

AstroHammer Series – 1

# Git: 版本控制与团队协作

Kai Wang  
2018.09.09

1. 创建仓库，添加文件
2. 撤销修改
3. 分支管理
4. 远程仓库
5. 工作模式
6. 自动生成文档
7. 福利时间

# 1. 创建仓库，添加文件

## 1. Initialize the repository

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest [10:02:17]
$ git init
Initialized empty Git repository in /Users/wangk/Desktop/gittest/.git/
```

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master o [10:02:23]
$ |
```

## 2. new a file

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master o [10:02:23]
$ echo hello git > README
```

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master x [10:03:56]
$ ls
README
```

## 3. add the file

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master x [10:03:58]
$ git add README
```

## 4. commit the adding

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master ✘ [10:04:07]
$ git commit -m "add the readme file"
[master (root-commit) 6e40179] add the readme file
  Committer: Kai Wang <wangk@KaideMacBook-Pro.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:
```

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

After doing this, you may fix the identity used for this commit with:

```
git commit --amend --reset-author
```

```
1 file changed, 1 insertion(+)
create mode 100644 README
```

校验和

也是版本库中每个版本的唯一标识

## 5. modify the file

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master o [10:07:12]
$ echo and hello world >> README
```

## 6. check the state

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master ✘ [10:10:22]
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
    (use "git checkout -- <file>..." to discard changes in working directory)

      modified:   README

no changes added to commit (use "git add" and/or "git commit -a")
```

## 6. add the file

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master ✘ [10:11:00]
$ git add README
```

## 7. commit the modification

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master ✘ [10:11:32]
$ git commit -m "modify the readme"
[master 4176bfa] modify the readme
Committer: Kai Wang <wangk@KaideMacBook-Pro.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:
```

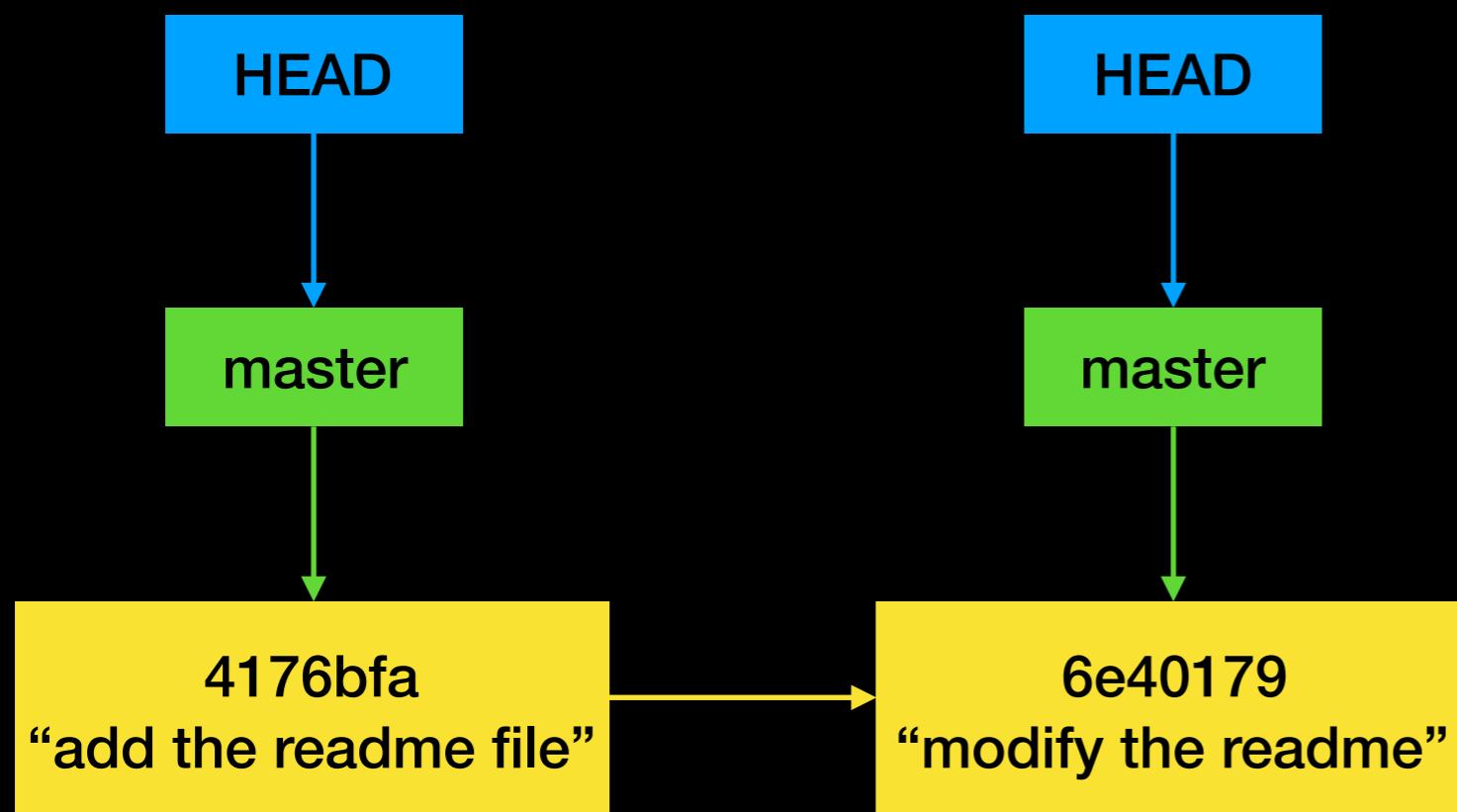
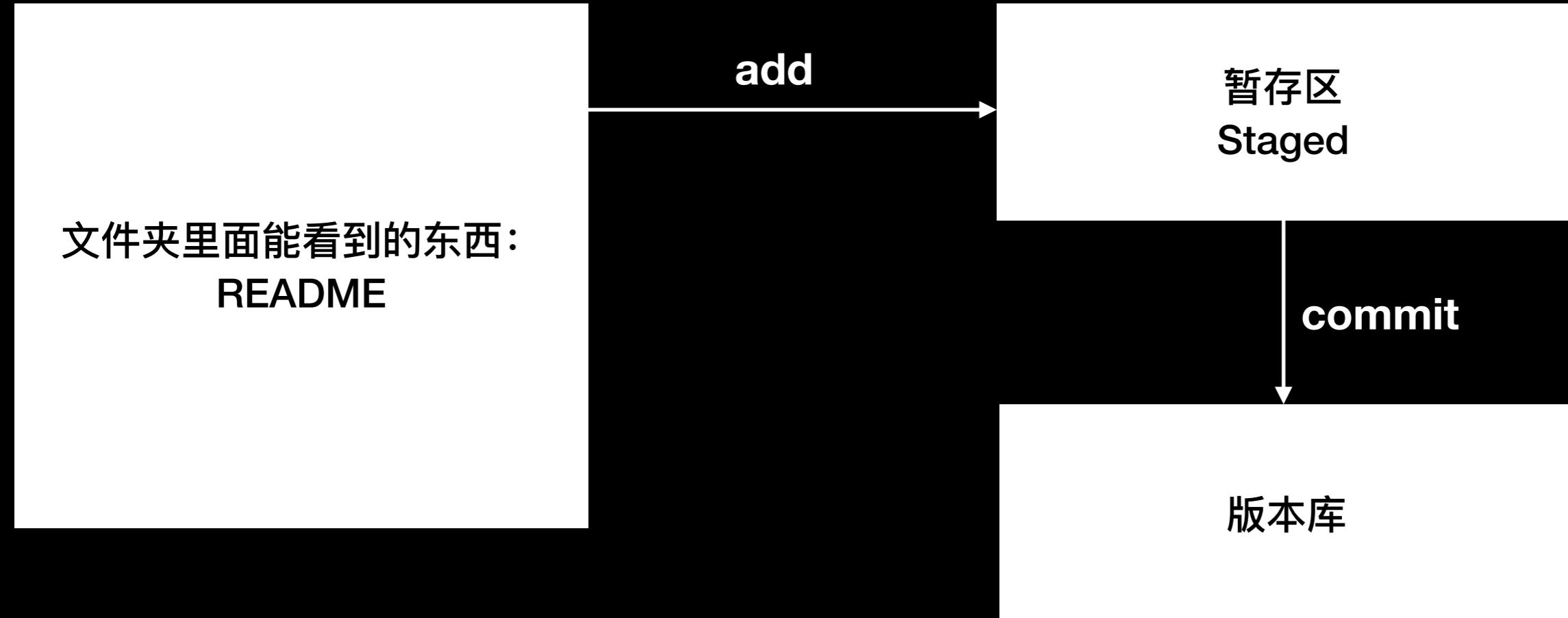
```
git config --global user.name "Your Name"
git config --global user.email you@example.com
```

After doing this, you may fix the identity used for this commit with:

```
git commit --amend --reset-author
```

```
1 file changed, 1 insertion(+)
```

校验和变了  
因为是不同的版本  
每次commit都会产生一个版本



```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master o [10:25:35]
$ git log
```



```
commit 4176bfafb1f1e39ba3f8cc026e8e3493711c8629 (HEAD -> master)
Author: Kai Wang <wangk@KaideMacBook-Pro.local>
Date: Thu Aug 30 10:12:07 2018 +0800
```

modify the readme

```
commit 6e401798edd1928a85644d57288b28437468c4cd
Author: Kai Wang <wangk@KaideMacBook-Pro.local>
Date: Thu Aug 30 10:04:27 2018 +0800
```

add the readme file



Time

```
# wangk @ KaideMacBook-Pro in ~/Documents/Project/9.GalaxyGroup/pyscript on git:master x [19:10:15]
```

```
$ cat ~/.gitconfig
```

[alias]

```
st = status
```

```
co = checkout
```

```
ci = commit
```

```
br = branch
```

```
last = log -1
```

```
lg = log --color --graph --pretty=format:'%Cred%h%Creset -%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold blue)<%an>%Creset' --abbrev-commit
```

```
unstage = reset HEAD
```

[credential]

```
helper = store
```

Untracked

Unmodified

Modified

Staged

Add the file

Remove the file

Edit the file

Stage the file

Commit

## 2. 撤销修改

修改文件

add 文件

commit 文件

撤销修改文件  
(使用暂存区文件)

找不回来的

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master o [10:33:42]
$ echo hello baba >> README
```

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master x [10:37:28]
$ cat README
hello git
and hello world
hello baba
```

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master x [10:37:39]
$ git checkout -- README
```

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master o [10:38:06]
$ cat README
hello git
and hello world
```

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master o [10:38:09]
$ echo hello erzi >> README

# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master x [10:39:05]
$ git add README

# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master x [10:39:12]
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   README
```

撤销add文件

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master x [10:39:37]
$ git reset HEAD README
Unstaged changes after reset:
M       README

# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master x [10:39:58]
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   README

no changes added to commit (use "git add" and/or "git commit -a")
```

还在文件中

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master ✘ [10:40:02]
$ echo hello guys >> README

# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master ✘ [10:41:10]
$ git add README

# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master ✘ [10:41:14]
$ git commit -m "dajiahao"
[master 5c50744] dajiahao
Committer: Kai Wang <wangk@KaideMacBook-Pro.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

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

After doing this, you may fix the identity used for this commit with:

```
git commit --amend --reset-author
```

```
1 file changed, 2 insertions(+)
```

## 撤销commit文件 版本库回退

>>git lg

```
* 5c50744 - (HEAD -> master) dajiahao (2 minutes ago) <Kai Wang>
* 4176bfa - modify the readme (32 minutes ago) <Kai Wang>
* 6e40179 - add the readme file (39 minutes ago) <Kai Wang>
```

|

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master o [10:44:42]
$ git reset --hard 4176bfa
HEAD is now at 4176bfa modify the readme
```

>>git lg

```
* 4176bfa - (HEAD -> master) modify the readme (33 minutes ago) <Kai Wang>
* 6e40179 - add the readme file (41 minutes ago) <Kai Wang>
```

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master o [10:45:52]
$ cat README
hello git
and hello world
```

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master o [10:45:56]
$ git reset --hard 5c50744
HEAD is now at 5c50744 dajiahao
```

>>git lg

