LEARN PYTHON WITH NO PROGRAMMING EXPERIENCE:

WHY, HOW, AND WHEN TO USE FUNCTIONS



JEAN PAUL KNIGHT

Learn Python With No Programming Experience: Why, How, and When to Use Functions

Jean Paul Knight

This book is for sale at http://leanpub.com/pythonbook

This version was published on 2018-08-02

© 2018 PythonLove.com

Contents

Why You Need Functions **1**

Why You Need Functions

If you think you don't need functions, you're kind of right. In theory, you can do almost anything without using functions. The only trouble is, it will be a long, painful struggle. In practice, functions serve a very important role.

> The truth is, for small scripts, functions may not make a big difference. However, if you want to write full-fledged programs, functions play a very important role. In fact, the more you understand functions, the greater your ability to write complex programs will be.

Functions make it easier to manage complex programs and applications.

Let's face it: looking at a mass of code can be frustrating and intimidating. Badly organized code put a strain on everybody. It is hard to read and difficult to fix.

However, functions allow you to organize your code into neat "chunks". You can then rearrange them in any way you see fit.

Functions allow you to simplify your code

> Your brain finds it hard to grasp several complex concepts all at once. That's why you need to turn each complex part into something simple. For example, you may have a task that takes 20 lines of code. You can put all that code into one function. Then, the next time you need to do that task, you'll only need *one line* of code to make it work.

Thanks to functions, it's much easier to find and fix bugs - especially in large programs.

Programmers spend a lot of time fixing errors. The more lines of code you write, the more likely you are to encounter an error. The good news is, functions reduce the number of errors you have to deal with.

With functions, you only need to fix the code in one place. If the function's task is causing issues, you simply make a change in that function. This speeds up your work. You'll also have less frustration having to search for the same error, over and over again.

When you use functions, you don't have to write the same code over and over again.

A function acts as a self-contained unit for a specific task. Once you've figured out how to make the task work, simply use a function for it. You don't need to rewrite the code.

Functions make it easier to understand what your code does.

As a programmer, you'll often find yourself going back to your code. For example, you might want to add extra features to a script you've been using. It can be very confusing trying to understand what each part of your code does. That's where you'll find functions to be very handy.

Well named functions make it clear what they do. When you do write functions the right way, it becomes easier to read your code. So, you'll often be able to grasp what the code is about just by reading the function's name. You'll also find it much easier to fix any bugs, errors, or mistakes. That will save you the time you'd have to use to comb over the code. Thank you functions!

When solving a complex problem, functions make it easier.

When writing Python code, you'll constantly find yourself breaking things down. Smaller chunks make each part of the problem small enough to be solved. Functions can store each chunk so you don't have to think about it again.

With functions, you can separate details from the big picture.

In the simplest scenario, you write your functions and keep them in one file. Then you use them in another file, focusing on the main logic of your program. This will make your life much, much easier!

On the other hand...

###without functions... ...life is pure misery. Here are some reasons why:

Without functions, it's almost impossible to write full-fledged Python programs.

A big, complex program is simply a collection of small solutions to various problems. Each solution is connected in a way that makes it useful to you and other people. Without functions, all you have is a single huge mass of code - rather than manageable chunks.

If you don't use functions, your skills will likely remain at the scripting level (as opposed to actual programming).

Advanced programming is all about solving problems and managing complexity. Without a good grasp of functions, you will have a limited ability to create useful solutions. This will affect the level of competency at which you can use your Python skills.

When you don't grasp functions and how they work, it's much harder to advance as a Python programmer.

Since functions are a basic part of Python, you'll be expected to understand them during job interviews. There's no way around it. Remember, a pro is someone who can use basic concepts without having to think about them.

Lack of deep understanding of functions make it harder to learn other concepts such as OOP (Object Oriented Programming).

As you keep using Python, you'll keep coming across invisible walls. The way to cross these walls is by developing your skills. For instance, without OOP it's difficult to handle many problems. But, if you can't use functions you won't be able to learn OOP.

Without functions, your code is boring.

Yep. I said it. Without functions, your code becomes a boring blob of code. It's very hard to understand what your code does. However, a well-named function is like a picture; it's worth a thousand words!

Without functions you have to work way too hard.

When you make a mistake, you'll have to correct in many places. This could lead to even more mistakes that are harder to find. Yet, with functions, you can fix it in one place and you're good to go!

Writing code without functions wastes time.

Without functions, you have to copy and paste the code to reuse it. But, once you have a function, you can keep using it as many times as you want. You can share that code with your friends and let them use it in their programs!

It looks like we're ready for the next step. It's time to answer the questions of the day: "**What in the world is a function?**" (You may need this answer in case someone asks you!)