1. To organize the program into chunks that match how we think about the problem in order to reduce the length of coding and save the time of coding
2. To define a function:

def name (parameter)

statement

1. To call a function:

Name the function being executed followed by a list of values, called argument, which are assigned to the parameters in the function definition.

1. Def: Return is a command to evaluate the expression following the return. Then the expression will be returned to the caller as the “fruit” of calling this function.

Why: Because the purpose of a fruitful function is to return a result and if programmers do not arrange to return a value, python will automatically return the value None, the return command is necessary for the caller to call the result.

How: return <value>

1. No, just for fruitful functions. Because purpose of void functions is to not result in a value but to execute to do something useful, there is no need to use return.
2. Parameters are variables that we need to provide to the Python so that it can execute the function. Arguments are the values that we assign to parameters.

Why: Because we want a flexible function that can change according to the arguments we assign to each parameter.

How:

* When defining function:

def name (parameters (separated by “,”)

* When calling function:

name (arguments (appearing in the exact order of parameters in function definition))

1. import <file\_name>

# then each time we use the functions:

file\_name. function\_name

1. from file\_name import function\_name