Git操作技巧

Oh Shit, Git!?!
ohshitgit.com

程序员必会的六条黄金 Git 命令，让你效...
Git 是程序员日常使用的最多的工具之一，但...
vikingz.me

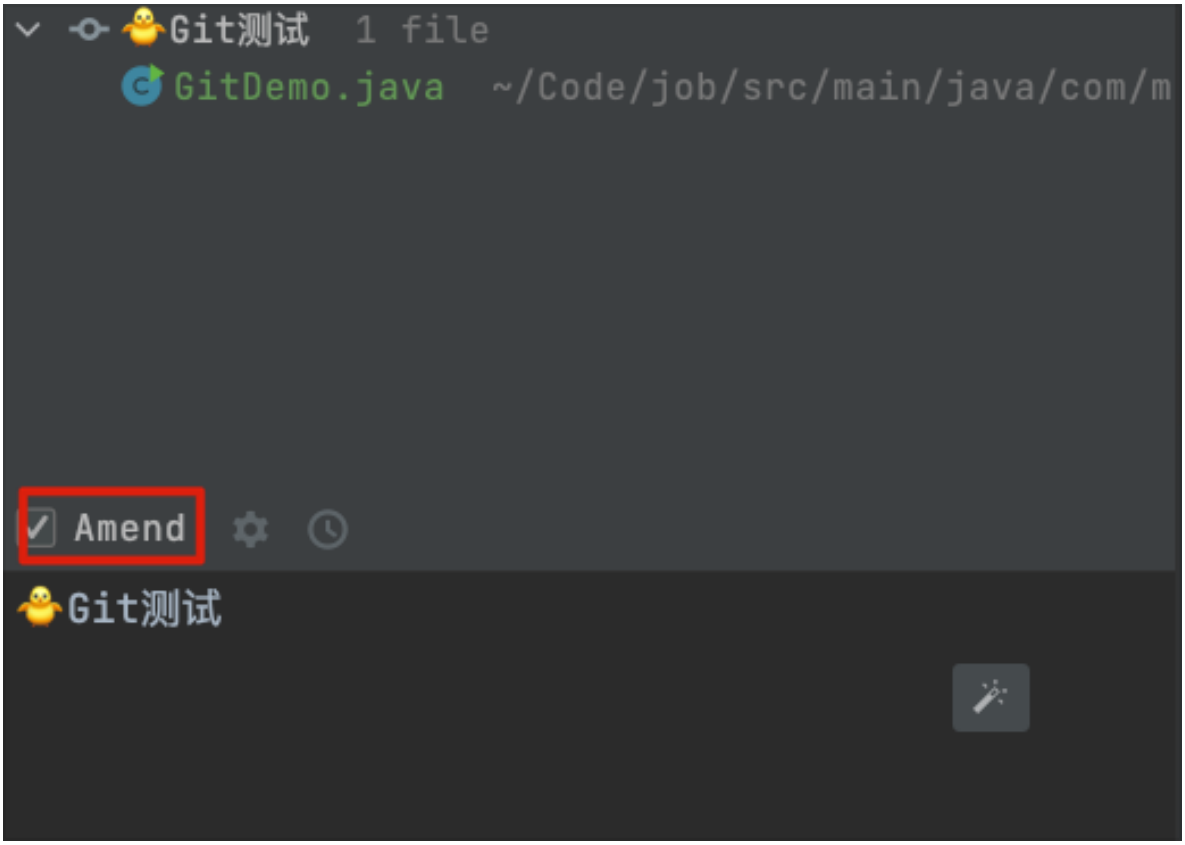
git reset --soft HEAD^：

- --soft：撤销 commit，不撤销 git add . 操作，不删除改动的代码
- --mixed ：撤销 commit，撤销 git add . 操作，不删除改动的代码
- --hard：撤销 commit，撤销 git add . 操作，删除改动的代码，危险操作
- 如果进行了 2 次 commit，都想撤销，使用：HEAD~2

