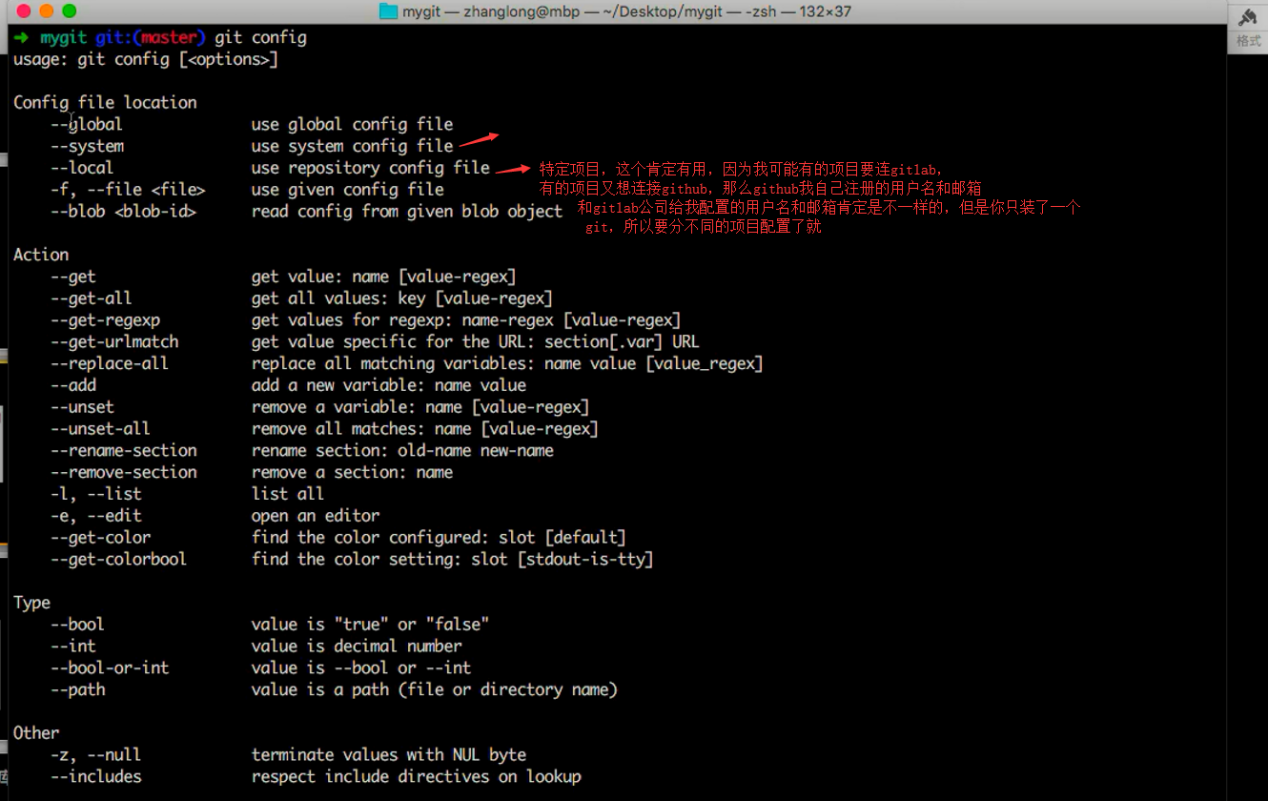


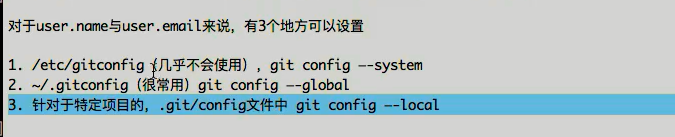
上面这个用到了ohmyzsh软件,linux上面有,有智能提示

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

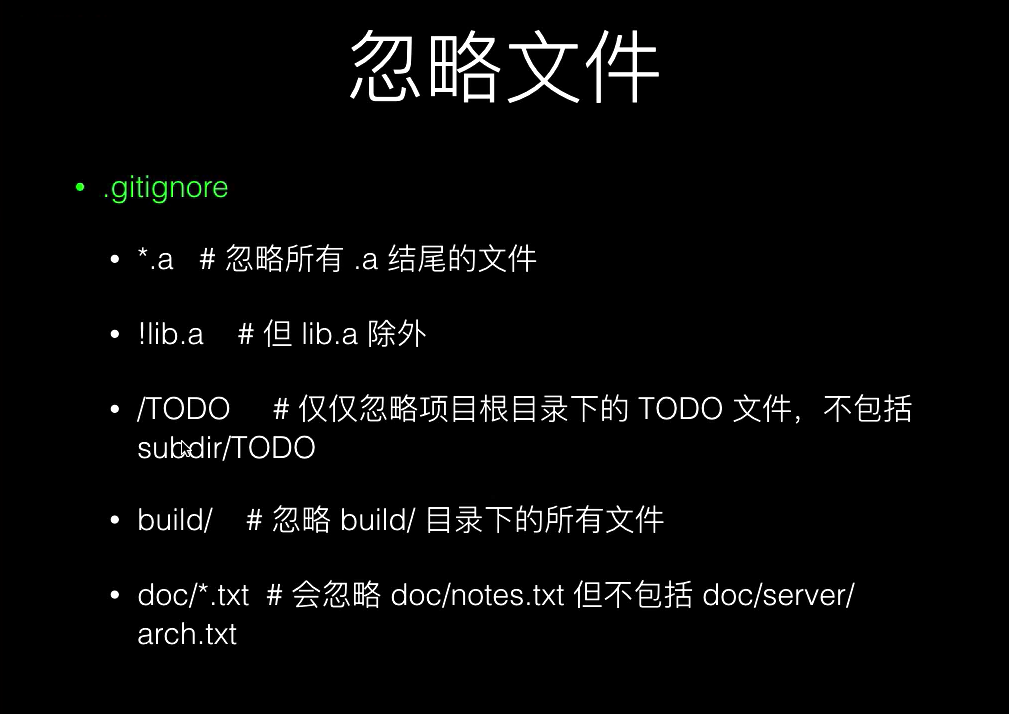
