# Git笔记

1. 概念
2. Git是分布式的版本控制系统。
3. .git下面的index是Git的暂存区
4. git diff是查看文件在工作目录和暂存区的区别

git diff HEAD 比较的是工作目录和本地仓库的区别

git diff --cached 比较的是暂存区和本地仓库的区别

4. 把整个文件从暂存区删除掉 git rm --cached 文件名

1. Git存储
2. 暂存区存放的是文件的名称以及内容的SHA值，对应的blob对象(即文件的sha1值和文件内容)是存在在git的对象库中(.git/objecs)

|  |  |
| --- | --- |
| name | a.txt |
| hash | 40bd001563085fc35165329ea1ff5c5ecbdbbeef |

1. 查看暂存区有哪些文件

git ls-files -s

查看某个commit点所指向的目录树的文件列表

git ls-tree [commit的SHA1值]

查看某个sha1值所对应的文件内容(不区分暂存区和本地仓库)

git cat-file -p [sha1值]

查看文件a.txt在暂存区中的内容

git cat-file -p :文件名

查看文件a.txt在本地仓库中的内容

git cat-file -p 分支名:文件名

1. 每个分支在./git/refs/heads下面有一个对应的文件，文件的内容是所对应的commit点的SHA1值。 HEAD是分支的引用，HEAD的内容为当前所在分支的名称。
2. 比如当前所在分支为dev，基于master分支创建分支的命令：

git branch bug\_22 master[或对应的SHA1值]

1. 用暂存区的某个文件的内容覆盖工作目录下该文件的内容(即撤销工作区的修改，对于未纳入git管理的文件不起作用)

git checkout filename(.代表全部)

用本地仓库的某个文件的内容同时覆盖暂存区和工作目录该文件的内容(即撤销工作区的修改和暂存区的修改，对于未纳入git管理的文件不起作用)

git checkout HEAD filename(.代表全部)

1. git stash 可以把当前工作区和暂存区的修改压入栈中不提交，然后再切换分支

git stash

git stash类似于栈，最新的修改压入栈顶 即 stash@{0}

查看stash@{0}的内容：git stash show -p stash@{0}

弹出栈 git stash pop stash@{0}(栈中删除此次修改)

git stash apply stash@{0}(栈中还有保存此次修改)

查看stash列表 git stash list

删除某个stash git stash drop stash@{0}

1. Git分支
2. Fast-Forward合并不会形成新的合并commit点，只需要修改分支对应的commit值。
3. 三方合并会形成一个新的commit点

四、远程仓库pull/push

在push时，要求远程仓库无新的提交，否则报错；必须先pull，完成本地分支与远程分支在本地的合并，然后推送到远程仓库，此时在远程仓库完成的是远程仓库当前分支与本地分支的Fast-Forward合并。

push时，在远程仓库发生的是Fast-Forward.