한 번에 끝내는
블록체인 개발 A to Z

Chapter 1

Rust Introduction

Chapter 1

Rust Introduction

Hello World

Installation

1. Installing rustup on Linux or macOS

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

2. If you met link error install, you need to install C compiler

```
$ sudo apt-get install build-essential # Linux
$ xcode-select --install               # macOS
```

Installation

1. Make Project Directory

```
$ mkdir ~/projects
$ cd ~/projects
$ mkdir hello_world
$ cd hello_world
```

2. Writing and Running a Rust Program

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

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

Hello, Cargo!

Cargo is Rust's build system and package manager, which is capable of building your code, downloading the libraries your code depends on, and building those libraries.

1. Check Installed Cargo

```
$ cargo --version
cargo 1.61.0 (a028ae42f 2022-04-29)
```

2. Create Project with Cargo

```
$ cargo new hello_cargo
$ cd hello_cargo
```

Hello, Cargo!

Cargo is Rust's build system and package manager, which is capable of building your code, downloading the libraries your code depends on, and building those libraries.

❸ Check Installed Cargo

```
[package]
name = "hello_cargo"
version = "0.1.0"
edition = "2021"

[dependencies]
```

❹ Write src/main.rs (Cargo expects your source files to live inside the src directory)

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

Hello, Cargo!

Cargo is Rust's build system and package manager, which is capable of building your code, downloading the libraries your code depends on, and building those libraries.

❺ Build and Run a Cargo Project

```
$ cargo build
   Compiling hello_cargo v0.1.0 (file:///projects/hello_cargo)
    Finished dev [unoptimized + debuginfo] target(s) in 2.85 secs

$ ./target/debug/hello_cargo
Hello, world!
```

❻ Run with Cargo

```
$ cargo run
    Finished dev [unoptimized + debuginfo] target(s) in 0.0 secs
     Running `target/debug/hello_cargo`
Hello, world!
```

Hello, Cargo!

Cargo is Rust's build system and package manager, which is capable of building your code, downloading the libraries your code depends on, and building those libraries.

❼ Check (make sure it compiles but doesn't produce an executable)

```
$ cargo check
   Checking hello_cargo v0.1.0 (file:///projects/hello_cargo)
    Finished dev [unoptimized + debuginfo] target(s) in 0.32 secs
```

❽ Build for Release (to compile it with optimizations)

```
$ cargo build --release
   Compiling hello_cargo v0.1.0 (file:///projects/hello_cargo)
    Finished release [optimized] target(s) in 0.33s
```