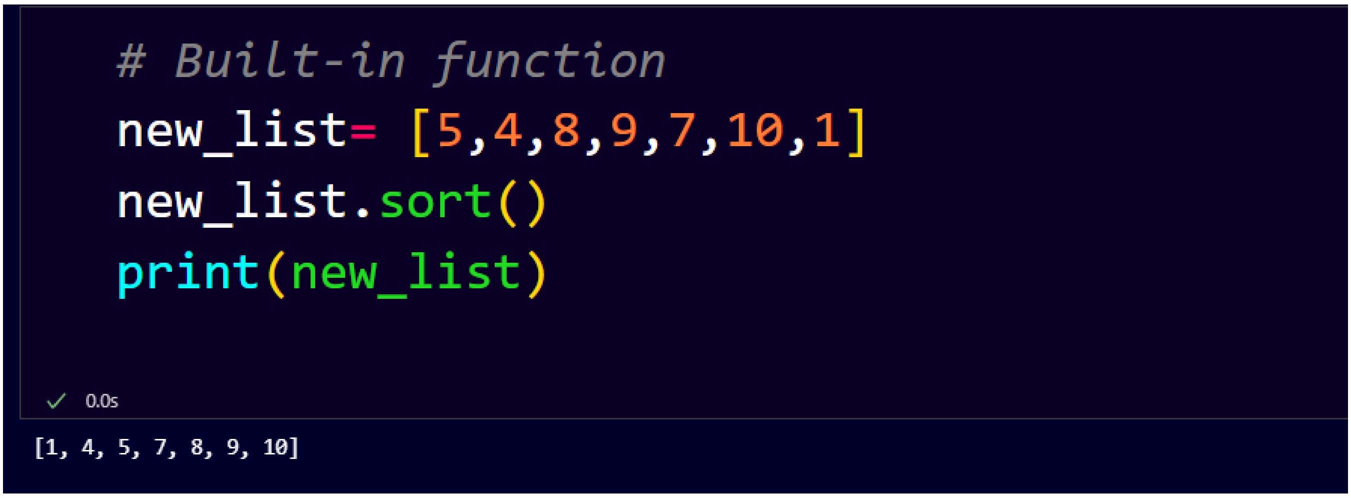
1. In Python, what is the difference between a built-in function and a user-defined function? Provide an example of each.

Solution:- Built-in function:- The functions which are provided by the python means you can directly use it you need to write the code only

Ex:- **Sort Function**



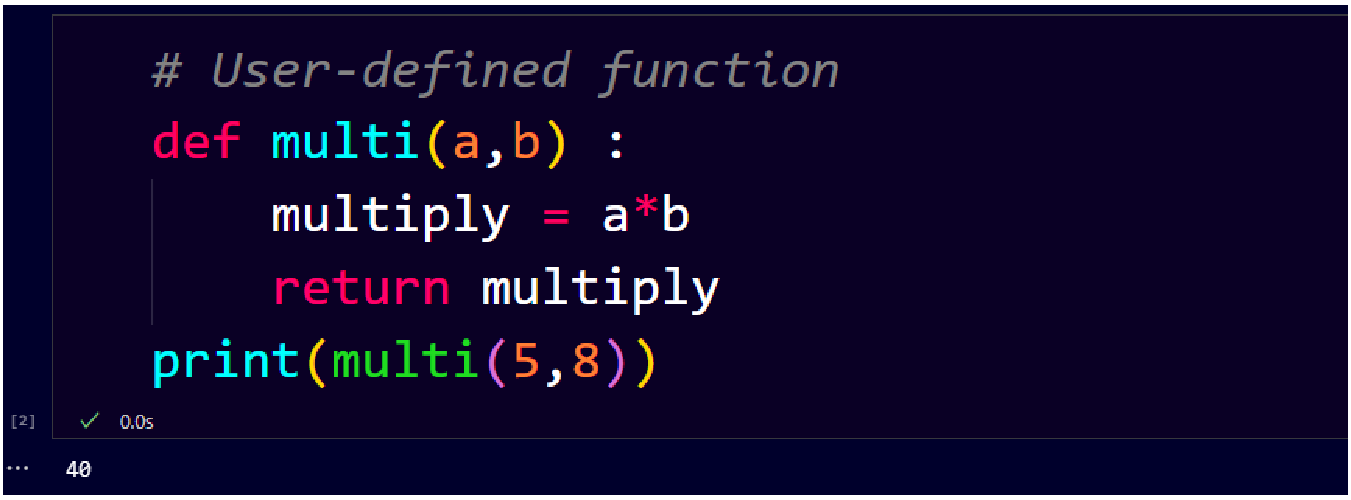
```
# Built-in function
new_list= [5,4,8,9,7,10,1]
new_list.sort()
print(new_list)
```

✓ 0.0s

```
[1, 4, 5, 7, 8, 9, 10]
```

User defined function:- These type of function are defined by the user for the user specific task

Ex:- **Multiplication of two number**



```python
# User-defined function
def multi(a,b) :
    multiply = a*b
    return multiply
print(multi(5,8))
```