```
* 5c50744 - (HEAD -> master) dajiahao (6 minutes ago) <Kai Wang>
* 4176bfa - modify the readme (35 minutes ago) <Kai Wang>
* 6e40179 - add the readme file (43 minutes ago) <Kai Wang>
|
```

撤销commit文件  
版本库回退

记忆力不好的人  
找不回来的

退回上一个版本库

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master o [10:47:33]
$ git reset --hard HEAD^
HEAD is now at 4176bfa modify the readme
```

>>git lg

---

```
* 4176bfa - (HEAD -> master) modify the readme (36 minutes ago) <Kai Wang>
* 6e40179 - add the readme file (44 minutes ago) <Kai Wang>
```

### 3. 分支管理

## 1. new a branch

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master o [11:13:31]
$ git branch dev
```

## 2. goto the “dev” branch

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master o [11:13:53]
$ git checkout dev
Switched to branch 'dev'
```

## 3. “add” and “commit” a file

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:dev o [11:16:25]
$ vim program1.py
```

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:dev x [11:16:45]
$ cat program1.py
print("hello world")
```

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:dev x [11:18:06]
$ git add program1.py
```

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:dev x [11:18:25]
$ git commit -m "the first program"
[dev 6c321e5] the first program
Committer: Kai Wang <wangk@KaideMacBook-Pro.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:
```

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

After doing this, you may fix the identity used for this commit with:

```
git commit --amend --reset-author
```

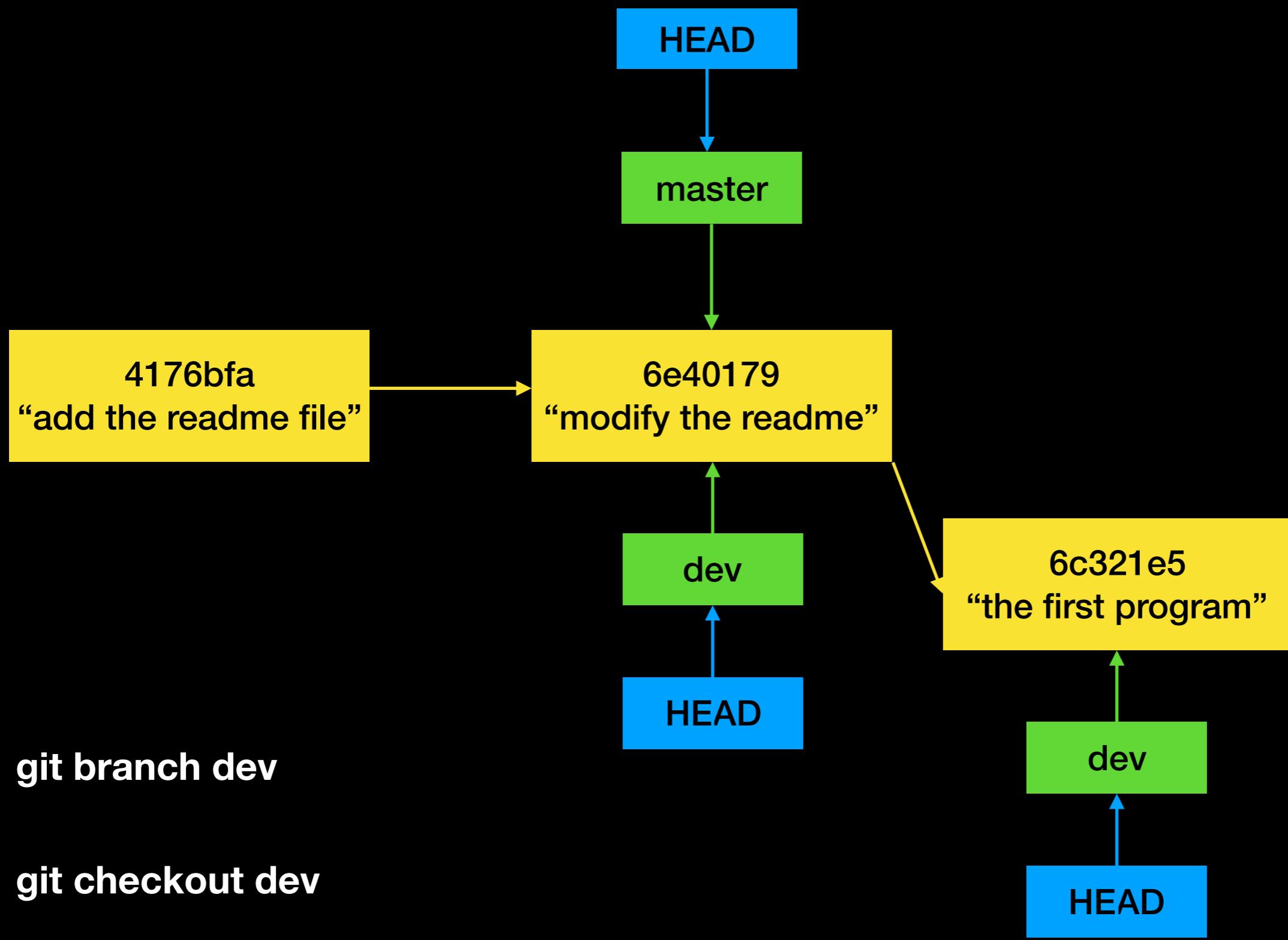
```
1 file changed, 1 insertion(+)
create mode 100644 program1.py
```

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:dev o [11:19:51]
$ ls
README          program1.py

# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:dev o [11:19:54]
$ git checkout master
Switched to branch 'master'

# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master o [11:20:08]
$ ls
README

# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master o [11:20:09]
$ git branch
  dev
* master
```



## 4. merge the branches

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:dev o [11:25:40]
$ git checkout master
Switched to branch 'master'

# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master o [11:25:45]
$ git merge dev
Updating 4176bfa..6c321e5
Fast-forward
  program1.py | 1 +
  1 file changed, 1 insertion(+)
  create mode 100644 program1.py

# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master o [11:25:54]
$ ls
README          program1.py
```

>> git log

```
* 6c321e5 - (HEAD -> master, dev) the first program (8 minutes ago) <Kai Wang>
* 4176bfa - modify the readme (75 minutes ago) <Kai Wang>
* 6e40179 - add the readme file (82 minutes ago) <Kai Wang>
```

## 5. delete the branch

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master o [11:27:26]
$ git branch -d dev
Deleted branch dev (was 6c321e5).

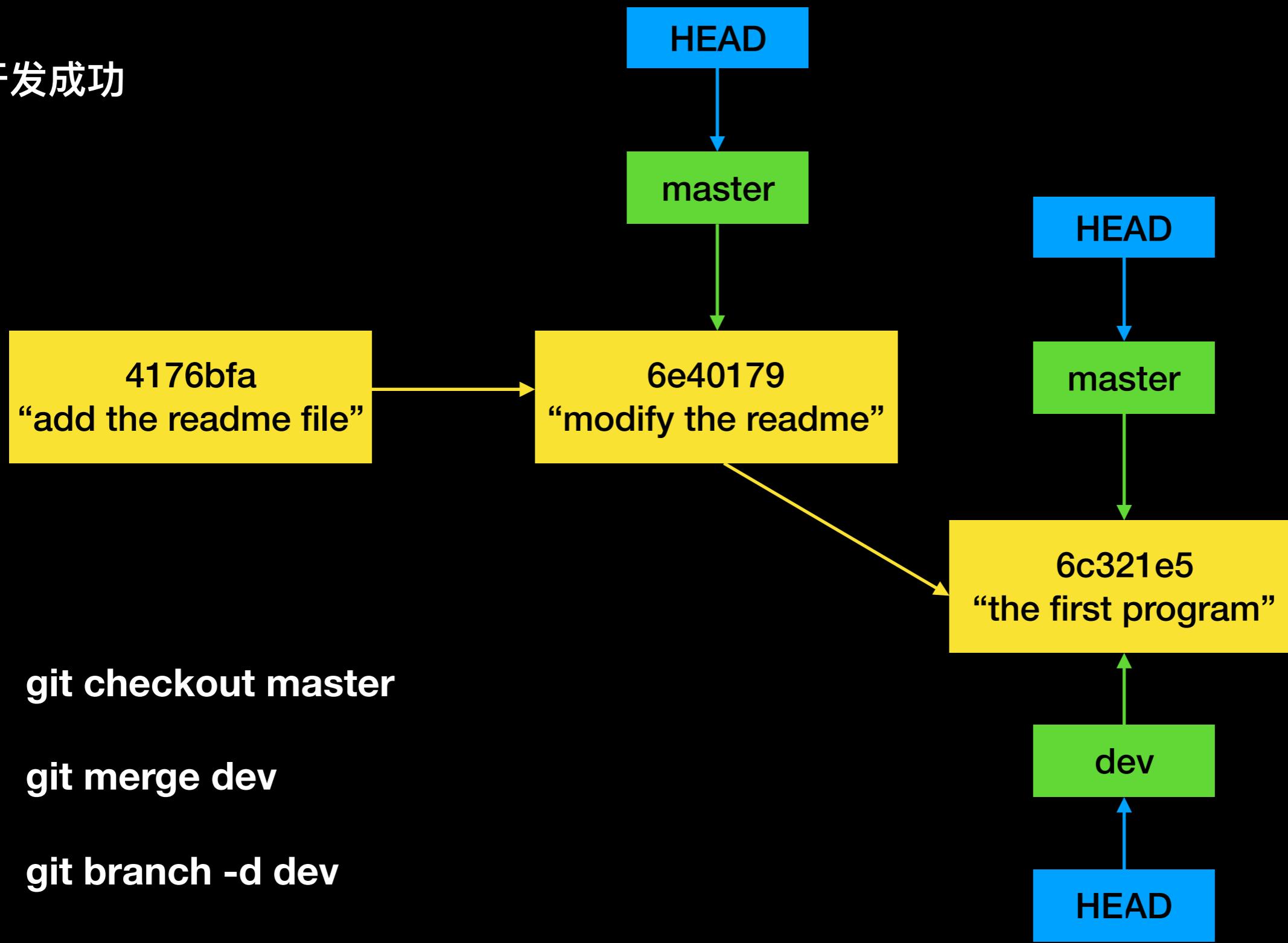
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master o [11:27:30]
$ git branch
* master
```

>> git log

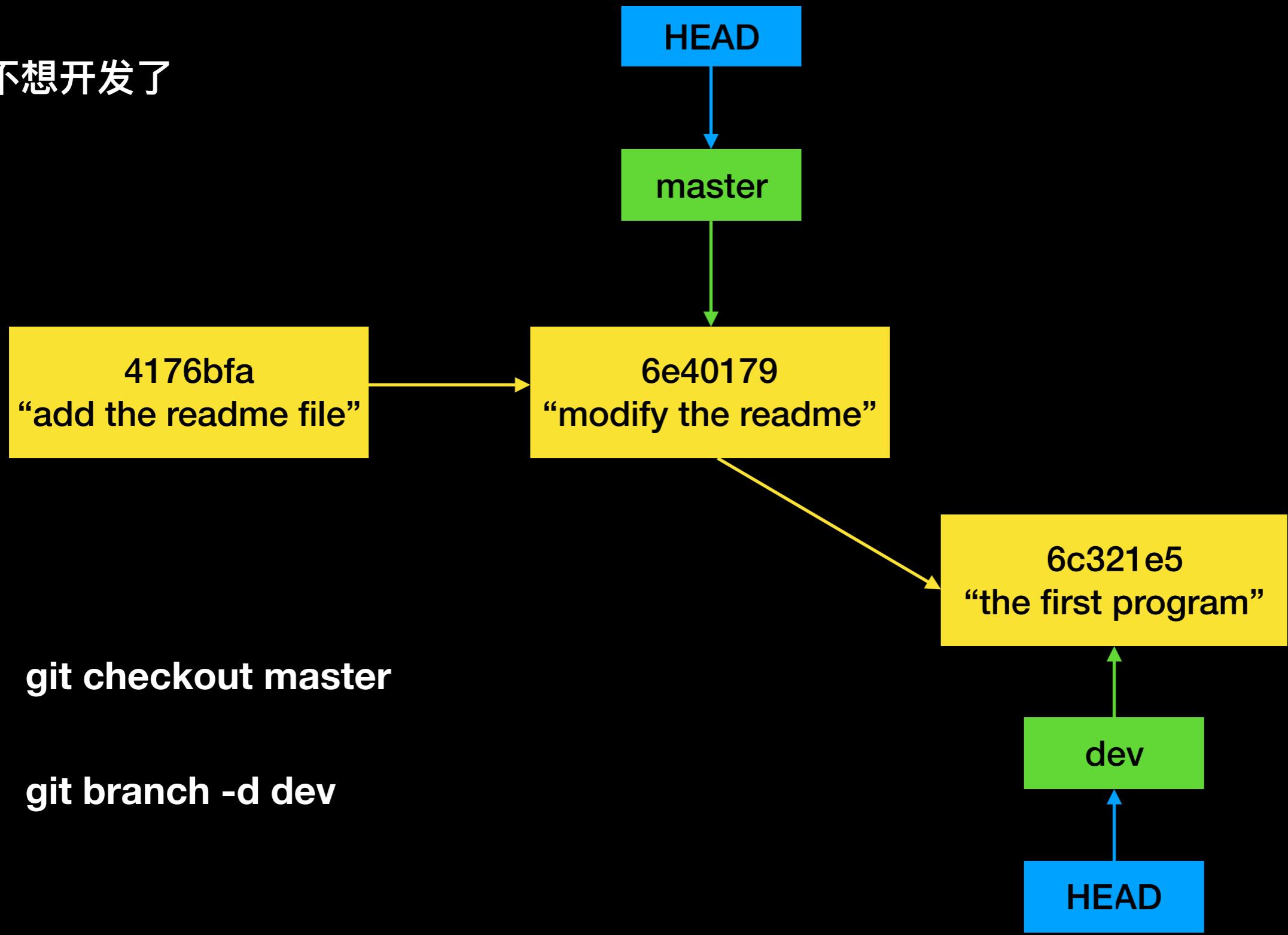
---

