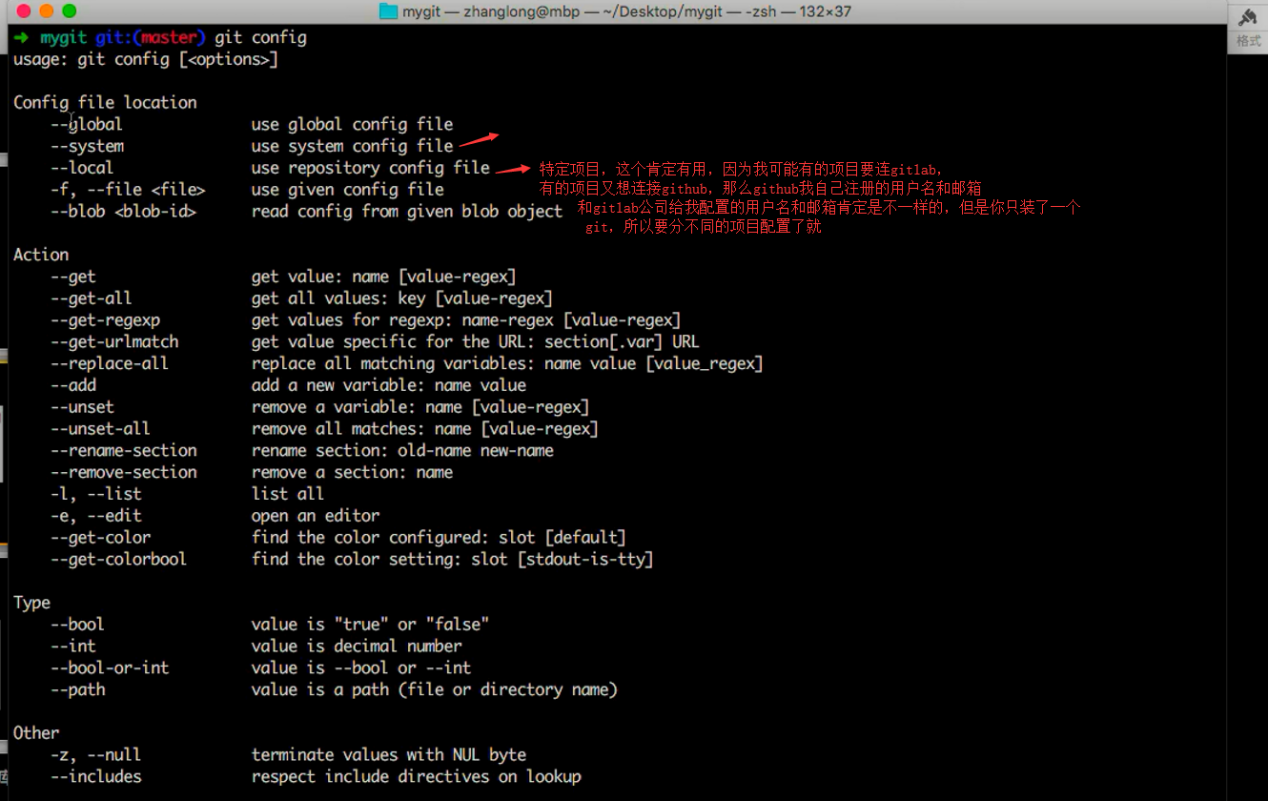


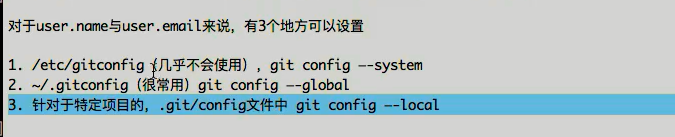
上面这个用到了ohmyzsh软件,linux上面有,有智能提示

git的邮箱全局配置(global)和分项目配置(local)



具体使用





windows的.gitconfig全局文件应该在C:\Users\Administrator目录下

# git添加删除修改与日志

ctrl+a 光标到最开始

ctrl+e 光标到最后

错误commit的了一个文件后进行删除：git rm test2.txt

如果git rm test2.txt之后但是还没执行git commit之前后悔了，要通过两个命令找回

1. git reset HEAD test2.txt
2. git chechout -- test2.txt

其实通过git rm删除一个文件后(但是还没有执行git commit,如果git commit应该恢复不删除的了就),要执行两个步骤才能恢复文件，还有一种方式删除文件通过linux rm -f test2.txt

