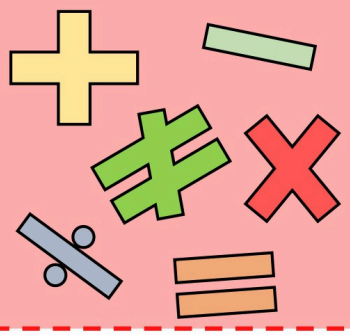
Operator

In Python, an operator is a symbol that determines the action that is to be performed or considered. We will learn about 4 main types of operators in python –

1. Arithmetic Operators
2. Assignment Operators.
3. Relational Operators
4. Logical Operators.
5. Identity Operators
6. Membership Operators.



1. Arithmetic Operators: Operators that can perform mathematical operators such as Addition, Subtraction, Multiplication, Division, and Modulus.

Operator	Description	Syntax
+	**Addition:** adds two operands	x + y
-	**Subtraction:** subtracts two operands	x - y
*	**Multiplication:** multiplies two operands	x * y
/	**Division:** divides the first operand by the second	x / y
%	**Modulus:** returns the remainder when first operand is dividend by the second	x % y

2. Assignment Operators: Assignment Operators are used in assigning values to the variables.

Operator	Description	Syntax
=	Assign value of right side of expression to left side operand	x = y
+=	Add AND: Add right side operand with left side operand and then assign to left operand	a += b a = a+b
-=	Subtract AND: Subtract right side operand with left side operand and then assign to left operand	a -= b a = a-b
*=	Multiply AND: Multiply right side operand with left side operand and then assign to left operand	a *= b a = a*b
/=	Divide AND: Divide right side operand with left side operand and then assign to left operand	a /= b a = a/b

3. Relational Operators: Relational Operators compare the values and either returns True or False according to the given condition.

Operator	Description	Syntax
>	**Greater than:** True if left operand is greater than the right.	x > y
<	**Less than:** True if left operand is less than the right.	x < y
>=	**Greater than or equal to:** True if left operand is greater than or equal to the right.	x >= y
<=	**Less than or equal to:** True if left operand is less than or equal to the right.	x <= y
==	**Equal to:** True if both operands are equal.	x == y
!=	**Not equal to:** True if operands are not equal.	x != y

4. Logical Operators: Logical Operators perform Logical AND, Logical OR, and Logical NOT operations.

Operator		Description	Syntax	
and	&&	**Logical AND:** True if both the operands are true	x and y	p && q
or	||	**Logical OR:** True if either the operands are true	x or y	p || q
not	!	**Logical NOT:** True if operand is false	not x	! p

p	q	p and q
1	1	1
1	0	0
0	1	0
0	0	0

p	q	p or q
1	1	1
1	0	1
0	1	1
0	0	0

p	not p
1	0
0	1

5. Identity Operators: Sometimes, in Python programming, need to compare the memory address of two objects; this is made possible with the help of the identity operator.

Identity operators are used to check whether both operands are same or not. Python provides 'is' and 'is not' operators which is called identity operators and both are used to check if two values are located on the same part of the memory. Two variables that are equal do not imply that they are identical.

Operator	Description	Syntax
is	Return true, if the operands are same. Return false, if the operands are not same	a = 3 ; b = 3 print(a is b) True
is not	Return false, if the operands are same. Return true, if the operands are not same	a = 3 ; b = 3 print(a is not b) False

Key differences:
- 'is' checks for object identity (memory address), while '==' checks for value equality.
- 'is' is used to verify if two variables point to the same object, while '==' is used to verify if two values are equal.

```
a = [1, 2, 3] ; b = [1, 2, 3]
c = a

print(a == b) # Output: True (values are equal)
print(a is b) # Output: False (different objects in memory)
print(a is c) # Output: True (same object in memory)

Memory addresses:
print(id(a)) # Output: 1040 (memory address of a)
print(id(b)) # Output: 1096 (memory address of b)
print(id(c)) # Output: 1040 (memory address of c, same as a)
```

6. Membership Operators: The membership operators in Python programming are used to find the existence of a particular element in the sequence and used only with sequences like string, tuple, list, dictionary etc.

Membership operators are used to check an item or an element that is part of a string, a list or a tuple. A membership operator reduces the effort of searching an element in the list.

Python provides 'in' and 'not in' operators which are called membership operators and used to test whether a value or variable is in a sequence.

Operator	Description	Syntax
in	True if value is found in list or in sequence, and false if item is not in list or in sequence.	x = "Hello World" print('H' in x) True
not in	True if value is not found in list or in sequence, and false if item is in list or in sequence.	x = "Hello World" print('H' not in x) False