

Git & GitHub Organization

"FINAL".doc



FINAL.doc!



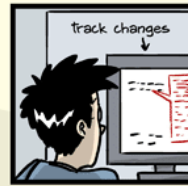
FINAL_rev.2.doc



FINAL_rev.6.COMMENTS.doc



FINAL_rev.8.comments5.
CORRECTIONS.doc



FINAL_rev.18.comments7.
corrections9.MORE.30.doc



FINAL_rev.22.comments49.
corrections.10.##\$%WHYDID
ICOMETOGRADSCHOOL????.doc

JORGE CILIM © 2012

版本控制

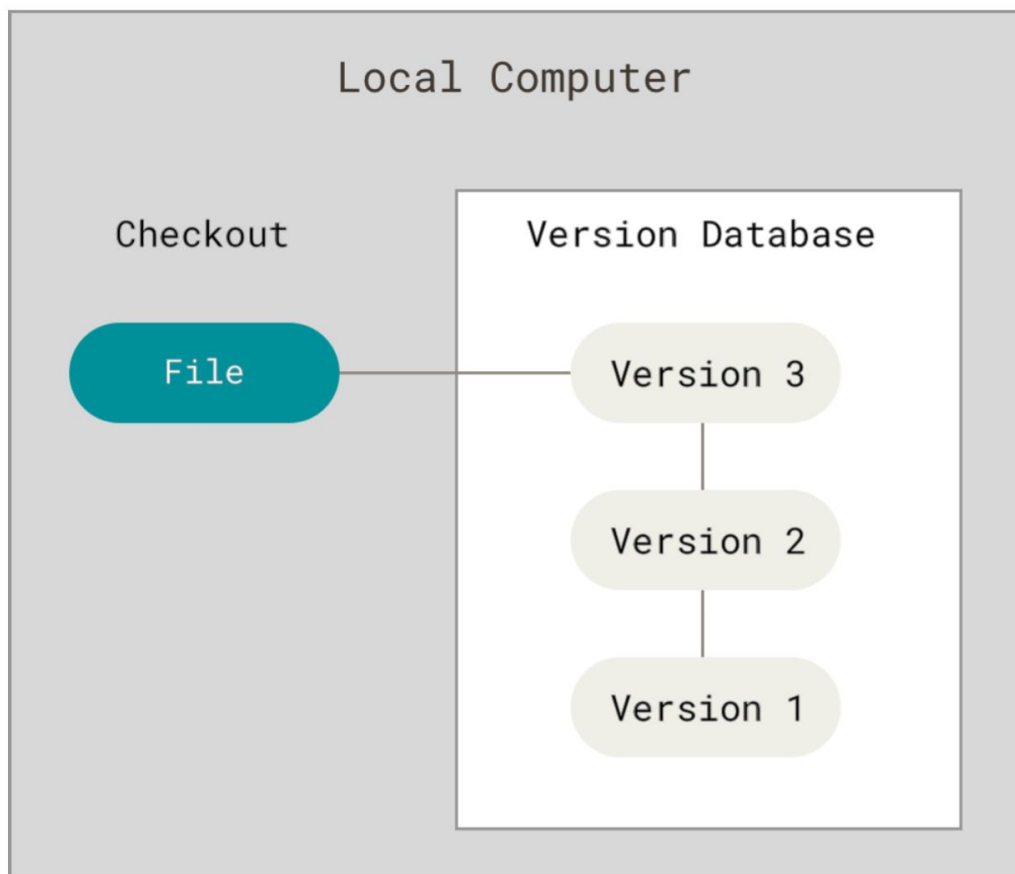
版本控制是一种记录一个或若干文件内容变化，以便将来查阅特定版本修订情况的系统。

- **回退**：将特定的文件回溯至之前的状态或整个项目回退至过去某个时间点
- **比较**：比较文件的变化细节
- **溯源**：查找某个人修改了某个地方或某个人报告了BUG

版本控制

本地版本控制系统

效果的好坏取决于记忆和组织的好坏，大多都是采用某种简单的数据库来记录文件的历次更新差异



版本控制 + 协同

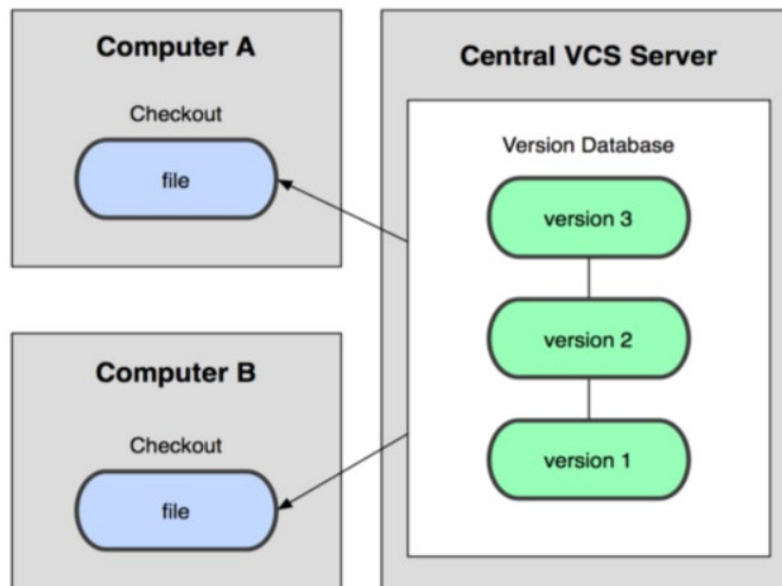
集中化的版本控制系统 (Centralized Version Control Systems, CVCS)

单一的集中管理的服务器，保存所有文件的修订版本

协同工作的人们都通过客户端连到这台服务器，取出最新的文件或者提交更新

优点

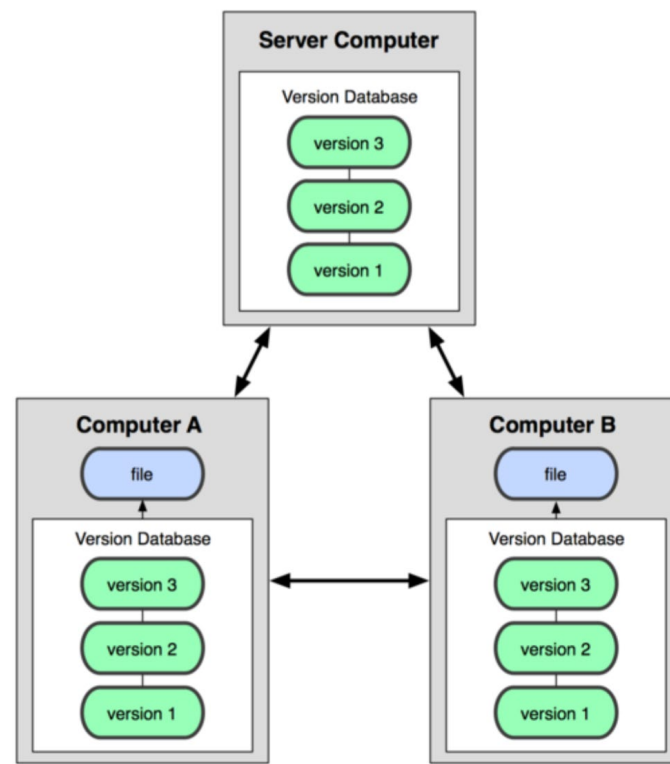
- 每个人都可以一定程度上看到项目中的其他人正在做些什么
- 管理员也可以轻松掌控每个开发者的权限
- 想比在各个客户端上维护本地数据库更轻松容易



版本控制 + 协同 + 分布式

分布式版本控制系统 (Distributed Version Control System, DVCS)

