**Setting Up a Rust Development Environment on Linux**

[[@imthemrluiz](https://devopsvision.medium.com/?source=post_page-----757cc78f536c--------------------------------)](https://devopsvision.medium.com/?source=post_page-----757cc78f536c--------------------------------)

[@imthemrluiz](https://devopsvision.medium.com/?source=post_page-----757cc78f536c--------------------------------)

·

Follow

5 min read

·

3 days ago

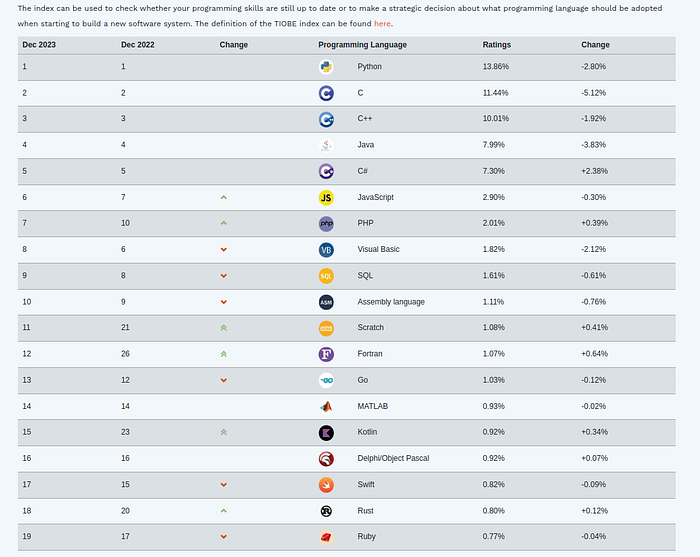
7

Getting a Ubuntu environment ready to start playing with Rust



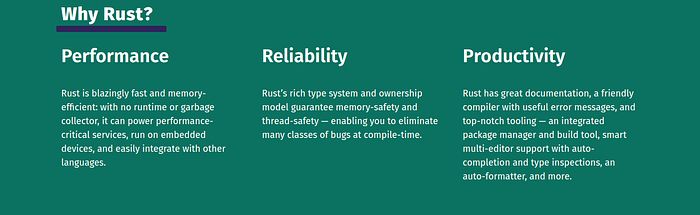
**Why learn rust?**

Simple and easy to undestand, I have worked in DevOps, SysAdmin, and SRE for a long time. During this journey in many cases, we need to use some programming language to create something, from a chatbot to a script to send some notification, pipeline, or whatever. Rust and Go are nice languages, I’ll not join in terms of comparing one language with another but, Rust got my attention for its big adoption in the past few years.



<https://www.tiobe.com/tiobe-index/>

The idea is not to get the master in rust but to understand its main propose and functions. Be able to build some sample projects and play for fun with the language.



<https://www.rust-lang.org/>

**What is Rust commonly used for?**

Thanks to its direct access to both hardware and memory, Rust is well-suited for embedded systems and bare-metal development. And since it’s a general-purpose language, it can also be used for a variety of applications.

**Rust Usage**

*Over 90% of survey respondents identified as Rust users, and of those using Rust, 47% do so on a daily basis — an increase of 4% from the previous year. 30% of Rust user respondents can write simple programs in Rust, 27% can write production-ready code, and 42% consider themselves productive using Rust.*

**What is Rust?**

Rust is a **systems programming language** that is perhaps most notable for being **memory-safe without garbage collection**. It was designed to solve problems observed maintaining the millions of lines of C++ in the Firefox web browser. It is fast, reliable, and runs on almost anything, and its enthusiasts are exceptionally enthusiastic about its potential.

On December 29, 2016, Steve Klabnik, author of [**The Rust Book**](http://doc.rust-lang.org/book/) and member of the Rust core team, blogged, [**“Rust is more than safety”**](http://words.steveklabnik.com/rust-is-more-than-safety), saying that memory safety alone is not convincing for some prospective Rust users. This catalyzed a blogsplosion the likes of which Rustaceans had never seen (and Rustaceans love blogging), where Rust’s creators and users offered their opinions about what makes Rust so exciting.

**Getting the Linux Environment Ready to Run Rust**

The first step is to install Rust. We’ll download Rust through rustup, a command line tool for managing Rust versions and associated tools.

If you’re using Linux or another Unix-like OS. To download Rustup and install Rust, run the following in your terminal, then follow the on-screen instructions. The following steps install the latest stable version of the Rust compiler.

curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh

Once you type the above command and make the default installation, it will start downloading the components, after that, it's ready to go and move on.

Rust is installed now. Great!  
  
To get started you may need to restart your current shell.  
This would reload your PATH environment variable to include  
Cargo's bin directory ($HOME/.cargo/bin).  
  
To configure your current shell, run:  
source "$HOME/.cargo/env"

Then as required we can run:

source "$HOME/.cargo/env"

We will also need a *linker*, which is a program that Rust uses to join its compiled outputs into one file. It is likely you already have one. If you get linker errors, you should install a C compiler, which will typically include a linker. A C compiler is also useful because some common Rust packages depend on C code and will need a C compiler.

Linux users should generally install GCC or Clang, according to their distribution’s documentation. For example, if we use Ubuntu, we can install the build-essential package.

