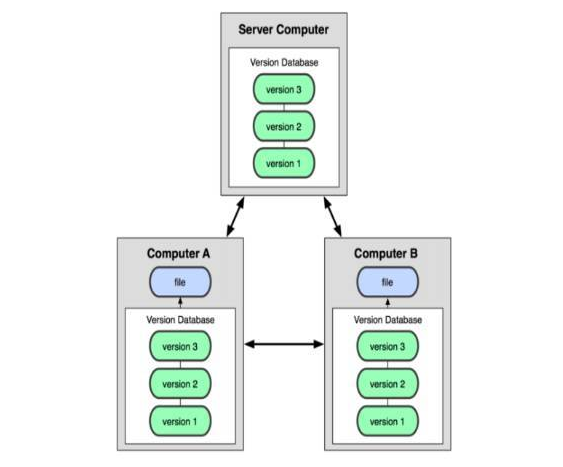
整理Git的使用方法

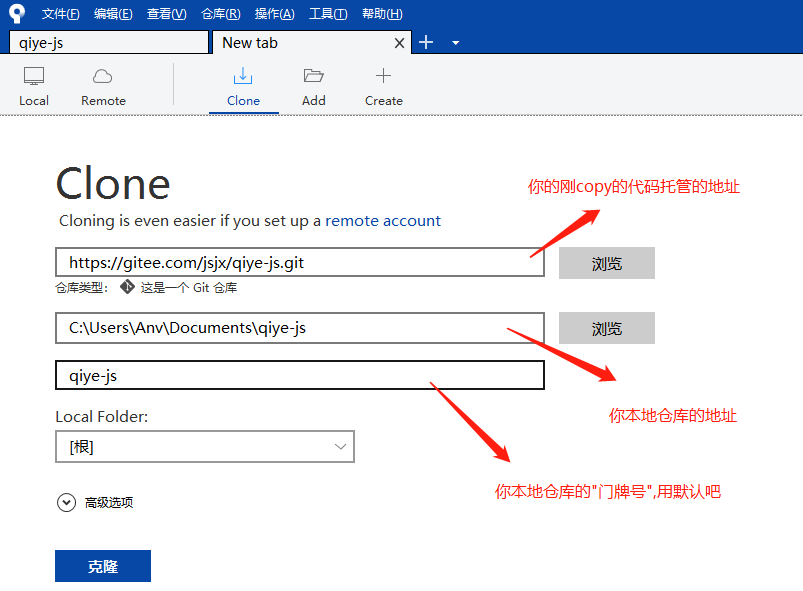
首先,在说Git使用方法之前,你要明白什么是Git,它是干什么的?Git是一个分布式版本控制软件.简单点说就是在项目里,每个人的电脑里都有一个完整的本地版本库,假如你的同事修改了文件A,就可以把修改的最新版本'推送'并更新到代码托干平台上(比如我们的竹简项目是托管在码云上),当你'拉取'的时候,此时最新的版本库也将会同步在你电脑的本地库里,项目里的每个人都可以对代码进行修改标注,互不影响,修改不妥还能吃后悔药.在多人多部门协作一个项目的时候,是无限受用的好帮手呢.从网上荡了个图,可以参考看一下:

如果还不是很明白Git的基本概念,请先阅读廖雪峰老师的教程吧.

接着开始你的使用吧

天才第一步:创建你本地的库

首先,自己去码云<https://gitee.com>去注册一个账号,并加入到企业,进入企业项目.这里我用”江苏界享电子商务有限公司 / qiye-js”这个项目作为实例,

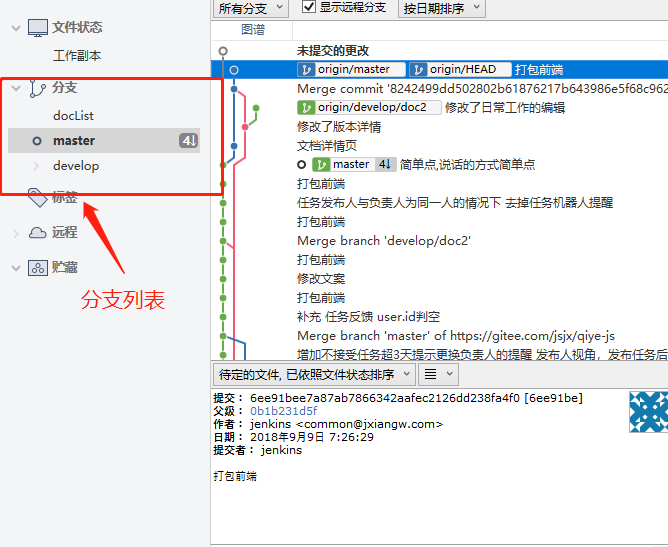
好了,(win下)打开你的Sourcetree,Ctrl+N,如下:

well,记住你仓库的门牌和地址,此时你的仓库可以叫上Webstorm开始营业干活了.

