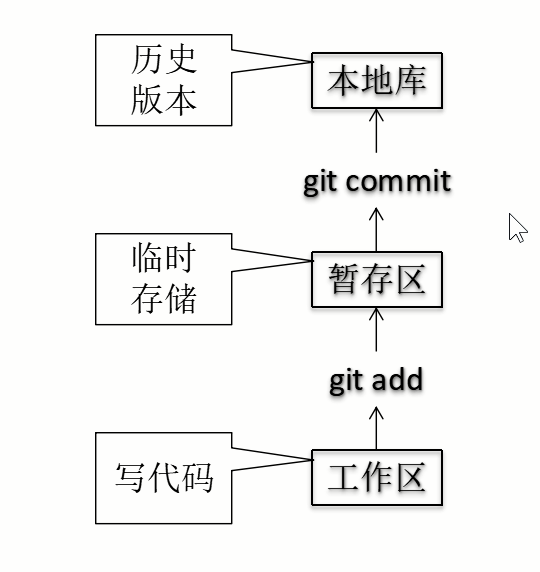
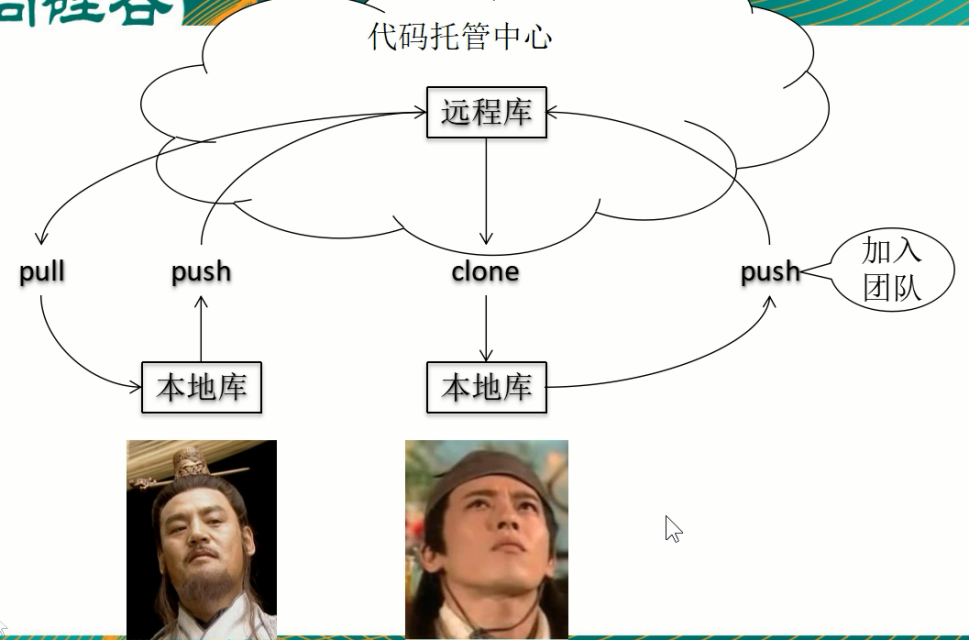
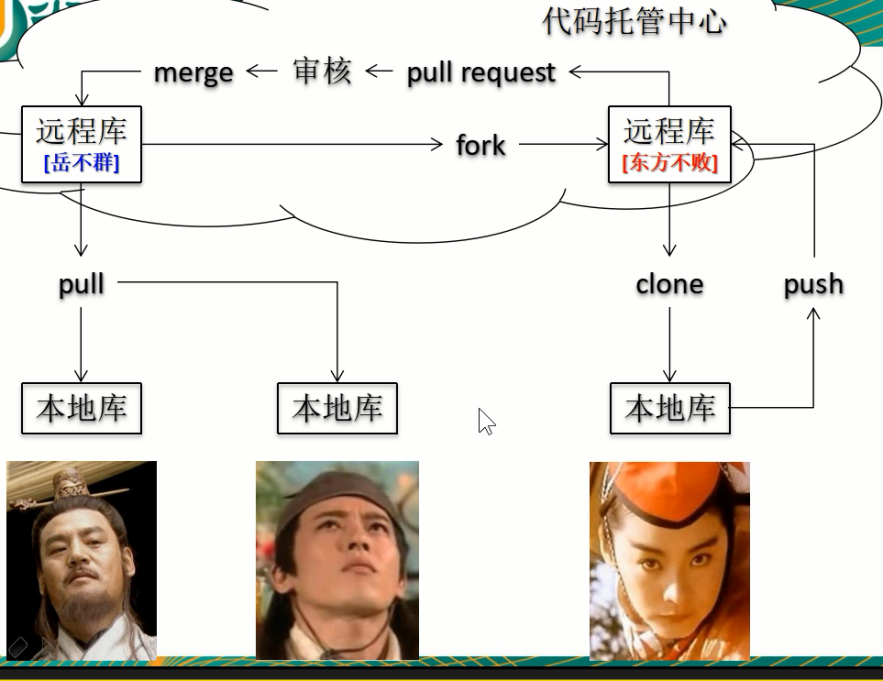
## Git



