Functions

We will now discuss how we use functions to control our code. So far, we have been creating variables, lists, tuples etc. but we must define these objects independently every time. If we want do the same thing over and over again then we can define a function.

Function can have no parameters, parameters or call other functions.

Whenever we invoke a function, it creates a new frame, the formal parameter gets bound to the value of actual parameter when the function is called

When the function is called, a new scope/frame is created and then the body of the function is executed

Scope is mapping of names to objects

def f(x): 🡪 formal parameter

x = x + 1

print(‘in f(x): x=’, x)

return x

* return only has meaning inside a function
* only one return executed inside a function
* code inside the function but after the return statement is not executed

SCOPE

* Inside a function, we can access variable defined outside
* Inside a function, we cannot modify a variable defined outside.

Run these examples

Def f(y): def g(y):

x = 1 (x is re-defined in scope of f) print(x) (x is out the scope so it can accessed)

x += 1 print(x + 1) (it uses x = 5 from the outer scope)

print(x) x = 5

x = 5 g(x)

f(x) 🡪 2

print(x) 🡪 5

x = 3