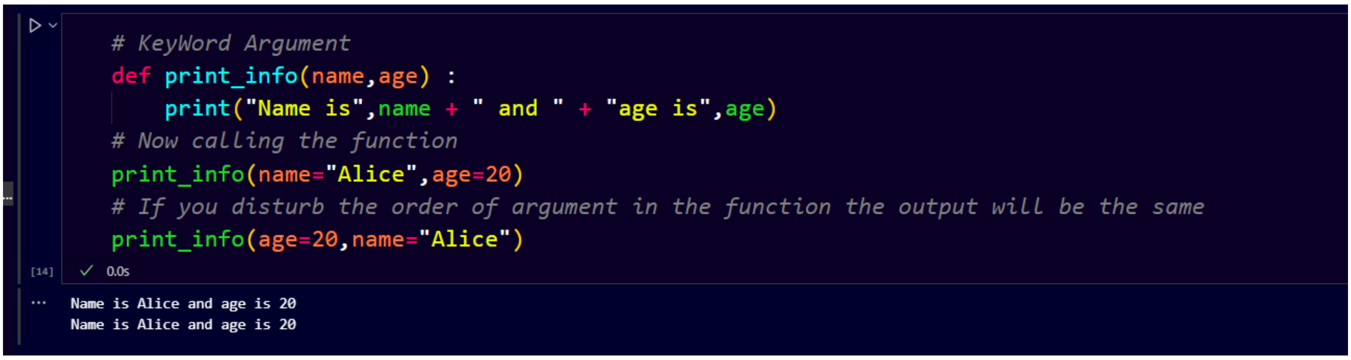
```
[2]  ✓ 0.0s
...  40
```

2. How can you pass arguments to a function in Python? Explain the difference between positional arguments and keyword arguments.

Solution:- At the time of the declaration of the function in the bracket you can pass the argument

Keyword Argument:- At the time of the function calling you put an equal sign and give the argument to the function in these case if you disturb the order of the argument then there is no effect on the result

Example:-

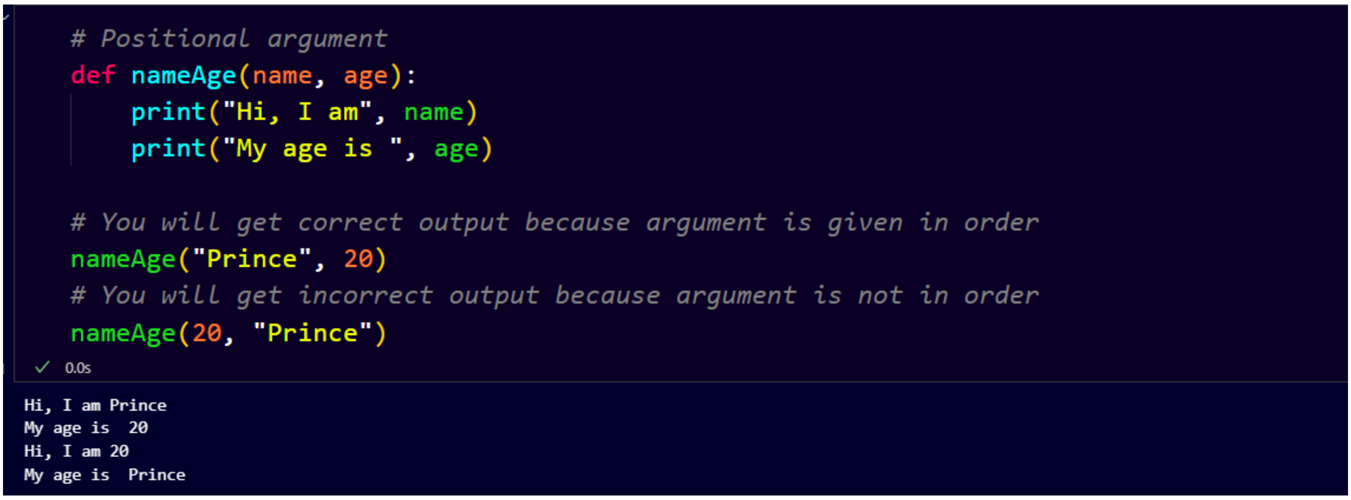


```python
# KeyWord Argument
def print_info(name,age) :
    print("Name is",name + " and " + "age is",age)
# Now calling the function
print_info(name="Alice",age=20)
# If you disturb the order of argument in the function the output will be the same
print_info(age=20,name="Alice")
```

```
Name is Alice and age is 20
Name is Alice and age is 20
```

Positional Argument:- In positional argument you will only the pass the value of the variable which are passed the argument of the function so here if we disturb the order then we get different output

