CT60A0203
Introduction to Programming: Python
Week 5

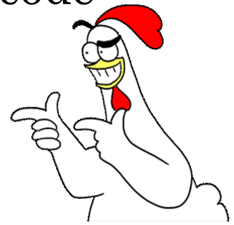
Learning objectives: Python procedure, function and function calling

- To learn procedure and function (subprograms) in Python
- To experience how to declare a procedure and function in Python
- To know how to call/use functions in Python programs
- To know Python namespaces and scope of the variables (self study)

At the conclusion of this lecutre, students will be able to define functions and learn the purpose of creating functions with return value in their programs.

- You have already used some **Python's built-in** functions in your code

  Do you know that?



- Do you remember random.randint()? # Python's built-in function

- Another built-in function → **math.sqrt(2)** → using sqrt() function already defined in Python to get the square root value of 2.
- So, if your program needs to do some square root functions, you can use **math.sqrt()** to do so. **So handy is it not?**

- **print() is a procedure we often used in our program is also a Python's built-in function.**

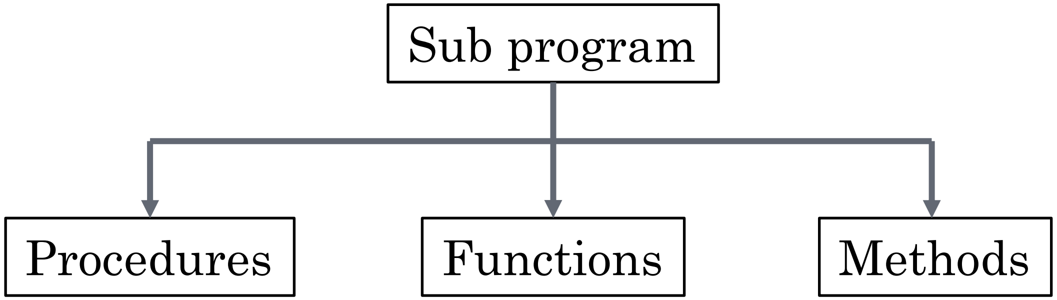
- Can we create our own subprograms (procedure/function)?

Yes of course, that's what you are going to learn and practice today.

➢What is sub program in programming? And why do we need it in programming?

It is a portion of the program which are set of instructions/block of commands to perform certain tasks.
The same part of program (function) can be used repeatedly instead of writing it again.

Example: **random.randint()** can be used in your programs repeatedly!



Sub program

Procedures

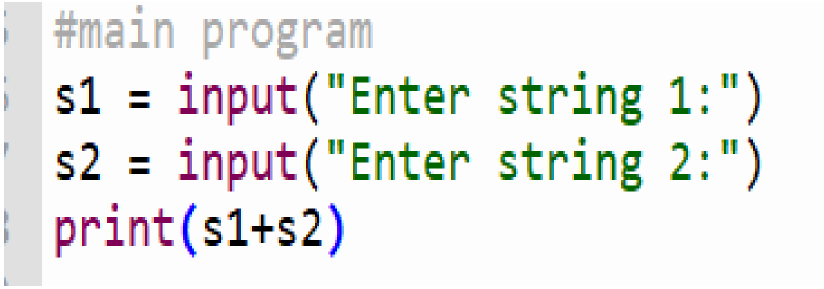
Functions

Methods

Ok. How to define our own procedure and function in Python?

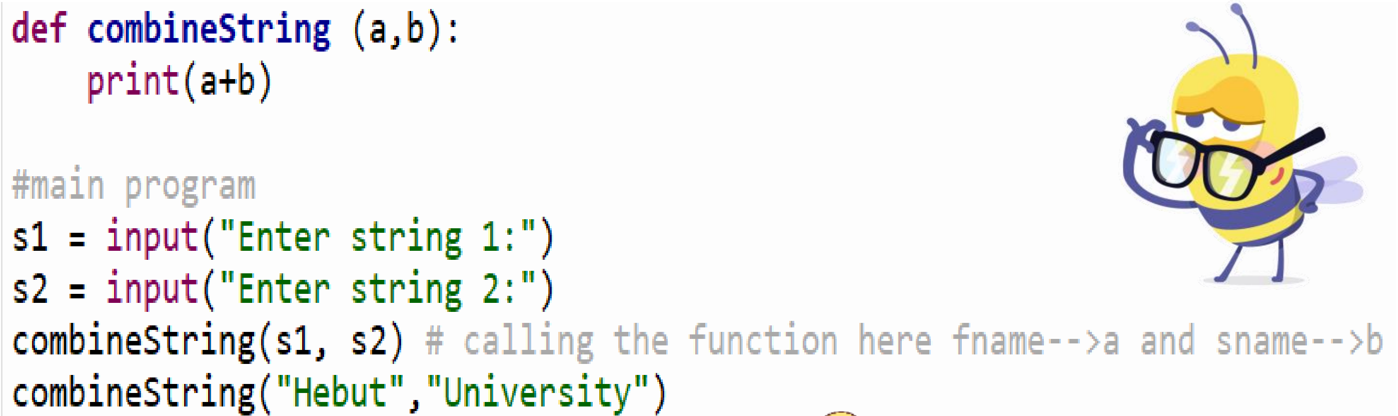
➢Creating our own function is called user defined function (created by user. Yes, we are).

