

Why the Rust programming language will be the future of programming

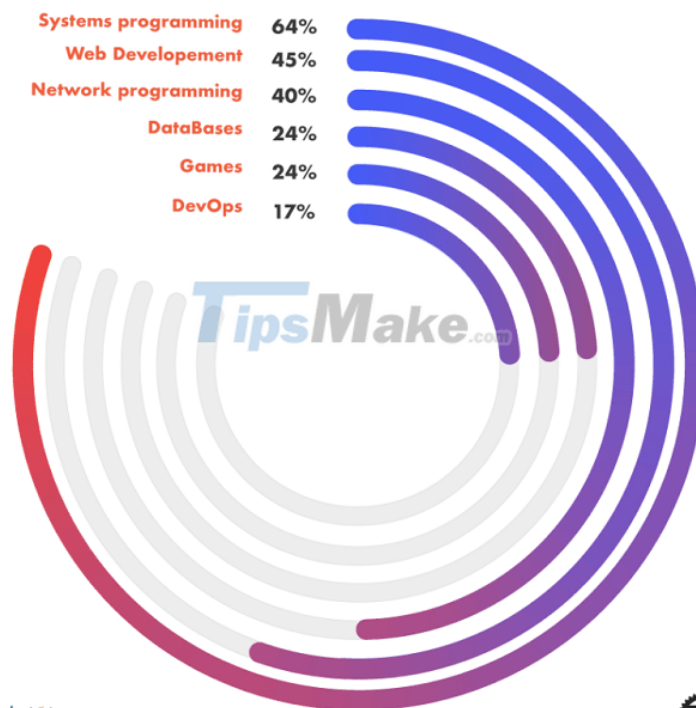
What is the Rust programming language? And why will the Rust programming language be the future of programming? Learn about the Rust . programming language

If it is said that a programming language will develop strongly in the future after Python, the name that is mentioned a lot today is Rust.

Rust is a programming language that is considered new, advanced and will be the future of programming. If you do not know, in today's article, we will learn a little bit about this new programming language.

#first. What is the Rust programming language?

What kind of projects do you develop in Rust ?



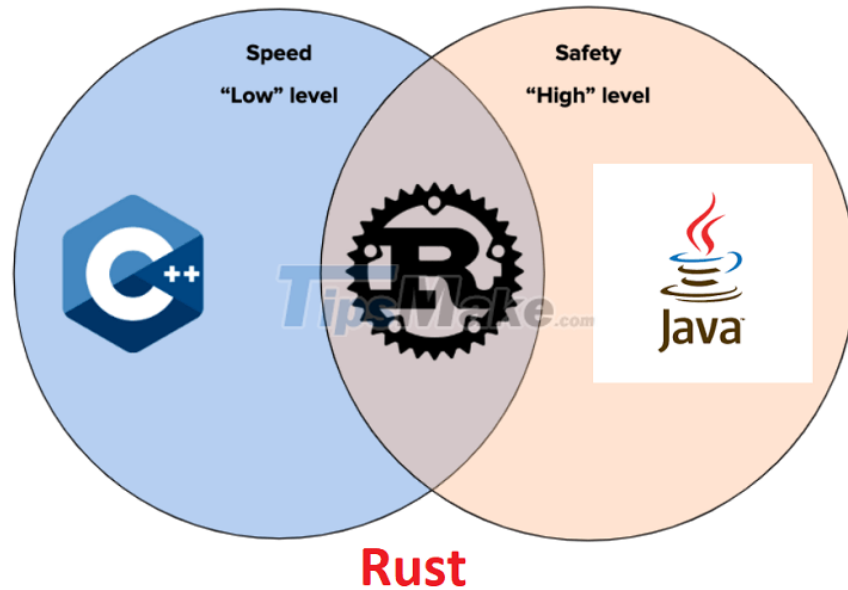
Rust is a low-level, statically typed, and multi-paradigm programming language. It focuses on writing safe and very well designed code that simplifies development.

The Rust programming language was developed to solve problems that C/C++ programming languages have struggled with for a long time, such as memory errors and building concurrent programs for example. .

Rust has been developed since 2006 by programmers at Mozilla Research (Rust's father is Graydon Hoare), but it was not until 2010 that it was officially known and used by the public.