Example:-

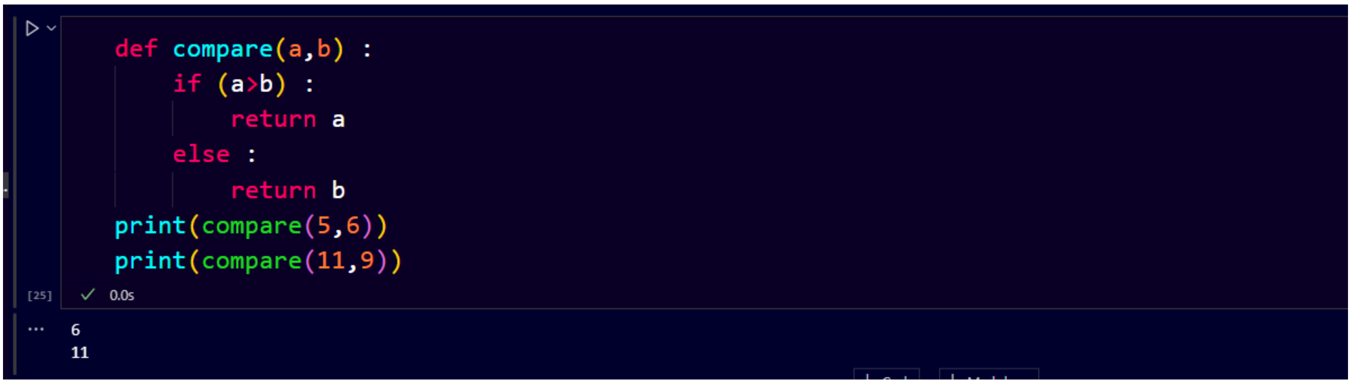


```python
# Positional argument
def nameAge(name, age):
    print("Hi, I am", name)
    print("My age is ", age)

# You will get correct output because argument is given in order
nameAge("Prince", 20)
# You will get incorrect output because argument is not in order
nameAge(20, "Prince")
```

```
Hi, I am Prince
My age is  20
Hi, I am 20
My age is  Prince
```

3. What is the purpose of the return statement in a function? Can a function have multiple return statements? Explain with an example.

Solution:- Return function return the value to the function now you can use these function anywhere outside the code and print the value. Yes the function can have multiple return statement



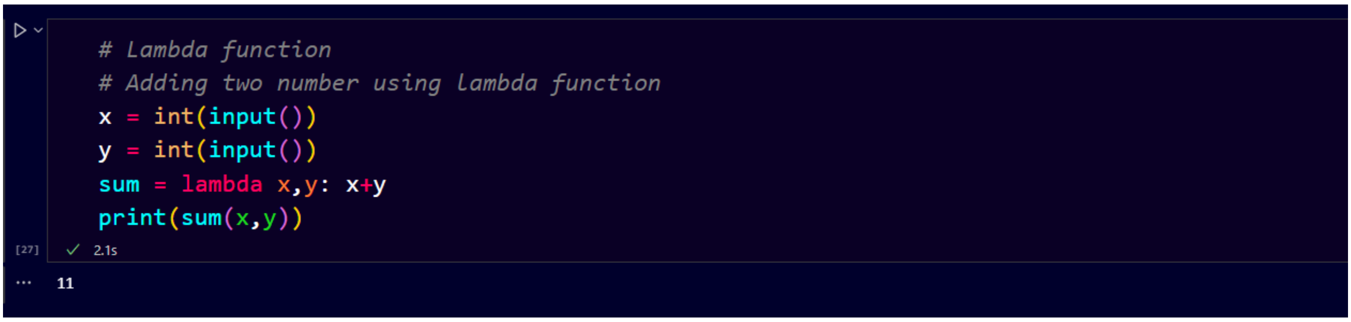
```python
def compare(a,b) :
    if (a>b) :
        return a
    else :
        return b
print(compare(5,6))
print(compare(11,9))
```

```
6
11
```

4. What are lambda functions in Python? How are they different from regular functions? Provide an example where a lambda function can be useful.

Solution:- Lambda function is also the user defined function which is mainly use for the calculating the arithmetic expression it is not useful for the multiple if -else or complex for loop

Example:-



```python
# Lambda function
# Adding two number using lambda function
x = int(input())
y = int(input())
sum = lambda x,y: x+y
print(sum(x,y))
```

```
11
```