客户端并不只提取最新版本的文件快照，而是把仓库完整地镜像下来，包括完整的历史记录。任何一处协同工作作用的服务器发生故障，事后都可以用任何一个镜像出来的本地仓库恢复。因为每一次的克隆操作，实际上都是一次对代码仓库的完整备份。



Git

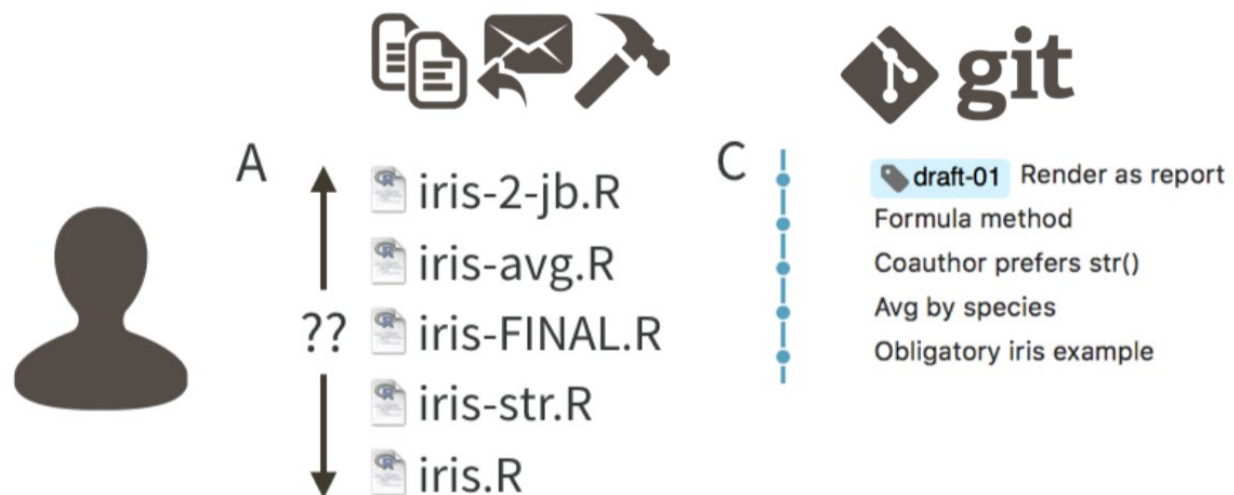
分布式版本控制软件，最初由Linus Torvalds创作
最初目的是为更好地管理Linux内核开发而设计

- 直接记录快照，而非差异比较
- 近乎所有操作都是本地执行
- Git 保证完整性
- Git 一般只添加数据



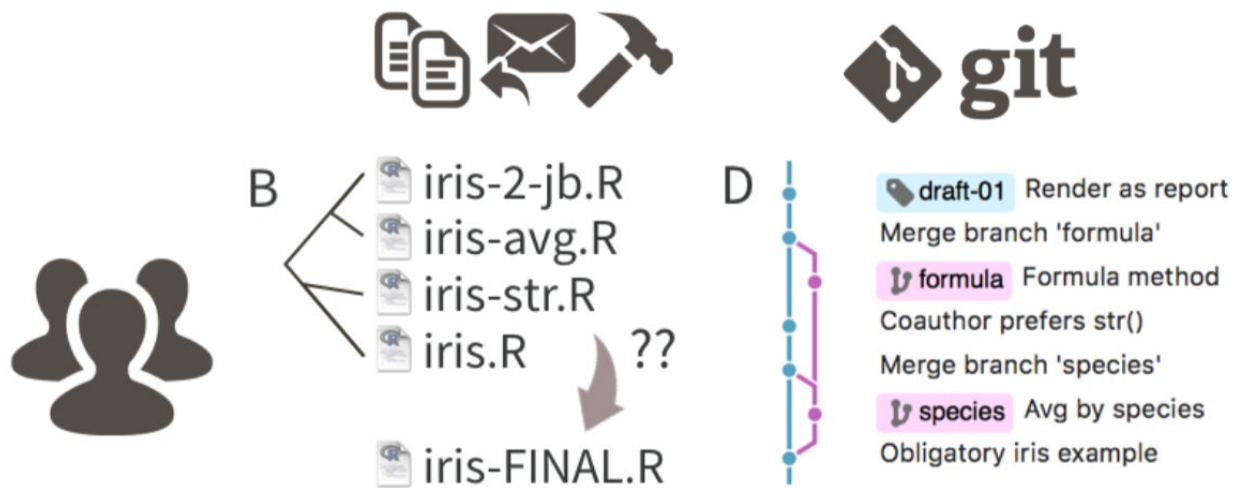
Git

Git允许通过提交信息对项目进行跟踪



Git

使用分支来进行团队工作或新功能开发



Git 安装

Linux (Debian-Ubuntu)

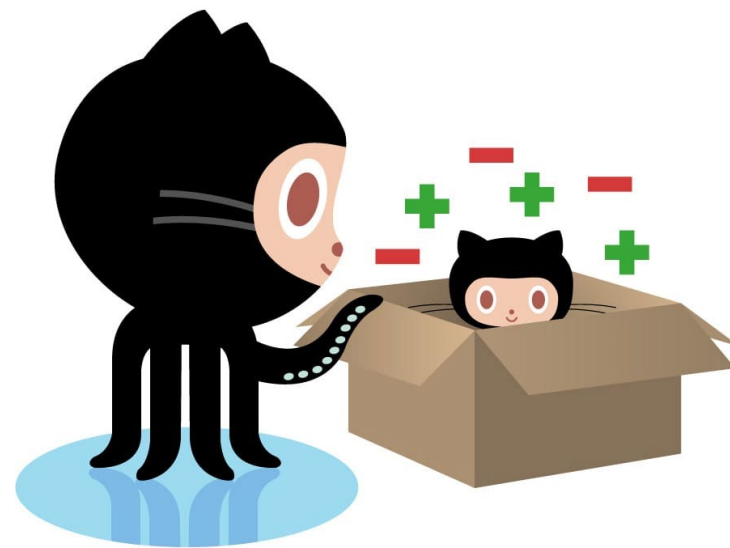
```
$ sudo apt install git
```

macOS

<https://git-scm.com/download/mac>

Windows

<https://git-scm.com/download/win>



Git 配置

```
wang@wang-virtual-machine:~/git$ git commit -m "test commit"

*** 请告诉我你是谁。

运行

    git config --global user.email "you@example.com"
    git config --global user.name "Your Name"

来设置您账号的缺省身份标识。
如果仅在本仓库设置身份标识，则省略 --global 参数。

fatal: 无法自动探测邮件地址 (得到 'wang@wang-virtual-machine.(none)')
```

设置名称和邮箱

- git config --global user.name “Your Name”
- git config --global user.email “you@example.com”

Git 使用

创建本地仓库

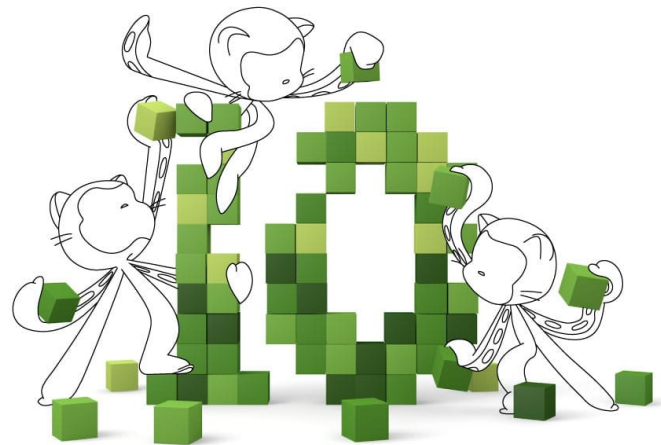
1. 创建本地目录

```
$ mkdir workspace
```

```
$ cd workspace
```

