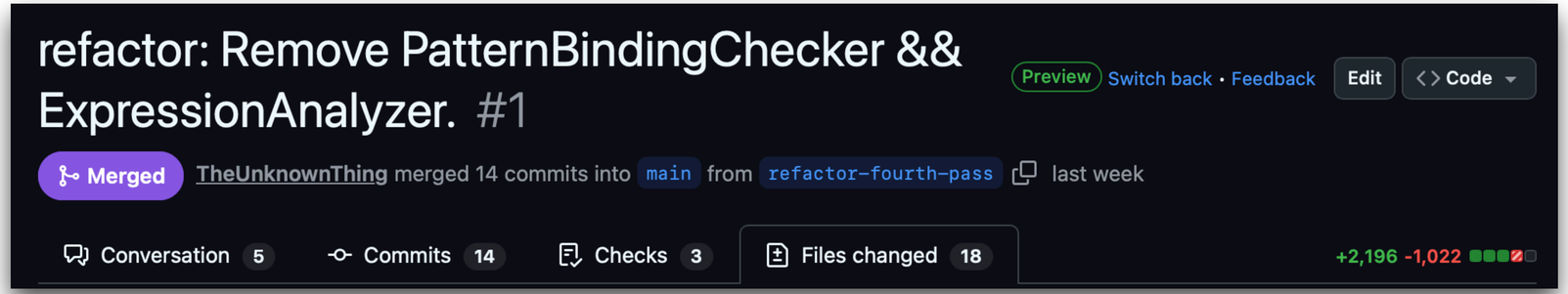
TheUnknownThing

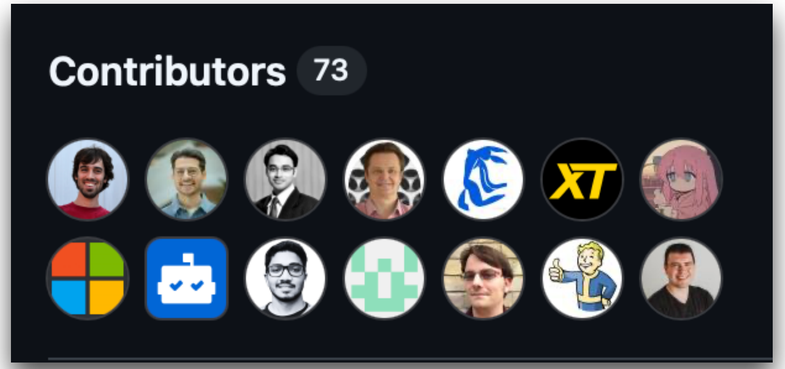
Git Operations

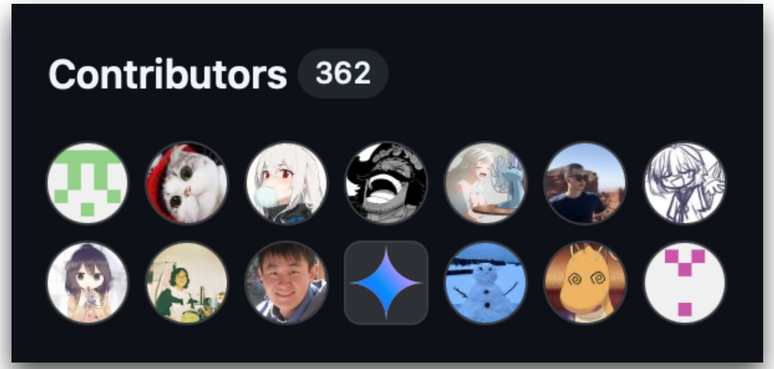
为什么要用 Git?

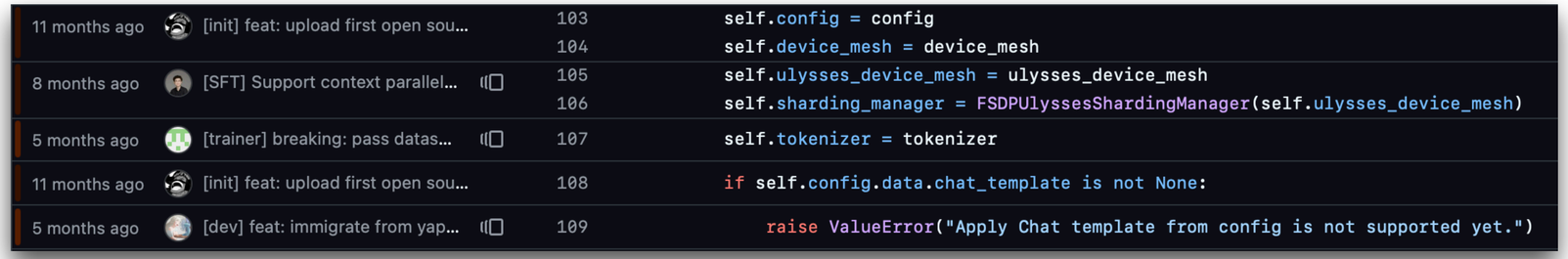
项目是复杂的

- 你会重写你的代码
- 有很多人会贡献代码
- 手贱了怎么办?
- 谁写了哪行代码?









我需要一个强大的时光机

- 你会重写你的代码
- 有很多人会贡献代码
- 手贱了怎么办?
- 谁写了哪行代码?
- 万一重构失败了怎么办😭
- 我和别人改了同一个文件怎么办🤔
- 我补药重写一遍😭
- 让我康康谁写的代码出问题了😈

“ 如果有一台时光机，该多好？ ”

为什么要用 Git?

什么是 Git?

为什么要用 Git?

git 1 of 2 noun

'git

Synonyms of *git* ›

British

: a foolish or worthless person

https://www.merriam-webster.com/dictionary/git

“I’m an egotistical bastard, and I name all my projects after myself. First ‘linux’, now ‘git’.”

Linus Torvalds

Git 是版本控制系统

版本控制系统能：

- 帮助我们管理代码的修改历史
- 让协作变得更方便。
- 自动维护一系列快照（文件 & 文件夹）

每一个仓库就是一个“时光机”

这些时光机“看起来”就是一个文件夹

```
❯ ls -la .
total 80
drwxr-xr-x@    theunknownthing  .
drwxr-x---+    theunknownthing  ..
drwxr-x--@     theunknownthing  .cache
-rw-r--r--@    theunknownthing  .clangd
drwxr-xr-x@    theunknownthing  .git
drwxr-xr-x@    theunknownthing  .github
```

你可能会问：

- 是什么魔法让 Git 与众不同？

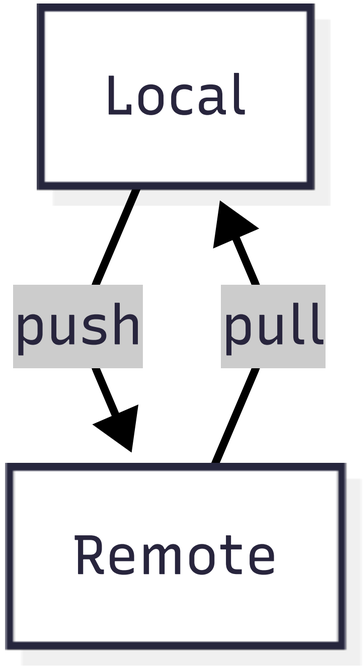
  Git 仓库 v.s. 普通文件夹

- Git 的时光机是存放在哪里的？

  .git 文件夹

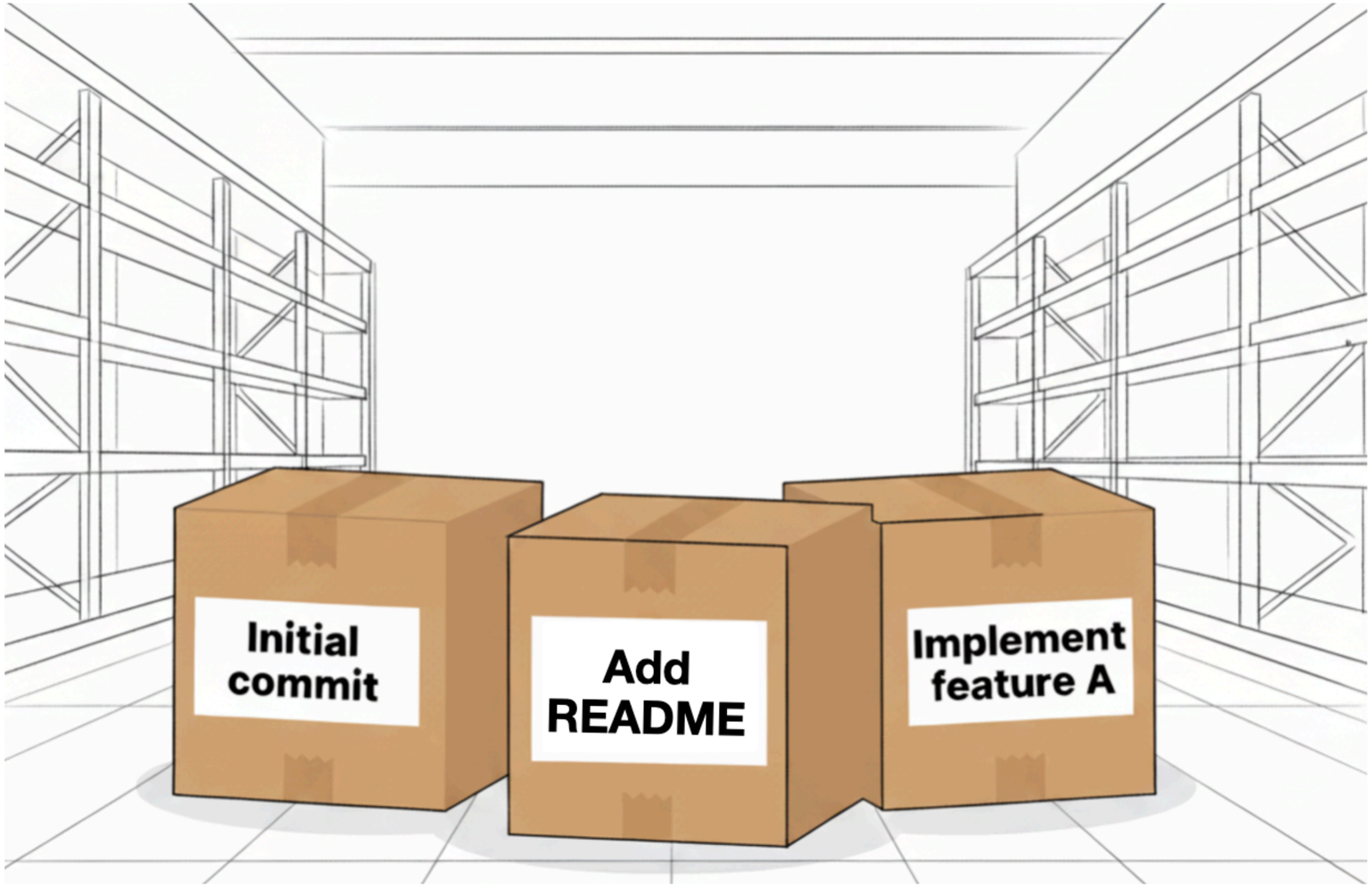
- 我要怎么和别人合作？

  本地仓库 + 云端仓库  + 



认识一些 Git 中的名词

仓库



认识一些 Git 中的名词

Commits

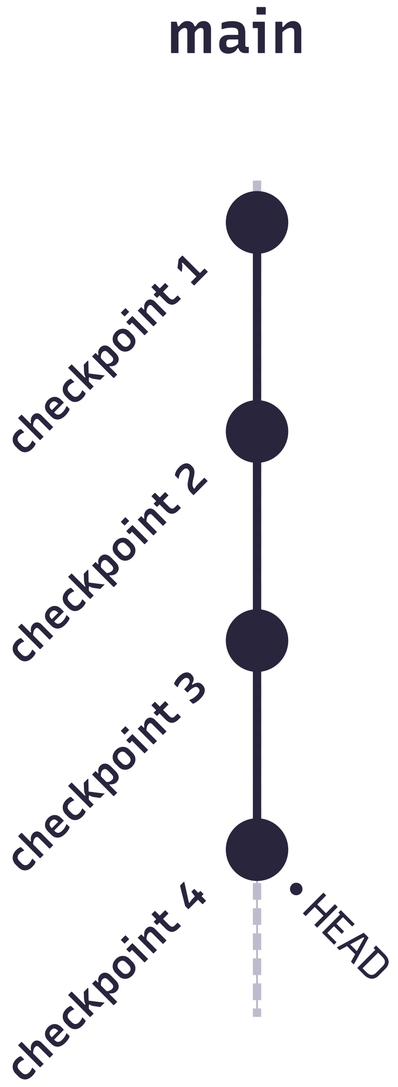
黑点代表 Commits，相当于“保存仓库某时刻的快照”

HEAD

告诉 Git，你现在处于哪个版本上工作?

Parent

之前的一系列快照，“父节点”



认识一些 Git 中的名词

Branch & Main Branch

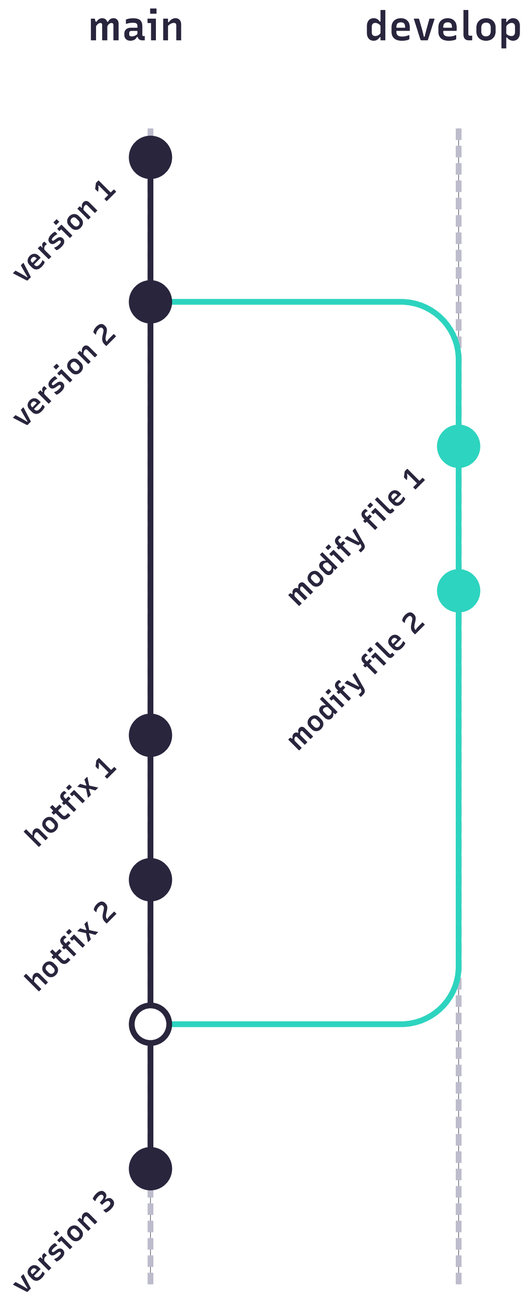
创建了一个分支，就相当于创建了一条独立的“时间线”

Merge

在分支上的工作结束后，合并到主线

Merge Commit

空心圆点；合并两条历史；有两个 PARENT



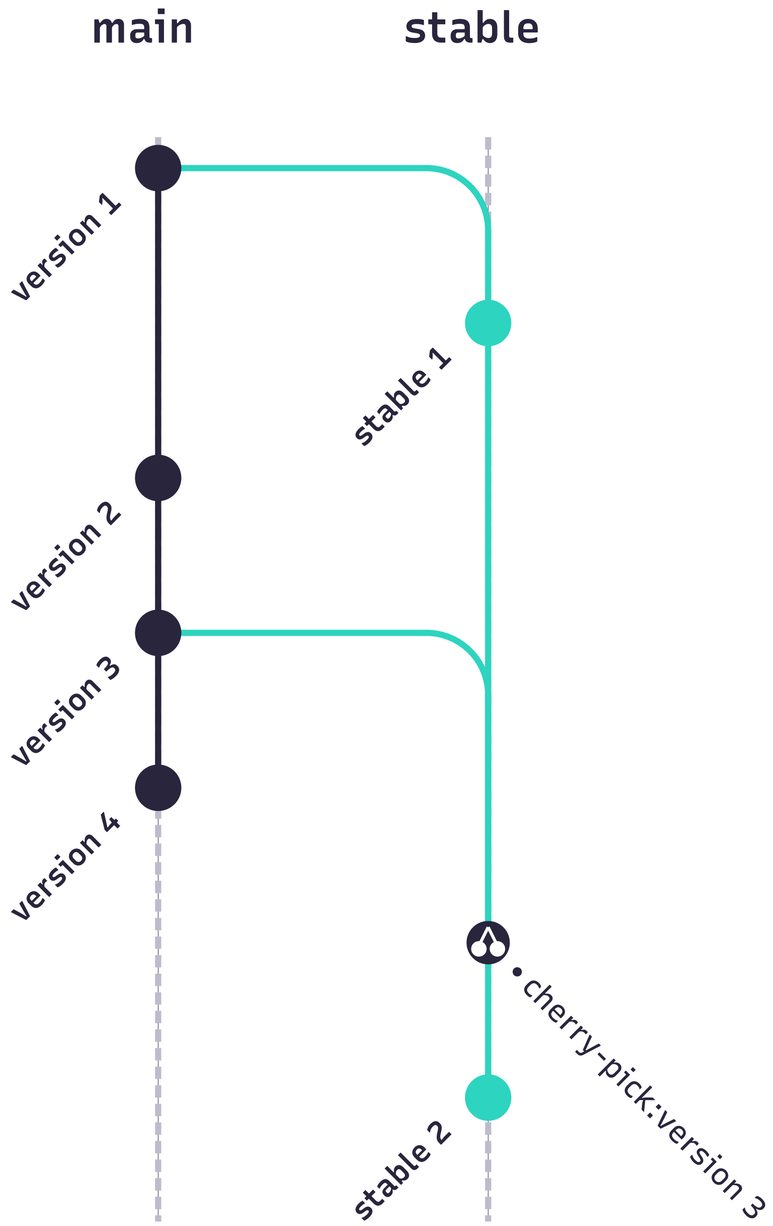
认识一些 Git 中的名词

Cherry-Pick

把某一个 Commit 的修改应用于所在的分支

Parallel Development

做版本与分支管理不是无用功，会在关键时候救你命



什么是 Git?