我用的最多的一个 git 小技巧，一次 commit 以后，突然发现还有点问题，又修改了几行代码，不想生成一个新的 commit 怎么办?
前提：提交但是没有 push 到远程分支
添加修改：git add .
使用原来的 commit：git commit --amend --no-edit

```
# 一个 feature 开发完了，已经提交
# 突然发现有几个小问题，修改完了以后，不想生成一个新的commit
# 添加这些小的修改
git add .
# 将修改直接合并到上一个 commit，并且使用上次的提交信息
git commit --amend --no-edit
```

amend 改进



🍺修改仓库对应的远程仓库地址：

git remote set-url origin 仓库地址

🎁查看当前仓库对应的远程仓库地址

git remote -v

这条命令能显示你当前仓库中已经添加了的仓库名和对应的仓库地址，通常来讲，会有两条一模一样的记录，分别是 fetch 和 push，其中 fetch 是用来从远程同步 push 是用来推送到远程
🗑️从 git 中移除某个文件

git rm 文件名 --cached

强制执行
git rm misc.xml --cached -f

代码回退
已提交，未 push
git reset --soft 撤销 commit
git reset --mixed 撤销 commit 和 add 两个动作

已提交，并且 push
git reset --hard 撤销并舍弃版本号之后的提交记录。使用需要谨慎。
git revert 撤销。但是保留了提交记录。

```
Author: zhouguangping <1445101933@qq.com>
~/C/job ››› git reset --hard 874967112a31801f37918bacb62db90ca9407e69
HEAD 现在位于 8749671 🐥查看JVM内存情况
```

提交回退变动
git push origin HEAD -f

```
~/C/job ››› git push origin HEAD -f
Username for 'https://github.com': xiaozhouzhoua
Password for 'https://xiaozhouzhoua@github.com':
总共 0（差异 0），复用 0（差异 0），包复用 0
To https://github.com/xiaozhouzhoua/job.git
 + 8b5a009...8749671 HEAD -> master (forced update)
```

