Git是由C语言开发

Windows安装Git

设置username和useremail

git config --global user.name "GuYiJun"

git config --global user.email "247951787@qq.com"

Git可安装在Linux 、Unix（最先只能安装在L和U两个平台上）、Windows、MacOS平台上

$ cd D:\Git 定位文件夹

$ mkdir one 新建文件夹

$ cd one 定位新建文件夹

$ pwd 显示当前目录

$ git init将该目录变成Git可以管理的仓库

新建readme.txt在one目录下

git add readme.txt 添加文件到仓库

git commit -m "wrote a readme file" 提交并提说明改动记录

git status

修改内容

git diff查看difference

git add readme.txt

git commit -m "add difference"

git status

git log日志查看

git log --pretty=oneline

21c7658cca05ca01c0482eb5a6c66deb6927b45c （版本号，16进制）(HEAD（当前版本） -> master) add Ambition

8cd45d60db2f38d6ad322e78ea59848d20bf3012 wrote a readme file

git reset --hard HEAD^回退上一个版本

git log查看提交历史第二版本已经看不到了

git reset --hard 21c7658cca05ca01c0482eb5a6c66deb6927b45c找到第二版本commit版本号，回到第二版本

cat readme.txt查看txt内容

git reflog查看命令历史，查看记录每次命令，查看第一版本的commit id

git commit -m " "将所有添加到暂存区的文件提交到分支

git管理的是修改，而不是文件

第一次修改 -> git add -> 第二次修改 -> git add -> git commit

撤销修改

git checkout -- readme.txt

将修改后而未放在暂存区的文件恢复到修改前

或将已添加到暂存区又作了修改的文件恢复到添加到暂存区

git reset HEAD readme.txt

将已add的文件恢复到工作区

git checkout -- readme.txt

修改不复存在了~

删除文件

手动删除：rm file.txt

1、彻底删除

版本库git rm file.txt

git commit -m 版本库同步删除

2.误删 checkout 恢复，只能恢复文件到最新版本

连接远程git remote add orgin git@github.com:pinkskies/note.git

$ git pull --rebase origin master（如果远程有readme.md文件，需pull最新版本）

推送到远程库git push -u origin master

远程库为空，第一次推送master分支时，加上-u参数，Git将本地master分支内容推送到远程新的master分支，还会把本地master分支和远程的master分支关联起来

以后提交后将最新修改推送到GitHub上：git push origin master

克隆

git clone git@github.com:pinkskies/note.git

ls查看目录

创建新指针

git checkout -b dev -b创建并切换

等同于git branch dev,git checkout dev

更改readme.txt并提交

git checkout master切换到master查看readme.txt并无变化

git merge dev 合并dev分支的工作成果到master分支上

$ git merge dev
Updating d46f35e..b17d20e
Fast-forward
readme.txt | 1 +
1 file changed, 1 insertion(+)

Fast-forward意为此次合并为“快进模式”，也就是直接将master指向dev的当前提交

git branch -d dev合并后删除dev分支

git branch查看当前所有分支

因为创建、合并和删除分支非常快，所以Git鼓励你使用分支完成某个任务，合并后再删掉分支，这和直接在master分支上工作效果是一样的，但过程更安全。

冲突：

新建branch feature1上修改commit，切换master又进行修改commit，merge出现冲突，将txt修改后再commit，冲突解决，git branch -d feature1将branch feature1删除

git log --graphy查看分支合并图

或者git log --graph --pretty=oneline --abbrev-commit

分支管理策略：

合并分支时，Git会使用Fast forward模式，删除分支后会丢掉分支信息

因此要强制禁用Fast forward模式，merge时会生成新的commit，从分支历史上即可看出分支信息

创建branch dev并修改commit，切换到maser，git merge --no-ff -m "merge with no-ff" dev

本次要创建新的commit，所以加上-m将commit描述写进去

git log --graph --pretty=oneline --abbrev-commit 查看分支历史

合并分支时，加上--no-ff参数就可以用普通模式合并，合并后的历史有分支，能看出来曾经做过合并，而fast forward合并就看不出来曾经做过合并。

Bug分支

git stash暂时隐藏当前工作现场

首先确定在哪个分支上修复bug，假如需要在master分支上创建，就从master创建临时分支

状态：two修改未commit，three add新加未commit

git checkout master

git checkout -b issue

git add commit提交修改

git checkout master

git merge --no-ff -m" " issue

git branch -D issue

git checkout dev

git stash list查看被保存的工作现场

恢复

工作现场还在，Git把stash内容存在某个地方了，但是需要恢复一下，有两个办法：

一是用git stash apply恢复，但是恢复后，stash内容并不删除，你需要用git stash drop来删除；

另一种方式是用git stash pop，恢复的同时把stash内容也删了

Git友情提醒，feature-vulcan分支还没有被合并，如果删除，将丢失掉修改，如果要强行删除，需要使用大写的-D参数。。

远程push：

必须先pull远程再push

git pull --rebase orgin master

git push orgin master

查看git远程库信息

git remote

或git remote -v

git push orgin dev指定推送dev分支

抓取分支

git clone git@github.com:pinkskies/note.git

git branch pink目录下只能看到本地master分支

多人合作更新冲突

必须创建远程origin的dev分支到本地分支

git checkout -b dev origin/dev(远程存在dev分支）

git add env.txt

git commit

git push origin dev

此时其他人对同样的文件做了修改

git add env.txt

git commit

git push origin dev报错此时，需要pull更新最新提交

git pull同样报错，未指定本地dev分支和远程dev分支的连接

git branch --set-upstream-to=origin/dev dev 建立本地分支和远程分支的关联

再次进行pull

git pull 合并有冲突，需要手工解决

git commit -m解决后重新提交

git push origin dev成功

git checkout master

git tag v1.0 打标签 默认

或找到commit id打上

git log --pretty=oneline --abbrev-commit

git tag v2.0 f4186(commit id)

git tag查看所有tag（按字母排序）

git show v1.0查看标签信息

git tag -a（指定版本） v1.0 -m""(说明）

git tag -d v1.0删除tag

标签只存储在本地，不会自动推送到远程

git push origin v1.0 推送tag到远程

git push origin --tags推送全部未推送到远程的本地标签

删除已推送到远程的tag：

先从本地删除

git tag -d v1.0

再从远程删除

git push origin :refs/tags/v1.0

多个远程库

git remote add github git@……/.git

.gitignore文件

文本编辑后另存为.gitignore

git add -f App.class -f强制添加到Git

<https://github.com/github/gitignore> 官方链接

git check-ignore -v App.class检查.gitignore某个规则写错了

git config --global alias.st status

用git st替代git status

同理git config --global alias.co checkout --global全局参数

git config --global alias.unstage 'reset HEAD'配置unstage别名

git unstage test.py

等同于git reset HEAD test.py

git config --global alias.last 'log -1'配置git last显示最后一次提交信息

git last 显示最近一次提交

git config --global **alias**.lg "log --color --graph --pretty=format:'%Cred%h%Creset -%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold blue)<%an>%Creset' --abbrev-commit"

同理

配置文件

每个仓库Git配置文件都放在.git/config文件中

cat .git/config

别名就在alias后

删除别名直接将对应行删除即可

当前用户Git配置文件在.gitconfig中

cat .gitconfig