为什么要用 Git?

怎么用 Git?

什么是 Git?

为什么要用 Git?

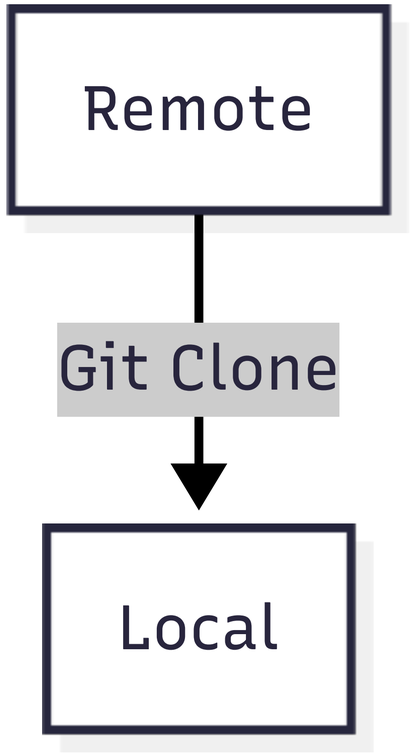
怎么用 Git?

- 这就是 Git。它通过一个优美的分布式图论树状模型来跟踪项目的协作。🤓
- **Cool!** 我们该怎么用它？😍
- 我也不知道。大概就是把几个 `shell` 命令背下来，然后输入它们。如果遇到报错，就把你的工作备份到别处，删掉整个项目，再重新下载一份新的。🤡

git clone

Clone an existing repo

> 克隆一个存在的仓库
> 下载包含所有历史的时光机



把几个 shell 命令背下来

你可能会用到的命令

`git clone url`

克隆仓库

`git clone url directory`

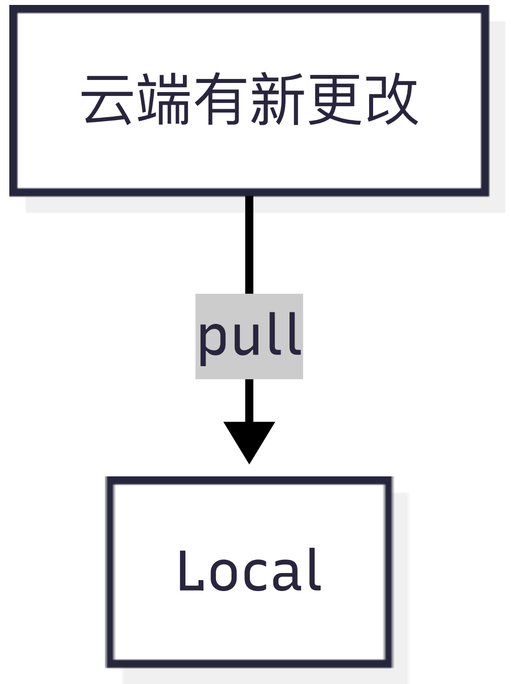
克隆仓库到指定目录

动手做一做：

git pull

Fetch remote, merge it with local

> 拉取更改并与本地合并
> 有人提交了新的代码!



