1.Git是一种分布式版本控制系统。（Version Control System）

①每个人电脑里都有完整的版本库。

②分支管理

2.集中式的版本控制系统（中央服务器）

①CVS—Concurrent(并行) Version System

②SVN—Subversion版本控制系统

3.查看git上的用户名和邮箱：

$git config user.name

$git config user.email

4.修改用户名和邮箱地址：

$git config --global user.name "username"

$git config --global user.email "email"

5.创建版本库（repository）

①切换至计算机e盘

$ cd e:/

②pwd:print working directory

③GBK:国标扩展，前三个字的第一个字母

④初始化一个Git仓库，使用

git init

⑤添加文件到Git仓库

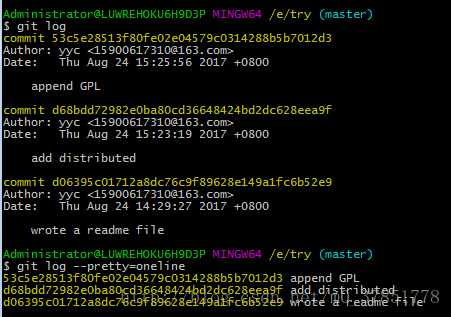
1.git add <file> 可重复多次使用，添加多个文件

2.git commit -m "description"

⑥git status命令可以让我们时刻掌握仓库当前的状态

git diff命令查看具体修改的内容（difference）

⑦git log --pretty=oneline



⑧cat = concatenate files and print on the standard output

6.版本回退

1.HEAD指向的版本就是当前版本。使用命令git reset --hard commit\_id 可获取任何一个历史版本。

2.git log查看提交历史，确认要推到哪个版本。

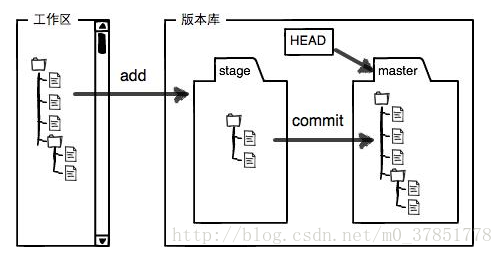
3.git reflog查看命令历史，确认要回到未来的哪个版本。

7.工作区和暂存区

1.工作区（working directory）---可见的目录。

2.版本库（repository）---工作区中的隐藏目录.git，算是Git的版本库。

3.工作区的文件通过git add <filename> 进入暂存区（Stage），通过git commit一次性将暂存区中的所有文件提交到分支（默认为master）。