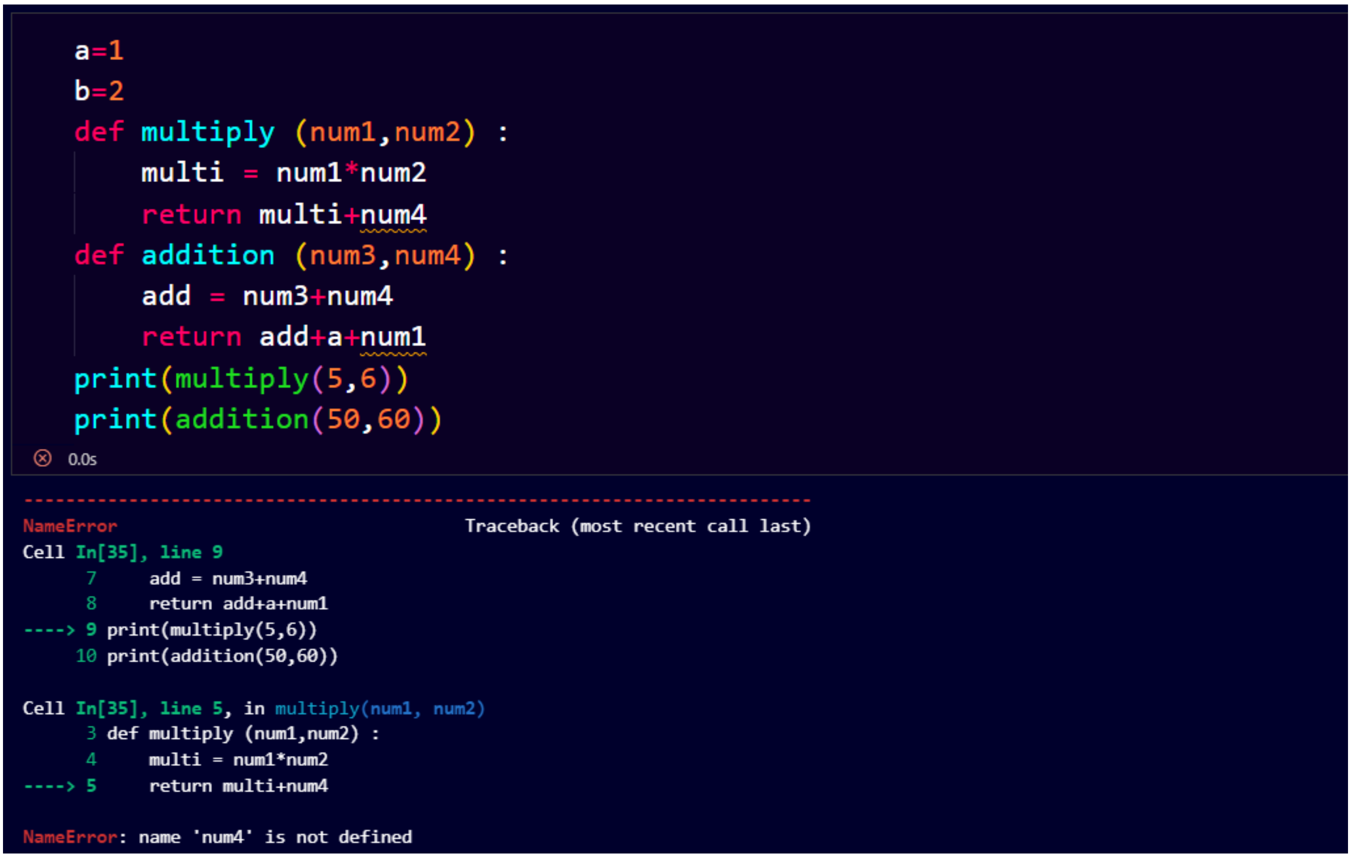
5. How does the concept of "scope" apply to functions in Python? Explain the difference between local scope and global scope.

Solution:- Local Scope-> Local scope is the scope which is belong to the particular block of the code and can be access to that block of the code only

Global Scope-> Global scope is the scope which is define outside the function or any class and can be use any where inside the code

Example:-

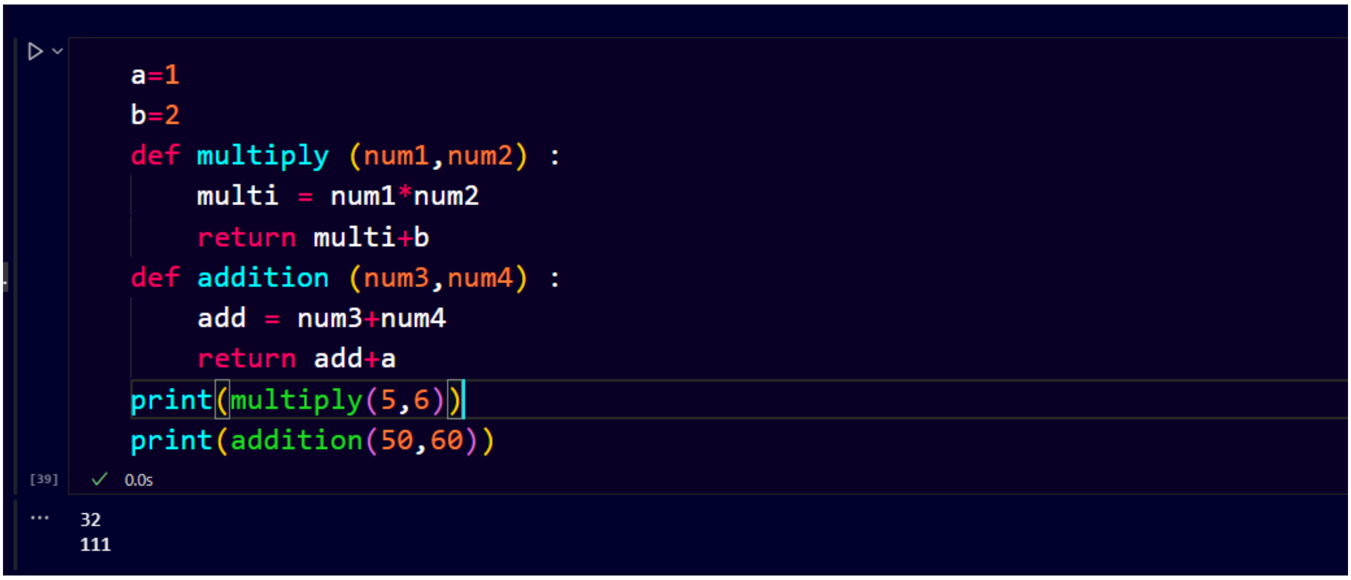
Local Scope-> Here we are getting the error because in the return statement we are using the local variable of the function



```
a=1
b=2
def multiply (num1,num2) :
    multi = num1*num2
    return multi+num4
def addition (num3,num4) :
    add = num3+num4
    return add+a+num1
print(multiply(5,6))
print(addition(50,60))
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[35], line 9
      7     add = num3+num4
      8     return add+a+num1
----> 9 print(multiply(5,6))
     10 print(addition(50,60))

Cell In[35], line 5, in multiply(num1, num2)
      3 def multiply (num1,num2) :
      4     multi = num1*num2
----> 5     return multi+num4

NameError: name 'num4' is not defined
```