## 团队内协作



## 跨团队协作



## 本地Git

1. mkdir test //创建文件夹
2. git init //初始化
3. git config user.name tom //配置项目级别签名 项目优先级高于系 统 保存于 .git/config中 cat或vim查看
4. git config user.email [good@163.com](mailto:good@163.com)
5. git config --global user.name tom //配置系统级别签名 保存于~/.gitconfig中
6. git config --global user.email [good@163.com](mailto:good@163.com)

使用：

1. git status
2. git add xxx.txt //添加到缓存区
3. git commit xxx.txt //提交到本地仓库 git commit -m “message” xxx.txt
4. git restore xxx.txt //撤销工作区的文件修改，恢复到上一次commit文件内容
5. git log //查看历史记录（提交过的）

git log --pretty=oneline,short

git log --oneline //只能显示当前版本的前面几个版本

git reflog

1. git diff xxx.txt //比较工作区文件与本地仓库最近版本的不同
2. git reset --hard HEAD //恢复到上一个版本

恢复指定版本：

git reflog

git reset --hard 70740d7

git reset --hard HEAD^^^

git reset --hard HEAD~3

（复制：ctrl+insert 粘贴：shift+insert）

8 1.  git diff  filepath 工作区与暂存区比较

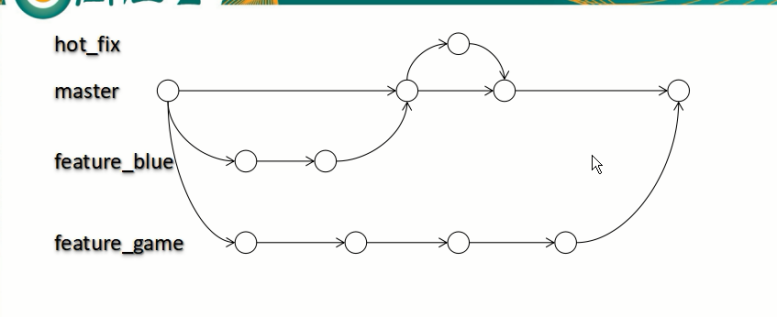
2. git diff HEAD filepath 工作区与HEAD ( 当前工作分支) 比较

3. git diff --staged 或 --cached  filepath 暂存区与HEAD比较

4. git diff branchName filepath  当前分支的文件与branchName 分支的文件进行比较

5. git diff commitId filepath 与某一次提交进行比较

##### 分支



1. git branch hot\_fix //创建分支
2. git branch -v //查看分支
3. git checkout hot\_fix //切换分支
4. git merge master //合并分支 （）（在一个工作区条件下，谁提交文件归谁，另一个人会失去改文件）（合并有冲突时，默认合并到工作区，修改后提交）
5. 合并之前要把工作区的修改都提交了
6. 合并之后，有冲突的文件要在单独提交（git add, git commit -m “” d.txt）,最后git commit -m “”

ll 查看文件细节

ls -A 查看隐藏文件

ll .git/ 查看.git

Vim :set nu //显示行号 command模式

Pwd //显示当前路径

## 远程仓库

1. git remote -v //查看远程仓库别名
2. git remote origin <https://github.com/CoderQiao/MyFirst.git> //给远程仓库起别名origin
3. git push origin master //本地库push到远程库
4. git clone <https://github.com/CoderQiao/MyFirst.git> //从远程库clone到本地
5. git fetch origin hot\_fix //fetch抓取，只下载不改 变本地工作区
6. git checkout origin/master //查看抓取来的远程库
7. git checkout master //切换回来
8. git merge origin/master //合并远程master与本地 master （不明白可不可以合并本地其他分支）
9. git pull origin master //抓取并合并本地库 pull=fetch+merge
10. 可以多人同时pull，但只有一个能提交，一人提交后其他人不能再提交。只能重新pull后拿到最新版才有提交权限。即待提交版本的前一个版本要与远程库一致，不能有代差。
11. 重新pull后，会有冲突，要一个一个解决冲突，再add commit。commit时，后没有文件名，因为提交整个库。之后才能push。
12. 冲突来源于fetch库与本地库不同，类似分支合并时的冲突
13. git remote add origin\_ssh [git@github.com:CoderQiao/MyFirst.git](mailto:git@github.com:CoderQiao/MyFirst.git) //ssh操作 给远程地址起一个别名叫origin\_ssh

### 关联MyEclipse

###### push

1. windows->preference->Team->Git->configuration //git 配置
2. init 右键项目->Team->share project->Git //初始化项目
3. add to index //项目添加到缓存区
4. 右键项目->Team->commit //项目提交到本地库
5. 右键项目->Team->Remote->push force update 可能会有文件丢失 项目为空时，不点force update 也可以顺利push

###### pull

1. file-> import-> Git-> projects from Git
2. import general project
3. 右键项目-> configure-> convert to maven project
4. 右键项目-> Team-> pull branch hot\_fix/master
5. 解决冲突：1.Team-> merge tool 2.team -> commit //先解决冲突再重新提交本地库，推送远程库

###### Git工作流

1. Team - > switch to - > new branch //本地创建新的分支
2. 修改文件 //
3. 在Git staging 中提交到本地库 //提交到本地库
4. Team - > push branch “workflow” //提交到远程库