3/9/2025 (day 1)

*Python : High level, interpreted, object oriented, independent dynamically is caued as Python

*Line code:
- complier execute all 100 lines at once
- interpreted execute is line by line execution

*Features of Python:
- simple syntax / easy to learn
- libraires
- open source
- Platform independent
- less memory
- execute faster compare to some Programmin

*Application of Python
- web development
- mobile app development
- machine learning / deed
- AI
- cyber security
- graphics
- IOT
- automation
- desktop app develoment... etc

4/9/2025 (day 2)

* comments:
They are part of code but they want consider in execution time

* Two type comment:
1. Single line comment # my 1st program
2. multi line comment ''' ... ''' """ ... """
or

* keywords:
They are reserved word uses for particular task and they cannot be used as identifier (varible, fun name..etc)

* variables:
it is place to store out values
ex: a = 5
a is variable 5 is values

* valid variable declaration:
a = 5
A = 7
num = 11
num1 = 89
num2 = 15
empid = 123
emp_name = "Rani" or 'Rani' (it support both)
_emp_no = 9901850122

* invalid variable declarations:

in - 35

stu #name@ = "arun"

stu id = 345

* Data types:

It's pre defined componet and specift the data categoey

- Persons = str
- name = str (alpha)
- age = int
- phone no = int (str)
- DoB = str (convert to date using libraries)
- height = float
- weight = float

* Python data types:

- int
- float 5.5 (all come under numeric)
- complex 5j+7
- str
- bool (all are single valve variable)

* special datatypes (data stucture)

- list
- tuple
- set
- dict

* a = 4; b = 5; c = 6 (single line u want to execute u can try)

a, b, c = 4, 5, 6

In Python line by line is execute

a = 4

b = 5

c = 7

* input function : input()

output function : Print

Print ("hello")

a = int(input ("enter a numbe"))

-> enter a number 3

take 2 float numbers from user

x = float(input ("enter a value :"))

y = float(input ("enter a value :"))

Print(x+y)

Print(x, "+", y, "=", x+y)

Print ("sum of :", x, "+", y, "=", x+y)

#. format method

Print ("sum of {} + {} = {}". format (x, y, x+y))

f starting method

Print f"sum of {x} + {y} = {x+y}")

5/9/2025 (day -3)

* operators:

1) arithmatic operators:

+ - * / % // **

ex: a = 5
b = 6
print (a+b)
print (a-b)
print (a*b)
print (a/b)

2) Relational operators:

> < >= <= == !=

ex: 5 > 12
false

5 < 7
true

3) Logical operators:

and or not xor xnor

A	B	AND	OR
fail	fail	false	false
fail	Pass	false	true
Pass	fail	false	true
Pass	pass	true	true

not (true) = false
not (false) = true

ex: print (3>5 and 12<7)
false

print (3<5 and 12>7)
True

④ Assignment operator:
= += -= *= /= %= // = ** =

ex: a = 5
a
=> 5

a += 3
a
-> 8

* special operator in Python:
① identity operators:
is , is not

ex: a = 9
b = 2
a is b
-> false

a is not b
-> True

8/9/2025 (day 4)

+ Membership operator
in not in

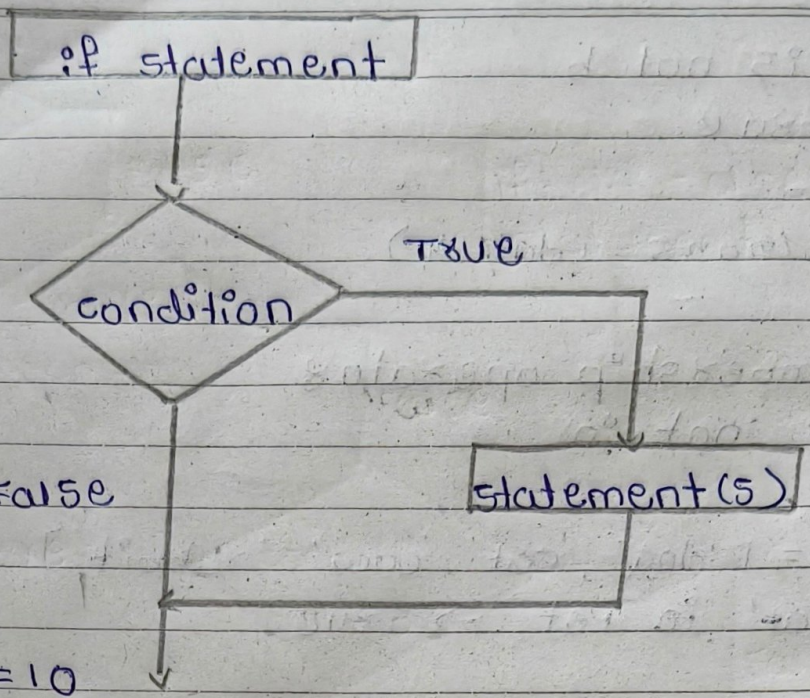
pet = ['dog', 'cat', 'cow', 'rabbit']
-> 'cat' in pet -> True

