Rust 工具之 cargo

作者：lingyundu
原文链接：https://ld246.com/article/1607699415160
来源网站：链滴

Cargo 是 Rust 的代码组织管理和项目构建工具，使用 rustup 安装 Rust 时，Cargo 默认也会被安
。

Cargo 的主要用途：

- 创建和管理 Rust 的模块系统 。
- 下载和管理依赖包。
- 调用 rustc或其他构建工具来构建项目（应用）。

Rust提供了一套模块系统来组织和管理代码，包括：模块（module）、Crate、包（package）和
作空间（workspace）。

其中，包（package）是包含一个或者多个 Crate的目录结构，类似于其他编程语言中**项目**的概念。而
rate 的英文意思是箱子，它是一个 模块树，并且是编译的基本单元，可以将其编译成可执行程序（ex
cutable）或者库（library）。

创建包

用来创建包的命令是： cargo new ，通过 cargo new --help 可以查看该命令的帮助信息。

创建一个简单的包：

```
$ cargo new hello_world --bin // --bin 表示创建的包中包含一个可编译成可执行文件的 Crate
```

hello_world 目录包含一个 Cargo.toml 文件和一个 src 目录。Cargo.toml 文件是一个配置文件，包
包名、版本、作者、依赖配置等信息。src 目录用来存放源码文件，其中 main.rs 是约定的可执行程
的入口文件。

```
├── Cargo.toml
└── src
    └── main.rs
```

管理依赖

在开发过程中，如果用到了其他的 包，只需要将它们配置到 Cargo.toml 文件。在编译的时候，Carg
会自动下载这些依赖 包，以及这些 依赖 包 的依赖包。

这些依赖包的信息存放在crates.io，我们也可以将自己开发的库上传到该网站分享给全世界(^_^)。

例如，要添加 time和 regex 依赖，将其名称和版本添加到 Cargo.toml 文件的 [dependencies] 的
方即可。

```
[package]
name = "hello_world"
version = "0.1.0"
authors = ["Your Name <you@example.com>"]
edition = "2018"

[dependencies]
time = "0.1.12"
regex = "0.1.41"
```

构建和运行

在 包目录下，运行 cargo build 命令构建项目，然后在 target/debug 目录中我们可以找到生成的可行文件或者库文件。

```
PS D:\rust\hello_world> cargo build
   Compiling hello_world v0.1.0 (D:\Github\hello_world)
    Finished dev [unoptimized + debuginfo] target(s) in 0.98s
```

另外，对于生成可执行文件的 包，Cargo 还提供了 cargo run 命令，该命令先构建项目，然后会运生成的可执行程序。

```
PS D:\rust\hello_world> cargo run
    Finished dev [unoptimized + debuginfo] target(s) in 0.01s
     Running `target\debug\hello_world.exe`
Hello, world!
PS D:\rust\hello_world>
```

相关资料

The Cargo Book

The Rust community’ s crate registry

Managing Growing Projects with Packages, Crates, and Modules