Although it was originally developed by Mozilla programmers, later on Rust was mainly developed by the open source community. It is considered a blend between ensuring the speed of execution and safety on the system.

#2. Pros and cons of Rust



Rust's advantages:

If with other programming languages, controlling memory when processing instructions is extremely complicated and potentially risky, with Rust, everything is much easier.

Rust has specialized functions and tools for managing memory during allocation. Especially in system programming to minimize memory errors when executing instructions.

Rust is a cross-platform language, it can run on all popular operating systems today and of course it can also make applications that run on those operating systems. In particular, with the ability to speed as well as good memory management, it is very suitable for embedded applications.

In addition, unlike languages like C, Rust is considered optimal and supports parallel applications. In other words, it supports more multi-core CPUs, most applications today use 2 to 3 cores mainly.

Disadvantages of Rust:

```
1 use std::collections::HashMap;
2 use std::net::{IpAddr, Ipv4Addr, SocketAddr};
3 use std::env;
4 use warp::{
5     http::{Response},
6     Filter,
7 };
8
9 #[tokio::main]
10 async fn main() {
11     let example1 = warp::get()
12         .and(warp::path("api"))
13         .and(warp::path("HttpExample"))
14         .and(warp::query::<HashMap<String, String>>())
15         .map(|p: HashMap<String, String>| match p.get("name") {
16             Some(name) => Response::builder().body(format!("Hello, {}. This is a response from the Rust programming language.", name));
17             None => Response::builder().body("404 Not Found");
18         });
19     warp::serve(example1).run(([127, 0, 0, 1], 3030));
20 }
```

```
antchu-macbook:20201123-rust-functions antchu$ cargo build --release
Finished release [optimized] target(s) in 0.43s
antchu-macbook:20201123-rust-functions antchu$ cp target/release/handler .
antchu-macbook:20201123-rust-functions antchu$
```

Rust is developed by the open source community so it is always updated regularly, but sometimes this is what hinders their use. You must regularly monitor changes, update to be able to keep up with different changes.

Although powerful, but Rust is considered a programming language that is quite difficult to access, because they were born to interact deeply with the system but are still under control, the way the commands are quite complicated for humans.

In essence, Rust was born to program on complex and high-speed systems.

#3. Rust's application?



With the Rust programming language, you can do a lot of things such as: making web, making PC applications .

In addition, with its capabilities, Rust is also used in high-speed information processing equipment such as vehicle control systems, signal systems .

But currently, Rust is still being used the most, especially in large systems, systems that require high complexity, as well as have to process many parallel data sources continuously at the same time.

In the future, Rust may be developed on ARM-based systems thanks to the advantages of memory management and execution speed.

These are all things that mobile devices desperately need because of hardware size limitations. It is even making some improvements and testing it in AI development, which needs to be as fast as possible.

```
1  #[prelude_import]
2  use std::prelude::v1::*;
3  #[macro_use]
4  extern crate std;
5  fn main() {
6      {
7          let _result =
8              match ::std::iter::IntoIterator::into_iter(::std::ops::Range { start: 1, end: 3 }) {
9                  mut iter => loop {
10                     let mut __next;
11                     match ::std::iter::Iterator::next(&mut iter) {
12                         ::std::option::Option::Some(val) => __next = val,
13                         ::std::option::Option::None => break,
14                     }
15                     let n = __next;
16                     {
17                         // Do something with n
18                     }
19                 },
20             };
21         _result
22     }
23 }
```

In addition, Rust is also used to write:

1. Compiler
2. Real-time application
3. Large and complex systems
4. Embedded system
5. Systems need high performance and low latency. For example in driving devices, in games, or broadcasting devices.
6. Distributed and parallel applications
7. Web Frameworks....

Any programming language is born with its own purpose, no programming language is perfect and so is Rust!

Usually, programmers will use the ability to think programming to limit the shortcomings of the programming language they use, or in other words, optimize the execution of the instruction. There will even be separate tools to support this.

With its fast speed, as well as excellent memory management, I believe that in the future the Rust programming language will go even further, more powerfully, and it will not stop at system programming. Simple.

This language deserves to be a programming language to interest you for your future application development career. Hopefully, the article has brought you the most basic information about Rust. Goodbye and see you in the next posts!

You finished reading the article "**Why the Rust programming language will be the future of programming**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.