What fundamental differences in syntax, performance and suitability exist between C# and Python?

Katerina Svetoslavova Slavova

Fontys University of Applied Science

Class: PCB 10

Student Number: 5363233

Due date: 11 October 2024

Table of Contents

**Table of contents**2

**Introduction3**

**Methods3**

Python - Overview3

C# - Overview3

Python - History3

C# - History4

Syntax4

“Hello world!” syntax**4**

Construction**5**

Performance6

Suitability6

**Conclusion6**

**References7**

## Introduction

The main goal of this research document is to outline the differences between syntax, performance and suitability of two of the most popular programing languages – Python and C#. According to a developer survey held in 2024 by Statista.com[[1]](#statista), Python leads with 23,9% more popularity among programmers than C#, and holds the third place in the rank before JavaScript and HTML/CSS. What makes one language more preferable than the other and how to know which to choose are two extremely important questions for each developer. As the famous American computer scientist once said: “The only way to learn a new programming language is by writing programs in it.”

# Methods

The methods that were used in this project are literature study and expert interviews, their main goal was to examine what is already done and found about this topic. All information was explored in google and google scholar. I gathered it from other researches, analysis documents, blogs and found expert interviews with the creators of both languages. The tools that were used are World and Scribbr.nl.

## Python - Overview

Python is object-oriented programming language, considered to be beginner-friendly and one of the easiest in the world. This language provides its users with the opportunity not to declare the type of the variable explicitly, as it is being checked during the execution of the program. That makes python part of the dynamically typed languages.

## C# - Overview

C# is also object-oriented programming language that is better suited for more experienced developers. It is considered to be part of the statically typed languages, because of the fact that the type of the variable should be specified at the time of its declaration. Due to its syntax’s specifications, it is easier to learn for programmers who have previous experience with languages like C++, C and Java.

## **Python – History**

Python was developed by Guido van Rossum in the Netherlands. He started working on his project in December 1989. It all started as a hobby that he used as an excuse for him to be busy during the Christmas holidays. The main goal of his idea was to provide readability and an easier way for programmers to create their code with fewer lines. With the evolution of Python, there were released different versions as the latest stable one is *3.12.1*. The most popular are the 2.x and 3.x, as there is a great competition between them. Another interesting fact about Python is that it gets its name from the BBC comedy series “Monty Python’s Flying Circus”, because Rossum wanted a short and unique name.

## **C# - History**

C# was developed by Microsoft in 2000. This programming language was created by Anders Hejlsberg in 1999 and was specially designed to run the .Net framework. It is based on Java and C++, but it has its additional extensions. Hejlsberg and his team called their creation “Cool”, which is the abbreviation of “C-like Object Oriented Language”. At first, Microsoft wanted to keep the name, but it was soon changed to C# due to trademark reasons. Microsoft have released many versions, each improving the quality of the previous. The most recent stable version is the C# 12.0, which is also the most popular one.

## **Syntax**

Python’s syntax could be described as readable and simple, while the C#’s one is more complex. The goal of the following examples is to outline the differences in syntax between these two languages.

## “Hello world!” syntax:

Python: C#:



Resource: Miczkowski, K. (13 March 2019). Boldare.com

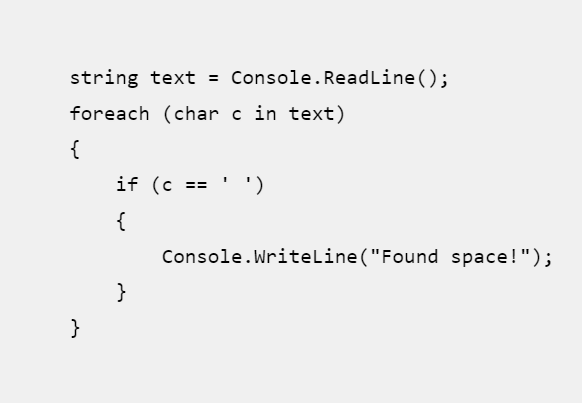
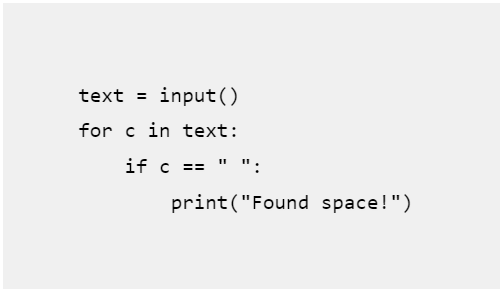
<https://www.boldare.com/blog/python-vs-c-sharp-a-comparison/>

Resource: Miczkowski, K. (13 March 2019). Boldare.com

<https://www.boldare.com/blog/python-vs-c-sharp-a-comparison/>

As you can see C# follows a specific structure that consists of namespace, a class, and main method, in which we start writing our main code. On the other hand, Python does not have any of these attributes. Although, the new versions of C# allow us to minimize the code only to “Console.WriteLine();”, all these attributes are still defined, but by the C# interactive compiler and not by us. Another small, but very important difference is the semicolon at the end of the C# printing method, which is missing in the Python code. There is also a difference in the name of the printing methods. Python’s print() method is short and easy to remember, while the C#’s Console.WriteLine() method is more tricky.