2. 初始化本地仓库

```
$ git init
```



Git 使用

1. 创建工作目录

```
$ mkdir workspace
```

```
$ cd workspace
```

2. 初始化本地仓库

```
$ git init
```

3. 记录更新，跟踪、暂存文件

```
$ git add <filename>
```

3.1 忽略文件

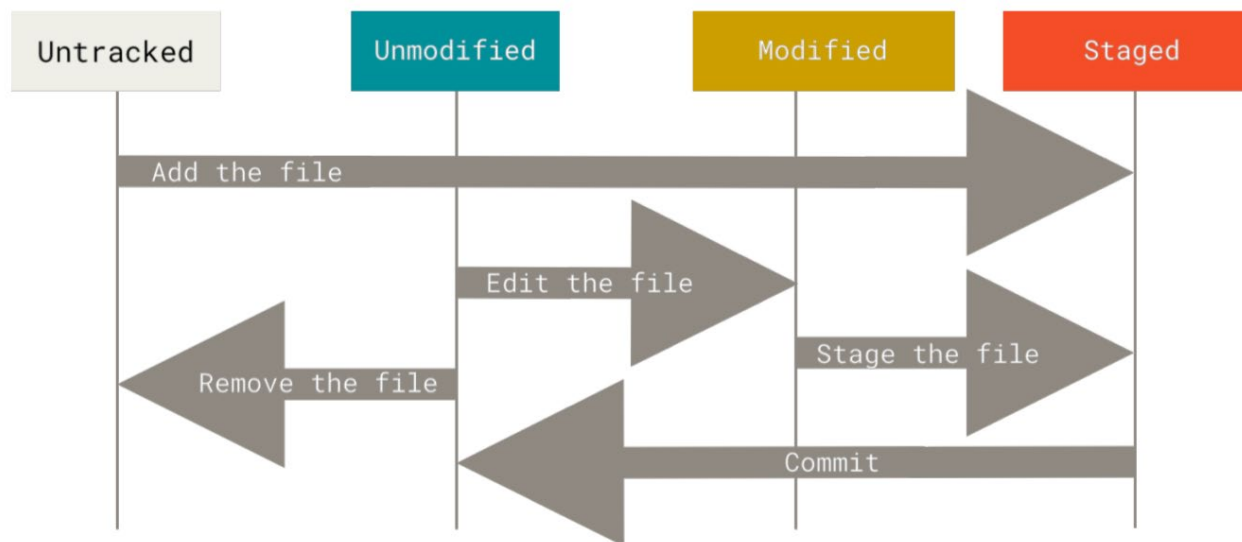
.gitignore

4. 提交更新

```
$ git commit [-amend] “”
```

5. 查看文件状态

```
$ git status / git diff
```



Git 使用

6. 再次跟踪、暂存文件，查看文件状态，提交更新

7. 提交历史查看

```
$ git log [--pretty=oneline]
```

8. 取消暂存和回退

```
$ git reset HEAD <FileName> #取消暂存
```

```
$ git reset HEAD^ #取消提交上一版本
```

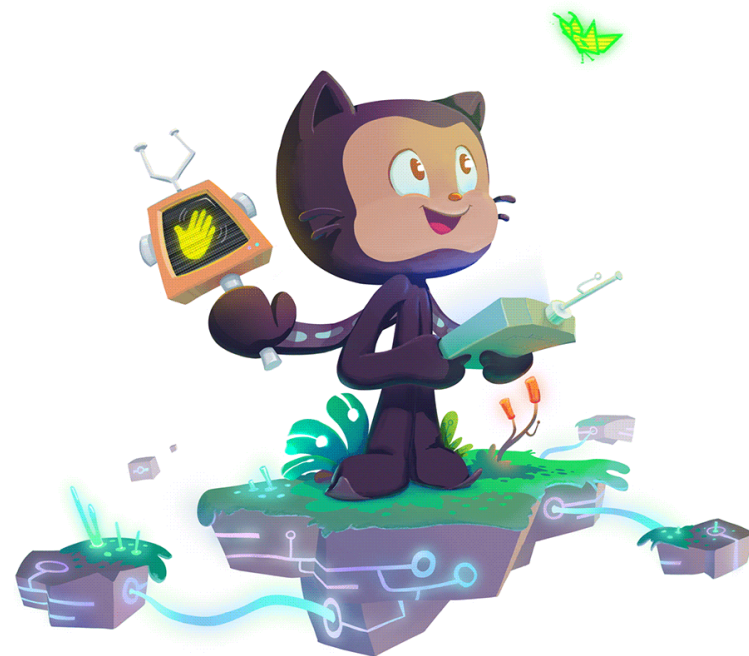
9. 撤销修改

```
$ git checkout -- <file> (危险操作)
```

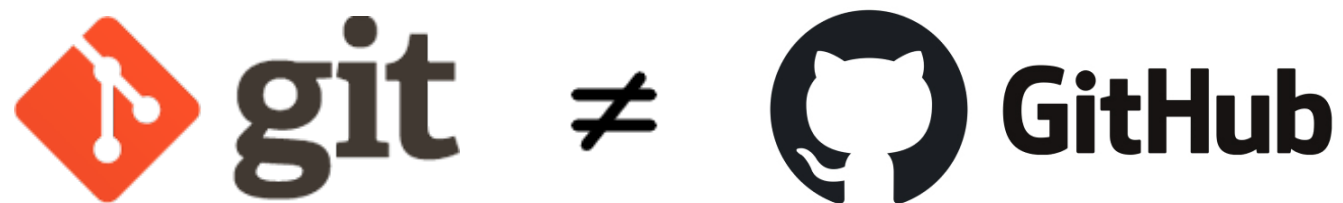
10. 删除文件

```
$ git rm
```

11. 远程仓库



Git & Github



GitHub是通过Git进行版本控制的软件源代码托管服务平台

GitHub提供一个GUI轻松地将存储库分支或克隆到本地机器上，从而促进团队之间的代码库共享

Git & Github



访问Github仓库的两种方式（命令行）： • HTTPS • SSH

两者采用不同的身份验证

验证方法取决于克隆仓库时选择的是 HTTPS 还是 SSH 远程 URL

Git & Github

HTTPS

HTTPS访问:

账号+密码 -> personal access token (PAT) [创建个人访问令牌](#)

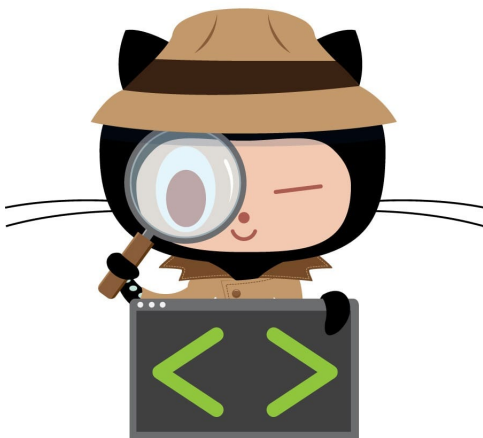
推荐使用HTTPS

1. HTTPS 不太可能被防火墙阻止
2. HTTPS 连接允许credential.helper缓存密码

Git & Github

凭证助手

```
$ git config --global credential.helper store
```



<https://blog.csdn.net/u012163684/article/details/52433645/>

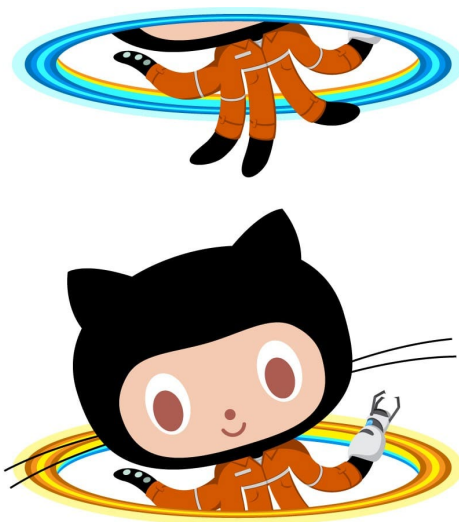
Git & Github

Git 项目仓库

- 将尚未进行版本控制的本地目录转换为Git仓库
- 从其它服务器克隆一个已存在的Git仓库

```
$ cd /home/user/my_project  
$ git init
```

```
$ git clone https://github.com/User/Demo
```




Git & Github

创建GitHub仓库

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner *

 h-wanng ▾

Repository name *

/ Demoo ✓

Great repository names are short and memorable. Need inspiration? How about [studious-enigma](#)?

