Q. Declare an int value and store it in a variable.

Check the type and print the id of the same.

In [1]:
```
a=200
print(a)
print(type(a))
print(id(a))
```

```
200
<class 'int'>
140715055271936
```

Q. Take one int value between 0 - 256.

Assign it to two different variables.

Check the id of both the variables. It should come the same. Check why?

In [2]:
```
a=150
b=150
print(a)
print(id(a))
print()
print(b)
print(id(b))
# Ids of these two numbers should come same because the Interning for intege
# 0 to 256. The Object Reusability Concept is valid for this integer range.
# value in this integer range, will be pointing to the ID of the same assign

```

```
150
140715055270336

150
140715055270336
```

Q. Take one int value either less than -5 or greater than 256.

Assign it to two different variables.

Check the id of both the variables. It should come different.Check why?

In [3]:
```
a=350
b=350
print(a)
print(id(a))
print()
print(b)
print(id(b))
# Ids of these two numbers should be different because the Interning for int
# from 0 to 256. So, any two variables which are assigned same value outside
# different IDs.

```

```
350
1691507995536

350
1691507995792
```

Q. Arithmetic Operations on integers

Take two different integer 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 [4]:

```
a=250
b=230
print(a+b)
print(a-b)
print(a*b)
print(a/b)
print(a%b)
print(a//b)
print(a**b)
```

```
480
20
57500
1.0869565217391304
20
1
3358938053783544375895482794475113564785211099681514328077275075485147750543954
0483511368046719765588053129902013382055038597972370792695651096193299710913303
4119651916316294752928864883916612775021628242576912509747179815202102683044405
4853820138880318462701215044956414002607692805002503888545106747187674045562744
1406250000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000
```

Q. Comparison Operators on integers

Take two different integer values.

Store them in two different variables.

Do below operations on them:-

Compare the 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]:
```
a=250
b=230
print(a>b)
print(a<b)
print(a>=b)
print(a<=b)
```

```
True
False
True
False
```

Q. Equality Operator

Take two different integer values.

Store them in two different variables.

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

Observe the output(return type should be boolean)

In [6]:
```
a=250
b=230
print(a==b)
print(a!=b)
```

```
False
True
```

Q. Logical operators

Observe the output of below code

Cross check the output manually

print (10 and 20)

#---------->Output is 20

print (0 and 20)

#---------->Output is 0

print (20 and 0)

#---------->Output is 0

```
print ( 0 and 0 )

#---------------------------------------->Output is 0

print ( 10 or 20 )

#---------------------------------------->Output is 10

print ( 0 or 20 )

#---------------------------------------->Output is 20

print ( 20 or 0 )

#---------------------------------------->Output is 20

print ( 0 or 0 )

#---------------------------------------->Output is 0

print ( not 10 )

#---------------------------------------->Output is False

print ( not 0 )

#---------------------------------------->Output is True
```

In [7]:
```
print ( 10 and 20 )
#---------------------------------------->Output is 20
print ( 0 and 20 )
#---------------------------------------->Output is 0
print ( 20 and 0 )
#---------------------------------------->Output is 0
print ( 0 and 0 )
#---------------------------------------->Output is 0
print ( 10 or 20 )
#---------------------------------------->Output is 10
print ( 0 or 20 )
#---------------------------------------->Output is 20
print ( 20 or 0 )
#---------------------------------------->Output is 20
print ( 0 or 0 )
#---------------------------------------->Output is 0
print ( not 10 )
#---------------------------------------->Output is False
print ( not 0 )
#---------------------------------------->Output is True
```

```
20
0
0
0
10
20
20
0
False
True
```

Q. Bitwise Operators

Do below operations on the values provided below:-

Bitwise and(&) ----------> 10, 20

-------> Output is 0

Bitwise or(|) ----------> 10, 20

-------> Output is 30

Bitwise(^) ----------> 10, 20

-------> Output is 30

Bitwise negation(~) ----------> 10

-------> Output is -11

Bitwise left shift ----------> 10,2

-------> Output is 40

Bitwise right shift ----------> 10,2

-------> Output is 2

Cross check the output manually

In [8]:
```
print(10 & 20)     #Bitwise and(&)
print(10 | 20)     #Bitwise or(|)
print(10 ^ 20)     #Bitwise xor(^)
print(~10)         #Bitwise negation(~)
print(10<<2)       #Bitwise leftshift (10,2)
print(10>>2)       #Bitwise Rightshift (10,2)
```

```
0
30
30
-11
40
2
```

Verification for the previous question

In [10]:
```
# print(bin(10))
# print(bin(20))
# #0000000000000000000000000000000000000000000000000000000000101000
# #0000000000000000000000000000000000000000000000000000000000000010
# print(int(0b101000))
# print(int(0b10))
```

Q. What is the output of expression inside print statement. Cross check before running the program.

In [11]:
```
a = 10
b = 10
print(a is b) #True or False?              is & is not compare the Ids of t
print(a is not b) #True or False?
a = 1000
b = 1000
print(a is b) #True or False?              Here, both a & b lies outside th
print(a is not b) #True or False?
```

```
True
False
False
True
```

Q. What is the output of expression inside print statement. Cross check before running the program.

In [12]:
```
print ( 10 +( 10 * 32 )// 2 ** 5 & 20 +(~( -10 ))<< 2 )
```

```
20
```

In [13]:
```
# STEP-BY-STEP Execution

# print ( 10 +( 10 * 32 )// 2 ** 5 & 20 +(~( -10 ))<< 2 )
# print ( 10 +( 10 * 32 )// 2 ** 5 & 20 +9 << 2 )
# print ( 10 +( 10 * 32 )// 2 ** 5 & 29 << 2 )
# print ( 10 +( 10 * 32 )// 2 ** 5 & 116 )
# print ( 10 +320// 32 & 116 )
# print ( 20 & 116 )
# 20

# bin(20)  # 0b0010100
# bin(56)  # 0b1110100
```

Q. Membership operation

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

In [14]:
```
print ( '2' in 'Python2.7.8' )
print ( 10 in [ 10 , 10.20 , 10 + 20j , 'Python' ])
print ( 10 in ( 10 , 10.20 , 10 + 20j , 'Python' ))
print ( 2 in { 1 , 2 , 3 })
print ( 3 in { 1 : 100 , 2 : 200 , 3 : 300 })
print ( 10 in range ( 20 ))
```

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

Q. An integer can be represented in binary, octal or hexadecimal form.

Declare one binary, one octal and one hexadecimal value and store them in three different variables.

Convert 9876 to its binary, octal and hexadecimal equivalent and print their corresponding value.

In [15]:
```
a= 0b1000100
b= 0o5734
c= 0x563cd
print(bin(9876))
print(oct(9876))
print(hex(9876))
```

```
0b10011010010100
0o23224
0x2694
```

Q. What will be the output of following:-

In [16]:
```
a = 0b1010000
print (a)
b = 0o7436
print (b)
c = 0xfade
print (c)
print ( bin ( 80 ))
print ( oct ( 3870 ))
print ( hex ( 64222 ))
print ( bin ( 0b1010000))
print ( bin ( 0xfade ))
print ( oct ( 0xfade ))
print ( oct ( 0o7436))
print ( hex ( 0b1010000))
print ( hex ( 0xfade ))
```

```
80
3870
64222
0b1010000
0o7436
0xfade
0b1010000
0b1111101011011110
0o175336
0o7436
0x50
0xfade
```