目录

[GIT是什么 2](#_Toc873610)

[安装启动和配置GIT 2](#_Toc873611)

[GIT常用命令 2](#_Toc873612)

[commit命令 3](#_Toc873613)

[log命令 3](#_Toc873614)

[reset --hard id 命令 3](#_Toc873615)

[git的作用 3](#_Toc873616)

[多人协同工作传代码 3](#_Toc873617)

[建立远程仓库 3](#_Toc873618)

[使用SSH协议push项目 4](#_Toc873619)

[给SSH建立关联 4](#_Toc873620)

[使用.gitignore文件 5](#_Toc873621)

[git分支 5](#_Toc873622)

[查看项目的分支 5](#_Toc873623)

[创建一个分支 5](#_Toc873624)

[切换分支 5](#_Toc873625)

[合并分支 5](#_Toc873626)

# GIT是什么

GIT: **G**lobal **I**nformation **T**racker(全球信息追踪)

https://blog.csdn.net/free\_wind22/article/details/50967723

在公司是一个团队在开发同一个项目那么就有：前端开发人员，后端开发人员，如何将他们写的**代码进行共享**？在公司电脑里写的项目，在家里的电脑上如何**拿到代码继续开发**？（用优盘传，网盘传，邮箱传等等，但是他们都不是最好的工作模式。）**还有就是对同一个文件大家都要改，改完后如何代码合并成了一个问题。**

如果有一个**中央仓库**，大家都在这里取代码，都在这里提交代码，就方便了。

**Git是一个协同开发工具 Git是一个分布式版本控制工具**

# 安装启动和配置GIT

1、下载安装：<https://www.git-scm.com/download/>

2、启动：在项目根目录右键，选择Git Bash Here

3、配置：配置邮箱、用户名

git config --global user.email [abc@qq.com](mailto:abc@qq.com)

git config --global user.name abc

配置如果想修改，重新执行即可

4、查看配置的邮箱和用户名

git config user.email

git config user.name

# GIT常用命令

* git init

该命令用来初始化一个项目，让项目中的文件受到git的监视。会在项目的根目下生成一个.git的文件夹。git init 之后，如果文件夹是空的，那么git是不会监视该文件夹的。

* git status查看文件的状态，被监视的文件(绿色)，untracker（红色）
* git add 文件名 让单个文件受到git监视 也可以空格隔开多个文件
* git add -A或者git add . 让所有文件受到git的监视

## commit命令

git commit -m ‘备注修改的信息’ 提交文件

这条命令的使用逻辑是：先修改项目下的文件，查看status，add文件，commit文件，再看status。

## log命令

git log

查看修改的日志，这里可以看到修改后提交的备注信息和日期时间形如

commit e64ae41b5a30e7c753147c800274227e478f2b4a (HEAD -> master)

Author: lichuang <59289488@qq.com>

Date: Sat Feb 2 21:22:56 2019 +0800

## reset --hard id 命令

git reset --hard e64ae41b5a

这条命令可以让项目下的文件还原到id的位置

# git的作用

0共享代码

1监视文件的修改

2保存文件的修改，将来可以回到以前文件的状态

3多人协同工作传代码合并代码

# 多人协同工作传代码

首先使用git init --bare命令创建一个中央仓库

然后团队中的人都可以到这个仓库中拿代码下来（git pull URL master）工作，拿代码下来也可以（git clone URL ）或 （git clone URL 别名）

也可以提交（git push URL master）代码上去

**git clone = git init + git pull + git remote add origin**

**clone是本地没有仓库时从远程仓库拿代码下来，pull是本地有仓库从远程拉取新内容下来（如果有的话）进行融合merge**

# 建立远程仓库

1、在github上注册一个账户

2、新建一个仓库

3、根据需要：使用https协议push项目或者使用ssh协议push项目

注意：使用https协议push代码每次都要输入github网站的账号和密码比较麻烦。



# 使用SSH协议push项目

1、在命令行中输入 ssh-keygen -t rsa 创建密钥

2、然后一路回车，看信息提示。

3、找到存放（C:\Users\Administrator \.ssh）的文件id\_rsa.pub,打开文件将里面的内容复制一下

4、将复制的内容放入github 个人中心的“设置”栏目下的 “SSH和GPG密钥”里，在这里要新建一个SSH密钥，标题随便取，内容就粘贴刚刚复制的id\_rsa.pub里面的内容。

5、在命令行中输入 git push [git@github.com:web007/study.git](mailto:git@github.com:web007/study.git) master 即可提交内容至远程仓库（不需要每次输入账号和密码）