So, how to define a procedure/function and use it in your code?



```
#main program
s1 = input("Enter string 1:")
s2 = input("Enter string 2:")
print(s1+s2)
```

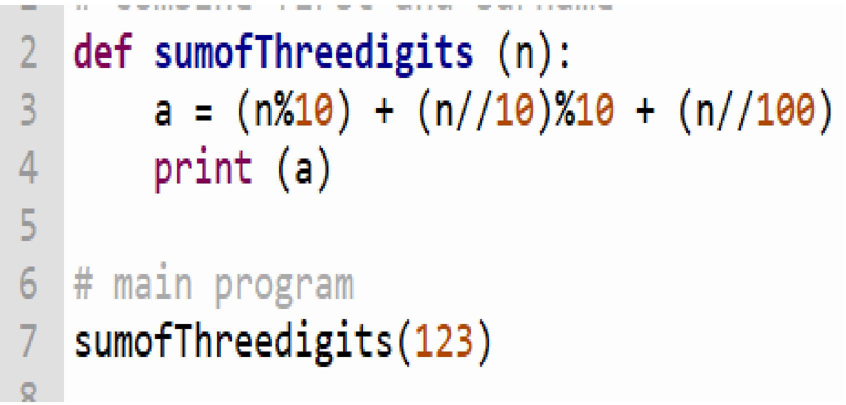
- Let us create a subprogram (procedure/function) “combineString” and use whenever we need to combine any two strings.



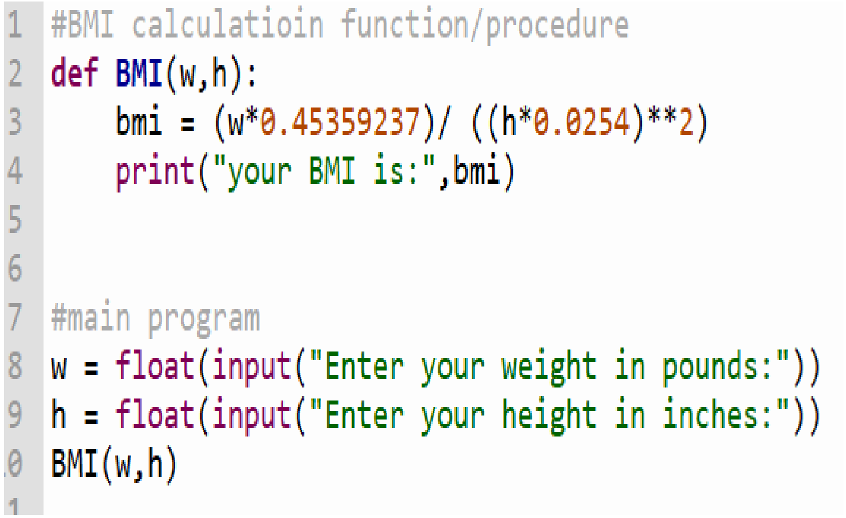
```
def combineString (a,b):
    print(a+b)

#main program
s1 = input("Enter string 1:")
s2 = input("Enter string 2:")
combineString(s1, s2) # calling the function here fname-->a and sname-->b
combineString("Hebut","University")
```

```
def  <function name> ( <with or without any parameters>):
          <block of code
            ............
            ..............
           <return or print or nothing> >
```

procedure

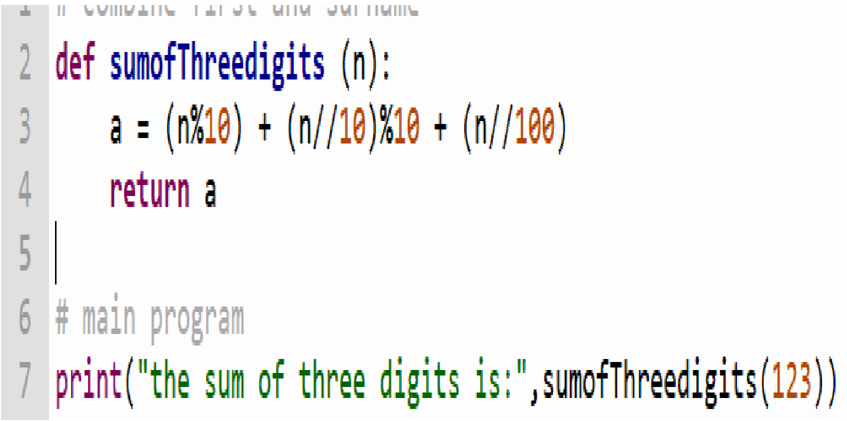


```
def sumofThreedigits (n):
    a = (n%10) + (n//10)%10 + (n//100)
    print (a)

# main program
sumofThreedigits(123)
```

