账号：605857324@qq.com密码:1024zhoumin123

**1.初始化本地git**

**git init**

**2.全局设置用户名，用户邮箱**

**git config --global user.name "MinZ2" 注：一定是自己github上的用户名**

**git config --global user.email "605857324@qq.com"**

**3.查看当前用户名，当前用户邮箱（可略）**

**git config user.name**

**git config user.email**

**4.连接远程仓库**

**git remote add origin https://github.com/MinZ2/test1.git**

**5.查看/删除远程连接仓库（可略）**

**远程仓库命名：git remote**

**远程仓库地址：git remote -v**

**删除仓库：git remote rm origin**

**显示仓库的详细信息： git remote show origin**

**6.将本地仓库文件push到远程仓库**

**git status 查看工作目录的状态===红色的文字（新建文件）**

**git add . 将文件添加到暂存区===绿色文字（上传到缓存区）**

**git commit -m “备注” 提交修改,添加备注信息（本地仓库）**

**git push origin master 将版本库分支push到远程仓库**

**删除远程文件**

**（1）对需要删除的文件、文件夹进行如下操作:===(绿色文字)**

**git rm test.txt (删除文件)**

**git rm -r test (删除文件夹)**

**（2）提交修改**

**git commit -m "Delete some files."**

**（3）将修改提交到远程仓库的xxx分支:**

**git push origin master**

**7.新建/切换/查看/删除分支**

**！！！新建的git仓库只有master一个分支，当第一次提交东西后才能查到分支名称**

**新建本地分支：git branch sit**

**切换分支：git checkout sit（切换完分支后，工作区的文件会不一样）**

**把新建的本地分支同步到远程分支去：git push origin sit:sit（远程没有这个分支）**

**查看当前本地分支并且定位当前分支：git branch（当前分支颜色会变绿色）**

**查看远程分支：git branch -r**

**查看本地和远程所有分支：git branch -a（远程分支是红色的）**

**！！！！在远程仓库新建完一个分支，个人git命令上查询不到，必须先要git pull一下git branch -r才能查，当切此分支后，git branch才能查到**

**删除本地分支:git branch -D test**

**！！！如果你本地test分支删除了（远程有），切换此test分支，git会自动设置test分支并且跟踪远程test分支**

**【下面慎用】**

**# 遍历本地仓库中的所有分支，如果该分支在远程仓库中不存在，则在远程仓库中创建该分支（本地有远程没有，远程创建）**

**# 遍历远程仓库中的所有分支，如果该分支在本地仓库中不存在，则在远程仓库中删除该分支（本地没有远程有，远程删除）**

**git push --all --prune**

**8.