* **Why should we use functions at all?**

**A function is a box**.

In this box we place code that works together to accomplish some specific task.This code may be simple or may be very long and complex.

If we don't use a function to store this code, every time we need the same functionality we will have to write the whole thing again and that is a lot of work, right?

< Nguồn internet >

* **How to define/declare a function?**

def NAME( PARAMETERS ):  
 STATEMENTS

* **How to call/use a function?**

To call a function, use the function name followed by parenthesis:

*Eg: + In file:*

*ft(4, 5, 6)*

*+ Out file:*

*from ft import <file name>*

*ft(4,5,6)*

* **What is return, why and how do we use it?**

Return is to let function return a value:

*def my\_function(x):*

*return 5 \* x*

*print(my\_function(3)) => result : 15*

*def my\_function(x):*

*5 \* x*

*print(my\_function(3)) => result : none*

* **Do we have to use return in every function?**

No. **void function** The opposite of a fruitful function: one that does not return a value. It is executed for the work it does, rather than for the value it returns

For example:

**def my\_function():**

**print("Hello from a function")**

This function return nothing. It just needs to print out the string

* **What are function arguments/parameters, why and how we use it?**

**parameter** A name used inside a function to refer to the value which was passed to it as an argument.

Information can be passed to functions as parameter. Parameters are specified after the function name, inside the parentheses. You can add as many parameters as you want, just separate them with a comma.

*def my\_function(fname,lname):*

*print(fname + lname)*

*my\_function(“Phi",”Long”)*

* **How to use function from a different file other than our currently working file?**

There isn't any need to add file.py while importing. Just write**from file import function**, and then call the function using **function(a, b)** or **import file** and then call the function using **file.function(a,b)**