Comparing Python & C#

SDV602

Ezekiel P. Brown
NMIT TePukenga

NMIT SN - 13491119

18/11/22

The Heritage & Philosophy

Python

Python was first released in February of 1991 by the creator Guido van Rossum. Most may believe that it was named after a giant snake but actually, Guido called it after the popular television show Monty Python. Python 2 was released in 2000, incorporating new features such as list comprehensions, garbage collection that detects cycles and references, reference counting, as well as Unicode support, before being discontinued in 2020. This was followed by the release of Python 3 in 2008, a version that has continued to be supported to this day.

Python's design philosophy was explained well by Tulchak and Marchuk (2016). They say, “Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C++ or Java.”

CPython was the original language in which Python was implemented. CPython is the name given to this version of Python in order to distinguish it from other Python implementations which follow. The Python interpreter produces bytecode from your Python code, which is then compiled into bytecode using CPython. A Python language is developed using CPython as the base for the development of the language.

Python supports a number of programming paradigms, including imperative, functional, object-oriented, and procedural programming paradigms. A program written in the imperative paradigm is elegant and simple, and it can be used to manipulate data structures extremely well since computation is carried out by directly making changes to the state of the program itself. Functional paradigms are characterized by the use of each statement as an expression and are free of all forms of states and mutable values. This approach has the main advantage of being able to accommodate parallel processing since there are no states to consider, and this lends itself well to parallel processing. It is a useful style for recursive functions and lambda functions. To implement object-oriented programming, data fields are treated as objects and can be manipulated only by methods prescribed by the object-oriented programming paradigm. Almost everything in Python is an object, which has several properties and methods that can be used to manipulate it. The Python language, however, does not fully support this paradigm, since it cannot implement features like encapsulation and does not force the creation of classes as part of the language. The procedural paradigm is based on a step-by-step iterative approach where tasks that are common to a given task are placed in functions and are called when necessary based on the fact that they are usually repeated. Generally speaking, this style of coding is based on iteration, sequencing, selection, and modularization of code.

A study from Stack Overflow Developer Survey 2022. (2022) shows that from over 71,000 responses that 48% of them use python which equates to around 34,000 responders. 53,000 of those openly admit to being professional developers and 6,000 are learning to code. Stack Overflow ranks Python as the 4th highest programming language only behind JavaScript,

HTML/CSS, and SQL. Python is ranked 3rd for respondents learning to code. It is also worth noting that Python ranks first along with Rust for all developers that are interested in learning these languages. There are still many big companies that make use of Python on a regular basis such as Intel, IBM, Pixar, Netflix, Facebook, Spotify, and Nasa to mention a few.

C#

A team of Microsoft employees, including the designer of C#, Ander Hejlsberg, began working on the language in January 1999. Originally, the language was called "C-like object-oriented language" or Cool, but due to copyright reasons, Microsoft was forced to change the name. During the year 1988, Microsoft had a project called C# that was never completed, so it was repurposed for the Ander Hejlsberg project.

The C# programming language is an object-oriented language that was developed by Microsoft as part of its attempt to combine the computing power of C++ with the programming ease of Visual Basic. In addition to its C++-based foundation, C# contains similar features to Java and is based on the C++ language. Friedman (2021) states that “C# was also created with a goal to be a multi-use, general-purpose programming language; or, in other words, an object-oriented programming language.”

From Stack Overflow Developer Survey 2022. (2022) study, 27% of 71,000 respondents use C# in their day-to-day lives. This equates to around 19,000. C# is still a popular language 20 years later. Stack Overflow ranked them 7th most popular with professional developers. C# is still widely used within the IT industry ranging from web applications to game development. Companies such as Microsoft, Trustpilot and City National Bank still use C#

Platforms

C#

The C# programming language can be executed on several platforms, including Windows, Linux and macOS. With the aid of the Mono project, C# can now run on a variety of platforms. This is because it was originally built to run only on Windows.

There are many useful libraries for C# that can be used to accomplish various tasks. These libraries include AutoMapper, FluentValidation, MediatR, Swashbuckle, Ocelot and Polly, just to mention a few.

Using Swashbuckle, you can generate beautiful API documentation based on Swagger UI, enabling you to explore and test API operations in real time. In a few minutes, you will be able to configure this in your project and it will take only a few minutes of your time.

A common object-to-object mapper library is AutoMapper which has been designed to map the properties of two different objects similarly. As a result, this will assist in reducing developer work and ensure that the mapping of values is performed without any unwanted code.

Python

Python can be run on Linux, macOS and Windows.

Python's standard library consists of over 200 core modules, all of which contribute to its functionality. As a result of the combination of all these factors, Python has become one of the most popular high-level programming languages available today. The Python Standard Library is an essential component of Python, as it is responsible for accessing the functionality of Python for programmers.

Some of these libraries include Matplotlib, TensorFlow, Pandas, Numpy and SciPy just to name a few. These are especially easy to start using. Importing libraries can be easy as typing “import pandas” at the top of your code.