git log --pretty=oneline 简单直观的显示消息

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)

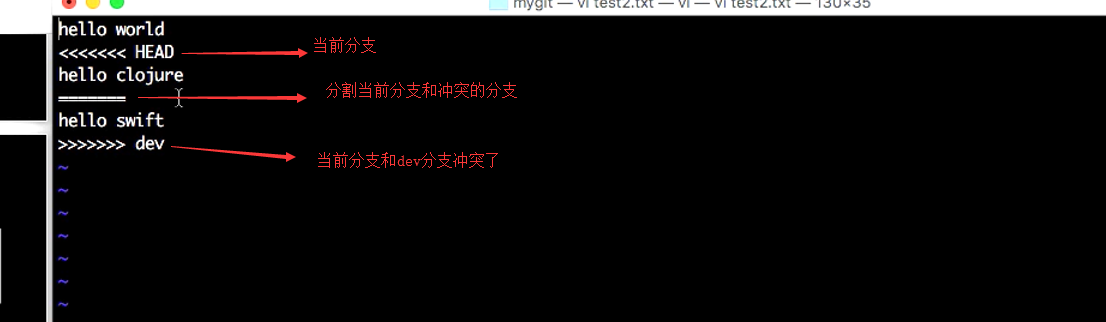
切换分支：git checkout new\_branch

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:

