Git详解

**Git简介**

Git的创始人同时也是Linux的创始人Linus。Git的作用分为3块：

* 管理项目的版本
* 多人协同开发一个项目
* 分布式保存项目

Git不像SVN一样，它可以没有中央仓库服务器，每个开发人员都有一个完整的仓库，在开发完毕后在进行互相push。当然对于多个人的团队开发git也是有远程仓库方便开发人员快速修改合并代码的。

参考地址：<https://blog.csdn.net/weelyy/article/details/82823798>

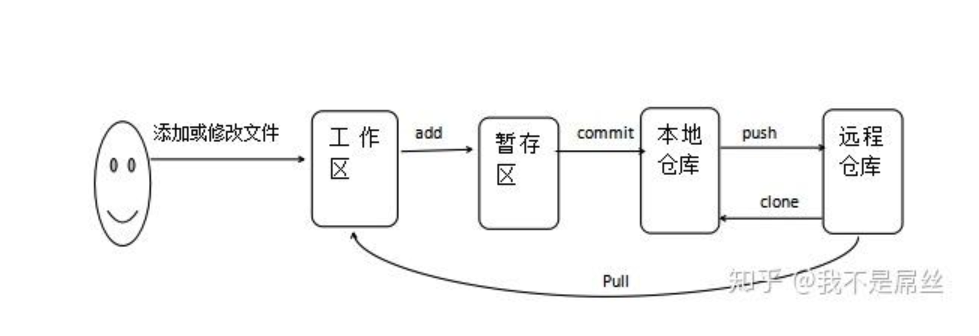
**Git的工作流程**

Git的工作区域主要有四个，分别是：

* 远程仓库
* 本地仓库
* 工作区
* 暂缓区

执行操作如下：

* 首先我们先远程从仓库clone到本地仓库，然后在工作区中添加或者修改文件。
* 提交文件到本地仓库，我们需要先通过add命令将文件放入暂缓存区中，然后再通过commit命令将文件放入到本地仓库中
* 我们通过push命令将文件从本地仓库推送到远程仓库中。
* 更新别人提交的代码需要我们通过pull命令将代码更新到工作区。



**Git的几个主要命令**

git add: 将工作区的修改提交到暂存区

git commit: 将暂存区的修改提交到当前分支

git reset：会退到某一个版本【可以先用git log查看需要回退的commit id】,然后git reset --hard id。

git stash: 保存某次修改

git pull: 从远程更新代码

git push: 将本地代码更新到远程分支上

git reflog: 查看历史命令

git status: 查看当前仓库的状态

git diff: 查看修改

git log: 查看提交历史

git revert: 回退某个修改

git add的用法：

* git add -A：保存所有的修改
* git add .：保存新的添加和修改，但是不包括删除
* git add -u：保存修改和删除，但是不包括新建文件
* 一般默认使用git add -A就行

git commit用法：

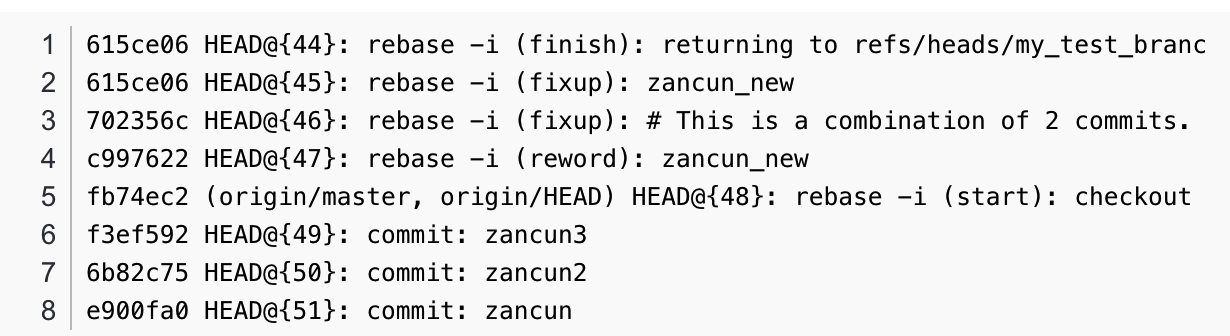
* git commit -m “本次提交描述”：该命令会将git add .存入暂存区修改内容提交至本地仓库中，若文件未添加至暂存区，则提交时不会提交任何修改。
* git commit -a: 相当于运行git add -u 把所有目录下的文件加入暂存区域在运行git commit。注意，对于新增的文件，并没有被commit。
* git commit -am “本次提交描述”或者git commit -a -m “本次描述”等同于上面的-a 和-m。
* git commit --amend：修改最后一次提交，有时候如果提交注释书写有误或者漏提文件，可以使用此命令。对于漏提的文件，需要git add到缓存区之后，git commit --amend才能将修改追加到最近的一次提交上。

