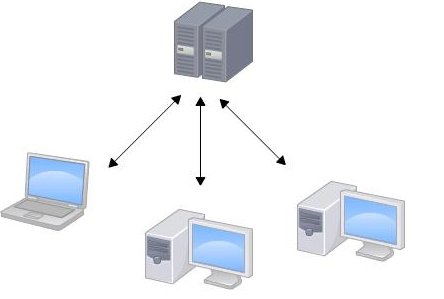
# Git笔记

#### 1.1 Git的诞生

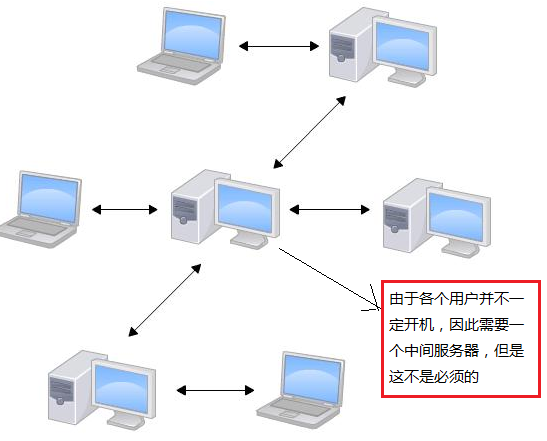
Linus 两周时间就搞定了。

##### 集中式与分布式

集中式:



分布式：



##### 安装Git

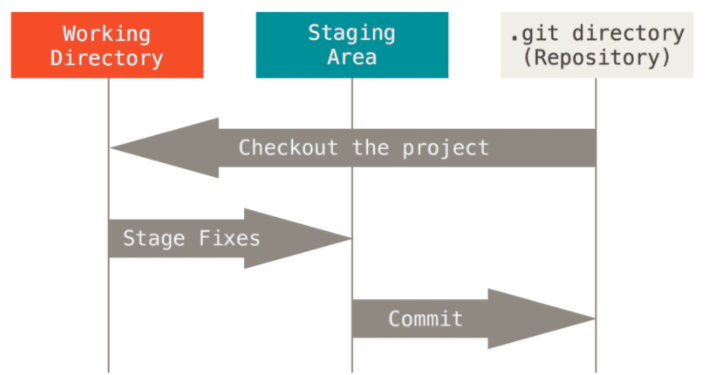
设置邮箱和用户名：



##### Git原理

Git对文件的存储更像是对小型文件系统的**快照**。而想svn等集中式会比较文件的差异性，并记录差异性。

**Git中存在*三种状态*，已修改，已提交，已暂存。*三种目录*：工作目录，暂存目录，Git仓库目录**



Git的基本工作流程：

1. 在工作目录中修改文件  **vi xxx**
2. 暂存文件，将文件快照放入暂存区域 **git add**
3. 提交更新，将快照永久存到Git仓库中 **git commit**

