git init 创建版本库

git add <file> 添加文件，可反复多次使用，添加多个文件

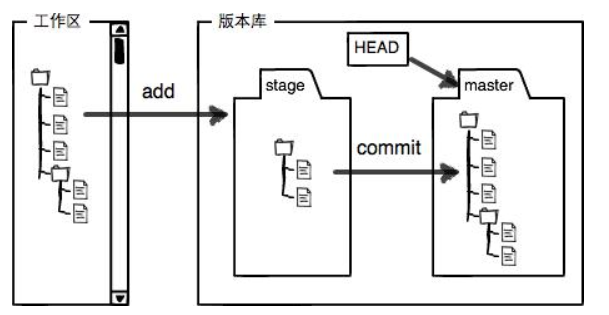
git commit -m <message> 完成

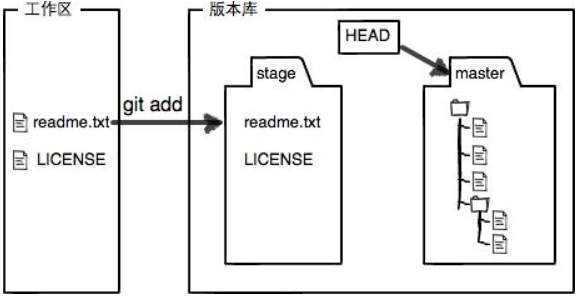
git reset --hard commit\_id 版本的切换

git log 查看提交历史

git reflog 查看命令历史

git status 查看工作区和暂存区的状态





git checkout -- file 丢弃工作区的修改或者是删除，都可以一键还原

git reset HEAD <file> + git checkout -- file 丢弃暂存区的修改

git rm <file> 删除一个文件

**添加远程库**

git remote add origin [git@github.com](mailto:git@github.com):ly934609871/learngit.git 关联一个远程库

git push -u origin master 第一次推送master的所有分支

git push origin master 推送最新修改

**从远程库克隆**

git clone [git@github.com:ly934609871/gitckills.git](mailto:git@github.com:ly934609871/gitckills.git) 从远程库克隆到本地

**创建分支**

git branch 查看当前分支

git checkout -b dev = git branch dev + git checkout dev

创建并切换到dev分支

git branch dev 创建dev分支

git checkout dev 切换到dev分支

git merge dev 合并dev内容到当前分支

git branch -d dev 删除dev分支

git log --graph / git log --graph --pretty=oneline / git log --graph --pretty=oneline --abbrev-commit 查看分支合并图

**--no-ff方式的git merge**

git merge --no-ff -m “merge with no-ff”dev

git stash 手头工作没有完成，需要修复bug时

git stash pop 回到工作现场

git branch -D dev 强行删除dev分支

git remote 查看远程库的信息

git remote -v 查看远程库更详细的信息

**推送分支**

git push origin master 推送分支

git push origin dev 推送到dev分支

**多人协作**

git remote -v 查看远程库信息

git checkout -b dev origin/dev 在本地建立和远程库对应的分支

git push origin dev 推送自己的修改

推送失败时

1.合并 git pull

2.若合并有冲突，先设置dev与origin/dev的链接

git branch --set-upstream-to=origin/dev dev

再pull git pull

3.git push origin dev 推送

**标签**

git tag <tagname> 创建一个新标签

git tag 查看所有标签

git tag <tagname> <commit id> 对指定id创建标签

git show <tagname> 查看标签信息

git tag -a <tagname> -m “说明文字” <commit id> 带有说明的标签

git tag -d <tagname> 删除标签

git push origin <tagname> 推送标签到远程

git push origin tags 推送本地所有标签到远程

git push origin :refs/tags/<tagname> 删除远程标签