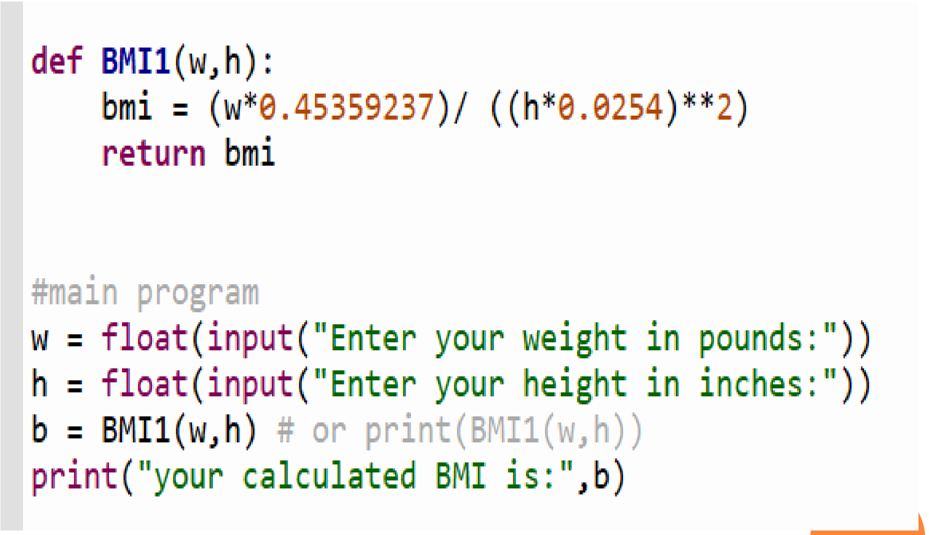


```
#BMI calculatioin function/procedure
def BMI(w,h):
    bmi = (w*0.45359237)/ ((h*0.0254)**2)
    print("your BMI is:",bmi)


#main program
w = float(input("Enter your weight in pounds:"))
h = float(input("Enter your height in inches:"))
BMI(w,h)
```

function

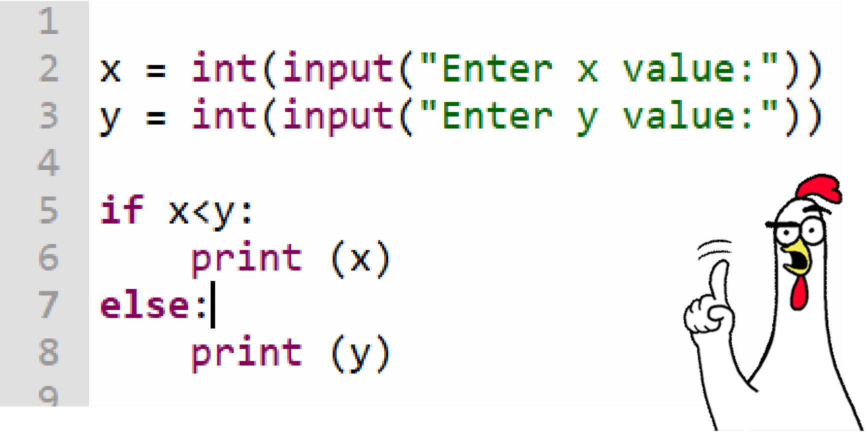


```
def sumofThreedigits (n):
    a = (n%10) + (n//10)%10 + (n//100)
    return a

# main program
print("the sum of three digits is:",sumofThreedigits(123))
```



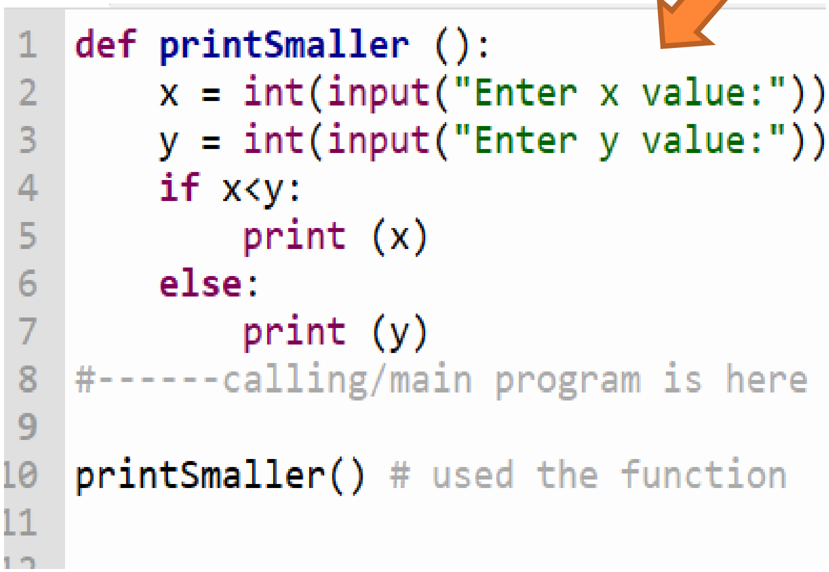
```
def BMI1(w,h):
    bmi = (w*0.45359237)/ ((h*0.0254)**2)
    return bmi


#main program
w = float(input("Enter your weight in pounds:"))
h = float(input("Enter your height in inches:"))
b = BMI1(w,h) # or print(BMI1(w,h))
print("your calculated BMI is:",b)
```

- Suppose you write a program that prints the smallest one among two numbers.

