

Membership Operator

Membership operator is used for searching a given value into group of values (collections)

1. in
2. not in

It is a binary operator and required 2 operands. This operator returns boolean value (True/False).

in operator returns True, if given value exists in group of values else return False.

not in operator returns True, if given value not exists in group of values else return False.

```
>>> "a" in "java"
True
>>> "x" in "python"
False
>>> "naresh@nareshit.com" in ["suresh@gmail.com",
...                           "ramesh@nareshit.com",
...                           "naresh@gmail.com"]
False
>>> "x" not in "python"
True
>>> "o" not in "python"
False
>>> 10 in 10
Traceback (most recent call last):
  File "<pyshell#7>", line 1, in <module>
    10 in 10
TypeError: argument of type 'int' is not iterable
```

Example:

write a program to find input character is vowel
or not

```
ch=input("Enter any character")
print("vowel") if ch in "aeiouAEIOU" else print("not vowel")
```

Output:

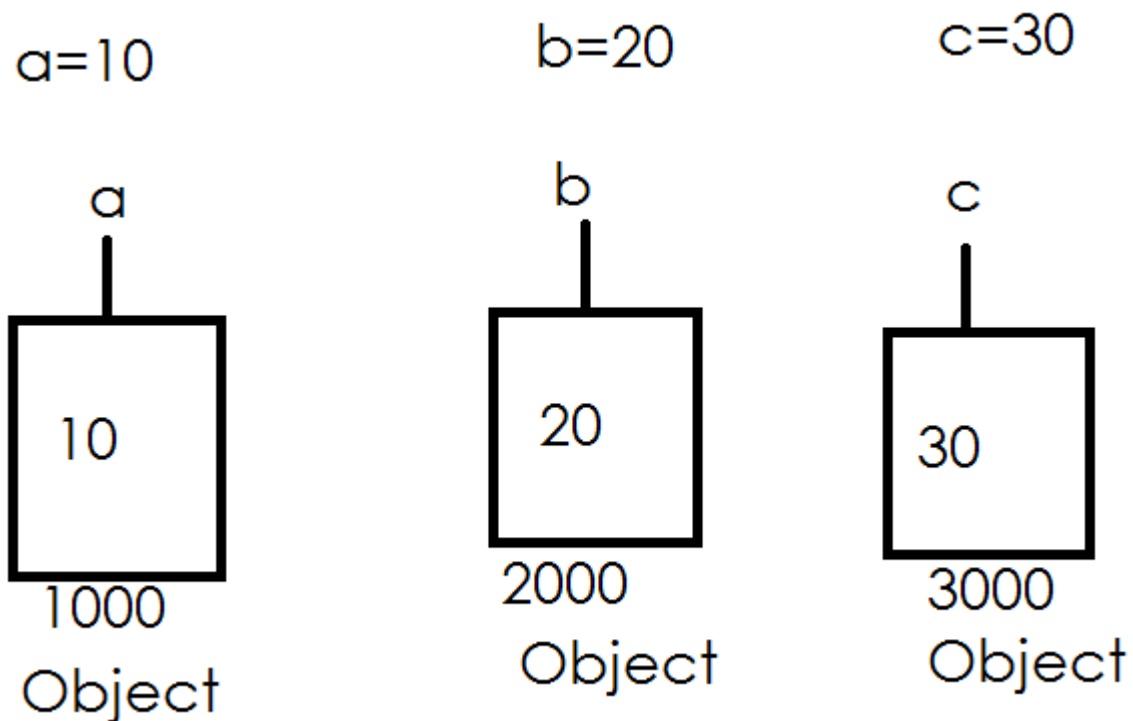
Enter any character
vowel

Enter any character
A
vowel

Enter any character
x
not vowel

Identity Operator

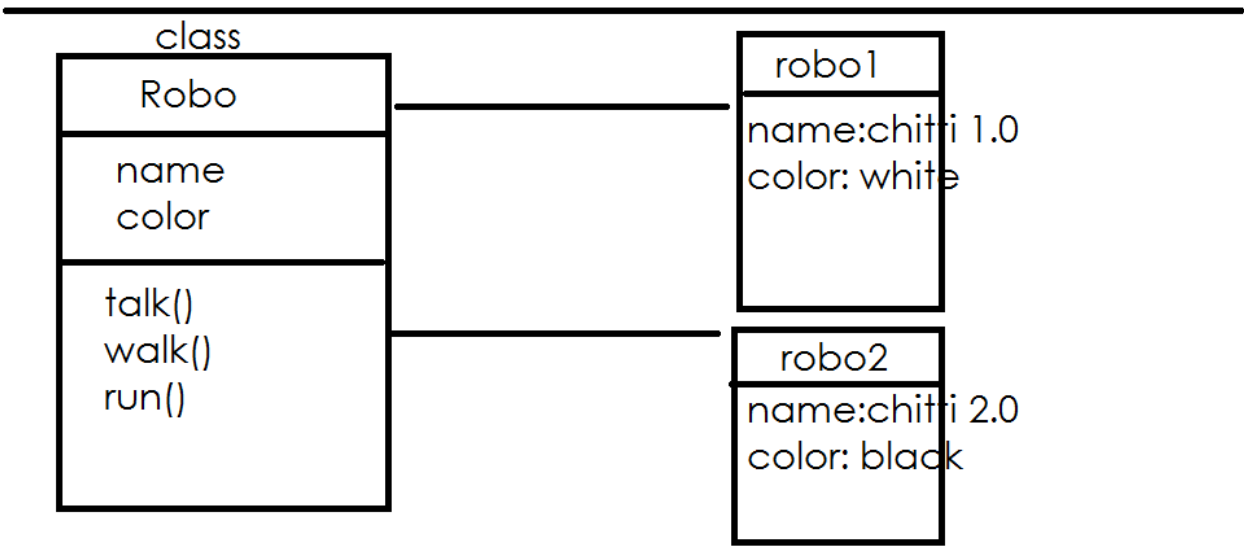
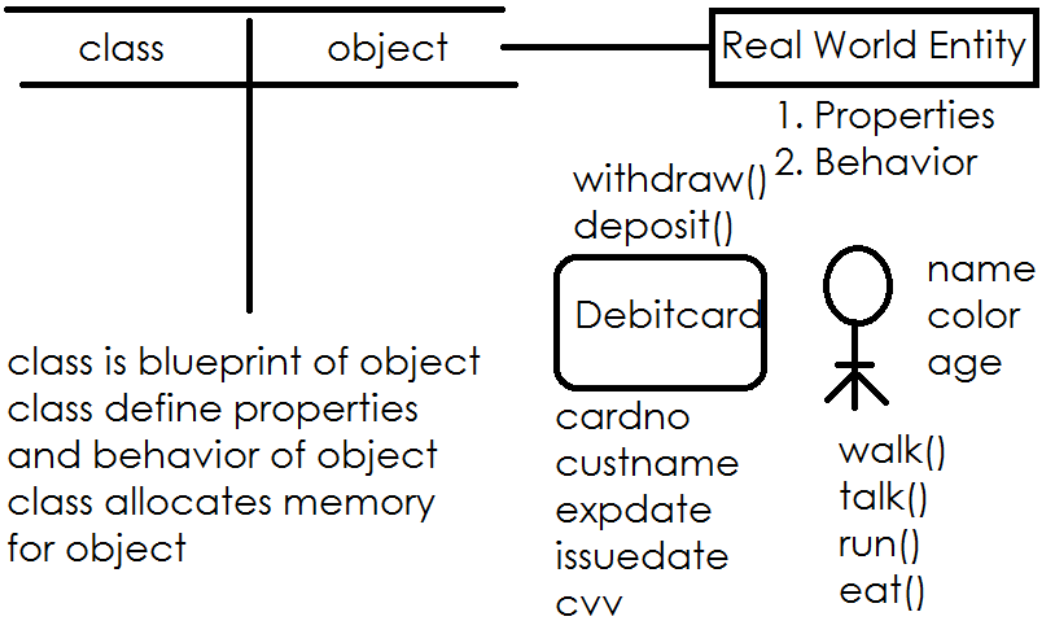
Every object created in memory is identified with unique number called address.