分支功能了解一下

什么是分支

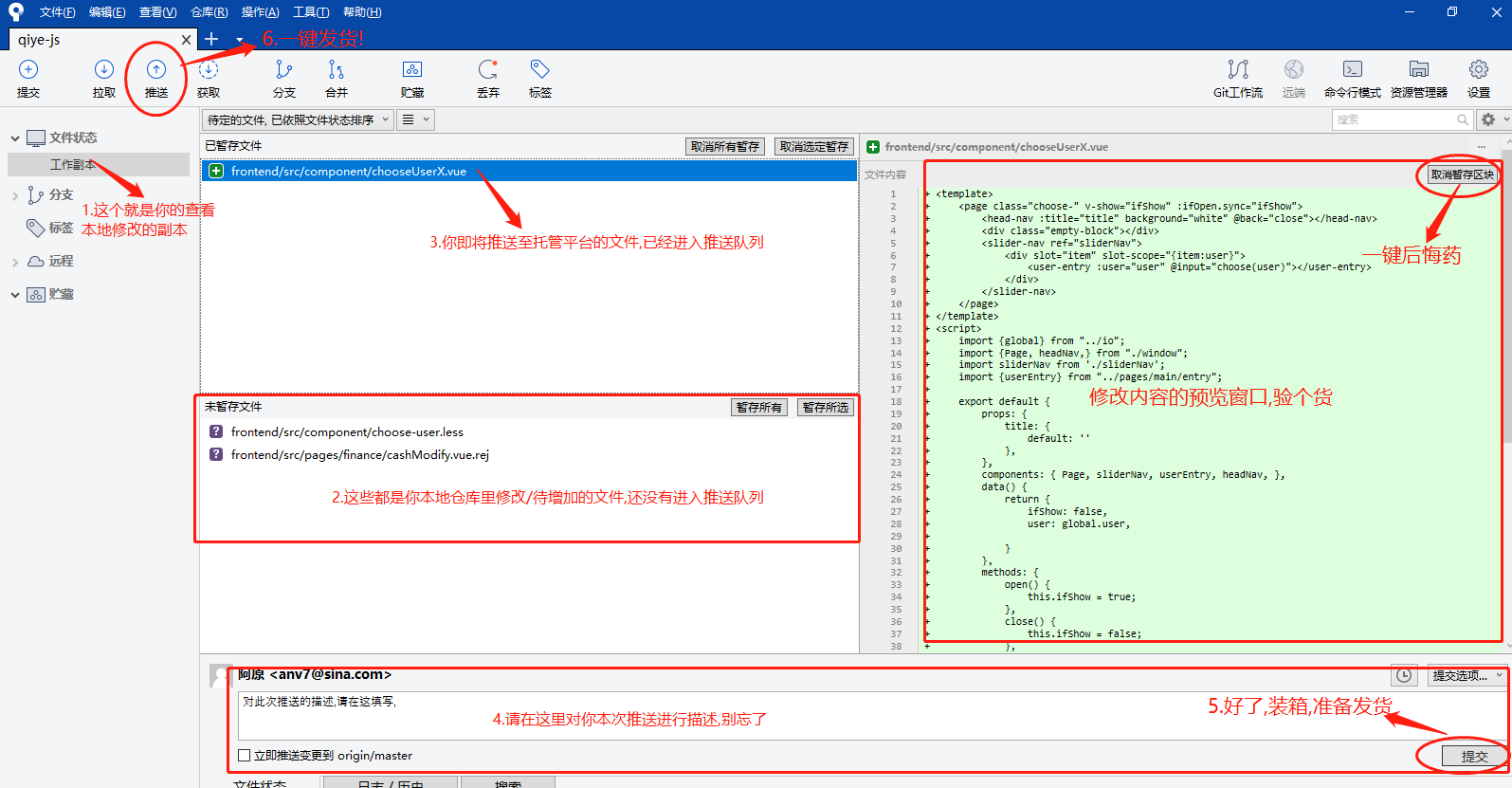
假设你准备开发一个新功能，但是需要两周才能完成，第一周你写了50%的代码，如果立刻提交，由于代码还没写完，不完整的代码库会导致别人不能干活了。如果等代码全部写完再一次提交，又存在丢失每天进度的巨大风险。

