# 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

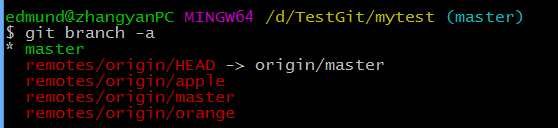
git remote show origin 查看某一个远程仓库的更多信息

git remote add 命令添加一个新的远程仓库引用到当前的项目(git remote add origin git://127.0.0.1/testgit.git)

git remote remove <name> 删除一个远程仓库引用

## clone

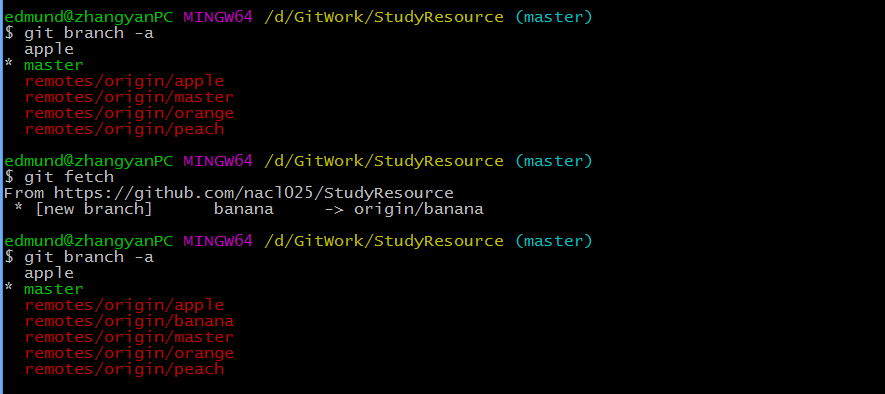
git clone url 克隆远程仓库到本地，并且在本地创建 master 分支和所有的远程跟踪分支，并不创建除master外的其他远程分支的本地Copy，如下图：



## 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）。

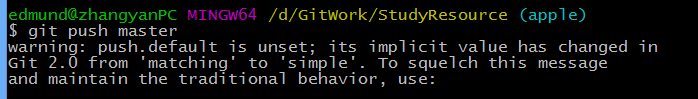
可以使用 git checkout -b branchname remote/branchname（快捷方式git checkout --track remote/branchname ）建立一个本地工作分支。如下图：

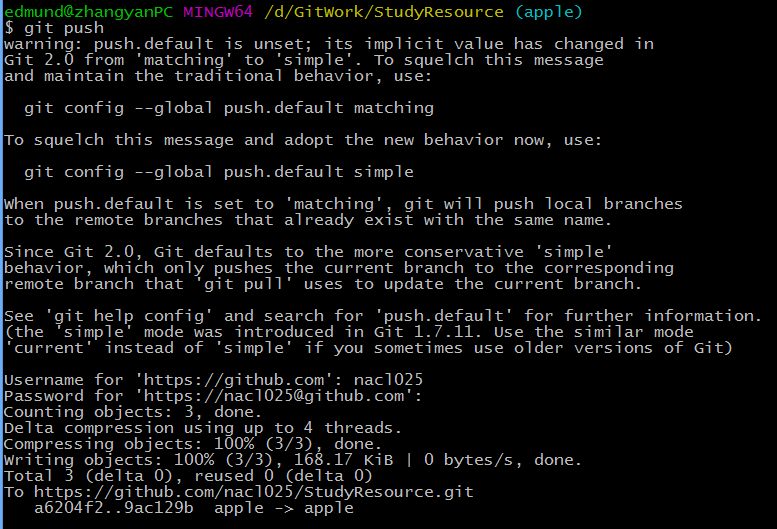


## push

**git push【branchname】 把本地head所在的分支推送远程相应分支上。**

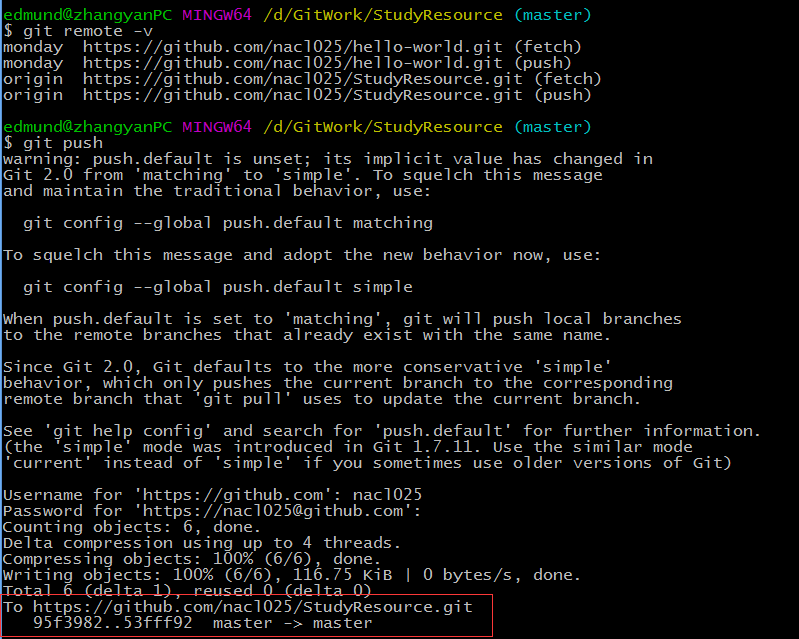
注：1 branchname可以缺省，如果填写，必须为本地head分支名，例如：当前处于head为apple分支，执行git push master，推送失败，如下图：