NumPy and Matplotlib are two Python libraries that can be used to visualize data and plot graphical information. It has therefore provided a viable alternative to MATLAB as well as an open-source program. Developers may also use the Matplotlib APIs to embed plots within GUI applications when using the matplotlib APIs.

The Pandas Python package is one of the most widely used open-source Python packages available for data science, data analysis, and machine learning. This library is based on Numpy, which is a package that manages multidimensional arrays and utilizes Numpy to support them. Data processing tasks that involve repetitive, time-consuming processes can be simplified using Pandas.

Integrated Development Environment (IDE)

It is a very common practice for computer programmers to develop their software in integrated development environments. A good IDE will consist of a source code editor, a build automation tool, and a debugger, which are all essential components of any development environment.

There are several good IDEs for Python development, including PyScripter, PyCharm, Spyder, PyDev, Idle and Wing.

For C#, some of the most useful IDEs are Visual Studio Code, SlickEdit, Ellipse aCute, and Notepad++.

Characteristics

C#

Advantages

In the design of C#, object-oriented programming (OOP) has been at the core from the beginning. It is implied in code like this one. This is because you can define the type and structure of the data to be able to apply standard functions to it. A simple way to create, manage, and combine applications is by breaking data into objects, which makes it easier for the user to build, manage, and combine applications. When we talk about OOP, the class declaration describes how an object behaves without having to interact with any of its inner attributes. By programming in OOP languages, you can test your code easier, read your code more easily, and respond to problems more quickly as well as write your code more efficiently.

There is no doubt that C# is considered a high-level programming language since its syntax is similar to human language. For this reason, the code written in C# needs to be compiled, for the hardware to understand its commands since it has a very high level of abstraction from the machine code.

Maintaining the performance of an application is one of the most crucial tasks that must be managed. C# has an inbuilt garbage collector that can be used for this purpose.

A C# developer is often referred to as a .NET developer since the stack of technologies used for programming is similar in most cases. The Stackoverflow Developer Survey 2022 (2022) found that C# is the 7th most popular programming language among professional developers according to the ranking of the most popular programming languages.

Disadvantages

While learning C# on its own may not be a simple process due to its complexity, using .NET libraries adds yet another layer of complexity. There are a lot of libraries that are updated regularly in .NET as well as thousands of resources that you need to know. If you want to develop cross-platform applications, you have to use multiple runtimes as C# or .NET apps are not natively compatible with Linux or macOS, which means you will be forced to work with multiple runtimes if you want to do so.

There is no doubt that C# heavily relies on .NET resources to enable it to run on various operating systems or platforms. Despite this, on its own, the .NET technology stack does not provide that much flexibility if you do not consider it as your primary technology stack.

Python

Advantages

It is easy to use Python even if you have no previous programming experience. In terms of its syntax, it is similar to the English language, which is a high-level programming language. The ease of use and learnability of the language can be attributed to these reasons. In comparison with Java and C, Python uses far fewer lines of code to perform the same task as Java or C does. Of its ease of learning, Python is a much faster language than other languages when it comes to the execution of principles.

The Python programming language is known for its versatility and productivity. As a result of Python's simple nature, developers can pay more attention to the actual problems they encounter when working with it. A user can understand the syntax and behaviour of the programming language more easily since she or he does not have to spend hours on this, so more work is done.

Since the Python language has been around for a long time, it has developed a mature community of programmers that can support the needs of all types of developers, be they beginning developers or professionals. To create a better understanding of the Python programming language, there are plenty of guides, tutorials, and documentation available. This will help the developers to learn the language faster and in a more efficient manner. Compared to other languages, Python has been able to grow rapidly as a result of its supportive community.

Python offers a vast library to its users. Python has a huge and comprehensive standard library, so all of the functions one might need to carry out a given task will be found in these libraries. As a result of an extremely supportive community and the sponsorship of several corporations, it has been able to succeed.

Disadvantages

In comparison to Java or C, Python runs at a slower rate. Python programming language is an interpreted, dynamically typed language. Since the language gets interpreted, each line of code must be explicitly ordered before it can be executed. The execution process is slowed down by this time-consuming process. The dynamic structure of Python also contributes to the slow speed of its execution since when the code is executed, the excess work needs to be done as well during the execution process. Consequently, Python isn't used very often in cases where fast acceleration is required.

Several issues were raised by Python users regarding the language design. In Python, variables can change type at any time as the language is dynamically typed. As a result, it needs more frequent testing, as well as displaying errors during runtime in the language.

The Python programming language consumes a great deal of memory. This is because it is flexible when it comes to data types. As a result, it consumes a lot of memory. When it comes to tasks where the user wants to optimize memory utilization, Python is not the best choice.

References

Friedman, J. (2021, November 13). *When Was C# Created? A Brief History*. C# Station. https://csharp-station.com/when-was-c-sharp-created-a-brief-history/

Stack Overflow Developer Survey 2022. (2022). Stack Overflow. https://survey.stackoverflow.co/2022/

Tulchak, L. V., & Marchuk, A. O. (2016). *History of Python*. http://ir.lib.vntu.edu.ua/bitstream/handle/123456789/10471/461.pdf?sequence=3&isAllowed=y