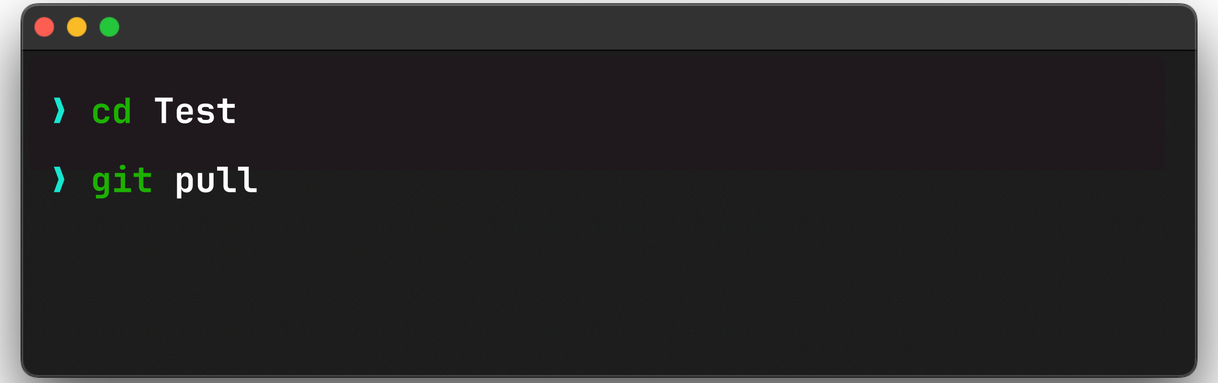
把几个 shell 命令背下来

你可能会用到的命令

`git pull`

拉取更改

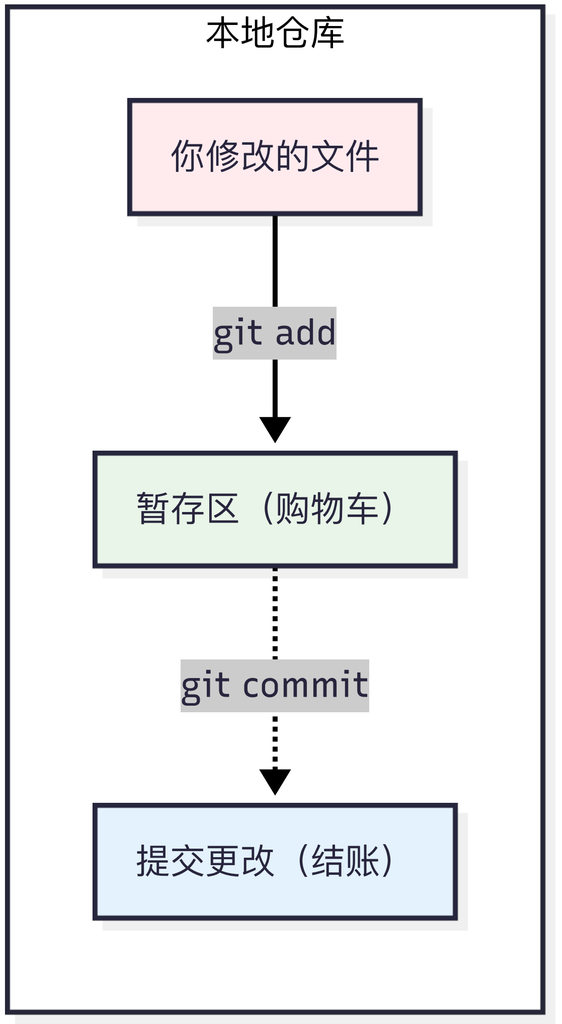
动手做一做：



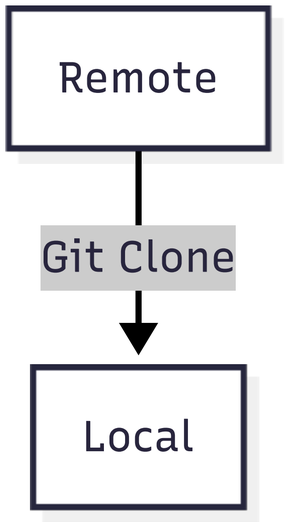
git add

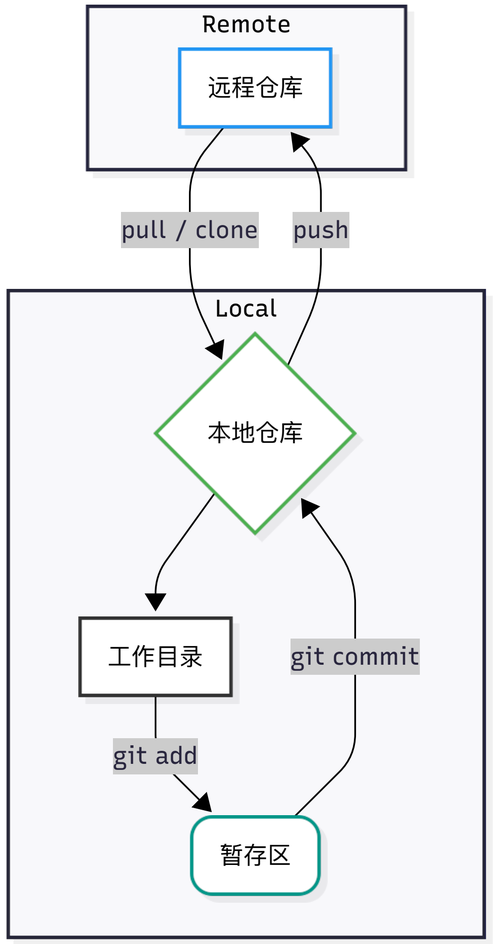
Adds changed files to the index

> 添加更改的文件至 Staging Area
> 结账之前放到购物车里 🛒



暂存区是什么 🤔





工作目录 📁：

- 正如名字所说，是一个你电脑上的目录
- 存储着你代码的文件夹

暂存区 🛒：

- 存储 准备在下一次提交中修改的内容
- 你可以选择你要提交什么更改

本地仓库 📦：

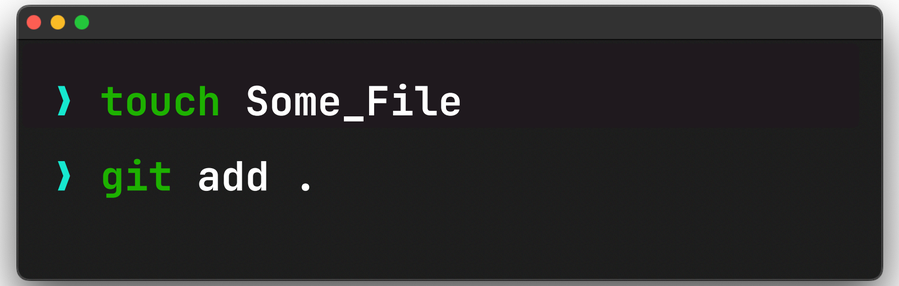
- 就是那个 .git 文件夹
- 它存储着你仓库的一切历史记录

把几个 shell 命令背下来

你可能会用到的命令

`git add file`	添加文件至暂存区
`git add .`	添加目录下所有文件至暂存区

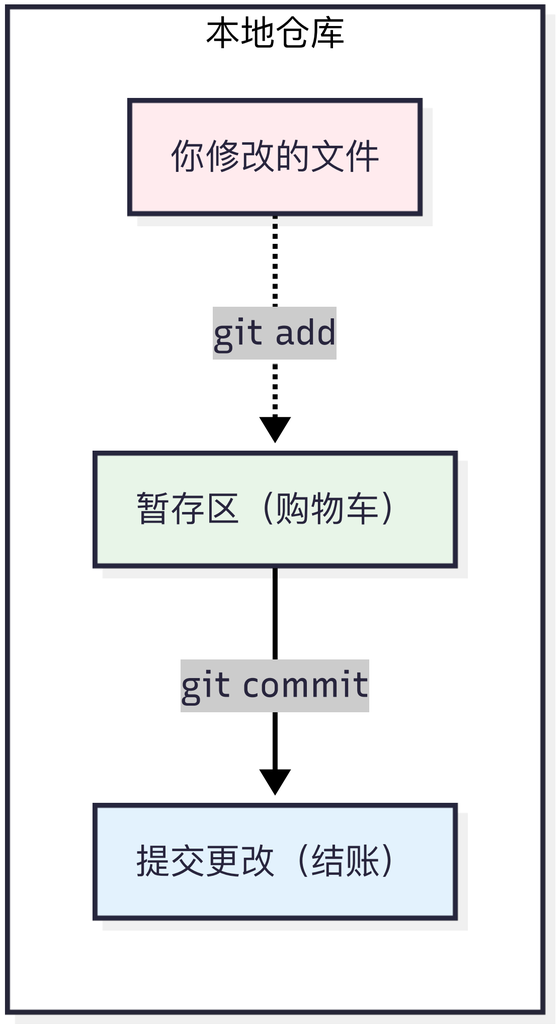
动手做一做：



git commit

Commit changes to local repo

> 提交更改至仓库中
> 你将更改的代码合并到仓库中（本地）



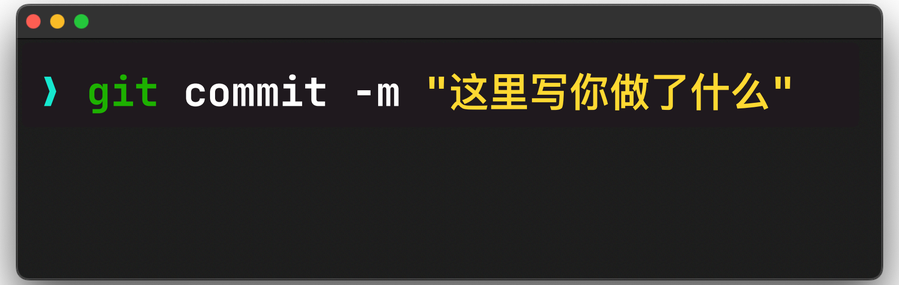
把几个 shell 命令背下来

你可能会用到的命令

```
git commit -m "message"
```

提交暂存区中的文件并附上描述

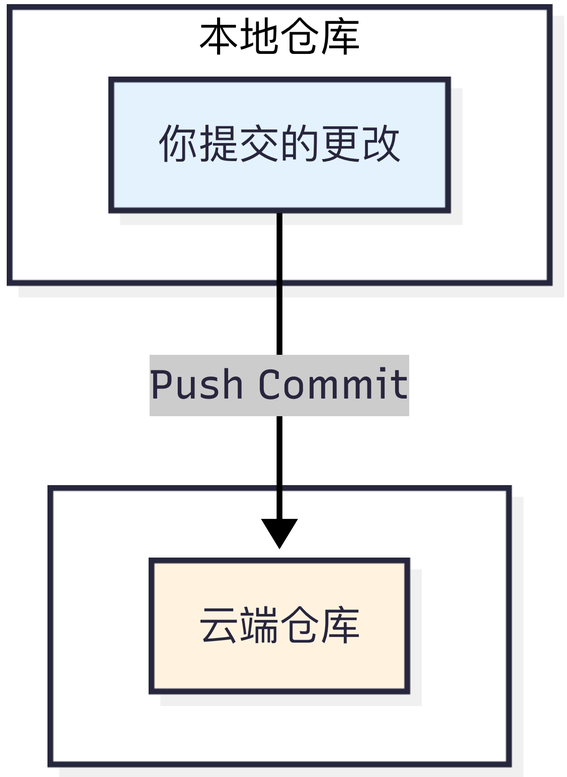
动手做一做：



git push

Push Commits to a remote repo

> 推送更改至远端仓库
> 把本地的更改上传（大家都能看见）



把几个 shell 命令背下来

你可能会用到的命令

`git push`

推送更改

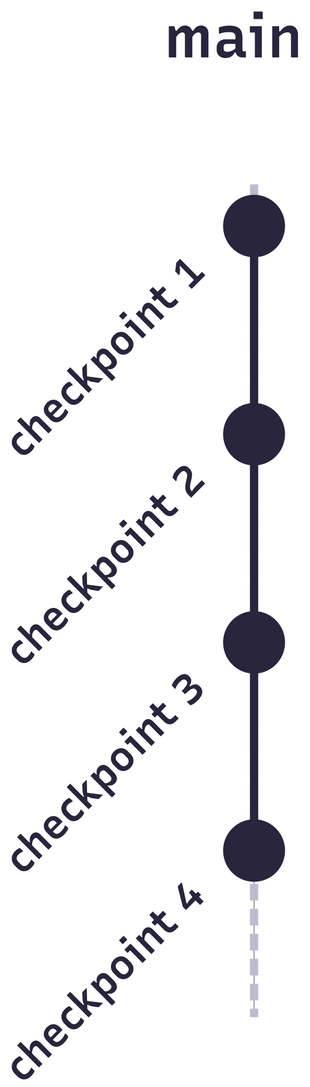
这次不动手实操了

你们也没我仓库的写入权限

git log

View the commit history of repo

> 检查仓库的历史提交
> 谁在什么时候做了什么 👀



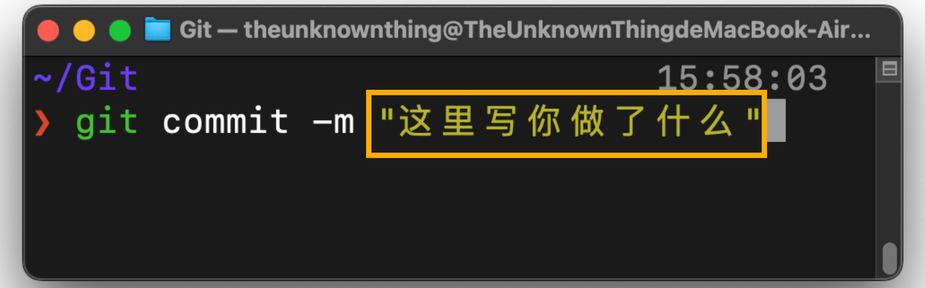
把几个 shell 命令背下来

你可能会用到的命令

`git log`	查看提交历史（时间顺序）
`git log --stat`	查看具体更改了哪些文件

动手做一做：

Git Conventions

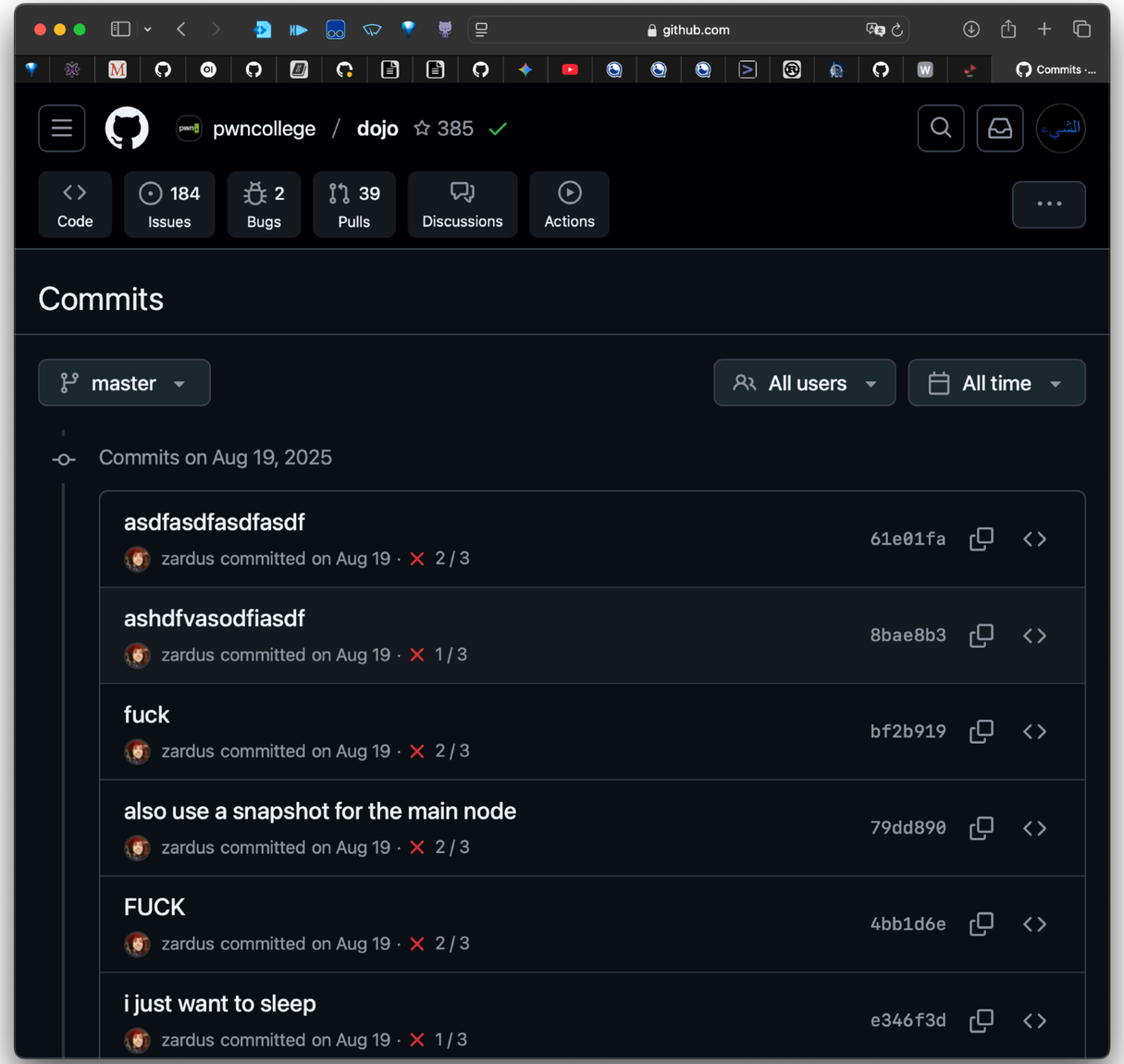


Commit Message 不许发疯!

坏的: *Fk; ASDF; LOL;***

好的: feat: add Polish language

www.conventionalcommits.org



Conventional Commits

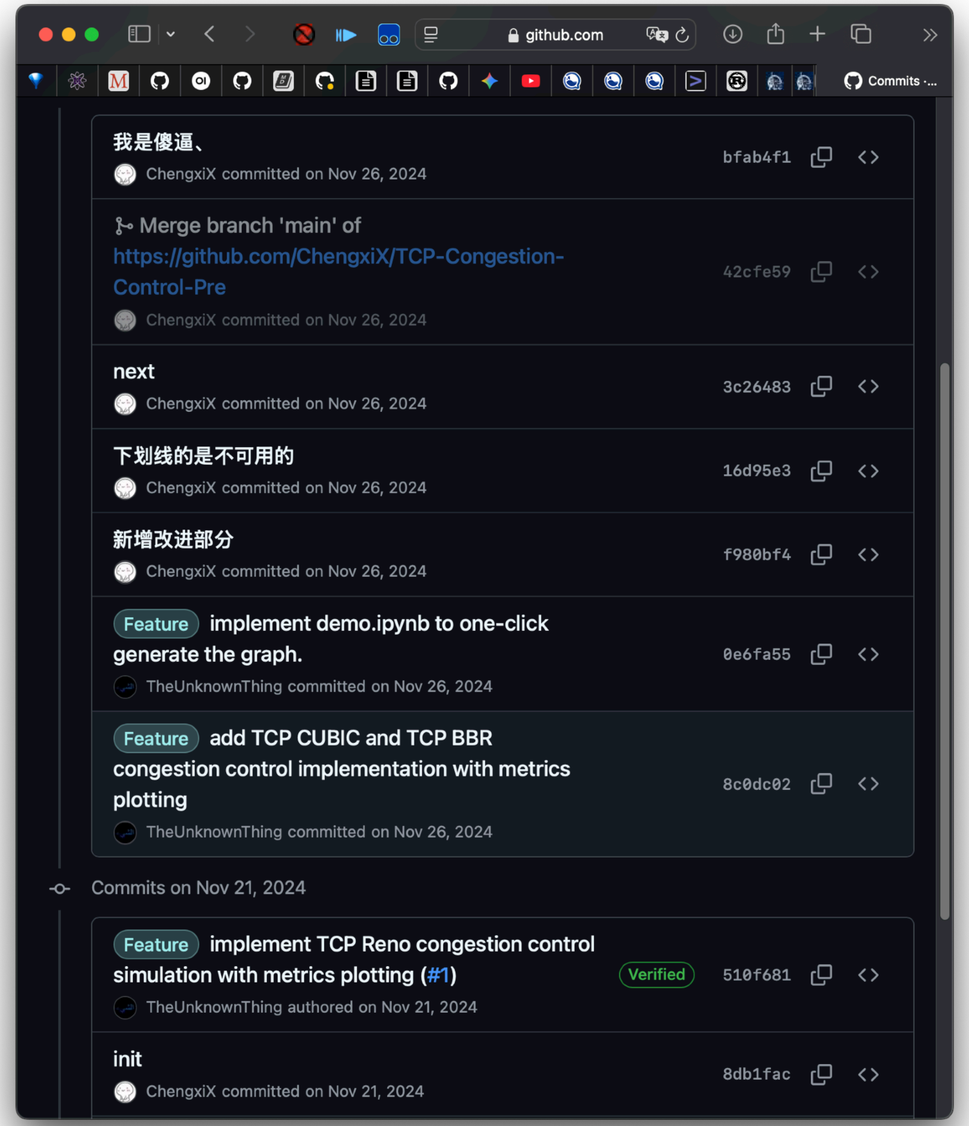
```
<type>[optional scope]: <description>

[optional body]

[optional footer(s)]
```

👍 Examples

```
docs: correct spelling of CHANGELOG
feat(lang): add Polish language
```



Git Ignore

包含密钥的文件

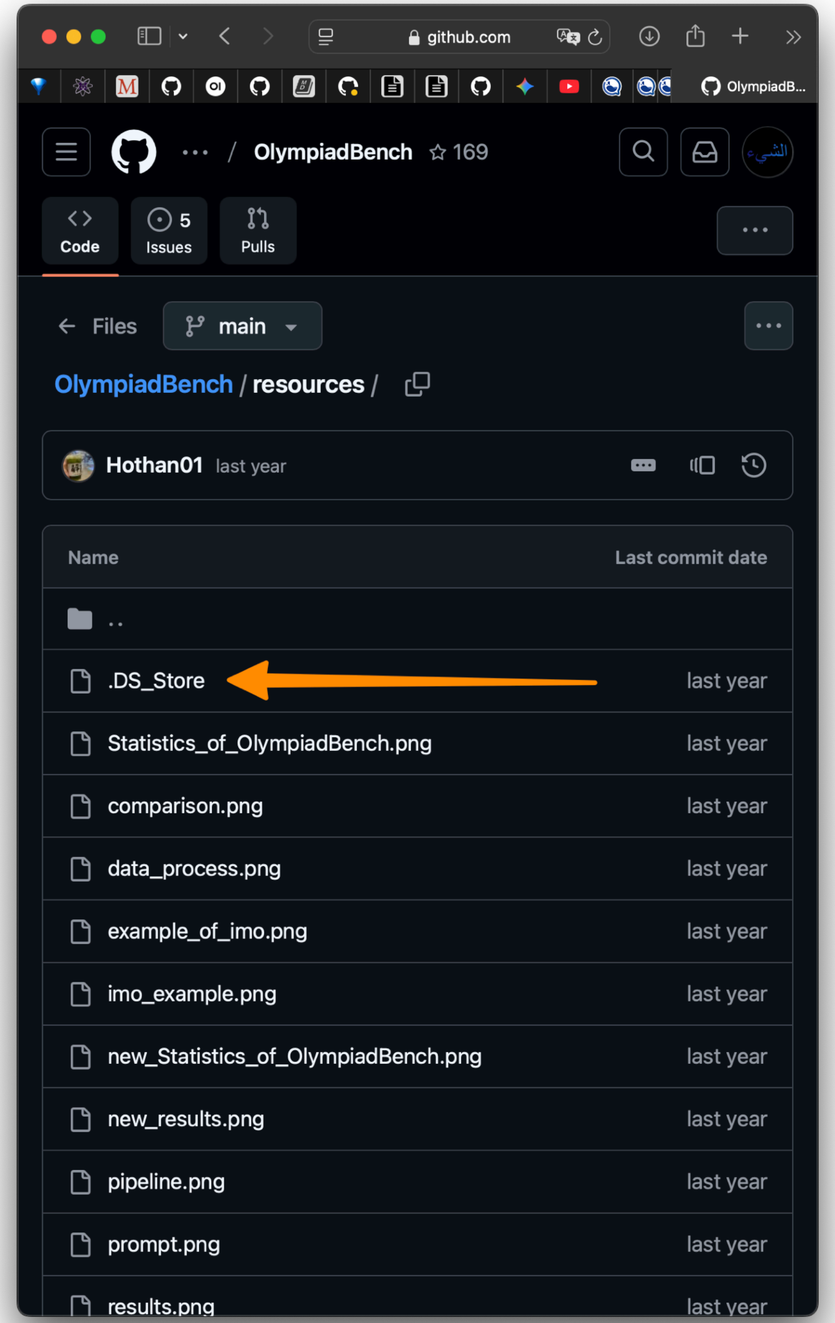
macOS 会拉屎 💩

代码的编译会产生很多文件

……

不想把他们传到仓库中，难道每次要手动删除?

