PYTHON VS RUST: CHOOSING YOUR LANGUAGE

Python is one of the most ubiquitous and popular programming languages nowadays. It can power anything from simple scripts and web applications to data analytics. On the other hand, Rust is an up-and-coming programming language quickly gaining popularity in the tech community.

Both these programming languages offer their own distinct advantages and disadvantages. In this article, let's compare how each language stacks up against the other and find the preferable language for specific development needs.

What is Python?

Python was first introduced in 1991 by Guido van Rossum. It is a multiparadigm programming language designed to be easily extensible and help users work efficiently.

Python eliminates staples of other programming languages like semicolons and curly brackets while providing a simple programming experience with a simple syntax that increases code readability. Python is considered a more beginner-friendly language due to its simplicity.

Python's extensibility and versatility allow using it across many domains, from system administration and application development to data analytics, machine learning, and artificial intelligence development.

(See why Python is perfect for big data.)

Advantages of Python

- **Python has a relatively smaller learning curve compared to other languages.** It can provide a simpler development experience without compromising functionality. The asynchronous coding style allows developers to easily handle complex coding requirements.
- **A massive collection of libraries and frameworks is available.** Python has gained an impressive number of libraries and frameworks due to its maturity and popularity. As a developer, there is a high chance that you can find a library or framework for any kind of functionality.
- **Python integrates with a wide variety of software,** including enterprise applications and databases. It can be easily integrated with other languages like PHP and .NET.

Disadvantages of Python

- Python is slower compared to compiled options such as C++ and Java since it is an interpreted language.
- While Python is easy to debug, some errors won't be shown until runtime.

(Read our comparisons of Python to Java & Go.)

What is Rust?

Rust is a multiparadigm general-purpose programming language introduced by Graydon Hoare from Mozilla Research. Rust is focused on safety, stability, and performance. It is a statically typed programming language with a memory-efficient architecture and is C/C++ compliant.

Even though Rust is a newer language compared to Python, it has quickly gained popularity within the developer community and is the most loved technology, according to the 2021 StackOverflow developer survey. Rust can also be used in many different domains such as:

- System developments
- Web applications
- Embedded systems
- Blockchain
- Game engines

Advantages of Rust

- Rust is performance-oriented compared to other languages with its fast and memory-efficient architecture with no runtime or garbage collection.
- Enforces strict safe memory allocations and secure coding practices.
- Direct safe control over low-level resources. (Comparable to C/C++)

Disadvantages of Rust

- Relatively higher learning curve compared to languages like Python. A higher degree of coding knowledge is required to use Rust efficiently.

- Low level of monkey patching support.
- The compiler can be slow compared to other languages.

Comparing Python vs Rust

Since we have a basic overview of Python and Rust now, let's compare them to understand how they stack up against each other.

Ease of coding

Python is inherently designed to provide a simpler development experience. Its highly readable code structure and simple syntax provide developers with a better coding experience. Additionally, users can easily adapt Python to any need and start development quickly as it can be used for many use cases.

Meanwhile, Rust is geared more towards system programming and better suited for specific use-cases. Rust will have a steeper learning curve to utilize its features properly.

Performance

As an interpreted language, Python is slower, even with options like CPython geared towards speed. Rust is faster and can be more than twice as fast as Python. Since Rust is compiled directly into machine code, there is no interpreter or virtual machine between the code and the hardware. Another factor that improves the performance of Rust is its memory management.

Even without a garbage collector like in Python, Rust ensures proper memory management from the get-go by enforcing checks for memory leaks and irregular memory behaviors. Overall, Rust has comparable performance to languages such as C and C++ without overheads.

Garbage collection

Rust provides developers the option to store data on the stack or the heap. This feature can be used during the program compilation to determine when the memory is no longer required and should be cleaned.

Moreover, this option clears out data without the need for the program to decide on allocating and cleaning memory. Rust can be easily integrated with other languages without adversely affecting them as it eliminates the need to run a garbage collector constantly.

Python utilizes a garbage collector to check for memory that is not in use and cleans up that unnecessary memory while the program is running.

Documentation

Both these programming languages have excellent documentation by official sources as well as community-provided sources. Python documentation is more beginner-friendly, and even community contributions are clearly defined in an easily understandable manner.

Rust also has simple-to-understand documentation. Yet it's a bit geared towards more technically experienced users and can be a little complex compared to Python.

Extensibility

Python offers a clear advantage in terms of extensibility due to the sheer number of libraries, frameworks, software, and services that are available for Python or support Python.

While Rust is a relatively new language, it has a rapidly growing ecosystem due to its popularity. However, it is not comparable to the options available with Python.

Error handling

How Python and Rust handle errors is entirely different. Python will throw an exception when an error is encountered. Rust will return a value when an error is found, while Python will simply throw an error without providing any suggestions on how to fix it. Meanwhile, Rust will provide some recommendations to easily pinpoint and fix the issues.

Rust will provide an improved development experience and a better and easier debugging experience than other compiled languages due to its features like:

- Guaranteed memory safety
- Reliability and consistency
- Comparative user-friendliness

Unlike Python, Rust prevents the need for users to wait until runtime to determine some errors.

Security

Rust emphasizes security, and the guaranteed memory is what separates Rust from other similar languages like C or C++. Rust is completely memory safe unless explicitly specified by the developer. According to Secure Rust Guidelines, the compiler tracks how many variables refer to given data and enforce a set of rules to manage and secure memory at any point for a Rust program.

In contrast, Python requires developers to configure the memory management and prevent any memory leaks by themselves.

Community

As both languages are open-source projects, the community is directly involved with the development and improvement of them. Python has a considerably larger community as the more mature platform. You can easily find resources for any kind of Python development need, and you can find guides or answers to most problems with a quick google search.

Rust also has a small yet friendly and highly active community. However, you might need to spend a bit more time finding resources for your exact needs.

Choosing between Python & Rust

Both these languages have their unique approaches to development, with each language excelling in one aspect or another. What to choose depends on the specific use case of the developer.

In general, Python:

- Will be the easiest language to learn, if you are starting up
- Also provides a simpler development experience

Meanwhile, Rust:

- Can have a higher barrier to entry with its complex feature set that will be daunting to new developers.
- May be an ideal candidate if you want to expand your skillset and learn a second language.

The versatility and extendability of Python are still unmatched. The ability to use Python across many disciplines from web and backend developments, DevOps (Scripting), scientific computing, enterprise applications to machine learning has contributed to the immense popularity of the language. When this versatility is coupled with the user-friendliness of the language, Python is one of the most sought-after languages.

(Learn about Python tools for beginners.)

However, Rust will be the more attractive option if speed and safety are your primary considerations. Its memory-safe nature and speed make Rust the ideal language for tasks like system development, embedded integrations, game engine development, file systems, and VR. Rust provides a programming language with modern sensibilities while facilitating comparable speeds to languages like C/C++.

Rust is quickly gaining popularity between the wider technological community as well as organizations like Mozilla, Dropbox, and Cloudflare. AWS has even provided its SDK for Rust.

Both support powerful programming

Both these languages are powerful programming tools. However, as mentioned previously, the best language depends on the specific requirement of the user. Python will continue to be a popular language with its maturity, widespread adoption, and ease of use. Yet, Rust is also quickly gaining inroads to the developer community with its superior speed and secure architecture.

Related reading

- BMC DevOps Blog
- The Lua Programming Language Beginner's Guide
- The Spring Framework Beginner's Guide: Features, Architecture & Getting Started
- Chaos Monkey: A Beginner's Guide
- API/Developer Portals: How To Create Great API Portals
- Top Conferences for Programming & Software Development