这样通过操作系统删除的文件想要恢复只需要通过 git checkout -- test2.txt直接就恢复了,不用执行git reset HEAD test2.txt这个步骤了

重命名:

git mv a.txt b.txt

修正commit的消息

1)错误的消息

git commit -m “错误的消息”

2)正确的消息

git commit --amend -m “正确的消息”

1. git log 发现错误提交的消息不见了

消息日志：

git log -3 最近三条

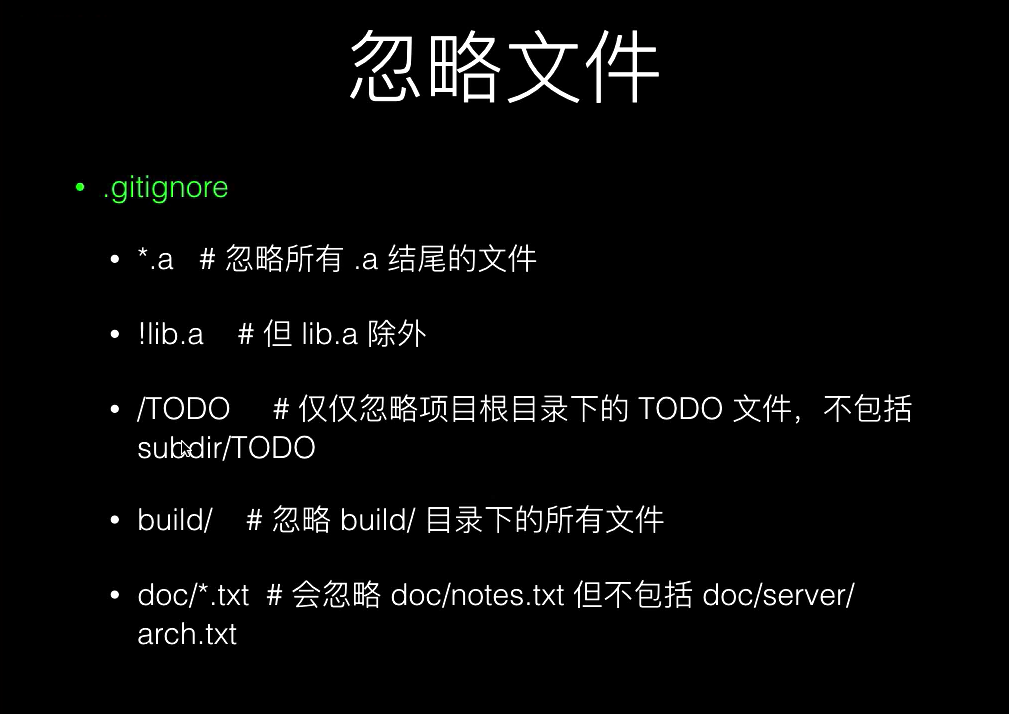
git log --pretty=oneline 简单直观的显示消息

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

.gitignore文件:(注意只要git add完的文件 gitignore就失效了)

\*.b 以b结尾的文件

!a.b 但是a.b除外



注意空目录的话git自动忽略

单层目录下的文件忽略

/\*/a.txt

多层目录下的只要有a.txt就忽略

/\*\*/a.txt

git分支：

创建分支：git branch new\_branch(比如当前分支在develop上，则这个命令会初始创建一个和develop分支一模一样的内容)

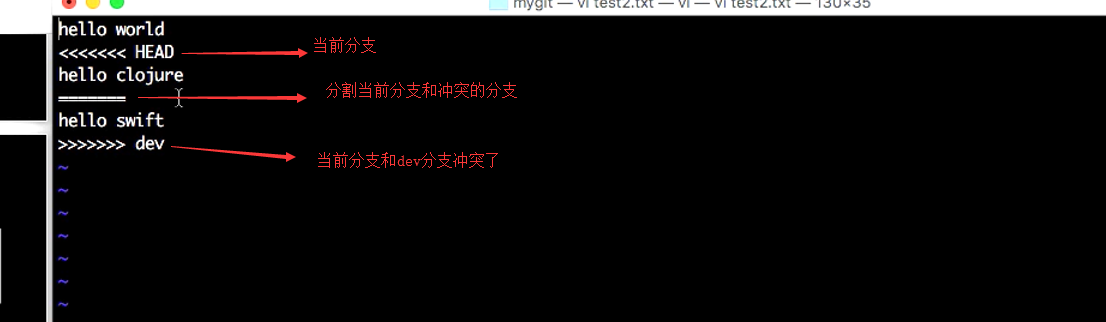
切换分支：git checkout new\_branch

删除分支：git branch -d new\_branch

合并分支：

git merge new\_branch（比如当前在master分支上，执行这个命令是把new\_branch的内容合并到当前的master分支上）

冲突分支：



解决完冲突后先执行git add 文件名 然后git commit 文件名就解决完冲突了



容易混淆的两个命令:

