https://www.cnblogs.com/chenlogin/p/6592228.html

**[git fetch 更新远程代码到本地仓库](https://www.cnblogs.com/chenlogin/p/6592228.html)**

理解 fetch 的关键, 是理解 FETCH\_HEAD，FETCH\_HEAD指的是: 某个branch在服务器上的最新状态’。这个列表保存在 .Git/FETCH\_HEAD 文件中, 其中每一行对应于远程服务器的一个分支。

当前分支指向的FETCH\_HEAD, 就是这个文件第一行对应的那个分支.

一般来说, 存在两种情况:

* 如果没有显式的指定远程分支, 则远程分支的master将作为默认的FETCH\_HEAD.
* 如果指定了远程分支, 就将这个远程分支作为FETCH\_HEAD.

git fetch origin branch1

这个操作是git pull origin branch1的第一步, 而对应的pull操作,并不会在本地创建新的branch。设定当前分支的 FETCH\_HEAD' 为远程服务器的branch1分支`。

这个命令可以用来测试远程主机的远程分支branch1是否存在, 如果存在, 返回0, 如果不存在, 返回128, 抛出一个异常.

git fetch origin branch1:branch2

首先执行上面的fetch操作，使用远程branch1分支在本地创建branch2(但不会切换到该分支),如果本地不存在branch2分支, 则会自动创建一个新的branch2分支,

如果本地存在branch2分支, 并且是`fast forward', 则自动合并两个分支, 否则, 会阻止以上操作.

fetch更新本地仓库两种方式：

[复制代码](javascript:void(0);)

//方法一

$ git fetch origin master //从远程的origin仓库的master分支下载代码到本地的origin master

$ git log -p master.. origin/master//比较本地的仓库和远程参考的区别

$ git merge origin/master//把远程下载下来的代码合并到本地仓库，远程的和本地的合并

//方法二

$ git fetch origin master:temp //从远程的origin仓库的master分支下载到本地并新建一个分支temp

$ git diff temp//比较master分支和temp分支的不同

$ git merge temp//合并temp分支到master分支

$ git branch -d temp//删除temp

[复制代码](javascript:void(0);)

1、git reset

没有push，这种情况发生在你的本地代码仓库,可能你add ,commit 以后发现代码有点问题.

首先，Git必须知道当前版本是哪个版本，在Git中，用HEAD表示当前版本，也就是最新的提交commit\_id(79f673d631b08907496ce792f429e1f00da25b73)，上一个版本就是HEAD^，上上一个版本就是HEAD^^，当然往上100个版本写100个^比较容易数不过来，所以写成HEAD~100。

* HEAD指向的版本就是当前版本，因此，Git允许我们在版本的历史之间穿梭，使用命令git reset --hard 79f673d631b08907496ce792f429e1f00da25b73。
* 穿梭前，用git log可以查看提交历史，以便确定要回退到哪个版本。
* 要重返未来，用git reflog查看命令历史，以便确定要回到未来的哪个版本。

2、git revert

已经push，对于已经把代码push到线上仓库,你回退本地代码其实也想同时回退线上代码,回滚到某个指定的版本,线上,线下代码保持一致.你要用到下面的命令

git revert用一个新提交来消除一个历史提交所做的任何修改.

revert 之后你的本地代码会回滚到指定的历史版本,这时你再 git push 既可以把线上的代码更新.(这里不会像reset造成冲突的问题)

revert 使用,需要先找到你想回滚版本唯一的commit标识代码,可以用 git log 或者在adgit搭建的web环境历史提交记录里查看.

git revert c011eb3c20ba6fb38cc94fe5a8dda366a3990c61

3、两者区别

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看似达到的效果是一样的,其实完全不同.

第一:上面我们说的如果你已经push到线上代码库, reset 删除指定commit以后,你git push可能导致一大堆冲突（或git push -f强制推送）.但是revert 并不会.

第二:如果在日后现有分支和历史分支需要合并的时候,reset 恢复部分的代码依然会出现在历史分支里.但是revert 方向提交的commit 并不会出现在历史分支里.

第三:reset 是在正常的commit历史中,删除了指定的commit,这时 HEAD 是向后移动了,而 revert 是在正常的commit历史中再commit一次,只不过是反向提交,他的 HEAD 是一直向前的.

[**好文要顶**](javascript:void(0);) [**关注我**](javascript:void(0);) [**收藏该文**](javascript:void(0);) **[https://common.cnblogs.com/images/icon_weibo_24.png](javascript:void(0);)** **[https://common.cnblogs.com/images/wechat.png](javascript:void(0);)**

[https://pic.cnblogs.com/face/343184/20150813214653.png](http://home.cnblogs.com/u/chenlogin/)

[圣耀](http://home.cnblogs.com/u/chenlogin/)  
[关注 - 0](http://home.cnblogs.com/u/chenlogin/followees)  
[粉丝 - 14](http://home.cnblogs.com/u/chenlogin/followers)

[+加关注](javascript:void(0);)

4

0

[«](https://www.cnblogs.com/chenlogin/p/6591918.html)上一篇：[Git 同步远程仓库](https://www.cnblogs.com/chenlogin/p/6591918.html)  
[»](https://www.cnblogs.com/chenlogin/p/6607436.html)下一篇：[微信小程序](https://www.cnblogs.com/chenlogin/p/6607436.html)