



TUTORIAL

RUST

Andres Ardila
Andres Ricardo Perez
Diego Alejandro Bayona
Alejandra Zambrano

01

¿Que es Rust?

Introducción y características

02

Ventajas y desventajas

Lo bueno y lo malo de desarrollar en Rust

03

Usos

En qué aplicaciones se utiliza Rust hoy en día

04

¡Tutorial!

Ejemplos prácticos de las funcionalidades del lenguaje



01

Que es Rust

Introducción y características

Rust

“Rust is a modern systems programming language focusing on safety, speed, and concurrency. It accomplishes these goals by being memory safe without using garbage collection.”

~ Rust documentation

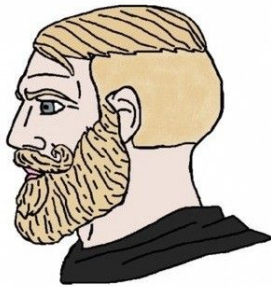


Ferris the crab

Relación con programación concurrente



Lock



Unlock

Send y Sync

Send: este trait le indica al compilador que algo de un tipo $\langle T \rangle$ pueda transferir la pertenencia entre hilos de forma segura.

Sync: le indica al compilador que algo de tipo $\langle T \rangle$ no tiene posibilidad de introducir inseguridad en memoria cuando es usado de manera concurrente por múltiples hilos de ejecución.

Características importantes de Rust



Ownership

Los valores son como libros tienen dueño



Borrowing

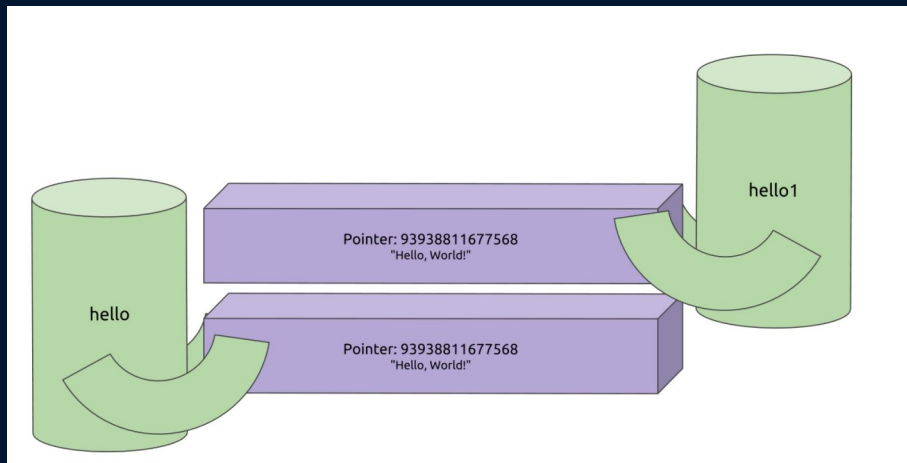
Las variables para acceder a los valores necesitan “pedirlos prestados”



Lifetimes

Este es el tiempo entre la creación de la variable hasta cuando sale de su alcance

Ownerships



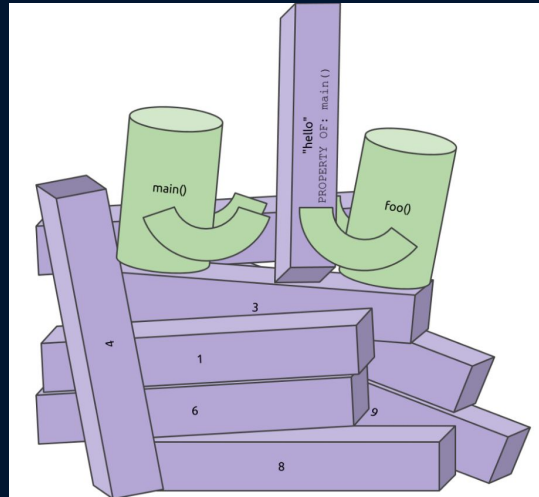
Each value in Rust has a variable that's called its *owner*.

There can only be one owner at a time.