```
* 6c321e5 - (HEAD -> master) the first program (10 minutes ago) <Kai Wang>
* 4176bfa - modify the readme (77 minutes ago) <Kai Wang>
* 6e40179 - add the readme file (84 minutes ago) <Kai Wang>
```

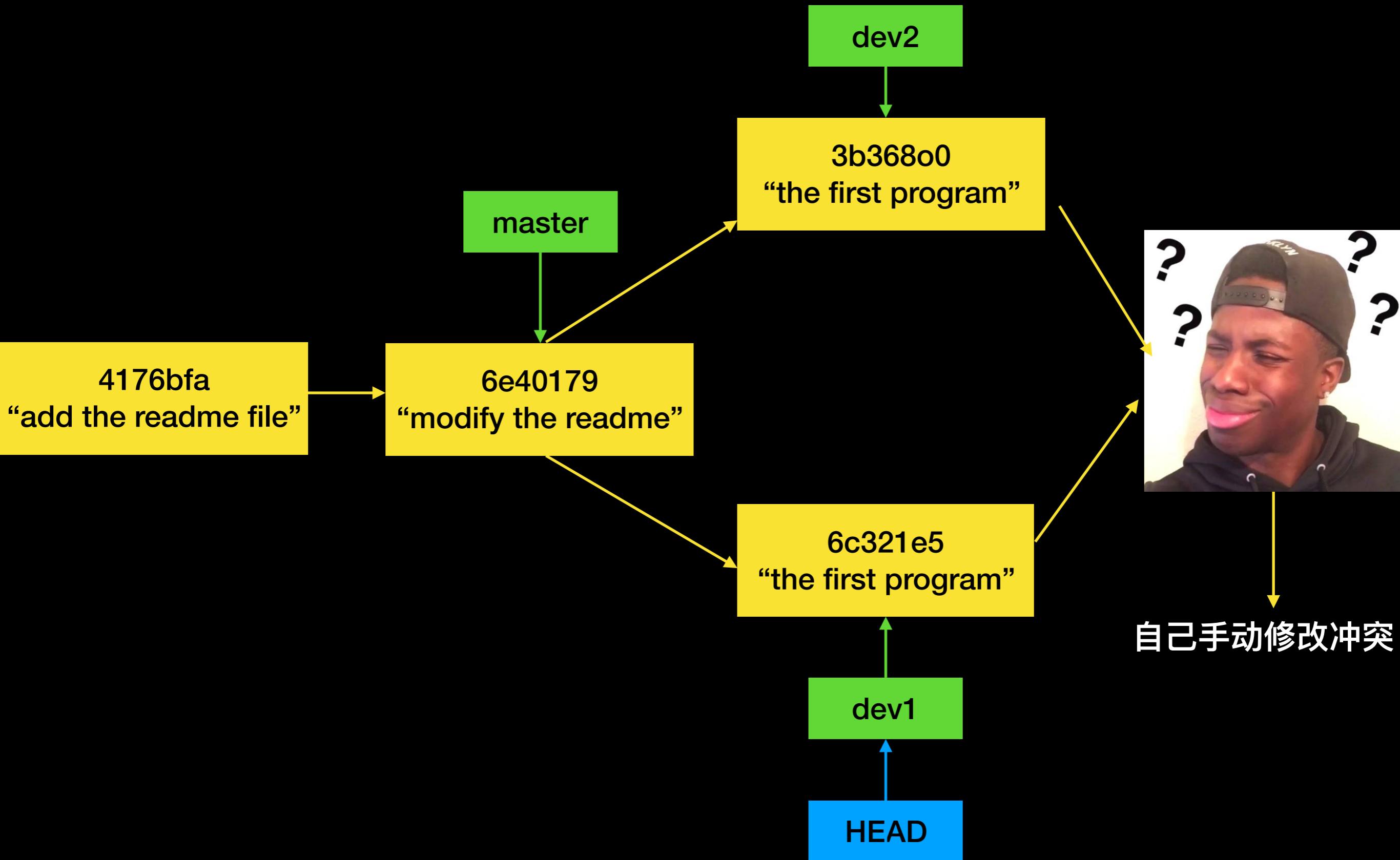
开发成功



不想开发了



# 冲突



## 4. 远程仓库



## Projects

Your projects Starred projects Explore projects

Filter by name...

Last updated

New project

All Personal

G K Wang / groupools Maintainer

★ 0 🔒  
updated 1 day ago

Blank project

Create from template

Import project

CI/CD for external repo

Project path

https://gitlab.com/wangk/

Project name

gittest

Want to house several dependent projects under the same namespace? [Create a group](#)

Project description (optional)

it is a test for git



Visibility Level ?

 🔒 Private

Project access must be granted explicitly to each user.

 🛡 Internal

The project can be accessed by any logged in user.

 🌎 Public

The project can be accessed without any authentication.

 Initialize repository with a README

Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.

Create project

Cancel

# 添加公钥

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master o [16:36:54]
```

```
$ ls ~/.ssh
```

```
config          id_rsa          id_rsa.pub      known_hosts
```

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master o [16:36:54]
```

```
$ ssh-keygen -t rsa -C "wkcosmology@gmail.com"
```

## 一路回车

The screenshot shows the GitLab User Settings interface. On the left sidebar, the 'SSH Keys' option is highlighted with a red box. The main content area is titled 'User Settings > SSH Keys' and displays the 'SSH Keys' section. It includes a brief description of what SSH keys are used for, a red box around the 'Add an SSH key' heading, and a text input field for pasting the public SSH key. Below the input field, there's a 'Title' input field and a 'Add key' button. At the bottom, it shows a summary of the added key: 'Your SSH keys (1)' followed by the email address and key fingerprint.

GitLab

Projects Groups Activity Milestones Snippets

User Settings

Profile Account Billing Applications Chat Access Tokens Emails Password Notifications

**SSH Keys**

SSH keys allow you to establish a secure connection between your computer and GitLab.

Add an SSH key

To add an SSH key you need to generate one or use an existing key.

Key

Paste your public SSH key, which is usually contained in the file '~/.ssh/id\_rsa.pub' and begins with 'ssh-rsa'. Don't use your private SSH key.

Typically starts with "ssh-rsa ..."

Title

e.g. My MacBook key

Name your individual key via a title

Add key

Your SSH keys (1)

wkcosmology@gmail.com a1:6b:54:84:bf:66:81:85:5a:4d:92:e6:cf:ff:cd:b9

last used: n/a

created 37 seconds ago

# 添加远程库

## push

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master o [16:31:18]
$ git remote add origin git@gitlab.com:wangk/gittest.git

# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master o [16:36:38]
$ git push -u origin master
Counting objects: 9, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (9/9), 753 bytes | 753.00 KiB/s, done.
Total 9 (delta 0), reused 0 (delta 0)
To gitlab.com:wangk/gittest.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
```

The screenshot shows a GitLab project page for 'gittest'. At the top, there's a navigation bar with 'K Wang > gittest > Details'. Below the header, the repository name 'gittest' is displayed with a 'G' logo, followed by a description 'it is a test for git' and 'Project ID: 8151457'. The main interface includes a star count (0), fork count (0), SSH link, and a copy button. Below this is a toolbar with buttons for 'Add Changelog', 'Add License', 'Add Contribution guide', 'Enable Auto DevOps', 'Add Kubernetes cluster', and 'Set up CI/CD'. A search bar at the bottom allows filtering by 'master' or 'gittest' and includes 'History', 'Find file', 'Web IDE', and a refresh icon. The commit history section shows a single commit by 'Kai Wang' from 5 hours ago, titled 'the first program'. The file list shows two files: 'README' and 'program1.py'. The 'README' file contains the text 'hello git and hello world'.

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master o [16:48:59]
```

```
$ vim program2.py
```

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master x [16:49:37]
```

```
$ git add program2.py
```

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master x [16:49:41]
```

```
$ git commit -m "the second program"
```

```
[master bf25850] the second program
```

```
Committer: Kai Wang <wangk@KaideMacBook-Pro.local>
```

Your name and email address were configured automatically based  
on your username and hostname. Please check that they are accurate.

You can suppress this message by setting them explicitly:

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

After doing this, you may fix the identity used for this commit with:

```
git commit --amend --reset-author
```

```
1 file changed, 1 insertion(+)
```

```
create mode 100644 program2.py
```

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master o [16:49:52]
```

```
$
```

K Wang > gittest > Repository

New file Template Choose type

Y master / program3.py

```
1 print("hello world 3)
```

Commit message

```
the third program
```

Target Branch

```
master
```

Commit changes

## 未和远端库同步，不能push

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master o [16:49:52]
$ git push origin master
To https://gitlab.com/wangk/gittest.git
  ! [rejected]          master -> master (fetch first)
error: failed to push some refs to 'https://gitlab.com/wangk/gittest.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

**git fetch**  
将远端库拉下来  
放在 origin/master  
(远端库)/(branch)

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master o [16:53:10]
$ git fetch origin
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From https://gitlab.com/wangk/gittest
  208753b..f24fa61  master      -> origin/master
```

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master o [16:55:09]
$ git checkout origin/master
Note: checking out 'origin/master'.
```

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may do so (now or later) by using -b with the checkout command again. Example:

```
git checkout -b <new-branch-name>
```

HEAD is now at f24fa61... the third program

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:f24fa61 o [16:55:55]
$ ls
README           program1.py     program3.py
```

## 将master和 origin/master 并合

