* Python

→ Python is a high level, interpreted, object oriented, dynamically typed programming language that uses indentation for code structure. (Guido Van Rossum)

→ It is widely used in data Science, AI, ML, web development & more.

1. High level language: Python is a high level language, it shadows away many low level details, allowing to focus on the logic of the program.

2. Interpreted language: Python code is interpreted line by line, making it easier to write and test code.

3. object Oriented: Python Supports object oriented programming (OOP) Concepts like Classes, objects, inheritance and Polymorphism.

4. Intended: Giving Space or tabs at the begining of the line of code

5. Dynamic typing: Python is dynamically typed, which means you don't need to declare the data type of a variable before using it.

• Features

1. Simple Syntax: Easy to learn and write, with a focus on readability

2. Open Source: Free to use, distribute and modify with a large Community Contributing with its growth.

3. Libraries: large collection of libraries and framework for various tasks Such as data analysis, web development and more.

4. Less memory: while python's memory can vary, it's often not the primary Concern for most applications.

5. platform independent: Runs on different operating systems, eg: windows, MacOS, Linux etc.

• Applications

1. web development: Django, Flask and Pyramid.

2. Gaming: Pygame and Panda 3D.

3. Machine Learning: Scikit - Learn, Tensor flow

4. Cybersecurity: used for penetration testing, vulnerability assessment and Security research.

5. Deep learning: Frameworks like Tensor flow, Keras and pytorch.

- Comments

→ A Comment is a line of code that is ignored by the interpreter. Comments are used to explain code; help others and urself to understand what the Code does.

Two types:

(i) Single line Comment.

```
# This is a Single line Comment.
Print ("Hello world!")
```

(ii) Multiline Comment.

```
""" This is a multiline Comment"""
print ("Comments are ignored by the interpreter").
```

Eg: """ write a Program to calculate Student results""".

- Keywords

→ These ~~words~~ are reserved words that have Special meanings and are used to define the Syntax and structure of the language. They Cannot be used as Variable names, function names or identifiers.

→ Example:

1. if, else, elif.
2. for, while.
3. def, class.
4. import, from.
5. try, except, finally.
6. True, false, none.

→ keywords define the syntax or structure of Python Code.

- Datatypes

In Programming, datatype is the type of data based on its format, size and set of values it can hold.

1. Integers (int): whole numbers like 1, 2, 3 etc.
2. float (float): Decimal numbers like 3.14, -0.5.
3. Strings (str): Text like "hello" or 'hello'.
4. Boolean (Bool): True or false values.
5. list (list): Ordered Collection of items [1, 2, 3].

→ Special data-type in Python.

1. str (string): A sequence of characters like "hello" or 'hello'.

2. list: An ordered collection of items, like [1, 2, 3] or ["a", "b", "c"]

3. tuple: An ordered immutable collection of items, like [1, 2, 3] or ["a", "b", "c"].

4. set: An ~~or~~ unordered collection of unique items, like {1, 2, 3} or {"a", "b", "c"}

5. dict (dictionary): An unordered collection of key value pairs, like {"name": "john", "age": 30}.

* VARIABLES

→ Variable is a name given to a value. (which we store our values)

eg: x = 10

'x' is a variable storing the value 10

Print (x) # Output: 10

→ Variables are case sensitive (x and X are different).

→ They must start with a letter or underscore (_).

→ The variable type is not needed to declare.

• Rules to declare variables:

→ Valid / variable declaration.

a = 2   A = 4

num = 20   num1 = 31   num2 = 90   num3 = 12.1

Stuid = 123

_stu_name = "Varsha"

→ Invalid variable declaration.

Initially we can't use numbers.

apart from _ we can't use any other special characters.

@num$ = 8.

* Datatype example.

name - string   age - int   pho - str   address - str   email - id - str

height / weight - float   DOB - str   work - str   salary - float.

eg: print ("Hello good morning!")

```
name = str (input ( "enter your name"))
age = int (input ("enter your age:"))
print (name, type (name))
print (age, type(age))
```

```
Hello good morning.
enter your name: Varsha.
enter your age: 23.
Varsha <class 'str'>
age <class 'int'>
```

* Operators

→ Operators are used to perform operations on variables and values.

