

git --version 查看git版本

git config --global user.name //获取当前登录的用户

git config --global user.email //获取当前登录的用户邮箱

git config -l 以列表的形式查看配置项

配置用户

git config --global user.name 'username' //username是你的git账号

git config --global user.email 'email' //email是你的git邮箱

初始化git仓库

git init //初始化一个仓库，此时文件里会得到一个.git的隐藏文件

显示目录的状态 有没有添加或者修改文件

git status

添加到暂存区中

git add index.html

git add . //全部添加到缓存区

增加到本地版本库中

git commit -m '备注信息'

查看版本

Git log --oneline

比较差异

git diff 比较的是暂存区和工作区的差异

添加远程仓库

git remote add origin xxxx 连接远程仓库

Ssh相关

ls -al ~/.ssh 检查ssh keys是否存在

ssh-add ~/.ssh/id\_res 生成新的key

ssh-keygen -t rsa -C 'xxx' 添加一个ssh

ssh -T [git@github.com](https://github.com) 检测是否建立连接成功

git init

git add README.md

git commit -m "first commit"

git remote add origin [git@github.com:wanglogn/hahanihao.git](https://github.com/wanglogn/hahanihao.git)

git push -u origin master

`git clone '远程仓库的地址'`

新建分支

`git branch xxx`

查看分支

`git branch`

查看所有分支

`git branch -a`

在本地建立和远程一样的分支

`git branch -t 分支名`

切换分支

`git checkout 分支名`

合并分支

`git merge 分支名`

更新主分支

`git pull origin master`

首先由管理员创建一个远程仓库

所有的组员pull或者clone远程仓库

所有的组员不能操作master分支，只能在自己新建的分支下进行开发

开发完成后将你的分支push到远程仓库

管理员不定时的pull远程仓库的代码，检查每个分支下的开发情况

如果开发没有问题，没有bug，管理员将该分支下的代码和主分支合并