

git pull和git fetch区别

git有个本地库和远程副本库。我们使用git pull。会从远程仓库中拉取数据，并更新本地库和远程副本库。如果使用git fetch，只是将数据拉取到远程副本库，本地库未发生变化，然后我们要通过git merge进行合并操作。最终更新代码。git pull就相当于fetch和merge的合体。没冲突的会自动合并，有冲突会提示，进行merge。

git pull发生冲突的解决方式

1、忽略本地修改，强制拉取远程到本地

主要是项目中的文档目录，看的时候可能多了些标注，现在远程文档更新，本地的版本已无用，可以强拉

```
git fetch --all
git reset --hard origin/dev
git pull
```

2、未commit先pull，视本地修改量选择revert或stash

同事 有新提交
我 没有pull -> 修改了文件 -> pull -> 提示有冲突

修改量小的时候，直接revert。修改量大的时候，可以先用idea的stash将自己的代码隐藏起来，然后pull拉取过来，然后再展示自己的代码，然后编辑冲突。

3、已commit未push，视本地修改量选择reset或直接merge

同事 有新提交
我 没有pull -> 修改了文件 -> commit -> pull -> 提示有冲突

reset head，使用hard模式，将远程副本库进行恢复。然后再进行pull，然后再修改。

也可以直接merge，然后再commit。

git fetch和git rebase的区别

git merge 的合并分支会让两个分支的每一次提交都按照提交时间，并且会将公共分支（master）和当前分支（feature）的最新一次commit点合并在一起，形成一个新的commit，commit的时间点在分支时间点之后。

git rebase操作实际上是将当前分支（feature）的所有commit点取消，保存成一个一个的临时patch（保存在.git/rebase"目录中），然后把当前分支（feature）更新到最新的原分支（master），最后把这些保存的临时patch文件，应用到当前分支（feature）上。这种会导致master的历史时间点都改变了。会把我们分支的commit的时间点放在主干的前面。就打乱且篡改了历史记录。也许会出现很多诡异现象。