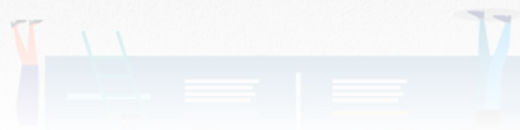


2021

C# vs Python Comparison



C#
Vs
Python



Surafel Zelalem

ETR/8571/10

Computer programming

Assignment 1

C#

C#, pronounced as “C sharp” was developed by Anders Hejlsberg and his team at Microsoft around 2000. It is a general-purpose, multi-paradigm programming language. It shares similarities with C and C++ as it was developed as their alternative. C# became open-source in 2017. It can be easily integrated with the .NET framework of Microsoft. The core syntax of the C# language is similar to that of other C-style languages such as C, C++ and Java, particularly:

- Semicolons are used to denote the end of a statement.
- Curly brackets are used to group statements. Statements are commonly grouped into methods (functions), methods into classes, and classes into namespaces.
- Variables are assigned using an equal's sign, but compared using two consecutive equals signs.
- Square brackets are used with arrays, both to declare them and to get a value at a given index in one of them.

C# is a high-level language, which means it's relatively easy to read and write. This programming language is also statically-typed, which means that errors are detected before the application goes live. This makes it a lot easier to detect small flaws in your stack that would otherwise be almost imperceptible—not to mention incredibly aggravating.

Features of C#

- Multi-paradigm Language: C# supports multiple programming paradigms such as Structured, Imperative, Task-oriented, functional and Object-oriented programming.
- Strong typing: A strongly typed language has strict typing rules. It is checked during compile time which shows errors and exceptions at compile time. C# is a strongly typed language.
- Cross-Platform Support: Using Mono, a free and open-source project helps to develop cross-platform compiler and runtime environment for C# making it cross-platform compatible.
- Standard Library: C# has a rich collection of community class and commercial libraries. Mostly built on the .NET framework to increase the application and functionality.

Advantages of C#

- Reusable Code: As C# predominantly supports the object-oriented approach, it is easier to design large scale software. You can easily create reusable classes and modules to increase the maintainability and modularity of the software.
- Support: As C# is maintained by Microsoft, you can get direct support from Microsoft's developer community.
- CLI (Common Language Infrastructure): It is the technical standard set by Microsoft which describes the executable code and runtime environment. It allows multiple high-level programming languages to be run on different platforms without the support of specific architecture or rewriting.

Like other general-purpose programming languages, C# can be used to create a number of different programs and applications: mobile apps, desktop apps, cloud-based services, websites, enterprise software and games. Lots and lots of games. While C# is remarkably versatile, there are three areas in which it is most commonly used.

- Website development
- Windows applications
- Games

Python

Python was released in 1991 by Guido van Rossum. It is a high-level multipurpose programming language. Python was created around the 1980s. It has been open-source since the beginning and has strong community support. There are numerous frameworks for different purposes based on python. Flask and Django for web development. Tensorflow and Keras for deep learning. Python has huge library support. Python is an interpreted language. Python has been predominantly used in artificial intelligence and machine learning.

Features of Python

- Dynamically Typed: Python is an interpreted language. This allows the dynamic typing feature to come along. It executes programming behaviour during the run time which is commonly performed by statically typed language at the time compilation.
- Code Readability: Python has simpler syntax as compared to other object-oriented programming languages. It has a pseudo-code structure. It looks clean with no semicolons and brackets. It relies on indentation to describe the block of code.
- Garbage Collection: Python has the feature of garbage collection which allows the programmer to get rid-off manual memory management.
- Multiple-programming Paradigm: Programming paradigm is referred to as the execution model of the program depending upon the user requirement. Python supports both functional and object-oriented programming. This gives it the flexibility to be used for various purposes.

Advantages of Python

- Vast Library Support: Python has more than 200 thousand packages for almost all purposes. It makes the developer's work easy by simple importing of modules and direct implementation.
- Productivity: Python offers powerful integration, unit testing frameworks and support on multiple IDEs and active community support. This makes it an ideal choice for developers who are into rapid development.
- Learning Curve: Python is one of the easiest languages to get started. It is one of the most popular choices among code newbies. It has a smooth learning curve, making it easier to become proficient in less time.

Comparison between python and C#

Speed

When referring to speed it means the speed of the programmer not the speed of the of the program itself, python was designed taking the programmer into account, it has a lot of white space and easy readability.

It also has much simpler syntax than `c#`. Also, Python doesn't require you to end every line with a semicolon as C languages do.

C# is familiar. If you know Java or any other C language then learning C# is only a step to the left. While Python operates on many similar structural principles—Like being object-oriented and a high-level language—the syntax is a lot different than C# or other C languages.

Python is dynamically typed, while C# is static. This means that when you call a variable in Python, it generally doesn't matter what it is, the Python will figure it out at runtime. It could be a float, a string, or an integer; they will all print as what they are when the program is run.