→ It is a special symbole perform certain operation b/w operands

eg: a = 3

= is operator a, 3 operand.

→ Types of Operators

(i) Arithematic operators:

Arithematic operators are used with numeric values to perform common mathematical operations.

Operator	Name	eg:
+	Addition	x+y
-	Substraction	x-y
*	Multiplication	x*y
/	Division	x/y
%	Modulus (Divisible no)	x%y
**	Exponentiation	x**y
//	floor division.	x//y

(ii) Assignment Operators

Assignment operators are used to assign values to variables.

Operator	example	
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
//=	x //= 3	x = x // 3
**=	x **= 3	x = x*
&=	x &= 3	x = x & 3
|=	x|= 3	x = x| 3
^=	x^ = 3	x = x^3
>>=	x >>= 3	x = x >> 3
<<=	x <<= 3	x = x <<= 3

(iii) Comparison operators

Operator	~~example~~ Name	example
==	Equal	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

(vi) Membership operators : {"In", not In}

I am member of your fam : False.

I am notmember of your fam : True.

You are a member of your fam : True.

You are not a member of your fam : False.

* CONDITIONAL STATEMENT

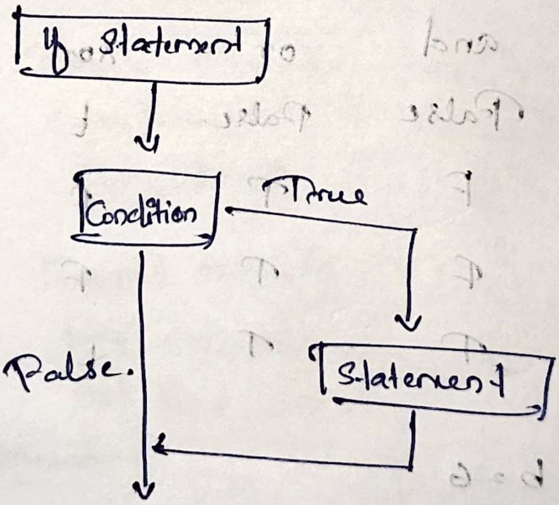
→ A Conditional Statment, is used to execute different blocks of code based on certain Conditions.

→ It allow use to make decission in Code., and execute diff blocks of Code accordingly.

• Types

(i) If Statement : Used to execute a block of code if a Condition is true.

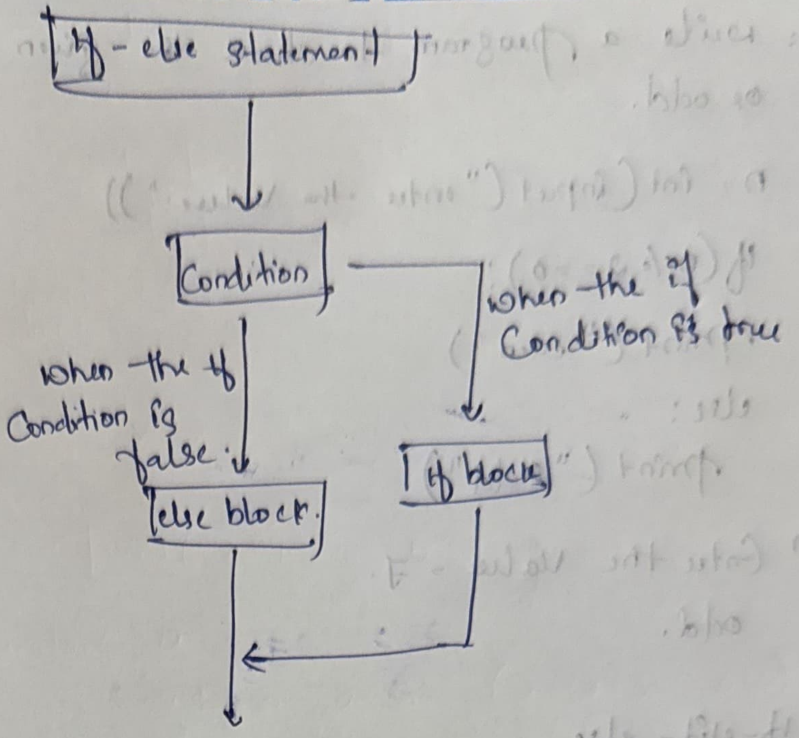
```
x = 10
if(x > 5):
   Print("x is greater than 5")
```