Description (optional)

Demo Project

☐ Public

Anyone on the internet can see this repository. You choose who can commit.

☒ Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ Add a README file

This is where you can write a long description for your project. [Learn more.](#)

☐ Add .gitignore


Choose which files not to track from a list of templates. [Learn more.](#)

☐ Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

Quick setup — if you've done this kind of thing before

 Set up in Desktop

or

HTTPS

SSH

<https://github.com/h-wanng/Demo.git>

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a README file.

...or create a new repository on the command line

```
echo "# Demo" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/h-wanng/Demo.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/h-wanng/Demo.git
git branch -M main
git push -u origin main
```

Git & Github

13. 连接远程仓库

```
$ git remote add origin https://github.com/h-wanng/Demo.git
```

14. 本地仓库推送到远程仓库

```
$ git push -u origin main
```

15. 删除远程仓库

```
$ git remote -v #查看远程库信息
```

```
$ git remote rm origin
```

16. 创建分支

```
$ git checkout -b dev (-b创建并切换分支)
```

```
$ git branch [-r/-a] #查看分支
```

Git & Github

17. 合并分支

```
$ git merge dev
```

18. 删除分支

```
$ git branch -d dev
```

删除远程分支

```
$ git push origin --delete <BranchName>
```

19. 分支的使用情况

Bug, Feature...

从主分支创建新分支

```
$ git checkout main
```

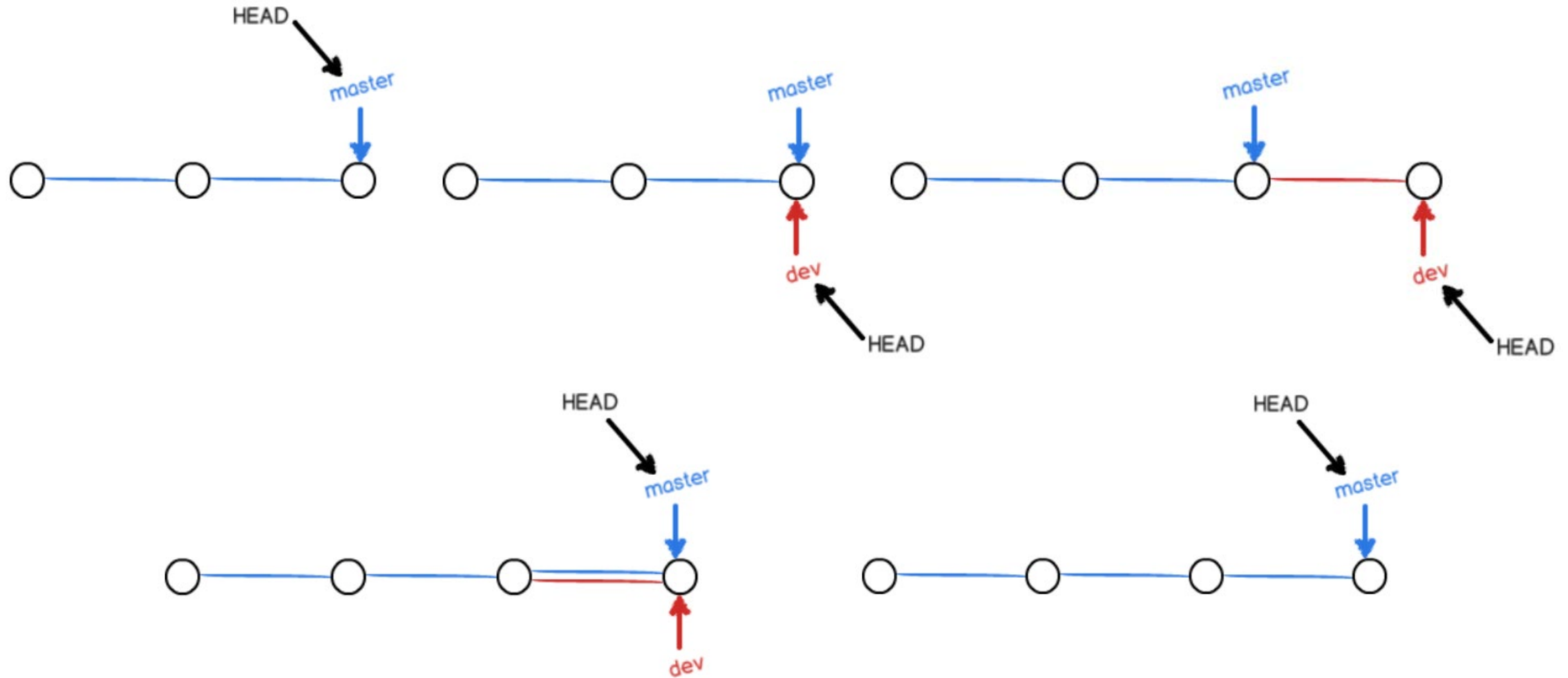
```
$ git checkout -b issue-001
```

20. 远程仓库克隆

```
$ git clone https://github.com/h-wanng/Demo.git [FoldName]
```

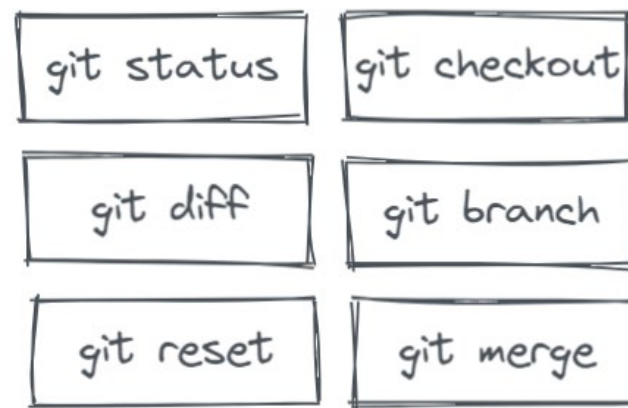
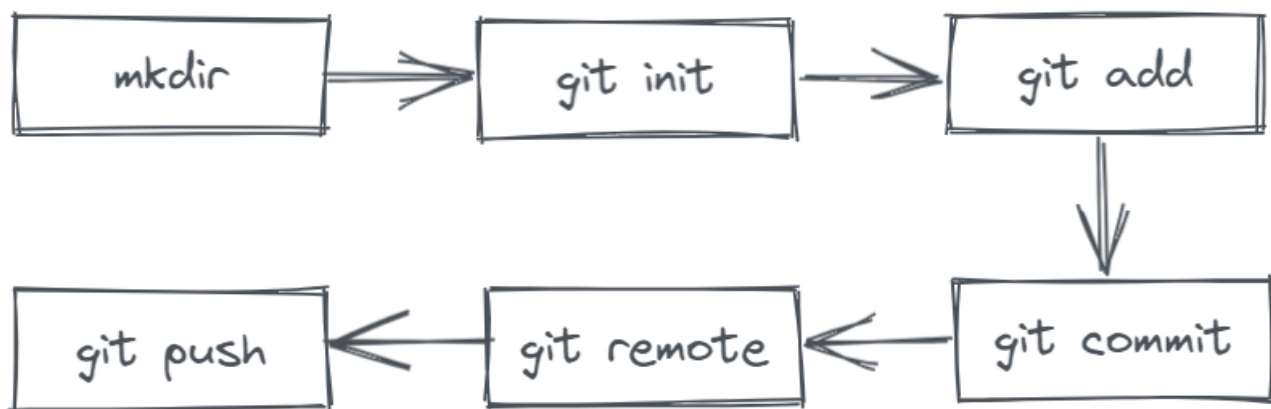
Git & Github

