Usually, we don't need to install a program every time we need to use it, we just run it when we need it. We can do the same with a part of our algorithm or program code that occurs more than once. Repetitive actions can be recorded only once, and then simply run as a separate program within a large one. This is called a function. Let's take a closer look at what it is and how it works.

What is a function

So, a function is some sequence of repetitive actions within a larger program. For example, in our code, we need to execute the same algorithm three times:

add two numbers

multiply the result by two

output to the console

add two numbers

multiply the result by two

output to the console

add two numbers

multiply the result by two

output to the console

Since the actions are the same, we can generalize our algorithm and write these actions as a function. Every function should have a name, so let's call our function calculate. It will look like this:

add two numbers

multiply the result by two

output to the console

Now it is clear that for any two numbers, we will need to perform these exact actions. We can call the function any number of times from anywhere in the program.

Only one question remains then: how will the generalized function understand what numbers it needs to add in each case? For that, it needs arguments, which we will discuss below.

Function arguments

Function arguments are some data that the function takes as the input to perform some actions with it. In our case, these are two numbers. Arguments are usually passed in parentheses, separated by commas: calculate(5, 3) or calculate(123,1346) and so on.

A function can have as many arguments as you want, it depends on the specific task. Also, it might not have arguments at all. For example, there may be an open\_all\_files function that opens all text files from a directory.

Another example of a function with no arguments is an output function that prints the phrase I did it! to the console after each completed block of code. It takes nothing as input and just prints out the string multiple times.

In any case, no matter how many arguments a function has, there is always the result of its work: a number, a string, an open file, and so on. In some cases, this result is visible to us. For example, when you output it to the console, as we showed to you earlier with the calculate function or the line I did it!.

However, after we see the result, we cannot do anything with it. It cannot be further used in the code, passed as an argument to another function. Functions that perform certain actions but do not return a result are often referred to as procedures.

If we need to do something else with the result, we need to make it returnable. Let's take a closer look at how to do this.

Return value

If the function returns a result, it means that the function has received some specific answer such as a number or a string that can be used further in the program. Thus, you can add such action to the calculate function that will return the result of its work. Then the algorithm will be like this:

add the numbers

multiply the result by two

output to the console

return the result

The return is an operator with which we return the result. It is used in all programming languages.

Great! Now the function gives us the resulting number and we can use it to perform further calculations.

Functions that take no arguments can return results too. Usually, it will always be the same result (but sometimes there are exceptions, which we will talk about later). For example, such a function may return 0 indicating that the program has successfully finished its work. For now, we won't go deep into the details of such functions. Just remember that they also exist. In later topics, we will talk about why they are needed and how to write them.

Conclusion

To sum up,

a function is a part of a program that contains an algorithm for repetitive actions.

a function is needed in order not to duplicate the same actions in the code several times.

a function can take arguments as input and do something with them.

a function can return the result of its actions, which can be used later in the program.