

Python vs. C#

Python is an open-source, general-purpose programming language that powers websites such as YouTube, Reddit, Pinterest, and Instagram. Python language was one of the first technology stacks Google used to develop its search engine with. It's one of the most popular programming languages today, which finds excellent use in recent development trends such as machine learning.

C# (C-Sharp) is a programming language developed by Microsoft that runs on the .NET Framework. C# is used to develop web apps, desktop apps, mobile apps, games and much more.

Ease of use

One glance at the TIOBE index is enough to see that Python and C# go hand in hand as far as the popularity among developers is concerned. Developers like C# because of its design and origin in the object-oriented programming paradigm.

Those who've had experience with Java can pick it up quickly – and it's a smart choice if they want to develop applications for Microsoft platform because C# is deeply integrated with the .NET framework.

But there are some things developers can get done much quicker with Python, mostly because it's a dynamically typed language.

The basic syntax of Python requires the knowledge of fewer language constructions, while in C#, you just can't write a program without knowing what compilation, assemblies, namespaces, classes, and methods are (and many more!).

It's easier to learn Python iteratively. Developers can use an increasing number of features as they dive deeper into the language.

In C#, that's not an option. Developers need to learn way more before they write their first line of code.

Consequently, a project in C# will require hiring developers who are already highly proficient in C#. Python developers can keep on learning new things and run quick tests on the go.

Dynamic vs. static

Python is a more dynamic language than C#. But what does that imply for your project?

When it comes to dynamic languages, the development process is relatively fast and easy. That's why they require the expertise of a team leader, who will oversee the process to make sure that developers build a robust and scalable application.

Since C# is a static language, it includes a build/compile step, which some developers aren't fond of. The build process adds a step to the entire web development process and impacts its productivity. But the compiler also detects syntax errors before they become a problem.

Performance

- C# may slightly beat Python in performance, but since it doesn't run on open-source technologies, it takes away these gains in the cost of infrastructure software.
- C# might take fewer server resources for the same volume of concurrent users than Python, but you'll be paying Microsoft for these servers anyway.
- Developers can improve Python's performance with PyPy's implementation of the JIT compiler. Writing and deploying code in C# is slower than in Python – the latter is known for its rapid development process, which it owes to its elegant syntax, dynamic nature, and the availability of libraries of pre-written code for easy reuse.
- All in all, developers are more productive in Python's environment.

Wrap up: Python or C#?

If your project depends on high speed and performance, Python will be the optimal choice. It's easy to learn, it offers options for smooth cross-platform development, and provides developers with a wealth of useful open-source libraries that speed up the development process.

However, if your project requires excellent Microsoft integration, top performance, and reliance on a standard syntax and libraries, C# is a better choice.

Prepared By : Abel Seyoum

ID No : ETR./4211/08