# 给SSH建立关联

为了方便 git push [git@github.com:web007/study.git](mailto:git@github.com:web007/study.git) master 这个命令，建立如下设置关联

git remote add origin git@github.com:web007/study.git

origin以后就是 [git@github.com:web007/study.git](mailto:git@github.com:web007/study.git) 这个地址的别名了。

git push origin master

git pull origin master

# 使用.gitignore文件

**使用此文件可以将项目下的文件忽略掉不会add到暂存区。**

生成.gitignore文件

1、用记事本直接保存，文件类型设为所有文件类型即可

2、在.gitignore文件中输入我们想要忽略的文件名或文件夹名即可

# git分支

使用git分支功能能解决现实开发中多任务的问题，这里的任务是指要开发的功能，并且任务之间代码是相互独立的。比如（开发场景）：你开发好了一个登录模块，现在正在开发购物车模块，但此时有人告诉你，你的登录模块出现了bug，那么你就要使用git的分支来解决代码合并共享的问题了。

## 查看项目的分支

使用 git branch 查看项目的分支（带\*号的表示的是你当前正处在的分支）

当你第一次git commit -m ‘第一次提交’ 提交之后就会生成主分支 master

每当你开发一个任务的时候就应该创建一个分支

## 创建一个分支

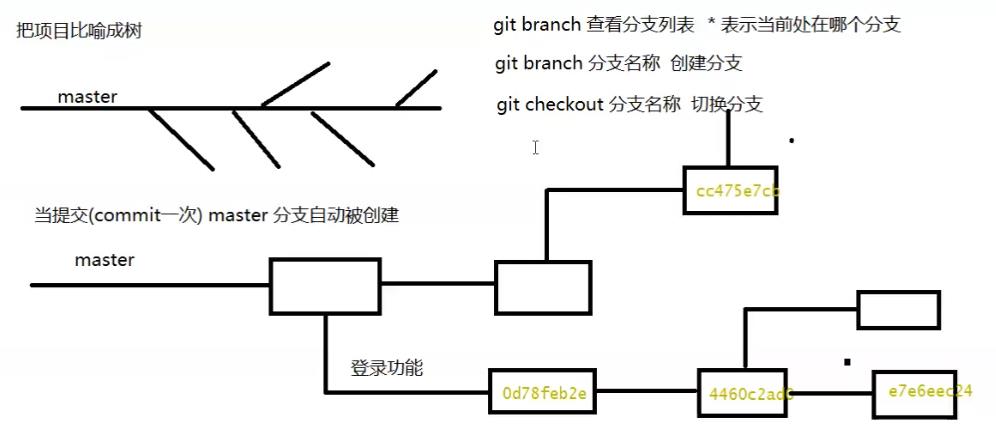
使用 git branch 分支名 创建一个分支

## 切换分支

使用 git checkout 分支名 切换到你想要的分支

## 合并分支

使用 git merge 分支名 将分支和当前所在的分支合并



# VIM

今天用git commit -m “注释”提交的时候，注释写错了，于是各种查资料开始了和git bash vim的纠缠。。。（网上的资料我真是没操作成功，不过最后还是摸索出来了。

首先 使用 git commit --amend 命令（修改最近一次提交的注释信息），会进入到vim 编辑器

然后 你会发现编辑器里你怎么输入都没反应，这是因为vim处在不可编辑状态，

按下字母键 c（此时进入编辑状态），可以开始修改注释信息了

修改好后，你会发现怎么都退出不了，然后如下操作：

按下Esc (退出编辑状态)； 接着连按两次大写字母Z，你会惊喜的发现，终于保存好退出来了！

windows 系统 ，亲测有效！！拿走不谢！

输入

:q

wq

两个任意一个都可以退出

1、vi & vim 有两种工作模式：

（1） 命令模式：接受、执行 vi & vim 操作命令的模式，打开文件后的默认模式；

（2） 编辑模式：对打开的文件内容进行 增、删、改 操作的模式；

#在编辑模式下按下 ESC 键，回退到命令模式。

2、创建、打开文件：$ vi [filename]

（1）使用 vi 加 文件路径（或文件名）的模式打开文件，如果文件存在则打开现有文件，如果文件不存在则新建文件，并在终端最下面一行显示打开的是一个新文件。

（2）键盘输入字母 “i”或“Insert”键进入最常用的插入编辑模式。

3、保存文件：

（1）在插入编辑模式下编辑文件。

（2）按下 “ESC” 键，退出编辑模式，切换到命令模式。

（3）在命令模式下键入"ZZ"或者":wq"保存修改并且退出 vi 。

（4）如果只想保存文件，则键入":w"，回车后底行会提示写入操作结果，并保持停留在命令模式。

4、放弃所有文件修改：

（1）放弃所有文件修改：按下 "ESC" 键进入命令模式，键入 ":q!" 回车后放弃修改并退出vi。

（2）放弃所有文件修改，但不退出 vi ，即回退到文件打开后最后一次保存操作的状态，继续进行文件操作：按下 "ESC" 键进入命令模式，键入 ":e!" ，回车后回到命令模式。

# git提交后出现nano界面,解决方法

这个是使用nano进行编辑提交的页面，退出方法为：  
Ctrl + X然后输入y再然后回车，就可以退出了  
如果你想把默认编辑器换成别的：  
方法一、在GIT配置中设置 core.editor: git config --global core.editor "vim"  
  
  
方法二、编辑~/.gitconfig文件。在core中添加editor = vim。如此以后在使用git的时候就自动使用vim作为编辑器