13/11/2022

Identity Operators in Python

- Compare the objects to check if both the objects are actually same objects and share same memory location

e.g. - any real life example.

Python is purely Object based language. So everything in Python is an object & every object is stored at a memory location. So each object is having its own unique id.

→ To find id of an object. syntax is/-

```
id(object_name)
```

So it will return memory address of the object which have been passed as an argument.

eg:-
```
a = 10
print(id(a))
print(id(10))
```
→ So it will return address of 10

both lines will print same address means id for both "a" and "10" is same. Why so?

Time Point

Reason:- Memory Manager in Python reuses the objects instead of creating another object for the same data (with same data type)

So identity operators compares these unique ids or we can say compare objects based on unique ids.

Types of Identity operators :- 2 types

"is" "is not"

is operator :-

returns true if both variables are same object

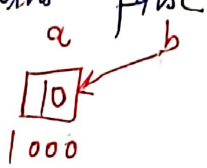
or

returns true if both operands have the same unique id (points to the same memory location)

otherwise it returns false

eg:- ① a = 10

b = 10



→ print (a is b) ⇒ returns True

→ print (a is not b) ⇒ false

② a = 10

b = 10.12

print (a is b) ⇒ returns False.

③ a = 10

b = '10'

→ print (a is b) ⇒ return false

→ print (a is not b) ⇒ true

④ a = 10

print (id(a) o/p = ? ⇒ 2371900342800

a = 9

print (id(a) ⇒ 2371900342768

print (a is a) — true How?

exercise ⇒

is not :- works in reverse manner

return true if memory location of two objects are not same

return false if memory location of two objects are same.

NOTE:- Identity operator is not equality operator

equality operator (==) checks for equality means if values are equal or not

e.g:- a = 5

b = 5

print(a == b) ⇒ true bcoz it will compare the values of a & b. value of a is 5 & value of b is also 5. So it gives true.

But

Identity operator checks identity means it is going to check if the values are same in terms of being same object in memory. So basically it will check the memory location of objects. if memory location of both objects are same means both objects are equal.

a = 5

b = 5

print(a is b) - true but it will not check the values of a & b. it checks the memory address of a & b.

eg:- ① banana shake == Mango Shake → false

② banana shake == banana shake → true

③ banana shake is banana shake → false ⇒ bcoz objects are different 2 glasses.

But if we share same glass of shake then

(4) banana shake is banana shake → true

(5) banana shake == banana shake → true

example:-

```
a = 5
b = 5
print( id(a) == id(b) ) or
print ( a is b)
```