Global Scope:- Here we are not getting the error because we are adding the a and b in the return statement which is the global variable



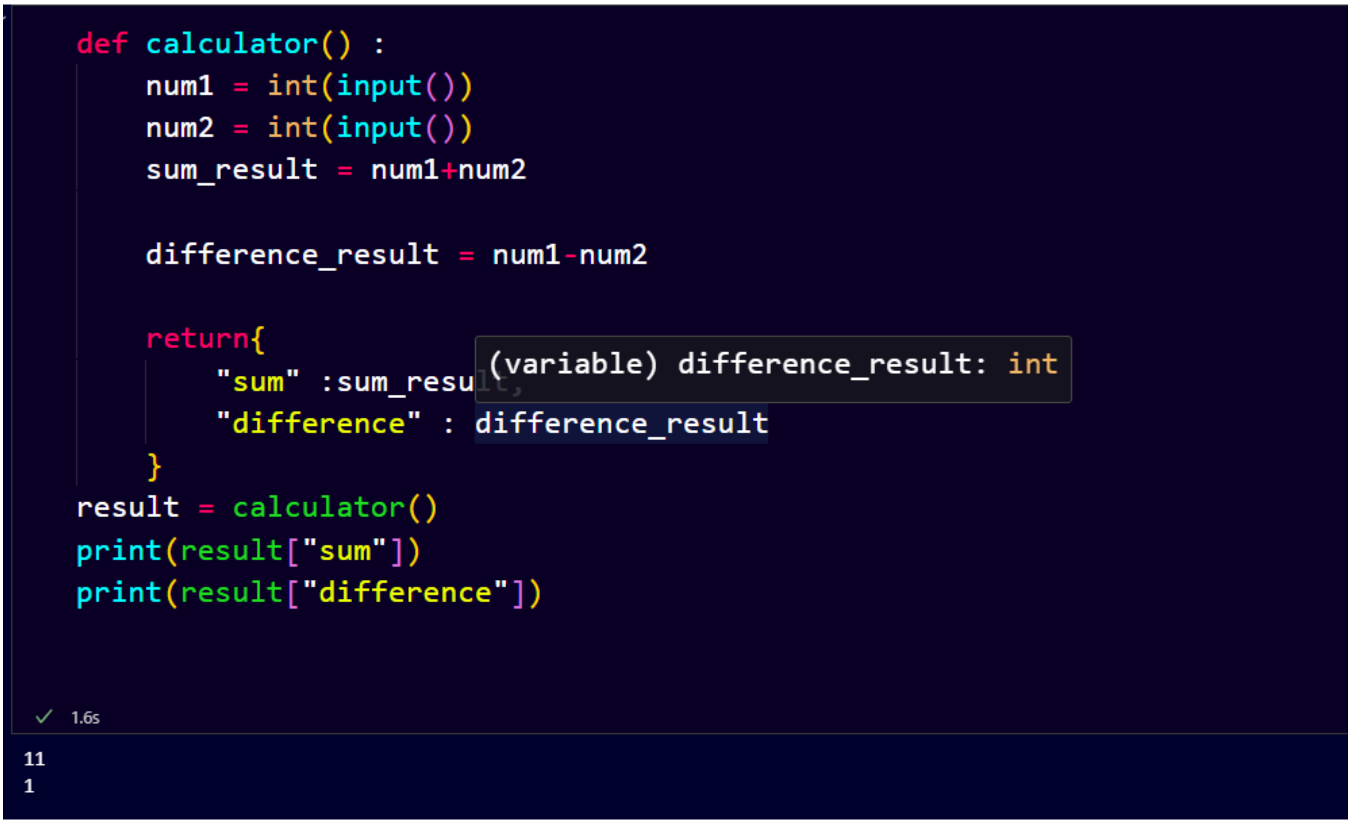
```
a=1
b=2
def multiply (num1,num2) :
    multi = num1*num2
    return multi+b
def addition (num3,num4) :
    add = num3+num4
    return add+a
print(multiply(5,6))
print(addition(50,60))
```

```
32
111
```