```
# wangk @ KaideMacBook-Pro in ~/Desktop/gittest on git:master o [17:05:50]
$ git merge origin/master -m "synchronize the repo"
Updating 208753b..f24fa61
Fast-forward (no commit created; -m option ignored)
 program3.py | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 program3.py
```

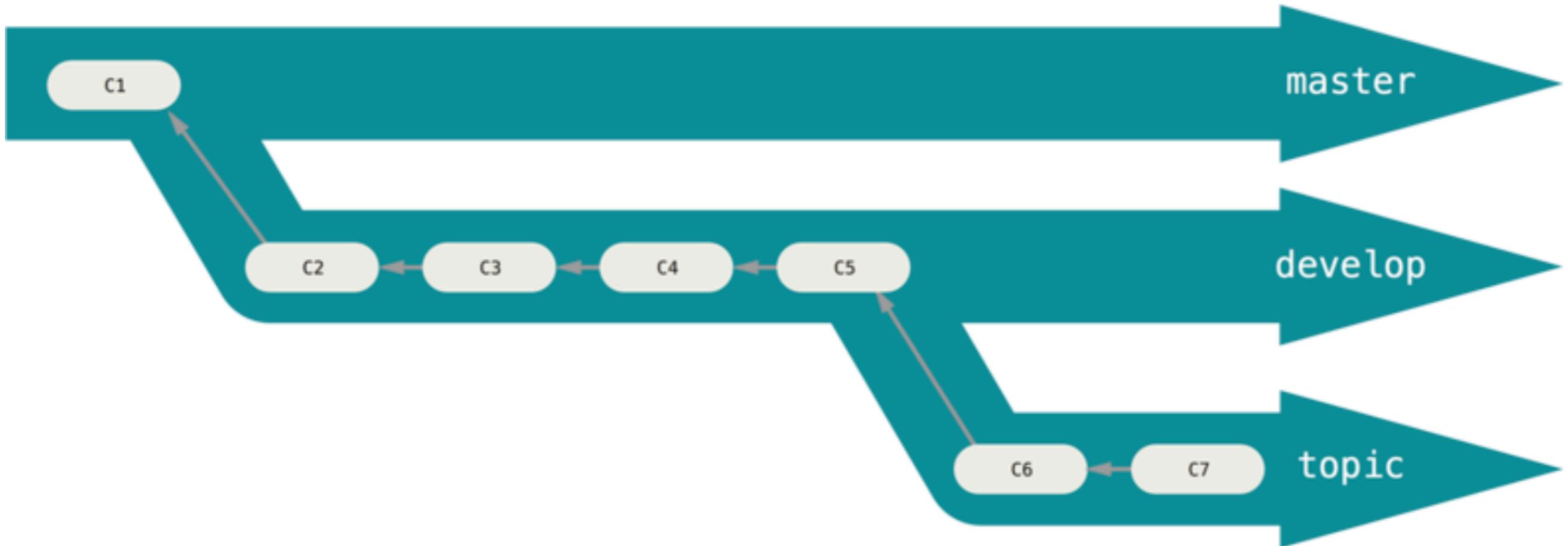
### >> git log