Branch



Git & Github


完整流程

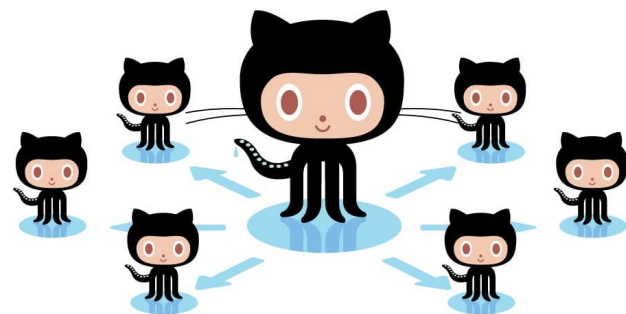
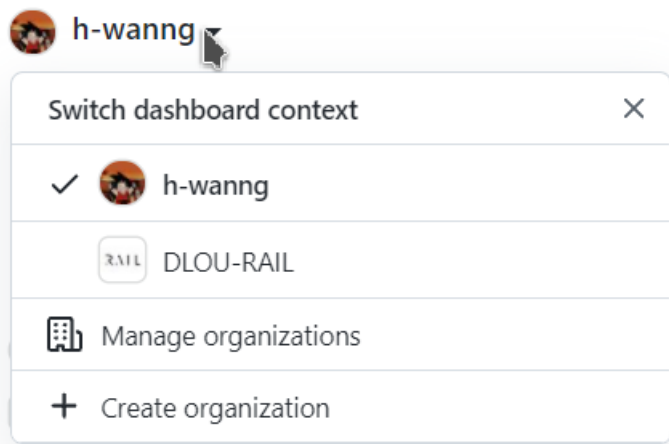


Github Organization

组织是共享帐户，其中业务和开源项目可一次协助处理多个项目

所有者和管理员可通过复杂的安全和管理功能管理成员对组织数据和项目的访问

GitHub 上任意页面的左上角点击Github图标 



Github Organization

组织中的仓库


近期活动


RAIL DLOU-RAIL ▾

[View organization](#)

Repositories [New](#)

Find a repository...


 [dlou-rail.github.io](#)

 [Demo](#)



Recent activity

When you take actions across GitHub, we'll provide links to that activity here.


所有活动

 h-wanng pushed to DLOU-RAIL/dlou-rail.github.io 2 days ago


4 commits to [main](#)


-  7d66234 rename
-  c527c5e title change

2 more commits »

 h-wanng pushed to DLOU-RAIL/dlou-rail.github.io 3 days ago

1 commit to [main](#)

-  f8ade4f favicon and image fix

 h-wanng created a branch in DLOU-RAIL/dlou-rail.github.io 3 days ago

[main](#) in DLOU-RAIL/dlou-rail.github.io

Updated Oct 23

Github Organization



DLOU RAIL

Dalian Ocean University(DLOU) - Robotics and Artificial Intelligence Lab

📍 China 🔗 <https://dlou-rail.github.io/>

[🏠 Overview](#) [📁 Repositories 5](#) [📦 Packages](#) [👤 People 9](#) [👥 Teams 4](#) [📅 Projects 1](#)

📁 Repositories

🔍 Find a repository...

Type ▾

Language ▾

Sort ▾

🖥️ New

[pull-request](#) Private

test repo for pull request

🐍 Python ☆ 0 🍴 1 🔄 0 🔗 0 Updated 3 hours ago

[Gaoming-HU](#) Private

☆ 0 🍴 0 🔄 0 🔗 0 Updated 5 hours ago

[MPC-and-PID-demo](#) Public

py01

🐍 Python ☆ 0 🍴 0 🔄 0 🔗 0 Updated 28 days ago

[dlou-rail.github.io](#) Public

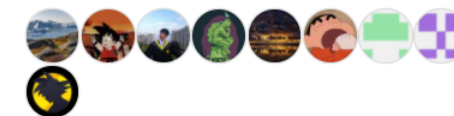
Home page.

🌐 HTML ☆ 1 🍴 0 🔄 0 🔗 0 Updated on 23 Oct

[Demo](#) Private

Test repository.

People



Top languages

🐍 Python 🌐 HTML

Github Organization

 DLOU RAIL

[Overview](#) [Repositories](#) [Packages](#) [People](#) [Teams](#) [Projects](#) [Settings](#)

 Find a repository...

Type ▾

Language ▾

Sort ▾

 New repository

MPC-and-PID-demo Public

py01

 Python  0  0  0  0 Updated yesterday

dlou-rail.github.io Public

Home page.

 HTML  0  0  0  0 Updated 4 days ago

Demo Private

Test repository.


 0  0  0  0 Updated 6 days ago

Github Organization

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner *

 DLOU-RAIL ▾

Repository name *

/

Great repository names are short and memorable. Need inspiration? How about [crispy-palm-tree?](#)

Description (optional)

☐  **Public**

Anyone on the internet can see this repository. You choose who can commit.

☒  **Private**

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ **Add a README file**

This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**

Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**

A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

- 读取**：建议授予要查看或讨论项目的非代码参与者
- 分类**：建议授予需要主动管理议题和拉取请求的参与者，无写入权限
- 写入**：建议授予积极向项目推送的参与者
- 维护**：建议授予需要管理仓库的项目管理者，没有执行敏感或破坏性操作的权限
- 管理员**：建议授予需要完全项目权限的人员，包括执行敏感和破坏性操作，例如管理安全性或删除仓库

Note：在clone private时需要账号密码

Github Organization

RAIL

DLOU RAIL

Overview

Repositories

Packages

People

Teams

Projects

Settings

Organization permissions

Members8

Outside collaborators

Pending collaborators









Pending invitations

Failed invitations

Find a member...

Export

Invite member

Members							2FA	Role
<input type="checkbox"/>		老安LA ANZHIQIANG	2FA	Private	Member	0 teams		
<input type="checkbox"/>		Dr.Pengyu Dr-Pengyu	2FA	Private	Member	0 teams		
<input type="checkbox"/>		H. Wang h-wanng	2FA	Private	Owner	4 teams		
<input type="checkbox"/>		Lovemyse1f	2FA	Private	Member	0 teams		
<input type="checkbox"/>		WANG-DR	2FA	Private	Member	0 teams		
<input type="checkbox"/>		WX wang-xin1998	2FA	Private	Member	0 teams		
<input type="checkbox"/>		吴旭航 wuxuhang	2FA	Private	Member	0 teams		
<input type="checkbox"/>		Yanxiaopang666	2FA	Private	Member	1 team		

Github Organization



DLOU RAIL



Overview



Repositories



Packages



People



Teams



Projects



Settings



Find a team...

New team



Select all

Visibility ▾

Members ▾



Cage Inspection

Cage Inspection Team.



2 members

3 teams ^



Image



1 member

1 team ^



Representation Learning



1 member

0 teams



Sonar




1 member

0 teams

Github Organization

DLOU RAIL / Cage Inspection

 Discussions

 Members 2

Teams 3

Repositories 1

Projects 1

 Settings

Cage Inspection

@DLOU-RAIL/cage-inspection

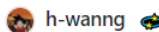
Cage Inspection Team. - Edit

2 members



1 is a member of child teams

Member statuses



Teams

3

Image



Start a discussion with [@DLOU-RAIL/cage-inspection](#)

 Recent

Pinned 0

 Private

Hello World!



h-wanng yesterday

Hello world!