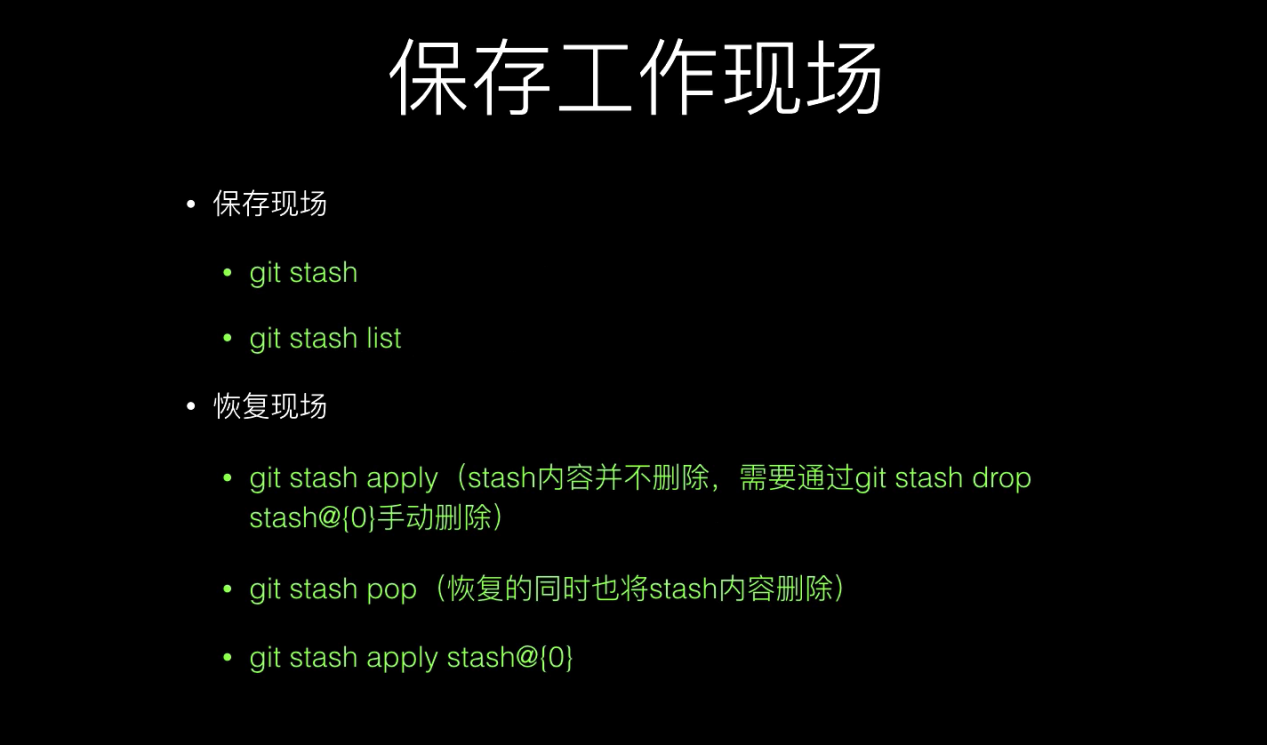
git checkout -- a.txt ：没有git add之前命令生效，恢复到暂存区的最后一次内容，丢弃修改

git reset HEAD a.txt ：代码不会有任何变更，git add之后有效，相当于add到暂存区的内容移除到工作区。

git stash:当我在a分支开发，这时候需要紧急改另b分支内容修复bug，但是我a分支还不能提交代码，因为可能有bug，或者不能编译都，那么就要把a分支临时stash

git stash pop:等我又切换成a分支的时候，执行这个命令，就可以接着在a分支上开发了

git stash save ‘hello world’:stash的时候注释一下可以（git stash命令的增强版，要不自己都忘了a分支开发到哪里了）



git标签

轻量级标签 ：git tag v1.0

带注释的标签 ：git tag -a v2.0 -m ‘2.0released’