6. How can you use the "return" statement in a Python function to return multiple values?

Solution:- We can create the dictionary to return the multiple value in

Example:-

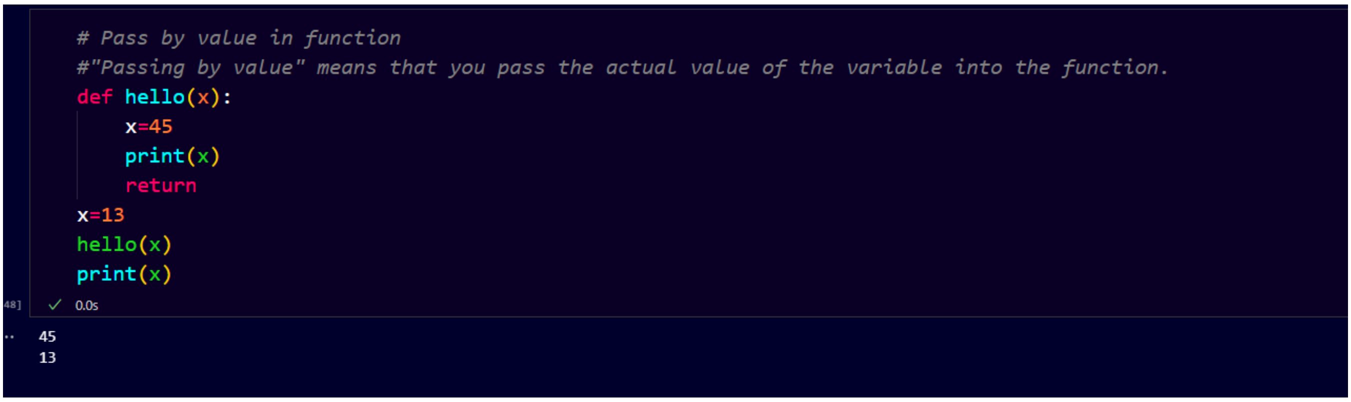


```
def calculator() :
    num1 = int(input())
    num2 = int(input())
    sum_result = num1+num2

    difference_result = num1-num2

    return{
        "sum" :sum_result,
        "difference" : difference_result
    }
result = calculator()
print(result["sum"])
print(result["difference"])
```

```
11
1
```

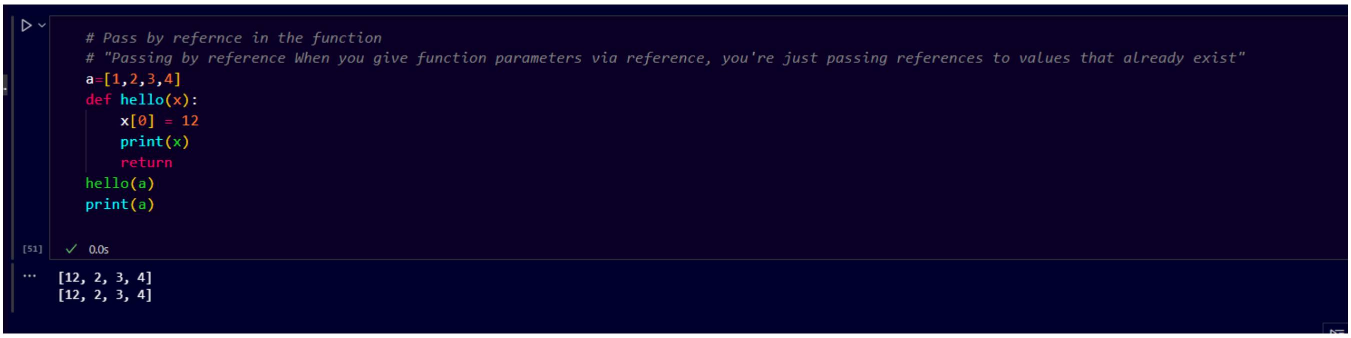
7. What is the difference between the "pass by value" and "pass by reference" concepts when it comes to function arguments in Python?

Solution:-



```python
# Pass by value in function
#"Passing by value" means that you pass the actual value of the variable into the function.
def hello(x):
    x=45
    print(x)
    return
x=13
hello(x)
print(x)
```

```
45
13
```



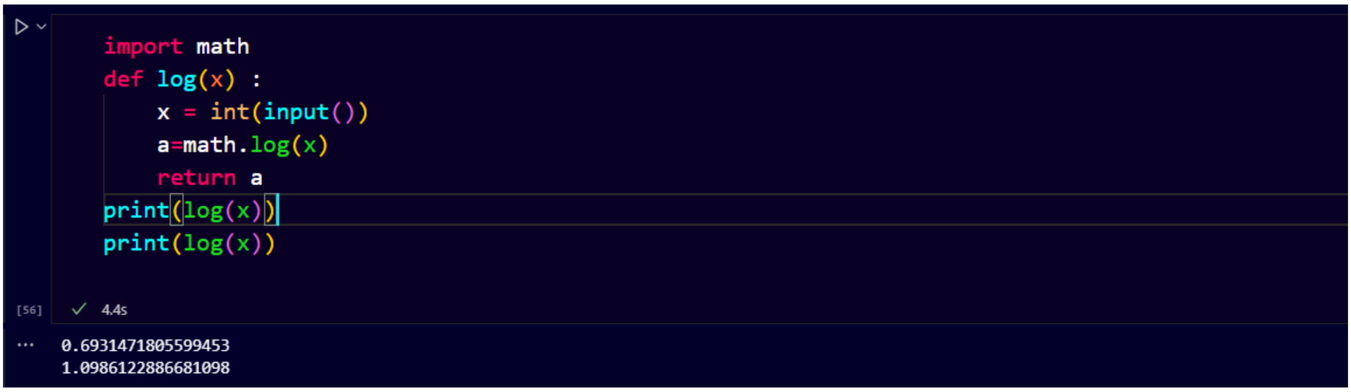
```python
# Pass by refernce in the function
# "Passing by reference When you give function parameters via reference, you're just passing references to values that already exist"
a=[1,2,3,4]
def hello(x):
    x[0] = 12
    print(x)
    return
hello(a)
print(a)
```

```
[12, 2, 3, 4]
[12, 2, 3, 4]
```