-> 'lion' not in pet
True

-> 'lion' in pet
false

* conditional statements:
it is allow us to make dicision in code. they check condition (expression that result in True or False) and execute diffent blocks of code accordingly

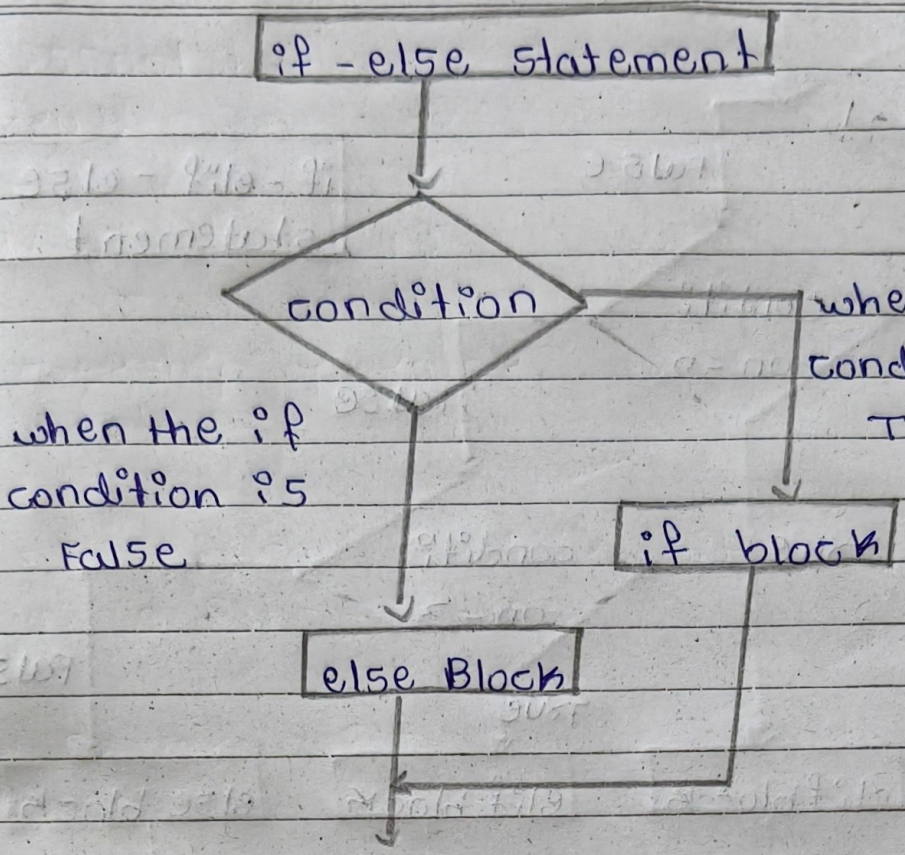
+ Types of conditional statements:
1) if statement - execute a black only if the condition is True
Sy:
if (conditional):
statement



example: x = 10
if (x > 5):
Print ("x is greater than 5")

