

## 1、git add .

git diff 显示工作区和暂存区的差异

git diff HEAD 显示工作区与当前分支最新commit之间的差异

git diff --cached 比较暂存区和版本库差异

## 2、git checkout [file] 恢复暂存区的指定文件到工作区

git checkout . 恢复暂存区的所有文件到工作区

git reset [file] git reset [commit] 重置暂存区的指定文件，与上一次commit保持一致，但工作区不变

git reset --hard git reset --hard [commit] 重置暂存区与工作区，与上一次commit保持一致

git reset --keep [commit] 重置当前HEAD为指定commit，但保持暂存区和工作区不变

git revert HEAD git revert [commit] 恢复最后一次提交的状态

git stash 暂时将未提交的变化移除，稍后再移入

git stash apply

git stash drop

git stash list

## 3、git commit -m [message]

git commit [file1] [file2] ... -m [message]

git commit -a 提交工作区自上次commit之后的变化，直接到仓库区

git commit -v 提交时显示所有diff信息

git commit --amend -m [message] 如果代码没有任何新变化，则用来改写上一次commit的提交信息# 使用一次新的commit，替代上一次提交

git commit --amend [file1] [file2] ... 重做上一次commit，并包括指定文件的新变化

## 4、git status 显示分支，未跟踪文件，更改和其他不同

git push [remote] [branch] git push 相当于 git push origin master# git push <远端> <分支>  
+ 本地分支的分支更新到远端+ 分支的分支更新到远端+ 分支的分支更新到远端+ 分支的分支更新到远端

又># 把本地的分支更新到远端origin的master分支上# 上传本地指定分支到远程仓库

git push [remote] --force 强行推送当前分支到远程仓库, 即使有冲突

git push [remote] --all 推送所有分支到

5、git branch #列出本地所有的分支

git branch -r #列出所有远程分支

git branch -v #列出本地所有的分支, + hash 信息

git branch -vv #列出本地所有的分支, + hash 信息 + 与远程的关联信息

git branch -a #列出所有的分支(远程和本地)

git branch [branch-name] #创建一个新的分支

git branch -m [branch-name] [new-branch-name] #git branch -m <旧名称> <新名称>#  
重命名分支

git branch [branch-name] --edit-description # 编辑分支的介绍

git checkout -b [branch] # 新建一个分支, 并切换到该分支

git branch [branch] [commit] # 新建一个分支, 指向指定commit

git branch --track [branch] [remote-branch] # 新建一个分支, 与指定的远程分支建立追踪关系

git checkout [branch-name] # 切换到指定分支, 并更新工作区

git checkout - # 切换到上一个分支

git branch --set-upstream [branch] [remote-branch] # 建立追踪关系, 在现有分支与指定的  
远程分支之间

git merge [branch] # 合并指定分支到当前分支

git cherry-pick [commit] # 选择一个commit, 合并进当前分支

git branch -d [branch-name] # 删除分支

git branch -D [branch-name] # 强制删除分支

git push origin --delete [branch-name] git branch -dr [remote/branch] # 删除远程分支

`git checkout [commit]` #把某次历史提交记录checkout出来，但无分支信息，切换到其他分支会自动删除

`git checkout [commit] -branch [branch-name]` # 把某次历史提交记录checkout出来，创建一个分支

`git branch --merged` # 查看已经被合并到当前分支的分支

`git branch --no-merged` #查看尚未被合并到当前分支的分支