## Construction:

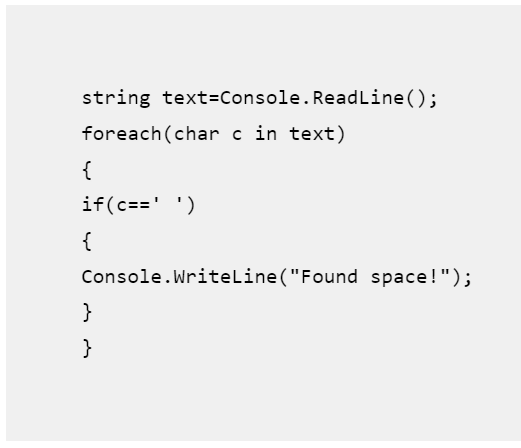
 Python: C#:

Resource: Miczkowski, K. (13 March 2019). Boldare.com

<https://www.boldare.com/blog/python-vs-c-sharp-a-comparison/>

Resource: Miczkowski, K. (13 March 2019). Boldare.com

<https://www.boldare.com/blog/python-vs-c-sharp-a-comparison/>



Resource: Miczkowski, K. (13 March 2019). Boldare.com

<https://www.boldare.com/blog/python-vs-c-sharp-a-comparison/>

As you can see, Python does not use curly brackets for cycles and if – statements. Due to this fact, the spaces after them called whitespaces are very important, because without them the program will not know if the following line should be inside or outside of the cycle and it will not compile. On the other hand, the C# program will work correctly both ways, due to the brackets, but every professional developer knows how important it is to follow the good practices shown in the first picture.

**Performance**

Another important difference, between the two programming languages, is that Python is interpreted language, while C# is a compiled one. Interpreted languages tend to be slower and more inefficient than compiled languages. This is due to the fact that compiled languages are being directly converted into a machine code executed by the processor. On the other hand, languages like Python, are first executed by interpreters and then by the target machine. These interpreters go through the code line by line, which slows down the execution process. Because of this, compiled languages have lower memory usage – interpreters save more information about the program during its runtime.

# Suitability

When it comes to suitability, Python has various tools, libraries and frameworks, such as NumPy, Django, TensorFlow, SciPy, PyTorch and Keras, that make it convenient for machine learning, web development, data analysis and artificial intelligence (AI). In comparison, thanks to its integration in .NET framework, C# gives programmers the opportunity to use the numerous .NET tools, libraries and frameworks such as Xomarin, Entity framework and .NET core. This makes the statically typed language suitable for game development, web development and mobile applications.

# Conclusion

All things considered, both languages have their advantages and disadvantages. The simplicity in its syntax makes Python perfect for beginners, who have never programmed before, but as a member of the interpreted languages its slower and has more memory usage. The variety of tools it possesses, creates the perfect ecosystem for advanced programmers in the machine learning field. Python developers’ salaries run between 33 000 - 72 500 euros per year - the amount depends on the programmer’s skills and abilities, so the more professional you are - the better.

C# has a challenging and tricky syntax, but has so much to offer when it comes to libraries and tools for game development. It is faster and economical, when it comes to memory usage. Once learned, it offers so much opportunities to its users. C# developers can earn from 45 000 up to 75 000 euros per year, as the total depends again on the skills.

# References

*C Sharp (Programming Language). (2024, October 5). Wikipedia. Retrieved October 9, 2024, from https://en.wikipedia.org/wiki/C\_Sharp\_(programming\_language)*

*C Sharp (Programming Language)*. (2024, October 5). Wikipedia. Retrieved October 9, 2024, from https://en.wikipedia.org/wiki/C\_Sharp\_(programming\_language)

Vailshery, L. S. V. (2024, September 18). *Most used programming languages among developers worldwide as of 2024*. Statista. Retrieved October 9, 2024, from https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/

*EP:43 Anders Hejlsberg*. (2023, February 11). The Aarthi and Sriram Show. Retrieved October 9, 2024, from https://www.aarthiandsriram.com/p/our-dream-conversation-anders-hejlsberg

Ray, P. R. (2023, December 2). *In-Depth comparison between C# and Python*. Olibr. Retrieved October 9, 2024, from https://olibr.com/blog/in-depth-comparison-between-c-and-python/

Protasiewicz, J. P. (2024, July 29). *Python vs. C # : A Comprehensive Guide for 2024*. Netguru. https://www.netguru.com/blog/python-vs-c-sharp

*Python Tutorial*. (n.d.). Tuorialspoint. Retrieved October 9, 2024, from https://www.tutorialspoint.com/python/index.htm

*History of Python*. (2024, August 12). Geeksforgeeks. Retrieved October 9, 2024, from https://www.geeksforgeeks.org/history-of-python/

Miczkowski, K. M. (2019, March 13). *Python vs C# - A detailed comparison*. Boldare. Retrieved October 9, 2024, from https://www.boldare.com/blog/python-vs-c-sharp-a-comparison/

Karaci, A. K. (2015, January). *A Performance Comparison Of C# 2013, Delphi Xe6, And Python 3.4 Languages*. Samsun University. Retrieved October 9, 2024, from https://www.researchgate.net/publication/319464630\_A\_Performance\_Comparison\_Of\_C\_2013\_Delphi\_Xe6\_And\_Python\_34\_Languages

Van Rossum, G. R., & Drake, F. D. (2010). *Python Tutorial*. https://github.com/KSlavova/Fontys-University/blob/36fcdb3c7032361baddced98de53ca0c4d8851d1/Professional%20standard/Week6/tutorial.pdf

Guervos, J. J. M. G., Alvarez, I. B. A., Castillo, P. A. C., Romero, G. R., Sanchez, P. G. S., Rivas, V. M. R., Valdez, M. G. V., Aguila, A. H. A., & Roman, M. R. (n.d.). Ranking the Performance of Compiled and Interpreted Languages in Genetic Algorithms. In *Sciterpress*. Retrieved October 9, 2024, from https://www.scitepress.org/Link.aspx?doi=10.5220/0006048101640170