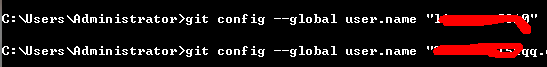
保持数据完整性：

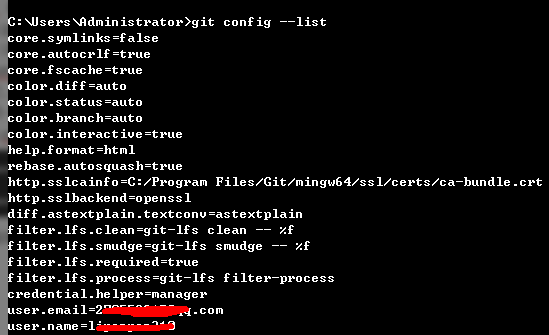
* 在保存到 Git 之前，所有数据都要进行内容的校验和（checksum）计算，并将此结果作为数据的唯一标识和索引。Git 的工作完全依赖于这类指纹字串，所以你会经常看到这样的哈希值。实际上，所有保存在 Git 数据库中的东西都是用此哈希值来作索引的，而不是靠文件名。Git 使用 SHA-1 算法计算数据的校验和，通过对文件的内容或目录的结构计算出一个 SHA-1 哈希值，作为指纹字符串。该字串由 40 个十六进制字符（0-9 及 a-f）组成。
* 对于任何一个文件，在 Git 内都只有三种状态：已提交（committed），已修改（modified）和已暂存（staged）。已提交表示该文件已经被安全地保存在本地数据库中了；已修改表示修改了某个文件，但还没有提交保存；已暂存表示把已修改的文件放在下次提交时要保存的清单中；
* git中如果命令是全写，则可以用两个-，即—
* Git 保存的不是文件差异或者变化量，而只是一系列文件快照;
* Git 中的分支，其实本质上仅仅是个指向 commit 对象的可变指针; 由于 Git 中的分支实际上仅是一个包含所指对象校验和（40 个字符长度 SHA-1 字串）的文件，所以创建和销毁一个分支就变得非常廉价。说白了，新建一个分支就是向一个文件写入 41 个字节（外加一个换行符）那么简单，当然也就很快了。

关于Git，和Linux内核是一个父亲。其先用BitMover公司的Bitkeeper软件进行版本管理；在05年左右开发出Git（分布式版本管理系统），Git 是一个软件，它允许你通过提交对一个系统（或一组）文件的历史进行注释。这些提交便是在给定时间点对系统做出的差异“快照”。 Github 获取 Git 中的提交历史，并将其存储在互联网上，因此你可以从任一一台电脑访问它。你在本机（例如：你当前正在使用的电脑）推送（pushing）提交到 Github，然后，从另一台新的或不同的电脑上拉取（pulling）这些提交。

通过cmd注册git账户，命令如下：



红色遮盖部分分别为用户名和邮箱，使用下面命令查看注册结果：（截图结果为成功）



Git的命令行模式：

1 Git保存的是独立的版本，有工作区域，暂存区域，Git仓库三部分；

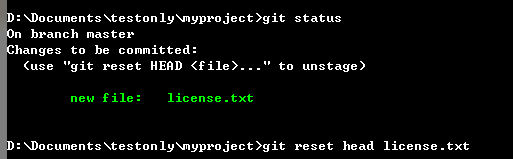
2 在你要建立版本迭代的文件夹位置输入命令git init，效果如下：

 **提交命令如下：**

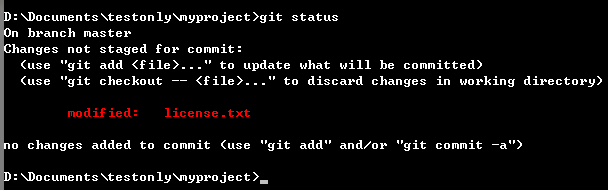


第一步： git add + 文件名 （\*为目录下所有文件都提交）

第二步： git commit –m ”对文件的描述“



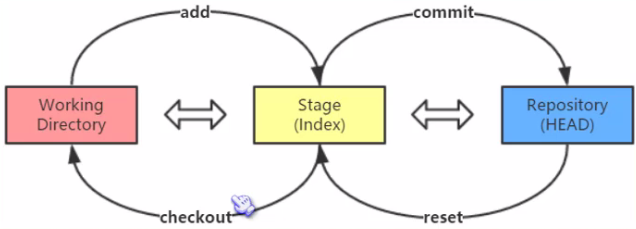
这个命令是查看状态，且下面的git reset head (可以加文件名)，不加的话是直接把暂存区域恢清空，不提交，加的话是只清空指定文件。用于commit操作之前。