`.gitignore`



```
# Build directories
build/
bin/
lib/

# CMake generated files
CMakeCache.txt
CMakeFiles/
cmake_install.cmake
Makefile

# Object files
*.o
*.obj

# Executables
*.exe
*.out
rcompiler
unit_tests
integration_tests
```

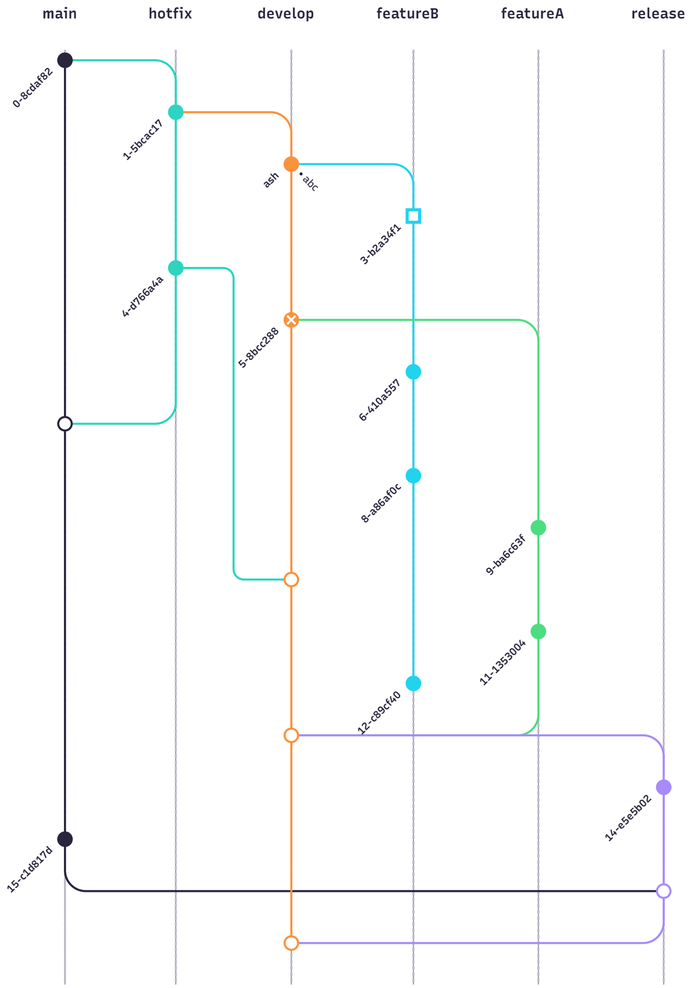
编译目录

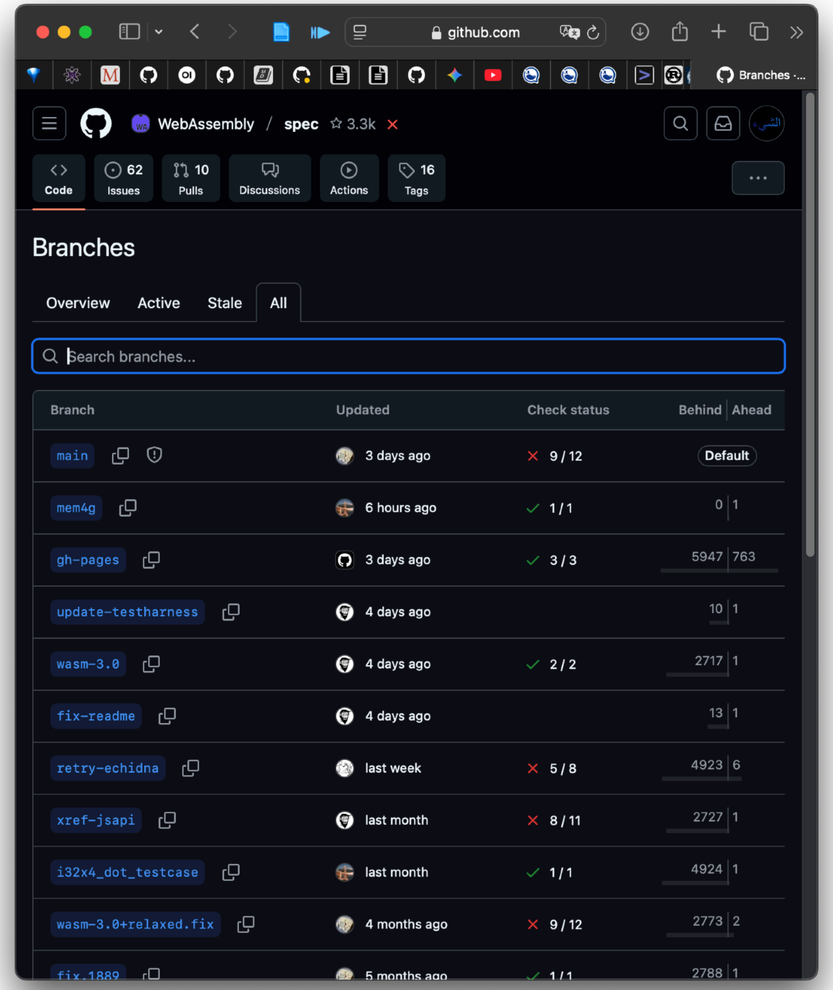
CMake 生成的文件

库文件

可执行文件

Branches


main
hotfix
develop
featureB
featureA
release
0-8cdaf82
1-5bcac17
a8h
a1c
3-b2a34f1
4-d766a4a
5-8bcc288
6-410a557
8-a86af0c
9-ba6c63f
11-1353004
12-c89cf40
14-e5e5b02
15-c1d817d


github.com
Branches ·...
WebAssembly / spec ☆ 3.3k
Code
62 Issues
10 Pulls
Discussions
Actions
16 Tags
Branches
Overview
Active
Stale
All
Search branches...
Branch
Updated
Check status
Behind | Ahead
main 3 days ago 9 / 12 Default
mem4g 6 hours ago 1 / 1 0 | 1
gh-pages 3 days ago 3 / 3 5947 | 763
update-testharness 4 days ago 10 | 1
wasm-3.0 4 days ago 2 / 2 2717 | 1
fix-readme 4 days ago 13 | 1
retry-echidna last week 5 / 8 4923 | 6
xref-jsapi last month 8 / 11 2727 | 1
i32x4_dot_testcase last month 1 / 1 4924 | 1
wasm-3.0+relaxed.fix 4 months ago 9 / 12 2773 | 2
fix.1889 5 months ago 1 / 1 2788 | 1

git checkout

Navigating between different versions

切换时光机的检查点

```
git checkout <branch> # 切换已有分支
git checkout -b <branch> # 新建并切换分支
git checkout <commit_hash> # 查看该提交时仓库的状态
```

git branch

Managing your branches

管理分支的主要命令

```
git branch -a # 显示所有分支
git branch --show-current # 显示当前分支
git branch <name> # 在当前提交基础上新建分支
git branch <name> <commit_hash> # 在制定提交基础上新建分支
```

git 1 of 2 noun

'git

Synonyms of *git* >

British

: a foolish or worthless person

Noun

That ***git*** of a brother of yours has ruined everything!

> oh, don't be such a silly ***git***, of course your mates want you around

https://www.merriam-webster.com/dictionary/git

Oh Shit, Git!?!

https://ohshitgit.com/

Oh Shit, Git!?!

Git is hard: screwing up is easy, and figuring out how to fix your mistakes is fucking impossible. Git documentation has this chicken and egg problem where you can't search for how to get yourself out of a mess, *unless you already know the name of the thing you need to know about* in order to fix your problem.

So here are some bad situations I've gotten myself into, and how I eventually got myself out of them *in plain english*.

Oh shit, I did something terribly wrong, please tell me git has a magic time machine!?!

```
git reflog
# you will see a list of every thing you've
# done in git, across all branches!
# each one has an index HEAD@{index}
# find the one before you broke everything
git reset HEAD@{index}
# magic time machine
```

You can use this to get back stuff you accidentally deleted, or just to remove some stuff you tried that broke the repo, or to recover after a bad merge, or just to go back to a time when things actually worked. I use `reflog` A LOT. Mega hat tip to the many many many many many people who suggested adding it!

Oh shit, I committed and immediately realized I need to make one small change!

```
# make your change
git add . # or add individual files
git commit --amend --no-edit
# now your last commit contains that change!
# WARNING: never amend public commits
```

Oh Shit, Git!?!

用好 Git 很难: 很容易就犯错了，然后想自己弥补犯下的错，简直太难了。查阅 Git 文档简直就像是个 “鸡生蛋 蛋生鸡” 的问题，*你得知道你要的是啥*，但如果我知道的话，我还他妈查个毛文档啊！

所以接下来我会分享一些我遇到过的抓狂的经历，然后用 *白话* 来说说我是如何解决的。

哎呦我去，我刚才好像犯了个大错，能不能给我台时光机啊!?!

```
git reflog
# 你将看到你在 git 上提交的所有改动记录被列
# 了出来，而且囊括了所有的分支，和已被删除的
# commit 哦！
# 每一条记录都有一个类似 HEAD@{index} 的索
# 引编号
# 找到在犯错前的那个提交记录的索引号，然后执
# 行：
git reset HEAD@{index}
# 哈哈，这就是你要的时光机！
```

你可以用这个方法来找回那些你不小心删除的东西、恢复一些你对 repo 改动、恢复一次错误的 merge 操作、或者仅仅想退回到你的项目还能正常工作的那一时刻。我经常使用 `reflog`，在此我要向那些提案添加这个功能的人们表示感谢，太谢谢他们了！

哎呦我去，我刚提交 commit 就发现还有一个小改动需要添加！

```
# 继续改动你的文件
git add . # 或者你可以添加指定的文件
git commit --amend --no-edit
# 你这次的改动会被添加进最近一次的 commit 中
# 警告：千万别对公共的 commit 做这种操作
```

怎么用 Git?

什么是 Git?

为什么要用 Git?

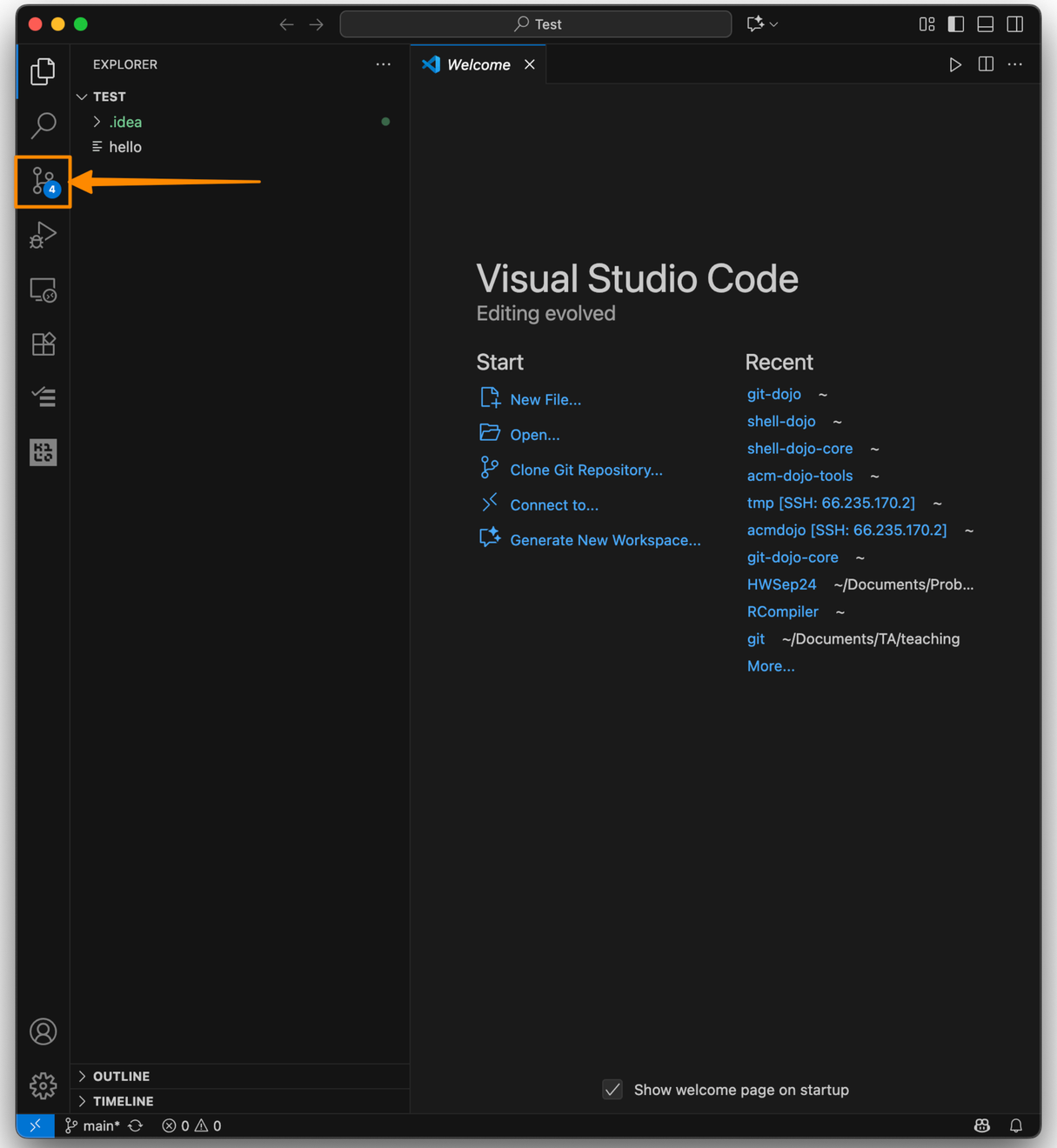
图形化操作 Git

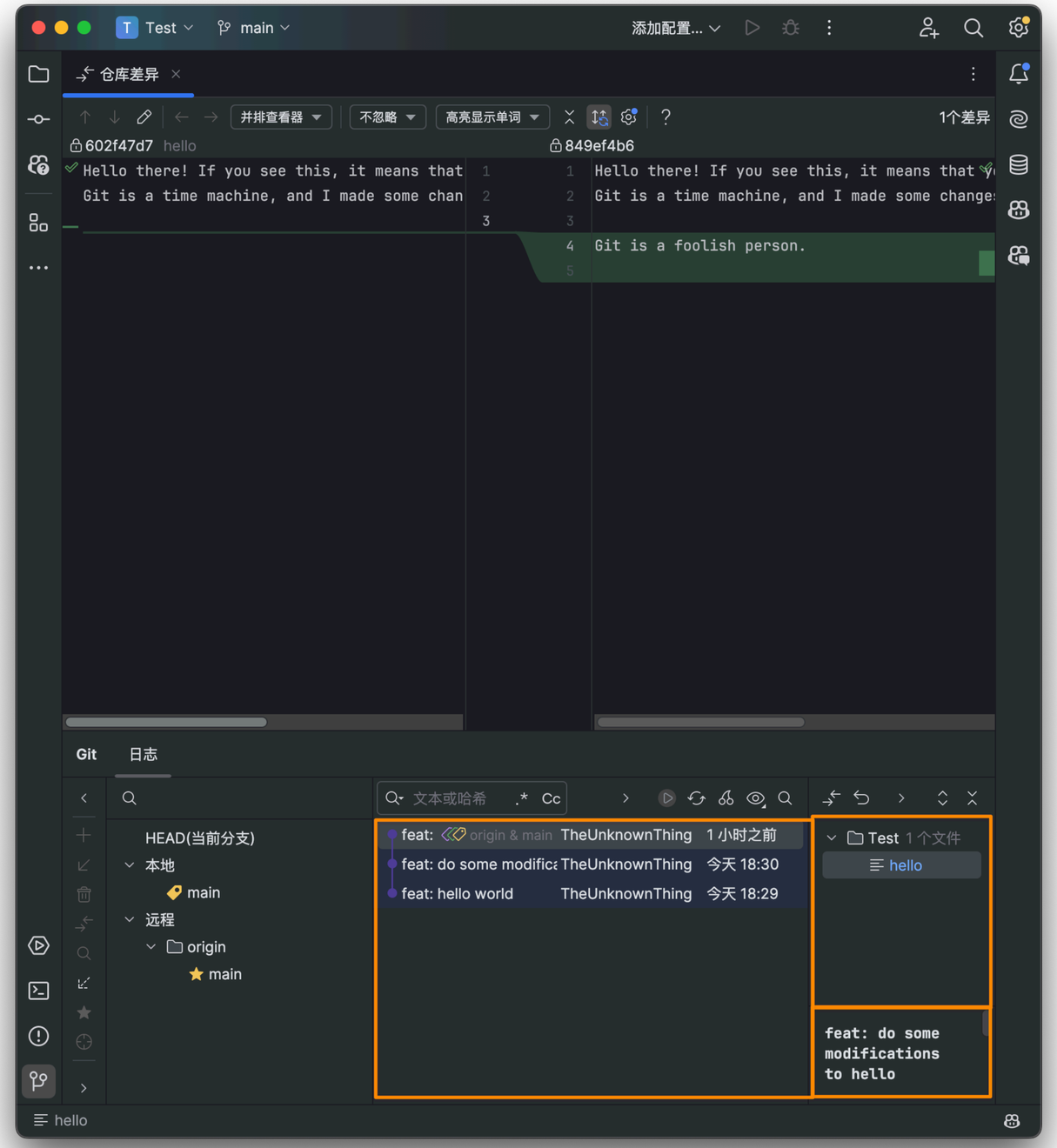
怎么用 Git?

