**Git的使用**

**（本地库和远程库只是纯粹的映射关系，分支都是对应的，如果没有对应的可以建立对应，然后才能推送和拉取，也因为这种映射关系，导致推送和拉去过程中会有冲突问题，Push时就如同本地的两个分支合并一样，pull时则是push的反向。）**

本地库、远程库的建立映射：添加好公钥后，

git remote add origin git@github.com:michaelliao/learngit.git

本地分支远程分支建立映射：（建立映射后就可以push到对应的分支了，并且如果名字一样的话就不需要在命令后面添加远程分支名了）建立映射后，本地的.git文件目录：.git/refs/remotes/【正在跟踪的分支】下就有了关联的远程分支（. git/refs/head/【本地分支】），于是git pull和git fetch的区别就出来了：git fetch 的作用就是把远程的提交commitid更新到.git/refs/remotes/【正在跟踪的分支】对应的分支下，然后再git merge 对应的分支，合并到. git/refs/head/【本地分支】对应的分支下，然后git commit，git push就可以了；而git pull则是同时将远程的更新commitid更新到.git/refs/remotes/【正在跟踪的分支】以及 . git/refs/head/【本地分支】，相当于直接把git merge 也一起做了。

本地分支跟远程分支进行关联：git branch --set-upstream-to=origin/远程名 本地名

本地的新建的分支推送到远程（远程没有）：git push --set-upstream origin 分支名

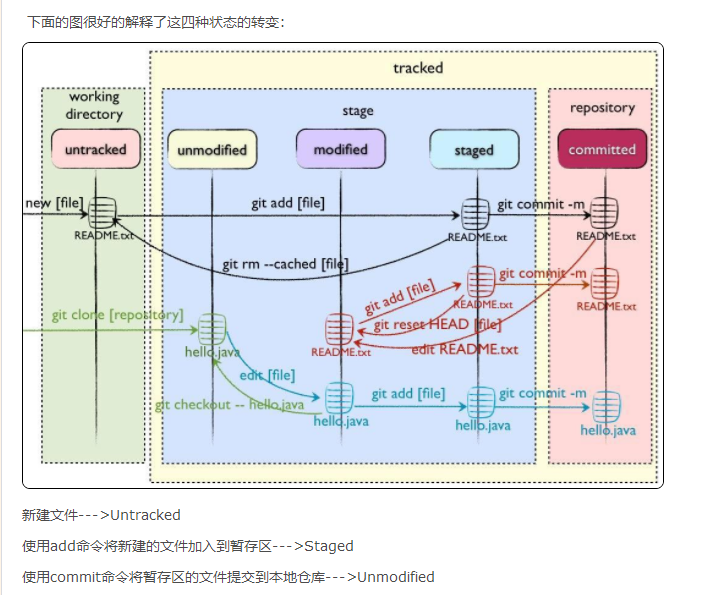
远程分支拉取到本地（本地没有）：git checkout -b 本地分支名 origin/远程分支名（不行就先git fetch）

一般从远程库clone时只克隆其中一个分支。

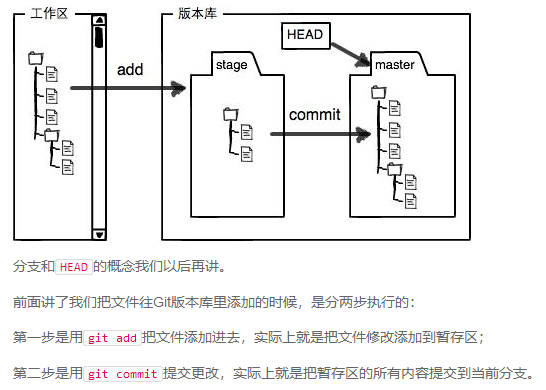
查看远程分支和本地分支的关联关系：git branch -avv

查看远程地址：git remote -v

查看远程/本地所有分支：git branch -a



1、只有在git仓库中的文件才能被管理，可以创建N个库；并且库中的文件显示的是head指向的分支；git reset的命令实质上是跳转到某个分支，不光可以回退也可以穿梭到未来存在的分支；下面是本地库的存储分配方式：



2、分支的使用

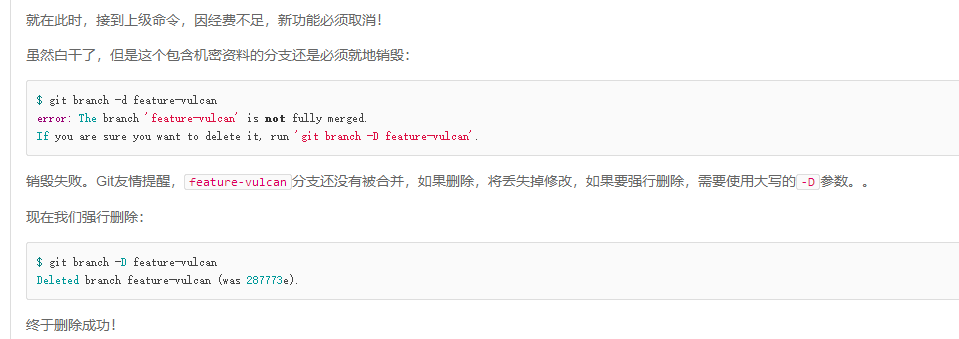




3、对于修复bug和开发新功能，都要新建分支，完成后再删除分支，有紧急bug要修复时：



新开发功能时，正常情况下新建分支，然后提交、合并，删除分支，但当开发好后git add之后，还没提交就要删除：



合并时禁用fast模式：

：

4、多人协作

