

excaliburdraco / Rustlings

🔍

👤

▼

📁

👤

<> Code

🔗 Pull requests

🎬 Actions

📁 Projects

📖 Wiki

🛡 Security

📊 Insights

⚙ Settings

👁

🔗

☆

first2025rustlings

📄 MIT license

☆ 0 stars

🔗 257 forks

👁 0 watching

🔗 Branches

🏠 Activity

🏠 Tags

🌐 Public repository · Forked from [opencamp-cn/Rustlings](#)

🔗 main ▼

🔗 1 Branch

🏠 0 Tags

🔗

🏠

🔍 Go to file

t

Go to file

+

Add file ▼

Code

⋮

This branch is up to date with opencamp-cn/Rustlings:main .

🔗 Contribute ▼

🔄 Sync fork ▼

👤 kunyuanxu-star

doc: update README

341827f · 3 weeks ago

🔄

📁 .devcontainer	Initial commit	last month
📁 .github	update ci env	last month
📁 .vscode	Initial commit	last month
📁 exercises	init project and change CI	last month
📁 src	Initial commit	last month
📁 tests	Initial commit	last month
📄 .all-contributorsrc	Initial commit	last month
📄 .editorconfig	Initial commit	last month
📄 .gitignore	Initial commit	last month
📄 .gitpod.yml	Initial commit	last month
📄 .markdownlint.yml	Initial commit	last month
📄 AUTHORS.md	Initial commit	last month
📄 CHANGELOG.md	Initial commit	last month
📄 CONTRIBUTING.md	Initial commit	last month
📄 Cargo.lock	Initial commit	last month
📄 Cargo.toml	Initial commit	last month
📄 LICENSE	Initial commit	last month
📄 README.md	doc: update README	3 weeks ago
📄 flake.lock	Initial commit	last month
📄 flake.nix	Initial commit	last month
📄 info.toml	init project and change CI	last month
📄 install.ps1	Initial commit	last month
📄 install.sh	Initial commit	last month
📄 oranda.json	Initial commit	last month
📄 shell.nix	Initial commit	last month

[README](#) [License](#)

1. 在网络浏览器中用自己的 github id 登录 github.com。
2. 请将本仓库 fork 到您的 github 账号下，然后参照以下步骤完成环境配置和实验提交：
 - 本地环境：
 - i. **安装Linux的环境。**对于windows的用户，推荐使用wsl2安装Ubuntu 22.04，也可以使用vmware等虚拟机进行安装。如果在这一步存在问题，请联系助教。
 - ii. **创建ssh key，用于ssh方式克隆github代码。**在linux环境下，使用 `ssh-keygen -t rsa -b 4096 -C "你的邮箱"` 命令，创建ssh key，下面的选项全部直接敲回车即可。随后使用 `cat ~/.ssh/id_rsa.pub` 命令查看生成的公钥，并完整的复制下来。在github仓库界面点击自己的头像，选择 settings。进入到设置页面后，点击左侧的 SSH and GPG keys 选项。点击 New SSH key 选项，并将复制下来的内容粘贴上去，添加该ssh key的描述。随后点击 Add SSH key，并一路点击确认即可。
 - iii. **本地安装rust。**进入linux环境下，参考Arceos 教程 [Rust 开发环境配置 - ArceOS Tutorial Book \(rcore-os.cn\)](#) 中，找到Rust 开发环境配置的章节，相应配置即可，你可以同时将后续需要的环境也配置好。
 - iv. ****clone实验仓库到本地。****在前面点击链接生成的仓库中，同样点击醒目的 code 绿色按钮，选择 local 下的 ssh 选项，复制下面的链接。随后回到本地linux环境下，使用 `git clone` 复制的链接 的方式，将目标仓库clone到本地。随后，使用 `ls` 命令查看自己clone下来的文件夹，再使用 `cd` 命令进入到该文件夹下，使用 `cargo install --force --path .` 安装rustlings。
 - v. **练习rustlings。**使用vscode等编辑器，进入clone下来的目录下的 exercises 文件夹，执行 `rustlings watch` 依次查看完成情况，并依次完成对应的练习。执行 `rustlings run` 练习名称 去运行对应练习，也可以使用 `rustlings hint` 练习名称 查看题解。
 - vi. **提交完成情况。**当做完部分或所有练习之后，在rustlings目录下执行 `git add .` ; `git commit -m "update"`; `git push` 命令，把更新提交到GithubClassroom的CI进行自动评测。你可以在github仓库页面的actions页面，看到你的CI提交结果，或者 <https://opencamp.ai/Rust/camp/S01/stage/0?tab=rank> 上面查看自己的评分。

- 在线环境：
 - i. 如果使用在线环境，在本网页的中上部可以看到一个醒目的 `code` 绿色按钮，点击后，可以进一步看到 `codespace` 标签和醒目的 `create codespace on main` 绿色按钮。请点击这个绿色按钮，就可以进入到在线的ubuntu +vscode环境中
 - ii. 再按照下面的环境安装提示在vscode的 `console` 中安装配置开发环境：`rustc`等工具。
 - iii. 然后就可以基于在线vscode进行测试 (执行命令 `rustlings watch`)，编辑代码的循环实验过程了。
3. 上述步骤有任何问题都可以找助教。
4. 下面是官方的Rustlings的布置，可以参考，**请务必不要拉取下面的仓库！**

Greetings and welcome to `rustlings` . This project contains small exercises to get you used to reading and writing Rust code. This includes reading and responding to compiler messages!