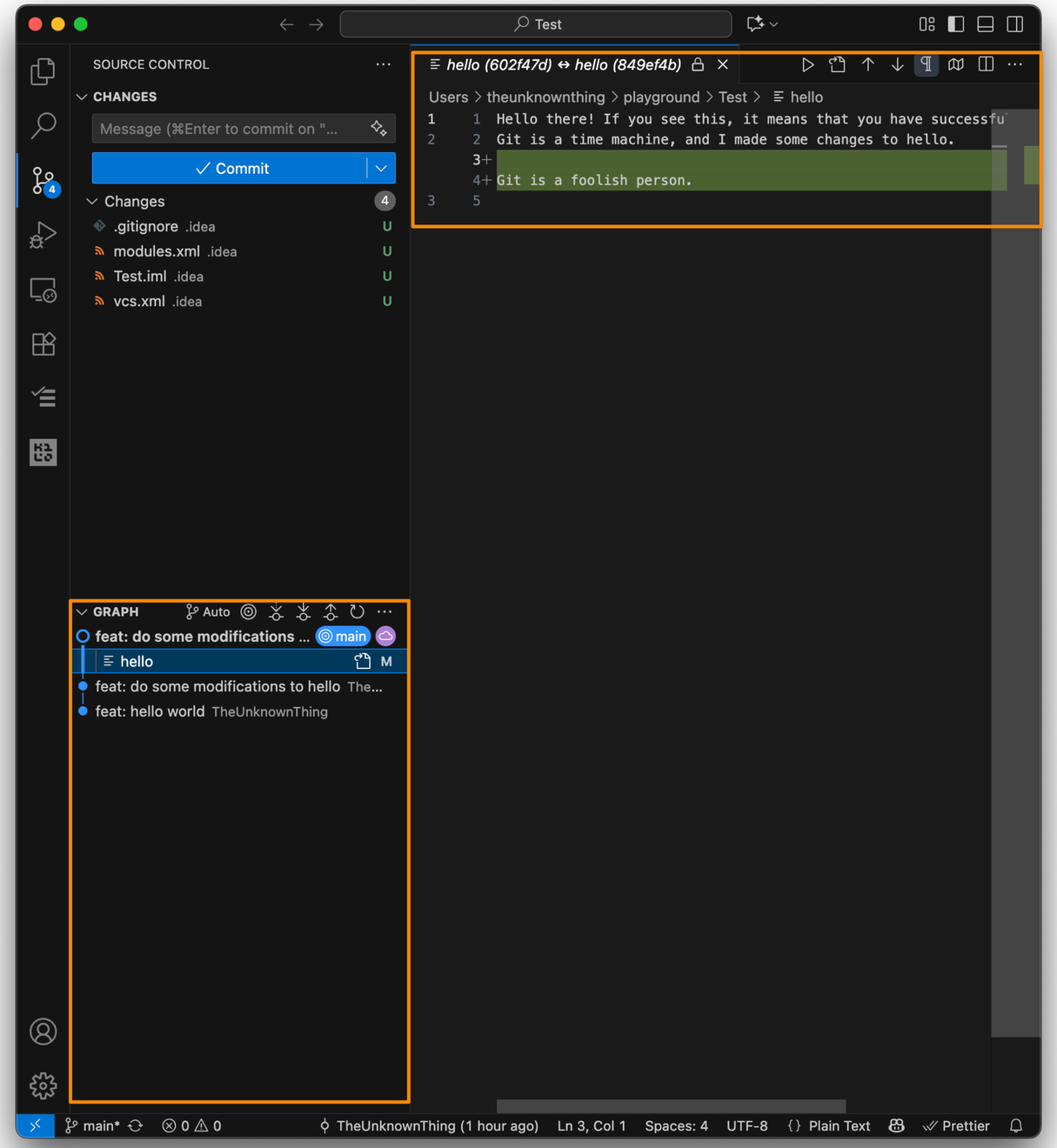
什么是 Git?

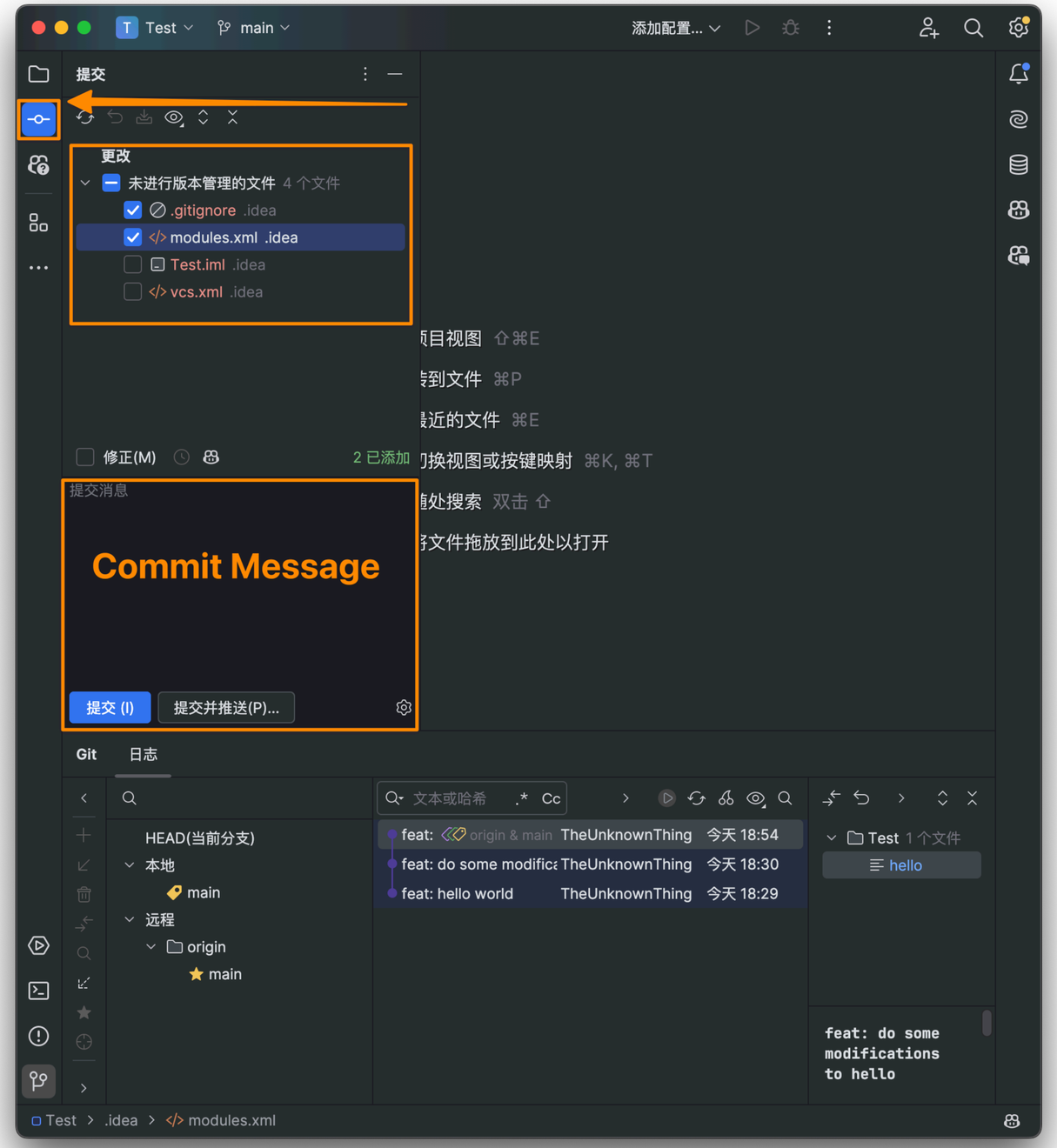
为什么要用 Git?

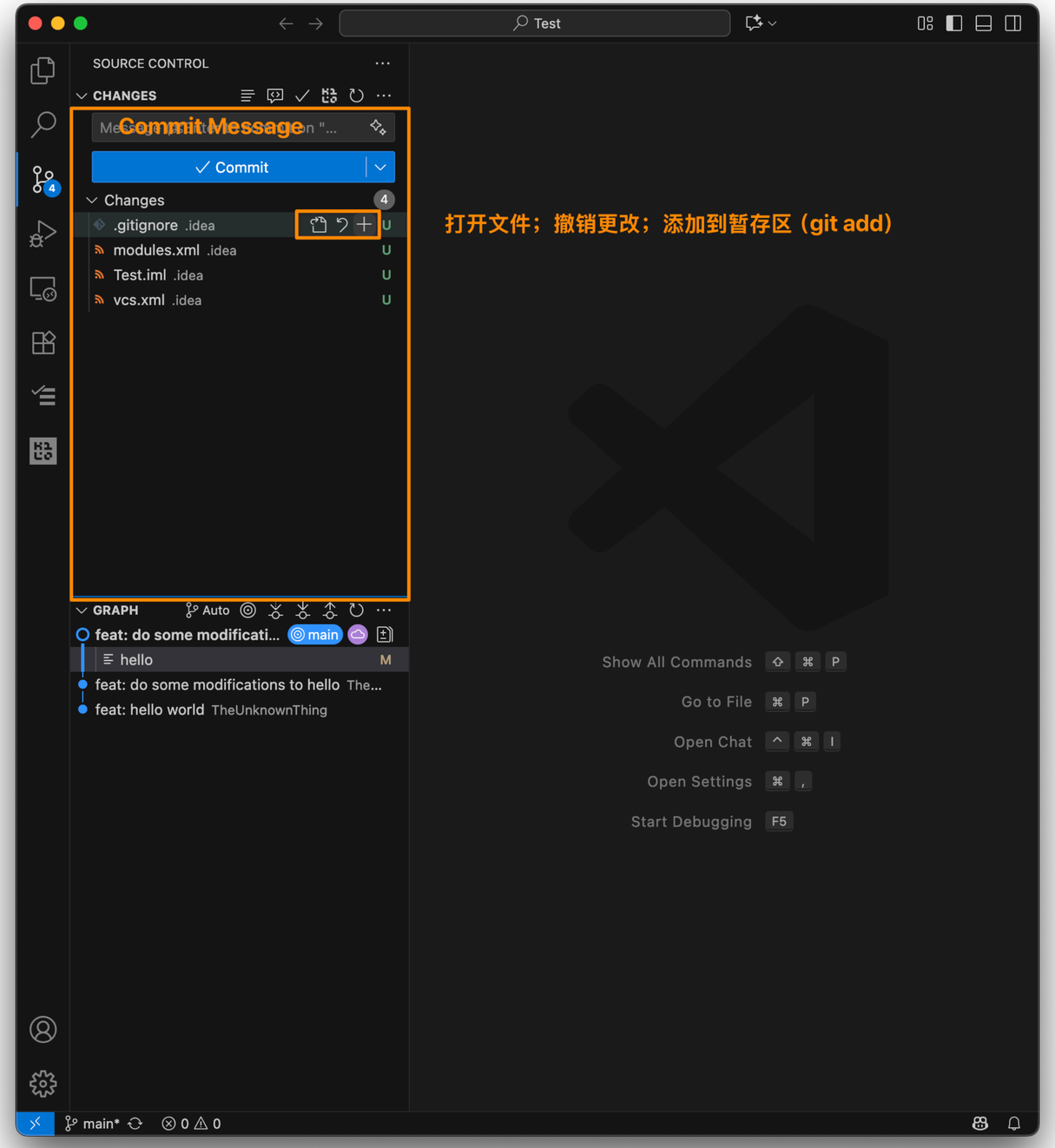

Test
main
添加配置...
项目视图 ⇧⌘E
转到文件 ⌘P
最近的文件 ⌘E
切换视图或按键映射 ⌘K, ⌘T
随处搜索 双击 ⇧
将文件拖放到此处以打开
Test

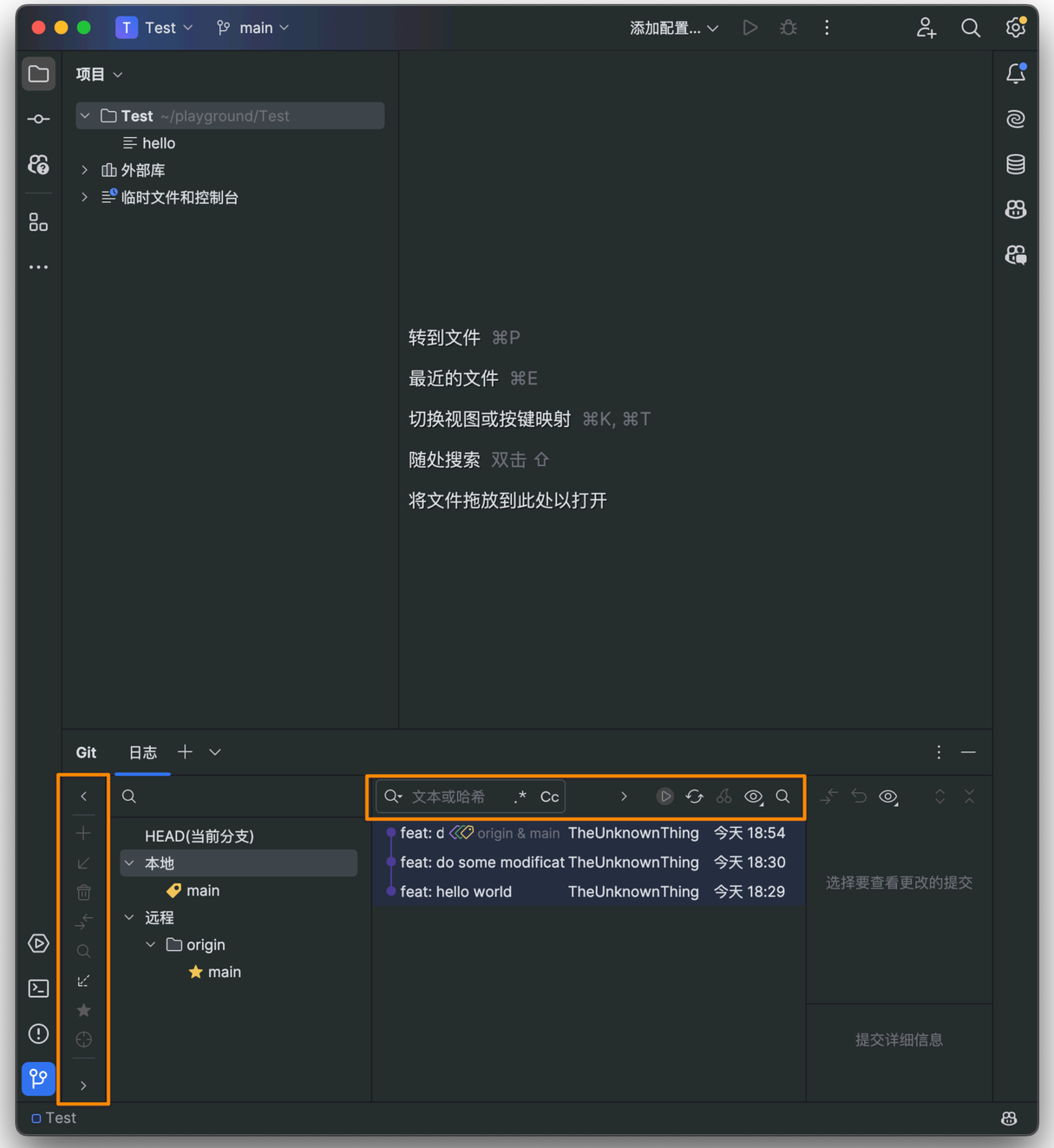

Test
EXPLORER
TEST
.idea
hello
Welcome
Visual Studio Code
Editing evolved
Start
New File...
Open...
Clone Git Repository...
Connect to...
Generate New Workspace...
Recent
git-dojo ~
shell-dojo ~
shell-dojo-core ~
acm-dojo-tools ~
tmp [SSH: 66.235.170.2] ~
acmdojo [SSH: 66.235.170.2] ~
git-dojo-core ~
HWSep24 ~/Documents/Prob...
RCompiler ~
git ~/Documents/TA/teaching
More...
Show welcome page on startup
OUTLINE
TIMELINE
main*
0 0

