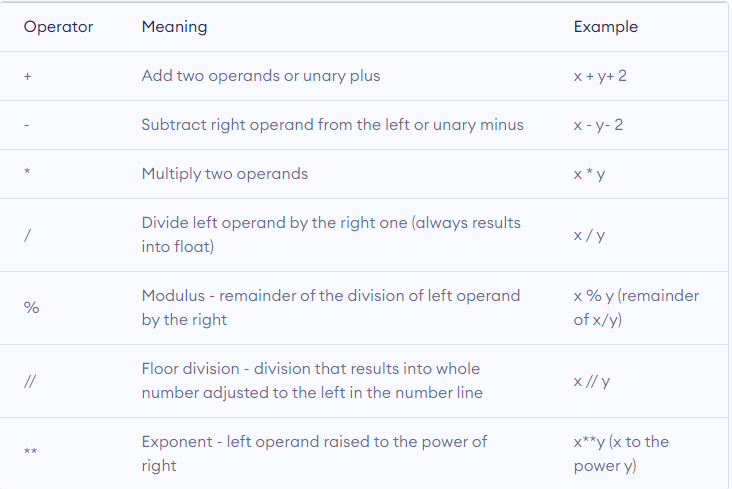
# **What are Operators?**

In computer programming, **operators** are constructs defined within **programming languages** which behave generally like **functions**, but which differ **syntactically or semantically**.

Common simple examples include arithmetic (e.g., addition with +), comparison (e.g., "greater than" with >), and **logical** operations (e.g., AND, also written && in some languages). More involved examples include **assignment** (usually = or:=), **field** access in a record or **object** (Usually.), and the **scope resolution operator** (often:: or.). Languages usually define a set of built-in operators, and in some cases allow users to add new meanings to existing operators or even define completely new operators.

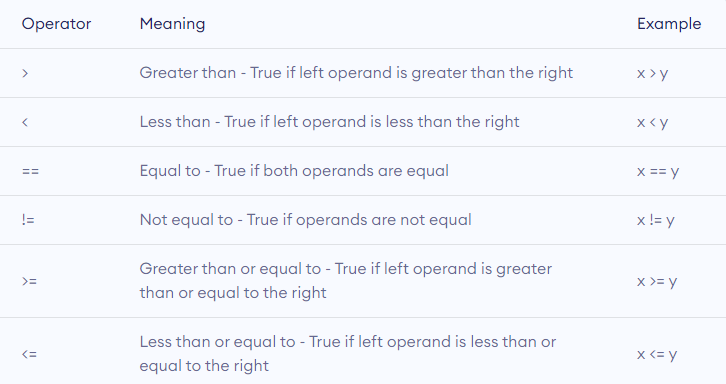
## Arithmetic operators

Arithmetic operators are used to perform mathematical operations like addition, subtraction, multiplication, etc.



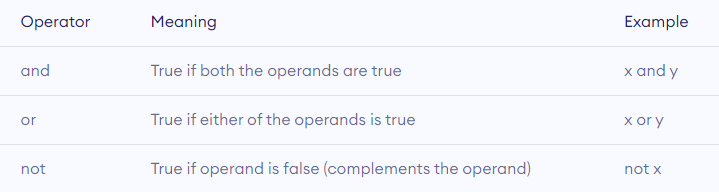
## Comparison operators

Comparison operators are used to compare values. It returns either True or False according to the condition.



## Logical operators

Logical operators are the and, or, not operators.