Github 上也已经没有之前的错误提交了

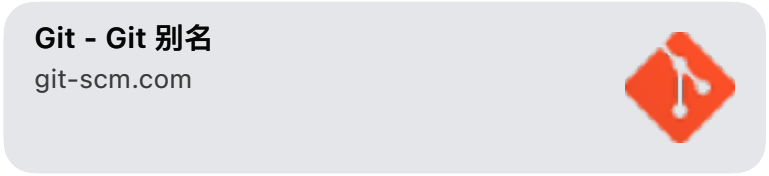
gist.github.com

```
git init                                          # 初始化本地git仓库（创建新仓库）
git config --global user.name "xxx"               # 配置用户名
git config --global user.email "xxx@xxx.com"      # 配置邮件
git config --global color.ui true                 # git status等命令自动着色
git config --global color.status auto
git config --global color.diff auto
git config --global color.branch auto
git config --global color.interactive auto
git config --global --unset http.proxy            # remove  proxy configuration on git
git clone git+ssh://git@192.168.53.168/VT.git     # clone远程仓库
git status                                        # 查看当前版本状态（是否修改）
git add xyz                                       # 添加xyz文件至index
git add .                                         # 增加当前子目录下所有更改过的文件至index
git commit -m 'xxx'                               # 提交
git commit --amend -m 'xxx'                       # 合并上一次提交（用于反复修改）
git commit -am 'xxx'                              # 将add和commit合为一步
git rm xxx                                        # 删除index中的文件
git rm -r *                                       # 递归删除
git log                                           # 显示提交日志
git log -1                                        # 显示1行日志 -n为n行
git log -5
git log --stat                                    # 显示提交日志及相关变动文件
git log -p -m
git show dfb02e6e4f2f7b573337763e5c0013802e392818 # 显示某个提交的详细内容
git show dfb02                                    # 可只用commitid的前几位
git show HEAD                                     # 显示HEAD提交日志
git show HEAD^                                    # 显示HEAD的父（上一个版本）的提交日志 ^^为上两个版本 ^5为上5个版本
git tag                                           # 显示已存在的tag
git tag -a v2.0 -m 'xxx'                          # 增加v2.0的tag
git show v2.0                                     # 显示v2.0的日志及详细内容
git log v2.0                                      # 显示v2.0的日志
git diff                                          # 显示所有未添加至index的变更
git diff --cached                                 # 显示所有已添加index但还未commit的变更
git diff HEAD^                                    # 比较与上一个版本的差异
git diff HEAD -- ./lib                            # 比较与HEAD版本lib目录的差异
git diff origin/master..master                    # 比较远程分支master上有本地分
```

```
支master上没有的
git diff origin/master..master --stat                    # 只显示差异的文件，不显示具体内容
git remote add origin git+ssh://git@192.168.53.168/VT.git # 增加远程定义（用于push/pull/fetch）
git branch                                               # 显示本地分支
git branch --contains 50089                              # 显示包含提交50089的分支
git branch -a                                            # 显示所有分支
git branch -r                                            # 显示所有原创分支
git branch --merged                                      # 显示所有已合并到当前分支的分支
git branch --no-merged                                   # 显示所有未合并到当前分支的分支
git branch -m master master_copy                         # 本地分支改名
git checkout -b master_copy                              # 从当前分支创建新分支master_copy并检出
git checkout -b master master_copy                       # 上面的完整版
git checkout features/performance                        # 检出已存在的features/performance分支
git checkout --track hotfixes/BJVEP933                   # 检出远程分支hotfixes/BJVEP933并创建本地跟踪分支
git checkout v2.0                                        # 检出版本v2.0
git checkout -b devel origin/develop                     # 从远程分支develop创建新本地分支devel并检出
git checkout -- README                                   # 检出head版本的README文件（可用于修改错误回退）
git merge origin/master                                  # 合并远程master分支至当前分支
git cherry-pick ff44785404a8e                            # 合并提交ff44785404a8e的修改
git push origin master                                   # 将当前分支push到远程master分支
git push origin :hotfixes/BJVEP933                       # 删除远程仓库的hotfixes/BJVEP933分支
git push --tags                                          # 把所有tag推送到远程仓库
git fetch                                                # 获取所有远程分支（不更新本地分支，另需merge）
git fetch --prune                                        # 获取所有原创分支并清除服务器上已删掉的分支
git pull origin master                                   # 获取远程分支master并merge到当前分支
git mv README README2                                    # 重命名文件README为README2
git reset --hard HEAD                                    # 将当前版本重置为HEAD（通常用于merge失败回退）
git rebase
git branch -d hotfixes/BJVEP933                          # 删除分支hotfixes/BJVEP933（本分支修改已合并到其他分支）
git branch -D hotfixes/BJVEP933                          # 强制删除分支hotfixes/BJVEP933
```

```
git ls-files                                    # 列出git index包含的文件
git show-branch                                 # 图示当前分支历史
git show-branch --all                           # 图示所有分支历史
git whatchanged                                 # 显示提交历史对应的文件修改
git revert dfb02e6e4f2f7b573337763e5c0013802e392818      # 撤销提交dfb02e6e4f2f7b573337763e5c0013802e392818
git ls-tree HEAD                                # 内部命令：显示某个git对象
git rev-parse v2.0                              # 内部命令：显示某个ref对于的SHA1 HASH
git reflog                                      # 显示所有提交，包括孤立节点
git show HEAD@{5}
git show master@{yesterday}                     # 显示master分支昨天的状态
git log --pretty=format:'%h %s' --graph         # 图示提交日志
git show HEAD~3
git show -s --pretty=raw 2be7fcb476
git stash                                       # 暂存当前修改，将所有至为HEAD状态
git stash list                                  # 查看所有暂存
git stash show -p stash@{0}                     # 参考第一次暂存
git stash apply stash@{0}                       # 应用第一次暂存
git grep "delete from"                          # 文件中搜索文本“delete from”
git grep -e '#define' --and -e SORT_DIRENT
git gc
git fsck
```

git设置别名




```
git ci -am'add "feature" to the readme'

git log --oneline
git log --graph --abbrev-commit
git log --pretty=format:'%h %s' --graph
```

删除分支

Delete a Git Branch Locally and Remotely
baeldung.com

Useful Git Commands - Java Code Geeks - 2022
javacodegeeks.com