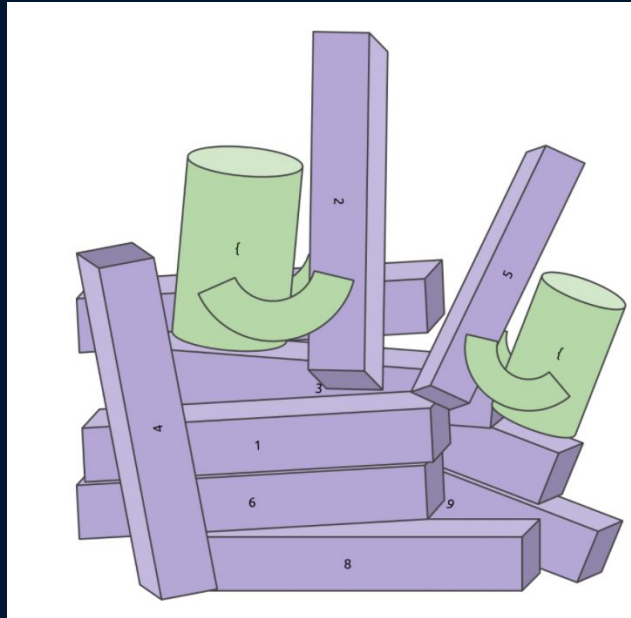
When the owner goes out of scope, the value will be dropped.

Borrowing

`main()` gives `foo()` access to `string`, but, (as indicated by the label), `main()` is still the owner of `string`. This means that at the end of `foo()`'s scope, `string` will not be dropped from memory; `main()` is still responsible for `string`'s space in memory.



Lifetimes





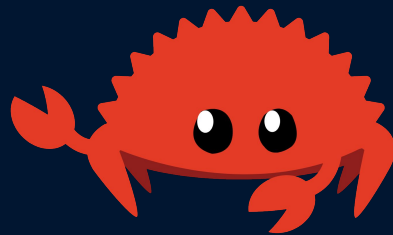
02

Ventajas y desventajas

Razones para usar o no Rust

Ventajas

- Mejor manejo de memoria mediante el compilador
 - Permite control de los errores de memoria de bajo nivel sin utilizar un garbage collector
- Concurrencia más sencilla
 - Gracias al modelo de Ownership
 - Evita las condiciones de carrera con el borrow checker
- Abstracciones sin costo
 - No se tiene una diferencia en tiempo de ejecución entre código de bajo nivel y código con abstracciones



Desventajas

- Curva de aprendizaje lenta
 - El desarrollo se puede tornar más largo en tiempo
- Compilación más lenta
- Por ser tan reciente, no hay muchos recursos por la comunidad
- Archivos ejecutables más pesados que en otros lenguajes





03

Usos de Rust

Para que se está utilizando el
lenguaje hoy en día

Usos

- Herramientas para línea de comandos multi-plataforma
- Servicios distribuidos en línea
- Dispositivos embebidos
- Programación de **sistemas**
 - Diseñar y escribir códigos que permiten al hardware tener interacción con el programador y el usuario, que permitan la ejecución del código de aplicaciones
 - Por ejemplo programas como los motores de los navegadores, e incluso algo como un kernel.

Plataformas que usan Rust



POSTMATES

moz://a



deliveroo

coursera

ACADEMIA



Dropbox



CLOUDFLARE





04 | Tutorial!

Bibliografía

[1] S. Klabnik and C. Nichols, The Rust Programming Language. 2018. Disponible en: <https://doc.rust-lang.org/book/title-page.html>

[2]"Introduction - Rust By Example", Doc.rust-lang.org. [Online]. Disponible en: <https://doc.rust-lang.org/stable/rust-by-example/>. [Accesado en: 10- Jun- 2021].

[3] T. Countz, "Ownership in Rust, Part 1", Medium, 2018. [Online]. Disponible en: <https://medium.com/@thomascountz/ownership-in-rust-part-1-112036b1126b>. [Accesado en: 12- Jun- 2021].

[4]G. Dreimanis, "Introduction to Rust", Serokell Software Development Company, 2020. [Online]. Disponible en: <https://serokell.io/blog/rust-guide>. [Accesado en: 13- Jun- 2021].

[5]"What is System Programming? - Definition from Techopedia", Techopedia.com. [Online]. Disponible en: <https://www.techopedia.com/definition/9616/system-programming>. [Accesado en: 13- Jun- 2021].

GRACIAS!

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, and infographics & images by Freepik.

Please keep this slide for attribution.