1、git一些常用的命令介绍：

1. git init 创建一个空的仓库
2. git status 查看状态
3. git add 添加文件
4. git commit –m ‘注释’ 提交添加的文件并备注说明
5. git remote add origin [git@gitlub.com:\*\*\*/git.git](mailto:git@gitlub.com:***/git.git) 连接远程仓库
6. git push –u origin master 将本地仓库文件推送到远程仓库
7. git log 查看变更日志
8. git branch 查看分支
9. git branch newname 创建一个叫newname的分支

10、git checkout newname 切换到叫newname的分支上

11、git merge newname 把newname分支合并到当前分支上

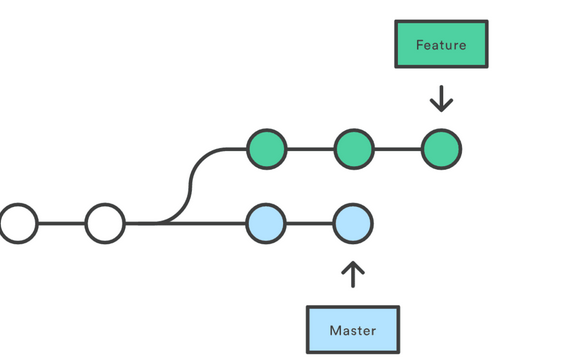
12、git pull origin master 将master分支上的内容拉到本地上

2、merge 和 rebase

在git中，可以使用git merge 和 git rebase 两个命令来进行分支的合并。

Git rebase 和 git merge 一样都是用于从一个分支获取并且合并到当前分支，但是采取的是不同的工作方式

工作场景：



在一个feature分支进行新特性的开发，与此同时，master分支也有新的提交，为了将master上新的提交合并到你的feature分支上，有以下两种选择：

Merging or rebasing

Merge:

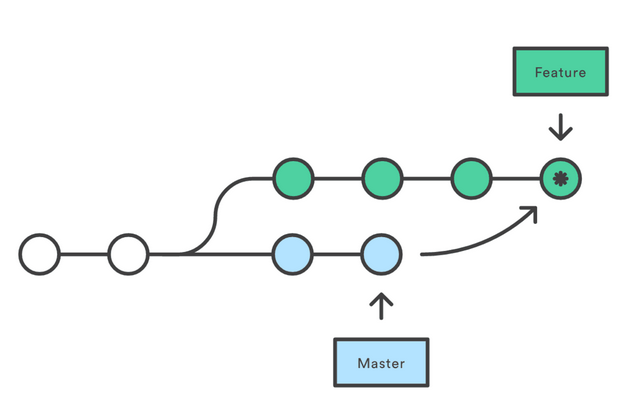
执行以下命令：

git checkout feature

git merge master

或者更简单的git merge master feature

那么此时在feature上git 自动会产生一个新的commit(merge commit)



marge 特点：自动创建一个新的commit

  如果合并的时候遇到冲突，仅需要修改后重新commit   
优点：记录了真实的commit情况，包括每个分支的详情   
缺点：因为每次merge会自动产生一个merge commit，所以在使用一些git的 GUI tools，特别是commit比较频繁时，看到分支很杂乱。

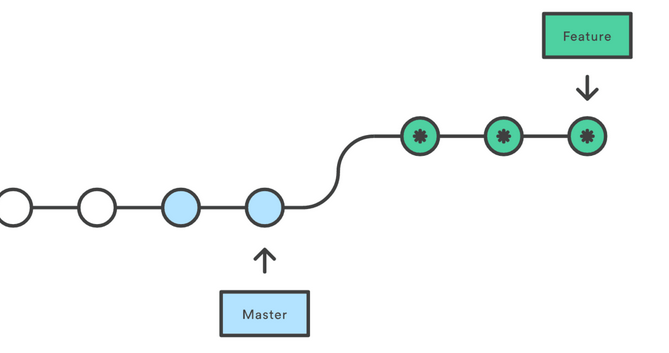
Rebase(变基)：

作用：可以对某一段线性提交历史进行编辑、删除、复制、粘贴；因此，合理使用rebase命令可以使我们的提交历史干净、简洁！

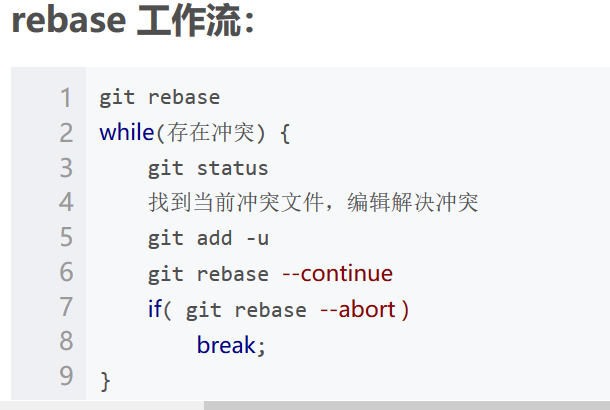
执行以下命令：

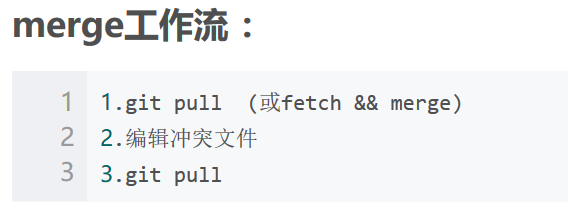
git checkout feature

git rebase master



rebase 特点：会合并之前的commit历史   
优点：得到更简洁的项目历史，去掉了merge commit   
缺点：如果合并出现代码问题不容易定位，因为re-write了history

