Ques. Python Operators?

- Arithmetic Operators
- Assignment operators
- Comparison operators
- Logical operators
- Identity operators
- Membership operators
- Bitwise operators
- Arithmetic Operators

Operator	Name	Example
+	Addition	x + y
-	Subtraction	x-y
*	Multiplication	x * y
/	Division	x / y
%	Modulus	x % y
**	Exponentiation	x ** y
//	Floor division	x // y

- Assignment operators

Operator	Example	Same As	Try it

=	x = 5	x = 5	x = 5
			print(x)
			5

+=	x += 3	x = x + 3	x = 5
			x += 3
			print(x)
			8

-=	x -= 3	x = x - 3	x = 5
			x -= 3
			print(x)
			2

```
*=  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 | 3
^=  x ^= 3  x = x ^ 3
>>= x >>= 3 x = x >> 3
<<= x <<= 3 x = x << 3
```

- Python Comparison Operators

Ques. difference between membership and identity operators?

- **Membership operators:-** Membership operators are operators used to validate the membership of a value.

1. **in operator :** The ‘in’ operator is used to check if a value exists in a sequence or not.

```
x = ["apple", "banana"]
print("banana" in x)

Output:- True
```

```
list1=[0,2,4,6,8]
list2=[1,3,5,7,9]

check=0
for item in list1:
    if item in list2:
        #Overlapping true so check is assigned 1
        check=1

if check==1:
    print("overlapping")
else:
    print("not overlapping")

Output:- not overlapping
```

2. **'not in' operator-** Evaluates to true if it does not finds a variable in the specified sequence and false otherwise.

```
x = ["apple", "banana"]
print("pineapple" not in x)
```

```
Output:- True
```

```
a = 70
b = 20
list = [10, 30, 50, 70, 90 ];

if ( a not in list ):
 print("a is NOT in given list")
else:
 print("a is in given list")

if ( b not in list ):
 print("b is NOT present in given list")
else:
  print("b is in given list")

Output:-
a is in given list
b is NOT present in given list
```

- **Identity operators** evaluate whether the value being given is of a specific type or class. These operators are commonly used to match the data type of a variable.

1. **is operator:-** The is operator returns true if the variables on either side of the operator point to the same object. Otherwise, it returns false.

```
x = 'Educative'
if (type(x) is str):
    print("true")
else:
    print("false")
Output:- True
```

2. **is not operator:-** The is not operator returns false if the variables on either side of the operator point to the same object. Otherwise, it returns true.

```
x = 6.3
if (type(x) is not float):
    print("true")
else:
    print("false")

Outpur:- False
```