

Rust is a general-purpose programming language emphasizing performance, type safety, and concurrency. It enforces memory safety, meaning that all references point to valid memory. It does so without a conventional garbage collector; instead, memory safety errors and data races are prevented by the "borrow checker", which tracks the object lifetime of references at compile time.

Rust does not enforce a programming paradigm, but was influenced by ideas from functional programming, including immutability, higher-order functions, algebraic data types, and pattern matching. It also supports object-oriented programming via structs, enums, traits, and methods. It is popular for systems programming.[13][14][15]

Software developer Graydon Hoare created Rust as a personal project while working at Mozilla Research in 2006. Mozilla officially sponsored the project in 2009. In the years following the first stable release in May 2015, Rust was adopted by companies including Amazon, Discord, Dropbox, Google (Alphabet), Meta, and Microsoft. In December 2022, it became the first language other than C and assembly to be supported in the development of the Linux kernel.

Rust has been noted for its rapid adoption, and has been studied in programming language theory research.

## History

### Early years (2006–2009)

Mozilla Foundation headquarters in Mountain View, California

Rust began as a personal project in 2006 by Mozilla employee Graydon Hoare.[16] Hoare has stated that Rust was named for the group of fungi that are "over-engineered for survival".[16] During the time period between 2006 and 2009, Rust was not publicized to others at Mozilla and was written in Hoare's free time;[17]: 7:50 Hoare began speaking about the language around 2009 after a small group at Mozilla became interested in the project.[18] Hoare emphasized prioritizing good ideas from old languages over new development, citing languages including CLU (1974), BETA (1975), Mesa (1977), NIL (1981), Erlang (1987), Newsqueak (1988), Napier (1988), Hermes (1990), Sather (1990), Alef (1992), and Limbo (1996) as influences, stating "many older languages [are] better than new ones", and describing the language as "technology from the past come to save the future from itself." [17]: 8:17 [18] Early Rust developer Manish Goregaokar similarly described Rust as being based on "mostly decades-old research." [16]

During the early years, the Rust compiler was written in about 38,000 lines of OCaml.[17]: 15:34 [19] Early Rust contained features such as explicit object-oriented programming via an `obj` keyword (later removed),[17]: 10:08 and a `typestates` system that would allow variables of a type to be tracked along with state changes (such as going from uninitialized to initialized).[17]: 13:12

#### Mozilla sponsorship (2009-2012)

Mozilla officially sponsored the Rust project in 2009.[16] Brendan Eich and other executives, intrigued by the possibility of using Rust for a safe web browser engine, placed engineers on the project including Patrick Walton, Niko Matsakis, Felix Klock, and Manish Goregaokar.[16] A conference room taken by the project developers was dubbed "the nerd cave," with a sign placed outside the door.[16]

During this time period, work had shifted from the initial OCaml compiler to a self-hosting compiler, i.e., written in Rust, based on LLVM.[20][note 4] The Rust ownership system was also in place by 2010.[16] The Rust logo was developed in 2011 based on a bicycle chainring.[22]

The first public release, Rust 0.1 was released on January 20, 2012[23] for Windows, Linux, and MacOS.[24] The early 2010s saw increasing involvement from open source volunteers outside of Mozilla and outside of the United States. At Mozilla, executives would eventually employ over a dozen engineers to work on Rust full time over the next decade.[16]

#### Evolution (2012-2015)

The years from 2012 to 2015 were marked by substantial changes to the Rust type system, especially, removal of the `typestates` system, consolidation of other language features, and the removal of the garbage collector.[17]: 18:36 [16] Memory management through the ownership system was gradually consolidated and expanded to prevent memory-related bugs. By 2013, the garbage collector feature was rarely used, and was removed by the team in favor of the ownership system.[16] Other changes during this time included the removal of pure functions, which were declared by an explicit `pure` annotation, in March 2013.[25] Specialized syntax support for channels and various pointer types were removed to simplify the language.[17]: 22:32

Rust's expansion and consolidation was influenced by developers coming from C++ (e.g., low-level performance of features), scripting languages (e.g., Cargo and package management), and functional programming (e.g., type systems development).[17]: 30:50

Graydon Hoare stepped down from Rust in 2013.[16] This allowed it to evolve organically under a more federated governance structure, with a "core team" of initially 6 people,[17]: 21:45 around 30-40 developers total across various other teams,[17]: 22:22 and a Request for Comments (RFC) process for new language features added in March 2014.[17]: 33:47 The core team would grow to 9 people by 2016[17]: 21:45 with over 1600 proposed RFCs.[17]: 34:08

According to Andrew Binstock writing for Dr. Dobbs' Journal in January 2014, while Rust was "widely viewed as a remarkably elegant language", adoption slowed because it radically changed from version to version.[26] Rust development at this time was focused on finalizing the language features and moving towards 1.0 so it could begin promising backward compatibility.[17]: 41:26

Six years after Mozilla sponsored its development, the first stable release, Rust 1.0, was published on May 15, 2015.[16] A year after the release, the Rust compiler had accumulated over 1,400 contributors and there were over 5,000 third-party libraries published on the Rust package management website Crates.io.[17]: 43:15

