# 1，先拷贝项目，然后建立分支，自动创建分支

## git checkout -b zhangyj

# 2.查看分支

## git branch，切换到分支 git checkout zhangyj

# 3.将自己改动的提交到本地

## git status

## git add –A

## git commit -m"add index.html"

# 4.拉下系统的项目（develop为原始项目分支的名字）

## git pull origin develop

# 5.如果是提交 (zhangyj为我们自己设置的分支的名字)

## git push origin zhangyj

# 6.打开git网址



## 1.因为我们是使用的develop分支，默认提交的分支是master，所以这里要进行改变









# 7.完成

# 8.出错总结

## 1.如果自己的代码完全错误，跑步起来，到代码齐全的分支上重新

### git clone -b develop <http://git.minixiao.com/wangyj/cn.edu.dlut.career.git>

## 2.如果自己的分支上传后，网络端代码错误，则删除自己的分支

### git branch -D zhangzyj 删除分支

## 3、 如果提示，还有进程在上传，目前的进程上传不了，

Git操作的过程中突然显示Another git process semms to be running in this repository, e.g. an editor opened by ‘git commit’. Please make sure all processes are terminated then try again. If it still fails, a git process remove the file manually to continue…   
翻译过来就是git被另外一个程序占用，重启机器也不能够解决。

原因在于Git在使用过程中遭遇了奔溃，部分被上锁资源没有被释放导致的。

解决方案：进入项目文件夹下的 .git文件中（需要显示隐藏文件夹）删除index.lock文件即可。

# 9.回退版本

## 1.回退到上一个版本

### git reset --hard head^ 或

### git reset --hard head~x

## (1) 回退所有内容到上一个版本

### git reset HEAD^

## (2) 回退a.py这个文件的版本到上一个版本

### git reset HEAD^ a.py

## (3) 向前回退到第3个版本

### git reset –soft HEAD~3

## (4) 将本地的状态回退到和远程的一样

### git reset –hard origin/master

## (5) 回退到某个版本

### git reset 057d

# 10、删除项目里面所有.svn和.git 文件

转载 2016年08月05日 09:40:19

* 2406

# 项目代码上传至svn/git后会产生.svn/.git文件，项目打包时需要将这些文件删除。

打开终端，cd到项目文件夹

**[plain]** [view plain](http://blog.csdn.net/xo764834598/article/details/52126045) [copy](http://blog.csdn.net/xo764834598/article/details/52126045)

1. //删除文件夹下的所有 .svn 文件
3. find . -name ".svn"| xargs rm -Rf
5. //删除文件夹下的所有 .git 文件
7. find . -name ".git"| xargs rm -Rf

不如直接删除.git文件给力,啥也别说了

# 11、特殊重要的

## 有时候，如果我们上传的文件太大，失败了，但是又已经commit了，很难受，又不能撤回，有时候又不知道到底是哪个文件太大了。所以用到了撤回到上次commit的状态。也就是说本地的修改还会通过git status显示出来

你的意思是用git commit提交了一个更改，但是你又不需要这个更改了，想退回到上一个comiit ID上去，但是又想把你更改的代码保留吗？可以用git reset --soft commit-id （，然后敲git status看看，根据说明可以用相应的命令了

### git log 查看comit 的id

## git reset --soft fadsfadsfadsf123 (你运行gitcommit 之前的那个最新commit ID)

## git status 就可以看到我们本地刚刚做的修改了哦 哈哈哈，这个才是真正牛逼的东西