Python is also quicker when it comes to running your code. Since Python doesn't have a compile step, it runs immediately—unlike C#, which does have a compile step. While compile steps aren't always long, it can significantly increase the amount of time needed for testing and debugging: you'll have to wait for a compile step every time you try to run the code, even when you are just making small changes.

Readability

C# uses what many languages use to delineate its blocks of code: nested curly braces and brackets. While this kind of code can be made readable, it doesn't have to be. The interpreter doesn't require any indentations. So, you could end up with lines and lines of brackets and braces; like a painfully unreadable coding hellscape.

Python, on the other hand, has whitespace built into its DNA. It uses whitespace to delineate blocks of code. This means that instead of a grassy field of curly braces, you either code with neat indents or your code just won't run. So, while both languages can make for neat, readable, code, Python basically forces it. Whether Python is more attractive without the curly brace fringe is up to the reader, but many prefer it over C#.

Performance

When it comes to performance there is a clear distinction between C# and Python. C# is a compiled language and Python is an interpreted one. Python's speed depends heavily on its interpreter; with the main ones being CPython and PyPy. Regardless, C# is much faster in most cases.

For some applications, it can be up to 44 times faster than Python. This is for a number of reasons—from Python's garbage collector to its dictionary lookups. It's also partly due to C# being a compiled language: it takes a bit more work to write but runs more efficiently because of it.

Application

C# and Python are both general-purpose languages. They can be used for anything from game development to machine learning. But that doesn't make them equal on all fronts. Machine learning might be the best example of how they differ.

Python is machine learning royalty. Python has an enormous number of software choices for a machine learning programmer: Numpy, SciPy, TensorFlow, PyTorch, Apache Spark, Keras, and more. Because of this gold standing in machine learning, Python also has a wealth of videos, tutorials, and tried and true examples of function.

When it comes to machine learning in C#, you get ML.NET, and some older libraries like Accord.Net, and bindings for TensorFlow. Microsoft's Cognitive Toolkit (CNTK) has support for both C# and Python, but it's Python API is much more polished. There just isn't a lot available for C# programmers when it comes to this field. Popularity has its perks, and being a more popular language for machine learning brings Python more support for that field.

License

Python is fully open-source. It sits under the GPL license and it's available to everyone to use and contribute too. In the same vein, most of its accouterments are also open-source; from packages to IDEs.

C# is also (mostly) open-source, but this is a fairly new development. Microsoft has been making C#'s source available little by little to the public. However, this doesn't mean everything C# touches is open-source or free like Python. For example, the official IDE for C languages is Visual Studio, which has a free version for individuals and small companies but isn't open-source. While there are open-source alternative IDEs for C#, the average workflow for C# developers will be using Visual Studio. In this sense, C# can be more expensive than Python, even if you could potentially use it for free.

Comparison Table: C# Vs Python

Python	C#
Developed in the 1980s as open-source distribution.	Developed by Microsoft around 2000 as Microsoft proprietary. Recently open-sourced in 2017.
Python is used as multi-paradigm programming. It is equally preferred for functional as well as an object-oriented paradigm.	C# supports multiparadigm programming but is majorly used for object-oriented programming.
Python is framework independent. It can be easily used with CLI (command line interface), IDEs, web frameworks and integrate with Java, JavaScript and C++.	C# essentially uses .Net framework to run. It is inter operational with F#, JavaScript, Python and VB.NET.
It is dynamically typed. Errors and exceptions show up during run time.	It is statically typed. Errors and exceptions show up during compile time.
It requires an interpreter.	No requirement of an interpreter.
The development lifecycle is faster but performance-wise there is lag.	With the use of CLI, C# proves to be fast and has better performance.
Multi-threading requires multiple processes.	Multi-threading is easier in the .Net framework
Syntactically, Python has clean code which makes debugging easier.	C# has the typical use of braces and semi-colon making it slightly tough to debug.
Python has a vast variety of library support for almost all applications. Some of them are Apache spark, Pytorch, Sci-kit Learn and many more.	C# also has libraries to support development but limited when compared to python. Some of them are ML.net, Accord.net.

Conclusion

C# has a more clear and organized structure. Like an OOP language, there are no inconsistencies in the formatting rules or syntax. Because of this, C# is just a bit slower to learn and to code. C# can also do almost anything that Python is able to do, and C# is much faster at runtime partly because of all the extra effort you put into it.

Python is easy to learn (certainly easier to learn than C#) and easy to write. Python is so good at getting you into coding with good practices, that it's usually recommended as the first language you learn (and it doesn't have the phalanx of curly braces as we see in C#). Python also has extensive standard libraries and is easier to use for machine learning.

In the end, both C# and Python are excellent languages, and picking one over the other isn't picking wrong. Both languages are free, they both have mature tooling, active communities, and a number of frameworks and libraries. They both have extensive applications in many programming fields. They both have their own strengths and weaknesses when it comes to user input, programming speed, and runtime performance.