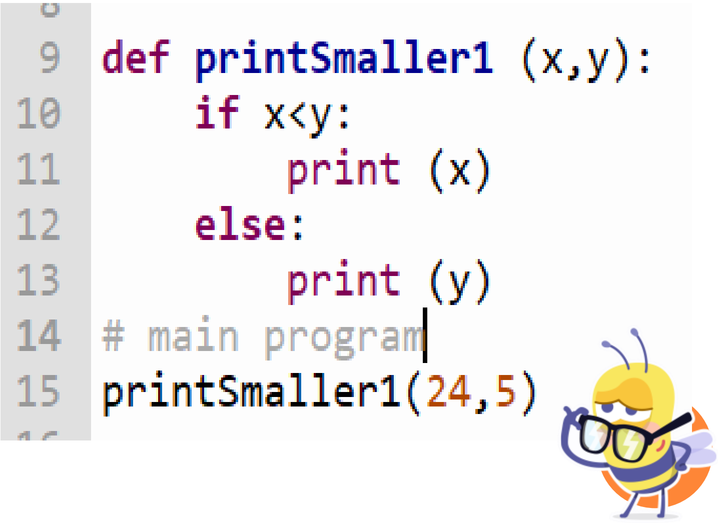


```
x = int(input("Enter x value:"))
y = int(input("Enter y value:"))

if x<y:
     print (x)
else:
     print (y)
```

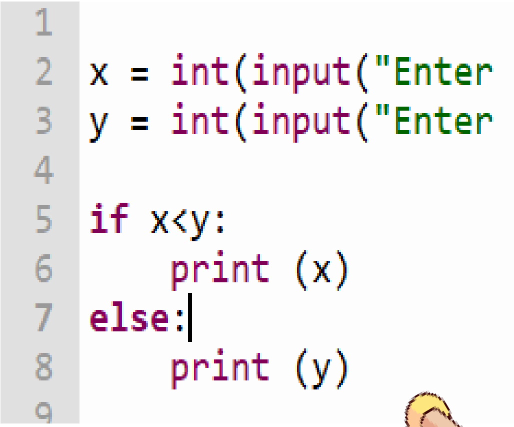
- Your work may need sometimes to use the same code repeatedly. So, you want to create this portion as a readymade function and would like to use it whenever is required.
- **How to do that?**



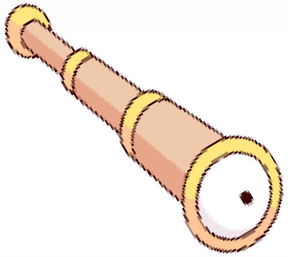
```
def printSmaller ():
    x = int(input("Enter x value:"))
    y = int(input("Enter y value:"))
    if x<y:
        print (x)
    else:
        print (y)
#------calling/main program is here

printSmaller() # used the function
```

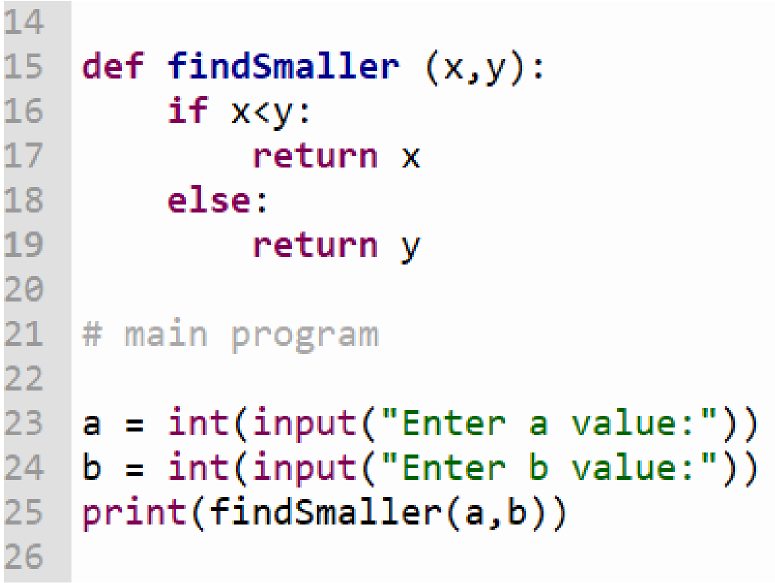


```
def printSmaller1 (x,y):
    if x<y:
        print (x)
    else:
        print (y)
# main program
printSmaller1(24,5)
```