git reset的用法：

* git reset根据-soft -mixed -hard，会对working tree 和index和HEAD进行重置。
* git reset HEAD^: 回退版本，一个^表示一个版本，可以多个，另外也可以使用git reset HEAD~n这种形式
* git reset commitId： 可以先用git log 获取commitId，然后回退到指定的版本。
  + soft 参数：git reset –soft HEAD~1 意为将版本库软回退一个版本，所谓软回退表示将本地版本库的头指针全部重置到指定版本，且将这次提交之后的所有变更都移动到暂存区。
  + 默认的mixed参数：git reset HEAD~1意为将版本回退1个版本，将本地版本库的头指针全部重置到指定版本，且会重置暂存区，即这次提交之后的所有变更都移动到工作区。
  + Hard参数：git reset –hard HEAD~1 意为将版本库回退一个版本，但是不仅仅是将本地版本库的头指针全部重置到指定版本，也会重置暂存区，并且会将工作区代码清空（工作区是clean状态）
  + 注意,soft参数和默认参数都不会修改工作区代码，只有hard参数才会修改工作区代码。
* git reset HEAD filename回退文件，将文件从暂存区会退到工作区，此时不带hard和soft参数。

git reflog的用法：

* 如果在回退之后，又再次想回到之前的那个版本，git reflog可以查看所有分支的所有操作记录（包括commit和reset的操作），包括已经被删除的commit记录，git log则不能察看已经删除了的commit记录



比如说，会退到commit: zancun3，只需要：

git reset –hard f3ef592（或者HEAD@{49}）即可

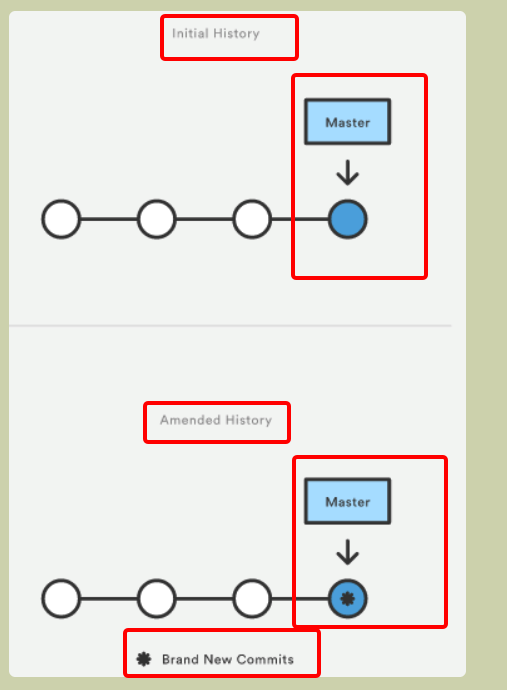
这个命令对于找回丢失的代码非常有用。

git revert的用法：

* git revert反转提交。撤销一个提交的同时会创建一个新的提交，也就是用一个新的提交来消除历史提交所做的任何修改。
* git revert commitId：revert指定的一个commit
* git revert HEAD~3：revert到指定的倒数第四个commit

git commit --amend:

git commit --amend命令可以用来修复最近一次的commit，可以让你合并你缓存区上的修改和上一次的commit，而不是提交一个新的快照，还可以用来编辑上一次的commit描述。



记住amend不是修改最近一次commit，而是整个的替换掉它，对于git来说是一个新的commit。

git commit --amend会合并缓存区的修改和最近一次的commit，然后用生成的新的commit替换掉的。如果缓存区没有内容，那么利用amend可以修改上一次commit的描述。

开发过程中很容易填写了不够准确的commit描述，--amend就是用来fix这些错误的。

例子：





编辑器的默认描述会是上一次commit的描述， --no-edit能让我们修复commit，而且不要修改commit描述。