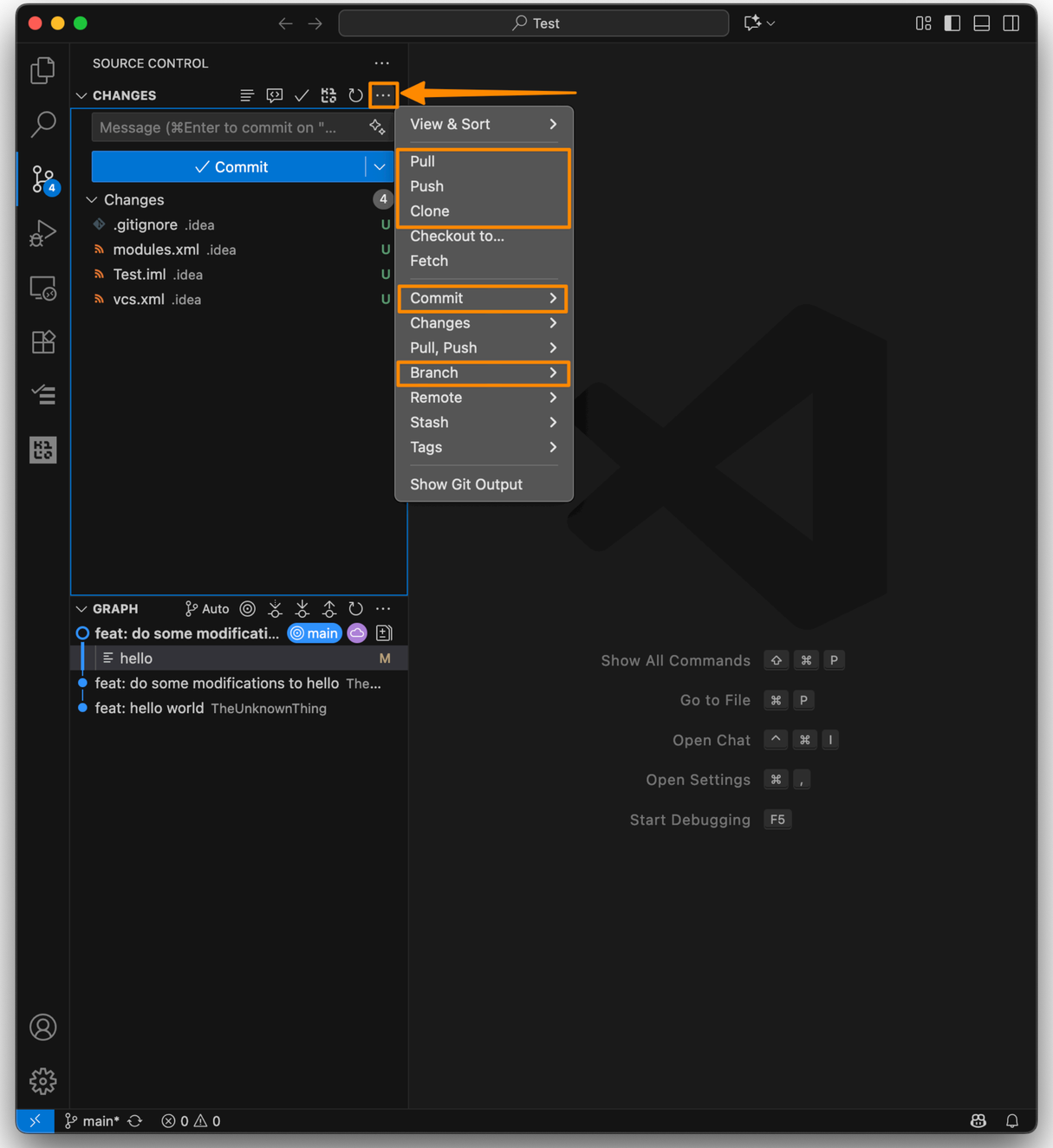

Test
main
添加配置...
仓库差异
并排查看器
不忽略
高亮显示单词
1个差异
602f47d7 hello
849ef4b6
Hello there! If you see this, it means that
Git is a time machine, and I made some chan
Hello there! If you see this, it means that
Git is a time machine, and I made some change
Git is a foolish person.
Git
日志
文本或哈希
HEAD(当前分支)
本地
main
远程
origin
main
feat: origin & main TheUnknownThing 1 小时之前
feat: do some modific TheUnknownThing 今天 18:30
feat: hello world TheUnknownThing 今天 18:29
Test 1 个文件
hello
feat: do some modifications to hello
hello


Test
SOURCE CONTROL
CHANGES
Message (⌘Enter to commit on "...
Commit
Changes 4
.gitignore .idea U
modules.xml .idea U
Test.iml .idea U
vcs.xml .idea U
hello (602f47d) ↔ hello (849ef4b)
Users > theunknownthing > playground > Test > hello
Hello there! If you see this, it means that you have successfu
Git is a time machine, and I made some changes to hello.
Git is a foolish person.
GRAPH
Auto
feat: do some modifications ... main
hello M
feat: do some modifications to hello The...
feat: hello world TheUnknownThing
main*
TheUnknownThing (1 hour ago)
Ln 3, Col 1
Spaces: 4
UTF-8
Plain Text
Prettier


Test
main
添加配置...
提交
更改
未进行版本管理的文件 4 个文件
.gitignore .idea
modules.xml .idea
Test.iml .idea
vcs.xml .idea
修正(M)
2 已添加
提交消息
Commit Message
提交 (I)
提交并推送(P)...
Git
日志
HEAD(当前分支)
本地
main
远程
origin
main
feat: origin & main TheUnknownThing 今天 18:54
feat: do some modific TheUnknownThing 今天 18:30
feat: hello world TheUnknownThing 今天 18:29
Test 1 个文件
hello
feat: do some modifications to hello
Test > .idea > modules.xml


SOURCE CONTROL
CHANGES
Commit Message
Commit
Changes 4
.gitignore .idea U
modules.xml .idea U
Test.iml .idea U
vcs.xml .idea U
GRAPH
Auto
feat: do some modificati... main
hello M
feat: do some modifications to hello The...
feat: hello world TheUnknownThing
main*
0 0
打开文件；撤销更改；添加到暂存区（git add）
Show All Commands
Go to File
Open Chat
Open Settings
Start Debugging
F5


Test
main
添加配置...
项目
Test ~/playground/Test
hello
外部库
临时文件和控制台
转到文件 ⌘P
最近的文件 ⌘E
切换视图或按键映射 ⌘K, ⌘T
随处搜索 双击 ⇧
将文件拖放到此处以打开
Git
日志
文本或哈希
HEAD(当前分支)
本地
main
远程
origin
main
feat: d origin & main TheUnknownThing 今天 18:54
feat: do some modificat TheUnknownThing 今天 18:30
feat: hello world TheUnknownThing 今天 18:29
选择要查看更改的提交
提交详细信息
Test


Test
SOURCE CONTROL
CHANGES
Message (⌘Enter to commit on "...
Commit
Changes 4
.gitignore .idea U
modules.xml .idea U
Test.iml .idea U
vcs.xml .idea U
View & Sort
Pull
Push
Clone
Checkout to...
Fetch
Commit
Changes
Pull, Push
Branch
Remote
Stash
Tags
Show Git Output
GRAPH
Auto
feat: do some modificati... main
hello M
feat: do some modifications to hello The...
feat: hello world TheUnknownThing
Show All Commands ⇧ ⌘ P
Go to File ⌘ P
Open Chat ⌃ ⌘ I
Open Settings ⌘ ,
Start Debugging F5
main*
0 0

“不要让使用 Git 成为你的负担。养成版本管理的好习惯非常重要，况且这不难。”

我 · 自己说的

图形化操作 Git

怎么用 Git?

什么是 Git?

为什么要用 Git?

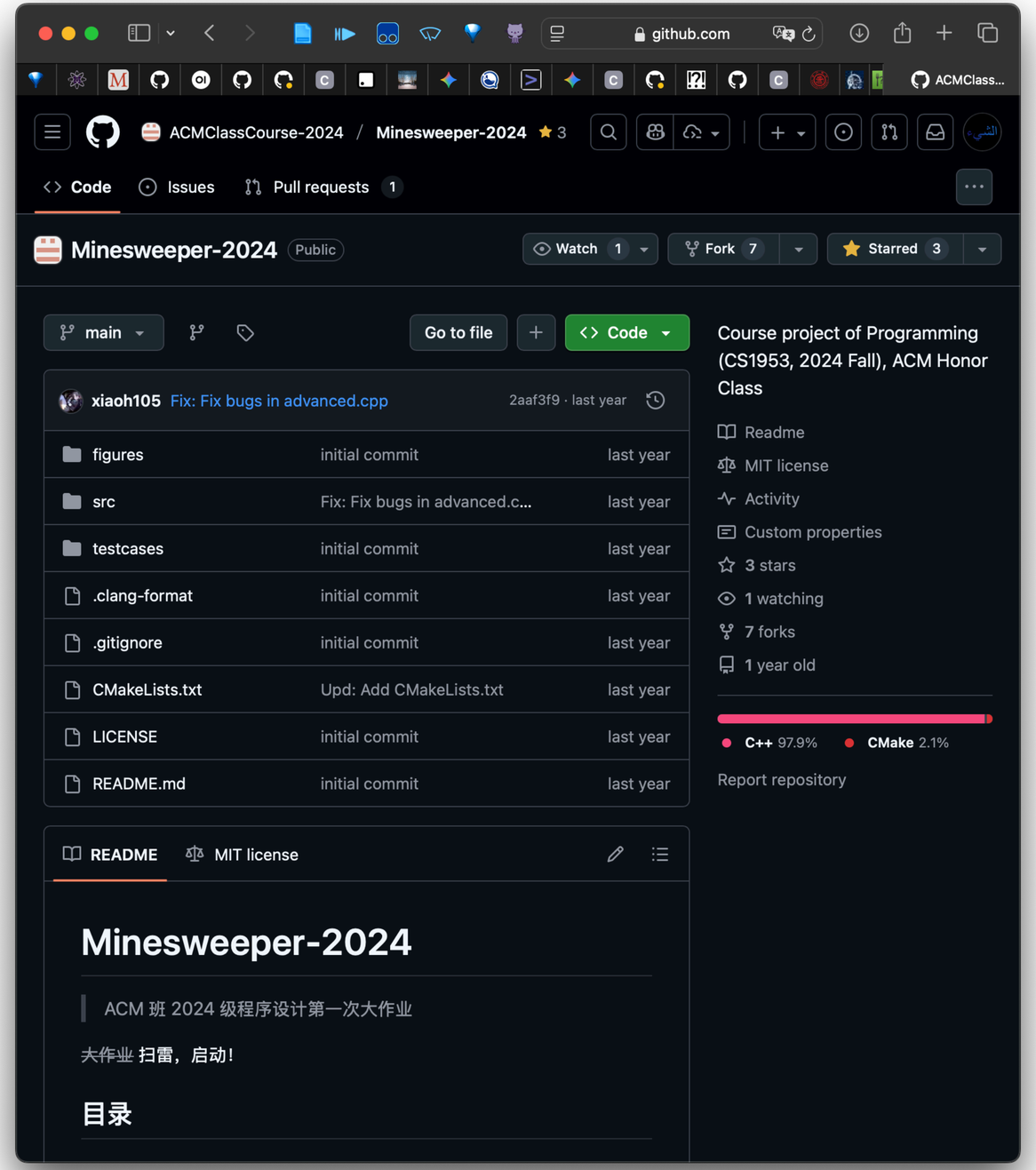
GitHub

图形化操作 Git

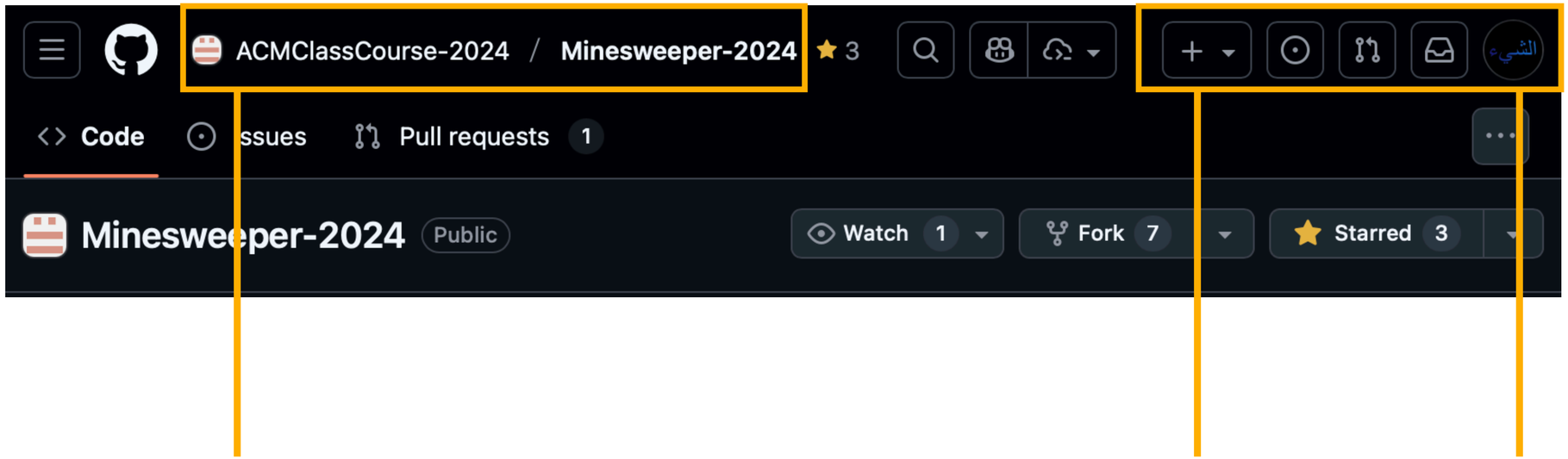
怎么用 Git?

什么是 Git?

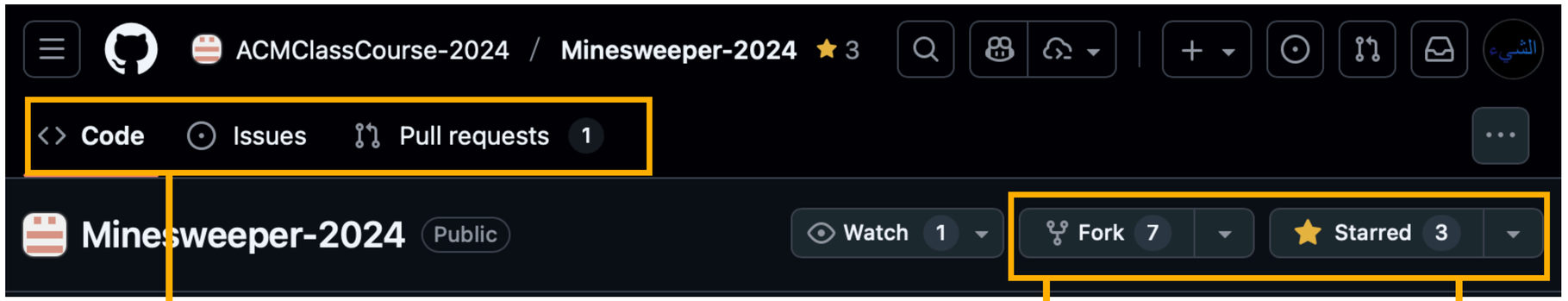
为什么要用 Git?


github.com
ACMClass...
ACMClassCourse-2024 / Minesweeper-2024 3
Code
Issues
Pull requests 1
Minesweeper-2024 Public
Watch 1
Fork 7
Starred 3
main
Go to file
Code
xiaoh105 Fix: Fix bugs in advanced.cpp 2aaf3f9 · last year
figures initial commit last year
src Fix: Fix bugs in advanced.c... last year
testcases initial commit last year
.clang-format initial commit last year
.gitignore initial commit last year
CMakeLists.txt Upd: Add CMakeLists.txt last year
LICENSE initial commit last year
README.md initial commit last year
README
MIT license
Minesweeper-2024
ACM 班 2024 级程序设计第一次大作业
~~大作业~~ 扫雷，启动!
目录
Course project of Programming (CS1953, 2024 Fall), ACM Honor Class
Readme
MIT license
Activity
Custom properties
3 stars
1 watching
7 forks
1 year old
C++ 97.9%
CMake 2.1%
Report repository

GitHub 上你会使用...



GitHub 上你会使用...



