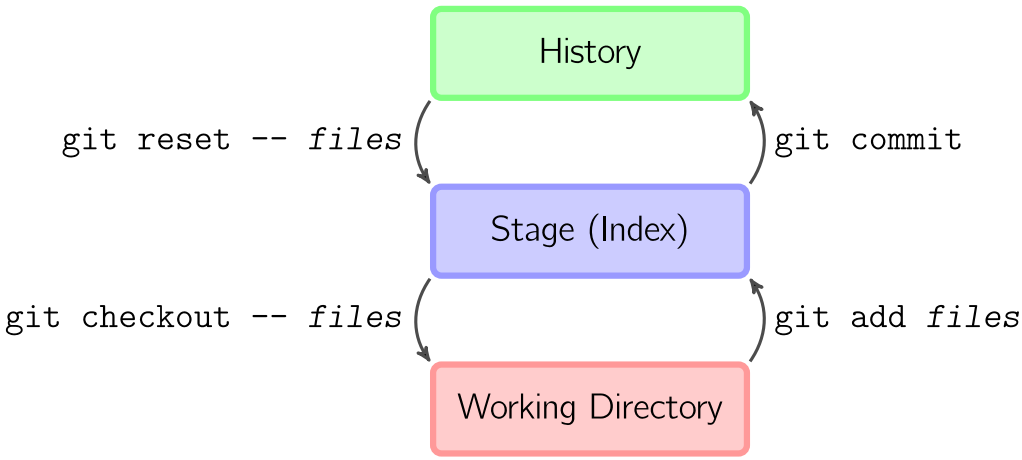
Checkout和Rest的所有谜题 Topic source

Lochuan_Chang
#1 Created at 2017-4-19 21:25

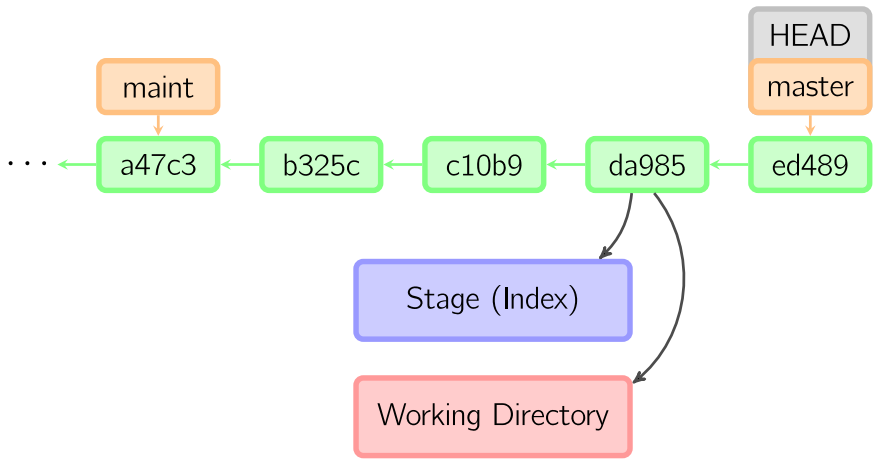


文件层面操作

- git add files 把当前文件放入暂存区域。
- git commit 给暂存区域生成快照并提交。
- git reset -- files 用来撤销最后一次git add files，你也可以用git reset 撤销所有暂存区域文件。
- git checkout -- files 把文件从暂存区域复制到工作目录，用来丢弃本地修改。

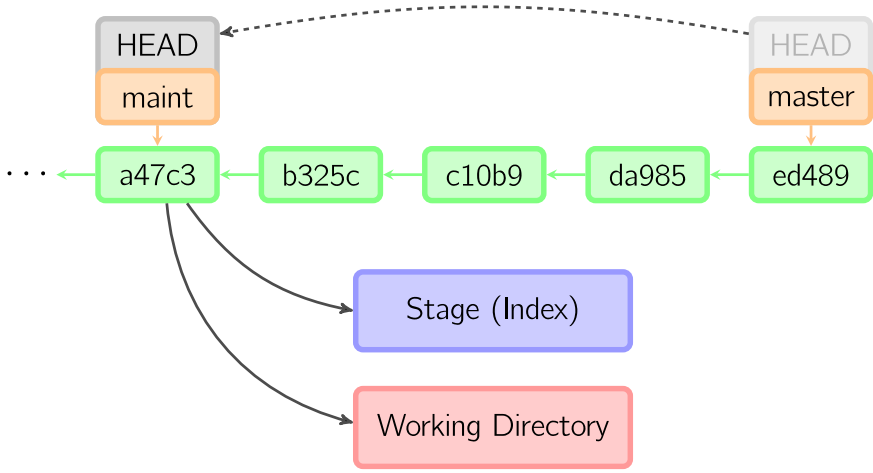
```
A double dash(--) is used in bash built_in commands and many other commands to signify the end of command options, after which only positional parameter are accepted.
```

```
git checkout HEAD~ files
```



