1、安装git

git config --global user.name 'XXX'

git config --global user.email 'XXX'

2、创建本地库

mkidir learngit //自定义文件夹  
cd learngit

touch test.md //创建test.md文件  
pwd //显示当前目录

3、常用CRT

git init //初始化代码仓库  
git add learngit.txt                               //把所有要提交的文件修改放到暂存区  
git commit -m 'add a file'                      //把暂存区的所有内容提交到当前分支  
git status                                            //查看工作区状态  
git diff                                                //查看文件修改内容  
git log                                                //查看提交历史  
git log --pretty=oneline                       //单行显示  
git reset --hard HEAD^　　　　　　　　 //回退到上一个版本，其中（HEAD^^(上上版本),HEAD~100(往上100个版本)）  
  
commit id                                          //(版本号) 可回到指定版本  
git reflog                                           //查看历史命令

其中说明【  
工作区（Working Directory）  
版本库（Repository） #.git  
stage(index) 暂存区  
master Git自动创建的分支  
HEAD 指针

】

git diff HEAD -- <file>                                  //查看工作区和版本库里最新版本的区别  
git checkout -- <file>                                   //用版本库的版本替换工作区的版本，无论是工作区的修改还是删除，都可以'一键还原'  
git reset HEAD <file>                                   //把暂存区的修改撤销掉，重新放回工作区。  
git rm <file>                                               //删除文件，若文件已提交到版本库，不用担心误删，但是只能恢复文件到最新版本

4、创建SSH Key，建立本地Git仓库和GitHub仓库之间的传输的秘钥

ssh-keygen -t rsa -C 'your email'                                                    //创建SSH Key  
git remote add origin git@github.com:username/repostery.git          //关联本地仓库，远程库的名字为origin  
//第一次把当前分支master推送到远程，-u参数不但推送，而且将本地的分支和远程的分支关联起来  
git push -u origin master   
git push origin master                                                                  //把当前分支master推送到远程  
git clone git@github.com:username/repostery.git                            //从远程库克隆一个到本地库

5、分支  
git checkout -b dev                                   //创建并切换分支  
#相当于git branch dev 和git checkout dev   
git branch                                                //查看当前分支，当前分支前有个\*号  
git branch <name>                                   //创建分支  
git checkout <name>                                //切换分支  
git merge <name>                                   //合并某个分支到当前分支  
git branch -d <name>                               //删除分支  
git log --graph                                          //查看分支合并图  
git merge --no-ff -m 'message' dev            //禁用Fast forward合并dev分支

git stash                                                 //隐藏当前工作现场，等恢复后继续工作  
git stash list                                            //查看stash记录  
git stash apply                                         //仅恢复现场，不删除stash内容  
git stash drop                                          //删除stash内容  
git stash pop                                           //恢复现场的同时删除stash内容  
git branch -D <name>                              //强行删除某个未合并的分支

//开发新feature最好新建一个分支  
git remote                                               //查看远程仓库  
git remote -v                                           //查看远程库详细信息

git pull                                                   //抓取远程提交  
git checkout -b branch-name origin/branch-name                  //在本地创建和远程分支对应的分支  
git branch --set-upstream branch-name origin/branch-name   //建立本地分支和远程分支的关联

6、其他---标签  
git tag v1.0                                                                      //给当前分支最新的commit打标签  
git tag -a v0.1 -m 'version 0.1 released' 3628164                 //-a指定标签名，-m指定说明文字  
git tag -s <tagname> -m 'blabla'                                        //可以用PGP签名标签  
git tag                                                                             //查看所有标签  
git show v1.0                                                                   //查看标签信息  
git tag -d v0.1                                                                 //删除标签  
git push origin <tagname>                                               //推送某个标签到远程  
git push origin --tags                                                       //推送所有尚未推送的本地标签