

Please visit website: <http://cxyroad.com>

赖，只需在Cargo.toml中指定库名和版本即可。执行cargo build或cargo run命令后，Cargo会自动解析依赖关系，下载所需的依赖包，并编译项目的源代码。构建完成后，可以在target/debug或target/release目录下找到生成的可执行文件或库文件。

```
...  
[dependencies]  
rand = "0.8" # 添加随机数生成库rand，版本号为0.8  
clap = "3.0"
```

```
...
```

添加测试

Cargo内置了测试框架，我们可以编写测试代码来验证项目的正确性。在Rust项目中，测试文件通常放在tests目录下，并以_test.rs为后缀。比如：可以创建一个名为project_test.rs的测试文件，并编写以下测试代码。

```
...  
fn add(a: i32, b: i32) -> i32 {  
    return a + b;  
}  
  
#[test]  
fn test_add() {  
    assert_eq!(add(2, 3), 5);  
    assert_eq!(add(-1, 1), 0);  
}  
...
```

在Rust中，#[test]是一个属性宏，用于标识一个函数为测试函数。当运行cargo test命令时，Rust会自动执行所有标记了#[test]属性的函数，用于验证

```
ate::print_text;  
///  
/// print_text("World");  
/// ````  
pub fn print_text(text: &str) {  
    println!("{}", format!("{}", World", text));  
}
```

```
fn main() {  
    print_text("Hello");  
}
```

...

2、打开终端，导航到项目的根目录（包含Cargo.toml的地方），然后运行：
cargo doc。这个命令会编译项目并为所有公共接口生成文档，输出结果默认位于target/doc目录下。打开target/doc/<crate_name>/index.html文件，即可查看生成的文档。

3、如果要在本地启动一个Web服务器实时预览文档，可在生成文档后执行：
cargo doc --open，这样rustdoc将自动在浏览器中打开文档索引页。

总结

通过以上的介绍，我们可以看到Cargo在Rust项目中的重要作用。它不仅简化了构建和分发过程，还提供了丰富的功能和灵活的扩展性。无论是初学者还是经验丰富的开发者，都可以通过Cargo来更加高效地创建、测试、文档化和分享自己的Rust项目。

原文链接: <https://juejin.cn/post/7372094258792628278>