git分支管理说明文档

针对一个项目，默认拥有一个master分支，通常用来发布稳定版本，极少进行更改。为此，在开发过程中，可创建一个develop分支

1.创建develop分支（从master分支拉出）

git checkout -b develop master

创建的develop为开发分支，通常情况下作为开发的主线，不同版本的开发代码最终都将合到develop分支上。

\*一般情况下，一个项目只有master和develop两个分支长期存在，其他延伸出来的分支在完成其使命后需要进行删除

2.develop分支延伸

develop作为开发的主线，需要合并起多人开发的代码，而在开发过程中，直接在develop上进行开发并不合适，因此，可以从develop分支再拉出不同的分支：

1）feature分支

git checkout -b featureXXXX develop

feature分支作为新需求的分支，不同开发人员都可从develop分支拉出

2）release分支

git checkout -b releaseXXXX develop

release分支作为预发布的分支

3.fixbug分支

从master分支拉出，用于解决正式版本上出现的bug，解决完之后，需要合并到master和develop分支上

\*\*1--3所创建出来的分支存在于本地仓库上，如要推送到远程仓库github上，使用命令

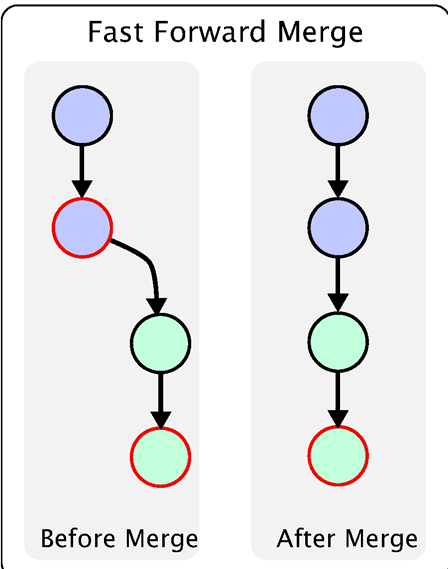
git push origin develop（feature/release/fixbug）

4.合并分支（举例：将feature合并到develop分支） git branch

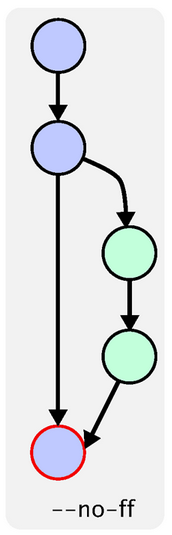
1）切换到develop分支 git checkout develop

2）git merge --no-ff feature

默认合并是快进式合并，会指向feature分支



--no-ff采用之后则从develop分支拉出节点来



合并之后需要push到远程仓库

5.版本回退

1）git log首先查看日志，确定要回退到哪个版本（日志填写很重要），获取到版本的commit id

2）git reset --hard \*\*\*\*\* （达到本地退回）

3)强调：原则上来说，回退到本地某一版本之后，对于远程仓库是不允许修改的，如果需要推送，使用：

git push origin 分支名 -f

-f起到强制作用，因为本地的版本相对较低

\*\*\*回退之后，做了修改进行push或pull可能会出现本地仓库分支对应不到远程仓库分支的现象，需要（git pull origin 分支名）

可使用

git branch --set-upstream feature origin/分支名

这样每次pull/push就无需origin

5.删除分支：

git branch -d 分支名 （本地仓库删除）

git push origin ：分支名 （远程仓库删除，意思为推送一个空的分支替换掉远程分支）

6.冲突解决

>>>>>>>>HEAD

<<<<<<<<<

大致示意图

