Git操作

git config --global 所有git仓库都会使用这个配置

git init 在一个路径下使用，将这个路径初始化为git仓库

git add file 将文件放进暂存区（文件要存在）

git commit -m “提交注释” 将文件提交到仓库

git status 检查是否还有文件未提交

git diff file 查看文件的修改

add只能一个一个提交, commit可以一次性提交全部

git log 提交日志

git log --pretty=oneline 简化显示日志

git reset --hard HEAD^ 回退版本

git reset --hard HEAD^^ 回退两个版本

git reset --hard HEAD~100 回退多个版本

cat file 查看文件内容

git reset --hard 版本号 回退到任意版本（可以是新的版本）

git reflog 获取版本号

git checkout -- file 丢弃工作区的修改（包括删除文件）

rm file 删除文件

git push origin master 将本地分支的最新修改推送到github

git clone 地址 将远程库内容克隆到本地库

git checkout -b 分支名 创建并切换分支

git branch 查看所有分支

git branch 分支名 创建分支

git checkout 分支名 切换分支

git merge 分支名 合并某分支到当前分支

git branch –d 分支名 删除分支

不同分支对应着不同的版本，在某一分支下做出的修改，不会影响其他分支

首先master主分支应该是非常稳定的，一般情况下不允许在上面干活，干活一般情况下在新建的dev分支上干活，干完后，比如上要发布，或者说dev分支代码稳定后可以合并到主分支master上来

git一般使用”Fast forward”模式，在这种模式下，删除分支后，会丢掉分支信息

git merge –no-ff (-m “注释”) 分支名 禁用Fast forward模式

git stash 暂存工作现场

git stash list 暂存工作现场列表

git stash drop 删除第0条现场

git stash apply 恢复现场（不删除stash）

git stash pop 恢复现场（删除stash）

git remote 查看远程库的信息

git remote –v 查看远程库的详细信息

远程库的默认名称是origin

git push origin 分支名 将该分支推送到远程库对应的远程分支

master分支是主分支，因此要时刻与远程同步，一些修复bug分支不需要推送到远程去，可以先合并到主分支上，然后把主分支master推送到远程去

git pull 从远程获取最新版本并merge到本地

多人协作工作模式：

首先，可以试图用git push origin 分支名 推送自己的修改.

如果推送失败，则因为远程分支比你的本地更新早，需要先用git pull试图合并

如果合并有冲突，则需要解决冲突，并在本地提交。再用git push origin 分支名推送

开发主要有5个分支

1、主分支Master：只发布提供给用户的正式版本

2、开发分支Develop：开发所用的分支，开发完成后合并到主分支上

3、功能分支feature：从Develop分出，用于开发某种功能

4、预发布分支release：用于Develop合并到Master之前，进行预发布版本的测试

5、修复bug分支fixbug：从Master分支分出，用于修复bug，要合并回Dev和Master

后三个属于临时分支，用完后要及时删除