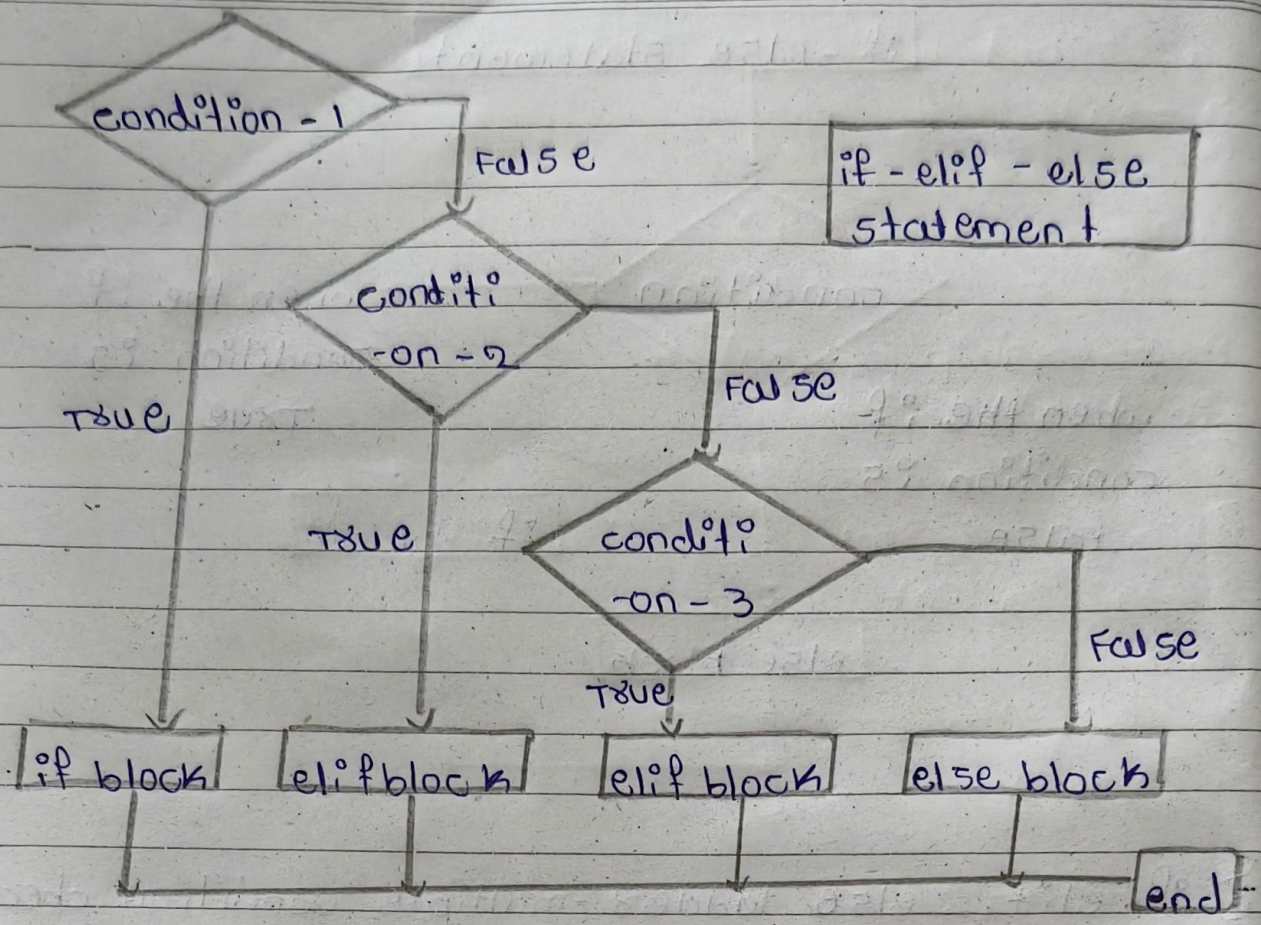
2) if... else statement - Provides two paths:
one if condition is true, another if false
Sy: if (condition)
statements
else:
statements

ex: x = 2
if (x > 5):
Print ("x is greater than 5")
else:
Print ("x is not greater than 5")



3) if. elif... else ladder - multiple condition checked one by one

Sy: if (condition):
    statement 1
elif (condition 2):
    statement 2
elif (condition 3)
    statement 5
else:
    statement 5


condition - 1
False
if - elif - else statement
Conditi -on - 2
False
True
True
conditi -on - 3
False
True
if block
elif block
elif block
else block
end