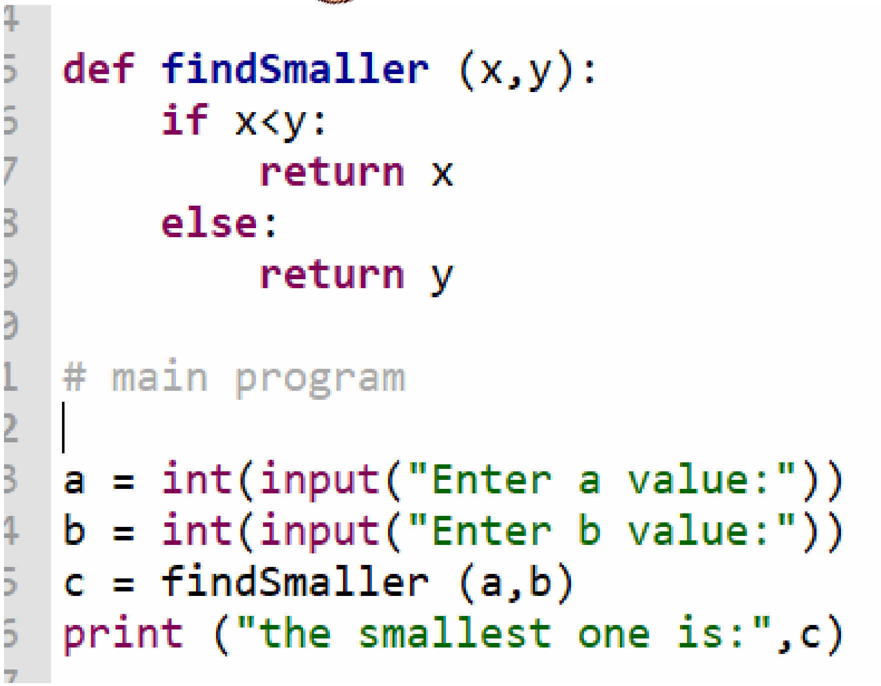


```python

x = int(input("Enter x value:"))
y = int(input("Enter y value:"))

if x<y:
    print (x)
else:
    print (y)

```

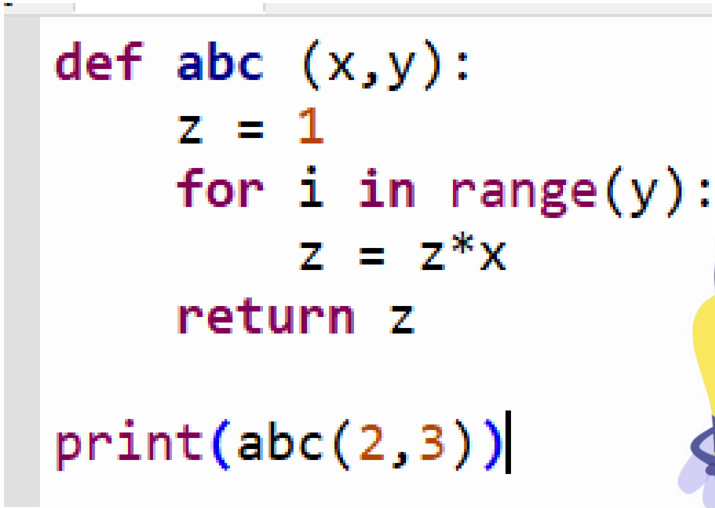




```python

def findSmaller (x,y):
    if x<y:
        return x
    else:
        return y

# main program

a = int(input("Enter a value:"))
b = int(input("Enter b value:"))
print(findSmaller(a,b))

```

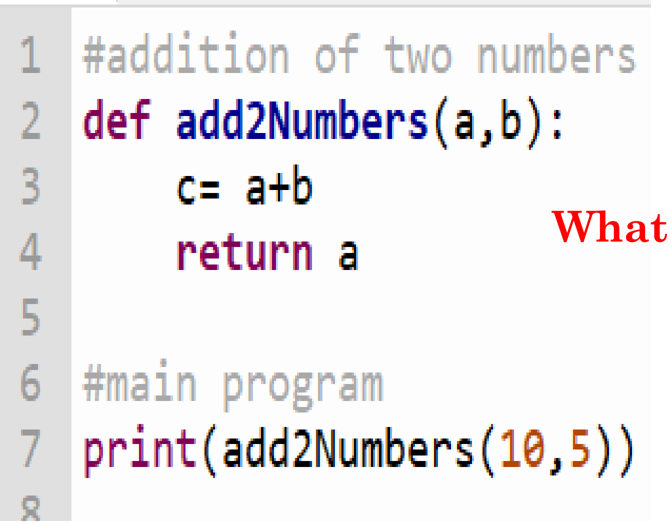


```python

def findSmaller (x,y):
    if x<y:
        return x
    else:
        return y

# main program

a = int(input("Enter a value:"))
b = int(input("Enter b value:"))
c = findSmaller (a,b)
print ("the smallest one is:",c)

```



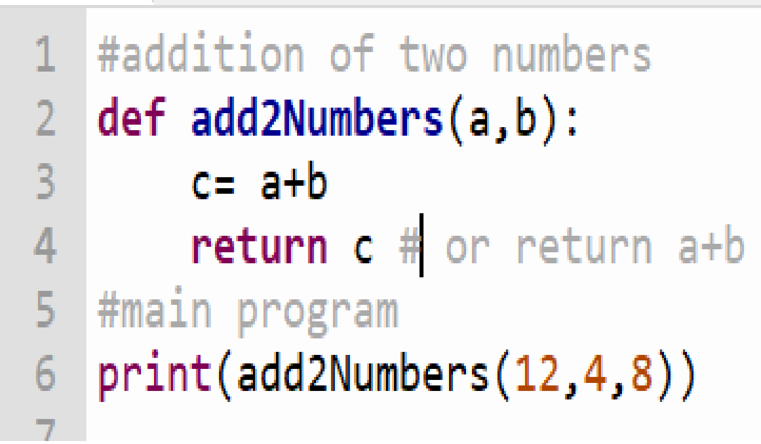
```
def abc (x,y):
    z = 1
    for i in range(y):
        z = z*x
    return z

print(abc(2,3))
```

- ❑ What abc() functions does?
- ❑ What is the output of the code?
- ❑ The name of the function should be given based on its work nature. So, what name would you suggest?

