

# 使用Rust构建wasm

现在我们开始在本机安装Rust环境

## 1.安装 Rust 工具链

## 2.安装 wasm workflow tool! (划重点):

Rust是一门很新的编程语言，类似C++，无GC、无Run time有着很好的性能，在FaceBook内使用Rust构建了一个Libra项目，使Rust名声大噪。2017–2018年蝉联两年stackoverflow 最受欢迎的语言第一名。有着替代C++的潜力。

对于我们前端来讲更关注的是如何使用一种高效的编程语言来实现javascript里的高密度计算的代码，能够解决javascript和webassembly传递复杂参数的问题，并且能够方便快捷的编译成.wasm文件。 社区内大部分工具都支持C/C++编译成.wasm，不过如果想在本地编译wasm 本地需要安装很复杂的依赖 python 3+、xcode等，这是一个工具链，并且这些过程跟javascript 一点关系也没有，不是工作的重点，却很耗时费力。方便的是有很多在线编译工具比如 [WasmExplorer](#)。虽然编译工具链可以通过在线工具解决，但是javascript和webassembly之间复杂参数传递的问题还是一个头疼的问题。

Rust，是目前对wasm编译最友好的语言，一键编译成.wasm文件。并且还有“npm”包 [wasm-bindgen](#) 用来解决javascript和webassembly间复杂参数传递。这也是我们使用Rust的主要原因。

## 现在我们开始在本机安装Rust环境

(Rust的安装非常简单)

## 1.安装 Rust 工具链

Rust 工具链包括：

- **rustup**：负责安装 Rust、切换 Rust 版本、下载标准库文件等
- **rustc**：Rust 的编译器（一般通过 Cargo 命令调用）
- **cargo**：Rust 的项目管理工具（类似NPM）

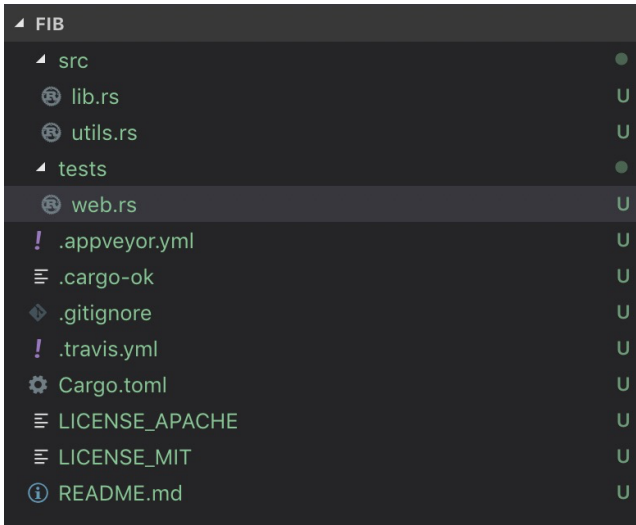
运行：`curl https://sh.rustup.rs -sSf | sh` 即可安装

安装完成后可以参照[Rust官方](#)上的demo，试着Run起你的第一个用Rust实现的Hello World。

首先使用cargo 生成一个Rust项目：

```
1 cargo generate --git https://github.com/rustwasm/wasm-pack-template
```

生成的项目目录：



## 2.安装 **wasm workflow tool!** (划重点):

wasm-pack 用于将 Rust 项目打包成单个 WASM 文件 (类似 Webpack)

```
curl https://rustwasm.github.io/wasm-pack/installer/init.sh -sSf | sh
```

按照好 wasm-pack后, cd到项目目录下, 运行**wasm-pack build** 即可编译出.wasm文件 是不是比 C/C++方便很多~

**FIB**

- pkg
- src
  - lib.rs
  - utils.rs
- target
  - release
    - wasm32-unknown-unknown
      - release
        - .fingerprint
        - build
        - deps
        - examples
        - incremental
        - .cargo-lock
        - fib.d**
        - fib.wasm**
        - libfib.d
        - libfib.rlib
        - .rustc\_info.json
- tests
- .appveyor.yml
- .cargo-ok
- .gitignore
- .travis.yml
- Cargo.lock

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL**

```
admindeMacBook-Pro-3:fib joy$ wasm-pack build
[INFO]: Checking for the Wasm target...
info: downloading component 'rust-std' for 'wasm32-unknown-unknown'
10.8 MiB / 10.8 MiB (100 %) 441.6 KiB/s in 28s ETA: 0s
info: installing component 'rust-std' for 'wasm32-unknown-unknown'
[INFO]: Compiling to Wasm...
Compiling proc-macro2 v1.0.10
Compiling unicode-xid v0.2.0
Compiling log v0.4.8
Compiling wasm-bindgen-shared v0.2.60
Compiling syn v1.0.17
Compiling cfg-if v0.1.10
Compiling bumpalo v3.2.1
Compiling lazy_static v1.4.0
Compiling wasm-bindgen v0.2.60
Compiling quote v1.0.3
Compiling wasm-bindgen-backend v0.2.60
Compiling wasm-bindgen-macro-support v0.2.60
Compiling wasm-bindgen-macro v0.2.60
Compiling console_error_panic_hook v0.1.6
Compiling fib v0.1.0 (/Users/joy/code/fib)
warning: function is never used: `set_panic_hook`
--> src/utils.rs:1:8
1 | pub fn set_panic_hook() {
  | ~~~~~
  = note: `#[warn(dead_code)]` on by default

Finished release [optimized] target(s) in 23.91s
[INFO]: Installing wasm-bindgen...
```

[Rust背景](#)

[Rust社区](#)