用例1：回滚GitLab或者Github的远端代码库提交

我们在本地commit代码，然后也成功push到了Gitlab远端代码库上。此时我们将撤回本次提交。

方法1（保留提交记录）：

我们在远端的代码库上，选择该提交，然后鼠标点击revert.这样远端数据库的上一次提交将撤回，并且生成一次提交记录。然后我们到本地代码库上同步， git pull. 此时远端本地都会撤回上一次提交，但是本地和远端的代码库都会保留上一次的提交记录以及撤回提交记录

方法2 （不保留提交记录）：

我们现在本地代码库上，git reset –hard commit\_id. 此时本地代码库的提交将被撤回，并且提交记录将被消除。此时我们要让远端代码库也撤回操作，并且不保留提交记录。 此时我们在本地库上无法直接push, 因为本地的提交记录已经落后与远端的提交记录无法push. 我们需要git push origin 分知名 –force进行提交，让远端撤回上一次提交并且消除提交记录。[ git push只能用在非保护分支上，所以如果想对master分支进行该操作，需要管理员同意]

2.如果我们想放弃所有本地工作，可以使用git reset –hard commit\_id, 这样会把本地目录以及文件，暂存区，以及提交记录全部回到commit\_id的地方，达到请空所有修改的目的。

3.提交回滚操作

git reset --hard HEAD～ 回滚一次提交操作， 文件更改回到更改前的状态，log取消上一次操作记录， index中取消更改状态记录

git reset --soft HEAD～ 回滚一次提交操作， 文件更改内容保存，log取消上一次操作记录，staging area不会取消更改状态记录

git reset HEAD～ 回滚一次提交操作， 文件所更改内容保存，log取消上一次操作记录，staging area中取消更改状态记录

另外我们在本地working directory下面做的修改，经过多次commit以后，会产生多次的commit log.我们不仅仅可以回滚到上一次，还可以回到到前几次。

git reset –-(mixed|hard|soft) <commit sha1 | reference>

--mixed 代表修改仍然留在working directory中，但是staging area记录清除，repository，log清除

--hard 完全回到原状态，包括working directory中的内容也会到原样, 达到放弃修改本地文件，还原本地文件的作用

--soft 修改仍然留在working directory中，staging area记录不被清除， log被清除，