编码设置**

**$ git config --global core.quotepath false   显示 status 编码**

**$ git config --global gui.encoding utf-8  图形界面编码**

**$ git config --global i18n.commit.encoding utf-8  提交命令的时候使用utf-8编码集提交**

**$ git config --global i18n.logoutputencoding utf-8  输出 log 编码**

**$ export LESSCHARSET=utf-8 设置LESS字符集为utf-8**

**9.远程强制覆盖本地**

**从远程仓库下载最新版本：git fetch --all**

**将本地设为刚获取的最新内容：git reset --hard origin/master**

**10.查看提交历史（更多用法：https://git-scm.com/docs/git-log）**

**（1）git log**

**（2）历史记录的简洁版本：git log --oneline**

**（3）逆向时间显示日志：git log --reverse**

**（4）查看指定用户的提交日志： git log --author=MinZ2【 --oneline -5】后面接的【】表示显示多少条记录**

**（5）指定日期，Git 项目中三周前且在四月十八日之后的所有提交，我可以执行这个（我还用了 --no-merges 选项以隐藏合并提交）：**

**git log --oneline --before={3.weeks.ago} --after={2010-04-18} --no-merges**

**11.过滤文件**

**创建过滤配置文件命令：touch .gitignore ！！！名字不要写错了**

**/abc.txt #忽略此目录下abc.txt文件**

**\*.a #忽略所有.a结尾的文件**

**!lib.a #但lib.a除外**

**todo/ #忽略todo/目录下的所有文件**

**doc/\*.txt #会忽略doc/a.txt 但不包括doc/a/b/b.text**

**【问题集】：**

**问题：git切换分支或者add或者commit时出现[fatal: Unable to create 'H:/技术/Git使用/.git/index.lock': File exists.]**

**解决方案：在.git同级目录，执行[rm -f .git/index.lock],将文件删除即可成功。**

**问题：git上字体颜色代表什么**

**回答：**

**绿色，已经加入控制暂未提交**

**红色，未加入版本控制**

**蓝色，加入，已提交，有改动**

**白色，加入，已提交，无改动**

**灰色：版本控制已忽略文件。**

**问题：出现[Already up to date.]**

**回答：**

**①当前分支的文件和远程仓库的文件不统一(解决：远程仓库同步到本地)**

**②本地仓库有新增的文件（解决：add和commit）**

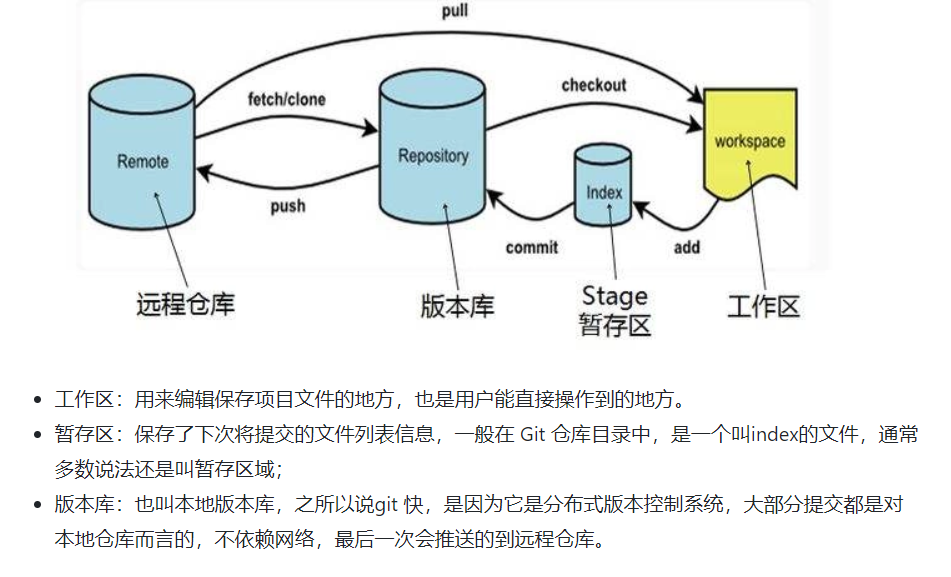
**工作区（本地）：在电脑里面能看到的目录**

**暂存区(缓存区)：一般存放在 ".git目录下" 下的index文件（.git/index）中，所以我们把暂存区有时也叫作索引（index）。Git为我们自动创建的第一个分支master，以及指向master的一个指针叫HEAD。**

**版本库：工作区有一个隐藏目录.git，这个不算工作区，而是Git的版本库。**

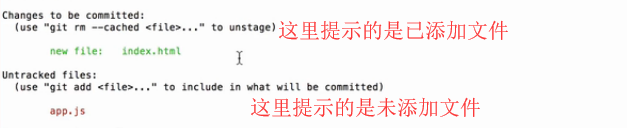
**工作区——add——暂存区——commit——版本库——push——远程仓库**



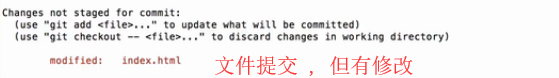


## git status

(查看状态 , 也能查看到添加的文件)



当文件提交后 , 文件再次被修改; 通过查询状态 可知



modified 提示 index.html 有变动 , git本地仓库需要同步

.

