The Gist of Rust Plan

Mustafif Khan

July 22, 2021

Contents

Chapter 1

Getting Started

1.1 What is Rust?

- When was Rust developed?
- Why learn Rust?
- Who created Rust?
- The idea of Memory safety and zero abstraction
- Low level programming with high level functionalities
- Statically typed language
- No garbage collector
- How to install via Rustup

1.2 What is Cargo

- Rust's package manager
- Exploring Crates.io
- The difference between bin & lib

1.3 Program Structure

- Creating a new cargo project
- Inspecting the hello world program
- Insepcting Cargo.toml

Chapter 2

Basics Of Rust

2.1 Variables & Data Types

- The let keyword
- Mutability vs Immutability
- Implicit and explicit type declaration
- Comments
- Primitive data types

2.2 Collection Types

- Vectors (compare to C++)
- Arrays
- Tuples
- HashMaps (compare to Ruby)

2.3 Loops & Conditionals

- While loops

- For loops
- Loop
- If/Else statements
- if let statement
- Match statement

2.4 Functions

- Declaring a function
- Implicit and explicit returns
- Public and private functions

Chapter 3

More into Rust

3.1 Ownership in Rust

- Rust's borrow checker
- References and Borrowing
- Rust's two string types
- Smart Pointers

3.2 Structs & Enums

- Three types of structs
- Implementations
- Enumerations
- Structs in Enums

3.3 Generics, Traits & Lifetimes

- Generic functions
- Generic types

- Generic implementations
- Traits
- Marker Traits
- Simple Traits
- Generic Traits
- Associated type traits
- Trait bounds
- Scopes
- Lifetimes

3.4 Closures

- What are closures
- Fn closures
- FnMut closures
- FnOnce closures

3.5 Error Handling

- Recoverable errors with Option/Result
- More on Option/Result
- The `?` operator
- Nonrecoverable errors
- Custom errors/traits

3.6 Macros in Rust

- What are Macros
- When and when not to use Macros
- `macro_rules!`
- Builtin Macros in standard library
- Token types
- Repitition in Macros
- Procedural Macros
- Derive Macros
- Debugging Macros