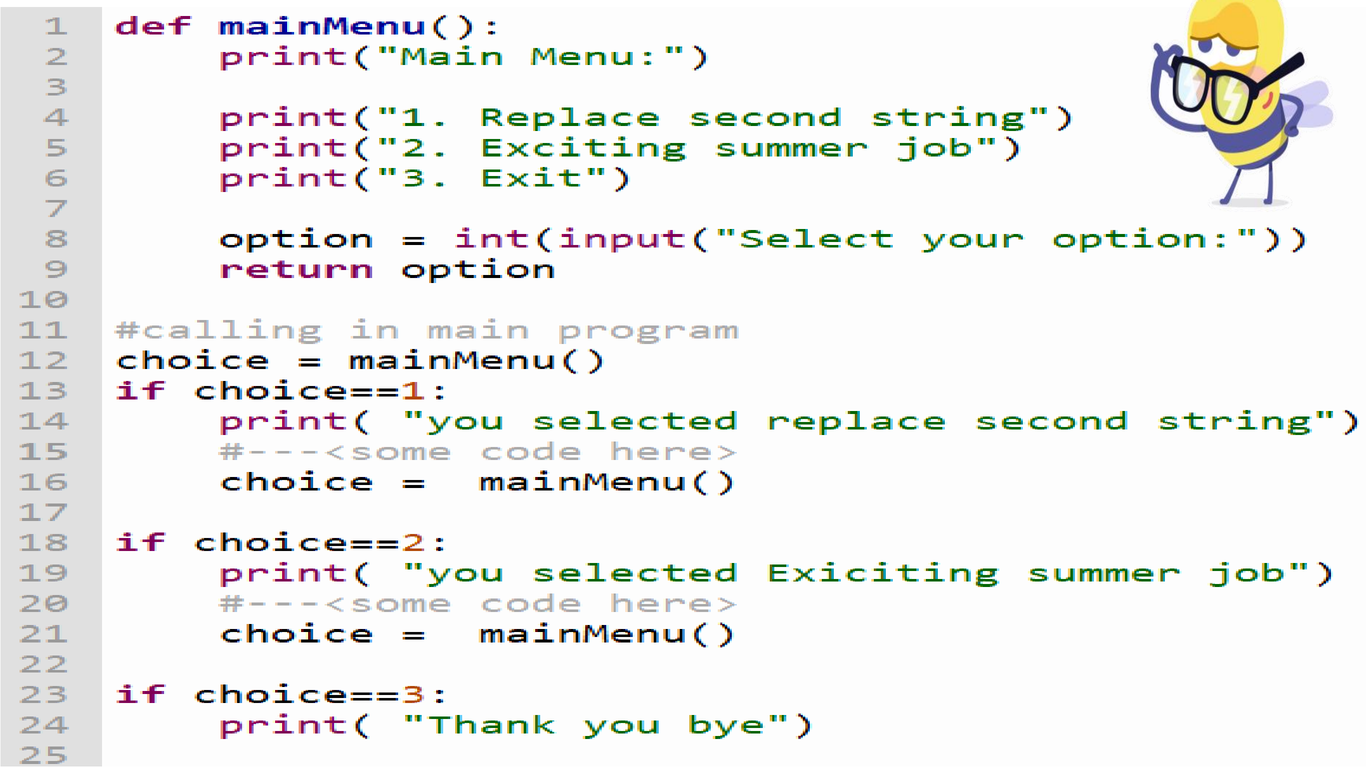


```
#addition of two numbers
def add2Numbers(a,b):
    c= a+b
    return a

#main program
print(add2Numbers(10,5))
```

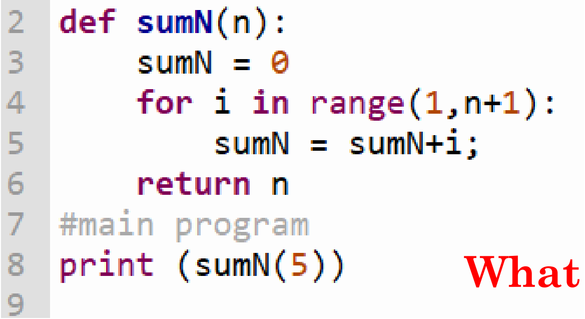
What is the output?



```
#addition of two numbers
def add2Numbers(a,b):
    c= a+b
    return c # or return a+b
#main program
print(add2Numbers(12,4,8))
```

