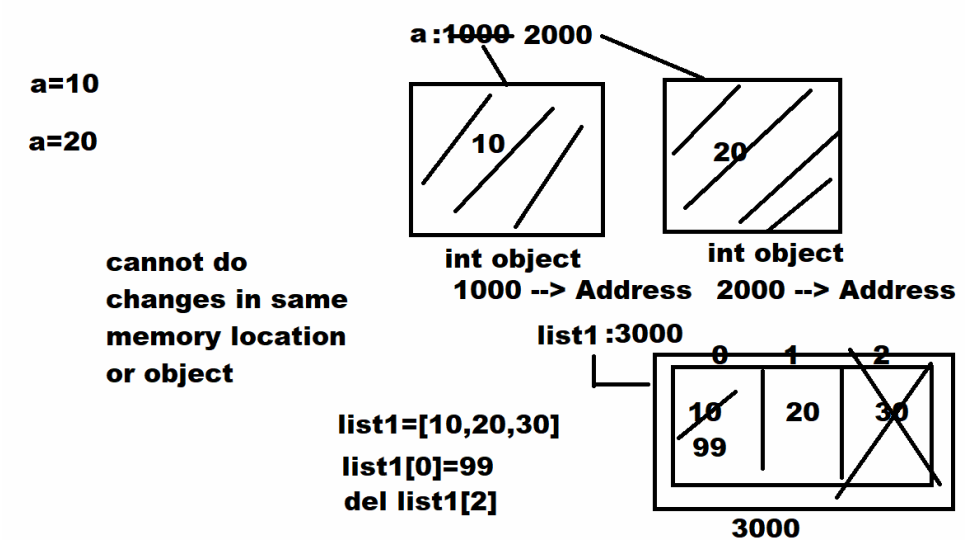
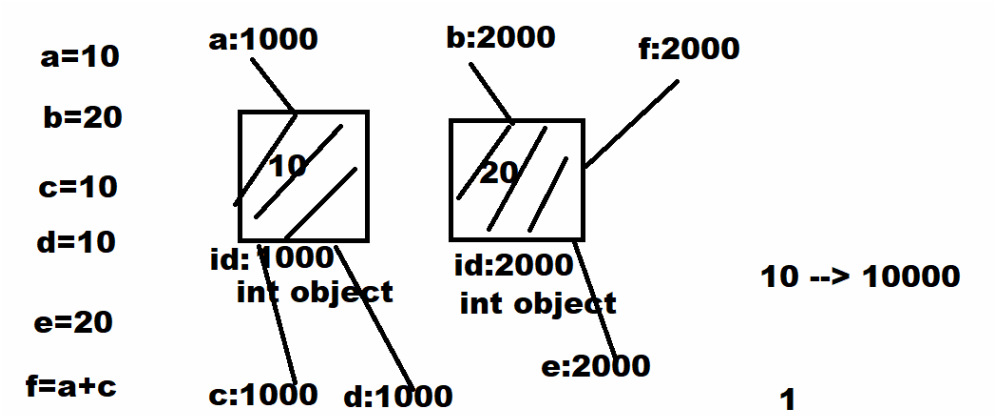
Python Data types

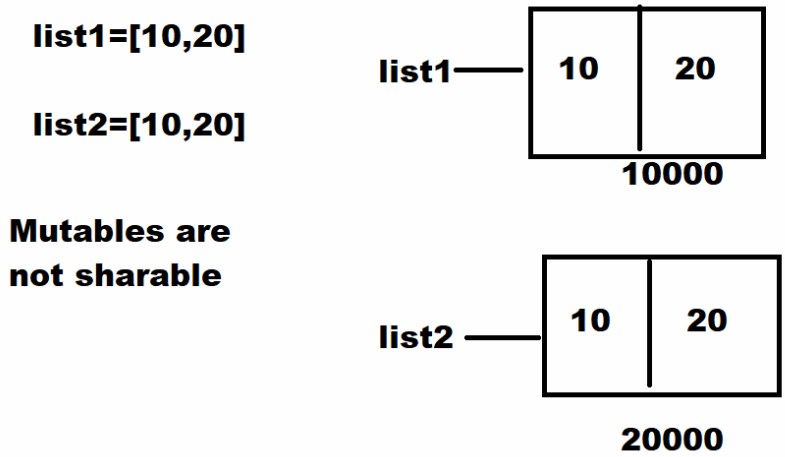
1. Mutable Data types/Object: An object whose values can be changed after creation is called mutable object.
    a. List → Sequence
    b. Set → Sets
    c. Dictionary → Mapping
    d. Bytearray → Sequence

