Q. Declare a boolean value and store it in a variable.

Check the type and print the id of the same.

In [1]:
```
a=True
print(type(a), id(a))
# Boolean data-types follow the Object Reusability Concept
```

```
<class 'bool'> 140715054982992
```

Q. Take one boolean value between True and False.

Assign it to two different variables.

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

In [2]:
```
a=True
b=True
print(id(a))
print(id(b))
```

```
140715054982992
140715054982992
```

Q. Arithmetic Operations on boolean data

Take two different boolean values.

Store them in two different variables.

Do below operations on them:-

Find sum of both values

Find difference between them

Find the product of both.

Find value after dividing first value with second value

Find the remainder after dividing first value with second value

Find the quotient after dividing first value with second value

Find the result of first value to the power of second value.

Find the result of first value to the power of second value.

In [3]:
```
a=False
b=True
print(a+b)
# print(int(a))
# print(int(b))
print(a-b)
print(a*b)
print(a/b)
print(a%b)
print(a//b)
print(a**b)
```

```
1
-1
0
0.0
0
0
0
```

Q. Comparison Operators on boolean values

Take two different boolean values.

Store them in two different variables.

Do below operations on them:-

Compare these two values with below operator:-

Greater than, '>'

less than, '<'

Greater than or equal to, '>='

Less than or equal to, '<='

Observe their output(return type should be boolean)

In [4]:
```
a=True
b=False
print(a>b)
print(a<b)
print(a>=b)
print(a<=b)
```

```
True
False
True
False
```

Q. Equality Operator

Take two different boolean values.

Store them in two different variables.

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

Observe the output(return type should be boolean)

In [5]:
```
a=True
b=False
print(a==b)
print(a!=b)
```

```
False
True
```

Q. Logical operators

Observe the output of below code

Cross check the output manually

In [6]:
```
print ( True and True )
#---------------------------------------->Output is True
print ( False and True )
#---------------------------------------->Output is False
print ( True and False )
#---------------------------------------->Output is False
print ( False and False )
#---------------------------------------->Output is False
print ( True or True )
#---------------------------------------->Output is True
print ( False or True )
#---------------------------------------->Output is True
print ( True or False )
#---------------------------------------->Output is True
print ( False or False )
#---------------------------------------->Output is False
print ( not True )
#---------------------------------------->Output is False
print ( not False )
#---------------------------------------->Output is True
```

```
True
False
False
False
True
True
True
False
False
True
```

Q. Bitwise Operators

Do below operations on the values provided below:-

Bitwise and(&) ----------> True, True -------> Output is True

Bitwise or(|) ----------> True, False -------> Output is True

Bitwise(^) ----------> True, False -------> Output is True

Bitwise negation(~) ---------> True -------> Output is -2

Bitwise left shift ---------> True,2 -------> Output is 4

Bitwise right shift ---------> True,2 -------> Output is 0

Cross check the output manually

In [7]:
```
print(True & True)
print(True | False)
print(True ^ False)
print(~True)
print(True<<2)
print(True>>2)
```

```
True
True
True
-2
4
0
```

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

check before running the program.

In [8]:
```
a = True
b = True
print (a is b) #True or False? #
print (a is not b) #True or False?
a = False
b = False
print (a is b) #True or False?
print (a is not b) #True or False?
```

```
True
False
True
False
```

Q. Membership operation

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

In [10]:
```
print ( True in [ 10 , 10.20 , 10 + 20j , 'Python' , True ])
print ( False in ( 10 , 10.20 , 10 + 20j , 'Python' , False ))
print ( True in { 1 , 2 , 3 , True })
print ( True in { True : 100 , False : 200 , True : 300 })
print ( False in { True : 100 , False : 200 , True : 300 })
```

```
True
True
True
True
True
```