工作区----添加---🡪暂存区----提交---🡪本地仓库----推送---🡪远程仓库

**总体上的步骤**

**第一步** 项目文件夹之中

**git init** 初始化 表示使用git管理

**第二步**记录代码到暂存区

**git add . / git add 指定文件 工作区进入暂存区**

**第三步 提交代码到本地仓库**

**git commit -m “此次记录的说明” 暂存区进入本地仓库**

**git commit -a -m “注释” 工作区直接进入暂存和本地仓库，执行这个命令 必须第一次要执行过 git add . git commit -m “此次记录的说明” 这两个命令**

**第四步本地仓库推送到远程仓库**

1. **建立连接**

git remote add 远程仓库地址别名（自定义） 远程仓库地址

1. **先拉取后推送**

**git pull**

1. **首次推送**

git push -u 远程仓库地址别名 本地分支:远程分支`

如果本地分支名和远程分支名一样，则可以省略 `:远程分支

**3.后续的每一次推送**

**git push**

**多人合作 协作开发的时候**

**仓库拥有着**

邀请

**再次编写代码 先拉取在推送**

**合作者**

1. **先克隆**

git clone 远程仓库地址` --> 把远程的文件拷贝到本地

1. **编写好的代码先提交到本地仓库**
2. **拉取**

**git pull**

1. **推送**

git push

**----------------知识点--------------------------**

初次提交代码会要求输入账号邮箱、

**git config --global user.name "xxx"**

**git config --global user.email "xxx"**

**恢复代码也就是撤回**

暂存区-🡪工作区

git checkout . **文件名没有引号**

仓库区—>暂存区

git reset.

仓库区—>工作区和暂存区

git reset –hard

**当你在想要恢复到某一天的代码的时候**

1. **查看版本号**

git log或者git log –oneline

1. **切换**

- git checkout 版本号`

**- `git checkout master` --> 到达最后的版本**

**分支：主分支名字 master 不变的**

**创建分支**： **将原来的代码复制一份的意思**

`git branch 分支名`（自定义）

**切换分支**

git checkout 分支名`

`git checkout -b 分支名` --> 创建并直接切换分支

**合并分支：先切换到主分支**

**切换到主分支 ：git checkout master**

`git merge 分支名` （注意先切换分支，然后在合并）将新增改变的东西合并

**删除分支**

切换到主分支

`git branch -d 分支名`

**查看所有的分支**

`git branch`

----------------解决冲突+----------------------

