- Bitwise operators
- Membership operators
- Identity operators

3.7.1 Arithmetic operators

Arithmetic operators are used to perform arithmetic operations between any two operands. The various arithmetic operators supported by Python are given below:

- The '+' operator is used to perform the addition operation between the two input data values.
- The '–' operator is used to perform the subtraction operation between the two input data values.
- The '*' operator is used to perform the multiplication operation between the two input data values.
- The '/' operator is used to perform the division operation between the two input data values.
- The '%' operator is used to return the remainder of the division operation performed between the two input data values.
- The '**' operator is used for calculating the exponential.
- The '//' operator is used to perform the floor division operation between the two input data values. It returns the quotient of the division operation. The decimal points in the quotient result will be removed. If any one of the input is negative, then the quotient result is rounded towards negative infinity.

  For example, if the two numbers considered are 20 and 10, then the result of

  a + b = 30
  a – b = 10
  a * b = 200
  a / b = 2
  a % b = 0 (remainder is 0)
  a ** 2 = 400 (202)
  –a//b = –2 (a is negative, so the quotient will be rounded towards negative infinity)

3.7.2 Comparison (relational) operators

These operators are used to estimate the relation between two given input values. The various relational operators are given below:

- The '==' operator is used to compare whether the two given input values are equal or not. It returns True when both are equal; otherwise it returns False.
- The '!=' operator is used to compare whether the two given input values are equal or not. It returns True when both are not equal; otherwise it returns False.
- The '>' operator is used to compare the two given input values, whether the left side value is greater than right side value or not. It returns True if the left side value is greater than the other value otherwise it returns False.
- The '<' operator is used to compare the two given input values, whether the left side value is less than right side value or not. It returns True if the left side value is less than the other value otherwise it returns False.
- The '>=' operator is used to compare the two given input values, whether the left side value is greater than or equal to right side value or not. It returns True if the left side value is greater than or equal to the other value otherwise it returns False.
- The '<=' operator is used to compare the two given input values, whether the left side value is less than or equal to right side value or not. It returns True if the left side value is less than or equal to the other value otherwise it returns False.

For example, if the two numbers considered are 20 and 10 then the result of

(a == b) → False
(a != b) → True
(a > b) → True
(a < b) → False
(a >= b) → True
(a <= b) → False

3.7.3 Assignment operators

The assignment operator is used to assign the input value to the variable. The assignment operator can be grouped with arithmetic operators and are called as compound assignment operators. The various assignment operators are as follows:

- The '=' operator is used to assign the input value (right-side value) to the variable (left-side variable).
- The '+=' is used to add the right-side value and the left-side value and assigns the result to the left-side variable.
- The '-=' is used to subtract the right-side value from the left-side value and assigns the result to the left-side variable.
- The '*=' is used to multiply the right-side value and the left-side value and assigns the result to the left-side variable.
- The '/=' is used to divide the left-side value by the right-side value and assigns the result to the left-side variable.
- The '**=' is used to compute the exponential of the left-side value raised to the right-side value and assigns the result to the left-side variable.
- The '//=' is used to compute the result of floor division of the left-side value by the right-side value and assigns the result to the left-side variable.

The results of the following operators are

(a = 20) → value of a is 20
(a = 20), (a += 10) → value of a is 30
(a = 20), (a -= 10) → value of a is 10
(a = 20), (a *= 10) → value of a is 200
(a = 20), (a /= 10) → value of a is 2
(a = 20), (a **= 2) → value of a is 400
(a = 20), (a //= 10) → value of a is -2

3.7.4 Logical operators

The logical relation between two input values is computed by logical operators. The various logical operators are

- The 'and' is called as Logical AND operator. It returns True only when both the inputs are True.
- The 'or' is called as Logical OR operator. It returns True when at least one of input is True.
- The 'not' is called as Logical NOT operator. It returns the negativity of the input value.

For example, if two input numbers a and b are 1 and 0, then the results of the following operations are

(a and b) → False
(a or b) → True
(not (a)) → False

3.7.5 Bitwise operators

The bitwise operators work bit by bit, on the input data value. For each value, the 8-bit ASCII code will be considered and on these bits, the following operations are performed.

- The '&' is called as binary AND operation. It will select the resultant bit to True only when the corresponding two input bit values are True; otherwise it is False.
- The '|' is called as binary OR operation. It will select the resultant bit to True when at least one of the corresponding two input bit values are True; otherwise it is False.
- The '^' is called as binary XOR operation. It will select the resultant bit to True only when the corresponding two input bit values are different; otherwise it is False.
- The '~' is called as binary ones complement operation. It will set the resultant bit to the complemented input bit value.
- The '<<' is called as binary Left Shift Operation. It will left shift the left-side value. The right-side value will specify the required number of shift operations.
- The '>>' is called as binary Right Shift Operation. It will right shift the left-side value. The right-side value will specify the required number of shift operations.

For example, if the two numbers considered are a = 101 and b = 011. The results of the following operations are

(a & b) → 001
(a | b) → 111
(a ^ b) → 110
(~ a) → 010

3.7.6 Membership operators

The membership operators in Python are used for strings, lists or tuples. They focus on testing the membership in the sequence. The various operators are

- The 'in' operator is used to search the input element in the sequence. If the input element is found, then it returns True; otherwise it returns False.
- The 'not in' operator is used to search the input element in the sequence. If the input element is not found then returns it True; otherwise it returns False.

```
x=(1,2,3)
y=2
a=(y in x)
print("in operator result=",a)
b=(y not in x)
print("not in operator result=",b)
```

Output

```
D:\PythonPrograms\aa>python a2.py
in operator result= True
not in operator result= False
```

3.7.7 Identity operators

Identity operators are used to evaluate whether the given two objects point to the same memory location or not. The various operators are

- The 'is' operator is used to return True if both objects points to the same memory location; otherwise it returns False.
- The 'is not' operator is used to return True if both objects are not pointing to the same memory location; otherwise it returns False.

```
x=(1,2,3)
y=2
a=(y is x)
print("is operator result=",a)
b=(y is not x)
print("not is operator result=",b)
```

Output

```
D:\PythonPrograms\aa>python a2.py
is operator result= False
not is operator result= True
```