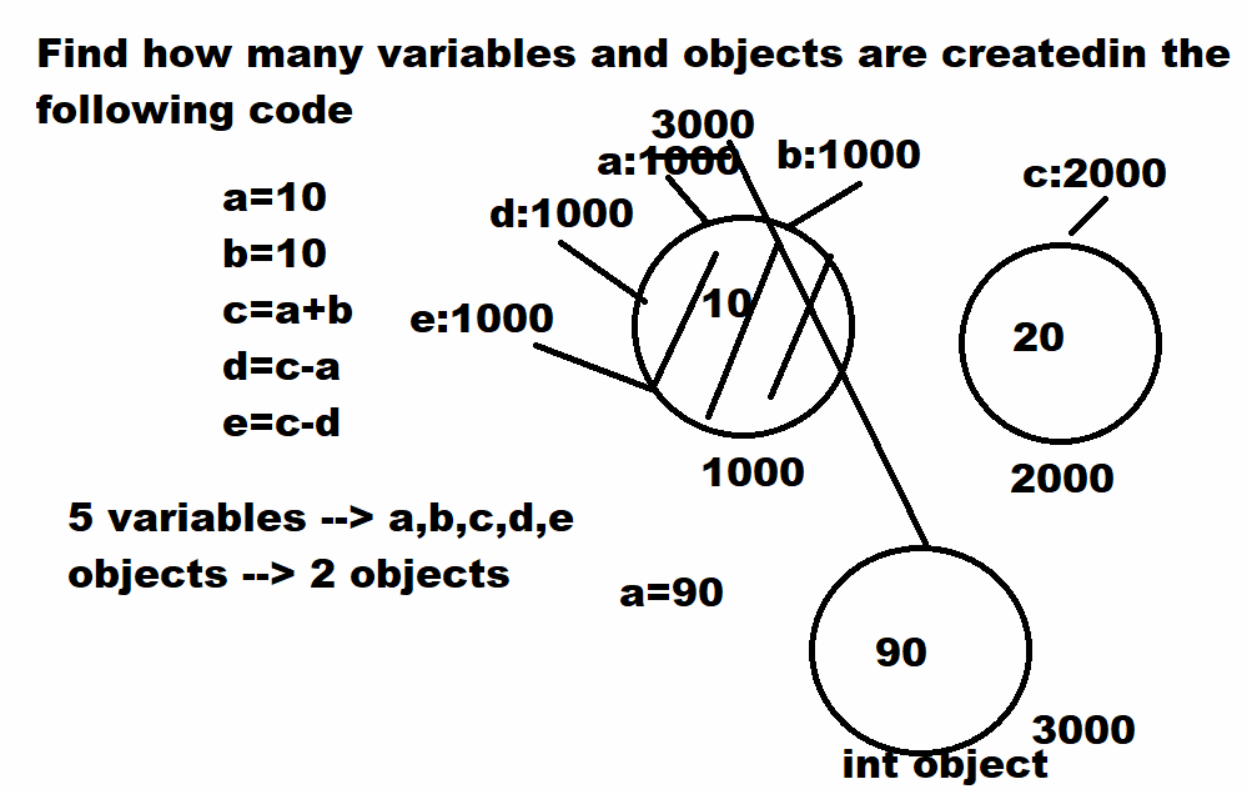
2. Immutable Data types/Object: An object whose values cannot changed after creation is called immutable object. All immutable types are constants.
    a. Int
    b. Float
    c. Complex
    d. Boolean
    e. Nonetype
    f. Tuple
    g. String
    h. Range
    i. Bytes
    j. Frozenset

Immutable objects are shared.




a=10
b=20
c=10
d=10
e=20
f=a+c
a:1000
b:2000
f:2000
10
20
id:1000
int object
id:2000
int object
e:2000
c:1000
d:1000
10 --> 10000
1


list1=[10,20]
list2=[10,20]
Mutables are
not sharable
list1
10
20
10000
list2
10
20
20000



id() : It is a predefined function in python, which returns address or id of object.

Syntax: id(object)

Example:

```
# Write a program to find identity or address of objects

a=10
b=10
c=a+b
d=c-a
e=c-d
print(a,b,c,d,e)
print(id(a),id(b),id(c),id(d),id(e))
a=99
print(a,id(a))
f1=1.5
f2=1.5
print(f1,f2)
print(id(f1),id(f2))
```

```
c1=1+2j
c2=1+2j
print(c1,c2)
print(id(c1),id(c2))
s1="abc"
s2="abc"
print(s1,s2)
print(id(s1),id(s2))

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

Output

