

Git 的基本原理

本质上，Git 是一套内容寻址（content-addressable）文件系统。那么 Git 是怎么进行寻址呢？其实，寻址无非就是查找，而 Git 采用 HashTable 的方式进行查找，也就是说，Git 只是通过简单的存储键值对（key-value pair）的方式来实现内容寻址的，而 key 就是文件（头+内容）的哈希值（采用 sha-1 的方式，40 位），value 就是经过压缩后的文件内容。

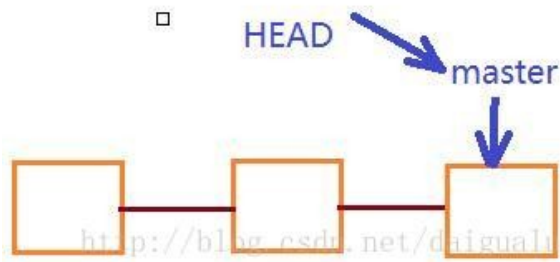
工作流程

基本的 Git 工作流程如下：

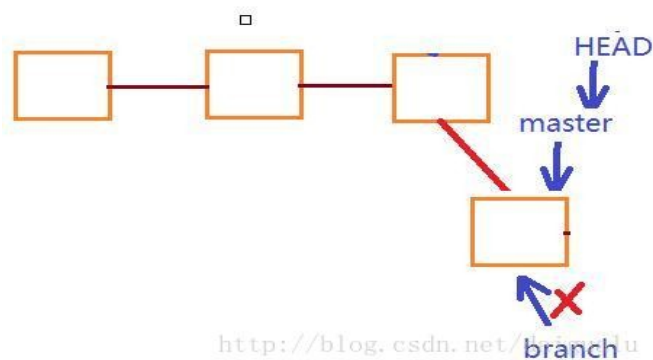
1. 在工作目录中修改文件。 Modified
2. 暂存文件，将文件的快照放入暂存区域。 Staged
3. 提交更新，找到暂存区域的文件，将快照永久性存储到 Git 仓库目录。 Committed

Git 的分支管理原理

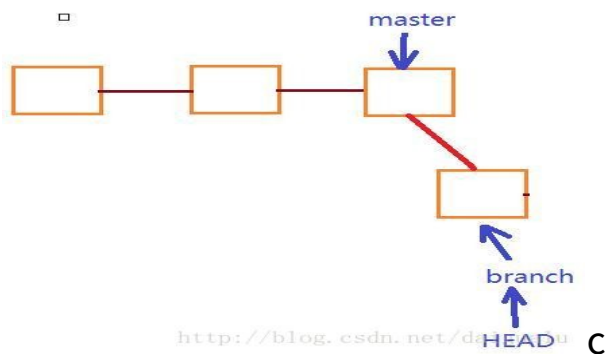
- 1、初始只有一个 master 分支，HEAD 指向 master，如下图 a 所示



