Efficiency or convenience?

For centuries, the world has been in development and change. Humanity has contributed day by day to this adventure, which it started with stones and sticks. The world in development has changed the direction of its horizon and advanced it with a new invention every day. This development has manifested itself with technology in the last centuries. Undoubtedly, one of the most important of these developments was the invention and advancement of programming languages.

In 1972, a related attack took place at the AT&T Bell lab. The C programming language, which is considered the father of all programming languages, has been published. It was the first type supported programming language ever made. It was prepared so that the kernel of the Unix operating system could be planned. Although many programming languages were published in different disciplines in the following years, no one could take the throne in system programming.

Nearly 20 years after the C programming language was released, another programming language was released that would leave its mark. That language was Python. It took about 10 years to stabilize. Compared to its predecessors, it was a language designed for intelligibility and convenience.

These two programming languages mentioned in the paragraphs above, according to the data on https://www.tiobe.com/tiobe-index/, are in the first 2 places in preference by programmers and are fighting for the first place. The interesting thing here is that although the Python programming language is derived from C, it is fundamentally different in many aspects from a technical point of view. Despite this, both managed to stay on the agenda. This shows us that programming languages with different purposes can remain popular at the same time. They differ from each other, but one is completely better than the other. Unacceptable.

When it comes to benchmarking in programming languages, the first thing to look at is the structure. C is a procedural language. Python, on the other hand, is basically object oriented, but has multiple constructs. There are 4 of them: oop, modular, interactive and interpretive. The Object Oriented programming structure gives us flexibility. Its reusability is higher than the procedural structure. OOP uses modular structure. This allows to reveal a different code by playing with the modules.

C is a low-intermediate programming language. It acts as a bridge between machine language and high-level languages. It uses Compiler for conversion. This system checks the code line by line and if it finds an error, the program closes and gives an error. If it finds no errors, it creates a file for the program to run. This makes C a faster language than Python. Python is a high level programming language. A python code is contained within the machine language and uses the interpreter for the conversion. Interpreter fully controls the python code. It suppresses the first error it encounters. So it makes it necessary to take action one by one to see the error. So Python becomes a slower language than C.

Programming languages have variables. They hold certain values for us. Sometimes these are integers, sometimes they are a group of words. In C, we define them one by one. Other languages based on OOP have both definition and automatic definition methods called 'auto'. Python, on the other hand, uses automatic identification directly. For example, if the value assigned to a variable is a number, it is treated as an integer. It provides definition as char if it is a single letter with quotation marks, and as a string if it is a sentence defined using double quotes.

One of the most important aspects of programming is memory management. In C we do this manually. We have to spend a line of code ourselves to allocate memory and clear it. There is a garbage collector in the object oriented programming structure adopted by Python. It makes these definitions and deletions automatically.

Regarding general memory management, automatic programs, object-oriented programming languages have a very powerful structure to provide this pipe. This is how Python will come to the C language.

There are variables that are used when writing code. They are assigned values as described in previous paragraphs. Since many operations are performed manually in procedural languages such as C, these variables need to be accessed in memory. Therefore, a kind of pointing tool called pointer is used. Since such operations are performed automatically in most object oriented programming languages such as Python, pointers are not needed and therefore the python language does not support it.

Since C is a procedural language, it uses functions as a function unit. Since Python is an object oriented language, it uses objects as a function unit.

As mentioned in the first part, C is a language defined for writing operating system kernels. Therefore, it is widely used in hardware applications. Embedded system programming is also very common. Python is a general purpose language. It tends to range from the simplest operation to important topics such as machine learning. Quantum programming, another topic that has become widespread recently, is also done with python.

C is a basic programming language. It asks you to define the processes yourself. There is hardly any ready-made mold. These ready-made patterns are very common in Python. While you can do the same in C in 10 lines, you can do it with just one word in Python. As it can be understood from here, the syntax structure of C is more difficult than Python. The complexity of the syntax structure also affects complexity. Python is an easier language to learn, but C is much more demanding. However, we cannot do in-line definition in python despite its ease of use. This ease also provides us with C.

