

## Rust Programming Language

Install

Windows

Cargo

Reference

Rust

# Rust Programming Language

`rust` 基于 `c++` 的一种语言，语法和 `c++` 类似，但是在内存管理有优势。也是一种预编译语言，需要使用 `c++` 的编译器。

## Install

---

### Windows

- (1) 存在两个版本，可以选择安装 `MinGW/MSVC`
- (2) 下载 `rustup-init.exe`。

下载地址: <https://www.rust-lang.org/zh-CN/tools/install>

`rustup` 是 `rust` 的环境版本管理工具。

- (3) `rustup` 自定义安装路径:
  1. 创建存放rust环境目录 `Rust/` ;
  2. 进入目录, 创建 `.cargo/` 和 `.rustup/` 目录, 这两个目录就是工具的安装路径;
  3. 配置系统环境变量。创建两个系统环境变量 `CARGO_HOME` 和 `RUSTUP_HOME` , 两个系统环境变量的位置分别为第二步创建的两个目录。然后在 `PATH` 中添加 `%CARGO_HOME%` 和 `%RUSTUP_HOME%` ;
  4. 启动 `rustup-init.exe` , 默认下载和使用MSVC版本的rust。输入3选择使用GNU版本;
  5. 然后输入2, 进行手动配置下载;

1. 输入 `x86_64-pc-windows-gnu` 版本
  2. 输入 `stable` 版本
  3. 输入 `complete` 进行完整安装
  4. 输入 `Y` 允许修改系统环境变量
6. 最后输入1按照配置进行安装。

#### (4) 检查

```
1 rustc --version
2 cargo --version
```

## Cargo

---

创建一个名为 `project_name` 的项目文件。

```
1 cargo new project_name
2 cd ./project_name # 进入项目目录
```

使用debug模式构建项目，生成 `project_name.exe` 可执行文件。

```
1 cargo build
2 ./target/debug/project_name.exe
```

使用debug模式构建并运行项目

```
1 cargo run
```

对项目进行构建，但是不生成可执行文件

```
1 cargo check
```

使用发布模式进行构建，生成的 `project_name.exe` 运行速度更快。

```
1 cargo build --release
2 cargo run --release
```

更新项目依赖版本

```
1 cargo update
```

查看依赖api文档

```
1 cargo doc --open # 生成当前项目的依赖的api文档
```

一般使用 `vscode` 或 `clion` 进行编写，写好程序后，在IDE的终端里面输入 `cargo run` 进行运行程序。

## Reference

---

官方英文文档地址: <https://www.rust-lang.org/>

Rust Programming language: <https://doc.rust-lang.org/book/>

中文翻译版本文档地址: <https://www.rustwiki.org.cn/>

Rust程序设计语言: <https://www.rustwiki.org.cn/zh-CN/book/>

Rust圣经: <https://course.rs/about-book.html>

## Rust

---

引入依赖。在 `Cargo.toml` 中写入依赖以及版本，然后重新构建项目。

```
1 [dependencies]
2 rand = "0.8.5"
```

下载依赖

```
1 cargo build
```