1. 将远程库克隆到本地   
   [**Git**](http://lib.csdn.net/base/git) clone https://github.com/gonghuiyun/gonghuiyun.github.io.git   
   2.查看远程库的详细信息   
   git remote show origin   
   3.进入clone下来的文件夹：如我的是gonghuiyun.github.io，执行git init   
   4.更改文件夹里面的内容   
   5.提交做出的更改：增加或是删除文件，要加选项-A，A后面的.代表当前文件夹（如gonghuiyun.github.io）的所有文件   
   git add -A .   
   6.观察状态   
   git status   
   7.正式提交到git远程库（-m 表示注释内容）   
   git commit -m "项目初始化"   
   8.推送到远程库   
   git push   
   注：若想从远程库取回数据，则执行git pull

**第一次提交** :

**方案一** : **本地创建项目根目录, 然后与远程GitHub关联, 之后的操作一样**;

-- **初始化[Git](http://lib.csdn.net/base/28" \o "Git知识库" \t "_blank)仓库** :**[Git](http://lib.csdn.net/base/git" \o "Git知识库" \t "_blank) init** ;

-- **提交改变到缓存** :**git commit -m 'description'** ;

-- **本地git仓库关联GitHub仓库** : **git remote add origin git@github.com:han1202012/TabHost\_Test.git** ;

-- **提交到GitHub中** : **git push -u origin master** ;

**方案二**: **方案二就是不用关联GitHub仓库, 直接从GitHub冲克隆源码到本地, 项目根目录也不用创建**;

-- **从GitHub上克隆项目到本地** :**git clone git@github.com:han1202012/NDKHelloworld.git** , 注意克隆的时候直接在仓库根目录即可, 不用再创建项目根目录 ;

-- **添加文件** :**git add ./\*** , 将目录中所有文件添加;

-- **提交缓存** :**git commit -m '提交'**;

-- **提交到远程GitHub仓库** : **git push -u origin master** ;

**之后修改提交** :

-- **与GitHub远程仓库同步** :**git pull** ;

-- **查看文件变更** : **git status** ;

-- **提交代码到本地缓存** : **git commit -m 'description'**;

--**提交代码到远程GitHub仓库** :**git push** ;

**.gitignore用法** : 开放模式 注明忽略的文件 直接列出文件名, 保守模式 注明保留的文件 !文件名 ;

**Git标签操作** : 轻量级标签, 带注释标签;

--**查看标签** :**git tag** ;

--**添加标签** : 轻量级标签**git tag tagName** , 带注释标签**git tag -a tagName -m 'description'** ;

--**删除标签** :**git tag -d tagName** ;

--**提交标签到GitHub中** : **git push origin --tags** ;

**Git分支操作**: 创建分支后, 分支操作不会影响master分支, 但是master分支改变会影其它分支;

--**列出分支** :**git branch** ;

--**切换分支** :**git checkout master** ;

--**提交分支** : **git push origin branchName** ;

--**删除分支** : **git branch -d branchName** , 强制删除分支 **git branch -D branchName** ;

--**合并分支** : **git merge branchName** ;

**常用命令：**

廖雪峰的网站

本地向远程库提交

$ git remote add origin git@github.com:michaelliao/learngit.git

$ git push -u origin master

要关联一个远程库，使用命令git remote add origin git@server-name:path/repo-name.git；

关联后，使用命令git push -u origin master第一次推送master分支的所有内容；

此后，每次本地提交后，只要有必要，就可以使用命令git push origin master推送最新修改；

从远程库克隆

$ git clone git@github.com:michaelliao/gitskills.git

**如何解决failed to push some refs to git(第一次往远程仓库发送文档时，要先把远程仓库里**的README.md文件拷回本目录**)**

git pull --rebase origin master

<http://jingyan.baidu.com/article/f3e34a12a25bc8f5ea65354a.html>

**本地库关联github遇到的问题(关联错误)**

<http://www.jianshu.com/p/5dfa49a97ce8>

意思是说，已经关联了远程库，再次关联就失败了！  
后来在网上看到说，先使用git remote rm origin,在进行关联远程库，果然这样就可以了！下面简单写一下步骤：

1.将关联错误的远程库移除  
$ git remote rm origin  
2.重新关联远程库  
$ git remote add origin git@github.com:watermin/DesDemo.git  
3.将本地库代码push到远程库

[**push到github时，每次都要输入用户名和密码的问题**](http://blog.csdn.net/gatieme/article/details/45033349)

原因是使用了https方式 push   
这种方式产生的原因，一般是我们clone是一定是使用了http的方式   
例如我们在github上创建了一个项目，然后我们clone到本地时使用了http而非[**Git**](http://lib.csdn.net/base/git)

我们可以看到右端的项目地址，默认是http的，clone时

git clone https://github.com/gatieme/AderSCloud.git

* 1
* 1

如果使用ssh的话，应该使用的路径是

git clone git://github.com/gatieme/AderSCloud.git

* 1
* 1

解决

在termail里边 输入

git remote -v

* 1
* 1

可以看到形如一下的返回结果

origin git@github.com:gatieme/AderSCloud.git (fetch)

origin git@github.com:gatieme/AderSCloud.git (push)

* 1
* 2
* 1
* 2

下面把它换成ssh方式的。   
移出旧的http的origin

git remote rm origin

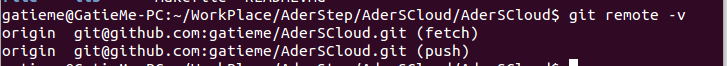
* 1
* 1

添加新的git方式的origin   
git remote add origin git@github.com:gatieme/AderSCloud.git

我们在查看一下push方式

git remote -v

* 1
* 1



再次push一下，已经不需要输入用户名密码了

git push origin

* 1
* 2
* 1
* 2

这里写图片描述