In [4]:
```
# -------------------- Arithmetic Operators ----------------
a = 4
b = 3

g = a
g += a
```

In [6]:
```
print(a * b + a)
print(a / b)
print(a // b)
print(a ** b)
print(g)
```

```
16
1.3333333333333333
1
64
8
```

In [13]:
```
#-------------------- Comparison Operators ----------------
# => the result is always in Boolean

print(a != b)
print(a >= b)
```

```
True
True
```

In [22]:
```
# Be Careful!
#It's wrong to use this form. -->  print(! (a >= b))
print(not (a >= b))
```

```
False
```

In [20]:
```
#-------------------- Logical Operators ---------------------
a = 7
b = 4
print((a > b and a - 2 < b) or (a > b))
```

```
True
```

In [32]:
```
#-------------------- Bit-Wise Operators ---------------------
a = 11    # a = 01011
b = 5    #  b = 00101

print(a | b)  #01111 --> 15
print(a & b)  #00001 --> 1
print(not(a))      # not (all none-zero number) = False
print(not(0))
print(a << 2) # n shift to left --> * 2^n
print(a >> 2)
```

```
15
1
False
True
44
2
```

In [40]:
```
#-------------------- Membership Operators ---------------------
name = "Kaveh"
my_list = [name , 123 , 1 , 1 , 2 ,3]
my_set = set(my_list)
my_tuple = (name , 625 , 25 , 5)
my_dictionary = {"my_name" : "Kaveh"}
print('K' in name)
print(1234 in my_list)
print(name not in my_list)


print("-----------")


print(1 in my_set)
print(5 ** 3 in my_tuple)
print(name in my_dictionary.values())
```

```
True
False
False
-----------
True
False
True
```

In [41]:
```
#-------------------- Indentity Operators ---------------------
# non-primitive data

a = [1 , 2 , 3]
b = [1 , 2 , 3]
c = a
d = a.copy()
print(a is b)
print(a is c)
print(a is d)
```

```
False
True
False
```

In [42]:
```
a.append(4)
```

In [44]:
```
print(a)
print(b)
print(c)
```

```
[1, 2, 3, 4]
[1, 2, 3]
[1, 2, 3, 4]
```

In [ ]:
```

```