8. Create a function that can intake integer or decimal value and do following operations:

   a. Logarithmic function (log x))

Solution:-

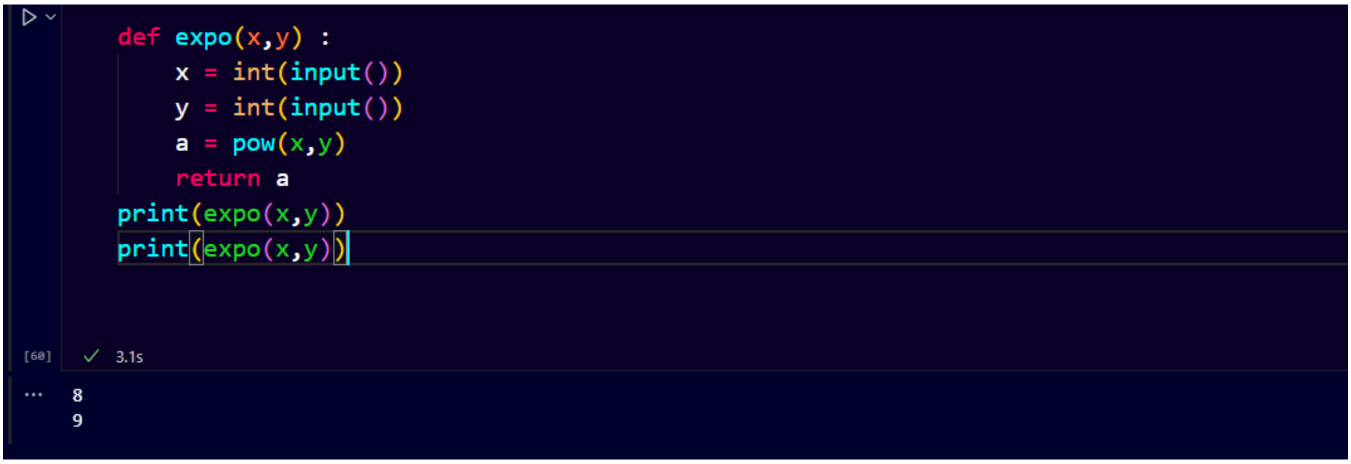


```
import math
def log(x) :
    x = int(input())
    a=math.log(x)
    return a
print(log(x))
print(log(x))
```

```
0.6931471805599453
1.0986122886681098
```

b. Exponential function (exp(x))

Solution:-

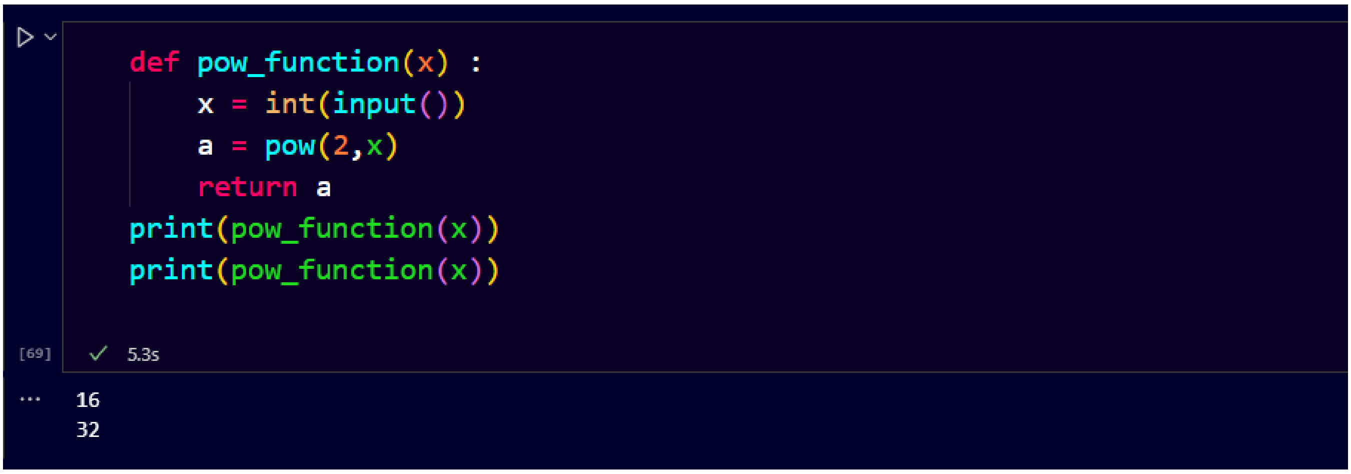


```
def expo(x,y) :
    x = int(input())
    y = int(input())
    a = pow(x,y)
    return a
print(expo(x,y))
print(expo(x,y))
```

```
8
9
```

c. Power function with base 2 (2^x)

