

Git V2.7.4

remote

git remote

当前配置的每一个远程仓库服务器的简写

git remote -v

显示需要读写远程仓库使用的 Git 保存的简写与其对应的 URL

git remote show origin

查看远程仓库的更多信息

git remote add myClone https://github.com/Kuri-su/Portflow-Monitor.git

添加远程仓库，并给远程库起名

git remote rename origin ori

重命名远程仓库

git remote rm origin

移除远程仓库

clone

git clone https://url.com/Kuri-su.git

克隆现有仓库，在当前目录新建的 Portflow-Monitor 文件夹下

git clone https://url.com/Kuri-su.git test

克隆现有仓库，在当前目录新建的 test 目录下

pull

git pull origin master

拉取分支最新内容 pull = fetch + merge

git pull --rebase

会以线性进行

fetch

git fetch origin master

从远程仓库中拉取（不会自动合并）

checkout

git checkout master

切换分支

git checkout -b test2

创建并切换到分支

git checkout -- readme.md

撤销对文件的修改

git checkout --track origin/dev

跟踪分支，并在本地创建同名分支  
(git checkout -b serverfix origin/serverfix) 的简写  
不过git checkout -b sf origin/serverfix 可以使本地分支和远程分支不同名

自由主题

branch

git branch test

创建分支

git branch still-a-branch 38b7da5e

从任意提交创建分支

git branch still-a-branch older-branch

从现有分支创建分支

git branch

列出全部分支

git branch -v

列出全部分支，并查看每个分支的最后一次提交

git branch -vv

将所有的本地分支列出来并且包含更多的信息

git branch --merge

查看哪些分支已经合并到当前分支

git branch --no-merged

查看所有包含未合并工作的分支

git branch -a

查看全部的本地分支和远程分支

git branch -r

查看全部的远程分支

git branch -u origin/serverfix

切换跟踪的远程分支

git branch -d test

删除分支

git branch -D test

强制删除分支

merge

git merge dev

合并分支

tag

git tag

列出所有标签

git tag -l 'v1.8.\*'

查找标签

git tag v1.4

创建轻量级标签

git tag 1.4.7 e7b084 -m "alpha beta"

给 e7b084 的提交创建一个名为1.4.7标签，注释为"alpha beta"

git tag -a v1.4 -m "my version 1.4"

创建附标签

git tag -d v1.4

删除标签

附标签是存储在 Git 数据库中的一个完整对象。它们可以被校验的，其中包含打标签者的名字、电子邮件地址、日期时间，还有一个标签信息，并且可以使用 GPG Privacy Guard (PGP) 签名与验证

diff

git diff foo.txt

工作目录中当前文件和暂存区域快照之间的差异

git diff --staged

查看暂存区文件和HEAD的区别

git diff 77d231f HEAD

对比两次提交

git diff 77d231f\*

和上一次提交进行比较

git diff 77d231f 658cfd1 - book/bisecton/

限制比较的范围

log

git log -2

显示最近的两条提交记录

git log -p -2

-p用来显示每次提交的内容差异

git log --stat

每次提交的简略的统计信息

git log --pretty=short

简短显示log

git log --pretty=full

完整显示log

git log --pretty=fuller

更加完整的显示log

git log --pretty=format:"%h - %an, %ar : %s"

更改显示的格式

git log --stat

图形化显示合并和分支提交历史

git log --graph

图形化显示提交历史

git log --after=2 weeks

此日期之后的提交

git log --before="2017-10-15"

此日期之前的提交

git log --author kurisu

仅显示包含作者的提交

git log --committer kurisu

仅显示指定提交者相关的提交

git log git.png

仅仅显示文件相关的提交

git log --grep kurisu

仅显示包含关键字的提交

git log -S function\_name

列出那些添加或移除了某些字符串的提交

rebase

git rebase master

对分支变基，消除钻石链

git rebase -i HEAD~3

变基，在 HEAD~3 ..HEAD 范围内，的每一次提交都会被重写，无论是否修改信息

然而我依旧不太明白变基在去除钻石链之外的作用..

init

git init

主要用于客户端

git init --bare

主要用于服务端

config

git config --global user.name "username"

初始化

git config --global user.email email@email.cc

初始化用户名称和邮箱

git config --global alias.br branch

设置别名

git config --list

查看配置

git config <key>

检查 Git 的某一项配置

git config --global core.editor emacs

设置默认的文编辑器

add

git add \*.txt

将文件添加到暂存区

git add -A

会一个文件一个文件的按是否需要添加

git add -p

跟踪文件

commit

git commit [-message] -m "update"

提交更新到版本库

git commit -a -m "update"

跳过使用暂存区，直接将文件从工作区提交到版本库

git commit --amend

将暂存区的文件重新提交

git commit --amend

删除，你提交后和提交记录了暂存区某些需要的修改，可以add到暂存区，然后git commit --amend

push

git push origin master

推送到远程仓库

git push origin v1.4

推送标签到远程仓库

git push origin --tags

推送多个标签到远程仓库

git push origin :dev

移除远程分支（只是移除指针，可恢复）

status

git status

查看工作目录状态

git status -s | --short

获得简短的状态输出

reset

git reset HEAD <file>

将指定文件还原到HEAD的状态

git reset --hard 39ea21a

重置分支指针

git reset --merge

取消本地工作区的合并

stash

git stash list

查看存储栈中的存储

git stash (save "message")

将现在工作区和暂存区的更改保存到存储栈

git stash -u

能尚未跟踪的文件

git stash --keep-index

存储工作区的更改而忽略暂存区的修改

git stash pop

从栈顶恢复，并删除（出栈）

git stash pop stash@{2}

从栈的任意位置恢复内容，并删除（出栈）

git stash apply

从栈顶恢复存储，而不删除

git stash apply stash@{2}

从栈的任意位置恢复存储，而不删除

git stash drop

从栈顶删除记录

git stash drop stash@{2}

从任意位置删除记录

git stash branch testChange

基于存储创建一个分支

git clean -d -n

如果清除清理全部的非跟踪的文件，只清除存储区快照命令

git clean -d -f

会通知 清理工作目录会清理那些文件（-n 参数的作用）

git show v1.4

可以看到标签信息与对应的提交信息

git grep -n gntime\_r

从提交历史中寻找某段代码

git rm \*.txt

将文件从暂存区移除，并不继续跟踪（会在文件系统中删除文件）

git rm \*.txt -f (强制)

把文件从版本库中和暂存区移除！

git rm --cached README

把文件从版本库中和暂存区移除！

git update-index --assume-unchanged foo.txt

忽略当前文件的本地修改

git update-index --no-assume-unchanged foo.txt

停止忽略 单个文件

git update-index --really-refresh

停止全部忽略

git mv a.txt b.txt

将文件重命名/移动

git gc

清除不必要的文件并优化本地数据库

gitk

打开内建的图形化工具

reflog

还未整理的命令

fsck

还未整理的命令

hash-object

还未整理的命令

cat-file

还未整理的命令

Rerere

还未整理的命令

blame

还未整理的命令

cherry-pick

还未整理的命令