function call:

- it helps to call a function to execute a task based on the argument passed to the parameter if any
- During execution, it is also requesting Python(PVM) to allocate a temporary memory on the stack for the execution of the function
- The memory for the local variables will be allocated within the stack memory within the respective function

return:

- it is a keyword, and last executable line of code within a function. Any executable line of code which is written after the return statement is triggered will not be executed
- It helps us to return a value from called function back to function call
- It returns the execution flow from called function back to function call
- During execution, it instructs the Python(PVM) to deallocate the memory which was assigned during function call

[Note: Python(compiler) will include a default "return" statement, if the programmer have not included it.]

What is recursion?

- Calling the function itself, i.e., calling back the called function until and unless the base condition fails.
- It is the process when function is calling itself
- 3 important parts of recursion:
    1. recursive call
    2. logic that has to be executed repeatedly
    3. base condition
- As per the PVM each recursive call is an individual function call

Syn:

1. There is no return value from the function:

```
def fun_name(parameter list):
        #base condition

        #logic
        #recursive function call

#initial function call(arguments)
```

2. There is return value from the function:

```
def fun_name(parameter list):
        #base condition

        #logic
        #return recursive function call

#var = initial function call(arguments)
```

2 types of function call in recursion:
1. **initial function call :** it should carry the initial val given
to the parameters
2. **recursive function call**: it should carry the updated val for
the parameters that is required for the next recursive cycle

How to analyze a recursive function?
1. design the function declaration
2. include the logic that has to be repeated
3. recursive fun call with updated arguments and decide the parameters
4. include the base condition

Parameters:
1. They are the variables that represents the updated value
which have to be carry forwarded in next cycle
2. The values that won't be updated but is used for checking some
condition

[Note:

1. Until and unless the function does not complete it's execution, it remains in the stack
2. When the execution of a method is completed it get's unloaded from the stack and gives the execution flow the calling method**]**

Base Condition:

- It is a condition which controls the recursive calls.
- without which an error will occur during run time, generally is **RecursionError,** when the memory of the stack will exceed

Why recursion?

- It helps in solving complex problems in simple way
- It helps to solve iteration problems using recursion and vice versa.
- usage of space is not constant

[Note: Understanding recursive tree is very important**]**