Solution:-

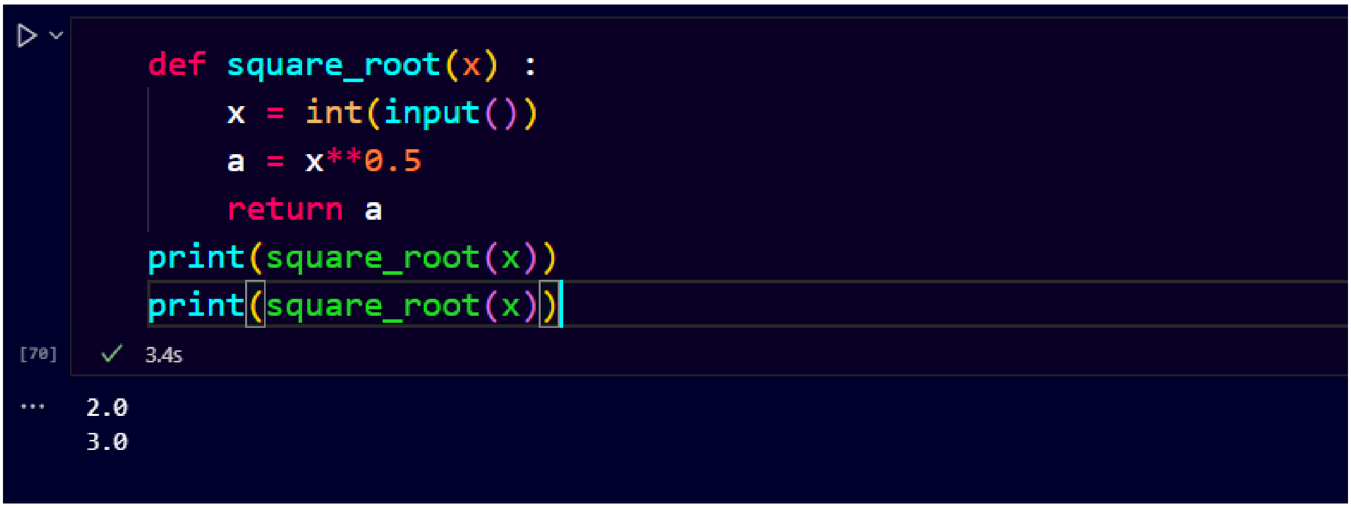


```python
def pow_function(x) :
    x = int(input())
    a = pow(2,x)
    return a
print(pow_function(x))
print(pow_function(x))
```

```
16
32
```

d. Square root

Solution:-

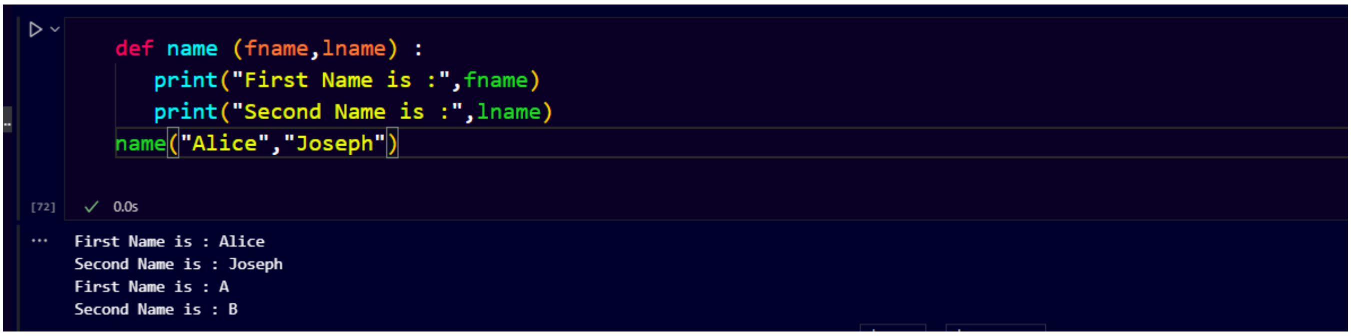


```python
def square_root(x) :
    x = int(input())
    a = x**0.5
    return a
print(square_root(x))
print(square_root(x))
```

```
2.0
3.0
```

9. Create a function that takes a full name as an argument and returns first name and last name.

Solution:-



```
def name (fname,lname) :
    print("First Name is :",fname)
    print("Second Name is :",lname)
name("Alice","Joseph")
```

```
First Name is : Alice
Second Name is : Joseph
First Name is : A
Second Name is : B
```