Operators:

- Operators are special symbols or keywords used to perform operation on one or more values (operands).

o Arithmetic assignment operators:

Operators		Example	Same as
=	(Assignment)	x = 5	x = 5
+=	(Add and assign)	x += 3	x = x + 3
-=	(Subtract and assign)	x -= 3	x = x − 3
*=	(Multiply and assign)	x *= 3	x = x * 3
/=	(Divide and assign)	x /= 3	x = x / 3
%=	(Modulus and assign)	x %= 3	x = x % 3
**=	(Exponentiation and assign)	x **= 3	x = x ** 3
//=	(Floor divide and assign)	x //= 3	x = x // 3

o Comparison Operators:

Used to compare two values, returning a Boolean (True or False) results.

Let x = 4 and y = 5

Operators	Example	Result
==	x == y	False
!=	x != y	True
>	x > y	False
<	x < y	True
>=	x >= y	False
<=	x <= y	True

o Logical Operators:

Used to conditional statements

Operator	Example	Description
and (Logical AND)	(condition1) and (condition2)	TRUE → If both conditions satisfies FALSE → if any of the condition(s) is/are False
or (Logical OR)	(condition1) or (condition2)	TRUE → of any or all conditions are satisfied FALSE → if all conditions are not satisfied
not (Logical NOT)	not (condition)	Invert the Boolean result

o Membership Operators:

Used to test if a sequence contains a specific value for e.g. strings, lists

Operator	Example	Description
in (Is present in)	x in y	Returns TRUE if a sequence with the specified value is present in the object, otherwise FALSE
not in (Is not present in)	x not in y	Returns TRUE if a sequence with the specified value is not present in the object, otherwise FALSE

o Identity Operators:

Used to check of two variables refers to the same object in memory

Operator	Example	Description
is (is the same object)	x is y	Returns TRUE if both variables are the same object, otherwise FALSE
is not (is not the same object)	x is not y	Returns TRUE if both variables are not the same object, otherwise FALSE

```
>>> x = [1,2,3]
>>> y = x
>>> z = [1,2,3]
>>> print(id(x))
2408268941440
>>> print(id(y))
2408268941440
>>> print(id(z))
2408268885440
>>>
>>> x is y
True
>>> x is z
False
>>> x == z
True
```