

# git使用

创建时间：2020/1/15 16:26

作者：天国之影-Relph

<https://learngitbranching.js.org>

## 基本命令

git checkout HEAD^ 退回到上一个版本

git checkout HEAD~3 退回到前3个版本

git branch -f master <logNum> 将master的版本指定退回到某个版本下

git rebase bugFix 将bugFix的提交并入master

## 高级命令

git cherry-pick <logNum> 将其他分支的提交并入master

交互式rebase

git rebase -i master~4

## 建立标签

git tag v1 C1

git describe <ref>

<ref> 可以是任何能被 Git 识别成提交记录的引用，如果你没有指定的话，Git 会以你目前所检出的位置（HEAD）。

它输出的结果是这样的：

<tag>\_<numCommits>\_g<hash>

tag 表示的是离 ref 最近的标签，numCommits 是表示这个 ref 与 tag 相差有多少个提交记录，hash 表示的是你所给定的 ref 所表示的提交记录哈希值的前几位。

当 ref 提交记录上有某个标签时，则只输出标签名称

当你用 git bisect（一个查找产生 Bug 的提交记录的指令）找到某个提交记录时，或者是当你坐在你那刚刚度假回来的同事的电脑前时，可能会用到这个命令。

git pull --rebase; git push

git checkout -b foo o/master; git commit; git push;

git branch -u o/master foo

git push <remote> <place>

切换到本地仓库中的"master"分支，获取所有的提交，再到远程仓库"origin"中找到"master"分支，将远程仓库中没有的提交记录都添加上去。

git push origin <source>:<destination>

git fetch origin foo

git push origin :foo 删除foo分支

git fetch origin :bar 创建一个新分支bar

git pull origin master:foo 在本地创建一个新分支foo，从远程仓库中的master分支中下载提交记录，并合并到foo，然后再merge到我们的当前检出的分支bar上。