Having ready-made molds and ease of use show themselves in many areas as well as in the data structure structure. While we define ourselves in C to use structures such as stack and queue, there are ready-made patterns in Python. It allows data usage and storage without identification.

As in all programming languages, file extensions are also different in accordance with its name. While .c is used in C, we use .py in python.

We said that Python uses an interpreter while using a C compiler. Although the compiler system saves us time, the ease of error solving makes it more practical to use the interpreter. Because when the program sees an error, it stops directly and prints an error. However, when using the compiler, it goes to the end and suppresses all errors at once. We can say that the use of interpreter provides ease of debugging.

A function defined in procedural languages such as C will only have a name and a definition. The compiler recognizes it and moves the debug line to the called function. But in python this is different. It provides an opportunity to redefine functions. It also makes it possible to call the same function with more than one name.

C is not a ready-made language. This makes it do less work in the background. It acts machine-friendly while doing a job. In return for this friendship, the running speed of C code is very high. Python is a practical language, but good for the user, not good for the machine.

C is a basic programming language. Since most common languages are derived from C, at the time of use, C is also a language that can evolve into them. It is open to development. Despite this, C is a language that has not been passed in many areas. One of its most prominent features is that it is a very secure language.

The C programming language does not support the basics of object oriented programming. Therefore, it does not support the polymorphism, which is one of its foundations. Python, on the other hand, supports it because it is a full object oriented language.

The C programming language does not make it appropriate to write code for every system and every environment. Requires gcc compiler. Since the compiler system of each operating system is different, the same c code may not work everywhere, as it is not the same or the same method on every system. But in python the situation is different. It can be run on any operating system and environment as long as the version is the same.

Python programming language has very high memory consumption. Defining ready-made patterns has its price. In C, this consumption is lower.

Python programming language has multithreding structure. This means being able to process with more than one thread. But while doing this, Global Interpreter Lock comes into play. More than one thread while working on a data or a file means confusion. That's why the thread that starts running should lock the doors so that no other thread can enter until it's done. This is a general producer-consumer problem. Python also suffers from the same problem. Therefore, each thread is included in the process in order. This slows down processes.

Even for basic operations in C, we need to define at least a library. There is no such requirement in Python. As we said in the previous sections, C is a functional language. It is also a compiler supported language. So after the compiler has done all the definition, the code starts from the execution main function. An error is received in a non-main project. This is not the case in Python. Without any definition, it can be directly defined and something can be printed on the screen.

One of the most basic structural differences is that in C, a semicolon must be added after each line. There is no such requirement in Python. Tab feature, which is used to increase readability in C, is mandatory in python because it does not use any curly braces or semicolons. This is how it closes. However, when writing python code, getting an error just because a tab is missing is not noticeable. So it is a simple but frustrating type of error that takes a long time.

When printing something to the screen in C, type designation is made with the % sign and at the end, which variable represents it. This is not the case in Python. It is possible to print the screen together by putting a plus sign between the variables.

C programming language is very difficult to use in image-based works such as graphic works. It does not contain graphics and data analysis visualization elements. So python is a much better option in this area.

A person who learns C for the first time learns other languages much more easily because he learns the basics of all operations. However, someone who starts coding with Python has difficulties when they need to use a more difficult language.

As it will be noticed, these two programming languages are very strong languages in their fields. For this reason, a new programming language has been written in recent years, in which the features of the two are blended. This is CPython. It is a highly functional, dynamic and object oriented language. However, since it is written in c, it is also traditional. But basically this language is a bytecode interpreter. It can communicate with many programming languages using its interface.

As a result, both languages are languages that do not fall off the agenda and seem to remain on the agenda. This choice, which changes according to the work done, the preferences of the programmer writing the code and the brain structure, will keep these languages in different procedures alive. Although they seem different, these languages, which are composed of and contain each other, are important building blocks of this journey.

We are renewing and changing every day. We are looking for the new with technology. Before this technology took us above the clouds we saw, we examined two of the best languages of the current conditions. With this review, we have examined almost all of the structural differences. Python 4 has the basic feature, while C has the only feature it doesn't. I respectfully greet these two languages that complement each other's gaps.