

Jake Gendreau
Jan 20, 2024
CS 210 Programming Languages
Travis DeVault

BRIEF OVERVIEW AND HISTORY:

The Rust Programming Language was developed by Graydon Hoare in 2006, who, after having to take the stairs up to his apartment in British Columbia due to an error in the elevator's code, decided that he could make something better. Knowing that most C++ applications suffered from some sort of memory vulnerability, he set out to make a programming language that was completely memory safe, but also fast, much like C and C++. He chose the name Rust after the fungus, which is known for being incredibly durable and hardy. It started out as a side project of his, but as it was developed, Mozilla, which Hoare worked for, picked up his project and dedicated a small team and funding to it. After 10 years in the hands of Mozilla, some layoffs lead to concerns about the future of the language. It was decentralized and put into the hands of the Rust foundation, where it continues to live today.

The objective of rust is to be a fast, reliable general purpose programming language. It is meant to be used where C and C++ are used, so it has many of the same concerns.

STRENGTHS, WEAKNESSES, APPLICATIONS:

The benefits that Rust has are numerous, but there are three biggest benefits. The first is that it is memory safe. Thanks to its mutability properties, and the properties of ownership, borrowing, and automatic bounds checking, it eliminates most of the problems that plague the C and C++ world. Another benefit is that it is widely supported. Firefox, Amazon, and Google all use Rust somewhere in their companies, and as a result, it has an abundance of support not only from these companies, but also from the developers that work at these companies. Lastly, Rust has a very high runtime performance, and is able to compete with C and C++ without the safety concerns.

Rust is not without its weaknesses though, one of which is that it has a steep learning curve. I attempted to learn Rust for a 240 assignment last semester, and I just couldn't figure it out in time to finish the project. This issue is coming from someone who has already learned 5 or so programming languages and has been programming seriously for more than 5 years. It is different to C and C++ in all the ways that make it very frustrating to learn coming from those languages. Another issue that it has is less library support, which is a side effect of it being publicly released for only about 10 years. Another weakness that comes as a result of this is that there's not a lot of jobs out there yet for Rust developers. It's a bit of a chicken and egg situation there. Another weakness is its compile time. The Rust compiler is very good at error checking and giving descriptive errors, but that comes at the cost of taking much longer to compile when compared to gcc or clang.

LANGUAGE STANDARD AND COMMITTEE:

The Rust programming language is maintained by the Rust Foundation, which is a non-profit, decentralized group that was founded by AWS, Huawei, Google, Microsoft, and Mozilla.

Rust is not maintained by ISO or ANSI, but is instead maintained by the Rust Foundation. Specifically, it's the Rust Programming Language team within the Rust Foundation. This team produces resources such as the Rust Reference and the Rustonomicon. The official documentation and standard for Rust can be found here <https://doc.rust-lang.org/reference/>.

PROJECTS CURRENTLY USING THE LANGUAGE:

The applications of Rust, like I said earlier, are essentially identical to that of C. It's meant to be fast and reliable. That being said, it does take up more space than C, so it can't be used in exactly the same situations. System controllers, such as elevator controllers, are the target devices for Rust to run on. One notable part of Rust is that it is one of the few languages that are supported natively by the Linux Kernel. Firefox is also partially written in Rust. Many tools that Rust uses, such as its package manager and compiler are written in rust, and Redox OS is an operating system that is written in Rust.

INITIAL IMPRESSIONS OF LANGUAGE AND DOCUMENTATION:

The docs for Rust - specifically Rust By Example - weren't perfect, but they are certainly much better than C. I did have to use other resources to figure out how to install Rust on my computer, but after I did that, it was fairly smooth sailing. Its very first demo program is a Hello, World program, and it covered many basic elements in a good way. However, input was never covered, at least simply, in Rust By Example. When I looked it up in the docs, I only found mentions of input to functions and input in unsafe blocks. Neither of which are what I needed for my program. Eventually, I found a thread on [stack_overflow](#) that solved my issue fairly simply. I really liked the String class. I'm not used to OO things, so being able to do `string.is_empty()` when checking for empty input was super useful!

I'm not a huge fan of the mutability and ownership properties, seeing as I'm very C-oriented. I'm sure that as I get used to it and as I use Rust more, I'll come to appreciate those properties, but for now they seem clunky, and in the way. Overall though, it seems like a very powerful and very stable language.