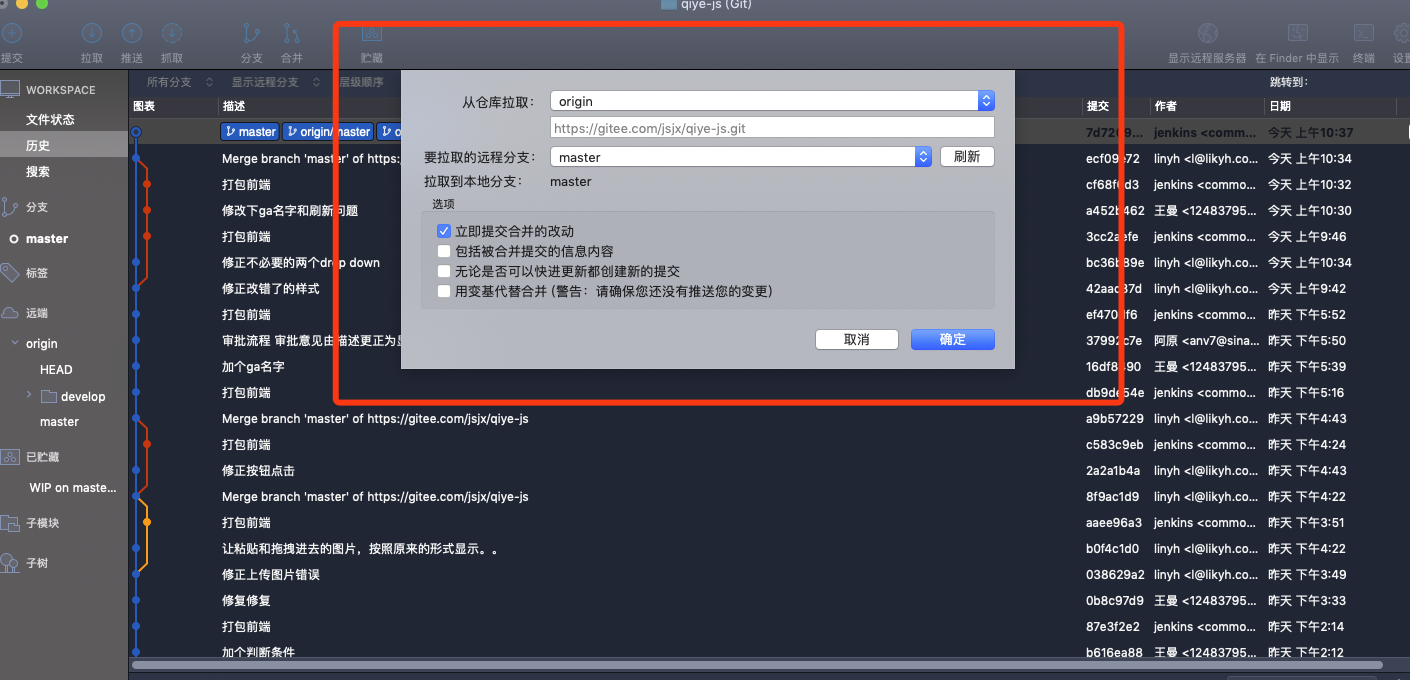
现在有了分支，就不用怕了。你创建了一个属于你自己的分支，别人看不到，还继续在原来的分支上正常工作，而你在自己的分支上干活，想提交就提交，直到开发完毕后，再一次性合并到原来的分支上，这样，既安全，又不影响别人工作。

你要知道，主分支叫master就像游戏里的主线任务一样的存在，如果把我们的项目比作一棵大树的话，master或许可以称为大树的主干，而且其他这些你自己的小分支，就是让“这棵大树成长”的一部分。

一般小的提交或者修改经检查无误后可以直接提交在master分支上，没做完的代码自己按照规范自己创建个分支提交上去，待你的小项目写完之后，选择主分支master，点击菜单栏“合并”按钮，并选择你分支的最新版本合并就ok了，也许你也会问，为什么只用选择最近一次的更新，其他的历史更新就不选了吗？当然不是这样，你的子分支是本地的版本库，选择最新更新，其实已经包括了你历史修改添加的所有内容。U get。

