Function Arguments

You can call a function by using the following types of formal arguments −

* Required arguments-must match and cannot be null if the function has x no of parameters.
* Keyword arguments
* Default arguments
* Variable-length arguments (covered in the main handout as arbitrary arguments.

Apart from defining a function with def keyword, one can also use lambda keyword. This is shown in the sample code.

See the different samples of Python function code below demonstrating the different functions

*# This is your function definition. the function name is demo. def is a keywork for define***def** demo1(a\_word): *# there is one parameter called a\_word* print(a\_word) *# what needs to be printed* **return***# calling the function using required arguments.*demo1(**"We are playing"**)  
demo1(**"We are learning online"**)  
  
  
*# ====================Calling using keyword arguments===========================***def** addition(x, y):  
 answer = x + y  
 **return** answer  
  
  
x = addition(x=5, y=7)  
print(x)  
  
  
*# ================Default arguments==========================================***def** addition(x, y=9):  
 answer = x + y  
 **return** answer  
  
  
*# check the function below ignores the default value 9 given above to give 11 as the answer when you print*print(addition(5, 6))  
*# Now, I left out second argument and the program uses value 9 to give answer 14 if you print*print(addition(5))

*# ============Variable length arguments==========================================***def** addition(x, \*y):  
 print(x)  
 **for** a **in** y:  
 print( a)  
 **return**print(addition(3))  
addition(4,2,3)  
  
*#===================#Using Lambda keyword=======================================*revision = **lambda** x,y,z: min(x,y,z)  
  
print(revision(45,23,46))  
  
*#Test each piece of code to understand function arguments  
  
#lets chat about global vs local variables. This was covered when we discussed variables but we will revisit them again.*

**Global vs Local variables**

Variables that are defined inside a function body have a local scope and these are local to the function, and those defined outside have a global scope and can be accessed anywhere in the program.