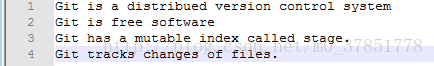
8.管理修改

1.Git跟踪并管理的是修改，而非文件。

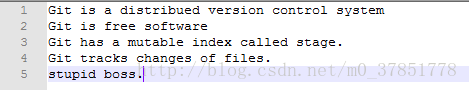
2.每次修改，如果不add到暂存区（Stage），就不会加入到commit中。

9.撤销修改

1. 初始状态—readme.txt

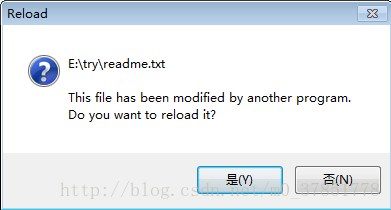


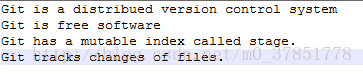
1. 一不小心这样了



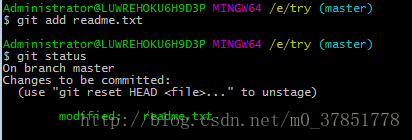
解决之法：

è¿éåå¾çæè¿°



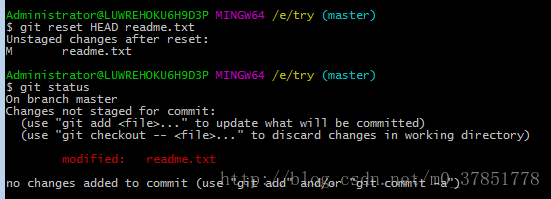


1. 又一不小心add到暂存区了

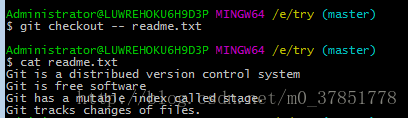


解决之法：

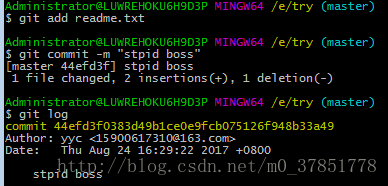
1.先从暂存区退到工作区



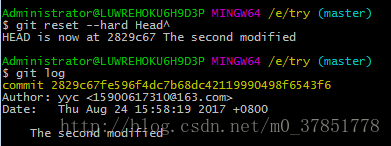
2.使用checkout将其从工作区中撤销



1. add + commit



解决之法：



小结：

场景1.想丢弃工作区的修改，用命令git checkout -- filename.

场景2.已经添加到暂存区了，用命令git reset HEAD filename 回到场景1，再执行场景1的操作。

场景3.都提交到版本库，用命令git reset --hard HEAD^,直接完事。（注意hard前没有空格）

10.删除文件

在工作区创建了一个文本文件test.txt，将其add到暂存器并commit至版本库。rm test.txt只能删除工作区中的该文件。

场景1.如果想要删除版本库中的该文件，git rm test.txt，通过git commit -m "remove test.txt"。

场景2.如果是误删了作用区中的该文件，可借助版本库中的该文件对其恢复：git checkout -- test.txt。只能恢复文件到最新的版本，但会丢失最近一次提交后你所修改的内容。

11.远程仓库

1.SSH:Secure Shell 安全外壳协议

12.添加远程库

1.关联一个远程仓库：$ git remote add origin https://github.com/server-name/learngit.git，在此之前确保在该目录下有个隐藏文件.git[git init]。

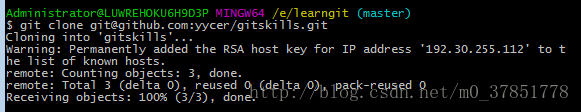
2.关联后，使用命令git push -u origin master 第一次推送master分支的所有内容。[u---upstream]

3.此后，每次本地提交后，有需要就可以使用命令git push origin master推送最新修改。

13.从远程仓库克隆

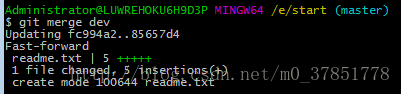
1.要克隆一个仓库，首先必须要知道仓库的地址，然后使用git clone命令克隆。

2.Git支持多种协议，包括https，但通过ssh支持的原生git协议速度最快。

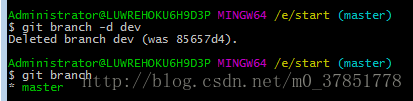


14.创建和合并分支

1.git merge命令用语合并指定分支到当前分支。（以下合并dev分支至master）



2.git branch -d dev 删除dev分支



3.小结

1.查看分支：git branch

2.创建分支：git branch <name>

3.切换分支：git checkout <name>

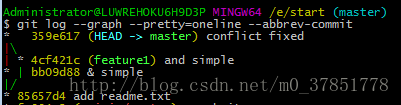
4.创建+切换分支：git checkout -b <name>

5.合并某分支到当前分支：git merge <name>

6.删除分支：git branch -d <name>

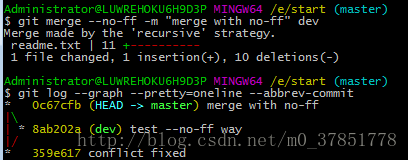
15.解决冲突

1.当一个文件同时被不同的分支修改时就会造成冲突。此时可能会导致Git无法自动合并分支，就必须首先解决冲突，再提交，合并完成。【master | feature1】



16.分支管理策略

--no-ff 方式的git merge



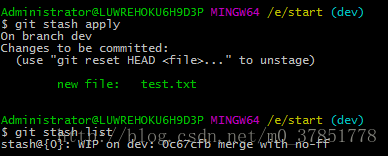
17.Bug分支

1.stash功能：保存现场，直至下次继续工作。

2.修复bug时，先保存当前分支的"现场"，然后创建新的分支进行bug修复，然后合并，最后删除。之后再返回"现场"，有以下两种方式：

a)git stash apply,但恢复后，stash内容并没删除，需要使用git stash drop删除。

b)git stash pop，一步到位。



è¿éåå¾çæè¿°