所有标签 ：git tag

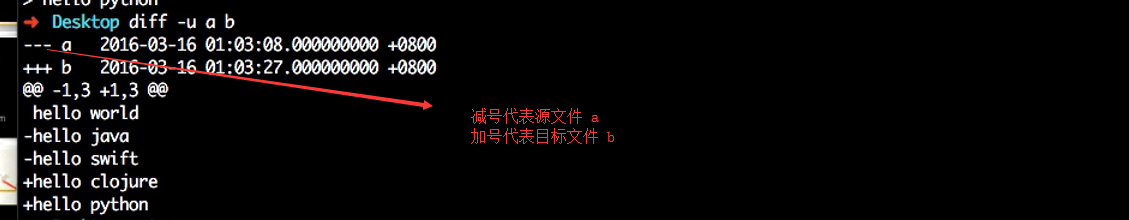
查找以v开头的tag ：git tag -l ‘v\*’

删除 : git tag -d v1.0

diff:

查看文件提交的历史及人（非常好用）：git blame a.txt

先看看linux的diff命令



git diff a (比较暂存区和工作区的文件差异 减号代表暂存区 加号代表工作区)

git diff HEAD(比较工作区与已经最新的版本库的差别 减号代表版本库 加号代表工作区)

git diff --cached [commit\_id](比较暂存区和版本库的差别 减号代表暂存区 加号代表版本库)

git 远程

push 推送

pull = fetch + merge

git获取ssh的远程仓库

1. 远程建立个仓库（github官网的图形界面）
2. 本地建立项目并初始化git init .
3. git remote add origin [git@github.com:gitlecture/gitlecture.git](mailto:git@github.com:gitlecture/gitlecture.git)
4. git remote show origin