#### 1.2 创建版本库

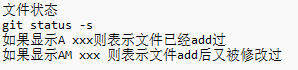
##### 1. 创建版本库



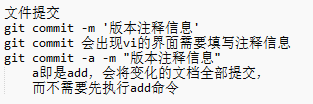
##### 2. 添加文件



##### 3. 文件状态

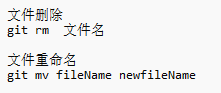


##### 4. 提交版本



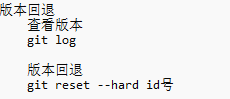
**注意: -a 并不会提交新的文件，只是会提交原本存在又被修改的文件**

##### 5. 文件删除与重命名



#### 1.3 时光穿梭

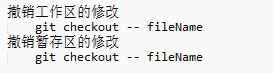
##### 1.3.1 版本回退



##### 1.3.2 管理修改



##### 1.3.3 撤销修改



##### 1.3.4 删除文件

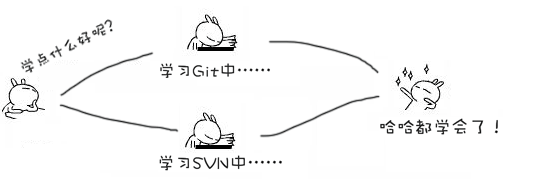
#### 1.4 仓库

添加远程仓库

从远程仓库克隆

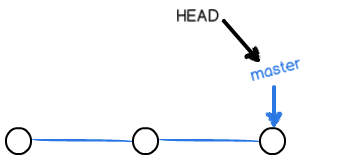
#### 1.5 分支管理

##### 1.5.1 分支的概念与原理

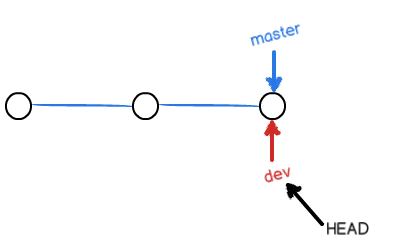


这样，可以使每个人都在编程，以防止，其中一个人代码不完整，其他人不能工作，这样每个人都做自己的，最后做好后合并就可以了。

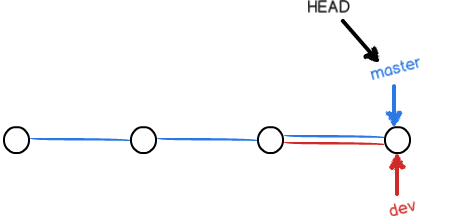
##### 1.5.2 分支的创建与合并



在上图中，HEAD表示自己的指针，master表示主干上的指针。

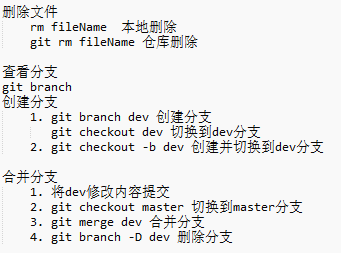


在上图中，可以看出建立新的分支就是建立新的指针dev，并将自己的指针指向dev就可以操作新的分支了。



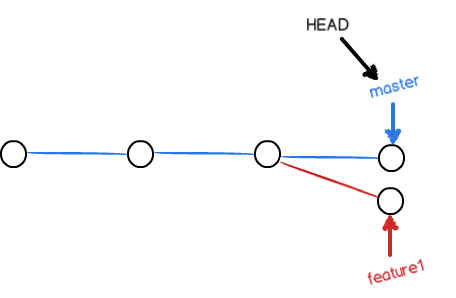
合并分支之后，dev和master都只想新的版本，就可以删除dev了。

**具体实现：**



##### 1.5.3 解决冲突

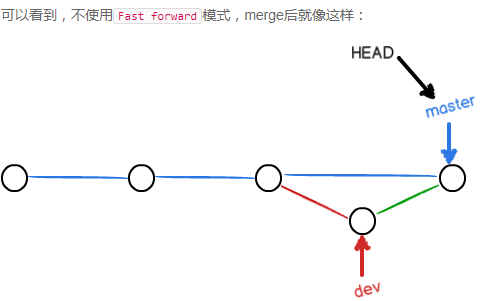
冲突情况：



1. 手动修改
2. 执行：git merge dev

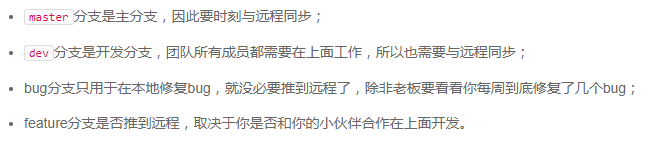
1.5.4 分支管理策略

执行git merge --no-ff -m “注释” ；这是一个特别的commit命令。会提交一个新的commit，--no-ff 会禁用 Fast forward.

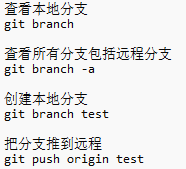
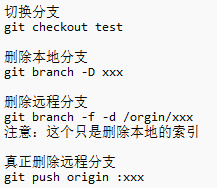


**区别：**合并分支时，加上--no-ff参数就可以用普通模式合并，合并后的历史有分支，能看出来曾经做过合并，而fast forward合并就看不出来曾经做过合并。

##### 1.5.5 协作中的各种分支



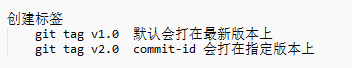
##### 1.5.6 远程分支控制

#### 1.6 标签管理

所谓的标签就和指针一样，只是标签一旦确定的就不会动了，其实每个版本本身含有标签，就是commit-id，但是这个数字太长，因此，我们可以自己创建标签，比如v1.0版本。

##### 1.6.1 创建标签



##### 1.6.2 操作标签





#### 1.7 自定义Git

##### 1.7.1 配置Git服务器

1. 安装Git

2. 在linux上创建一个专门负责管理代码的用户



1. 创建证书登陆
   1. **home/git/.ssh/authorized\_keys**
   2. **id\_rsa.pub ssh-keygen -t rsa**
2. 初始化Git仓库
   1. **git - - bare --shared init /home/git/repo/opendata.git**
   2. 将仓库所属用户改为git **chown -R git:git opendata.git**
3. 禁用git用户通过shell登陆
   1. **Vi /etc/passwd**
   2. git:x:503:503::/home/git:/bin/bash 修改为：git:x:503:503::/home/git:/sbin/nologin

//添加远程指针

git remote add “远程指针名” url

//修改添加到暂存区

git add –all

//暂存区提交

git commit -m “discription”

//推送到远程

git push 远程指针名 分支名