...looking for the old, web-based version of Rustlings? Try [here](#)

Alternatively, for a first-time Rust learner, there are several other resources:

- [The Book](#) - The most comprehensive resource for learning Rust, but a bit theoretical sometimes. You will be using this along with Rustlings!
- [Rust By Example](#) - Learn Rust by solving little exercises! It's almost like `rustlings`, but online

Note: If you're on MacOS, make sure you've installed Xcode and its developer tools by typing `xcode-select --install`. Note: If you're on Linux, make sure you've installed gcc. Deb: `sudo apt install gcc`. Yum: `sudo yum -y install gcc`.

You will need to have Rust installed. You can get it by visiting <https://rustup.rs>. This'll also install Cargo, Rust's package/project manager.

MacOS/Linux

Just run:

```
curl -L https://raw.githubusercontent.com/rust-lang/rustlings/main/install.sh | bash
```



Or if you want it to be installed to a different path:

```
curl -L https://raw.githubusercontent.com/rust-lang/rustlings/main/install.sh | bash -s mypath/
```



This will install Rustlings and give you access to the `rustlings` command. Run it to get started!

Nix

Basically: Clone the repository at the latest tag, finally run `nix develop` or `nix-shell`.

```
# find out the latest version at https://github.com/rust-lang/rustlings/releases/latest (on edit 5.5.1)
git clone -b 5.5.1 --depth 1 https://github.com/rust-lang/rustlings
cd rustlings
# if nix version > 2.3
nix develop
# if nix version <= 2.3
nix-shell
```



Windows

In PowerShell (Run as Administrator), set `ExecutionPolicy` to `RemoteSigned`:

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser
```



Then, you can run:

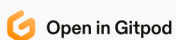
```
Start-BitsTransfer -Source https://raw.githubusercontent.com/rust-lang/rustlings/main/install.ps1 -Destination $env:TEMP
```



To install Rustlings. Same as on MacOS/Linux, you will have access to the `rustlings` command after it. Keep in mind that this works best in PowerShell, and any other terminals may give you errors.

If you get a permission denied message, you might have to exclude the directory where you cloned Rustlings in your antivirus.

Browser



Manually

Basically: Clone the repository at the latest tag, run `cargo install --path .`

```
# find out the latest version at https://github.com/rust-lang/rustlings/releases/latest (on edit 5.5.1)
git clone -b 5.5.1 --depth 1 https://github.com/rust-lang/rustlings
cd rustlings
cargo install --force --path .
```



If there are installation errors, ensure that your toolchain is up to date. For the latest, run:

```
rustup update
```



Then, same as above, run `rustlings` to get started.

Doing exercises

The exercises are sorted by topic and can be found in the subdirectory `rustlings/exercises/<topic>`. For every topic there is an additional README file with some resources to get you started on the topic. We really recommend that you have a look at them before you start.

The task is simple. Most exercises contain an error that keeps them from compiling, and it's up to you to fix it! Some exercises are also run as tests, but rustlings handles them all the same. To run the exercises in the recommended order, execute:

```
rustlings watch
```



This will try to verify the completion of every exercise in a predetermined order (what we think is best for newcomers). It will also rerun automatically every time you change a file in the `exercises/` directory. If you want to only run it once, you can use:

```
rustlings verify
```



This will do the same as `watch`, but it'll quit after running.

In case you want to go by your own order, or want to only verify a single exercise, you can run:

```
rustlings run myExercise1
```



Or simply use the following command to run the next unsolved exercise in the course:

```
rustlings run next
```



In case you get stuck, you can run the following command to get a hint for your exercise:

```
rustlings hint myExercise1
```



You can also get the hint for the next unsolved exercise with the following command:

```
rustlings hint next
```



To check your progress, you can run the following command:

```
rustlings list
```



Testing yourself

After every couple of sections, there will be a quiz that'll test your knowledge on a bunch of sections at once. These quizzes are found in `exercises/quizN.rs`.

Enabling rust-analyzer

Run the command `rustlings lsp` which will generate a `rust-project.json` at the root of the project, this allows [rust-analyzer](#) to parse each exercise.

Continuing On

Once you've completed Rustlings, put your new knowledge to good use! Continue practicing your Rust skills by building your own projects, contributing to Rustlings, or finding other open-source projects to contribute to.

Uninstalling Rustlings

If you want to remove Rustlings from your system, there are two steps. First, you'll need to remove the exercises folder that the install script created for you:

```
rm -rf rustlings # or your custom folder name, if you chose and or renamed it
```



Second, run `cargo uninstall` to remove the `rustlings` binary:



```
cargo uninstall rustlings
```

Now you should be done!

Contributing

See [CONTRIBUTING.md](#).

Development-focused discussion about Rustlings happens in the [rustlings stream](#) on the [Rust Project Zulip](#). Feel free to start a new thread there if you have ideas or suggestions!

Contributors ✨



Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

Languages

● Rust 86.6% ● Shell 6.4% ● PowerShell 4.0% ● Nix 3.0%

Suggested workflows

Based on your tech stack



SLSA Generic generator

Generate SLSA3 provenance for your existing release workflows

Configure



Rust

Build and test a Rust project with Cargo.

Configure

[More workflows](#)

Dismiss suggestions