Reply privately...


Github Organization

 DLOU RAIL

 Overview  Repositories  Packages  People  Teams  **Projects**  Settings

 Projects Beta 0


 **Projects** 1

 is:open

New project ▾

 1 Open  0 Closed

Sort ▾

 **UUV Cage** Private

No description

#1 updated yesterday

...

Github Organization

DLOU RAIL

Overview

Repositories

Packages

People

Teams

Projects

Settings

UUV Cage

Updated yesterday

Filter cards

+ Add cards

Fullscreen

Menu

3 To do

Welcome to GitHub Projects

We're so excited that you've decided to create a new project! Now that you're here, let's make sure you know how to get the most out of GitHub Projects.

☒ Create a new project

☒ Give your project a name

☒ Press the ? key to see available keyboard shortcuts

☐ Add a new column

☐ Drag and drop this card to the new column

☐ Search for and add issues or PRs to your project

☐ Manage automation on columns

☐ [Archive a card](#) or archive all cards in a column

Added by h-wanng

Cards

Cards can be added to your board to track the progress of issues and pull requests. You can also add note cards, like this one!

Added by h-wanng

0 In progress

0 Review in progress

0 Reviewer approved

0 Done

Collaboration

Fork (复刻)

 Fork

1

你所管理的仓库的副本。复刻用于更改项目而不影响原始仓库。可以通过拉取请求从原始仓库提取更新，或者提交更改到原始仓库。

- 可以使用**拉取请求**将更改从用户拥有的复刻提交到原始仓库，也称为上游仓库。
- 可以通过**同步复刻**与上游仓库，将更改从上游仓库提交到本地复刻。
- 删除复刻不会删除原始上游仓库，对复刻执行所需的任何更改不会影响原始仓库。复刻的仓库删除后无法恢复。
- 私有复刻继承上游或父仓库的权限结构。
- 从现有仓库的内容创建新仓库，但不想合并上游更改，可以使用复制仓库。
- 在开源项目中，复刻常用于迭代想法或更改，然后将其提交回上游仓库。

Collaboration

配置远程上游仓库

1. 列出当前为复刻配置的远程仓库

```
$ git remote -v  
> origin https://github.com/YOUR_USERNAME/YOUR_FORK.git (fetch)  
> origin https://github.com/YOUR_USERNAME/YOUR_FORK.git (push)
```

2. 指定同步的新远程上游仓库

```
$ git remote add upstream https://github.com/ORIGINAL_OWNER/ORIGINAL_REPOSITORY.git
```

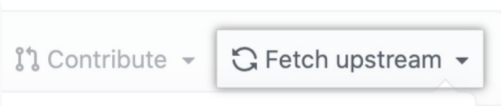
3. 查看新上游仓库

```
$ git remote -v  
> origin      https://github.com/YOUR_USERNAME/YOUR_FORK.git (fetch)  
> origin      https://github.com/YOUR_USERNAME/YOUR_FORK.git (push)  
> upstream    https://github.com/ORIGINAL_OWNER/ORIGINAL_REPOSITORY.git (fetch)  
> upstream    https://github.com/ORIGINAL_OWNER/ORIGINAL_REPOSITORY.git (push)
```

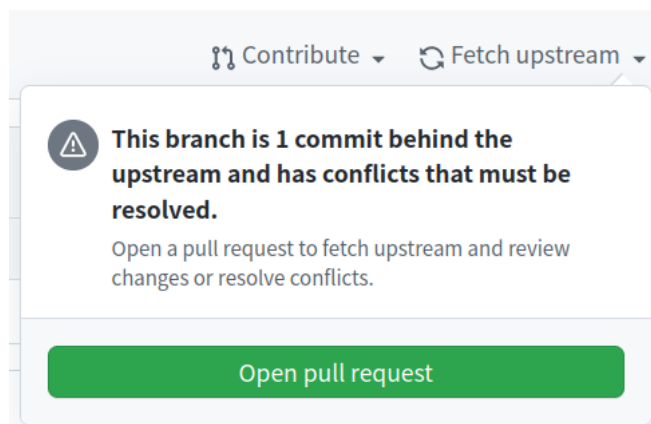
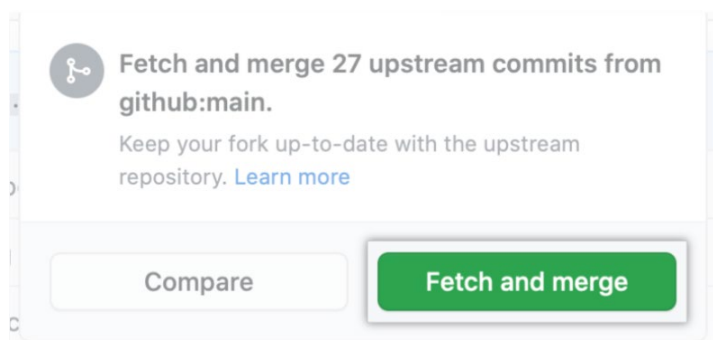
Collaboration

同步复刻

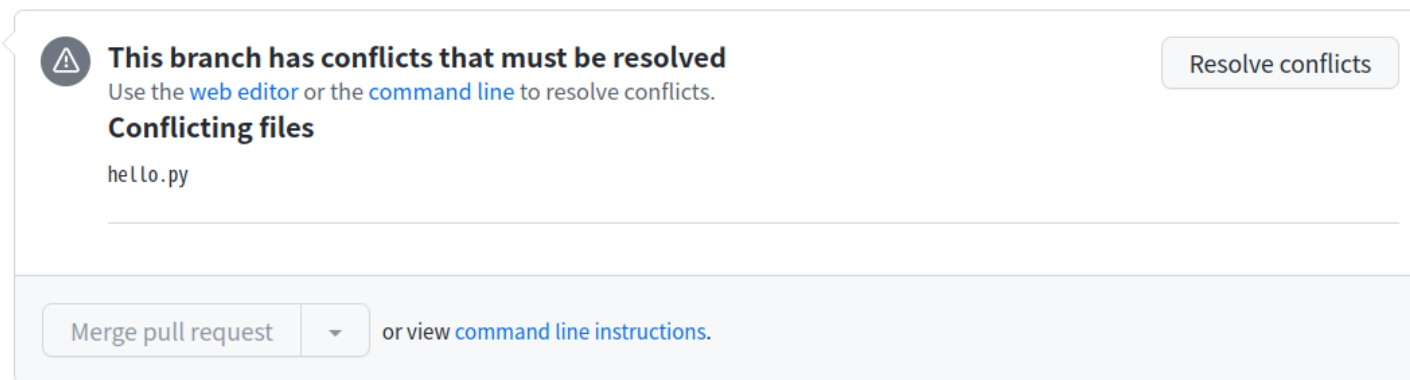
1.



2.

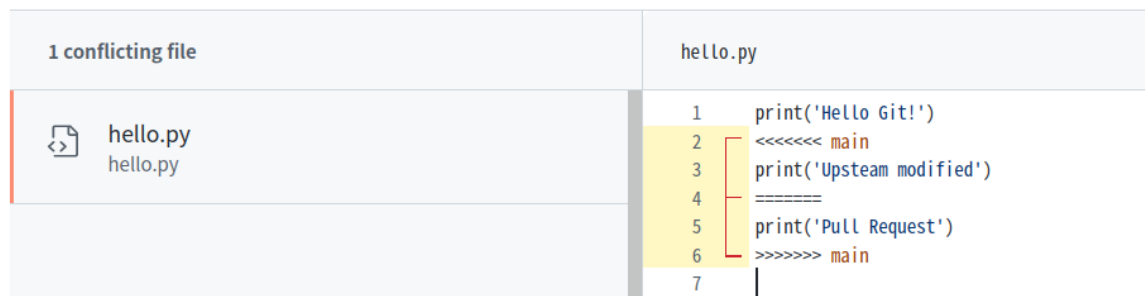


Add more commits by pushing to the `main` branch on `DLOU-RAIL/pull-request`.