1. 永久保存账号密码，免去git重复输入账号密码操作

**方法一**：安装好git之后一般会在C盘的C:\Users\Administator目录下生成 .gitconfig配置文件。用文档编辑工具打开该文件

添加：

[user]

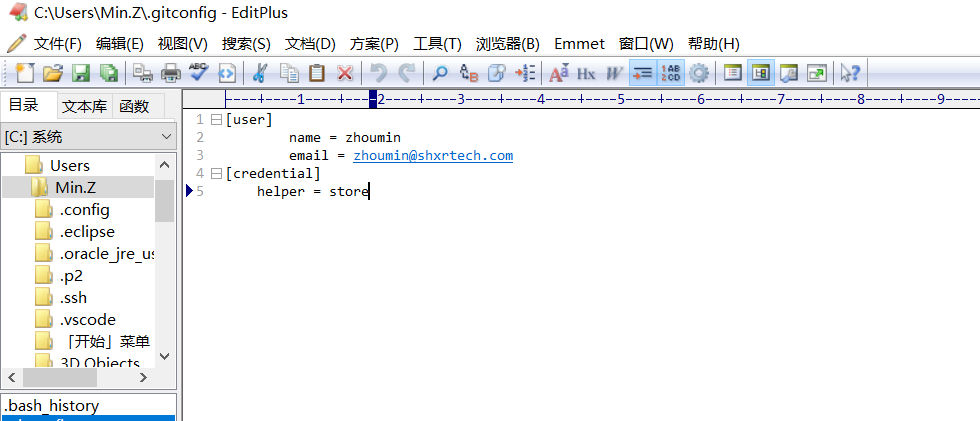
name = zhoumin

email = zhoumin@shxrtech.com

[credential]

helper = store

如图：



然后git只要输入一次账号密码就可以保存。

**方法二：**

直接在git bash 中执行命令：git config --global credential.helper store

在输入一次账号密码就可以保存了

1. Git branch

一般用于分支的操作，比如创建分支，查看分支等等，

**1.1 git branch**

　　　　　　不带参数：列出本地已经存在的分支，并且在当前分支的前面用"\*"标记

**1.2 git branch -r**

　　　　　　查看远程版本库分支列表

**1.3 git branch -a**

　　　　　　查看所有分支列表，包括本地和远程

**1.4 git branch dev**

　　　　　　创建名为dev的分支，创建分支时需要是最新的环境，创建分支但依然停留在当前分支

**1.5 git branch -d dev**

　　　　　　删除dev分支，如果在分支中有一些未merge的提交，那么会删除分支失败，此时可以使用 git branch -D dev：强制删除dev分支，

**1.6 git branch -vv**

　　　　　　可以查看本地分支对应的远程分支

**1.7 git branch -m oldName newName**

　　　　　　给分支重命名

1. Git checkout

　　　　1. 操作文件  2. 操作分支

**2.1 操作文件**

　　　　　　2.1.1 git checkout filename 放弃单个文件的修改

　　　　　　2.1.2 git checkout . 放弃当前目录下的修改

**2.2 操作分支**

　　　　　　2.2.1 git checkout master 将分支切换到master

　　　　　　2.2.2 git checkout -b master 如果分支存在则只切换分支，若不存在则创建并切换到master分支，repo start是对git checkout -b这个命令的封装，将所有仓库的分支都切换到master，master是分支名，

**2.3 查看帮助**

　　　　　　git checkout --help

it add .

# [rm 、git rm 、git rm --cached的区别](https://www.cnblogs.com/everfight/p/git_rm_usage.html)

rm

删除文件

git rm

git rm  
当我们需要删除暂存区或分支上的文件, 同时工作区也不需要这个文件了, 可以使用git rm  
git rm file = rm file+ git add file  
rm file删除本地文件  
git add file 提交删除的步骤同步到git仓库

git rm --cached file

当我们需要删除暂存区或分支上的文件, 但本地又需要使用, 只是不希望这个文件被版本控制, 可以使用 git rm --cached  
git rm --cached 会从index里面删除该文件，下次commit的时候会修改git仓库，但是本地的文件还是保留

Untrack and stop tracking files in git

有一些文件我们不想提交到git上去了，但是又被检测到有变化，显示Untracked files，我们可以采用  
git rm --cached file来停止跟踪这些文件