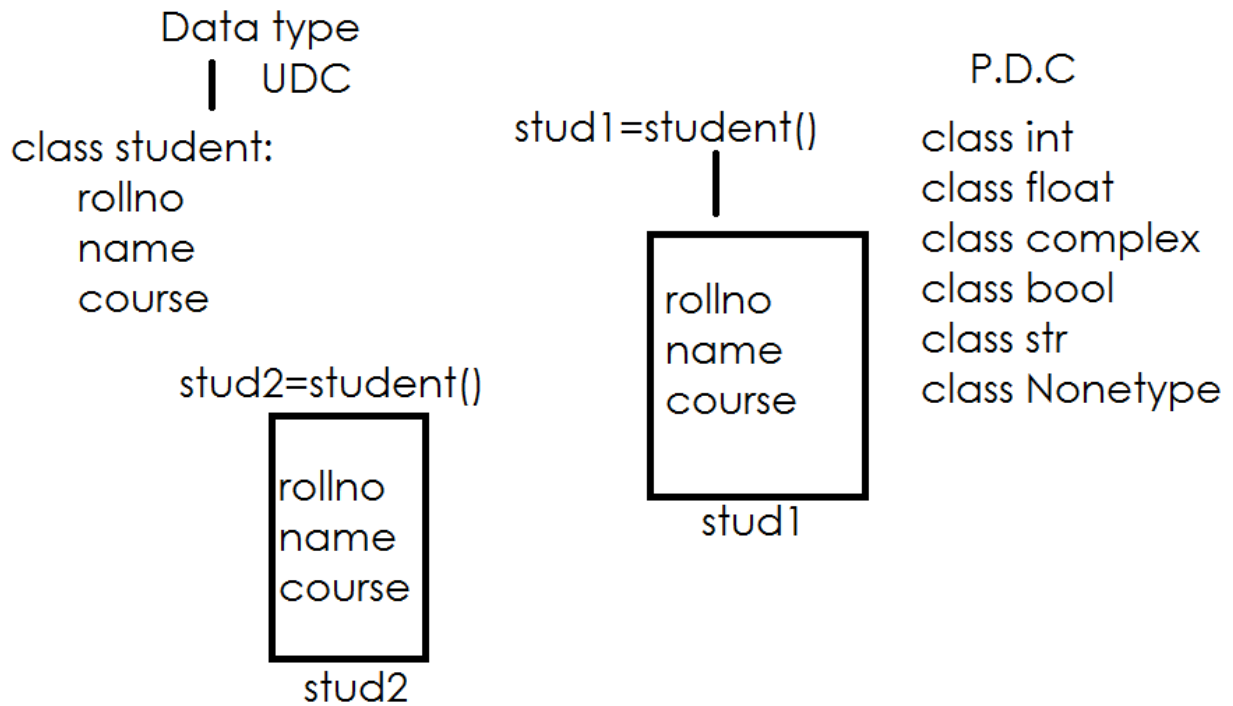


Python is an object oriented programming language

Data is represented as objects

Data type as class



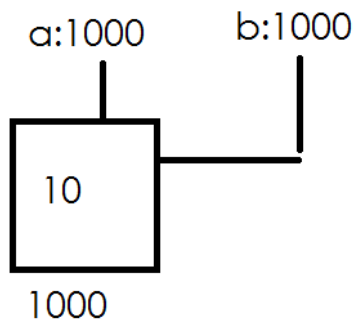


id(): It is a predefined function of python. This function returns identity or address of object.

id(object-name)

```
>>> a=10
>>> type(a)
<class 'int'>
>>> id(a)
140732974163016
>>> a
10
>>> b=20
>>> type(b)
<class 'int'>
>>> b
20
>>> id(b)
140732974163336
```

a=10 b=a

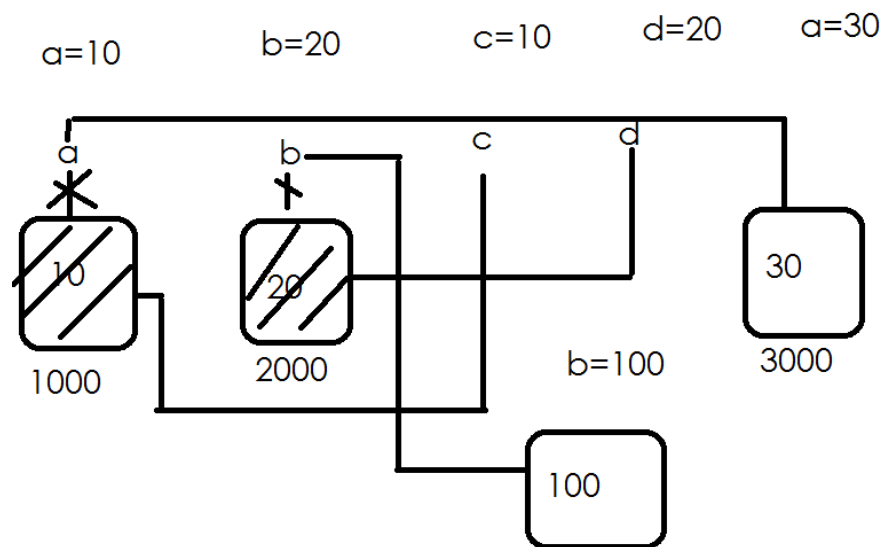


Identity operator is used to compare identity of variables (OR) it identify two variables are pointing to same object created within memory.

1. is
2. is not

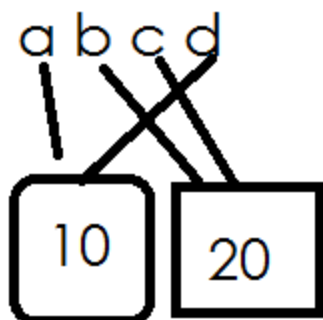
```
>>> a=10
>>> b=a
>>> id(a)
140732974163016
>>> id(b)
140732974163016
>>> a is b
True
```

All scalar data types are immutable types. Immutable objects sharable.



$a=10$
 $b=20$
 $c=a+10$
 $d=b-a$

how many variables
and object created



Immutable objects values cannot change after creation. Whenever changes are done PVM create new memory location/object and do changes.

Mutable object values can be changed after creation. Changes can be done in same object.

Example:

```
a=1.5
b=1.5
print(id(a),id(b))
c=10
d=10
print(id(c),id(d))
f1=1+2j
f2=1+2j
print(id(f1),id(f2))

list1=[10,20,30]
list2=[10,20,30]
print(id(list1),id(list2))

print(a is b)
print(c is d)
print(f1 is f2)
print(list1 is list2)
print(list1==list2)
```

Output:

```
657388392976 657388392976
140732974163016 140732974163016
657389702896 657389702896
657351548928 657397389824
True
True
True
False
True
```

Q: Difference between == and is operators in Python

The '==' is known as the equality operator. The 'is' is known as the identity operator. The == operator helps us compare the equality of objects. The is operator helps us check whether different variables point towards a similar object in the memory.