

Bitwise operators :-

- It's works on binary of numbers. (Bit-level)
 - Number → Binary operation once performed.
Result is converted back to decimal.

operator meaning

example . Binary exp result

& (AND) If both bits are 1
5 & 3 $\begin{array}{r} 0011 \\ 0001 \\ \hline 0001 \end{array}$

opposite

ONI 2019-02-11 19:54:21

<< leftshift shift by 5 & 1 101<< 1010 → 10

>> Rightshift shift bits right (terminate) 5 >> 1 101 >> 1
10>2

(OR) If anyone buys 1

16
8
4
2
1

formula = 2^n

(13) > 2)
8421

13. \leftarrow 1101 → Binary. Right shift no value (ignore)

$$4 \times 0 + 1 = 6$$

right

$0 \oplus 1 \rightarrow (0, 1, 2, 3, 4, 5, 0, 1) = 4 \text{ bit} \rightarrow (0, 1, 2, 3, 4, \dots, 15) = 16$.

Always start from 0

Combinational logic :-

$$\alpha^0 = (0, 1) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

— 0 — 0 — 0 —

$$2^{-} = (0, 1, 2, 3)$$

10

Conditional Statement:-

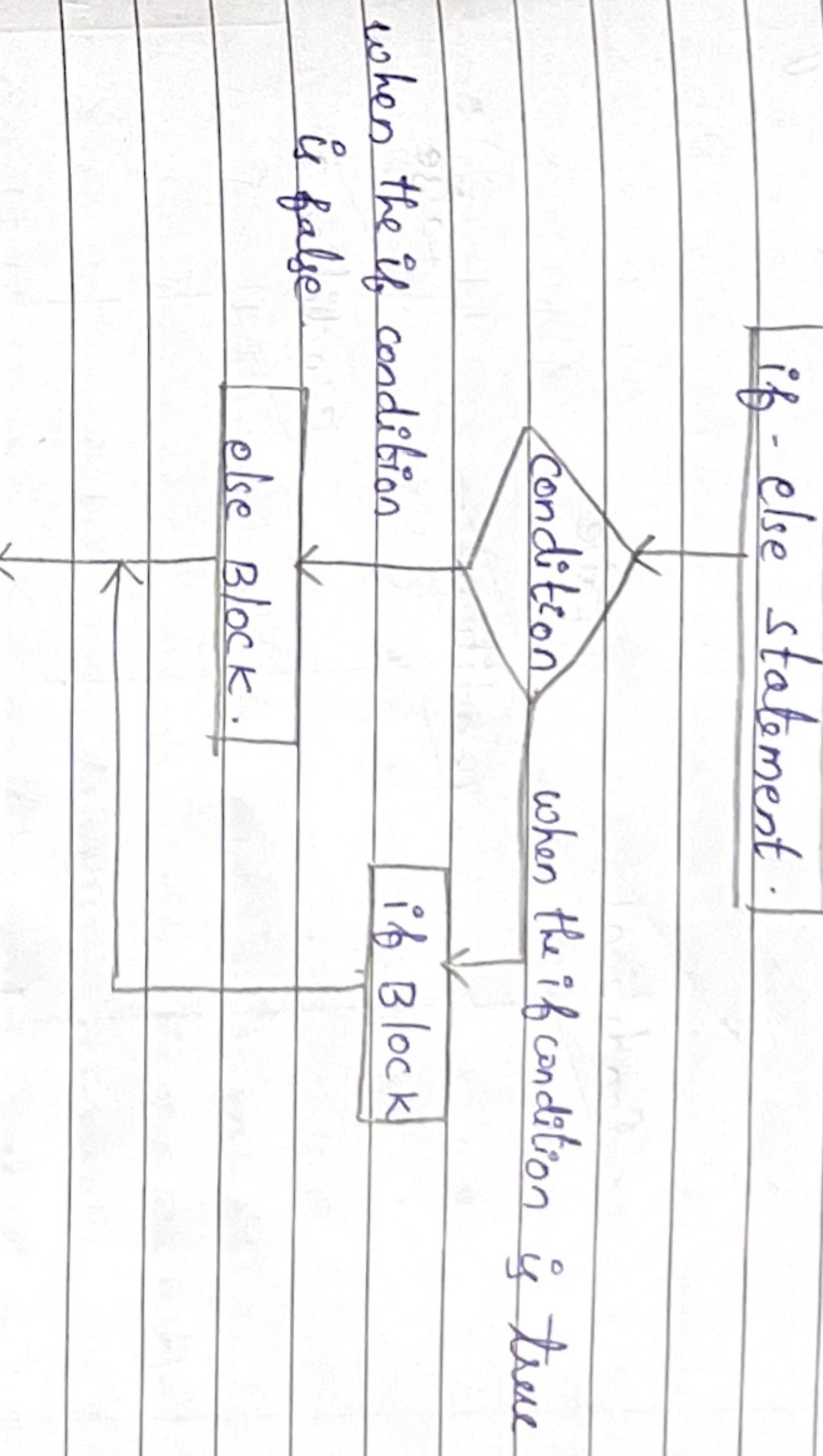
If condition:-

It allows us to make conditions in code. They check conditions (expression that results in T or F) and execute different blocks of code accordingly.

Types of conditional statements:-

- 1) If statement.
- 2) If...else...statement.
- 3) If...elif...else ladder.
- 4) Nested if -> using one if.

If statement:- [if statement] (single condition check)



If else statement provides two paths

one if condition is true another if false

Syntax:-

if (condition):
 Statement

else:
 Statement

execute a block only if the condition is true.

