# Git操作部署命令

**本地新建git分支并同步到远程同名分支**

* makdir ~/hello-world //创建一个项目hello-world
* cd~/hello-world //打开这个项目
* git init //初始化本地git仓库
* git clone [https://github.com/lovepli/git\_demo.git //](https://github.com/lovepli/git_demo.git%20//)克隆项目
* git remote add origin https://github.com/lovepli/git\_demo.git //连接远程github项目
* git branch //查看所有分支名

git branch -a //查看远程分支情况

git branch -r //查看远程分支和本地分支情况

* git fetch //如果没有看到你想要的分支,先获取所有分支
* git checkout origin/master //切换到远程master分支
* git pull //错误的说明
* git checkout -b dev //新建dev分支
* git branch -u origin/master dev //建立本地分支和远程分支的追踪关系
* git pull //验证正确
* 本地切换分支到新建分支：git checkout dev
* 新建分支同步远程分支（如果远程没有则自动创建）：git push --set-upstream origin dev
* **补充说明**：一般我们就用git push --set-upstream origin branch\_name来在远程创建一个与本地branch\_name同名的分支并跟踪；利用git checkout --track origin/branch\_name来在本地创建一个与branch\_name同名分支跟踪远程分支。

**提交代码到远程仓库的指定分支**

* git add . //提交所有文件到本地仓库
* touch README //新建readme文件
* git add README //提交readme文件到仓库
* git commit -m "提交说明"。//提交代码说明
* git remote add origin https://github.com/lovepli/git\_demo.git //连接远程github项目
* git push origin master //将本地代码推送到远程master分支
* 如果在github的remote上已经有了文件，会出现错误。此时应当先pull拉取一下，即：
* git pull origin master
* 然后再进行推送 git push origin master

**补充的git命令**

* git status //显示哪些项目发生了变化

new file //表示新建文件  
modified //表示修改文件

deleted //表示删除文件

* git merge +分支名 // merge功能是将一些分支的内容合并到某一个特定的分
* git branch -D yourbranch //删除分支
* git push origin --delete yourbranch //删除远程分支
* git branch -D Su-modify //强行删除本地分支