```
* bbf14e4 - (HEAD -> master) synchronize the repo (7 seconds ago) <Kai Wang>
|\ 
| * f24fa61 - (origin/master) the third program (19 minutes ago) <Kai Wang>
* | 0803e02 - the second program (25 seconds ago) <Kai Wang>
|/
* 208753b - modify the program (27 minutes ago) <Kai Wang>
* 6c321e5 - the first program (6 hours ago) <Kai Wang>
* 4176bfa - modify the readme (7 hours ago) <Kai Wang>
* 6e40179 - add the readme file (7 hours ago) <Kai Wang>
```

git pull = git fetch + git merge

## 5. 工作模式

# 1. 本地工作模式

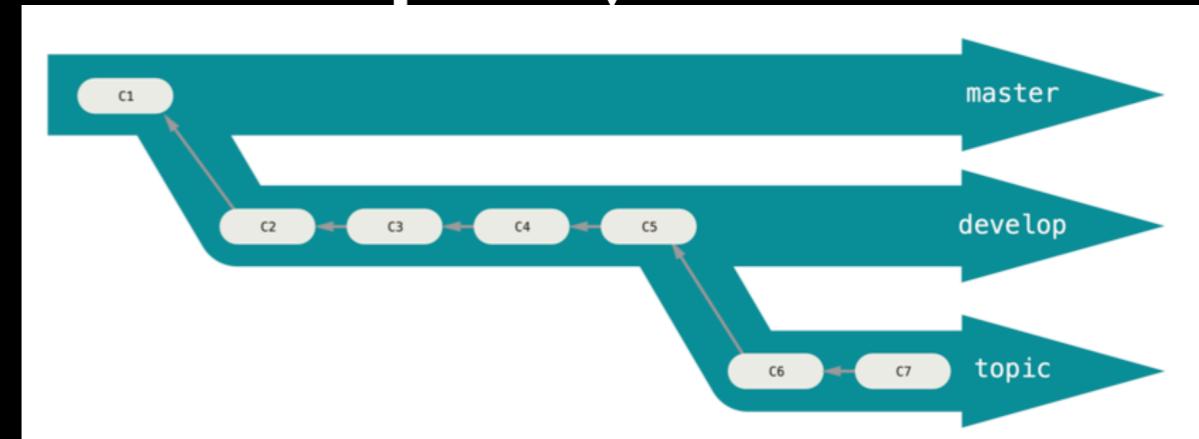
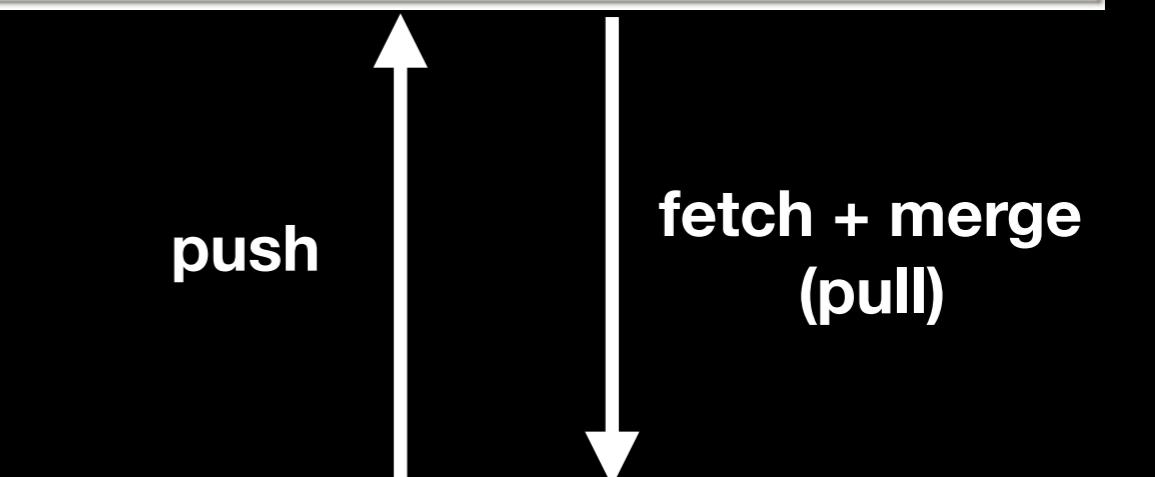
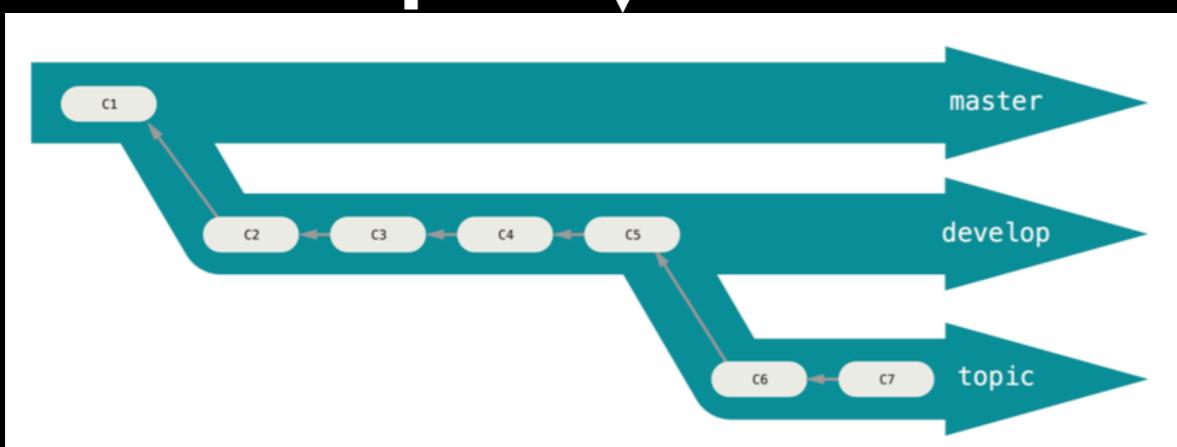
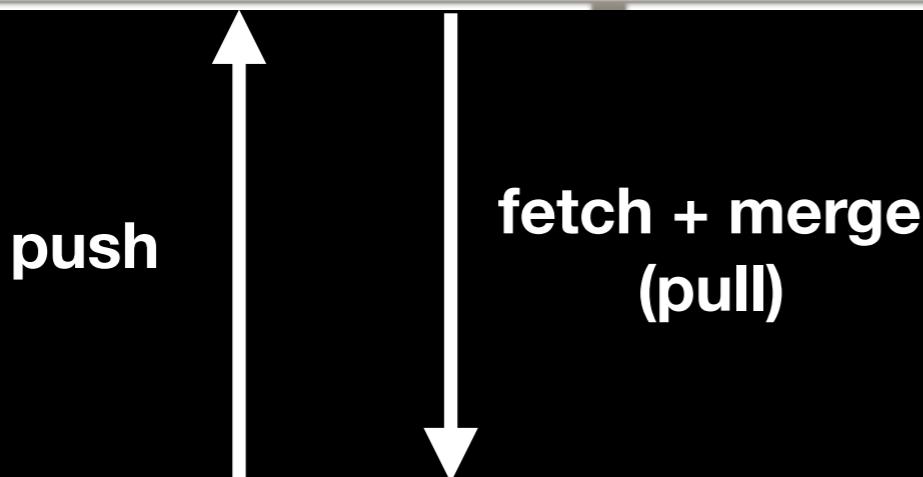
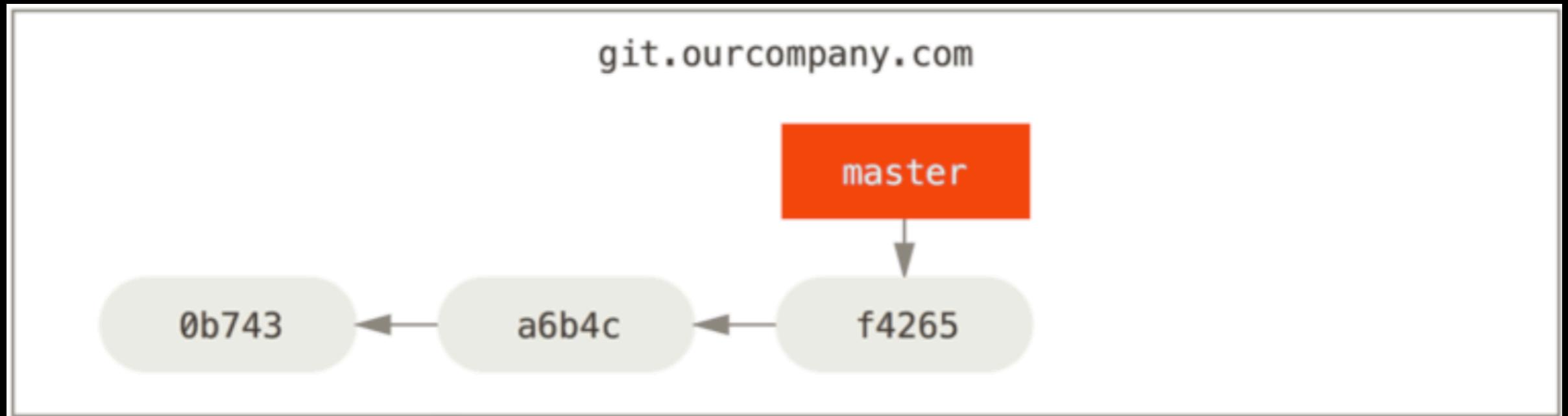


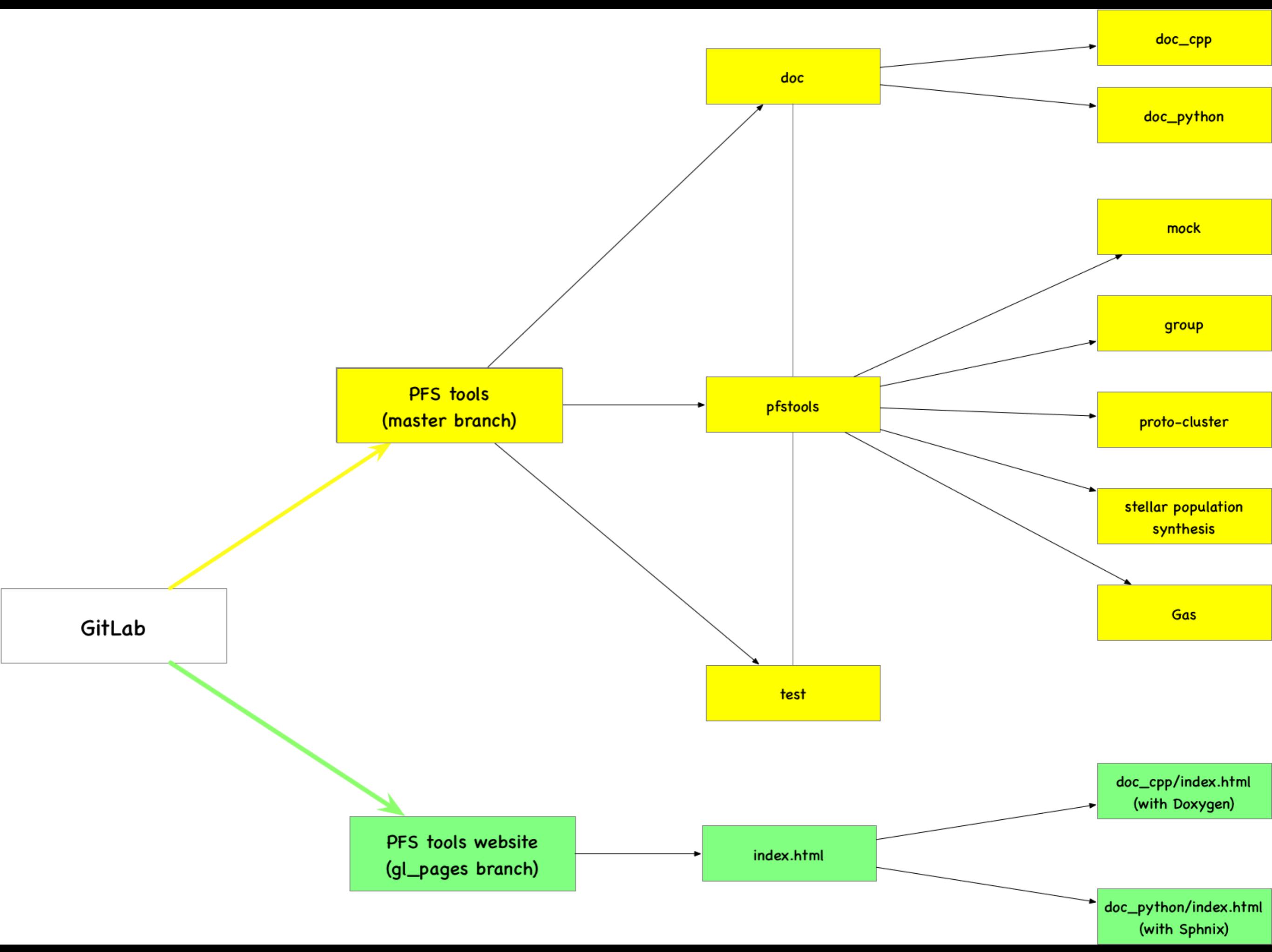
**master:** 最后的程序

**develop:** 增加功能, 修改功能

**topic:** debug, 测试

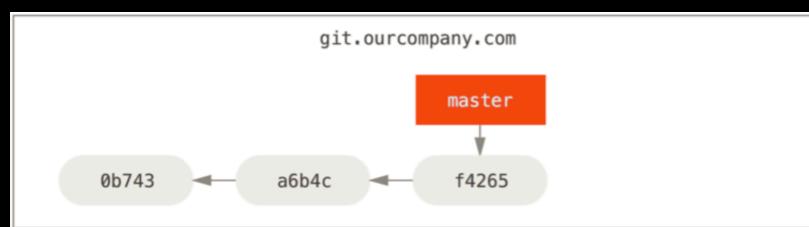
## 2. 合作工作模式





### 3. 本地-服务器同步 + 本地-远程合作模式

服务器仓库(用作备份和服务器跑程序)

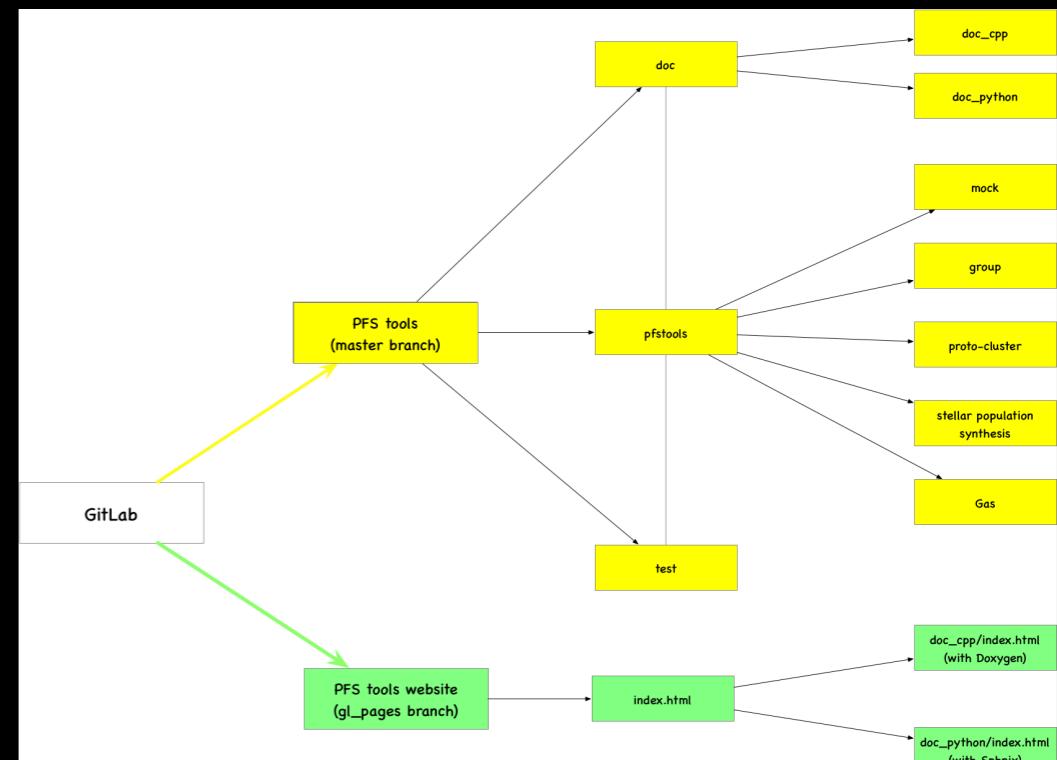


push

fetch + merge  
(pull)

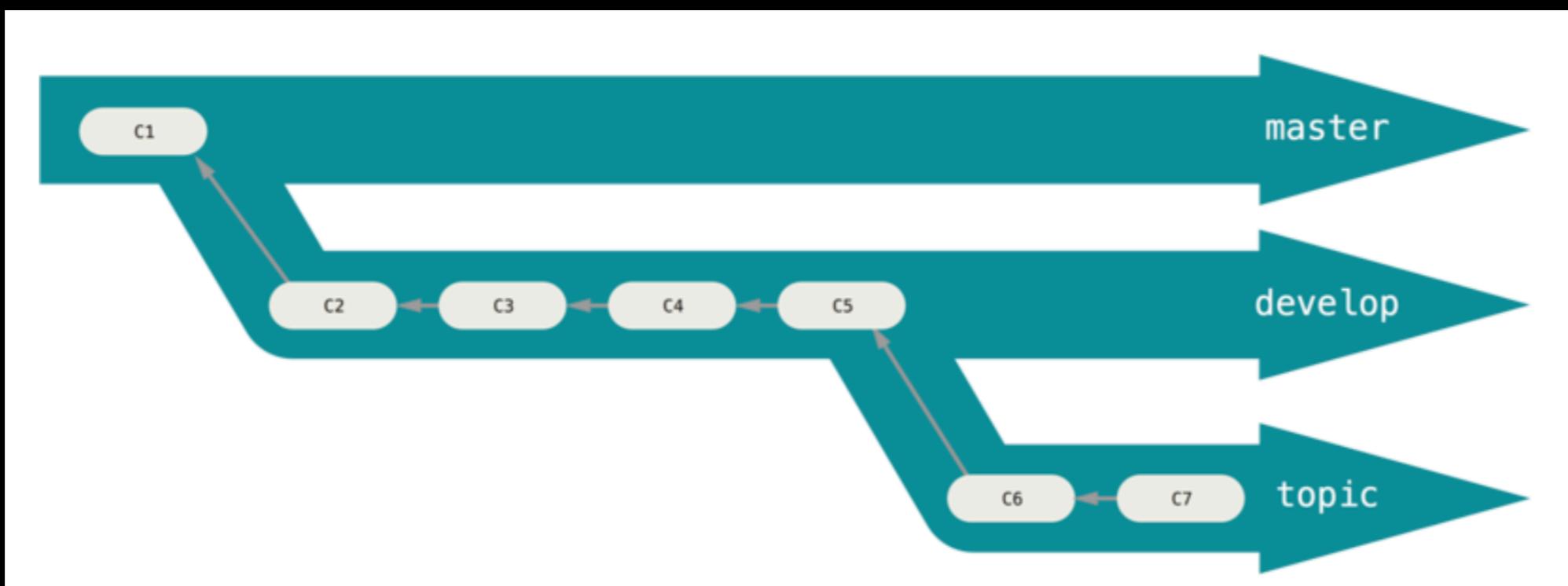
私人本地仓库

共有远程仓库 (用作共享)



push

fetch + merge  
(pull)



## 几个问题？

1. git有强大的版本管理功能，不用自己手动去做备份；
2. 本地写代码可以使用各种工具： vim, vscode, pycharm, sublime...  
但是本地机器性能有限，不能跑大型程序；
3. 服务器可以用来跑程序，但是网络有延迟， vim配置很垃圾，写代码不方便；
4. 本来可以只用服务器仓库作为共享和备份，但是考虑到服务器阶段性的维护和出问题，用gitlab保险一些；
5. 远程仓库： github, gitee(码云), gitlab。github私有仓库要钱，gitee是中文的，不international，所以选择gitlab；
6. 代码review一方面能够提升自己代码质量，“代码是写给人看的，其次才是来运行的”；另一方面也能学到别人代码中的技巧；
7. 写代码就是生产力。

**Rule #1**

**Create a Git repository for every new project**

**Rule #2**

**Create a new branch for every new feature**

**Rule #3**

**Use Pull Requests to merge code to Master**

# 配置服务器端同步本地代码

1. 进入服务器端git仓库所在目录，比如我要把/home/wangk/groupTools/pyscript设置成仓库

```
>> cd /home/wangk/groupTools/pyscript
```

```
>> git init
```

让仓库接受代码提交

```
>> git config receive.denyCurrentBranch ignore
```

禁止强制推送(可选)

```
>> git config --bool receive.denyNonFastForwards false
```

这样，远程仓库的ssh地址就是

```
ssh://wangk@venus.phys.tsinghua.edu.cn:/home/wangk/groupTools/pyscript/.git
```

2. 进入本地仓库地址，比如我的代码放在/Users/wangk/Documents/Project/9.GalaxyGroup/pyscript中

```
>> cd /Users/wangk/Documents/Project/9.GalaxyGroup/pyscript
```

```
>> git remote add server ssh://wangk@venus.phys.tsinghua.edu.cn:/home/wangk/groupTools/pyscript/.git
```

3. push代码到服务器

本地：