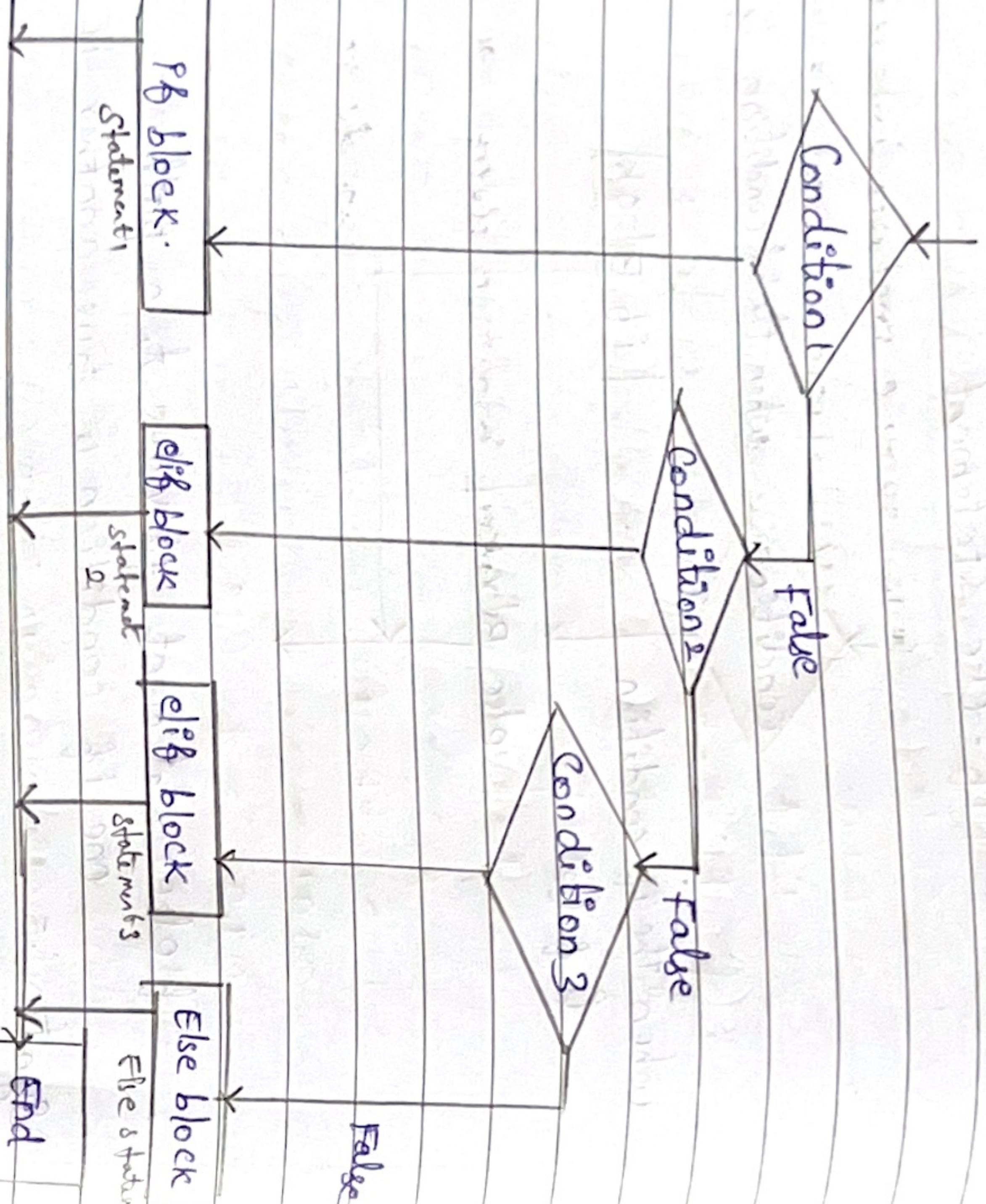
Syntax:-

```

if (condition):
    Statement
    if (x>5):
        print ("x is greater than 5")
    else:
        print ("x is not greater than 5")
  
```

(o/p):- x is not greater than 5

3) if...elif...else ladder - parallel [check condition one after another]
To check multiple conditions one by one



else: statements

use

- Nested if-else ladder :- When condition dependent one another.
- when one if is inside another if (or else), it is called a nested if-else. (Hierarchical) inside another.
- useful to check multiple conditions step by step.
- Dependent checks.

④ Become complex if deeply nested.

④ Step by step dependent conditions.

o

ex program:-
syntax:-

```

marks = 85
if (condition1):
    if marks >= 50 :
        if marks > 75:
            print ("Distinction")
        else:
            print ("First")
    else:
        print ("Second")
else:
    print ("Fail")
  
```

else is mandatory in all cases ?

No, else is optional in if, if-else, if-elif-else.

(05) multiple choices were but only one condition should run.

ex: program:-

- wrote a program to print eligibility for voting in India?

```
age = int(input("enter age:"))
Country = input("enter country:")
region = Country + "lowerc"
print(region)
if age >= 18 and region == "India":
    print("eligible to vote")
else:
    print("not eligible for vote")
```

O/P:-

```
Enter age: 26
```

```
India
```

```
Eligible for vote.
```

Q. positive or negative. of gives number.

```
a = input("enter a number:")
if (num > 0):
    print ("+ve")
else:
    if (num < 0):
        print ("-ve")
    else:
        print ("zero")
```

```
enter a number: -8
```

```
-Ve
```