Github（远程库）

Pull（fetch+merge），管理员将更改后的文件拉取到本地

Push（自己申请加入管理员团队后，可以将更改后的内容上传到github中）

Clone（将文本克隆到本地）

Push（将自己的文本上传到github中）

成员

管理员

Github：代码托管中心

指令：

pwd 命令是用于显示当前的目录

ll命令使用于显示包含文件的total数量

ls查看当前目录下有什么文件

cat xxx 参看xxx文件的内容

cd ..返回上一级目录

cd ../../返回上上一级目录

cd ~ 命令用于回到home目录

vim 命令进入vim编辑器编辑，完成后按esc，再输入 （ ：wq）保存退出

git init 将这个目录变成git可以管理的仓库

git config user.name xxx设置（检查）本地用户名

git config --global user.name xxx 设置（检查）全局本地用户名

git config user.email xxx 设置（检查）本地邮箱

git status 查看项目/工作区/暂存区状态。

状态分为3种:1、Changes to be committed：表示已经从工作区add到暂存区的file（文件或文件夹），可以通过 git restore --staged filename 命令将该file从暂存区移出

2、Changes not staged for commit:表示工作区，暂时区都存在的file（文件或文件夹），在工作区进行修改或删除，但是没有add到暂存区

3、Untracked files:表示只在工作区有的file（文件或文件夹），也就是在暂时区没有该file

当git add 成功的话，git status 会出现Changes to be committed:  
(use "git restore --staged <file>..." to unstage)

git commit将暂存区修改的内容提交到本地仓库

git commit –m “XX” –m 后面的是注释

git add[file name] 将文件从工作区添加到暂存区

git log 查看历史记录。多屏显示的控制方式：空格向下翻页、b向上翻页、q退出

git log --pretty=online/git log --online使历史记录显示更简洁

git reflog 获取所有版本号

git reset --hard 6fcfc89 返回到指定的6fcfc89版本（这个最好用）

git reset --hard HEAD^ 返回上一个版本

git reset --hard HEAD^^返回上上一个版本

git reset --hard HEAD~20 返回前20个版本

删除文件并找回：git reset --hard[指针位置]。使用前提：删除前，文件存在时的状态提交到了本地库

git diff[文件名]将工作区中的文件和暂存区进行比较

git branch -v查看分支

git branch xxx创建一个新分支xxx

git checkout xxx切换到xxx分支

git remote 查看远程库信息

git remote -v 查看远程库详细信息

git remote add origin <https://github.com/mengmengxiong0/huashan.git关联一个远程库>

git remote remove origin 删除一个关联的远程库

git push origin master[分支名] 将本地该分支上的文件推送到Github远程库上

git push -f强制推送

git clone [远程库文件地址] 将github远程库中文件克隆到本地（本地不需要是仓库）

git pull origin master[分支名] 将github远程库中文件拉取到本地某个分支

git fetch origin master[分支名] 将远程仓库内容下载到本地，注意git fetch不会改变本地的文件（只是下载下来了而已）

git merge origin/master[远程库的分支名] 将远程库的文件下载下来

git 分支1 rebase 分支2 将分支1合并到分支2

Github仓库中邀请成员和查看成员都在manage access中

git branch 失败的原因（Not a valid object name: 'master'.）：原因是没有提交一个对象，需要commit后才会真正建立一个maser分支，之后才可以branch新的分支。

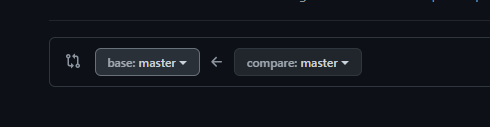
分支冲突：

多人冲突：如果不是局域github远程库的最新版所做的修改，是不能推送的，必须先拉取到最新版。拉取下来后如果进入冲突状态，则按照分支冲突的解决办法操作

git pull和git clone的区别：1pull必须为仓库（git init），clone不要是仓库；2分支问题：clone可以随意切换远程仓库中存在的分支，而push只能在当前分支；3推送问题：clone可直接跟地址名（如git push 地址名） ，而pull需要用origin来指定源再push，如（git remote add origin 地址名，再Git push origin 分支名）

以下示例主分支为main（git init后的主分支默认为main），远程库的default为master

跨团队协作（不是这个仓库的开发者或者管理员）：1（其他人点击）Fork（会有等待页面加载） 2（其他人操作）本地克隆、修改，然后推送到远程 3（其他人点击）Pull request选项中的 New pull request ，然后再点击Create pull request 4（原来的管理者、成员）在Pull request（1）选项中点击内容进去后，把“is:pr is:open ”去掉回车，再点击Merge pull request合并代码 ，再点击Comfirm merge确认合并

仓库管理员将其它分支内容合并到主分支：点击pull request，然后点击new pull request，然后会看到如图，箭头左边是要被合并的到的主分支，右边是要合并的其他分支。此时点击右边compare的小箭头会出现其他分支名，选择要合并上去的分支，再点击Merge pull request合并代码 ，再点击Comfirm merge确认合并。

开发者另外创建一个分支写代码（一般git init后的分支为maser或者main）：1先自己创建一个新的分支git branch haode；2再切换到创建好的分支git checkout haode（此时已经在haode分支上）；3写完代码后git add . (中间可以用git status查看是否add 到暂存区成功)，再git commit -m””，再git push 路径名 分支名。4再回到github上的仓库就会返现新增了一个haode分支（可以看到内容）

开发者将远程仓库中的内容和本地分支（主分支为main，自己另创的分支为haode）中的内容进行合并 ：1先将分支切换到主分支 git checkout main ；2再把远程仓库中的内容拉到自己的仓库分支中（这里拉到main分支，可以除haode其他的分支）git pull origin master（将远程库的master的内容pull到当前分支main）；3再将自己原来haode分支合并到的main分支中git merge haode。此时本地文件夹的内容就会和远程库的内容一样