日常的代码推送

 提交代码之前注意一下你所在的分支，你是要推送至主分支master还是要推送至你的子分支，不得不说这是新手村里面经常犯得一个错误，

代码的拉取

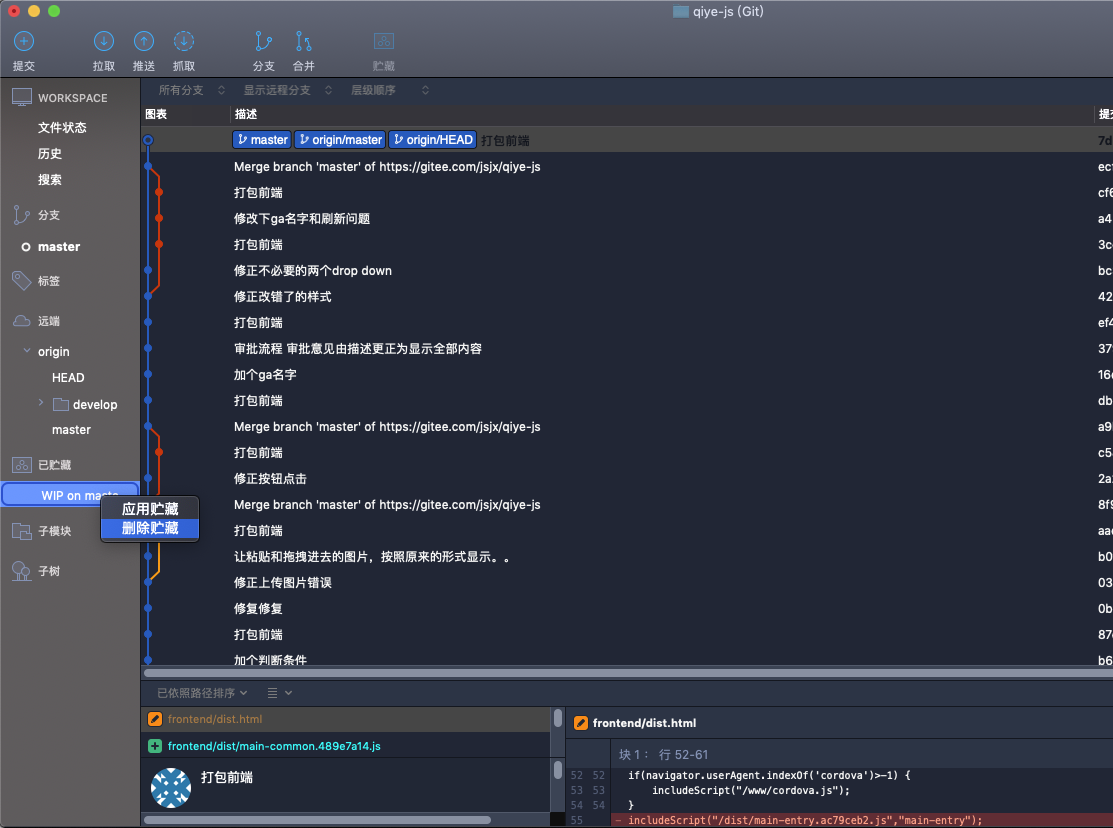
在你每次代码推送之前，先要把线上仓库别人完成的代码同步至你的本地仓库，保持你本地仓库是最新的代码版本后，才可以进行你刚写代码的推送，要不然，哼哼，报错喽。

Ok，我们说下sourcetree的贮藏功能

上节讲到当你所在的当前分支有他人的提交时，可以通过拉取，拉取到本地。如果拉取有冲突，可先把你本地修改临时贮藏之后再进行拉取。

如下图，选中你即将贮藏的修改，点击贮藏按钮后的描述可随意，能找到就行，然后点击拉取，同步线上仓库的代码修改。

拉取成功后，右键之前贮藏的文件，应用回去，没有冲突就行了



版本的悔棋操作

首先要说明的是在你的代码推送到git仓库或是服务器的时候，你务必要检查刚刚你写的代码，哪怕老板要的特别急，格式正不正确？代码缩进对不对?（ps:林总是会骂人的）…当然了，最重要的一点就是保证自己的代码要跑的起来，一个人的代码故障被push到了git里导致整个团队的环境故障，这真的不是一件光荣值得津津乐道的事情。

题外话 2018年马蜂窝前端bug事件了解一下

遇到冲突不怕怼，强制提交就是干

首先是代码冲突，没有解决，也没有检查，强制提交强制回怼就对了，最后能跑起来也是666的。

好了，言归正传，打个比方，由于你的失误操作，将问题代码推送到了远程，想重新修改上传代码，此时我们就可以将版本回退，如图