[**git 远程分支创建与推送**](http://www.cnblogs.com/wangkangluo1/archive/2011/09/02/2164313.html)

原文地址：<http://hi.baidu.com/lingzhixu/blog/item/4a9b830bb08a329fe850cd5b.html>

|  |
| --- |
| **本地分支的创建**  本地分支的来源为执行git checkout -b <branch name> 的那个分支  例如现在有两个分支，master和b1  master 分支下有一个commit：  commit1： add test1.c  b1分支下有两个commit：  commit2： add test2.c  commit1： add test1.c  如果在master分支下checkout，即：  git checkout master //进入master分支  git checkout -b frommaster //以master为源创建分支frommaster  git log后发现frommaster中有一个提交：  commit1： add test1.c  这说明他是从master分支为源头创建的  如果在b1分支下checkout，即：  git checkout b1 //进入master分支  git checkout -b fromb1 //以b1为源创建分支fromb1  git log后发现fromb1中已经有两个提交：  commit2： add test2.c  commit1： add test1.c  这说明他是从b1分支为源头创建的  **远程分支的创建**  从远程分支检出的本地分支，称为跟踪分支(tracking branch)。跟踪分支是一种和远程 分支有直接联系的本地分支。在跟踪分支里输入git push，Git 会自行推断应该向哪个服 务器的哪个分支推送数据。反过来，在这些分支里运行git pull 会获取所有远程索引，并  把它们的数据都合并到本地分支中来.  $ git push ssh://git@dev.lemote.com/rt4ls.git master // 把本地仓库提交到远程仓库的master分支中  $ git remote add origin ssh://git@dev.lemote.com/rt4ls.git $ git push origin master  这两个操作是等价的，第二个操作的第一行的意思是添加一个标记，让origin指向ssh://git@dev.lemote.com /rt4ls.git，也就是说你操 作origin的时候，实际上就是在操作ssh://git@dev.lemote.com/rt4ls.git。origin在这里完全可以理解为后者 的别名。  **需要说明的是，默认情况下这条语句等价于提交本地的master仓库到远程仓库，并作为远程的master分支。**  我从master分支创建了一个issue5560分支，做了一些修改后，使用git push origin master提交，但是显示的结果却是'Everything up-to-date'，发生问题的原因是git push origin master 在没有track远程分支的本地分支中默认提交的master分支，因为master分支默认指向了origin master 分支，这里要使用git push origin issue5560：master 就可以把issue5560推送到远程的master分支了。      如果想把本地的某个分支test提交到远程仓库，并作为远程仓库的master分支，或者作为另外一个名叫test的分支，那么可以这么做。  $ git push origin test:master         // 提交本地test分支作为远程的master分支 $ git push origin test:test              // 提交本地test分支作为远程的test分支  如果想删除远程的分支呢？类似于上面，如果:左边的分支为空，那么将删除:右边的远程的分支。  $ git push origin :test              // 刚提交到远程的test将被删除，但是本地还会保存的，不用担心 |