会弹出一个确认界面输入yes

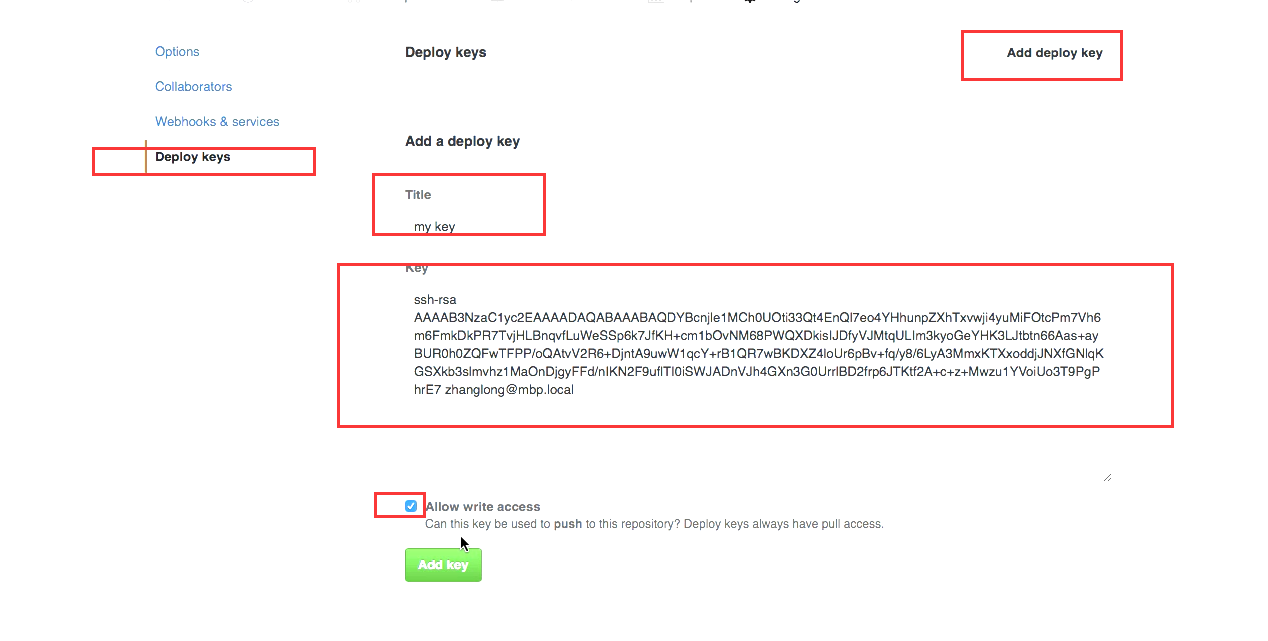
但是不会成功，因为没有把本地的公钥放到远程的github仓库中

1. cd .ssh

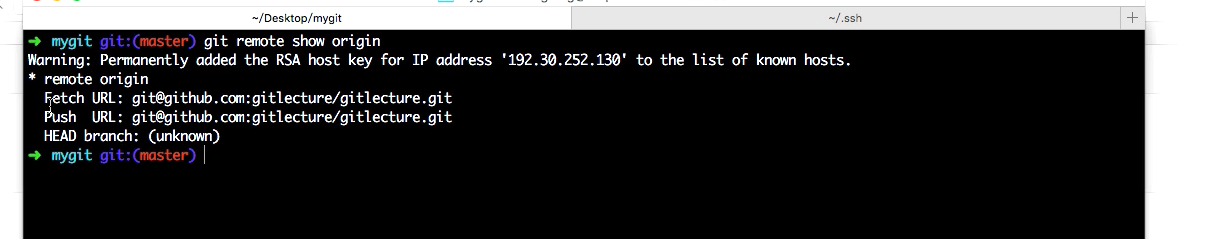
执行ssh-keygen 一路回车

会生成id\_rsa(私钥，自己留着谁也不给) id\_rsa.pub（公钥，放到github上面，具体怎么放，看下面的截图）

进入仓库->settings->Deploy keys->Add deploy key->（title随便写，key是本地的公钥）



1. 设置完了再次执行 git remote show origin



1. 本地和远程关联：git push -u origin master
2. **注意github在仓库级别和账户级别都可以建立公钥，但是两个仓库级别不能用同一个公钥（未测试），所以我觉得企业开发一般都在账号级别建立公钥，gitlab类似。**

git命令中经常出现的origin单词是什么意思呢？

**origin代表远程**的意思，但是实际是本地本地分支。

比如：

本地master->本地orgin master->远程master分支

可以看到orgin master是本地分支和远程分支的桥梁

如果能查看到本地origin分支呢？git branch -a

如果屏蔽查看本地origin分支呢？git branch

如何查看所有版本并指定各个版本的最后一次提交呢? git branch -av

注意可不要切换到远程分支！比如git checkout origin/master!(错误)

如何git clone下载的项目改成自己喜欢的名字呢？命令最后指定一个名字就行了

git clone [git@github.com:gitlecure/gitlecture.git](mailto:git@github.com:gitlecure/gitlecture.git) **mygit2**

# git图形界面

which gitk

在git的某个仓库目录下敲入gitk就启动图形界面了

感觉gitk太弱了，老师也没太讲解

git gui

gui也没咋讲

git设置别名

git config --global alias.br branch（branch简写为br）

git config --global alias.unstage ‘reset HEAD ’

拉取远程的分支并且下载切换：（这里要动手实验一下）

git pull

git checkout --track origin/develop

删除本地分支：

git branch -d develp

删除远程分支:

git push origin :develop（将本地的一个空分支推送到远程，相当于把远程删除掉了）

git push origin --delete develp（新版本用这种方式了）

创建远程分支：

git push --set-upstream origin develop

# tag

git tag v1.0

git tag -a v2.0 -m ’my tag’

git show v1.0

将当前分支推送到远程：git push origin v1.0

一次性的把本地创建的所有标签推送到远程：git push origin --tags

其他人获取标签：git pull

删除远程tag标签：git push origin :refstags/v1.0(新的写法：git push origin --delete tag v1.0)

删除本地tag：git tag -d v1.0

# git submodule

**submodule已经不推荐使用**

1. 先建立两个git项目git parent和git child
2. 在git parent仓库下git submodule add [git@github.com:gitlecture/git\_child](mailto:git@github.com:gitlecture/git_child) mymodule（注意mymodule目录不能存在）
3. git add .
4. git commit -m “add submodule”
5. git push
6. git child有更新了怎么办？只需要进入submodule目录后执行git pull就能把最新代码拉回到本地
7. 如果有很多子模块进入每个module的目录再执行太费事了，

只需要进入parent目录后执行git submodule foreach git pull，把所有子模块下载

8）更新后还要git add . git commit -m”子模块有更新”

9）其他人想下载整个项目呢？

git clone [git@github.com:gitlecture/git\_parent](mailto:git@github.com:gitlecture/git_parent).git git\_parent3 --recursive

# git subtree

1）在parent项目仓库下执行添加子项目

git remote add subtree-origin [git@github.com:gitlecture/git\_subtree\_child.git](mailto:git@github.com:gitlecture/git_subtree_child.git)

2）subtreedir:生成目录 subtree-orgin：第一步的名称 master:要下载的哪个分支

git subtree add --P subtreedir subtree-origin master --squash

1. child如果更新了，master怎么更新到代码呢？

git subtree pull --P subtreedir subtree-origin master --squash

1. 在master修改child分支的内容，然后推送

git add .

