→ Boolean :- Any expression that breaks down to true or false i.e 0 or 1.

⇒ Relational operators:

operand 1 (operator) operand 2 ⇒ True or False

1) Equality operator : It is represented by "=="

eg: 4 == 2 * 2 → True

name = "Akash"

name == "Jha" → False

2) Not Equal to (!=)

eg: 3 != 3 → True

3) Less than (<)

eg: 3 < 4 → True

4) Greater than (>)

eg: 3 > 4 → False

5) Greater than or equal to (>=)

eg: 3 >= 3 → True

6) Less than or equal to (<=)

eg: 3 <= 4 → True

→ If-statement

eg:
```
if 5<6:
    print("Yes, 5 is less than 6.")
```

⇒ Idendation (i.e 4 spaces) is very important in python program.

Syntax of if-statement:

```
if test expression:
    statement(s)
```

⇒ In if-statement something is true than it executes the statement otherwise it ignores it.

→ If-else statement

Syntax-

```
if condition:
    statement 1
else:
    statement 2
```

eg:
```
if 5>6:
    print("5 is greater than 6. Weird!")
else:
    print("5 is not greater than 6.")
```

output:- 5 is not greater than 6

⇒ Multiple-line comments

eg:
```
"""
This is a
multiple line
comment
"""
```

⇒ eg: Guess the game program

```
favourite_food = "Biryani"
ndime = input("what's your favourite food?")

if favourite_food == "Biryani":
    print("yep! so amazing")
else:
    print("yuck that's not it")

print("Thanks for playing")
```

→ Functions:- It is reusable block of programming statements to perform certain task.

Syntax - define
```
def function():
```
eg:
```
def say_hi():
    statement

function()
```

eg:
```
def say_hello():
    print("Hello friends")

say_hello()
```

⇒ ~~Arguments~~ Parameter :- It is specified after the function name, inside the parenthesis.
We can add as many ~~arguments~~ parameters as you want just seprate them with comma.

eg: def my_function (fname): → argument/parameter
print (fname + "Reference")

my_function ("Email")
→ argument

Syntax:- function_call (arguments):

⇒ A parameter is the variable listed inside the paranthesis in the function definition.

⇒ An argument is the value that is sent to the function when it is called.

→ Returning values from function

Syntax:- return my_value

A return statement consists of the return keyword and the value that function should return.

eg: A program to know should we deposit the money or not in our account

```
def withdraw_money (current balance, amount)
    if (current_balance >= amount):
        current_balance = current_balance - amount
        return current_balance
balance = withdraw_money (100, 80)
    if (balance <= 50):
        print ("We need to make a deposit")
    else:
        print ("Nothing to see here")
```