

# Git

## 1.Pull Request 流程

### 1.fork



如上图所示找到想要pull request的项目，然后点击fork按钮，此时变会在你的仓库中多出来一个仓库，名字为：自己的账户名/想要pull request的项目的名称

### 2.clone

通过运行命令：`git clone https://github.com/fengyanWang/first-pr.git`（后面的网址为你自己账户下刚刚生成的那个新的仓库的地址），将该仓库克隆到当前的开发环境中

### 3.branch

#### 1.确认分支

通过在终端运行命令：`git branch -a`查看当前所在的分支，通常我们都是查看分支后再进行代码的修改，这是一个好的习惯

```
(python27) wfy@wfy-ThinkCentre-M910t-N000:~/myStudy/githubStudy/Documents/myoData$ git branch -a
master
* work
remotes/origin/HEAD -> origin/master
remotes/origin/master
remotes/origin/work
```

<https://blog.csdn.net/wangzi111111111>

#### 2.创建特性分支

- 在应用github修改代码时，我们常常采用的策略是在主分支下在创建一个特性分支
- 在该特性分支下进行代码的修改然后通过该分支执行pull request操作
- 通过命令：`git checkout -b work master`(其中work为你新建的特性分支，master为你当前所在的分支)创建新的特性分支并自动切换

## 4.添加修改

### 1.修改工程

在刚刚新创建的分支下对fork下的工程进行修改

### 2.提交修改

首先使用命令：`git diff`查看修改的内容是否正确，然后使用命令：`git add readMe.md`（其中readMe.md为要添加的文件）向仓库中添加文件，在执行命令：`git commit -m "add readMe.md文件"`提交说明

### 3.创建远程分支

- 要从github发送pull request，github端的仓库中必须有一个包含了修改后的代码的分支，所以需要创建一个与刚刚创建的特性分支（修改所在的分支）相对应的远程分支

- 执行命令：**git push origin work1**(其中origin为当时fork的远程主分支的名称，一般默认为origin,work1为本地工作的特性分支)，然后执行：**git branch -a**进行查看是否创建成功

## 5.发送pull request

- 进入到自己的github账户下，并切换到创建的特性分支下，然后点击**create pull request**后，确定没问题，填写相关内容，然后点击**send pull request**

## 2.基于GIT做协同开发