一、基础使用操作

1. 将本地目录导入到git库
2. 建立自己的分支，git switch -c branchname
3. 做完修改后，提交修改
   1. git add .
   2. git status 查看当前暂存区状态
   3. git commit -m 修改描述
   4. git status 查看当前工作区状态
   5. d中显示工作区为空，则说明提交成功
4. 切换至master分支git switch master
5. 合并自己分支的操作 git merge branchname(如果出现冲突，说明有其他用户修改了你正在工作的文件，此时需要手动解决冲突后，再进行merge)
6. 合并完成后删除自己的分支 git branch -d brandname
7. 将本次修改上传至远程库 git push -u origin master
8. 在每次工作前，使用git pull命令，将自己的本地更新至最新版本。

二、常用分支

1、bug分支：

当手头事情没有做完又还不想提交，而又有新的东西需要处理，此时的操作为

1. 保存自己现在分支的工作 git stash
2. git status查看当前工作区状态，显示为空则正常
3. git stash list查看是否正常备份
4. 创建新的分支进行新东西处理
5. 处理完提交
6. 切换至master分支，进行合并
7. 删除新建的分支
8. 还原自己的原先的工作区内容 git stash pop
9. 继续自己的工作

三、常用命令：

查询git状态 git status

查看差异 git diff

版本查看 git log

版本回滚 git reset --hard xxxxxxx (xxxxxxx为版本号)

查看之前的命令 git reflog

查看文件内容 cat xxxxx(xxxxx为文件名)

查看工作区和版本库里面最新版本的区别git diff HEAD -- xxxxx (xxxxx为文件名)

撤销操作：

场景1：当你改乱了工作区某个文件的内容，想直接丢弃工作区的修改时，用命令git checkout -- file。

场景2：当你不但改乱了工作区某个文件的内容，还添加到了暂存区时，想丢弃修改，分两步，第一步用命令git reset HEAD <file>，就回到了场景1，第二步按场景1操作。

场景3：已经提交了不合适的修改到版本库时，想要撤销本次提交，参考版本回退一节，不过前提是没有推送到远程库。

链接远程：git clone xxxxxxxxx(xxxxxxxxx为远程库地址)

git add xxxx(xxxx为修改的文件或者文件夹)

git commit -m 'xxxx'(对此次操作的说明)

git push -u origin master(将本次操作上传至远程库)

创建分支

git checkout -b xxx(xxx为分支名)

查看分支：git branch

创建分支：git branch <name>

切换分支：git checkout <name>或者git switch <name>

创建+切换分支：git checkout -b <name>或者git switch -c <name>

合并某分支到当前分支：git merge <name>

删除分支：git branch -d <name>

快速合并会产生冲突情况：

每次进行分支操作后，需要切换至master分支后再进行合并。

当Git无法自动合并分支时，就必须首先解决冲突。解决冲突后，再提交，合并完成。

解决冲突就是把Git合并失败的文件手动编辑为我们希望的内容，再提交。

用git log --graph命令可以看到分支合并图。

普通合并方式：

普通方式合并分支 git merge --no-ff -m "desc" branchName (desc为合并描述，branchName为分支名称)

从远程库拉取最新版本：git pull