**Python:**

> High leval lang

> Scripting and OOPS

> Automation

> Built on C

**Data Types:**

>Int

>Str

>Float

>Boolean

>Complex

**variables :** Used to store data in the forms of data types

> In python no need to declare the variables

> We can just assign the value in the form of data-types

> Re-usabilty

**Functionality's:**

> User based > Devoloped by devolopers

> Standerd > Devoloped by python devolopers > These are stored in cache

Ex: Import sys

**Data Processing :** performing operations by using operators on top of data is called as data processing

> we can done the data processing or oprations by using operators

**Operators :** operators are 8 types

1. Arthematic (+,-,\*,/)

2. Relational (<,>,<=,==,!=) \*always result must be boolean \* Used for comparision

3. Logical (and, not, or) \*Used for combinations

4. Bitwise

5. Assignment

6. Membership

7. Identity

8. Ternary

**Interpretaion:** conversion of high leval lang to low leval is called interpretation

**Binary Operators:**

1. to convert "int" to "binery oparator" , we can use function called "bin".

ex: age = 27

bin\_val = bin(age)

print(bin\_value)

2. to convert "binary" value to "int" we can use function called "int".

ex: int\_val = int("bin\_val",2)

print(int\_val)

3. to convert into character we can use the function called "chr".

ex: FL = 's'

inter\_val = ord(FL)

bin\_val = bin(inter\_val)

int\_val = int("1110011",2)

chr\_val = chr(int('1110011',2))

print(chr\_val)

4. Bit-wise "And" : bit-wise "and" refers, in both levels it will need one.otherwise in results all Zero

5. Bit-wise "Or" : bit-wise "or" refers, in at any single stage it will need one. it results all Zero.

**Assignment Operator:**

x = 10

x\*=1

x/=1

x-=1

x+=1

print(x)

**Membership Operator:**

x=sandeep.burri@cyient.com

print('@' in x)

print("@" not in x)

**Identity Operator:** We can check the they are targetting same memory or not.

x=4

y=3

print(id(x))

print(id(y))

instead of above syntax we can directly use below operators for identity operators.

print(x is y)

print(x is not y)

**@@@ Useally python control flows from TOP to BOTTOM**

**@@@ To manage these flow, we can use conditions..**

**@ if you want to move right hand side, we can use "IF" condition**

**@ if you want to move left hand side we can use "ELSE" condition**

**@python is case sensitive.**

**Conditions:Condition can have**

1. Relational Expression

2. Logical Expression

3. Value's (True, False,'',111)

**Conditions are classified based on result:**

1.Passed condition - condition whih meant to pass the condition.

2. Failed Condition - condition is setted to fail and the result is fail, none, 0

**Condition With Relational Expression:**

x = 30

y = 20

print(x > y)

print(x>y)

print(x==y)

print(x!=y)

**Condition with relational and logical expression:**

print(x>y and x!=0)

print(x==y or x<y)

**Condition with value:**

py = 'python'  
if py:  
 print("yes, python")  
else:  
 print("No, in-correct")

x= 1000  
y=2000  
if (x>y):  
 print('x is greater than y')  
else:  
 print('x is not greater than y')