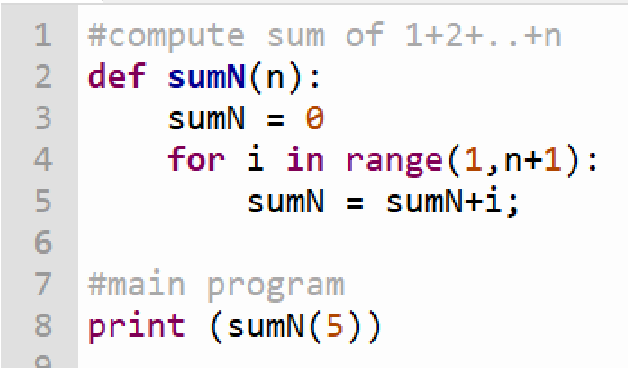


```python
def mainMenu():
     print("Main Menu:")

     print("1. Replace second string")
     print("2. Exciting summer job")
     print("3. Exit")

     option = int(input("Select your option:"))
     return option

#calling in main program
choice = mainMenu()
if choice==1:
     print( "you selected replace second string")
     #---<some code here>
     choice =  mainMenu()

if choice==2:
     print( "you selected Exiciting summer job")
     #---<some code here>
     choice =  mainMenu()

if choice==3:
     print( "Thank you bye")
```



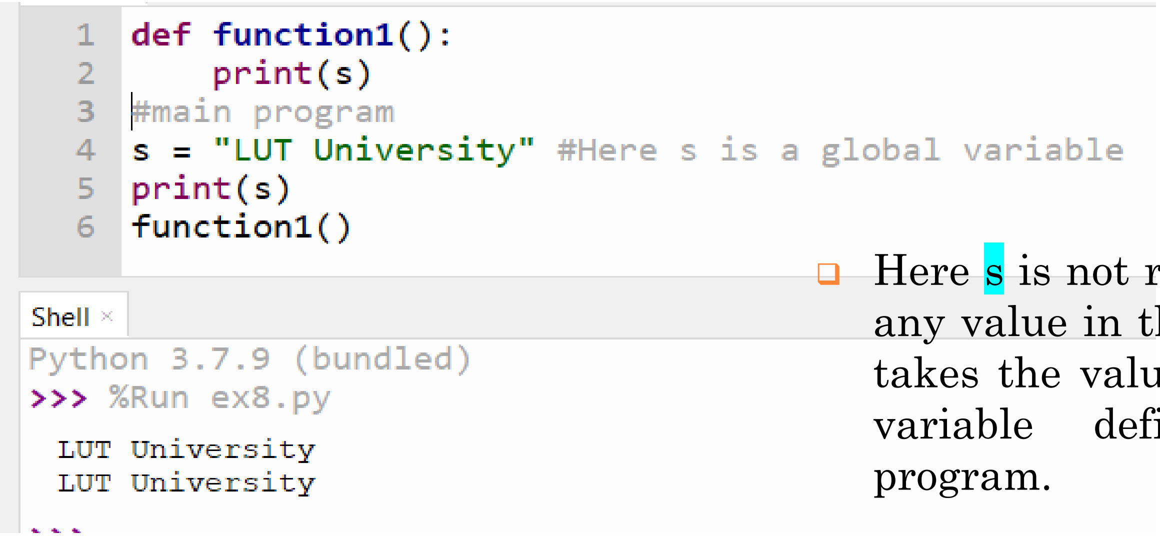
```python
def sumN(n):
    sumN = 0
    for i in range(1,n+1):
        sumN = sumN+i;
    return n
#main program
print (sumN(5))
```

What is the output?



```python
#compute sum of 1+2+..+n
def sumN(n):
    sumN = 0
    for i in range(1,n+1):
        sumN = sumN+i;

#main program
print (sumN(5))
```

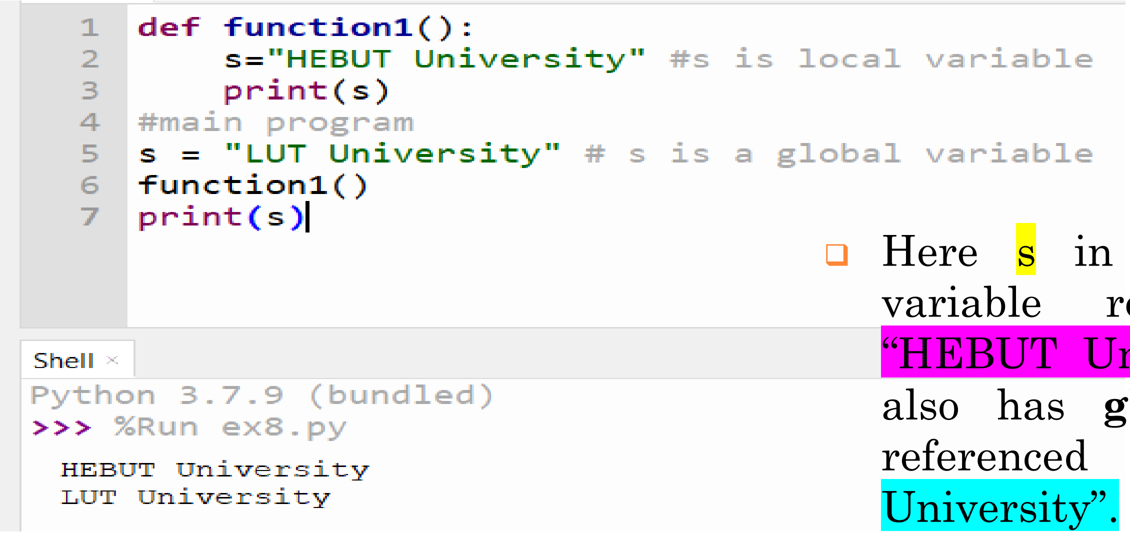
Global and local variables in Functions



```python
def function1():
    print(s)
#main program
s = "LUT University" #Here s is a global variable
print(s)
function1()
```

```
Python 3.7.9 (bundled)
>>> %Run ex8.py

 LUT University
 LUT University
```