## Servo and early adoption (2015–2020)

### Early homepage of Mozilla's Servo browser engine

The development of the Servo browser engine continued in parallel with Rust, jointly funded by Mozilla and Samsung.[27] The teams behind the two projects worked in close collaboration; new features in Rust were tested out by the Servo team, and new features in Servo were used to give feedback back to the Rust team.[17]: 5:41 The first version of Servo was released in 2016.[16] The Firefox web browser shipped with Rust code as of 2016 (version 45),[17]: 53:30 [28] but components of Servo did not appear in Firefox until September 2017 (version 57) as part of the Gecko and Quantum projects.[29]

Improvements were made to the Rust toolchain ecosystem during the years following 1.0 including Rustfmt, integrated development environment integration,[17]: 44:56 a regular compiler testing and release

cycle,[17]: 46:48 a community code of conduct, and community discussion organized through an IRC chat.[17]: 50:36

The earliest adoption outside of Mozilla was by individual projects at Samsung, Facebook (now Meta Platforms), Dropbox, and others including Tilde, Inc. (the company behind ember.js).[17]: 55:44 [16] Amazon Web Services followed in 2020.[16] Engineers cited performance, lack of a garbage collector, safety, and pleasantness of working in the language as reasons for the adoption, while acknowledging that it was a risky bet as Rust was new technology. Amazon developers cited the fact that Rust uses half as much electricity as similar code written in Java, behind only C,[16] as found by a study at the University of Minho, NOVA University Lisbon, and the University of Coimbra.[30][note 5]

Mozilla layoffs and Rust Foundation (2020-present)

In August 2020, Mozilla laid off 250 of its 1,000 employees worldwide, as part of a corporate restructuring caused by the COVID-19 pandemic.[31][32] The team behind Servo was disbanded. The event raised concerns about the future of Rust, due to the overlap between the two projects.[33] In the following week, the Rust Core Team acknowledged the severe impact of the layoffs and announced that plans for a Rust foundation were underway. The first goal of the foundation would be to take ownership of all trademarks and domain names, and take financial responsibility for their costs.[34]

On February 8, 2021, the formation of the Rust Foundation was announced by five founding companies: Amazon Web Services, Google, Huawei, Microsoft, and Mozilla.[35][36] The foundation, led by Shane Miller for its first two years, offered \$20,000 grants and other support for programmers working on major Rust features.[16] In a blog post published on April 6, 2021, Google announced support for Rust within the Android Open Source Project as an alternative to C/C++.[37]

On November 22, 2021, the Moderation Team, which was responsible for enforcing the community code of conduct, announced their resignation "in protest of the Core Team placing themselves unaccountable to anyone but themselves".[38] In May 2022, the Rust Core Team, other lead programmers, and certain members of the Rust Foundation board implemented governance reforms in response to the incident.[39]

The Rust Foundation posted a draft for a new trademark policy on April 6, 2023, including rules for how the Rust logo and name can be used,

which resulted in negative reactions from Rust users and contributors.[40]

On February 26, 2024, the U.S. White House released a 19-page press report urging software development to move to memory-safe programming languages; specifically, moving away from C and C++ and encouraging languages like C#, Go, Java, Ruby, Swift, and Rust.[41][42] The report was widely interpreted as increasing interest in Rust.[43][44] The report was released through the Office of the National Cyber Director.[41][45]

### Syntax and features

Rust's syntax is similar to that of C and C++, [46][47] although many of its features were influenced by functional programming languages such as OCaml.[48] Hoare has described Rust as targeted at frustrated C++ developers and emphasized features such as safety, control of memory layout, and concurrency.[18] Safety in Rust includes the guarantees of memory safety, type safety, and lack of data races.

### Hello World program

Below is a "Hello, World!" program in Rust. The `fn` keyword denotes a function, and the `println!` macro (see § Macros) prints the message to standard output.[49] Statements in Rust are separated by semicolons.

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

### Variables

Variables in Rust are defined through the `let` keyword.[50] The example below assigns a value to the variable with name `foo` and outputs its value.

```
fn main() {  
    let foo = 10;  
    println!("The value of foo is {foo}");  
}
```

Variables are immutable by default, and adding the `mut` keyword allows the variable to be mutated.[51] The following example uses `//`, which denotes the start of a comment.[52]

```
fn main() {  
    // This code would not compile without adding "mut".  
    let mut foo = 10;
```

```
println!("The value of foo is {foo}");
foo = 20;
println!("The value of foo is {foo}");
}
```

Multiple let expressions can define multiple variables with the same name, known as variable shadowing. Variable shadowing allows transforming variables without having to name the variables differently.[53] The example below declares a new variable with the same name that is double the original value:

```
fn main() {
    let foo = 10;
    // This will output "The value of foo is 10"
    println!("The value of foo is {foo}");
    let foo = foo * 2;
    // This will output "The value of foo is 20"
    println!("The value of foo is {foo}");
}
```

Variable shadowing is also possible for values of different types, going from a string to its length:

```
fn main() {
    let letters = "abc";
    let letters = letters.len();
}
```

### Block expressions and control flow

A block expression is delimited by curly brackets. When the last expression inside a block does not end with a semicolon, the block evaluates to the value of that trailing expression:[54]

```
fn main() {
    let x = {
        println!("this is inside the block");
        1 + 2
    };
    println!("1 + 2 = {x}");
}
```

Trailing expressions of function bodies are used as the return value:[55]

```
fn add_two(x: i32) -> i32 {
    x + 2
}
```

if expressions

An if conditional expression executes code based on whether the given value is true. else can be used for when the value evaluates to false, and else if can be used for combining multiple expressions.[56]