$ sudo apt-get install build-essential  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
build-essential is already the newest version (12.9ubuntu3).

**Troubleshooting**

To check whether we have Rust installed correctly, open a shell and enter this line:

~$ rustc --version  
rustc 1.74.1 (a28077b28 2023-12-04)

If you see this information, you have installed Rust successfully! If you don’t see this information, check that Rust is in your %PATH% system variable as follows.

$ echo $PATH  
/home/luiz/.cargo/bin:............

**Updating and Uninstalling**

Once Rust is installed via rustup, updating to a newly released version is easy. From your shell, run the following update script:

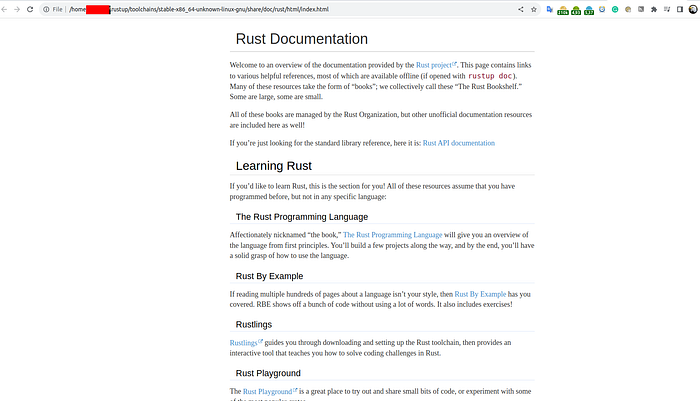
$ rustup update

To uninstall Rust and rustup, run the following uninstall script from your shell:

$ rustup self uninstall

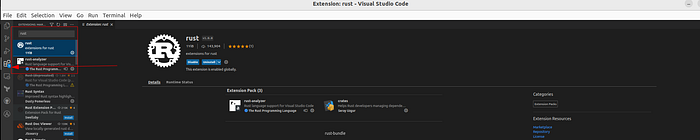
**Local Documentation**

The installation of Rust also includes a local copy of the documentation so that you can read it offline. Run rustup doc to open the local documentation in your browser.



Any time a type or function is provided by the standard library and you’re not sure what it does or how to use it, use the application programming interface (API) documentation to find out!

After that, on your IDE you can also install some plugins to help you code. In the image below, I’m installing the rust plugin on my VScode.



**Writing and Running a Rust Program**

Next, make a new source file and call it *main.rs*. Rust files always end with the *.rs* extension. If you’re using more than one word in your filename, the convention is to use an underscore to separate them. For example, use *hello\_world.rs* rather than *helloworld.rs.*

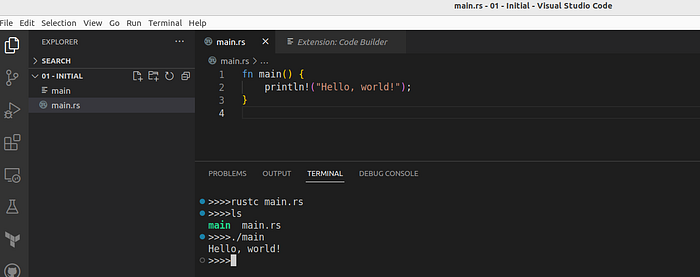
***main.rs***

fn main() {  
 println!("Hello, world!");  
}

Save the file and go back to your terminal.

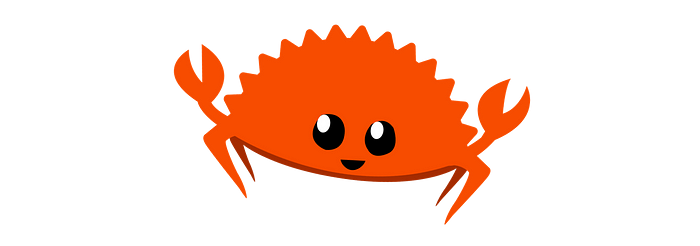
Enter the following commands to compile and run the file:

$ rustc main.rs  
$ ./main  
Hello, world!



And that’s it, this was the first Hello, world in rust! :). Step by step and practicing we get better.

That’s all for now. Let's continue to move on, creating some projects to get practice with Rust!



**Rust Programs Every Linux User Should Know About**

**Further reading**

* [Rust market overview: reasons to adopt Rust, Rust use cases, and hiring opportunities](https://yalantis.com/blog/rust-market-overview/)
* [Rust by the Numbers: The Rust Programming Language](https://thenewstack.io/rust-by-the-numbers-the-rust-programming-language-in-2021/)
* [The Rust Programming Language in the words of its practitioners.](https://brson.github.io/fireflowers/)
* [50 Project ideas in rust](https://www.youtube.com/watch?v=FPdqoz8c5y4&list=PL5dTjWUk_cPYuhHm9_QImW7_u4lr5d6zO&index=25&ab_channel=AkhilSharma)
* [Rust Installation Official Documentation](https://www.rust-lang.org/tools/install)
* See [“Other Installation Methods”](https://forge.rust-lang.org/infra/other-installation-methods.html) if you are on Windows.

**Book**

* [The Rust Programming Language](https://www.oreilly.com/library/view/the-rust-programming/9781098156817/)
* [Rust in Action](https://www.oreilly.com/library/view/rust-in-action/9781617294556/)