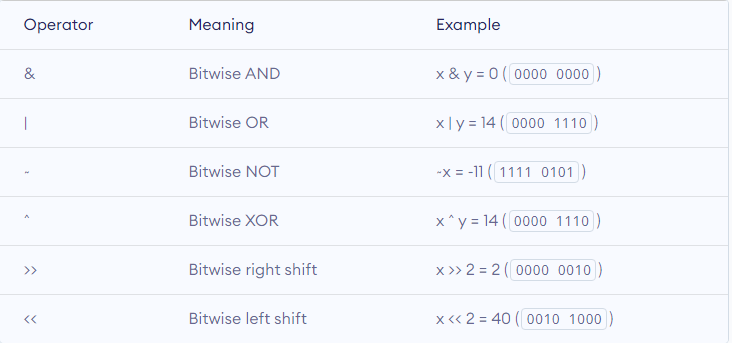


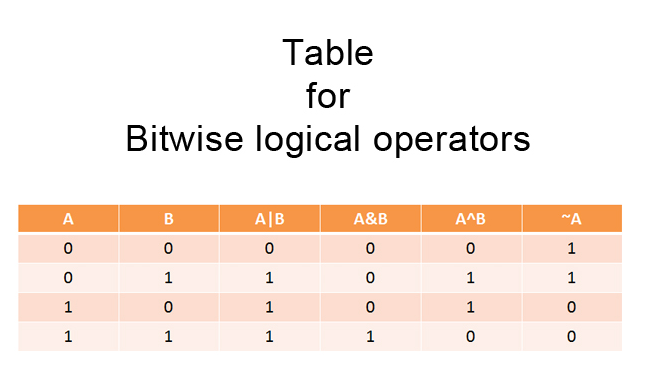
## Bitwise operators

Bitwise operators act on operands as if they were strings of binary digits. They operate bit by bit, hence the name.

For example, 2 is 10 in binary and 7 is 111.

**In the table below:** Let x = 10 (0000 1010 in binary) and y = 4 (0000 0100 in binary)



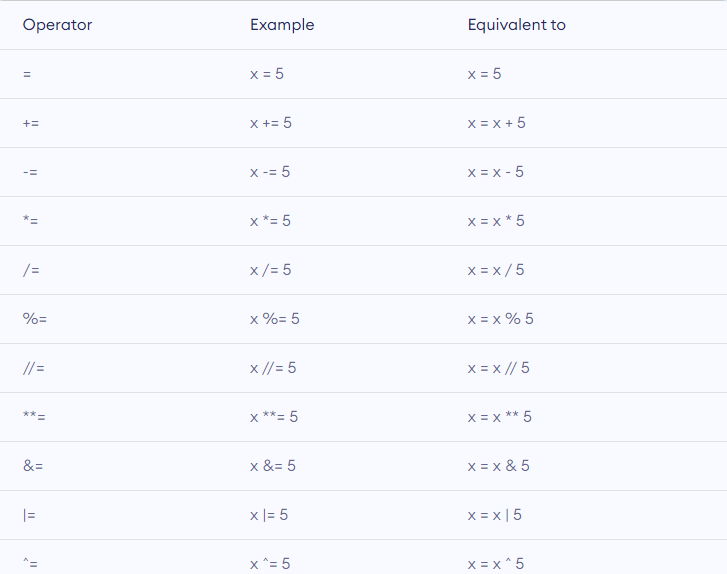


Assignment operators

Assignment operators are used in Python to assign values to variables.

a = 5 is a simple assignment operator that assigns the value 5 on the right to the variable a on the left.

There are various compound operators in Python like a += 5 that adds to the variable and later assigns the same. It is equivalent to a = a + 5.

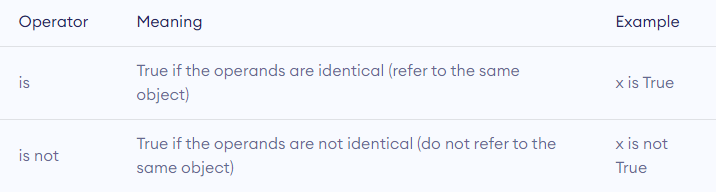


## Special operators

Python language offers some special types of operators like the identity operator or the membership operator. They are described below with examples.

### **Identity operators**

is and is not are the identity operators in Python. They are used to check if two values (or variables) are located on the same part of the memory. Two variables that are equal does not imply that they are identical.



### **Membership operators**

in and not in are the membership operators in Python. They are used to test whether a value or variable is found in a sequence (**string, list, tuple, set and dictionary**).

In a dictionary we can only test for presence of key, not the value.

