Function Analogy

Each function is like a person doing a specialized task in a company.

Whenever someone tells someone else to do a task though, rather than continuing their work, they patiently wait until the person they asked to complete a task does so…

**A function does nothing unless you CALL IT (employees are lazy inherently =))**

When we call the function, we may have to give it some information so that it can do its job…

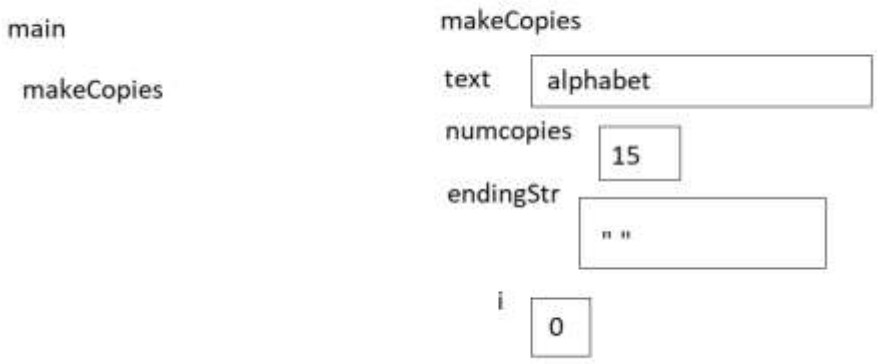
For example, the copy guy needs to know:

a) What to copy (string)
b) How many copies to make (int)
c) Mode (single/double sided) (Boolean)

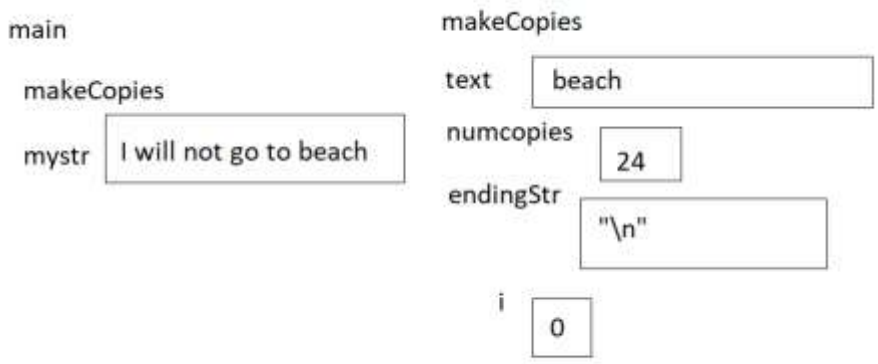
After they finish, (a) you will find out that they are done and (b) they may give you some information…

Copy guy returns to you the bill…

Picture for copy example:




And the second example:




**ONE BIG BENEFIT OF FUNCTIONS: CODE REUSE**

**ANOTHER: CODE ORGANIZATION ALLOWS FOR THE PROGRAMMER TO KEEP TRACK OF LESS AT ANY GIVEN TIME.**

**Key issues people have trouble understanding:**

**Difference between actual and formal parameters.**

**Building a proper mental model of what happens in the computer when a function is called.**

When writing a function, we typically assume that the **formal parameters** (the variables listed on the function definition line), already have values.

Income < 10000 $0 taxes

10000 < income < 20000, 10%

Income > 20000, you pay none of the first $10k, 10% on the next 10k, 20% on the rest.

40000-> 0 on the first 10000

1000 on the next 10000 (10% of 10000)

4000 on the last 20000 (20% of 20000)

---

5000 tax in total

When we test the function via unit testing, we must **CALL** the function, and we should call it multiple times with different test cases, which test out different options of the function. When we call the function, we use the name of the function we do NOT say “def”, and we provide **actual parameters**. These actual parameters do NOT need to be variables. Rather, they can be any expressions of the same type as the corresponding formal parameter.

So each actual parameter is **evaluated** and then those values are copied into the boxes for the corresponding formal parameters.