In [ ]:
```
Q1. Declare a float value and store it in a variable.
Check the type and print the id of the same.
```

In [1]:
```
a=10.05
print(type(a))
print(id(a))
```

```
<class 'float'>
1681744265872
```

In [ ]:
```
Q2. Arithmetic Operations on float
Take two different float values.
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 remainder after dividing first number with second number
 Find the quotient after dividing first number with second number
 Find the result of the first num to the power of the second number.
```

In [3]:
```
x=2.5
y=1.5
print(x+y) #sum of both
print(x-y) #difference
print(x*y) #product
print(x/y) #divide
print(x%y) #remainder
print(x//2) #quotient
print(x**y) #exponent
```

```
4.0
1.0
3.75
1.6666666666666667
1.0
1.0
3.952847075210474
```

In [ ]:
```
Q3. Comparison Operators on float
Take two different float values.
Store them in two different variables.
Do below operations on them:-
 Compare these two numbers with below operator:-
 Greater than, '>'
 Smaller than, '<'
 Greater than or equal to, '>='
 Less than or equal to, '<='
Observe their output(return type should be boolean)
```

In [5]:
```
x=20.05
y=10.05
print(x>y)
print(x<y)
print(x>=y)
print(x<=y)
```

```
True
False
True
False
```

In [ ]:
```
Q4. Equality Operator
Take two different float values.
Store them in two different variables.
Equate them using equality operators (==, !=)
Observe the output(return type should be boolean)
```

In [6]:
```
x=100.20
y=50.05
print(x==y)
print(x!=y)
```

```
False
True
```

In [ ]:
```
Q5. Logical operators
Observe the output of below code
Cross check the output manually
print(10.20 and 20.30) #both are true and second value taken
>Output is 20.3
print(0.0 and 20.30) #First is false so first value
taken->Output is 0.0
print(20.30 and 0.0) #Goes to till second and second value is
false so second is taken>Output is 0.0
print(0.0 and 0.0) #First is false so first value is
taken->Output is 0.0
print(10.20 or 20.30) #First is True so first value is
taken>Output is 10.2
print(0.0 or 20.30) #Goes to till second and second is true
second value is taken->Output is 20.3
print(20.30 or 0.0) #First is True so first value is
taken->Output is 20.3
print(0.0 or 0.0) #Goes to till second and second is also
false and second value is taken>Output is 0.0
print(not 10.20) #-Not of true is false->Output is False
print(not 0.0) #Not of false is True>Output is True
```

In [7]:
```
print(10.20 and 20.30) #both are true and second value taken
print(0.0 and 20.30) #First is false so first value
print(20.30 and 0.0) #Goes to till second and second value is
print(0.0 and 0.0) #First is false so first value is
print(10.20 or 20.30) #First is True so first value is
print(0.0 or 20.30) #Goes to till second and second is true
print(20.30 or 0.0) #First is True so first value is
print(0.0 or 0.0) #Goes to till second and second is also
print(not 10.20) #-Not of true is false->Output is False
print(not 0.0) #Not of false is True>Output is True
```

```
20.3
0.0
0.0
0.0
10.2
20.3
20.3
0.0
False
True
```

In [ ]:
```
Q6. What is the output of expression inside print statement. Cross check
before running the program.
a = 10.20
b = 10.20
print(a is b) #True or False? True 10.20<256
print(a is not b) #True or False? False
Why the Id of float values are different when the same value is
assigned to two different variables
ex: a = 10.5 b=10.5. but id will be same if I assign the variable
having float i.e. a=c then both a and c's Id are same
```

In [10]:
```
a = 10.20
b = 10.20
print(a is b)
print(a is not b)

print(id(a))
print(id(b))
c=a
print(id(c))
```

```
False
True
2548935758448
2548935757680
2548935758448
```

In [ ]:
```
Q7. Bitwise operation is not applicable between instances of float.
Why the Id of float values are different when the same value is
assigned to two different variables
ex: a = 10.5 b=10.5. but id will be same if I assign the variable
having float i.e. a=c then both a and c's Id are same
Object reusability concept is not applicable on float values.
```

In [16]:
```
#Bitwise operation on integers
a=1
b=2
print(a & b)
print(a | b)
print(a ^ b)
#print(a ~ b)
print(a << b)
print(a >> b)
```

```
0
3
3
4
0
```

In [17]:
```
#Bitwise operation is not applicable between instances of float.
x=1.2
y=2.4
print(x & y)
print(x | y)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[17], line 4
      2 x=1.2
      3 y=2.4
----> 4 print(x & y)
      5 print(x | y)

TypeError: unsupported operand type(s) for &: 'float' and 'float'
```

In [20]:
```
# Object reusability concept is not applicable on float values.
# ex: a = 10.5 b=10.5. but id will be same if I assign the variable
# having float i.e. a=c then both a and c's Id are same

a = 10.5
b = 10.5
print(a is b)
print(a is not b)

print(id(a))
print(id(b))
c=a
print(id(c))
```

```
False
True
2548935760240
2548935760272
2548935760240
```

In [ ]:
```
Q8. Membership operation
in, not in are two membership operators and it returns boolean value
print('2.7' in 'Python2.7.8') #True
print(10.20 in [10,10.20,10+20j,'Python']) #True
print(10.20 in (10,10.20,10+20j,'Python')) # True
print(20.30 in {1,20.30,30+40j}) # True
print(2.3 in {1:100, 2.3:200, 30+40j:300}) # True
print(10 in range(20)) # True
```

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

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

In [ ]:
```
```