```
10 10 20 10 10
140737479506648 140737479506648 140737479506968
140737479506648 140737479506648
99 140737479509496
1.5 1.5
2046049741296 2046049741296
(1+2j) (1+2j)
2046049738160 2046049738160
abc abc
140737478418928 140737478418928
[10, 20] [10, 20]
2046012478208 2046055527424
```

Identity operator

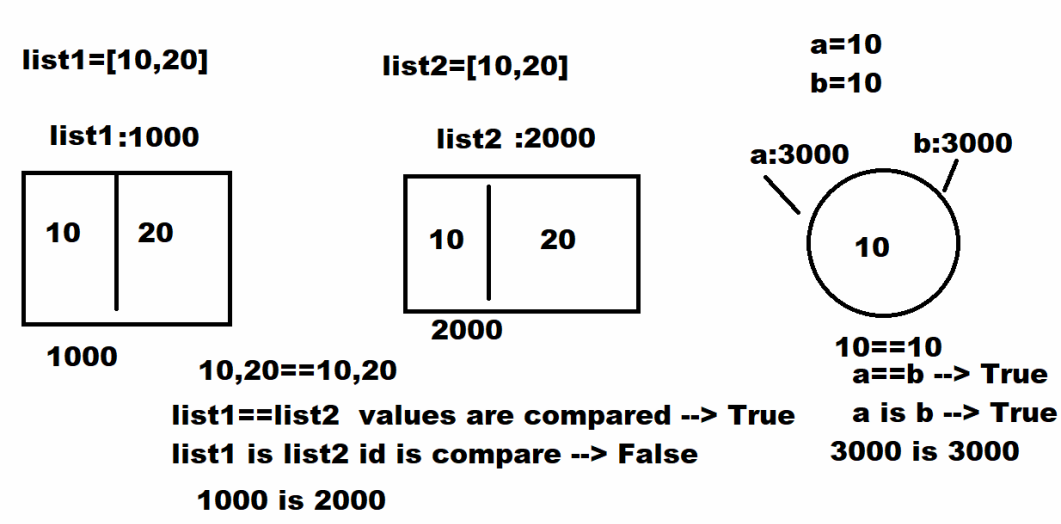
Identity operator is binary operator and required 2 operands.
Identity operator is used to find 2 variables are pointing to same object in memory.

1. is
2. is not

Identity operator returns Boolean value(True/False)

Q: what is difference between == and is operator in python?

The == operator compares the value or equality of two objects, whereas the Python is operator checks whether two variables point to the same object in memory.



Example:

```
list1=[10,20]
list2=[10,20]

print(list1==list2)
print(list1 is list2)

a=10
b=10
print(a==b)
print(a is b)
```

Output

```
True
False
```

True
True

Bitwise Operator
Bitwise operators are used to perform operation on binary data.

Applications of bitwise operators

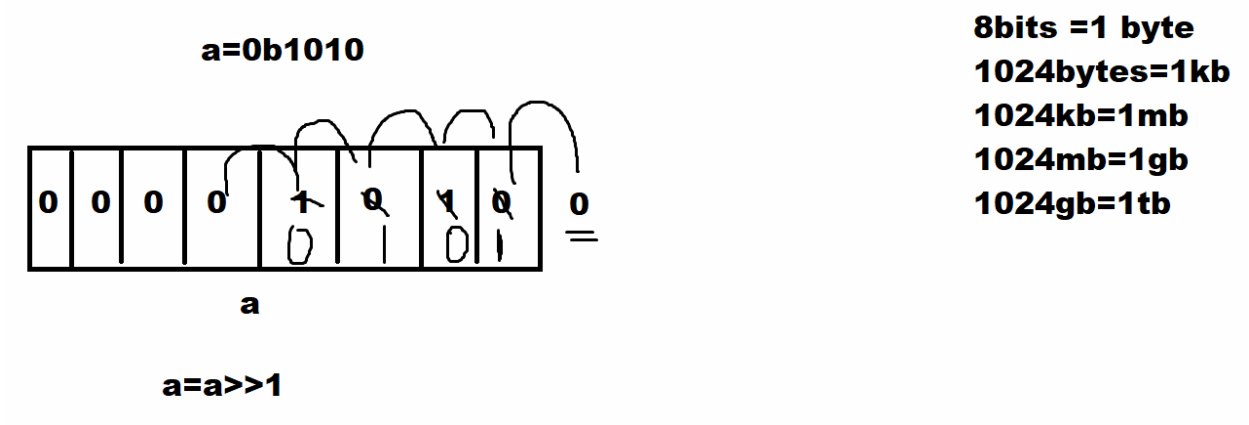
1. Embedded Applications (Logic Gates)
2. Memory management (Operating System or Programming Languages)
3. Image Processing

1. Shift Operators
    a. >> (Right shift operator)
    b. << (Left shift operator)
2. Bitwise & (and) operator
3. Bitwise | (or) operator
4. Bitwise ^ (xor) operator
5. Bitwise ~ (not) operator

Note: bitwise operators are applied only one integer data type.

Right shift operator
This operator is used to shift number of bits towards right side.
By shifting number bits towards right side the value get decremented or shifted bits are deleted.

Syntax: opr>>n



```
>>> a=0b1010
>>> print(a)
10
>>> print(bin(a))
0b1010
>>> b=a>>1
>>> print(b)
5
>>> print(bin(b))
0b101
>>> bin(25)
'0b11001'
>>> bin(0o12)
'0b1010'
>>> bin(0xb)
'0b1011'
>>> n1=65
>>> n2=n1>>3
>>> print(n1,n2)
65 8
>>> print(bin(n1),bin(n2))
0b1000001 0b1000
```

Formula : opr//2 pow n

```
>>> n1=65
>>> n2=n1>>3
>>> print(n1,n2)
65 8
>>> print(bin(n1),bin(n2))
0b1000001 0b1000
```

Left shift operator