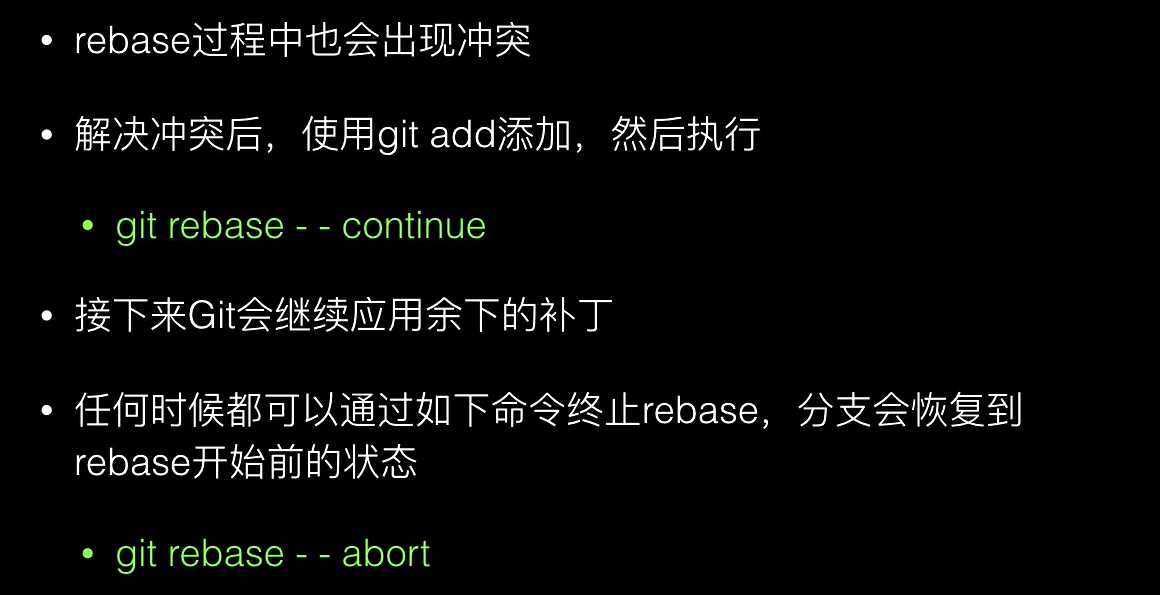
git commit -m “现在在master分支根目录下”

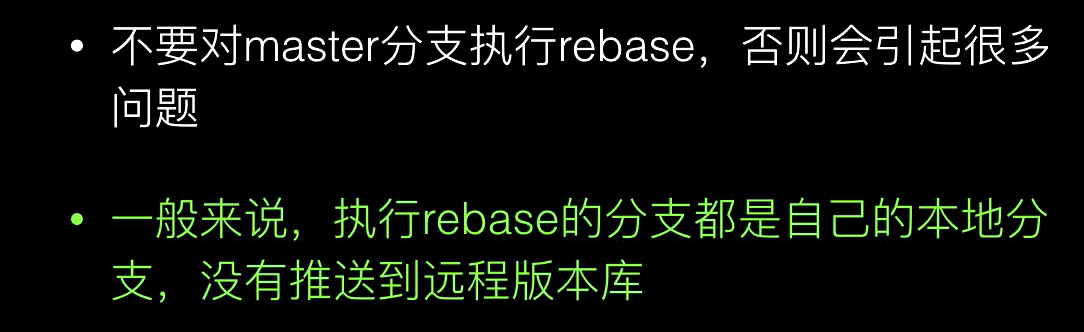
git push（推送到master的远程）

git subtree push -P subtreedir subtree-origin master（推送到child的远程）

# rebase

rebase类似merge





1. 不要在master分支上执行rebase
2. 不要在与其他人共享的分支上执行rebase