**git的结构：**  
工作区（working directory） 暂存区（staging index） 版本库（git directory）  
  
**git的四种状态：**  
1.untracked 未被追踪的  
2.Modifed 工作区，未加入暂存区的  
3.Staged 工作区修改的提交到暂存区，但未提交到版本库的  
4.Commited：数据安全的保存在本地库中  
5.一种Vim编辑状态：退出命令 ：Esc + : q ! （ 四个键 )   
**git指令：**  
git init ：初始化  
ls -a : 查看文件夹所有文件  
git log : 查看提交信息  
git add ：添加至暂存区  
git add . ：所有的（加小数点）  
git commit -m “first commit” ：提交至版本库 “first commit”  
git commit -am "all" :将工作区的所有内容提交到版本库  
git config --global user.name 用户名 ：提交用户名字，同改（全局的，去掉--global则为当前项目）  
git config --global user.email 邮箱 ：提交用户邮箱，同改  
git config -- list / git config -l ： 查看用户信息  
git pull 地址 : 取回远程地址某个分支的更新，与当前本地的指定分支合并（有冲突，则更改）  
git pull origin next：master ：取回origin主机的next分支，与本地的master分支合并  
git clone 网址：从远程仓库中克隆项目（HTTPS/SSH：H速度快 S用于用户认证）  
git push 网址 “” master：上推（提交）到master分支  
**问题及解决：**  
一：当之前Git log的user.name和user.email出错，则更改之前的信息：  
\* ./git-m 或者 git-m  
1）向导会让你输入需要修正的email（括弧内提示会自动给你找到不规范的email，你可以直接回车）  
2）输入需要替换成正确的用户名  
3）输入需要替换成正确的email（公司邮箱）  
参考网址：<http://baike.corp.taobao.com/index.php/Git-m>  
二：当git出现22 Time Out（网络超时）错误，可以采取http方式克隆上推  
**git参考文档：**  
根据我的经验，对GIT的使用最好通过命令行。   
前端有专门的命令行工具，比自带的要好用，如果有需要，可以找我要。  
**git基本命令如下：**  
git clone XXX //添加项目到本地  
git checkout -b 分支号 //建立并切换到创建的当前分支  
git checkout 分支号 //分支切换  
git branch //查看本地分支, 并且给出当前分支号  
git branch -a //查看本地分支&&查看远端分支（但并不全，查看全部远端分支请去gitlab上，或者查找其他命令）  
git pull //获取master分支或者已经绑定的远端分支上的代码  
git pull origin 分支号 //获取远端分支上的代码  
git add . //(注意有点号) , 相当于svn 的add 功能， 添加所有新创建的文件到缓存。  
git commit -am XXX //提交代码到本地缓存区域  
git status //查看当前代码状态，一般push代码前都要习惯性看一下当前代码状态。  
git push origin 分支号 //推送代码到远端分支， 注意push前应该先git pull 一下对应分支的远端代码， 才可以提交。  
git push origin 分支号 -f //强行推送代码到远端，（gitlab 上有些情况不允许你推送代码，比如你的版本领先master，但是你又不能去获取master代码， 有时候跳版本的时候会遇到这种问题）  
git merge 分支号 //合并当前和分支号的代码。  
git log //查看当前分支的本地提交记录  
git reset --hard 提交的UUID //把代码恢复到某次提交上去 （这个uuid可以是本地，也可以是远端， 注意，这个命令非常好用。值得掌握。）  
在git的世界里，如果你提交过代码， 那么理论上他就已经存在于你的缓存里，就算没有push也可以通过其他手段找回来。 所以要勇于删除没用的代码和无用的文件， 将来需要时用git找回，会是聪明的做法，始终保持代码和文件系 统清洁。  
  
**项目流程：当前分支：红色**  
本地自测->发预发->后台同学联测  
克隆：git clone 网址(http/ssh)  
修改完毕之后：  
git add . ：将所有的修改信息提交到暂存区（feat/blz）  
git commit -m "" ：将暂存区的内容提交到版本库（feat/blz）

// 提交代码可以用git-cz库来代替，代码规范  
git pull origin 分支（自己本地的）：将远程分支的代码拉下来，进行更新对比（feat/blz）  
git push origin 分支（自己远程的）：将本地分支的代码推上去（feat/blz）  
git checkout daily/0.66 ：切换成本地日程分支（feat/blz）—>（daily/0.66）  
git pull origin daily/0.66 ：拉取远程日常分支，进行更新对比 （daily/0.66）  
git merge origin feat/blz ： 合并（墨迹）feat/blz到daily/0.66 （daily/0.66）  
git push origin daily/0.66: 将本地代码推上远程日常分支上（daily/0.66）  
git checkout feat/blz：切换成本地开发分支（自己的）