(ii) If else Statement:

Used to execute one block of code if a Condition is true and another block if its false.

```
eg: X = 10
    if X > 5:
  Print ("x is greater than 5")
   else :
    Print ("x is lessthan or equal to 5")
```

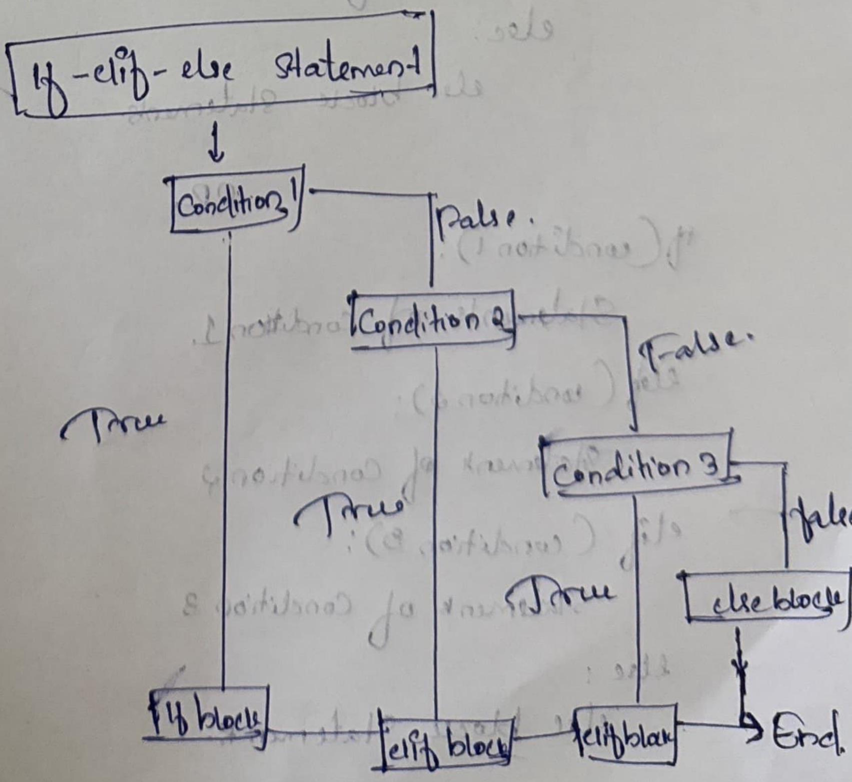

If - else statement
Condition
when the if Condition is true
when the if Condition is false.
If block
else block.

(iii) If - elif - else Statement :

used to check multiple Conditions and execute different blocks of Code based on which Condition is true.

eg: x = 20

```
If x >10:
  print("x is greater than 10")
elif x ==10:
  print("x is equal to 10")
else:
  print("x is less than 10")
```



eg: write a program to check the given number is even or odd.

```
n= int(input("enter the value:"))
if(n%2==0):
print("even")
else:
  print("odd")
```

→ Enter the value - 7.
odd.

→ if-elif-else

Syntax:

```
if (condition):
    Statement of condition 1
else:
    if (condition 2):
    Statement of condition 2
    else:
        if (condition 3):
        Statement of condition 3
        else:
            else block statements.
```

OR

```
if(condition 1):
    Statements of condition 1
elif (condition 2):
    Statements of condition 2
elif (condition 3):
    Statements of condition 3
else:
    else block statements.
```