Operators:

Operator is a symbol it is used to perform some operation on the operands

Operators are classified into 2 types

1.unary operators

> An operator which is used to perform some action[operation] on a operand

Eg: From C and Java

[Increment ++ | Decrement --]

Note: Unary operators are not supported in python

```
>>> x=10
>>> x++
SyntaxError: invalid syntax
>>> y=20
>>> ++y
20
>>> +++++++++++++++++++++++++++++y
20   # here +++++++ symbols are considered as sign only, but it doesn’t perform any action
```

2.Binary Operators

An Operator which is used to perform operation on more than one operand

1.assignment operator

* Assignment operator is “=”

* Assignment always from right to left

```
in C and Jav                                   Python
int x=10;                                       x=10
x=10+20;                                        x=10+20
----------------------------------
int x,y,z;
```

```
x=10;
y=10;
z=10;
x=y=z=10;                                              x=y=z=10
---------------------------------------------------
int x;
float y;
x=10;
y=3.14f;                                               x,y=10,3.14
```

#In python swapping of two numbers

```
a,b=10,20
print("Before interchange “)
print(“a:",a,"b: ",b)
a,b=b,a
print("After Interchange ")
print("a: ",a,"b : ",b)
```

#Arithmetic Operators

```
x=10
y=5

print("x=10 and y=5")
print("x+y ? : ",x+y) #15
print("x-y ? : ",x-y)   #5

print("x*y ? : ",x*y)  #50
print("x/y ? : ",x/y)   #2.0

print("x%y ? : ",x%y) #0
```

/ vs //

/ --> always used to return the result in the form float type only

// --> used to return the result based on the operand types ,Here if any operand is float then "Result will be in the form float only"

```
print(" 10/ 3 ? : ",10/3)   #3.33333 <class 'float'>
print(" 10/3.0 ? : ",10/3.0)
print(" 10.0/3 ? : ",10.0/3)
print(" 10.0/3.0 ?:",10.0/3.0)

#// floor div          ---->   10/3 --> 3.33333333
                              #     10//3 -->3
# int // int --> int
print("10//3 ? : ",10//3)  #3  --> <class 'int'>
# float // int --> float
print("10.0//3 ?:",10.0//3) # 3.0
# int // float --> float
print("10//3.0 ? : ",10//3.0) #3.0
# float // float --> float
print("10.0//3.0 ? : ",10.0//3.0) #3.0
```

# * vs **

```
print("10*2 ? : ",10*2)
print("10*10 ? : ",10*10)

print("10**2 ? : ",10**2)
print("10**3 ? : ",10**3)
print("2**3 ? : ",2**3)
```

Relational operators:

These are used to check the relation among two operands and return Boolean value is True or False

```
x,y=10,5

print("x=10 and y=5")
print("x>y ? : ",x>y) #True
print("x>=y ? : ",x>=y) #True

print("x<y ? : ",x<y) #False
print("x<=y ? :",x<=y) #False

print("x==y ? :",x==y) #False
print("x!=y ? : ",x!=y)  #True
```

Compound operators | short cut operators
Both Arithmetic and Assignment Operators

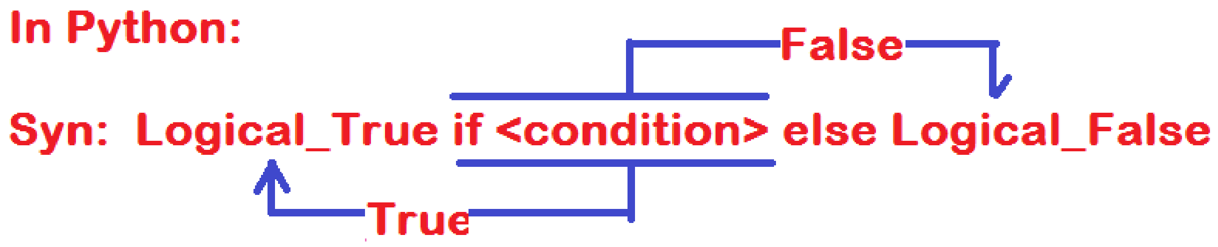
Operator	Action	Meaning
+=	x+=10	x=x+10
-=	y-=10	y=y-10
/=	x/=10	x=x/10
//=	y//=10	y=y//10
*=	x*=10	x=x*10
**=	x**=3	x=x**3

#Examples on Short had operators

```
x=10
x+=10  #x=x+10
print("x val is : ",x)

y=20
y/=10 #y=y/10
print("y val is : ",y)

z=20
z//=10
print("z val is : ",z)

x=10
x=x+1 #x+=1
    #x++ Syntax Error
x=x-1 #x-=1
   #x-- syntax Error
```

Conditional Operator :

It will check the condition first, If Condition evaluate True then it will be execute Logical True . If the Condition is False then it will execute Logical False



In Python:

Syn: Logical_True if <condition> else Logical_False

```
x,y=10,20
print(" x is big") if x>y else print("y is big")

Eg:
marks=90
print("PASS ") if marks>34 else print("FAIL")
```

Membership operator:

* Membership operators are used for iterable objects

* Iterable objects are nothing but Collections [str | list | tuple | set ….]

