Git 笔记

一、概念

1. Git 是分布式的版本控制系统。
2. .git 下面的 index 是 Git 的暂存区
3. 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

二、Git 存储

1. 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

2. 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

3. 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。

4. 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]

5. 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)

6. 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}

三、Git 分支

1. 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。
2. 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.