1. Create two int type variables, apply addition, subtraction, division and multiplications and store the results in variables. Then print the data in the following format by calling the variables:

First variable is___& second variable is __.
Addition:___+___= ___
Subtraction:___-___=_____
Multiplication:___*___=___
Division:___/___= ___

**ANSWER:**

In [5]:
```
def myfunc():
    a=int(input("Enter first variable:"))
    b=int(input("Enter second variable:"))
    print("First Variable is:",a)
    print("Second Variable is:",b)

    add=a+b
    print("Addition:{a}+{b}=",add)

    sub=a-b
    print("Substraction:{a}-{b}=", sub)

    mul=a*b
    print("Multiplication:{a}*{b}=", mul)

    div=a/b
    print("Division:{a}/{b}=", div)

myfunc()
```

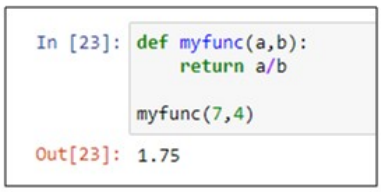
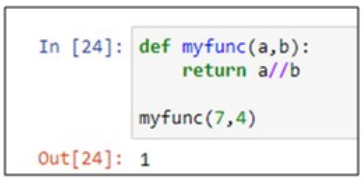
```
Enter first variable:4
Enter second variable:2
First Variable is: 4
Second Variable is: 2
Addition:{a}+{b}= 6
Substraction:{a}-{b}= 2
Multiplication:{a}*{b}= 8
Division:{a}/{b}= 2.0
```

2. What is the difference between the following operators:

(i) '/' & '//'

(ii) '**' & '^'

**ANSWER: i)**

| / | // |
|---|---|
| This method of division is considered as the 'classic division'. The '/' single slash carries out the float division. The output of this operator is always a quotient with a float datatype. The output remains float even if the input numbers are integer values. | This method of division is considered the 'true division'. The '//' double slash carries out integer division which is also known as floor division. The output of this operator will be the quotient rounded off to the closest whole number. |
|  |  |

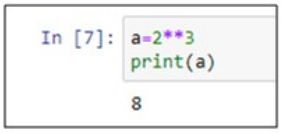
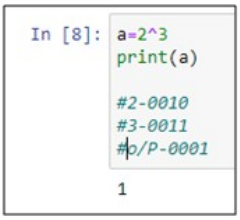
```
In [23]: def myfunc(a,b):
             return a/b

         myfunc(7,4)
Out[23]: 1.75
```

```
In [24]: def myfunc(a,b):
             return a//b

         myfunc(7,4)
Out[24]: 1
```

**ii)**

| ** | ^ |
|---|---|
| In python, ** is the exponentiation operator | ^ is the XOR operator |
|  |  |

```
In [7]: a=2**3
        print(a)
        8
```

```
In [8]: a=2^3
        print(a)

        #2-0010
        #3-0011
        #o/P-0001
        1
```

3. List the logical operators.

**ANSWER:** AND, OR and Not.

4. Explain right shift operator and left shift operator with examples.

**ANSWER:** The bitwise shift operators are the right-shift operator ( >> ), which moves the bits of an integer or enumeration type expression to the right, and the left-shift operator ( << ), which moves the bits to the left.

```
In [1]: #Bitwise Left shift
        a=0b0010
        b=a<<2
        print(b)
        8

In [4]: ##Bitwise Right shift
        a=0b1010
        b=a>>1
        print(b)
        5
```

5. Create a list containing int type data of length 15. Then write a code to check if 10 is present in the list or not.

**ANSWER:**

```
In [6]: l1=[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14]
        print(len(l1))
        print(10 in l1)

        15
        True
```