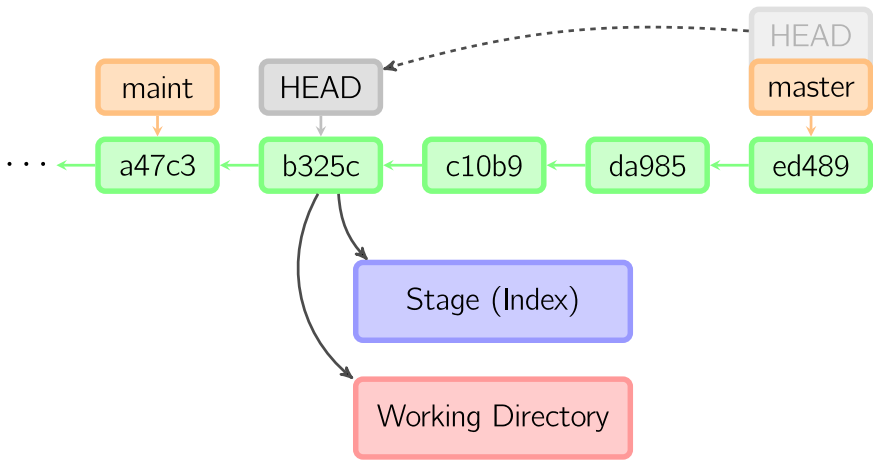
checkout命令用于从历史提交（或者暂存区域）中拷贝文件到工作目录，也可用于切换分支。当给定某个文件名时，git会从指定的提交中拷贝文件到暂存区域和工作目录。比如，git checkout HEAD~ foo.c会将提交节点HEAD~(即当前提交节点的父节点)中的foo.c复制到工作目录并且加到暂存区域中。（如果命令中没有指定提交节点，则会从暂存区域中拷贝内容。）注意当前分支不会发生变化。

git checkout maint



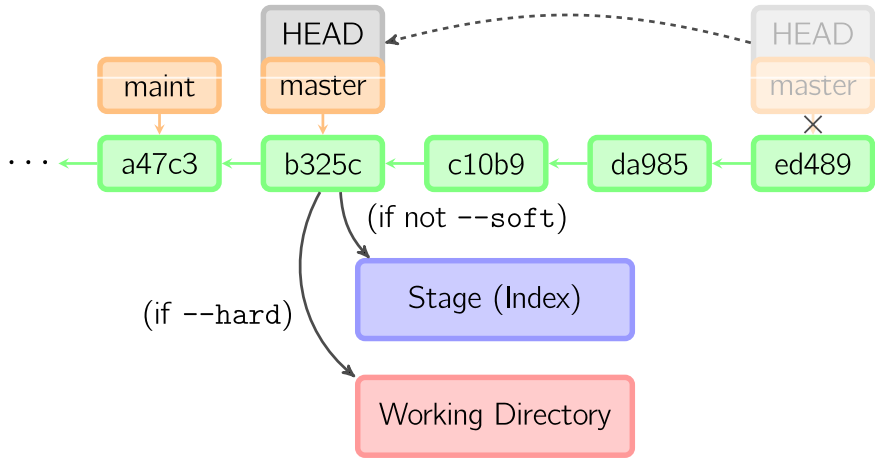
当不指定文件名，而是给出一个（本地）分支时，那么HEAD标识会移动到那个分支（也就是说，我们“切换”到那个分支了），然后暂存区域和工作目录中的内容会和HEAD对应的提交节点一致。新提交节点（~~下图~~上图中的a47c3）中的所有文件都会被复制（到暂存区域和工作目录中）；只存在于老的提交节点（ed489）中的文件会被删除；不属于上述两者的文件会被忽略，不受影响。

git checkout master~3



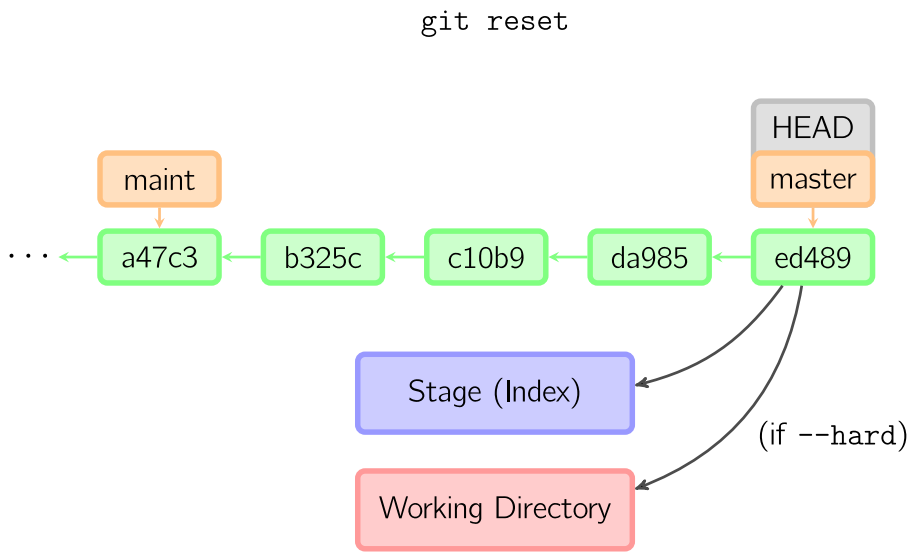
如果既没有指定文件, 也没有指定分枝. 而是只给出一段提交的历史Hash, 只有HEAD会移动到相应的历史提交. 这会造成HEAD分离, 非常危险的操作, 这个命令的说明只是为了满足你的好奇心而已, 不要使用这个命令.

git reset HEAD~3



reset命令把当前分支指向另一个位置，并且有选择的变动工作目录和索引。也用来在从历史仓库中复制文件到索引，而不动工作目录。

如果不给选项，那么当前分支指向到那个提交。如果用--hard选项，那么工作目录也更新，如果用--soft选项，那么都不变。



如果没有给出提交点的版本号，那么默认用HEAD。这样，分支指向不变，但是索引会回滚到最后一次提交，如果用--hard选项，工作目录也同样。

陈浩然MC

#2 Created at 2017-7-18 14:35

第一张图我感觉有地方要修改下，git reset -- file 是将暂存区的文件撤回到工作区，而git checkout -- file 是将工作区的修改恢复到之前未修改的状态（即之前的状态）。对应的关系有点问题

微梦创科网络

#3 Created at 1-3 15:03