```
>> git push server master
```

服务器端(每次push之后必须到服务器端执行这个命令才可以看到改变)：

```
>> git reset --hard
```

4. pull代码到本地

```
>> git pull <远程主机名> <远程分支名>:<本地分支名>
```

或者

```
>> git fetch <远程主机名>/<远程分支名>
```

```
>> git merge <远程主机名>/<远程分支名>
```

廖雪峰的官方网站    编程    读书    JavaSE课程    JavaEE课程    数字货币    JavaScript教程    Python教程

目录



## Git教程

Git简介

安装Git

创建版本库

时光机穿梭

远程仓库

分支管理

标签管理

使用GitHub

使用码云

自定义Git

期末总结

## 关于作者

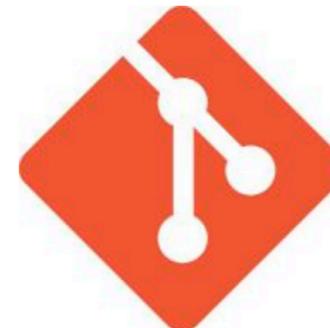


廖雪峰 V 北京朝阳区

+ 加关注

## Git教程

阅读: 6685772



# git

史上最浅显易懂的Git教程！

为什么要编写这个教程？因为我在学习Git的过程中，买过书，也在网上Google了不少资料，但都令人发指，就是简单得一笔带过，或者，只支离破碎地介绍Git的某几个命令，还有些教程是深入浅出，学完后能立刻上手的Git教程。

既然号称史上最浅显易懂的Git教程，那这个教程有什么让你怦然心动的特点呢？

首先，本教程绝对面向初学者，没有接触过版本控制概念的读者也可以轻松入门，不必担心起步难度。

其次，本教程实用性超强，边学边练，一点也不觉得枯燥。而且，你所学的Git命令是“充分且必要”的，能够完成你的工作。

文字+图片还看不明白？有视频！！！

本教程只会让你成为Git用户，不会让你成为Git专家。很多Git命令只有那些专家才明白（事实上我也明白一些命令可能你一辈子都不会用到）。既然Git是一个工具，就没必要把时间浪费在那些“高级”但几乎永远用不到，到时候再自行Google或者请教专家也未迟。

# Pro Git

EVERYTHING YOU NEED TO  
KNOW ABOUT GIT

Scott Chacon and Ben Straub

Apress®

## 6. 自动生成文档

**Page Contents****Group Finder API**

- [Classes](#)
- [Detailed Descriptions](#)

**Previous topic**[Source code notes of the Voronoi-Delaunay Method](#)**This Page**[Show Source](#)

Search

# Group Finder API

## Classes

[`groupTools.group\_finder.fof.FoFGroupFinder`](#)

This class is to apply the friend-of-friend group finder to find groups in the galaxy survey.

[`groupTools.group\_finder.vdm.VDMGroupFinder`](#)

This is an implementation of Voronoi-Delaunay Method to find galaxy groups.

## Detailed Descriptions

```
class groupTools.group_finder.fof.FoFGroupFinder(ra, dec, z, num_density, b, lmax, ratio,  
co_dist=None)
```

[\[source\]](#)

This class is to apply the friend-of-friend group finder to find groups in the galaxy survey.

### Methods

[`group\_id\_f\(\)`](#) output the found group id[`multiplicity\(\)`](#) return the multiplicity of the group catalogue[`n\_groups\(\)`](#) return the number of found groups[`\_\_init\_\_\(ra, dec, z, num\_density, b, lmax, ratio, co\_dist=None\)`](#)[\[source\]](#)

read the data and prepare for group finding

**Parameters:** `ra : array_like`

length N\_gal numpy array contains RA of galaxies

`dec : array_like`

length N\_gal numpy array contains DEC of galaxies

`z : array_like`

length N\_gal numpy array contains redshift of galaxies

`co_dist : array_like, optional`

length N\_gal numpy array contains comoving distance of galaxies

`num_density : array_like or int`

the average number density of the galaxies depending on redshift

`b : array_like or float`

linked length of on the projected direction

`lmax : array_like or float`

maximum linked length

`ratio : array_like or float`

ratio between the linked length on the line of sight and projected direction

[`group\_id\_f\(\)`](#)[\[source\]](#)

output the found group id

**Returns:** `group_id_f : array_like`

length N\_gal numpy array contains the found group ids

```
def foo(var1, var2, long_var_name='hi'):
    """A one-line summary that does not use variable names or the
    function name.
```

Several sentences providing an extended description. Refer to variables using back-ticks, e.g. `var`.

#### Parameters

-----

##### var1 : array\_like

Array\_like means all those objects -- lists, nested lists, etc. -- that can be converted to an array. We can also refer to variables like `var1`.

##### var2 : int

The type above can either refer to an actual Python type (e.g. ``int``), or describe the type of the variable in more detail, e.g. ``(N,) ndarray`` or ``array\_like``.

##### long\_var\_name : {'hi', 'ho'}, optional

Choices in brackets, default first when optional.

#### Returns

-----

##### type

Explanation of anonymous return value of type ``type``.

##### describe : type

Explanation of return value named `describe`.

##### out : type

Explanation of `out`.

##### type\_without\_description

#### Other Parameters

-----

##### only\_seldom\_used\_keywords : type

Explanation

##### common\_parameters\_listed\_above : type

Explanation

#### Raises

-----

##### BadException

Because you shouldn't have done that.

#### See Also

-----

##### otherfunc : relationship (optional)

newfunc : Relationship (optional), which could be fairly long, in which case the line wraps here.

##### thirdfunc, fourthfunc, fifthfunc

## example.foo(var1, var2, long\_var\_name='hi')

A one-line summary that does not use variable names or the function name.

Several sentences providing an extended description. Refer to variables using back-ticks, e.g. var.

#### Parameters:

##### var1 : array\_like

Array\_like means all those objects – lists, nested lists, etc. – that can be converted to an array. We can also refer to variables like var1.

##### var2 : int

The type above can either refer to an actual Python type (e.g. `int`), or describe the type of the variable in more detail, e.g. `(N, ) ndarray` or `array_like`.

##### long\_var\_name : {'hi', 'ho'}, optional

Choices in brackets, default first when optional.

#### Returns:

##### type

Explanation of anonymous return value of type `type`.

##### describe : type

Explanation of return value named `describe`.

##### out : type

Explanation of `out`.

##### type\_without\_description

#### Other Parameters:

##### only\_seldom\_used\_keywords : type

Explanation

##### common\_parameters\_listed\_above : type

Explanation

#### Raises:

##### BadException

Because you shouldn't have done that.

#### See also:

`otherfunc` relationship (optional)

`newfunc` Relationship (optional), which could be fairly long, in which case the line wraps here.

`thirdfunc, fourthfunc, fifthfunc`

#### Notes

Notes about the implementation algorithm (if needed).

This can have multiple paragraphs.

You may include some math:

$$X(e^{j\omega}) = x(n)e^{-j\omega n}$$

And even use a Greek symbol like  $\omega$  inline.

#### References

Cite the relevant literature, e.g. [1]. You may also cite these references in the notes section above.

- [1] (1, 2) O. McNoleg, "The integration of GIS, remote sensing, expert systems and adaptive co-kriging for environmental habitat modelling of the Highland Haggis using object-oriented, fuzzy-logic and neural-network techniques," *Computers & Geosciences*, vol. 22, pp. 585-588, 1996.

Mac 福利时间

# 1. Dash

- 文档查阅
- 代码片段

The screenshot shows a web browser window with the URL `numpy.histogram` in the address bar. The page content is the documentation for the `numpy.histogram` function. The sidebar on the left lists other functions like `histogram`, `histogram2d`, `histogramdd`, `histogram_bin_edges`, `Google`, and `Stack Overflow`. The main content area has a header `numpy.histogram` and a sub-header `numpy.histogram`. It includes the function signature `numpy.histogram(a, bins=10, range=None, normed=None, weights=None, density=None)`, a brief description "Compute the histogram of a set of data.", and detailed parameters for `a`, `bins`, `range`, `normed`, `weights`, and `density`. It also describes the returns `hist` and `bin_edges`. A "See also" section at the bottom lists related functions: `histogramdd`, `bincount`, `searchsorted`, `digitize`, and `histogram_bin_edges`.

`numpy.histogram(a, bins=10, range=None, normed=None, weights=None, density=None)`

Compute the histogram of a set of data.

**Parameters:**

- `a : array_like`**  
Input data. The histogram is computed over the flattened array.
- `bins : int or sequence of scalars or str, optional`**  
If `bins` is an int, it defines the number of equal-width bins in the given range (10, by default). If `bins` is a sequence, it defines the bin edges, including the rightmost edge, allowing for non-uniform bin widths.  
*New in version 1.11.0.*  
If `bins` is a string, it defines the method used to calculate the optimal bin width, as defined by `histogram_bin_edges`.
- `range : (float, float), optional`**  
The lower and upper range of the bins. If not provided, range is simply `(a.min(), a.max())`. Values outside the range are ignored. The first element of the range must be less than or equal to the second. `range` affects the automatic bin computation as well. While bin width is computed to be optimal based on the actual data within `range`, the bin count will fill the entire range including portions containing no data.
- `normed : bool, optional`**  
*Deprecated since version 1.6.0.*  
This is equivalent to the `density` argument, but produces incorrect results for unequal bin widths. It should not be used.  
*Changed in version 1.15.0:* DeprecationWarnings are actually emitted.
- `weights : array_like, optional`**  
An array of weights, of the same shape as `a`. Each value in `a` only contributes its associated weight towards the bin count (instead of 1). If `density` is True, the weights are normalized, so that the integral of the density over the range remains 1.
- `density : bool, optional`**  
If `False`, the result will contain the number of samples in each bin. If `True`, the result is the value of the probability `density` function at the bin, normalized such that the `integral` over the range is 1. Note that the sum of the histogram values will not be equal to 1 unless bins of unity width are chosen; it is not a probability `mass` function.  
Overrides the `normed` keyword if given.

**Returns:**

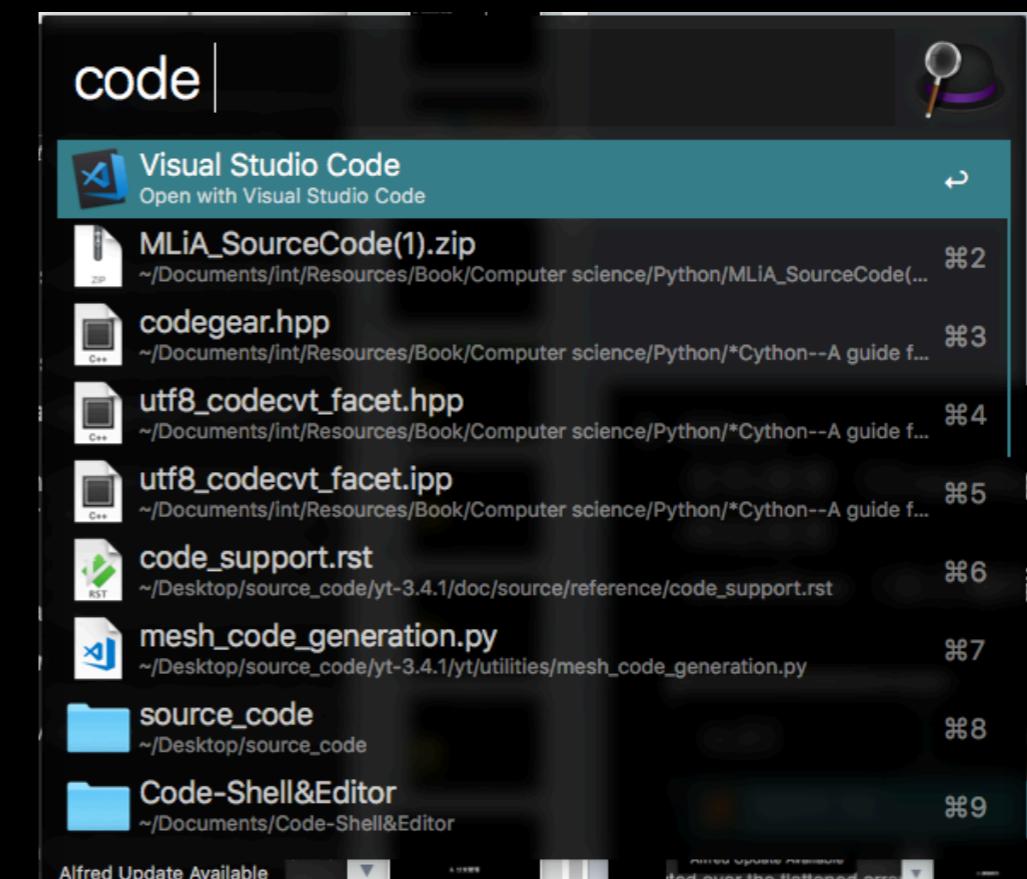
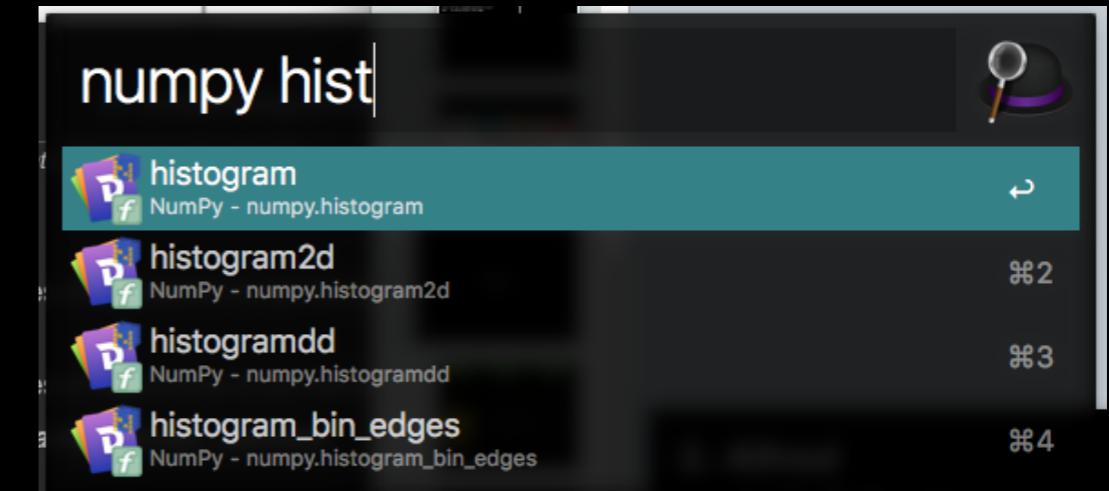
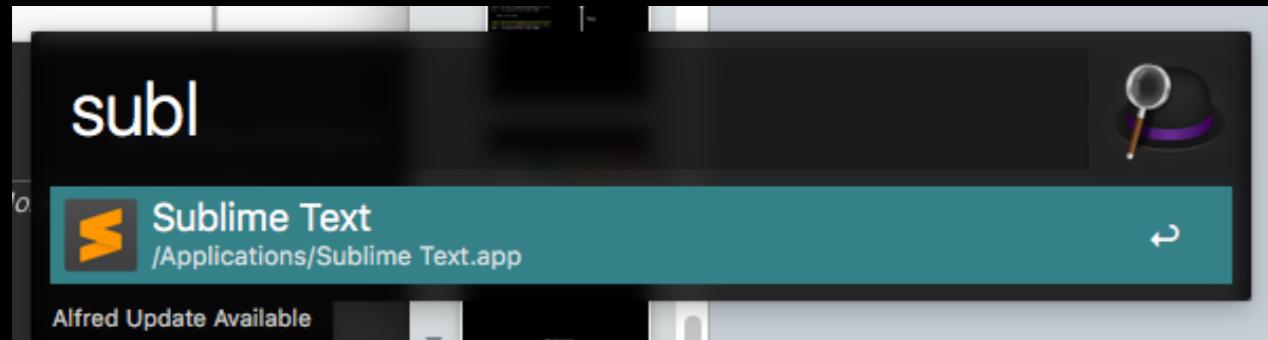
- `hist : array`**  
The values of the histogram. See `density` and `weights` for a description of the possible semantics.
- `bin_edges : array of dtype float`**  
Return the bin edges `(length(hist)+1)`.

**See also:**

`histogramdd`, `bincount`, `searchsorted`, `digitize`, `histogram_bin_edges`

## 2. Alfred

- 全机搜索（比spotlight快很多）
- 网站搜索
- workflow（自己编程实现很多功能，或者用别人编好的）



# 3. iTerm2

## - 分屏

# 快捷呼出

