# git命令行学习

1. git add .

这个命令是把文件提交到缓存仓库中

1. git commit –m “描述”

这个命令是提交版本到本地仓库的

1. git status

查看当前状态的

1. git log

查看当前版本记录的 可以加上 git log –pretty=oneline 可以简单看到版本号和版本描述

1. git reflog

如果有回退版本的，这里也可以看到回退时的版本号

1. git diff -- filename

##注意--后面有空格这个是工作区和暂存区文件的比较。git diff HEAD -- filename这个可以工作区和版本库比较(--后面有空格)

1. git reset --hard HEAD^

回归到上一个版本号,如果 是前两个版本 可以使用git reset --hard HEAD~2 或者查看日志，找到对应版本号 git reset –-hard xxxxxx ,xxxxxx表示版本号的前六位数字。###注意使用这个命令的时候，工作区和暂存区都会本回退版本覆盖。

git reset HEAD -- filename 是把暂存区的文件撤销，把刚才add进来的文件返回到工作区（在没有提交前，无论你add多少次进暂存区，只要reset HEAD 都会清空暂存区，实际就是把版本库覆盖到暂存区，把暂存区的东西覆盖到工作区）

1. git checkout -- filename

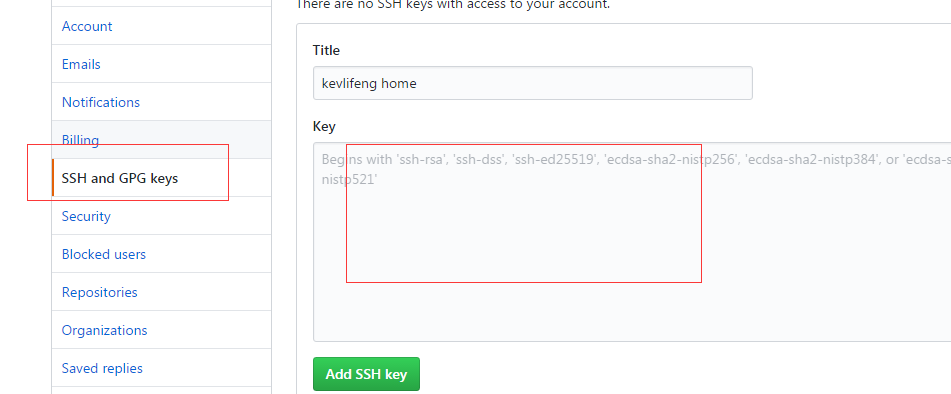
##注意—后面有空格，这个命令是把暂存区的数据复制到工作区。

进阶

下面是开始远程仓库

1. ssh-keygen –t rsa –C “[kevinlifeng1986@gmail.com](mailto:kevinlifeng1986@gmail.com)”

可以在用户目录下找到ssh文件目录，里面会有id\_rsa和id\_rsa.pub两个文件，这两个文件就是SSH Key的秘钥对，id\_rsa是私钥,然后把公钥的信息填写到github上的ssh上 settings->SSH and GPG keys



1. 在远程服务器上创建一个仓库。learngit.git
2. git remote add origin git@github.com: lifeng-github/learngit.git
3. git push –u origin master
4. 下次提交就是直接git push origin master

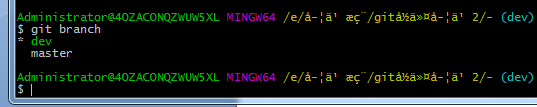
分支管理

1. git checkout –b dev

创建一个分支并切换到dev分支上。相当于两条命令git branch dev & git checkout dev .

1. git branch

查看当前有多少分支，正在时候的分支会加\*号



1. git checkout branchname

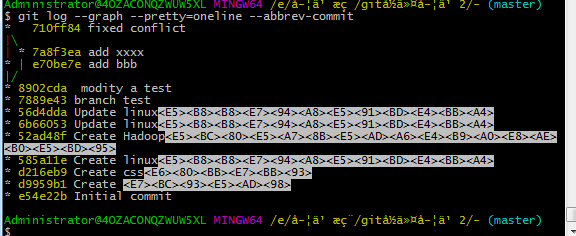
切换分支

1. git merge branchname

在当前的分支上合并branchname的分支。当分支上有上传，master分支也有上传，在合并的时候可能会有冲突，那么要进到文件里面去修改对应的东西，然后在提交。合并可以加上 --no-ff 参数就可以普通模式合并，合并后的历史分支，合并分支后有历史分支，能看得出曾今的合并

1. git log –-graph –-pretty=oneline –-abbrev-commit

这个可以看到对应的分支图



1. git branch –d branchname

删除分支,如果分支从来没有合并过主线就想直接删除，这个时候是删除不了，的除非你用 git branch –D branchname

1. git stash

保存当前分支下没有提交的代码，然后切换到其他的分支做其他事情，比如切到bug分支去修复bug，或者其他分支有紧急的事情处理。和下面的git stash list,git stash pop ,git stash apply.

1. git stash list

查看stash里面的内容。然后可以使用

1. git stash apply

这个是把stash最新的内容复制到工作区，但是stash里面的内容没有删除，如果要删除就要git stash drop;

1. git stash pop

这个是把stash最新的内容迁移到到工作区中，stash就没有保存的再上一次的内容了。

1. git remote

查看远程库的信息

1. git checkout –b dev origin/dev

这个是把远程的库的dev分支拉下来作为本地的dev分支，当你拉去下来时候，你修改好好代码提交到远程dev分支： git push origin dev是把本地的提交到远 程的dev分支，

###在提交分支的时候，可能人家也有提交了，你是提交不了，那么你就要使用 git pull拉去最新的提交。然而pull也失败。那是因为本地的dev分支和远程的分支origin/dev 没有链接。你要使用git branch –-set-upstream dev origin/dev, 然后在 git pull 这个时候就可以拉去最新的代码到本地库中，然后手动修改对应的conflict，然后提交

1. git pull

拉去最新的分支到本地库来，当提示“no tracking information”,表示你本地分支和远程分支没有建立链接。

标签管理

1. git tag v1.0

打标签，其实是在最新提交的commit上打上标签，如果忘记打标签，那么可以使用查看日志，想对那个提交时候打标签。git tag –a v0.1 –m “描述” xxxxxx 这个是-a表示制定标签名，-m 后面是这个标签的描述,xxxxx表示commit提交的版本号

1. git tag v0.9 xxxxx

xxxxx表示版本号，可以commit这个版本的时候打一个标签。

1. git tag

查看标签

1. git show v0.9

显示当前标签的信息

1. git tag –d v0.1

如果打错标签也可以删除

1. git push origin v0.1

把本地标签推到远程库中

1. git push origin –tags

一次性把标签全部推到远程仓库中

1. git tag –d v0.1

这个是删除本地的标签，如果要删除远程仓库的标签要使用

git push origin :refs/tags/v0.1

自定义git

1. git config –-global color.ui true

显示颜色，比较醒目

1. git config –global alias.co checkout
2. git config –-global alias.ci commit
3. git config –-global alias.br branch
4. git config –-global alias.unstage ‘reset HEAD’