

1. git clone \*\*(项目地址)

克隆一个git项目到本地,将git项目拉取到本地

1. git status

查看文件状态,列出当前目录没有被git管理，以及被修改过还未提交的文件

1. git add \*

将我们提交的文件添加到索引库中（添加到缓冲区），\*可以是路径也可以是.符号，git add . 代表将当前目录下的所有文件都添加到索引库中,如果指定路径则代表将制定路径的文件添加到索引库中。

1. git commit -m "备注"

将文件推送到本地仓库中,-m 后可以填写此次提交的备注如git commit -m "提交删除功能代码"，那么在git项目中的提交记录里面就能看见你的推送备注。这一步仅仅是放在缓冲区中，还未真正提交代码

1. git push origin 分支名

这一步才是推送代码推送时需要跟分支名，表示需要将代码推送至某个分支.如git push origin dev表示你要讲代码推送至dev分支。

1. git pull

更新当前分支的代码,获取最新的代码，会自动merge

1. git fetch

同git pull但不会自动merge

1. git checkout 分支名

切换分支

1. git merge 分支名

当前分支合并其他分支

所以一般的使用简易流程 git add ， git commit -m "", git push origin 。 当然如果远端有代码更新的情况需要git pull更新一下，中间也可以使用git status 查看不同时期的状态

1. git branch 命令

不仅仅能创建和删除分支，如果不加任何参数，它会给出当前所有分支的清单

1. git config --global user.name “”

git config --global user.email “”

1. git rm

在这里说一下git rm和rm的区别，虽然觉得这个问题有点肤浅，但对于刚接触git不久的朋友来说还是有必要的。

用 git rm 来删除文件，同时还会将这个删除操作记录下来；

用 rm 来删除文件，仅仅是删除了物理文件，没有将其从 git 的记录中剔除。

直观的来讲，git rm 删除过的文件，执行 git commit -m "abc" 提交时，会自动将删除该文件的操作提交上去。

而用 rm 命令直接删除的文件，单纯执行 git commit -m "abc" 提交时，则不会将删除该文件的操作提交上去，需要在执行commit的时候，多加一个-a参数，

即rm删除后，需要使用git commit -am "abc"提交才会将删除文件的操作提交上去。