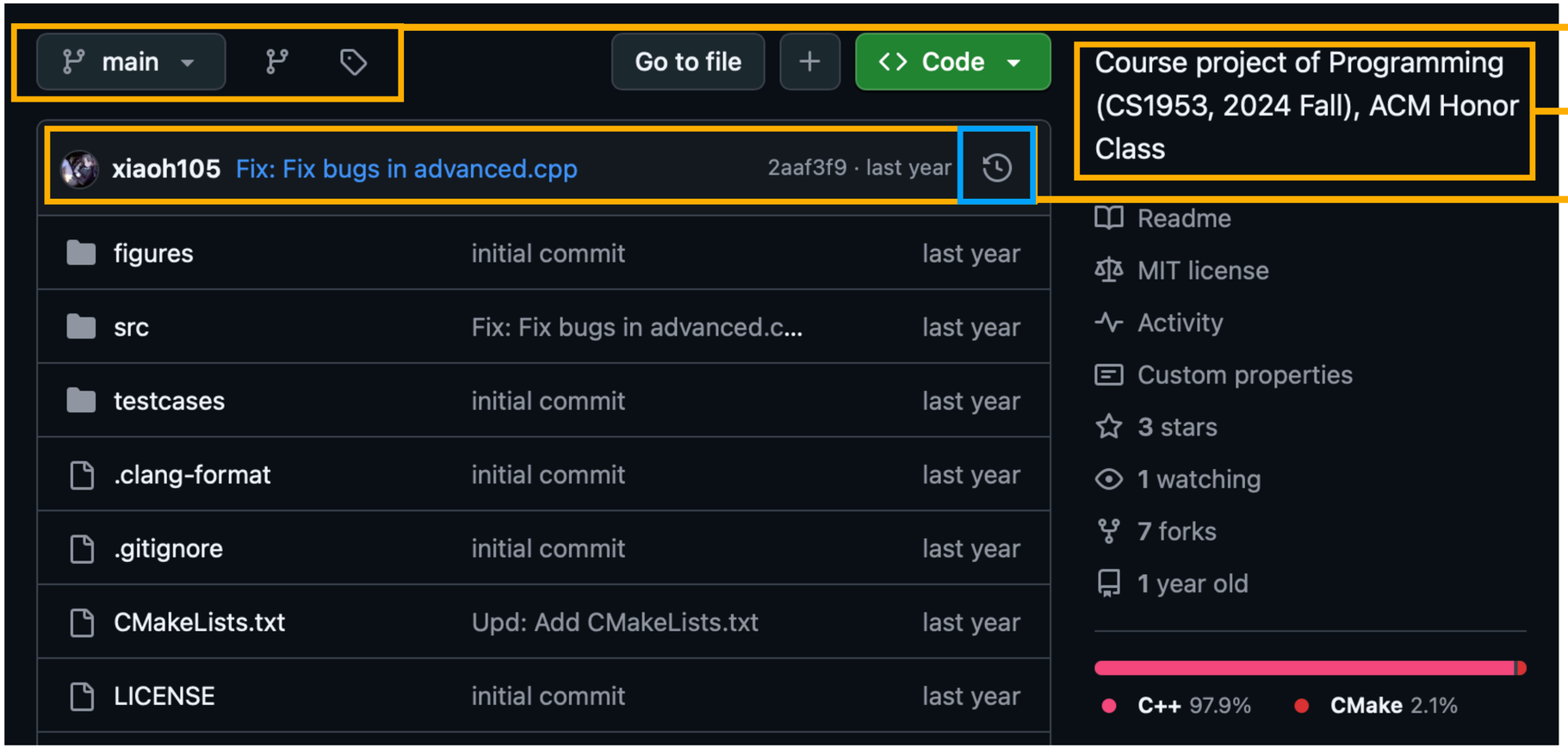
代码 / 议题 / 拉取请求

Fork 仓库

在你的账户中创建仓库副本

小星星

GitHub 上你会使用...



当前分支

仓库描述

提交记录

GitHub 上你会使用...

README

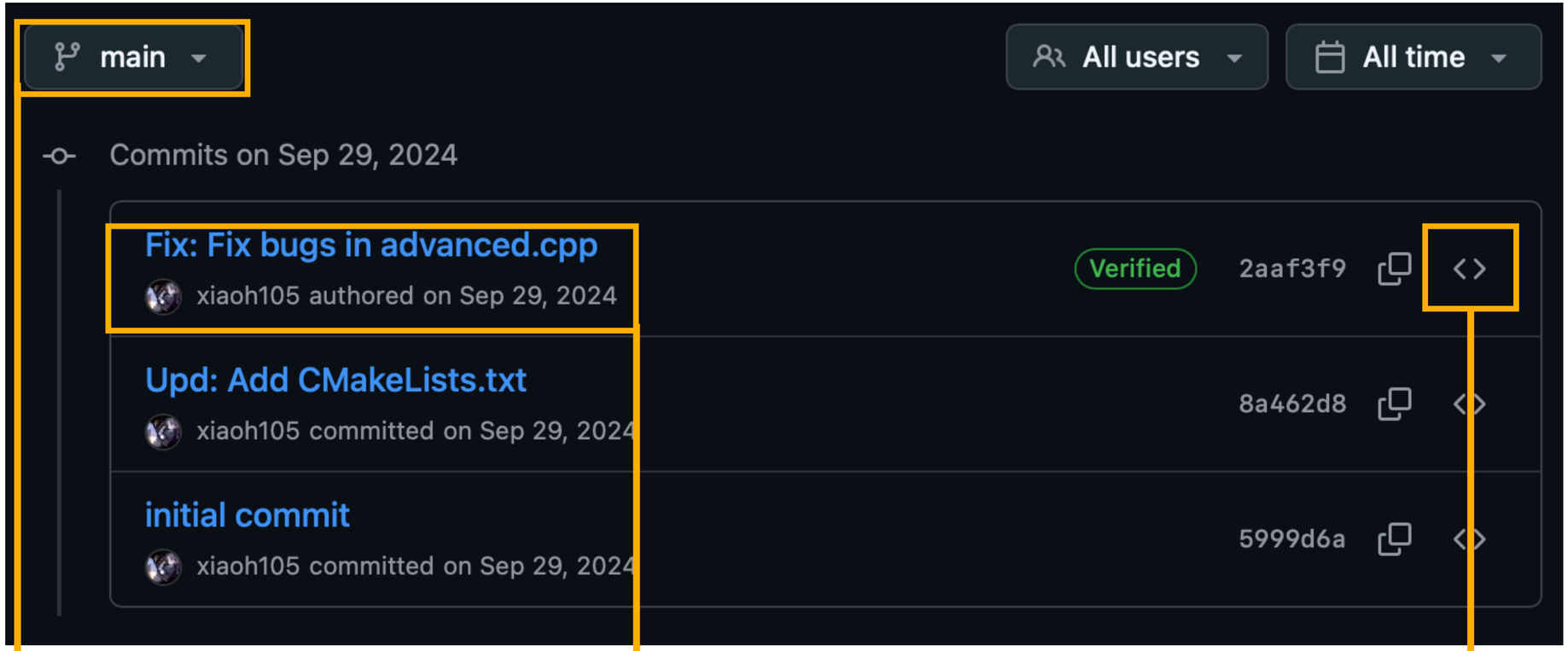
显示 README.md

许可证

如果你还不会 Markdown 的话:

https://www.markdownguide.org/cheat-sheet/

GitHub 上你会使用...

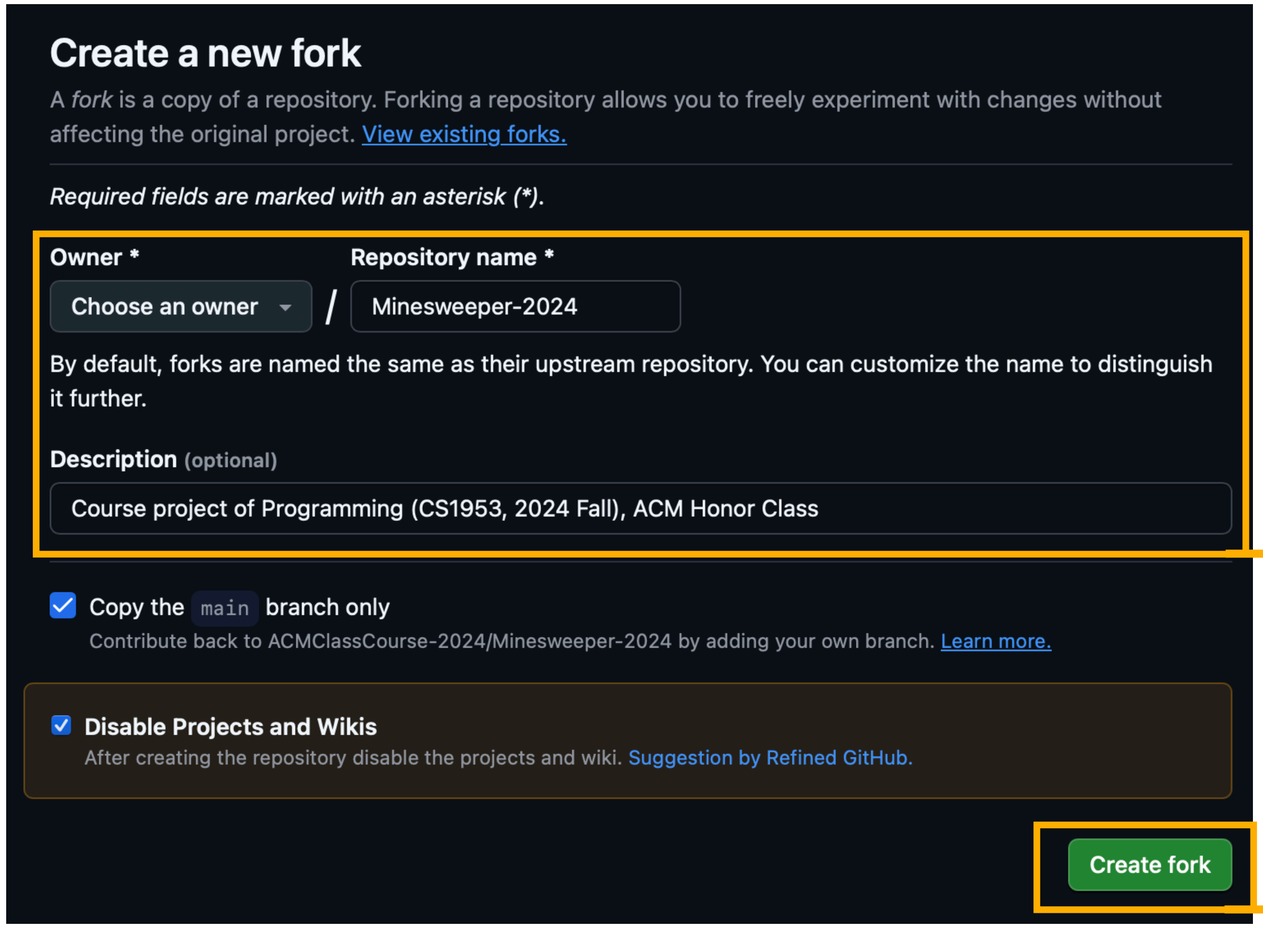


当前分支

看提交的更改

看那时的整个代码仓库

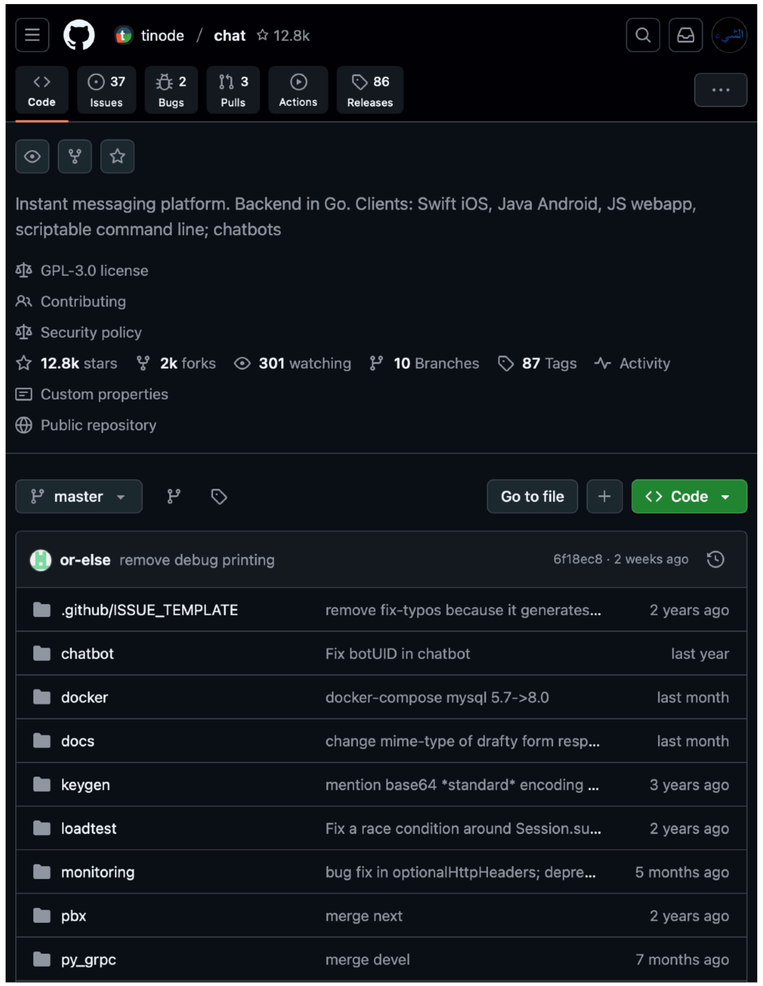
GitHub 上你会使用...