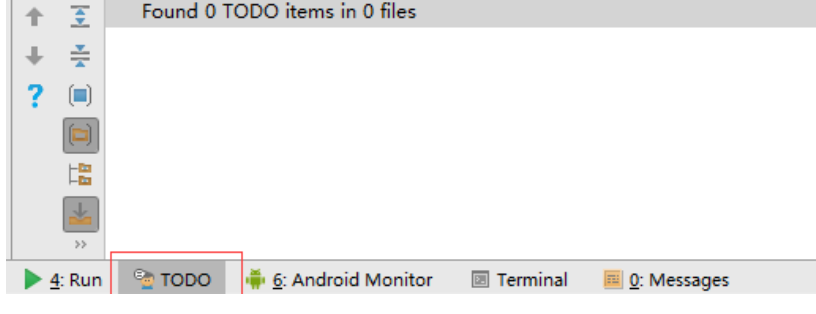




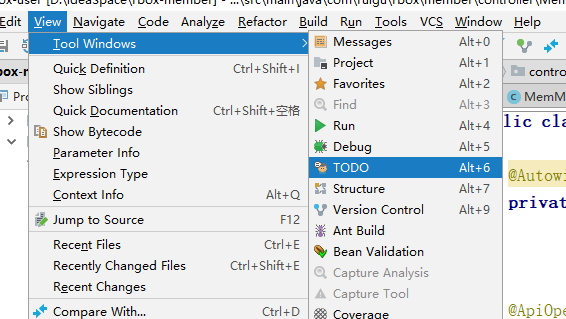
3、Check TODO：

TODO:表示在此处将要实现的功能，提醒你在后续阶段将会在此处添加代码，以表示待完成或待解决的问题

1、添加TODO：在你需要注释的地方输入TODO

2、查看TODO：左下角有一个TODO按钮，点开即可以查看

3、如果没有，可以在左上角的菜单可以打开。



4、完成TODO标记的事件之后，可以删除TODO注释

4、版本回退：

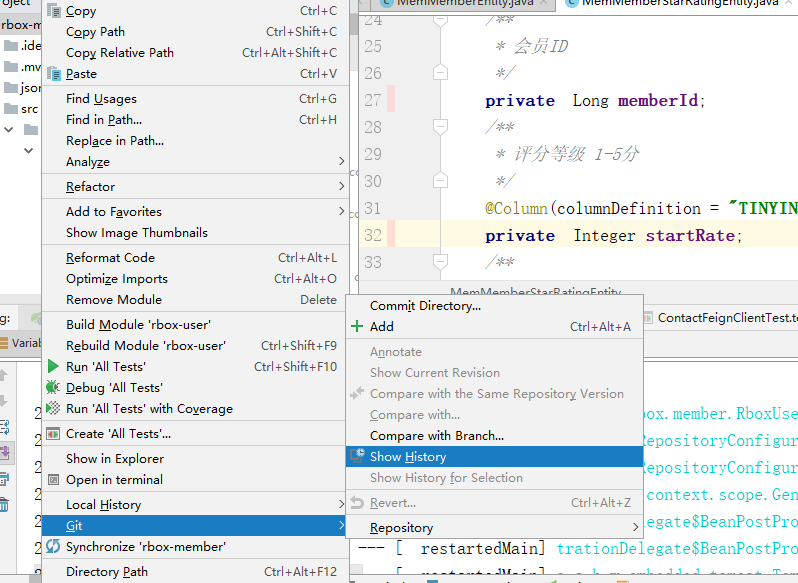
把当前版本回退到上一个版本 命令：git reset

想回退到最新版本：使用命令 git reset –hard 版本号

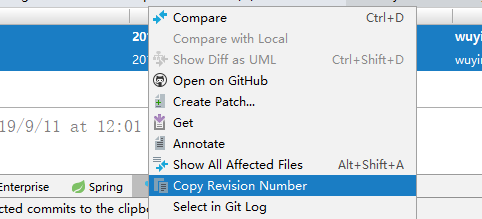
如果不知道版本号可以使用命令获取版本号：git reflog

Idea操作：

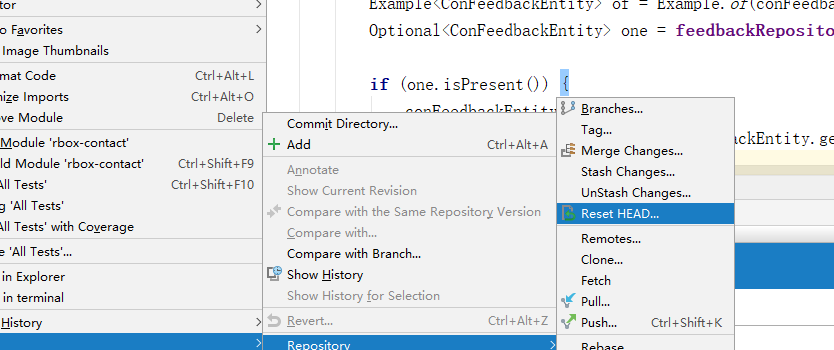
1. 找到show history



1. 选择版本号之后点击copy revision number 复制版本号



3、选择reset hard



4、reset type选择hard ，点击reset

