Q. Declare a complex number and store it in a variable.

Check the type and print the id of the same.

In [1]:
```
a=20+10j
print(a)
print(type(a), id(a))
```

```
(20+10j)
<class 'complex'> 1635992523600
```

Q. Arithmetic Operations on complex number

Take two different complex numbers.

Store them in two different variables.

Do below operations on them:-

Find sum of both numbers

Find difference between them

Find the product of both numbers.

Find value after dividing first num with second number

Find the result of the first num to the power of the second number.

In [2]:
```
a=20+10j
b=10+30j
print(a+b)
print(a-b)
print(a*b)
print(a/b)
print(a**b)
```

```
(30+40j)
(10-20j)
(-100+700j)
(0.4999999999999999-0.5j)
(-25412390.85566285-12788183.053630497j)
```

Q. Comparison Operation not applicable between instance of complex values.

Object reusability concept is not applicable on complex number

In [3]:
```
1 a=20+10j
2 b=10+30j
3 print(a>b)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-3-27786418da17> in <module>
      1 a=20+10j
      2 b=10+30j
----> 3 print(a>b)

TypeError: '>' not supported between instances of 'complex' and 'complex'
```

Q. Equality Operator

Take two different complex numbers.

Store them in two different variables.

Equate them using equality operators (==, !=)

Observe the output(return type should be boolean)

In [4]:
```
1 a=20+10j
2 b=10+30j
3 print(a==b)
4 print(a!=b)
```

```
False
True
```

Q. Logical operators

Observe the output of below code

Cross check the output manually

In [5]:
```
print ( 10 + 20j and 20 + 30j ) #20+30j
#---------------------------------------->Output is 20+30j
print ( 0 + 0j and 20 + 30j ) #0+0j
#---------------------------------------->Output is 0j
print ( 20 + 30j and 0 + 0j ) #0+0j
#---------------------------------------->Output is 0j
print ( 0 + 0j and 0 + 0j ) #0+0j
#---------------------------------------->Output is 0j
print ( 10 + 20j or 20 + 30j ) #10+20j
#---------------------------------------->Output is 10+20j
print ( 0 + 0j or 20 + 30j ) #20+30j
#---------------------------------------->Output is 20+30j
print ( 20 + 30j or 0 + 0j ) #20+30j
#---------------------------------------->Output is 20+30j
print ( 0 + 0j or 0 + 0j ) #0+0j
#---------------------------------------->Output is 0j
print ( not 10 + 20j ) #False
#---------------------------------------->Output is False
print ( not 0 + 0j ) #True
#---------------------------------------->Output is True
```

```
(20+30j)
0j
0j
0j
(10+20j)
(20+30j)
(20+30j)
0j
False
True
```

Q. What is the output of the expression inside the print statement.

Cross check before running the program.

In [6]:
```
a = 10 + 20j
b = 10 + 20j
print (a is b) #False #True or False?
print (a is not b) #True #True or False?
# This is because Object Reusability Concept does not apply on derived data-
```

```
False
True
```

Q. Membership operation

in, not in are two membership operators and it returns boolean value

In [7]:

```
print ( '2.7' in 'Python2.7.8' ) #True
print ( 10 + 20j in [ 10 , 10.20 , 10 + 20j , 'Python' ]) #True
print ( 10 + 20j in ( 10 , 10.20 , 10 + 20j , 'Python' )) #True
print ( 30 + 40j in { 1 , 20.30 , 30 + 40j }) #True
print ( 30 + 40j in { 1 : 100 , 2.3 : 200 , 30 + 40j : 300 }) #True
print ( 10 in range ( 20 )) #True
```

```
True
True
True
True
True
True
```