Update hello.py #1

Resolving conflicts between `DLOU-RAIL:main` and `h-wanng:main` and committing changes → `DLOU-RAIL:main`



Collaboration

同步复刻

1. 从上游仓库获取分支及其各自的提交

```
$ git fetch upstream
> remote: Counting objects: 75, done.
> remote: Compressing objects: 100% (53/53), done.
> remote: Total 62 (delta 27), reused 44 (delta 9)
> Unpacking objects: 100% (62/62), done.
> From https://github.com/ORIGINAL_OWNER/ORIGINAL_REPOSITORY
> * [new branch]      main      -> upstream/main
```

2. 切换本地分支

```
$ git checkout main
> Switched to branch 'main'
```

3. 将上游默认的更改合并到本地默认分支

```
$ git merge upstream/main
```

```
wang@wang-virtual-machine:~/pull-request$ git merge upstream/main
自动合并 hello.py
冲突 (内容): 合并冲突于 hello.py
自动合并失败, 修正冲突然后提交修正的结果。
```

Collaboration

Pull Request (拉取请求)

拉取请求用于在 GitHub 上向他人告知你已经推送到仓库中分支的更改。在拉取请求打开后，可以与协作者讨论并审查潜在更改，在更改合并到基本分支之前添加跟进提交。

- 任何对版本库有读取权限的人都可以创建拉取请求。
- 如果想为拉取请求创建一个新的分支，而又没有对版本库的写入权限，可以复刻该版本库。

Collaboration

Pull Request (拉取请求)

1. 克隆Fork Repo

```
$ git clone  
https://github.com/YOUR_USERNAME/YOUR_FORK.git
```

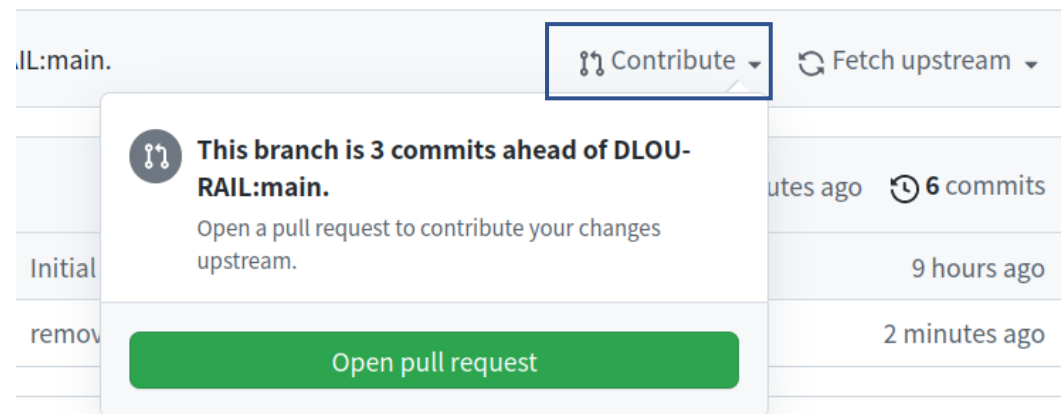
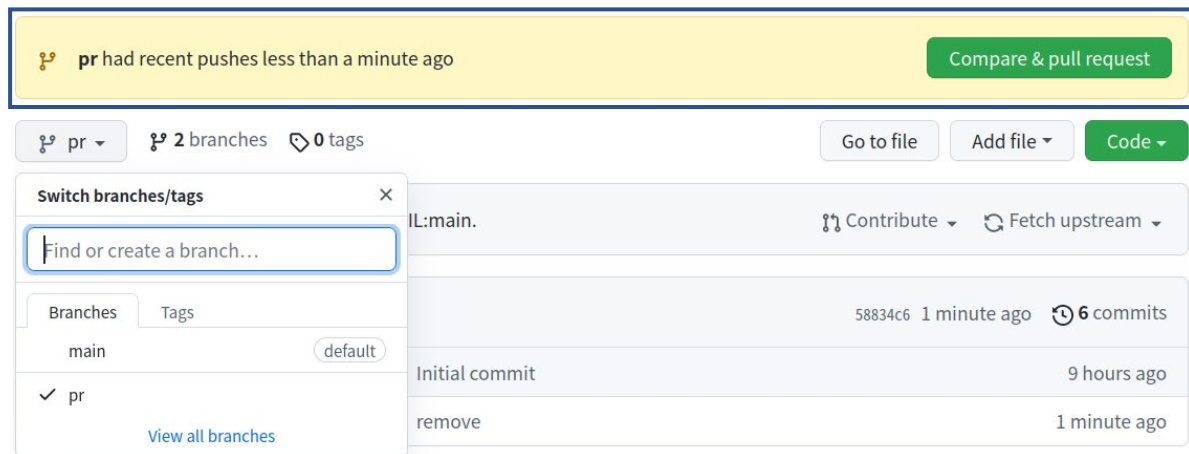
2. 创建新分支

```
$ git checkout -b test-pr
```

3. 修改增加代码

4. 跟踪, 提交, push

```
$ git add . && git commit -m "test-pr"  
$ git push origin test-pr
```



Collaboration

↻

base repository: DLOU-RAIL/pull-request ▾

base: main ▾

←

head repository: h-wanng/pull-request ▾

compare: pr ▾

✓ **Able to merge.** These branches can be automatically merged.

上游repo

fork repo

Pr

pull request标题

Write

Preview

H B I ≡ <> 🔗 ≡ ≡ ☑ @ ↗ ↶

Fix

Remove "upstream modified"

Add

Add "pull request"

pull request描述

Attach files by dragging & dropping, selecting or pasting them. M+

☒ Allow edits by maintainers ⓘ

Create pull request

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

Reviewers

请求审查人员

No reviews

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Linked issues

链接到议题

Use [Closing keywords](#) in the description to automatically close issues

Collaboration

3 commits

1 file changed

1 contributor

Commits on Nov 23, 2021 提交历史

pull request

 h-wanng committed 8 hours ago



974959a



fix

 h-wanng committed 1 hour ago



868ce5f



remove

 h-wanng committed 25 minutes ago



58834c6



Showing 1 changed file with 1 addition and 1 deletion.

比较

Split

Unified

2 hello.py

@@ -1,2 +1,2 @@

1 1 print('Hello Git!')

2 - print('Upstream modified')

2 + print('Pull Request')

Collaboration

审查、合并更改

DLOU-RAIL / pull-request Private

Watch 0

Star 0

Fork 2

<> Code Issues Pull requests 1 Actions Projects Security Insights Settings

Pr #2

Edit

<> Code

Open h-wanng wants to merge 3 commits into DLOU-RAIL:main from h-wanng:pr

Conversation 0

Commits 3

Checks 0

Files changed 1

+1 -1



h-wanng commented 17 minutes ago

...