IN :

- It will return True if specified Object is the member of iterable Object

NOT IN

- It will return True if the specified Object is not the member of iterable Object

#in

```
#Syn : object in iterable

lst=[10,20,30,40,50]
print(lst)
res=30 in lst
print("Result is : ",res)
print("10 in lst ? : ",10 in lst)
print("60 in lst ? : ",60 in lst)
```

#not in

```
print("60 not in lst ? : ",60 not in lst)
print("50 not in lst ? : ",50 not in lst)

#Eg 2:
s="welcome to sssit"
print("data of s : ",s)
print("P in s object : ",'p' in s)
print("o in s object : ",'o' in s)
```

#keyword module

```
# kwlist ---> variable of type <class 'list'>
# iskeyword( ) --> function from keyword

import keyword
print(keyword.kwlist)

print("import is keyword ? : ",'import' in keyword.kwlist)
print(keyword.iskeyword('import'))
```

Identity operator:

These are used compare id()'s of the Objects

Is

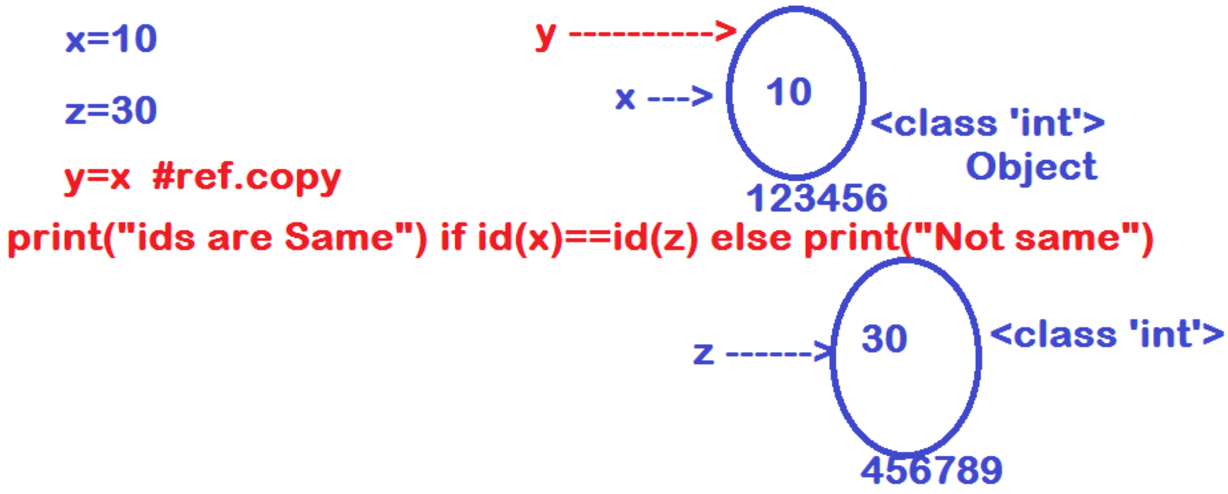
- It will returns True, if id’s of both Objects are same

Is not

- It will returns True, if id’s of both Objects are not same

Ex1:

```
x=10
z=30

#compare values of x and z
print("both are same") if x==z else print("Not Same")

#id() of both Objects
print("id(x) is : ? ",id(x))
print("id(z) is : ? ",id(z))
print("same ") if id(x)==id(z) else print("not same")

#is
print("id(x) and id(z) are Same ? : ",x is z)
```



```
x=10

z=30

y=x  #ref.copy
print("ids are Same") if id(x)==id(z) else print("Not same")
```

Ex2:

```
x=10
y=x #ref.copy
z=30

print("id(x) ? : ",id(x))
print("id(y) ? : ",id(y))
```

```
#is
print("id(x) and id(y) are Same ? : ",x is y)
print("id(x) and id(z) are Same ? : ",x is z)

#is not
print("id(x) and id(z) are not same ? : ",x is not z)
```