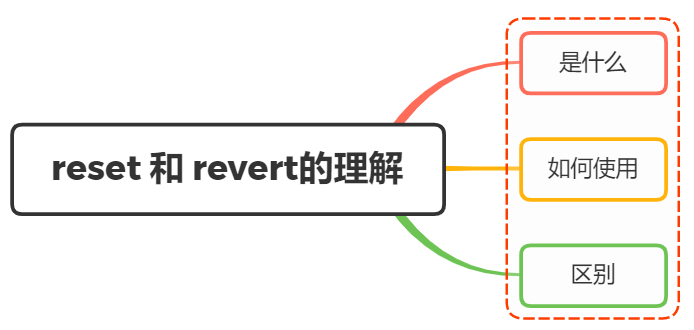
# 面试官：说说你对git reset 和 git revert 的理解？区别？

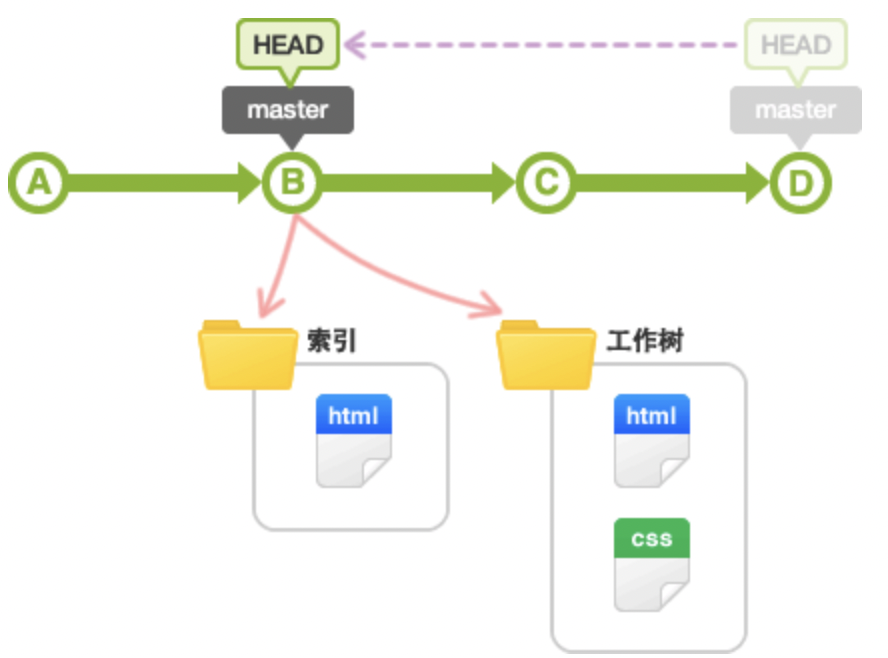


## 一、是什么

### git reset

reset用于回退版本，可以遗弃不再使用的提交

执行遗弃时，需要根据影响的范围而指定不同的参数，可以指定是否复原索引或工作树内容



### git revert

在当前提交后面，新增一次提交，抵消掉上一次提交导致的所有变化，不会改变过去的历史，主要是用于安全地取消过去发布的提交



## 二、如何用

### git reset

当没有指定ID的时候，默认使用HEAD，如果指定ID，那么就是基于指向ID去变动暂存区或工作区的内容

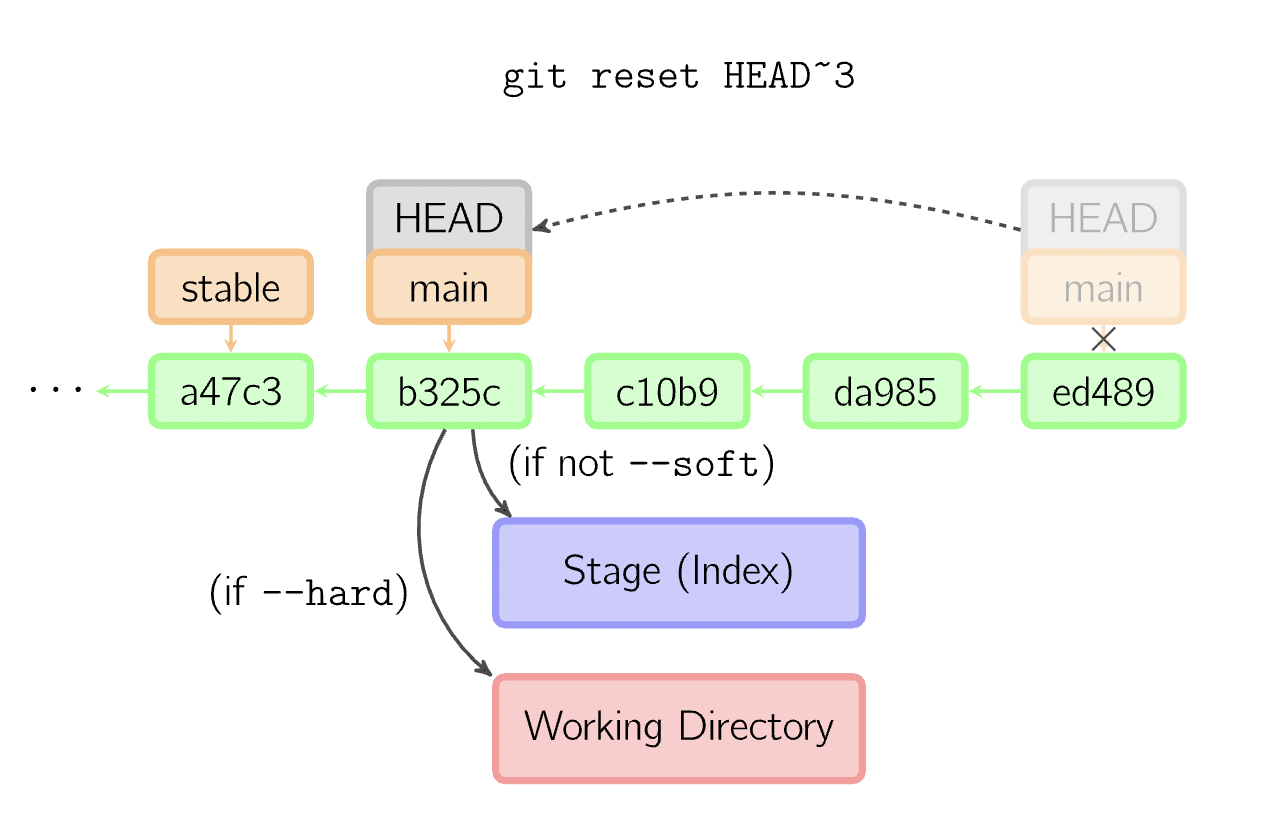
// 没有指定ID, 暂存区的内容会被当前ID版本号的内容覆盖，工作区不变  
git reset  
  
// 指定ID，暂存区的内容会被指定ID版本号的内容覆盖，工作区不变  
git reset <ID>

日志ID可以通过查询，可以git log进行查询，如下：

commit a7700083ace1204ccdff9f71631fb34c9913f7c5 (HEAD -> master)  
Author: linguanghui <linguanghui@baidu.com>  
Date: Tue Aug 17 22:34:40 2021 +0800  
  
 second commit  
  
commit e31118663ce66717edd8a179688a7f3dde5a9393  
Author: linguanghui <linguanghui@baidu.com>  
Date: Tue Aug 17 22:20:01 2021 +0800  
  
 first commit

常见命令如下：

* --mixed（默认）：默认的时候，只有暂存区变化
* --hard参数：如果使用 --hard 参数，那么工作区也会变化
