**一14一 一14一 一12一 一11一** 一10.5一 一10.5一 一10一

# 基础配置

配置userName和userEmail

git config --global user.name "jinxiaoshu"

git config --global user.email "[1024072563@qq.com](mailto:'1024072563@qq.com')"

config的三个作用域

git config --local(某个仓库) global（当前用户的所有仓库）system（系统所有登录的用户）

显示配置

git config --list --local global system

查看某个命令所有的参数（不输入--和之后的单词）

git config

直接在.git/confige文件中修改

# 初始化项目

新建项目初始化git目录

git init git\_learning

添加文件到暂存区

git add readme.txt

添加已经跟踪了的文件的改动到暂存区

git add -u

查看版本库的信息

git status

提交改动到本地版本库

git commit -m"your Reason"

查看提交日志信息

git log (-n1查看最近一次提交)

# 修改文件

从缓存区或直接删除某个文件

git rm fff.txt

命令直接重命名文件

git mv index1 index2

命令直接删除文件

git rm index1

查看分支信息

git log --online --all --n4 --graph 单行、所有分支、最近4次、图形化

图形化界面

gitk (--all)

# 查看仓库的信息

查看commit或者tree、blob的类型

git cat-file -t 1f6ee953d

查看commit或者tree、blob的信息

git cat-file -p 1f6ee953d

# Branch和Head

查看所有分支

git branch -av

基于master创建fixreadme分支，并切换到fixreadme分支

git checkout -b fixreadme master

切换到master分支

git checkout master

删除test分支

git branch -d test (如果test在分叉上，需要用-D)

修改最近一次commit的信息

git commit --amend（修改好之后Esc，不在编辑模式后输入：wq）

修改的commit信息(rebase的简单使用)

git rebase -i 2efec075b8（父commit）

差异的比较

head和缓存区差异的比较 git diff --cached

工作区和缓存区差异的比较 git diff

比较最近一次提交和上上次提交 git diff HEAD HEAD~2

比较两个commit之间的差异 git diff 01fb1 2eb9a

**比较两个分支** git diff master temp

针对性的比较某些文件 在后边加上 -- xxx xxx

恢复工作区和暂存区

暂存区恢复成和Head git reset HEAD -- xxx xxx

工作区恢复成和暂存区 git checkout -- xxx xxx

消除最近几次提交

git reset --hard 7cf519f6（你想退到的某一版本）如果没有push到远程仓库，该操作会不可逆的删除最近提交，并且缓存区工作区的内容会消失

stash

stash当前修改 git stash

查看现有stash git stash list

移除stash git stash drop stash@{0}

重新应用缓存的stash git stash pop（pop会调用drop或者apply不会调用drop）

# 远程仓库

克隆本地仓库到一个地方

git clone file://D:\gitdemo\git\_learning/.git zhineng（自动设置本地仓库为克隆仓库的origin）

git clone --bare file://D:\gitdemo\git\_learning/.git zhineng（裸仓库，没有git信息，不能提交）

查看、添加、删除远程仓库

查看 git remote -v

添加 git remote add zhineng [file://D:\gitdemo\gitRemote\zhineng](file://D:\\gitdemo\\gitRemote\\zhineng)

删除 git remote rm zhineng

推送本地仓库的改动到远程

git push zhinengRemote master

git push --set-upstream zhinengRemote master（第一次需要使用--set-upstream来关联）

git push zhinengRemote（想把本分支的推到远程，可以不用写本分支的名称）

关联本地和远程仓库

git branch --set-upstream zhinengRemote master

拉取远程仓库的变化（不写branch，默认是当前分支）

下载远程仓库的所有变动，还没和本地进行关联 git fetch [remote]

拉取远程仓库的变化，并与本地分支合并 git pull [remote] [branch]

# windows CMD 命令

单操作

|  |  |
| --- | --- |
| 清空屏幕 | cls |
| 切换盘符 | D: |
| 保存并退出 | :wq |
| 退出 | :q |
| 撤销 | :u |
| 显示当前目录 | cd |

文件和文件夹操作

|  |  |
| --- | --- |
| 回到上一层目录 | cd .. |
| 查看当前文件夹下所有的文件信息 | dir |
| 新建当前目录下的test文件夹 | md test |
| 新建非当前目录下的my文件夹 | md c:\test\my |
| 新建a.txt文件 | cd.>a.txt |
| 把字母a和回车换行覆盖输出到a.txt文件 | echo a>a.txt |
| 把字母b和回车换行追加输出到a.txt文件 | echo b>>a.txt |
| 删除a.txt的文件 | del a.txt |
| 删除所有后缀为.txt的文件 | del \*.txt |
| 删除名为test的空文件夹 | rd test |
|  |  |

切换到指定目录（如果不同盘，需先切换盘符）

cd D:\gitdemo\git\_learning

