**git push origin master 和 git push 区别：**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

master是主分支，还可以建一些其他的分支用于开发。

**git push origin master的意思就是上传本地当前分支代码到master分支。**

**git push是上传本地所有分支代码到远程对应的分支上。**

**Git 的origin和master分析**

首先要明确一点，对git的操作是围绕3个大的步骤来展开的（其实几乎所有的SCM都是这样）

1. 从git取数据（git clone）

2. 改动代码

3. 将改动传回git（git push）

这3个步骤又涉及到两个repository，一个是remote repository，再远程服务器上，一个是local repository，再自己工作区上。其中

1, 3两个步骤涉及到remote server/remote repository/remote branch，

2涉及到local repository/local branch。git clone 会根据你指定的remote server/repository/branch，拷贝一个副本到你本地，再git push之前，你对所有文件的改动都是在你自己本地的local repository来做的，你的改动(local branch)和remote branch是独立（并行）的。Gitk显示的就是local repository。

在clone完成之后，Git 会自动为你将此远程仓库命名为origin（origin只相当于一个别名，运行git remote –v或者查看.git/config可以看到origin的含义），并下载其中所有的数据，建立一个指向它的master 分支的指针，我们用(远程仓库名)/(分支名) 这样的形式表示远程分支，所以origin/master指向的是一个remote branch（从那个branch我们clone数据到本地），但你无法在本地更改其数据。

同时，Git 会建立一个属于你自己的本地master 分支，它指向的是你刚刚从remote server传到你本地的副本。随着你不断的改动文件，git add, git commit，master的指向会自动移动，你也可以通过merge（fast forward）来移动master的指向。

$git branch -a (to show all the branches git knows about)

\* master

remotes/origin/HEAD -> origin/master

remotes/origin/master

$git branch -r (to show remote branches git knows about)

origin/HEAD -> origin/master

origin/master

可以发现，**master就是local branch，origin/master是remote branch**（master is a branch in the local repository. remotes/origin/master is a branch named master on the remote named origin）

$git diff origin/master master （show me the changes between the remote master branch and my master branch).

需要注意的是，**remotes/origin/master和origin/master的指向是相同的**

$git diff origin/master remotes/origin/master

git push origin master

origin指定了你要push到哪个remote

master其实是一个“refspec”，正常的“refspec”的形式为”+<src>:<dst>”，冒号前表示local branch的名字，冒号后表示remote repository下 branch的名字。注意，如果你省略了<dst>，git就认为你想push到remote repository下和local branch相同名字的branch。听起来有点拗口，再解释下，push是怎么个push法，就是把本地branch指向的commit push到remote repository下的branch，比如

$git push origin master:master (在local repository中找到名字为master的branch，使用它去更新remote repository下名字为master的branch，如果remote repository下不存在名字是master的branch，那么新建一个)

$git push origin master （省略了<dst>，等价于“git push origin master:master”）

$git push origin master:refs/for/mybranch (在local repository中找到名字为master的branch，用他去更新remote repository下面名字为mybranch的branch)

$git push origin HEAD:refs/for/mybranch **（HEAD指向当前工作的branch，master不一定指向当前工作的branch，所以我觉得用HEAD还比master好些）**

$git push origin :mybranch （再origin repository里面查找mybranch，删除它。用一个空的去更新它，就相当于删除了）

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**1. 如何在本地仓库中修改子模块的内容后跟新到远程仓库中 ？**

当本地子模块有跟新时，在主目录下git commit ，然后回到子模块目录下git checkout master 如有必要git merge,然后回到主目录下进行git push，即可把代码跟新到包含子目标下的远程仓库下。

**2. 当远程子模块有更新，如何更新到本地含有的子模块的仓库中 ？**

在子模块上git pull一下实现更新，可以把子模块的更新映射到包含子模块的本地仓库中。

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/