2、创建一个 branch 分支。仅仅创建一个 branch 指针，等于 master，并修改 HEAD 使其指向 branch，如下图 b 所示



3、在分支上修改

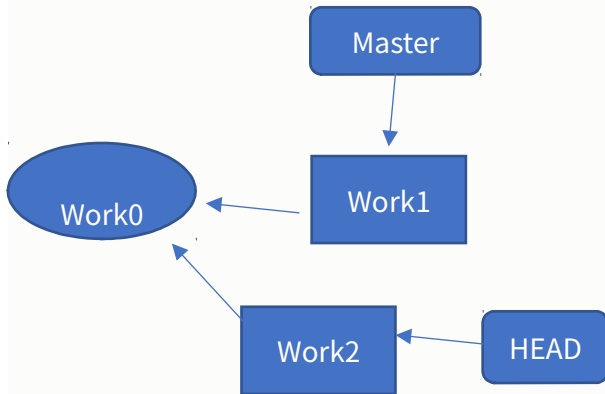


4、将分支上修改合并到 master 上。修改 master 指针指向，指向 branch 中的修改，HEAD 指向 master，删除 branch 指针，

实用：一般线性流程及合并分支

分支合并的各种方法：<https://blog.csdn.net/w605283073/article/details/80099644>

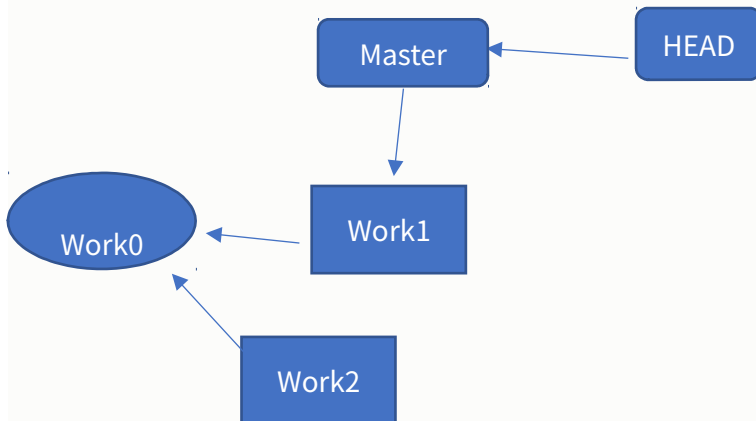
1、去自己的工作分支 `$ git checkout work2`



2、工作

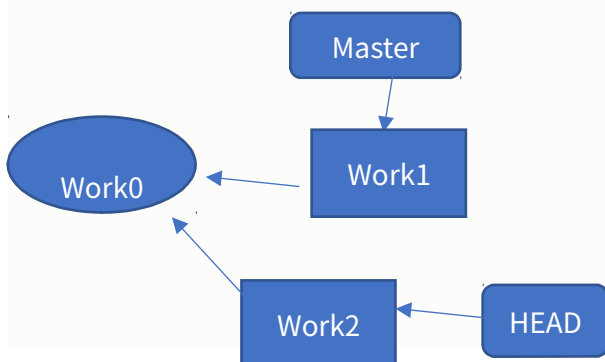
3、提交工作分支的修改 `$ git commit -a`

4、回到主分支 `$ git checkout master`



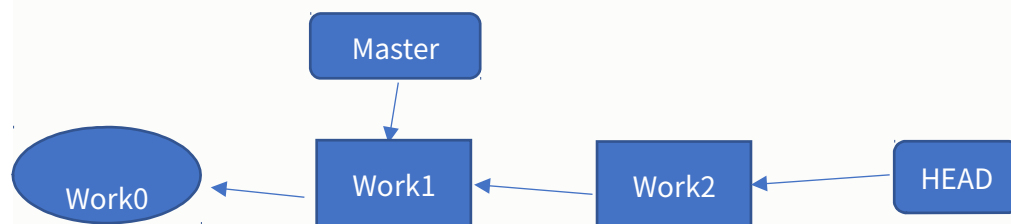
5、获取远程最新的修改，此时不会产生冲突 `$ git pull`

6、回到工作分支 `$ git checkout work2`

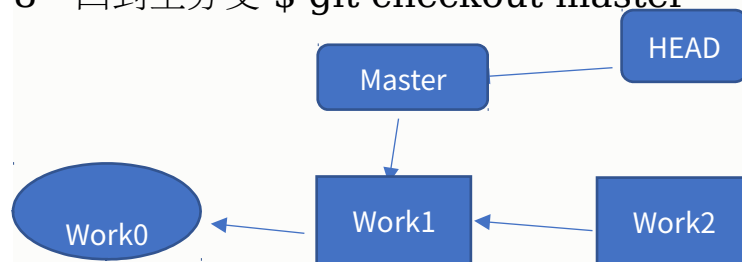


7、用 **rebase** 合并主干的修改，如果有冲突在此时解决

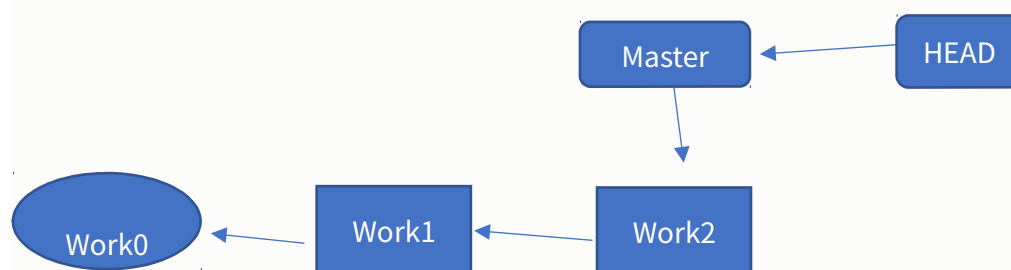
`$ git rebase master`



8、回到主分支 `$ git checkout master`



9、合并工作分支的修改，此时不会产生冲突。 `$ git merge work2`



10、提交到远程主干 `$ git push`