# Git

Linus在1991年创建了开源的Linux

代码管理?CVS、SVN?集中式?pass

商业的版本控制系统BitKeeper,Git

$ git config --global user.name "Your Name"

$ git config --global user.email "email@example.com"

git init

git add readme.txt 加入暂存区

git commit –m “操作说明” 提交到分支

git status

git diff查看工作区和暂存区区别

git　log　--pretty=oneline

git　log　–graph 查看分支合并图，按q退出

git　reset　--hard head~ **版本回退,回退到上一个版本,执行一次就往回回滚**

**提交版本(commit 后才算一个版本)**

**1.1,1.2,1.3**

**现在处于1.3版本,如果使用**git　reset　–hard　head~ 文件就会恢复到1.2版本

git　reset　--hard 90d92a277

git reflog查看命令历史，以便确定要回到未来的哪个版本

工作区

版本库(.git)

stage:暂存区

master分支

head指针



git　diff head　--　readme.txt查看工作区和版本库里面最新版本的区别

git　checkout　--　readme.txt 只能撤消还没有提交到暂存区修改

git　reset　head　readme.txt把暂存区的修改撤销掉

如果add后,想要从暂存区返回,可以使用这个命令

git　rm　A.txt 删除文件

# GitHub

SSH KEY

连接超时，首先找到git的安装目录，找到/etc/ssh/ssh\_config文件

把如下内容复制到ssh\_config文件的末尾处：并记得保存

Host github.com

User git

Hostname ssh.github.com

PreferredAuthentications publickey

IdentityFile ~/.ssh/id\_rsa

Port 443

git remote add origin https://github.com/halfmoon2014/learngit.git

git push -u origin master Git不但会把本地的master分支内容推送的远程新的master分支，还会把本地的master分支和远程的master分支关联起来，在以后的推送或者拉取时就可以简化命令

git clone <https://github.com/halfmoon2014/moreless.git>

git checkout -b dev

{

git branch dev 创建分支

git checkout dev 切换分支

}

git branch查看当前分支

//在切换分支前,当前分支最好commit,不然状态会被带到目标分支去了!

git checkout master

git merge dev 合并分支

git branch -d dev 删除分支

git pull 更新本地分支

git push origin dev:dev //将本地dev分支推送到远程dev分支

git rm --cached logs/xx.log，然后更新 .gitignore 忽略掉目标文件，最后 git commit -m "We really don't want Git to track this anymore!"

#### 添加同名多远程仓库

添加一个remote,这里是all,也可以是别的名字(如origin)

git remote add all https://github.com/wonux.test.git

再添加另一个:

git remote set-url --add all https://git.oschina.net/wonux/test.git

重复向同一个远程仓库名字添加需要set-url --add参数