Operators

1 Arithematic operators

2 Assignment "

3 Logical "

4 Identity "

5 Membership "

6 Bitwise "

7 Relational "

8 operator precedence

1. **Arithematic operators** :- It perform comman mathematical operation on the numeric operands.

→ The arithematic operators return the type of result depends on the type of operands as below:-

i) if either operand is a complex number the O/P is converted complex.

ii) if either operand is floating point number the result is converted to floating point;

iii) if both operands are integers then

the result is an integer and no conversion is needed.

Operator	Description	Syntax
+	Addition : Add two operands	x+y
−	Subtn : Subtract " "	x − y
*	Multn : Multi " "	x * y
/	Division (float) : Divide first operand by second.	x/y
//	Division (floor) : Divide 1st oper and by second	x//y
%	Modulus : returns the remainder when 1st operands divided by the 2nd	x % y
**	Power : Return first raised to power second	x ** y

Assignment operator :- The assignment operators are uses to assign values to variables.

List of assignment operator :-

Operator	Example	equivalent to
=	X = 5	X = 5
+=	X += 5	X = X+5
-=	X -= 5	X = X−5

*=	x*=5	*=x*5
/=	x/=5	x=x/5
%=	x%=5	x=x%5
//=	x//5	x= x//5
**=	x**=5	x=x**5
&=	x&=5	x=x&5
|=	x|=5	x= x|5

3. **Logical Operators :-**

This operator is uses to combine two boolean expr^n. It is applicable to all objects and support truth tests, identity tests and boolean operat^n.

i> and ii> or iii> not

a & b both true — only 1 or 2 or bot true — not a & b is False

4. **Python Membership Operators :-**

This operator is uses to test for membership in a sequence, such as strings, lists or tuples.

Operators	Description	Example
in	Evaluate to true if it finds a variable in the specified sequence and false otherwise	x in y, he in result in 1 if x is a me ber of sequence

not in	Evaluates to true if it does not finds a variable in the specified sequence and false otherwise	x not in y, here not in result in a 1 if x is not a member of sequence y.

Comparision or Relational Operator :-
This operator is uses to compare the values of both side and decide relation among them.

operator	Example
==	(a==b)
!=	
<>	
>	
<	
>=	
<=	

identity operator :-
is and is not.

'is' operator :- Evaluates to 'true' if the variable on either side of the operator point to the same object and otherwi 'false'.

```
x = 5
if( type(x) is int):
        print (" true ")
else :
        print (" false ")
```

11) is not :-> Evaluates to false if the variable on either side of the operator point to the same object and otherwise 'true'.

```
x=5.2
    if (type(x) is not int):
    print ("true")
  else
      print ("false")
```

3. Bitwise operator:-

It uses to perform bitwise calculn on integers.

-> In this first number is converted into binary form and then opern is performed on bit by bit.

Page :
Date : / /

Frozen set ( ) → It is an inbuilt ( ) which uses to change mutable values into immutable value i.e → The elements of sets is not added or remove once created

Syn:- frozenset (iterable_object_name)

Ex:-
```
nu = {1, 2, 3, 4, 5 ,6, 7 ,8 ,9}

frum = frozenset(nu)
print ("frozenset object is :" , frum)
```

o/p
```
frozenset object is : frozenset({1, 2, 3, 4, 5, 6, 7, 8, 9})
```