git高级相关的操作：

1. git reset：简单点就是用于“撤销提交、回退代码、修改历史”回退的命令。

命令格式：git reset [模式] [提交ID]。 其中模式有三种：hard soft max

--soft模式：只会回退commit，工作区、暂存区的代码不会回退

用法一：修改commit信息。比如你本地git commit后，但是想要修改commit的消息，你可以git reset到上一个commit，然后重新commit。

用法二：压缩合并commit。你可以git reset –soft head~10，回退到前面第10个commit，并且所有的工作区、暂存区的代码都不会变。然后commit提交一下，就可以将10个commit压缩为一个commit。

--maxed：回退暂存区代码，工作区不变。

用法一：用于撤销git add的代码，工作区不变，不会影响当前的代码。

--hard:彻底回退commit，工作区、暂存区全部回退到之前的commit。并且不可恢复

用法一：适合用于彻底放弃最近的改动。比如生产环境有问题，需要回退代码快速解决问题。

二．git merge

使用情况就是，多人开发时（A、B两人），大家都从一个feature分支拉去自己的分支。A提交了一个pr给feature分支，此时B需要用到A提交的代码，那么B就可以在自己的分支下git merge feature分支。（可能会出现冲突，可以手动解决冲突后，git并不知道冲突已经解决了，需要手动git add 冲突文件。最后git commit合并提交）

三．git rebase

和git merge的使用场景一样，但是并不会让你的分支出现merge的commit记录。git merge的逻辑是：A提交pr给feature后，B想要用到A提交的代码，就不要去git merge feature，而是通过git rebase feature。此时B的分支会回退到A和B共同的c**ommit祖先位置**，将A的提交全部应用到B分支上，然后再将B的提交应用上即可（此时会可能会出现冲突，手动解决冲突后需要git add冲突文件，然后继续git rebase –continue即可）。git rebase并不会像git merge一样将代码合并到自己的分支上，而是会启动一个零食分支用于处理git rebase，所以当你不想要git rebase时，可以通过git rebase –abort放弃本次git rebase。