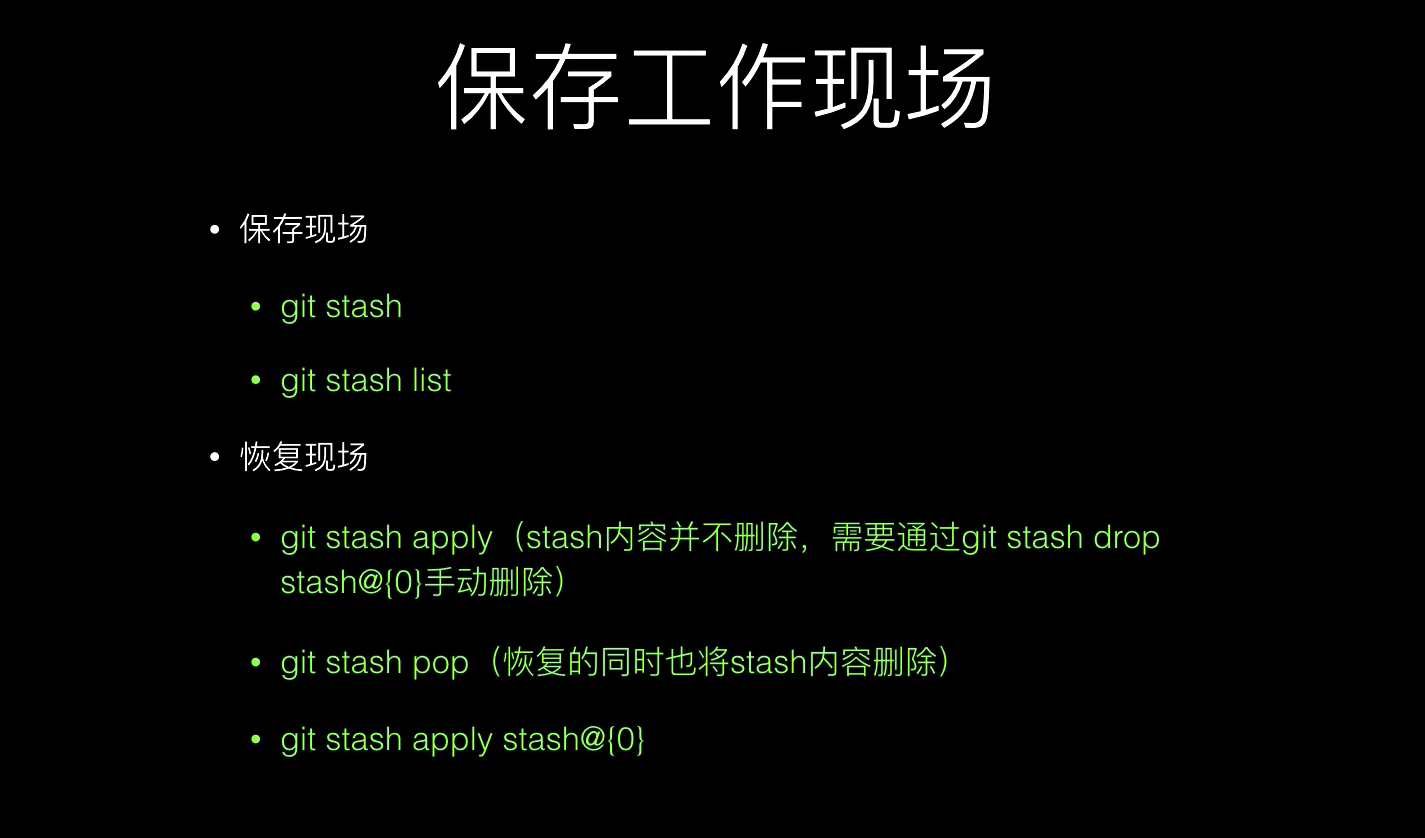
git checkout -- a.txt ：没有git add之前命令生效，恢复到暂存区的最后一次内容，丢弃修改

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

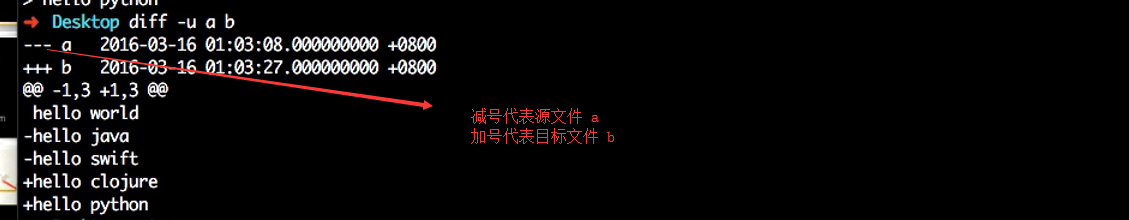
查找以v开头的tag ：git tag -l ‘v\*’

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