**问题及解决：**

* git pull出现vim编辑器窗口的解决方法：

首先按ESC键退出编辑状态，然后按shift + ；键，再按wq！保存退出，按q！则不保存退出

* 本地回退add（代码不变）

git reset HEAD index.js （后加文件即此文件，不加则全部add文件）

* 本地回退commit（代码不变）

**git reset --soft HEAD^**

* git log出现提交版本信息，如何退出？

按键小写英文字母q ，然后按键Enter即可退出

* git切换远程分支（新创建本地分支在origin的基础上）

\* git checkout -b 本地分支名 origin/远程分支名

\* git fetch origin 远程分支：本地分支

* git pull撤销

git reflog : 查看git记录（HEAD@{0}： 最近的记录）

git reset –hard <HEAD> ： <HEAD> ： 前方字符串识别码

git reset –hard qwerty ： qwerty： 前方字符串识别码

**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  
  
**git指令：**  
git init ：初始化  
ls -a : 查看文件夹所有文件

git branch : 查看本地所有分支

git branch -r : 查看远程分支

git checkout 本地分支\*\* ：将本地目前分支切换成本地分支\*\*

git checkout -b 本地分支名\*\* origin/分支名\*\* : 创建本地分支\*\*，并且拉取远程分支\*\*代码（以切换为分支名\*\*

git fetch origin origin/分支名\*\* : 同上，不过需要git checkout 为分支名\*\*（未切换为分支名\*\*）  
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 reset --soft HEAD^ :**  commit之后，想撤销commit  
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

**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**  
本地环境自测->发预发环境（测试同学）  
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/bailizhang）  
git commit -m "" ：将暂存区的内容提交到版本库（feat/bailizhang）  
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/bailizhang）  
git push origin 分支（自己远程的）：将本地分支的代码推上去（feat/bailizhang）  
git checkout daily/0.1.6 ：切换成本地日程分支（feat/bailizhang）—>（daily/0.1.6）  
git pull origin daily/0.1.6 ：拉取远程日常分支，进行更新对比 （daily/0.1.6）  
git merge origin feat/bailizhang ： 合并（墨迹）feat/bailizhang到daily/0.1.6 （daily/0.1.6）  
git push origin daily/0.1.6: 将本地代码推上远程日常分支上（daily/0.1.6）  
git checkout feat/bailizhang：切换成本地开发分支（自己的）