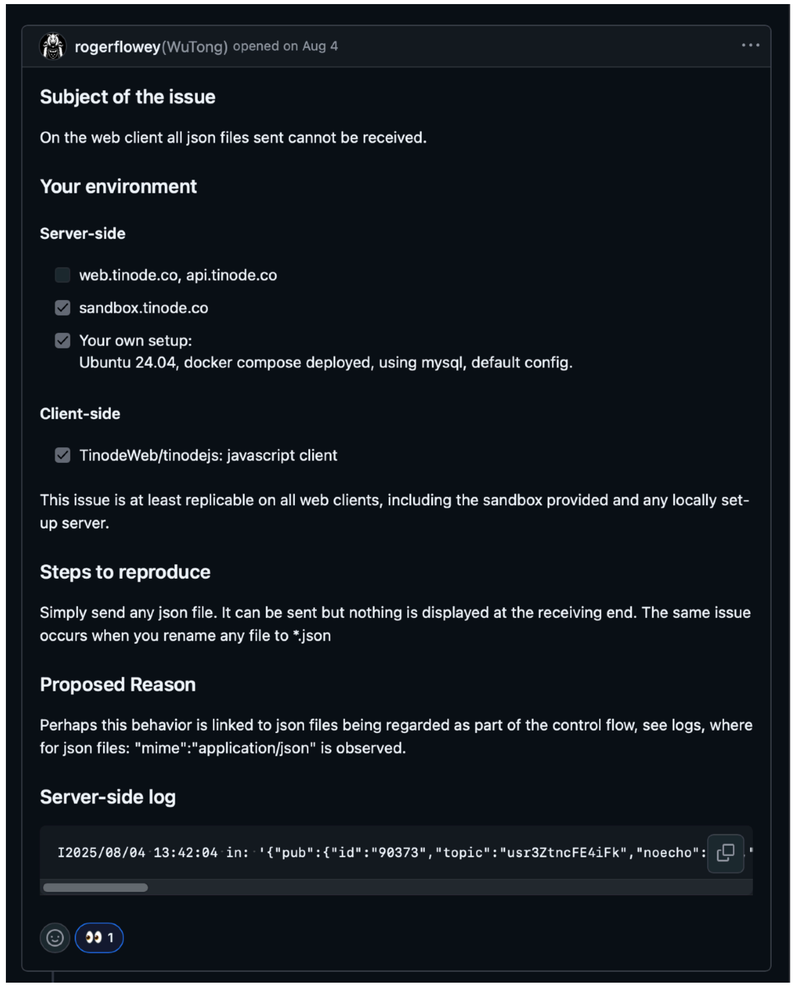


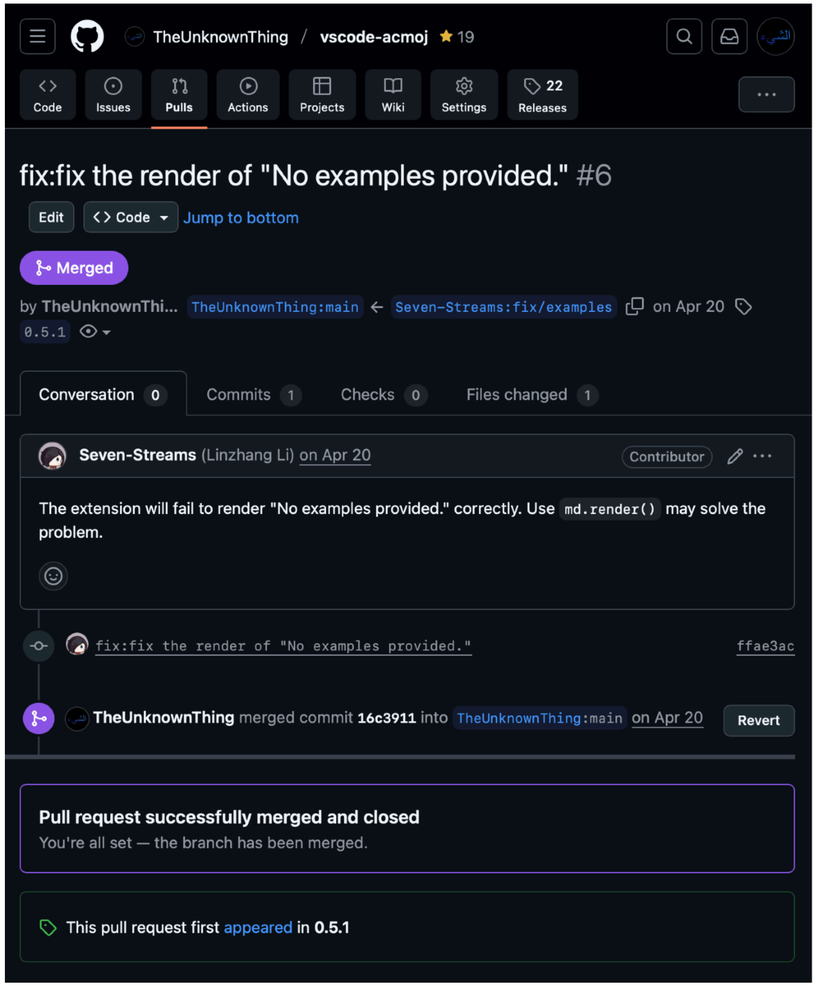
你 Fork 后仓库的信息

创建你的Fork

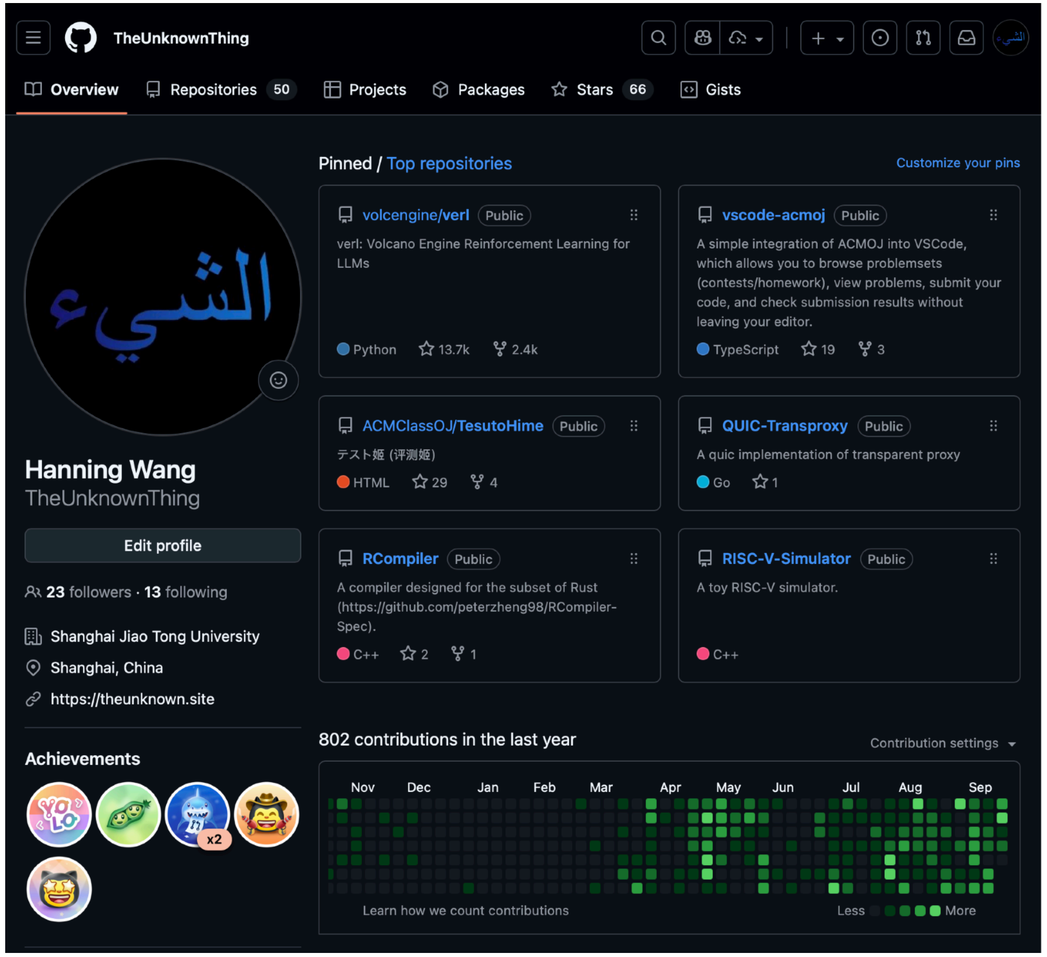
发现项目·提出议题·贡献代码

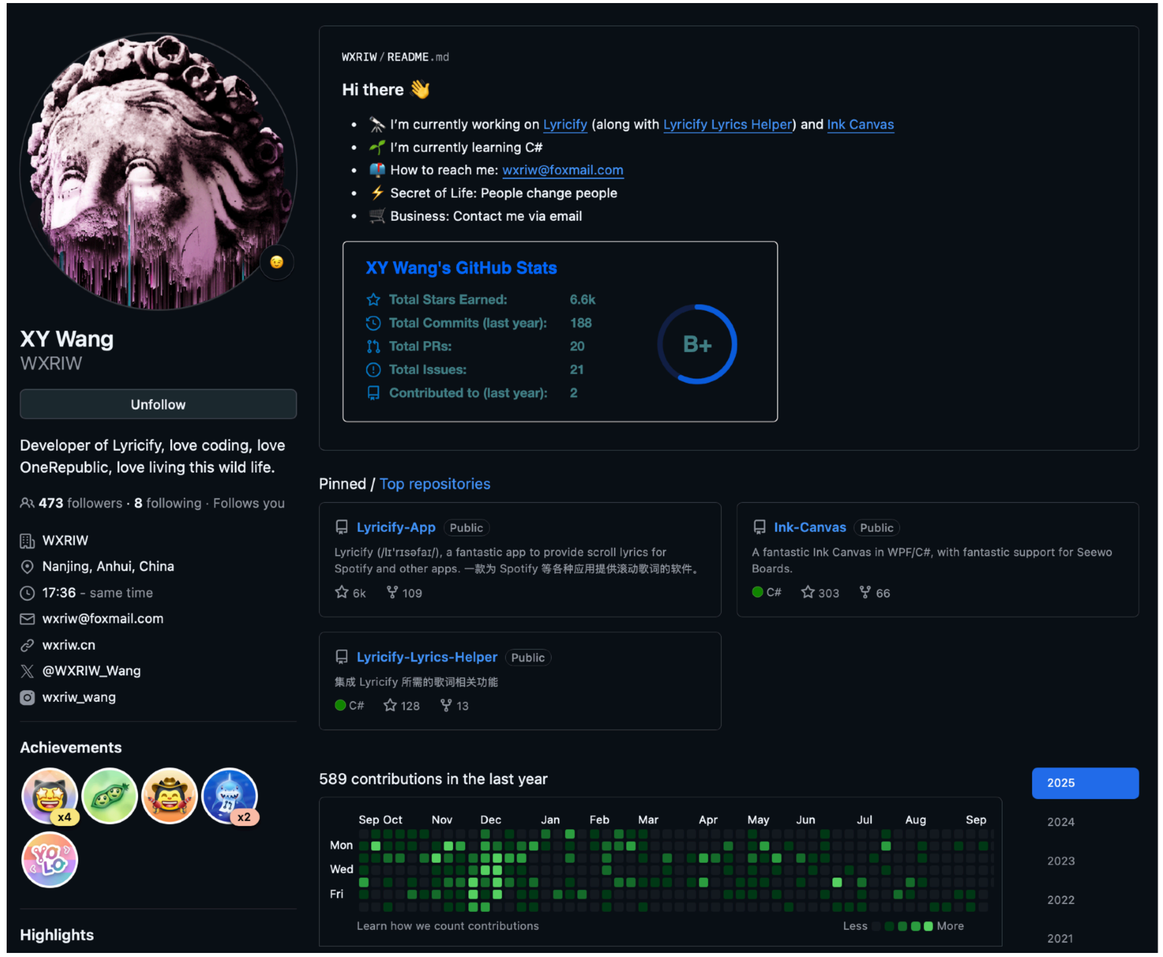






装点门面


TheUnknownThing
Overview
Repositories 50
Projects
Packages
Stars 66
Gists
Hanning Wang
TheUnknownThing
Edit profile
23 followers · 13 following
Shanghai Jiao Tong University
Shanghai, China
https://theunknown.site
Achievements
x2
Pinned / Top repositories
Customize your pins
volcengine/verl Public
verl: Volcano Engine Reinforcement Learning for LLMs
Python 13.7k 2.4k
vscode-acmoj Public
A simple integration of ACMOJ into VSCode, which allows you to browse problemsets (contests/homework), view problems, submit your code, and check submission results without leaving your editor.
TypeScript 19 3
ACMClassOJ/TesutoHime Public
テスト姫 (评测姬)
HTML 29 4
QUIC-Transproxy Public
A quic implementation of transparent proxy
Go 1
RCompiler Public
A compiler designed for the subset of Rust (https://github.com/peterzheng98/RCompiler-Spec).
C++ 2 1
RISC-V-Simulator Public
A toy RISC-V simulator.
C++
802 contributions in the last year
Contribution settings
Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep
Learn how we count contributions
Less More


WXRIW / README.md
Hi there 👋
I'm currently working on Lyricify (along with Lyricify Lyrics Helper) and Ink Canvas
I'm currently learning C#
How to reach me: wxriw@foxmail.com
Secret of Life: People change people
Business: Contact me via email
XY Wang's GitHub Stats
Total Stars Earned: 6.6k
Total Commits (last year): 188
Total PRs: 20
Total Issues: 21
Contributed to (last year): 2
B+
XY Wang
WXRIW
Unfollow
Developer of Lyricify, love coding, love OneRepublic, love living this wild life.
473 followers · 8 following · Follows you
WXRIW
Nanjing, Anhui, China
17:36 - same time
wxriw@foxmail.com
wxriw.cn
@WXRIW_Wang
wxriw_wang
Achievements
x4
x2
Highlights
Pinned / Top repositories
Lyricify-App Public
Lyricify (/lɪˈrɪsəfaɪ/), a fantastic app to provide scroll lyrics for Spotify and other apps. 一款为 Spotify 等各种应用提供滚动歌词的软件。
6k 109
Ink-Canvas Public
A fantastic Ink Canvas in WPF/C#, with fantastic support for Seewo Boards.
C# 303 66
Lyricify-Lyrics-Helper Public
集成 Lyricify 所需的歌词相关功能
C# 128 13
589 contributions in the last year
2025
2024
2023
2022
2021
Sep Oct Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep
Mon
Wed
Fri
Learn how we count contributions
Less More

GitHub

图形化操作 Git

怎么用 Git?

什么是 Git?

为什么要用 Git?

Dojo

GitHub

图形化操作 Git

怎么用 Git?

什么是 Git?

为什么要用 Git?

ACM.DOJO

Reference

http://ohshitgit.com/

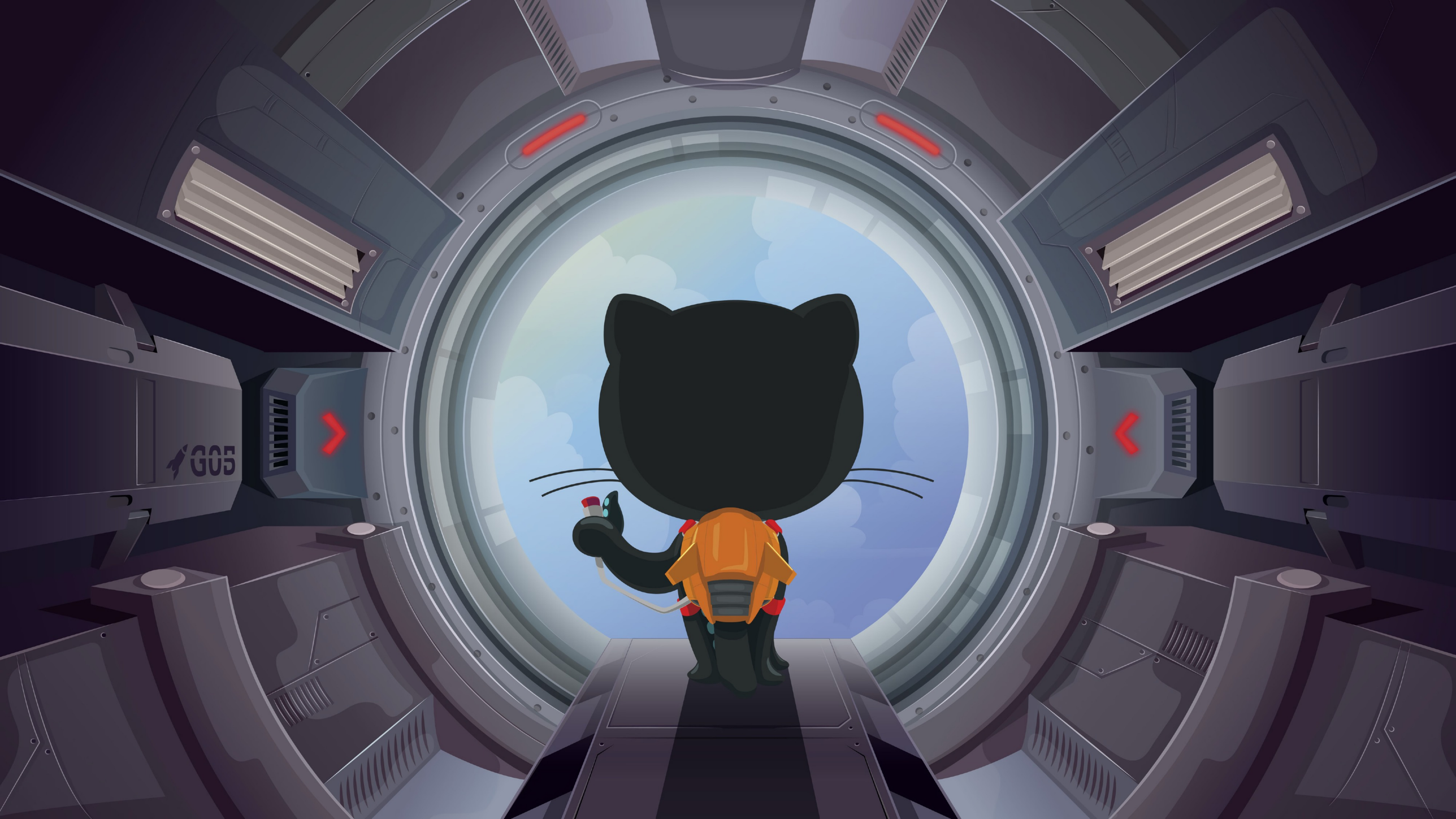
https://git-scm.com/docs

http://merriam-webster.com/dictionary/git

https://www.wolai.com/zymbox/gYBwBGsBpfqrnsXFiUBqFo

https://lau.yeeyu.org/blog-zh-cn/git-basic-usage-zh-cn/

https://missing-semester-cn.github.io/2020/version-control


G05