你需要知道的几个Git Command——Git中的后悔药

10 September 2013

pull之后发现忘记加--rabase参数了

我本地做了若干commit，准备push到remote，push前我需要pull一下，merge别人的工作：

`git pull`。

如果没有冲突，它会merge并创建一个commit，可是我发现我忘了rebase了，我需要做`git pull --rebase`，而不是`git pull`。咋办?

方案:

`git reflog`

`git reset HEAD@{1}`

然后再做`git pull --rebase`

事实上 git reflog可以做很多事。参加这篇

想把上次或上上次的版本拿出来看看

我当前在master最新版本上，可是好像有个Unit Test没通过，可是不像是这次commit引起的，那么我想看看上个版本是不是也有这个问题?

用`git checkout HEAD~1`,重复运行它，将一直往回走。（显然`git checkout HEAD~2`就是上上个版本了。）

这时后进入的是**detached HEAD**状态，我可以“审阅历史”。审阅完毕，想回到最新状态?

`git checkout master`即可。

本地白做了，想全部作废——git clean与git reset双剑合璧

我在本地做了很多修改：修改了已存在的文件，增加了文件，还增加了文件夹。

我还没有commit，不过我发现我做的都是废活，白做了，需要全部作废，回到“干净”的状态。

对于untracked的文件/文件夹，也就是我在本地新加的那些

`git clean -dn`或者`git clean -n`会**干跑**一下，告诉你它将删除哪些新加的文件夹/文件。有`-d`会处理文件夹，否则只会处理文件。

确定没问题后，

`git clean -df`或者`git clean -f`就会真正执行了。`-f`是force的意思。

对于tracked的文件，也就是我修改的那些

比如2.txt被我修改了，我想丢弃这个修改，执行

`git checkout 2.txt`即可

但如果我修改了30个文件，我不想一个一个checkout，那么用

`git reset --hard`

可是我已经执行过`git add .`了。

也就是我已经把我修改的2.txt添加到stage了。

这时`git checkout 2.txt`不好使了，必须用

`git reset 2.txt`

note: *git reset* only apply to tracked files, need *git clean* to clean untracked fiels

想undo上一个commit怎么办?

git reset有三种模式：soft，hard，default

`git reset --soft head~1`: undo git commit, 但本地index会保留，也就是回到`git add .`之后的状态。

`git reset head~1`: undo git commit, 但本地index不会保留，但本地做的改变都还在。也就是回到`git add .`之前的状态。

`git reset --hard head~1`: 这个最猛，回到上个commit，本地鸡犬不留。

commit之后发现注释写错了

commit并push之后发现注释写错了

```
git commit --amend -m "update comments here"
```

之后，还需push这次amend到remote

`git push -f`：需加上`-f`（force的意思），否则不会成功。但**不推荐**这样做，因为你是在一个团队里，别人也在基于上次commit继续开发……

更多参见How do I push amended commit to the remote git repo?

← Previous | Archive | Next →