* --soft：如果使用 --soft 参数，那么暂存区和工作区都不会变化



### git revert

跟git reset用法基本一致，git revert 撤销某次操作，此次操作之前和之后的 commit和history都会保留，并且把这次撤销，作为一次最新的提交，如下：

git revert <commit\_id>

如果撤销前一个版本，可以通过如下命令：

git revert HEAD

撤销前前一次，如下：

git revert HEAD^

## 三、区别

撤销（revert）被设计为撤销公开的提交（比如已经push）的安全方式，git reset被设计为重设本地更改

因为两个命令的目的不同，它们的实现也不一样：重设完全地移除了一堆更改，而撤销保留了原来的更改，用一个新的提交来实现撤销

两者主要区别如下：

* git revert是用一次新的commit来回滚之前的commit，git reset是直接删除指定的commit
* git reset 是把HEAD向后移动了一下，而git revert是HEAD继续前进，只是新的commit的内容和要revert的内容正好相反，能够抵消要被revert的内容
* 在回滚这一操作上看，效果差不多。但是在日后继续 merge 以前的老版本时有区别

git revert是用一次逆向的commit“中和”之前的提交，因此日后合并老的branch时，之前提交合并的代码仍然存在，导致不能够重新合并

但是git reset是之间把某些commit在某个branch上删除，因而和老的branch再次merge时，这些被回滚的commit应该还会被引入

* 如果回退分支的代码以后还需要的情况则使用git revert， 如果分支是提错了没用的并且不想让别人发现这些错误代码，则使用git reset

## 参考文献

* https://juejin.cn/post/6844903542931587086
* https://marklodato.github.io/visual-git-guide/index-zh-cn.html#reset