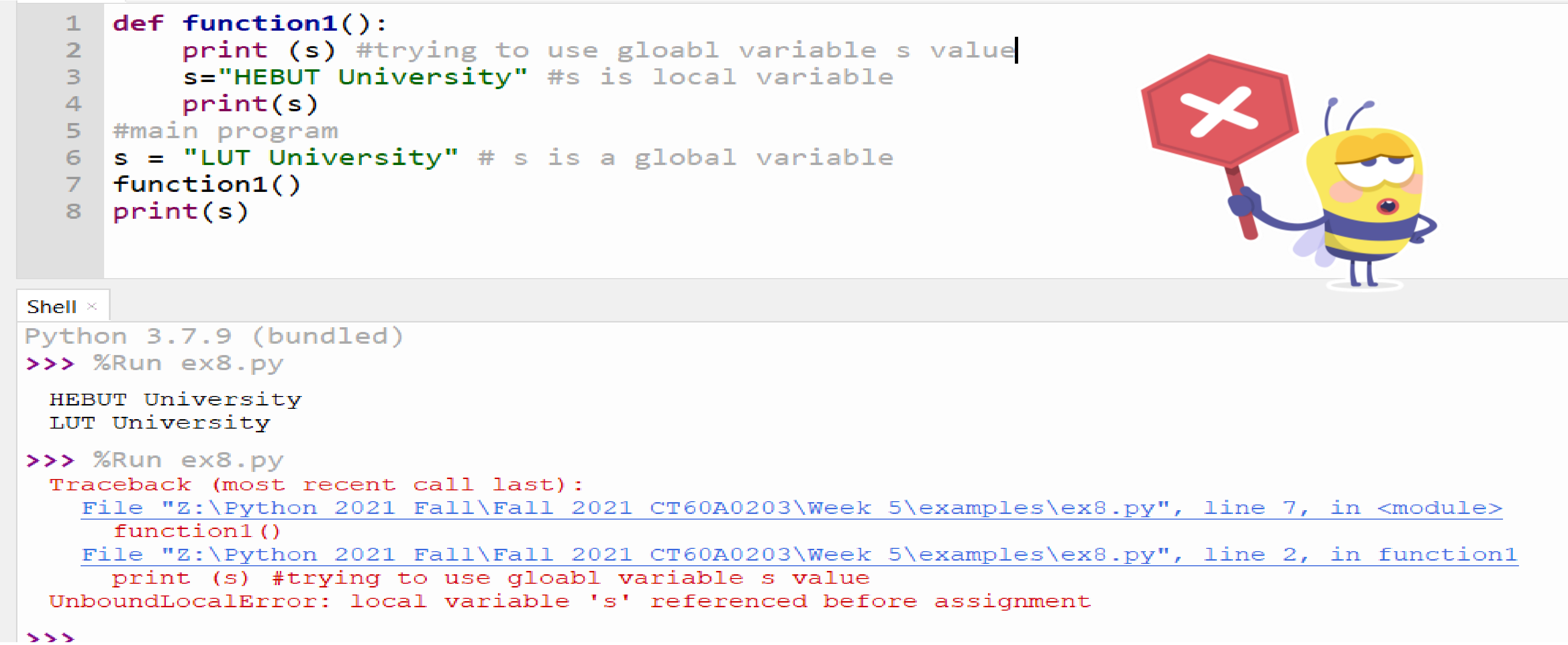
- ❑ Here s is not referenced or defined to any value in the in function1(). So, it takes the value of s which is global variable defined in the main program.



```python
def function1():
    s="HEBUT University" #s is local variable
    print(s)
#main program
s = "LUT University" # s is a global variable
function1()
print(s)
```

```
Python 3.7.9 (bundled)
>>> %Run ex8.py

 HEBUT University
 LUT University
```

- ❑ Here s in function1() is a local variable referenced to value→ "HEBUT University". And the code also has **global variable s** and referenced to value → "LUT University".



```python
def function1():
    print (s) #trying to use gloabl variable s value
    s="HEBUT University" #s is local variable
    print(s)
#main program
s = "LUT University" # s is a global variable
function1()
print(s)
```

```
Python 3.7.9 (bundled)
>>> %Run ex8.py

  HEBUT University
  LUT University

>>> %Run ex8.py
  Traceback (most recent call last):
    File "Z:\Python 2021 Fall\Fall 2021 CT60A0203\Week 5\examples\ex8.py", line 7, in <module>
      function1()
    File "Z:\Python 2021 Fall\Fall 2021 CT60A0203\Week 5\examples\ex8.py", line 2, in function1
      print (s) #trying to use gloabl variable s value
  UnboundLocalError: local variable 's' referenced before assignment

>>>
```

- Python does not allow to define both local variable and global variable to be defined and accessed like this to **avoid ambiguity**. That is, treat local variable is different from global variable or use global variable in both places.

- Refer https://www.python-course.eu/python3_global_vs_local_variables.php to learn more about Python namespaces, local and global variables.