```
# wangk @ KaitdeMacBook-Pro in ~/Documents/Project/9.GalaxyGroup/pyscript/test on git:master x [11:04:43]
$ ls
config_para.py      data_generator.py      fof_pfs.py      test.ipynb      vdm_pfs.py

# wangk @ KaitdeMacBook-Pro in ~/Documents/Project/9.GalaxyGroup/pyscript/test on git:master x [15:24:04]
$ cd
# wangk @ KaitdeMacBook-Pro in ~ [15:24:05]
$ cd Desktop

# wangk @ KaitdeMacBook-Pro in ~/Desktop [15:24:07]
$ ls
2018-08-30.pptx    8.27.pptx    8.30.key    books_tbd    courses    git.key    gitest    report.key    source_code    tester    ~$演示文稿5.pptx
# wangk @ KaitdeMacBook-Pro in ~/Desktop [15:24:09]
$ 
```

Fri, 30 Aug 2018

[1] arXiv:1808.09909 [pdf, ps, other]

### Massive Primordial Black Holes in Contemporary and Young Universe (old predictions and new data)

A.D. Dolgov

Comments: 10 pages, Invited talk at The Conference on Particles and Cosmology at the Nanyang Executive Centre from 5 to 9 March 2018 at NTU, Singapore

Subjects: Cosmology and Nongalactic Astrophysics (astro-ph.CO); High Energy Physics – Phenomenology (hep-ph)

[2] arXiv:1808.09823 [pdf, ps, other]

### Problems with the dark matter and dark energy hypotheses, and alternative ideas

Martin Lopez-Corredoira

Comments: 16 pages, 100 references, published in the proceedings of the Conference "Cosmology on Small Scales 2018: Dark Matter Problem and Selected Controversies in Cosmology" (Prague, September 26–29, 2018); this http URL arXiv admin note: substantial text overlap with arXiv:1701.08720

Journal-ref: in: Cosmology on Small Scales 2018. Dark Matter Problem and Selected Controversies in Cosmology, M. Krizek, Y. V. Dumin (Eds.), Inst. Mathematics, Czech Acad. Sciences, Prague, pp. 14–29 (2018)

Subjects: Cosmology and Nongalactic Astrophysics (astro-ph.CO); Astrophysics of Galaxies (astro-ph.GA); History and Philosophy of Physics (physics.hist-ph)

[3] arXiv:1808.09743 [pdf, other]

### Measuring our velocity from fluctuations in number counts

Nidhi Pant, Aditya Rotti, Carlos A.P. Bengaly, Roy Maartens

Comments: 15 pages, 7 figures

Subjects: Cosmology and Nongalactic Astrophysics (astro-ph.CO)

[4] arXiv:1808.09706 [pdf, other]

### Inflation, (P)reheating and Neutrino Anomalies: Production of Sterile Neutrinos with Secret Interactions

Arnab Paul, Anish Ghoshal, Arindam Chatterjee, Supratik Pal

Subjects: Cosmology and Nongalactic Astrophysics (astro-ph.CO); High Energy Physics – Phenomenology (hep-ph)

[5] arXiv:1808.09516 [pdf, other]

### Magnification bias in the shear-ratio test: a viable mitigation strategy

Sandra Unruh, Peter Schneider, Stefan Hilbert

Comments: 9 pages, 7 figures

Subjects: Cosmology and Nongalactic Astrophysics (astro-ph.CO)

[6] arXiv:1808.09815 (cross-list from gr-qc) [pdf, ps, other]

### Comment on Eur. Phys. J. Plus 133, 261 (2018) by Kholmetskii et al

Christian Corda

Comments: 8 pages, accepted for publication in The European Physical Journal Plus

Subjects: General Relativity and Quantum Cosmology (gr-qc); Cosmology and Nongalactic Astrophysics (astro-ph.CO); High Energy Physics – Theory (hep-th); Classical Physics (physics.class-ph)

[7] arXiv:1808.09643 (cross-list from gr-qc) [pdf, other]

### Pre-inflationary dynamics of Starobinsky and $\alpha$ -Attractor Inflation in Loop Quantum Brans-Dick Cosmology

Wei-Jian Jin, Yongge Ma, Tao Zhu

Comments: 15 pages, 8 figures, 3 tables

Subjects: General Relativity and Quantum Cosmology (gr-qc); Cosmology and Nongalactic Astrophysics (astro-ph.CO); High Energy Physics – Theory (hep-th)

[8] arXiv:1808.09456 (cross-list from hep-ph) [pdf, ps, other]

### Axion absorption and the spin temperature of primordial hydrogen

Adrien Auriol, Sacha Davidson, Georg Raffelt

Comments: 7 pages

### 3. zsh & Oh My Zsh

**zsh** 是一种**shell**解释器，目前一般系统默认的是**bash**

**Oh My Zsh**是对**zsh**的一种配置方式

- 好看

```
# wangk @ KaideMacBook-Pro in ~/Documents/Project/9.GalaxyGroup/pyscript on git:master ✘ [15:31:03]
$



Last login: Wed Aug 29 15:05:38 2018 from 101.5.232.110
[wangk@venus ~]$ cd group_pfs/
data/          data_playground/  mock/          pyscript/      relation_fitting/  result/      src/
[wangk@venus ~]$ cd group_pfs/
data/          data_playground/  mock/          pyscript/      relation_fitting/  result/      src/
[wangk@venus ~]$ cd group_pfs/pyscript/
[wangk@venus pyscript]$ ls
add_photo_z.py  confia_para.py  data_filter.py  data_generator.py  fof_pfs.py  fof.py   fof.pyc  group_match.py  group_match.pyc  muilt
```

- 自动补全功能强大：自动匹配，可以忽略大小写

## chucknorris

Maintainers: [apjanke maff](#)

Adds Chuck Norris fortune file from <http://www.k-lug.org/~kessler/projects.html>. Depends on fortune (and cowsay if using `chuck_cow`) being installed (available via homebrew, apt, ...). Perfectly suitable as MOTD.

Command	Description
<code>chuck</code>	print random Chuck Norris quote
<code>chuck_cow</code>	print quote in cowthink

Example: output of `chuck_cow` command

```
Last login: Fri Jan 30 23:12:26 on ttys001
```

```
( When Chuck Norris plays Monopoly, it )
( affects the actual world economy. )
```



## wd

Maintainer: [mfaerevaag](#)

`wd` (warp directory) lets you jump to custom directories in zsh, without using `cd`. Why? Because `cd` seems inefficient when the folder is frequently visited or has a long path. [Source](#)

Print usage with no opts or the `help` argument.

Examples:

Add warp point to current working directory:

```
wd add test
```

If a warp point with the same name exists, use `add!` to overwrite it.

Examples:

From an other directory, warp to test with:

```
wd test
```

# Rafael Bodill's Neo/vim Config

## 4. neovim +

Lean mean Neo/vim machine, 30-45ms startup time.

Best with Neovim or Vim 8 with +python3 extensions enabled.

### 自动补全

```
419
420     def get_grp_center(self, num_iteration, center_kind, rg_kind):
421         """
422             get the group center position(RA, DEC) using the information from
423             photometric data for low sampling rate galaxy survey.
424
425         Parameters
426         -----
427         num_iteration : int
428             number of iteration to determine the group center and radius
429         center_kind : str
430             kind of group center you choose to use in estimation
431         rg_kind assign_phot_gals(...)
432             kind data
433             generate_frac_map(...)
434         for i in get_grp_center(...)
435             self index
436                 _assign_vor_area(gals)
437                 _frac_map
438                 _gals
439                 _grp_cata
440                 _hm
441             self _initialize_grps(rg_kind, ...)
442             self _num_edges
443             self _phot_gals
444             prob _phot_z_error
445             self _r_edges
446             self.

