

## 1. 认识 cargo

cargo 是 Rust 的包管理工具，并且是所有语言中最好用的包管理工具之一。cargo 提供了一系列的工具，从项目的建立、构建到测试、运行直至部署，为 Rust 项目的管理提供尽可能完整的手段。

## 2. 创建一个“hello\_world”项目

```
cargo new hello_world
```

Rust 项目名的多个单词间可以用“\_”或“-”分隔，例如：“hello\_world”、“hello-world”，个人喜欢用“\_”分隔。Rust 项目分为 2 种，Binary 项目（默认）和 Library 项目，Binary 项目是可执行的，Library 项目是不可执行的。

示例：

```
cargo new --bin hello_world
cargo new --lib hello_world
```

## 3. 运行项目（当前目录位于 Cargo.toml 所在的目录中）

代码在对应项目的 main() 中。

```
cargo run          # 以 Debug 模式编译和运行项目
cargo run --release # 以 Release 模式编译和运行项目

cargo build        # 以 Debug 模式编译项目
cargo build --release # 以 Release 模式编译项目
```

## 4. 快速验证代码能否通过编译

```
cargo check
```

## 5. Cargo.toml 和 Cargo.lock

- Cargo.toml 是 cargo 的项目描述文件，存储着项目的所有元配置信息。

- Cargo.lock 是 cargo 根据 Cargo.toml 生成的项目详细依赖清单，不需要修改这个文件。

## 6. 覆盖默认的镜像地址

在 \$HOME/.cargo/config.toml 中添加以下内容：

```
[source.crates-io]
replace-with = 'ustc'
```

```
[source.ustc]
registry = "git://mirrors.ustc.edu.cn/crates.io-index"
```