上面这个命令是由于你提交之后，又修改了本地文件，与暂存区域文件不一样，所以提示你进行修改。

* git checkout -- (文件名) 是用来使用云端文件覆盖本地文件；（--左右各有一个空格）

我们可以使用git log查看提交历史（是commit记录，不是add哦）

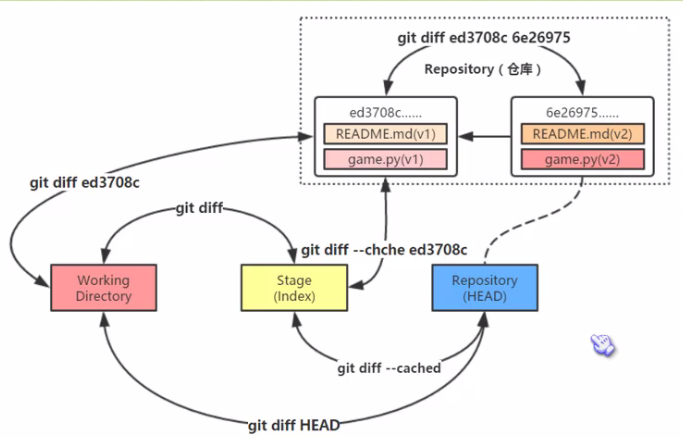


* git Reset head^ 移动head指向，在云端返回至上一次版本，但之前的版本不删除；
* git reset --soft：回退到某个版本，只回退了commit的信息，不会恢复到index file一级。如果还要提交，直接commit即可（只回退了仓库里面的内容）；
* git reset --mixed：此为默认方式，不带任何参数的git reset，即时这种方式，它回退到某个版本，只保留源码，回退commit和index信息（即回退仓库和暂存区域）；
* git reset --hard：彻底回退到某个版本，本地的源码也会变为上一个版本的内容，此命令慎用！（回退仓库与暂存区域还有覆盖本地文件至以前的版本）
* git reset + 版本快照 （也可以加文件名），这样可以回滚到指定的版本

(以上回滚也可以通过版本号往前回滚，上面的回滚属于mixed，当让，也可以在hard，sort后面加版本号回滚)

**比较不同：**

* git reflog可以查看之前的所有commit记录；
* git diff 是比较暂存区域和工作目录之间的区别
* git diff 快照id1 快照id2 比较两个快照之间的区别
* git diff 快照id 是比较仓库中的版本与当前工作目录中的版本的区别
* git diff –cached （+id） 比较的是仓库最新的快照（或指定版本）与暂存区域的对比



**删除文件：**

* git commit –amend （-m “”）进入界面更新或者直接更新；
* git rm +文件名 只是本地删除文件和暂存区域文件（版本一样时），表示在commit时不提交该文件
* git rm -f +文件名 只是本地删除文件和暂存区域文件（即使版本不一样），表示在commit时不提交该文件；
* git rm --cached + 文件名 只是删除暂存区域文件，表示在commit时不提交该文件

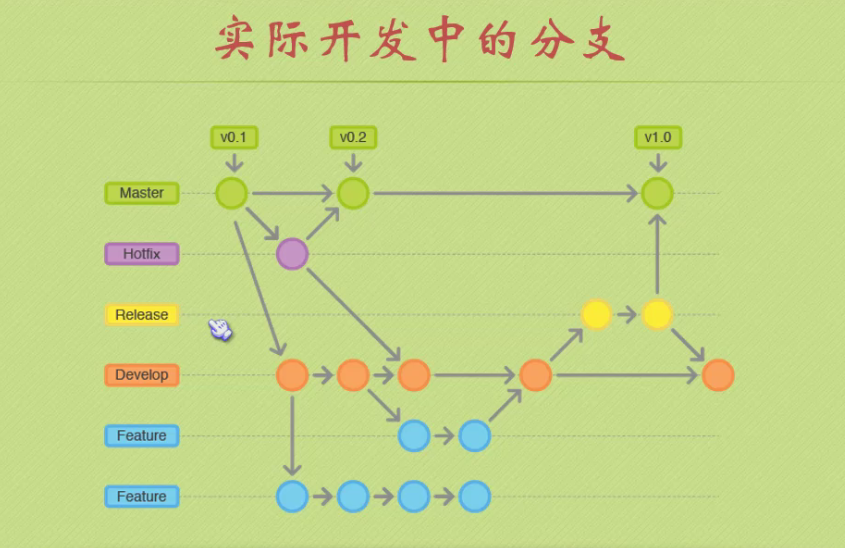
**修改文件名字：**

* git mv 旧的文件名 新的文件名

**Git 分支：**



* git branch 分支名 命令创建分支



分支的合并：

* git merge +分支名 合并分支，把分支合并到主体
* git branch –a 查看远程分支
* git branch 查看本地分支
* git branch –d 分支名 删除本地分支
* git checkout –b 分支名 创建并切换到这个分支
* git checkout 分支名 是切换到当前分支