Fix

Remove "upstream modified"

Add

Add "pull request"



h-wanng added 3 commits 9 hours ago



pull request

974959a



fix

868ce5f



remove

58834c6

Reviewers



No reviews

Assignees



No one—assign yourself

Labels



None yet

Projects



None yet

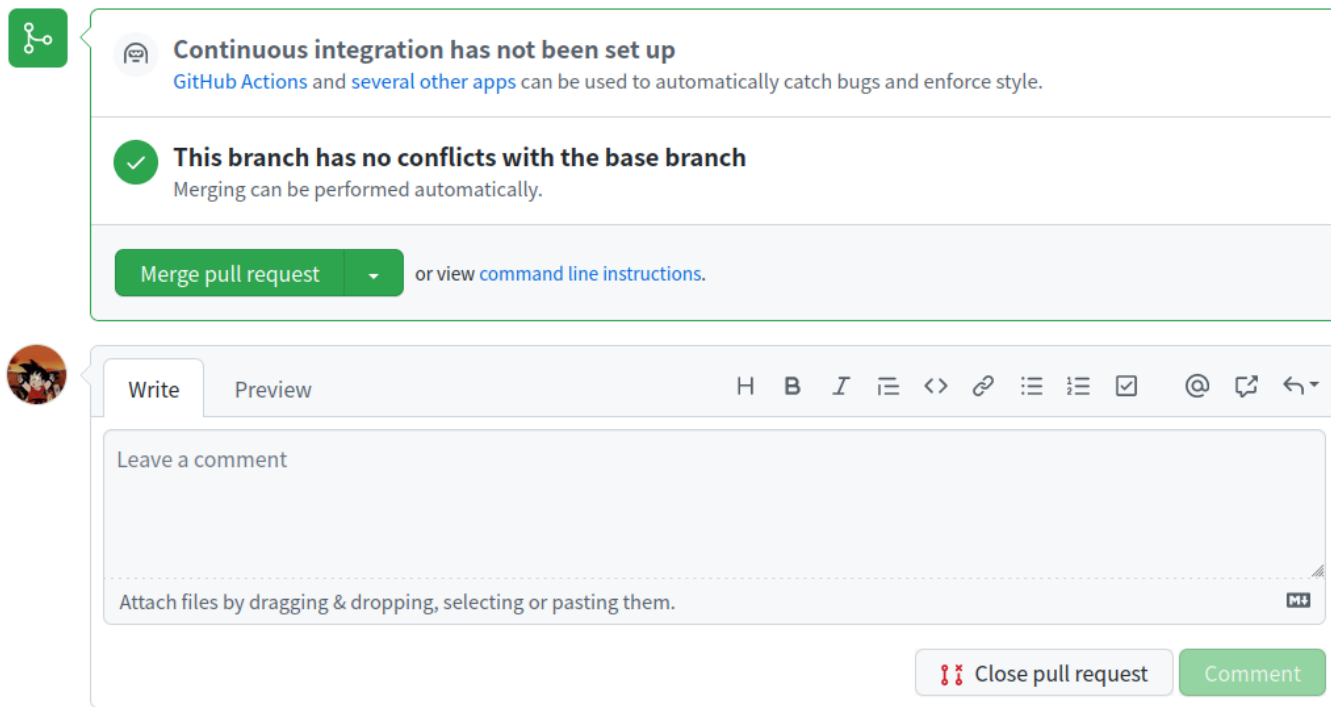
Milestone



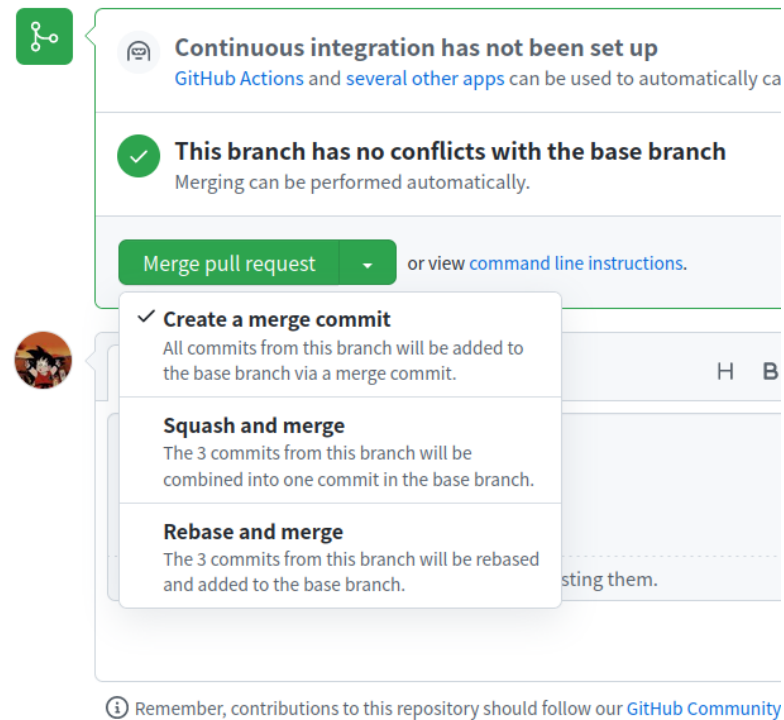
No milestone

Collaboration

Add more commits by pushing to the `pr` branch on `h-wanng/pull-request`.



The screenshot shows a GitHub pull request interface. At the top, a green box contains a status check icon and the text: "Continuous integration has not been set up. GitHub Actions and several other apps can be used to automatically catch bugs and enforce style." Below this, a green checkmark icon is followed by the text: "This branch has no conflicts with the base branch. Merging can be performed automatically." A green button labeled "Merge pull request" is visible, followed by the text "or view command line instructions." Below the status checks, there is a comment box with a "Write" tab selected and a "Preview" tab. The comment box contains the text "Leave a comment" and a placeholder "Attach files by dragging & dropping, selecting or pasting them." At the bottom right of the comment box, there are two buttons: "Close pull request" and "Comment".

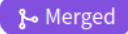



The screenshot shows a GitHub pull request interface. At the top, a green box contains a status check icon and the text: "Continuous integration has not been set up. GitHub Actions and several other apps can be used to automatically ca". Below this, a green checkmark icon is followed by the text: "This branch has no conflicts with the base branch. Merging can be performed automatically." A green button labeled "Merge pull request" is visible, followed by the text "or view command line instructions." Below the status checks, there is a dropdown menu with three options: "Create a merge commit", "Squash and merge", and "Rebase and merge". The "Create a merge commit" option is selected, and its description is visible: "All commits from this branch will be added to the base branch via a merge commit." The "Squash and merge" option is also visible, with its description: "The 3 commits from this branch will be combined into one commit in the base branch." The "Rebase and merge" option is also visible, with its description: "The 3 commits from this branch will be rebased and added to the base branch." At the bottom, there is a note: "Remember, contributions to this repository should follow our GitHub Community".

- Merge pull request: 将fork仓库的所有提交都合并到原仓库，并且产生了一个merge commit log。
- Squash and merge: 将多个commit合并为一个commit添加到原仓库中，会产生一个新的commit id。
- Rebase and merge: 将fork仓库的所有提交都会单独添加到基础分支，而无需合并提交。

Collaboration

Pr #2

 Merged **h-wanng** merged 3 commits into `DLOU-RAIL:main` from `h-wanng:pr`  now

 Conversation **0**  Commits **3**  Checks **0**  Files changed **1**



h-wanng commented 21 minutes ago




Fix
Remove "upstream modified"
Add
Add "pull request"



h-wanng added 3 commits 9 hours ago



 pull request


974959a



 fix

868ce5f



 remove

58834c6



 **h-wanng** merged commit `d81cf57` into `DLOU-RAIL:main` now

[Revert](#)



Pull request successfully merged and closed

[Delete branch](#)

You're all set—the `h-wanng:pr` branch can be safely deleted.

Reference

- Bryan, Jennifer. Excuse Me, Do You Have a Moment to Talk About Version Control? - American Statistician
- Pro Git - Scott Chacon and Ben Straub
- <https://www.liaoxuefeng.com/wiki/896043488029600>
- <https://docs.github.com/cn>
- <https://www.cnblogs.com/eyunhua/p/13215936.html>



Have an Octotastic day!