# 1. 常用命令

## commit

git commit -a -m 'tag' == git add .+ git commit -m 'tag'

(add的此处是添加到缓存区，如果文件还没有跟踪，需要先执行 git add进行跟踪)

## gitignore

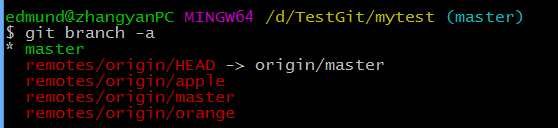
常用gitignore https://github.com/github/gitignore

## remote

git remote show origin 查看某一个远程仓库的更多信息

## clone

git clone url 克隆远程仓库到本地，并且在本地创建 master 分支和所有的远程跟踪分支，并不创建除master外的其他远程分支的本地Copy，如下图：



## fetch

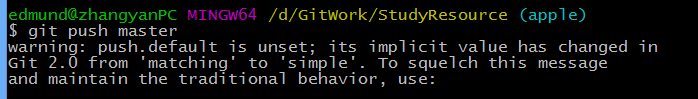
git fetch [remote] 会抓取新的远程跟踪分支，不会自动生成可编辑的本地Copy（同clone）。

可以使用 git checkout -b branchname remote/branchname建立一个本地工作分支。

## push

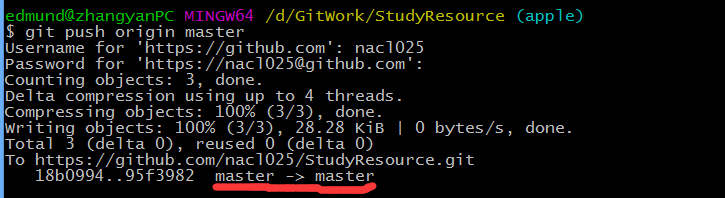
**git push【branchname】 会自动推送到本地分支相对应的远程分支上。**

注： branchname可以缺省，如果填写，必须为本地head分支名，例如：当前处于head为apple分支，执行git push master，推送失败，如下图：



**git push remotename branchname【:otherbranchname】 把本地分支推送到远程分支，**

注：1 branchname不一定是head所在的分支，而是branchname所代表的分支，例如：在本地apple分支上执行git push origin master,它的执行结果是把本地的master分支，推送到远程的master分支。如下图：



2 otherbranchname如果缺省，则推送到远程的branchname分支。另外，在远程必须真实存在，如果不存在会推送失败

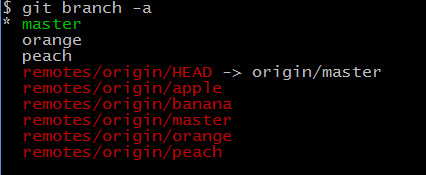
## branch

git branch -vv 将所有的本地分支列出来并且包含更多的信息，如每一个分支正在跟踪哪个远程分支与本地分支是否是领先、落后或是都有。

git branch -a 查看所有分支。包括远程分支在本地的远程跟踪分支（分支快照）：

# 2.使用

远程跟踪分支



其中 remotes/origin/branch-name 为远程跟踪分支，在本地可以编辑远程跟踪分支，但无法保存（及在切换到其他分支时，远程跟踪健会复原）。所以尽量不要操作远程跟踪分支。