INSERT +31 ~2 -11 <find_mmg.py[+] get_grp_center python utf-8[unix] 100% ≡ 446/446 ln : 18
W10: Warning: Changing a readonly file
```

### 函数签名

```
440             how="outer")
441             self._update_grp_center_no_weighted(self._gals)
442             self._update_rg_real(self._gals)
443             self.generate_frac_map()
444             prob, grp_id_f = self.assign_phot_gals()
445             self._phot_gals["group_id_f"] = grp_id_f[prob > 0.5]
446             self.get_grp_center()

INSERT +31 ~2 -11 </find_mmg.py[+] get_grp_center python utf-8[unix] 100% ≡ 446/446 ln : 33
get_grp_center(num_iteration, center_kind, rg_kind)
```

## 目录树

## 函数列表

py script group\_pr/find\_mmg.py

```
 403     sum_p = np.sum(grps_)
 404     prod1m = np.prod(1 - grps_)
 405     prob_i = grps_[max_id] * (1 - prod1m) / sum_p
 406     if prob_kind == "max":
 407         prob_i = grps_[max_id]
 408         max_prob.append(prob_i)
 409         assign_grp_id.append(
 410             self._grp_cata.loc[
 411                 prob.indices[prob.indptr[i]: prob.indptr[i + 1]][max_id],
 412                 "group_id_f"])
 413     else:
 414         max_prob.append(np.nan)
 415         assign_grp_id.append(-1)
 416     max_prob = np.array(max_prob)
 417     assign_grp_id = np.array(assign_grp_id)
 418     return max_prob, assign_grp_id
 419
 420 def get_grp_center(self, num_iteration, center_kind, rg_kind):
 421     """
 422     get the group center position(RA, DEC) using the information from
 423     photometric data for low sampling rate galaxy survey.
 424
 425     Parameters
 426     -----
 427     num_iteration : int
 428         number of iteration to determine the group center and radius
 429     center_kind : str
 430         kind of group center you choose to use in estimation
 431     rg_kind : str
 432         kind of group radius you choose to use in estimation
 433     """
 434     for i in range(num_iteration):
 435         self._gals = pd.merge(
 436             self._spec_gals.loc[:, "ra dec group_id_f".split()],
 437             self._phot_gals.loc[
 438                 self._phot_gals["group_id_f"] != -1,
 439                 "ra dec group_id_f".split()],
 440                 how="outer")
 441         self._update_grp_center_no_weighted(self._gals)
 442         self._update_rg_real(self._gals)
 443         self.generate_frac_map()
 444         prob, grp_id_f = self.assign_phot_gals()
 445         self._phot_gals["group_id_f"] = grp_id_f[prob > 0.5]
```

" Press <F1>, ? for help

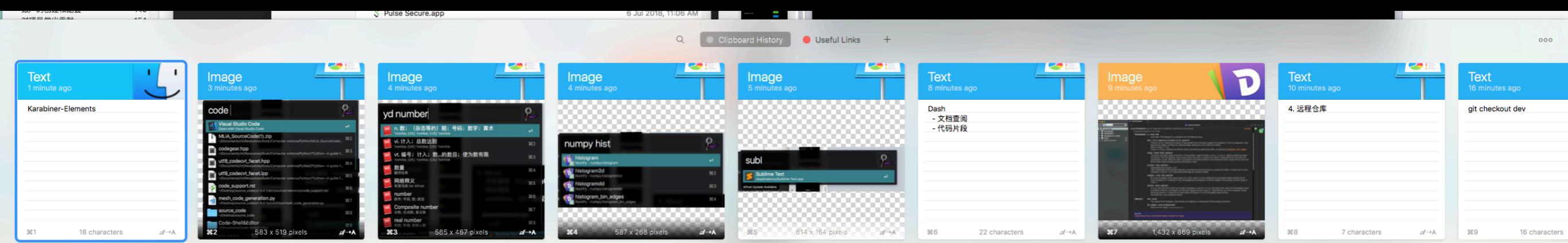
► imports

AssignPhotoGalToGrp : class

- +\_\_init\_\_ : function
- +assign\_vor\_area : function
- +initialize\_grps : function
- +update\_grp\_center\_no\_weighted : fu
- +update\_grp\_center\_vor\_weighted : f
- +update\_grp\_radius\_rms : function
- +update\_rg\_fudge : function
- +update\_rg\_real : function
- +assign\_phot\_gals : function
- +generate\_frac\_map : function
- +get\_grp\_center : function

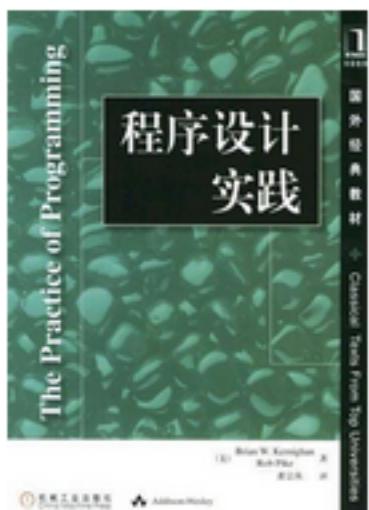
**TagbarToggle**

- . **sizeup** 屏幕快速规划
- . **Karabiner-Elements** 键位设置（中英文切换-> control）
- . **OmniFocus** 事务清单，规划（比较复杂）
- . **paste** 剪切板扩展
- . **pocket** 留着有时间看
- . **CheatSheet** 快捷键查看
- . **flux** 屏幕颜色自动调整，主要是晚上，护眼
- . **file cabinet pro** 放临时文件的地方，不要再堆在桌面上了
- . **OmniGraffle** 画流程图的（超级强大）



# 推荐书籍时间

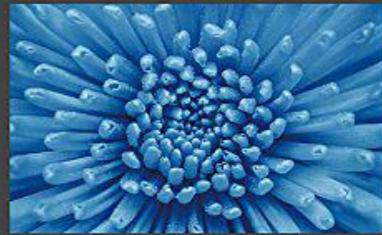
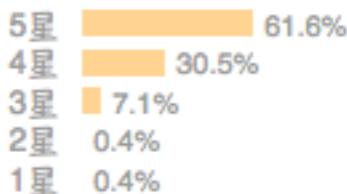
# 程序设计实践



作者: [美] Brian W. Kernighan / Rob Pike  
出版社: 机械工业出版社  
原作名: The Practice of Programming  
译者: 裴宗燕  
出版年: 2000-8  
页数: 221  
定价: 20.00元  
装帧: 平装  
丛书: 计算机科学丛书  
ISBN: 9787111075738

豆瓣评分

9.1 453人评价



## 程序设计实践 (英文版)

[美] Brian W. Kernighan / Rob Pike著  
**The Practice of  
Programming**

中国工信出版集团 人民邮电出版社  
POSTS & TELECOM PRESS

- Chapter 1: Style / 风格
- Chapter 2: Algorithms and Data Structures / 算法与数据结构
- Chapter 3: Design and Implementation / 设计与实现
- Chapter 4: Interfaces / 接口
- Chapter 5: Debugging / 调试
- Chapter 6: Testing / 测试
- Chapter 7: Performance / 性能
- Chapter 8: Portability / 可移植性
- Chapter 9: Notation / 记法

# 流畅的Python



作者: [巴西] Luciano Ramalho

出版社: 人民邮电出版社

原作名: Fluent Python

译者: 安道 / 吴珂

出版年: 2017-5-15

页数: 628

定价: 139元

装帧: 平装

ISBN: 9787115454157

豆瓣评分

9.4



244人评价

5星 75.0%

4星 20.1%

3星 4.1%

2星 0.4%

1星 0.4%

数据结构: list, tuple, dict, set这些内置类型

函数: 一等函数, 函数式编程, 装饰器和闭包

面向对象: 传值和传址, 垃圾回收机制, python中的继承

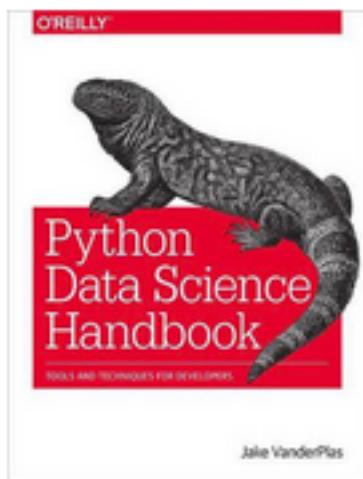
控制流程: 迭代器, 生成器, 上下文管理器, 协程, 并发

实例丰富, 不会有理解难度

风格幽默, 八卦也多

翻译质量很高

# Python Data Science Handbook



作者: Jake VanderPlas

出版社: O'Reilly Media

副标题: Tools and Techniques for Developers

出版年: 2016-6-25

页数: 500

定价: USD 59.99

装帧: Paperback

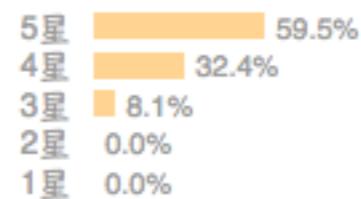
ISBN: 9781491912058

豆瓣评分

9.1

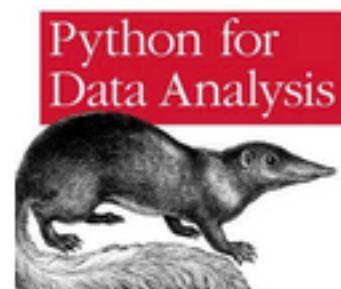


37人评价



numpy  
pandas  
matplotlib  
machine learning

# Python for Data Analysis



作者: Wes McKinney

出版社: O'Reilly Media

副标题: Data Wrangling with Pandas, NumPy, and IPython

出版年: 2012-11-1

页数: 466

定价: USD 39.99

装帧: Paperback

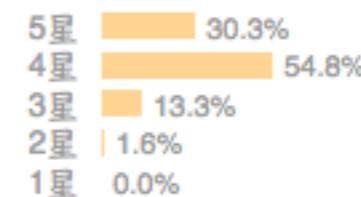
ISBN: 9781449319793

豆瓣评分

8.4

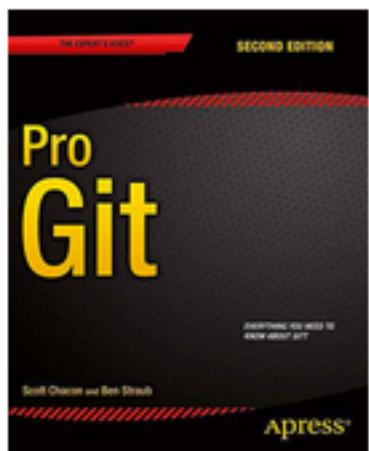


188人评价



主要讲pandas

# Pro Git (Second Edition)



作者: Scott Chacon / Ben Straub

出版社: Apress

出版年: 2014-11-9

页数: 350

定价: USD 59.99

装帧: Paperback

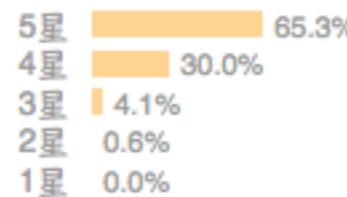
ISBN: 9781484200773

豆瓣评分

9.3



170人评价



图很多，很形象

中文版 <https://github.com/bingohuang/progit2-gitbook>

## Pro Git, 第二版, 简体中文

- 作为Git官方推荐书籍，《Pro Git》值得Git初学者和爱好者认真阅读一遍。
- 本人很喜欢 GitBook 的阅读书籍模式，喜爱开源共享。当前GitBook网站上只有《Pro Git》第一版，故我将《Pro Git》第二版（简体中文版）整理迁移至此，方便大家统一阅读学习。
- [《Pro Git》第二版书籍@GitBook](#)
- [《Pro Git》第二版源码@Github](#)：欢迎做 fork - 修改 - pull request
- GitBook生成的 PDF 格式书籍，对中文字体支持不好，这边上传完美 PDF 版本至百度网盘中，方便大家本地下载查看。  
[感谢  
`李志霞` 同学的建议和文件]
  - [百度网盘-《Pro Git 第二版 简体中文》](#)
- 《Pro Git》第二版原书格式是 AsciiDoc，和常用的 Markdown 格式略有不同，还好 GitBook 2.0.0 之后已支持，方便我整理到 GitBook 中，其中 AsciiDoc 帮助文档[点此](#)。
- 感谢原著作者及翻译者：[书籍官网地址](#)
- 我的邮箱[bingo727@gmail.com](mailto:bingo727@gmail.com)，欢迎一起交流~