or) Un Conditional statements : break, continue, Pass, return

Control Statements

if, Nested-if, if-else, if-elif, if-elif-else

```
if (Expression):
    Statement(s)
```
TRUE

Tab space

if

```
if(Expression):
    Statement(s)
```

```
if(Expression):
    Statement(s) #optional
    if(Expression 2):
        Statement(s) #optional
        if(Expression 3):
            :
```

Nested-if

```
if(Expression):
    Statement(s) #TRUE
else:
    Statement(s) #FALSE
```

if-else

NOTE: else should be immediately after if.

Nested if-else

```
if(Expression 1):
    Statement(s)
    if(Exp 2):
        Statement(s)
        :
    else:
        Statement(s)

else:
    Statements(s)
    if(exp3):
        stmt(s)
    else:
        stmt(s)
```

if-elif

```
if(Exp1):
    stmt(s)
elif(Exp2)
    stmt(s)
elif(exp3)
    stmt(s)
    :
```

```
if(Exp1):
    stmt(s)
elif(Exp2):
    stmt(s)
    :
else:
    stmt(s)
```

if-elif-else

Control Statements

Match - Case

: only matched case statement(s) Executes.

```
match  expression :

    Case  object :
          statement (s)

    Case  object :
          statement (s)
          :
          :

    Case  _ _ :
          statement (s)
```

1. break is Not required
2. default case (Case _ _) is optional
3. To write Empty-cases, use 'Pass' statement
4. objects Can be of any Fundamental datatype
5. objects Can be in Random order.
6. Non-Contiguous object Values are allowed.
7. Duplicate-Cases are allowed. [First Matched Case gets Executed]
8. Cases Can be of Mixed type.

Question

```
a = 10
b = 0
c = 1
if a or b and c : # if(a) True
    Print('Hi')
Print('Bye') # Independant Print, outside if, executes always.
```

o/p:

A) Hi
B) Bye
C) Hi Bye
D) No output

Question

```
i = 'TRUE'   # Non-Empty String ==True
j = False    # Pre-defined Constant
k = 'FALSE'
l = True
m = None
if i :  #True
    if k : #True
        Print ( j )
    else :
        Print (m)
#end of if
```

```
elif l :
    if j :
        Print ( k )
    else :
        Print ( i )
```

o/p:

A) False B) TRUE C) None

c) True D) FALSE

Non-Zero numbers (±ve), True, Non-Empty strings = True

False, None, Zero number, Empty string = False

Question

```
i = int(input())
match i:
    case 2:
        pass
    case -2:
        print('Yellow')
    case 9:
        print('Blue')
    case 'D':
        print('Orange')
    case 99:
        print('Red')
    case _:
        print('white')
```

i	o/p
9	Blue
100	white
2	No output
-2	Yellow
68 (Unicode char == 'D')	Orange
C (Unicode Value = 99)	Red
5	white

Character	Unicode Value
'A'	65
'a'	97
'Space'	32
'0'	48

Question

Home WORK

```
Count = 1
i = 0
while i <= 10 :
    match i :
        case 3 :   count = count + 2
        case 1 :   count = count + 4
        case 7 :   count *= 2
        case 0 :   count -= 2
        case 6 :   count //= 2
        case 9 :   count += 3
        case _ :   count = count - 1
    i = i + 2
print(count)
```

Summary

- Control Stmt ?
- Types
- Selection Stmts :
  - if
  - if-else
  - if-elif
  - if-elif-else
  - match-Case

To be contd...

Keep Hustling!