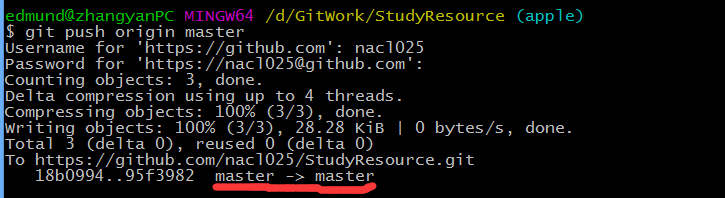
2 q:如果有两个以上remote会怎样？

a:自动匹配origin remote,如果想上传到除origin外的其他remote，需要加上remotename,如下一条指令所示。

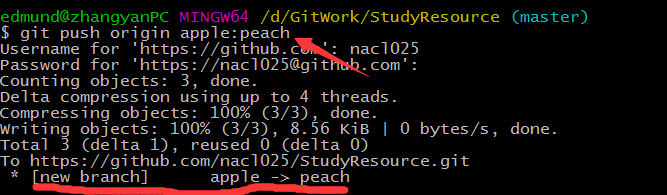


**git push remotename branchname【:otherbranchname】 把本地分支推送到远程分支，**

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otherbranchname如果不存在，会新建一个分支，并在本地新建对应的跟踪分支。如下图：



**git push remotename --delete branchname 删除远程分支**

注：remotename是必填项

## pull

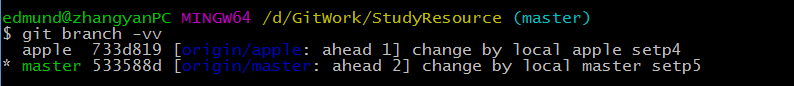
git pull == git fetch + git merge

注：尽量使用git fetch + git merge组合

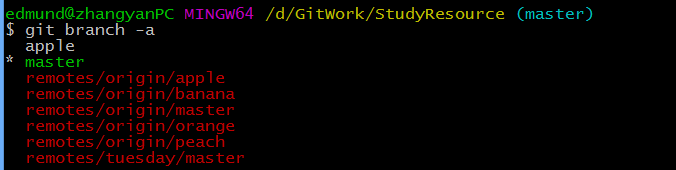
git pull https://github.com/onetimeguy/project 执行一个一次性的抓取，而不会将该 URL 存为远程引用

## 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。如下图：

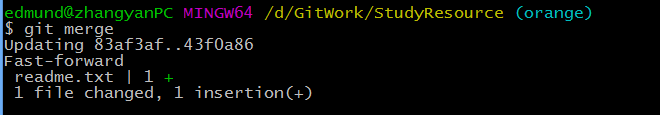


git branch -a 查看所有分支。包括远程分支在本地的远程跟踪分支（分支快照）。如下图：



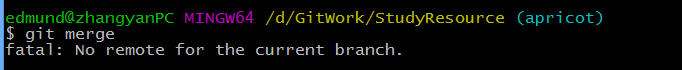
## merge

**git merge branchname 把分支branchname内容合并到当前分支（Head 分支）**

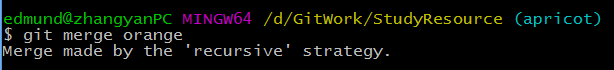


git merge 把当前分支（Head分支）的跟踪分支内容合并进来（如图）。

注：1如果当前分支为纯本地分支，并没有远程分支或跟踪分支，执行git merge会失败（如图）。



2 如果在一个新分支中merge旧分支，会失败。如下图：

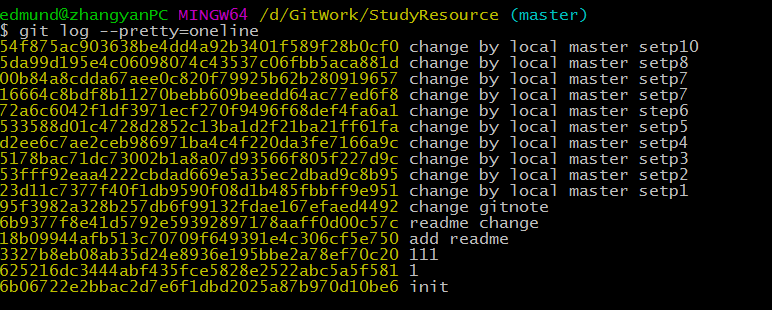


## log

--graph 显示 ASCII 图形表示的分支合并历史。

git log --no-merges issue54..origin/master 只显示所有在后面分支（在本例中是 origin/master）但不在前面分支（在本例中是 issue54）的提交的列表。

git log --pretty=oneline 显示历史提交的校验和 如下图所示：



## tag

git tag -a v1.4 -m 'my version 1.4' 打标签

git show v1.4 显示指定标签

git tag

git tag v1.4-lw 轻量标签

git tag -a v1.2 9fceb02 补打标签

git push origin v1.5 默认情况下，git push 命令并不会传送标签到远程仓库服务器上。 在创建完标签后你必须显式地推送标签到共享服务器上（git push origin --tags可把所有标签推送到远程）

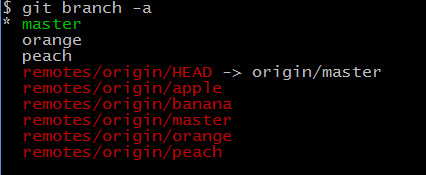
## gitk

## gui

git gui

# 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。

可以使用-u 或 --set-upstream-to 修改本地分支所对应的跟踪分支

