冰河亲自整理的Git命令汇总（建议收藏）

mp.weixin.qq.com/s/BojzCPOFRBktdCUtwX-Iww

大家好，我是冰河~~

Git目前是各大互联网公司使用的版本控制工具，进大厂，必须要学会Git的基本使用。这不，最近就有很多小伙伴私信我：冰河，可以帮我整理下Git的使用命令吗？网上的太零散了，买书看又没时间。我：可以啊！

于是乎，我熬夜整理了这篇文章。这篇文章主要是汇总讲解Git的使用命令。

冰河技术
分享各种编程语言、开发技术、分布式与微服务架构、分布式数据库、分布式事务、云原生、大数据与云计算技术和渗透技术。另外，还会分享各种面试题和面试技巧。

501篇原创内容

公众号

点击上方卡片关注我

Git的安装

通过 https://git-scm.com/downloads ，git官网下载需要的版本，一路下一步安装即可

装好后，在电脑文件夹的任意位置右键，即可看到git相关的命令。

git bash是命令行工具

git gui是图形化工具

打开git bash后，输入git --version ，能正确输出版本号，则证明安装成功。

基本命令

配置用户名和邮箱

```
git config --global [user.name](http://user.name) ‘自己的名字’
git config --global [user.name](http://user.name) ‘自己的邮箱’
```

- local 只对当前仓库有效
- global 所有仓库有效
- system 对系统所有用户有效

查看配置

```
git config --list --local
git config --list --global
git config --list --system
```

清除配置

```
git config --unset --local [user.name](http://user.name)
git config --unset --global [user.name](http://user.name)
git config --unset --system [user.name](http://user.name)
```

创建仓库

进入要被托管的文件夹，执行

```
git init
```

添加文件至暂存区

```
git add 文件名
```

提交文件

```
git commit -m '描述'
```

查看git状态

```
git status
```

查看修改内容

```
git diff 文件名
```

修改文件名字

```
git mv 原文件名    新文件名
```

查看日志

功能为查看日志
```
git log
```

查看日志，以单行显示
```
git log --pretty=oneline
```

功能为查看历史操作记录，比如回退版本后想要重返“未来”可以查看最新的提交版本
```
git reflog
```

通过可视化工具查看提交信息

```
gitk
```

版本回退

退回到上一个版本
```
git reset --hard head
```

当知道对应的版本号时，可以用这个命令，适用于回退和前往之前的新版本
```
git reset --hard 版本号
```

撤销操作

新版本git提示用该命令进行撤销
```
git restore 文件名
```

旧版本用此命令做撤销，新版本也可以用
```
git checkout – 文件名
```

如果已经add进暂存区
```
git restore --staged 文件名
```

新版本git用该命令此为旧版本git命令，新版本也可以用
```
git reset head 文件名
```

删除文件

```
git rm -f 文件名
```

使用分支

查看当前分支

```
git branch
```

创建dev分支并切换过去

-b表示创建并切换，相当于下面两条命令
```
git checkout -b dev
```

创建分支
```
git branch dev
```

切换分支
```
git checkout dev
```

注意：上面是老版本的命令，创建分支和撤销都用checkout容易分不清，因此新版本创建分支推荐用

```
switch
创建并切换到dev
git switch -c dev

直接切换到已有的dev分支
git switch dev
```

合并分支

将dev分支合并到当前分支，合并后会丢失原来分支的信息
```
git merge dev
```

合并dev到当前分支，–no-ff表示禁用fast forwad,之后查看日志时是可以看到已被删除分支的信息
```
git merge --no-ff -m “merge with no-ff” dev
```

删除分支

```
git branch -d dev
git branh -D dev
```

如果dev没有被合并过用大写 -D

查看分支合并情况

```
git log --graph --pretty=oneline --abbrev-commit
```

stash的使用（bug分支）

保存当前的工作现场
```
git stash
```

查看所有被保存的工作
```
git stash list
```

恢复并删除工作现场，等价于git stash apply + git stash drop
```
git stash pop
```

开发环境在dev分支下，bug修复是提交在master中，如何快速合并至dev下：转移至dev分支下，执行下面命令

```
git cherry-pick bug分支的提交版本号
```

远程克隆到本地

```
git clone 自己的git项目地址
```

如果是本地没有项目，从远程往下拉项目则是克隆

关联

```
git remote add origin 自己的git项目地址
```

如果本地先建好了项目，那么执行这个命令将本地仓库与远程仓库关联

拉取远程的更新

```
git pull
```

第一和远程关联上之后，在提交之前要先pull一下远程的更新才行

基本推送

```
第一次推送是要加上-u，可以把本地的master和远程的master关联起来，方便以后的推送或者拉取
git push -u origin master

之后推送可以直接用该命令
git push origin master
```

查看远程仓库信息

```
git remote

此命令可显示更详细信息
git remote -v
```

多人协作

- **git checkout -b 分支名 origin/分支名**，在本地创建和远程分支对应的分支，名称最好一致
- **git branch --set-upstream-to=origin/dev dev**，建立本地分支和远程分支的关联

- **git pull**，先抓取远程的更新，如果有冲突，手动解决冲突
- **git push origin 分支名**，解决冲突后推送

标签

基本操作

标签的作用可以简单理解为给版本起名字

```
查看所有标签
git tag

把当前分支的最新提交打上标签，标签名字自己起
git tag 标签名

把某个版本号的提交打上标签
git tag 标签名 对应commit版本号

可以用这种方式给标签增加说明，-a对应标签名，-m对应描述信息
git tag -a v0.1 -m “描述信息” 版本号

查看标签具体信息
git show 标签名

删除标签
git tag -d 标签名
```

推送标签

```
推送某个标签到远程
git push origin 标签名

推送所有标签到远程
git push origin --tags

删除远程标签：
先删除本地标签
git tag -d 标签名

 然后从远程删除
git push origin: refs/tags/标签名
```

好了，今天就到这儿吧，我是冰河，我们下期见~~

硬核专栏
推荐👍：《精通高并发系列》
推荐👍：《架构师进阶系列》
推荐👍：《一起进大厂系列》
推荐👍：《性能调优系列》
推荐👍：《深入理解JVM系列》

推荐👍：《精通分布式事务系列》
推荐👍：《Spring注解系列》
推荐👍：《吃透MySQL系列》
推荐👍：《Java8新特性》
推荐👍：《精通Nginx系列》

往期推荐

推荐👍：《发现一个超硬核学习宝藏！爱了！爱了！》

推荐👍：《实践出真知：全网最强秒杀系统架构解密！！》

推荐👍：《高并发分布式锁架构解密，不是所有的锁都是分布式锁！》

推荐👍：《这部电子书凭什么短短几个月全网累计下载突破16万？(目前已破45W+)》

推荐👍：《卧槽，冰河又写了一本电子书！！》

---END---

后台回复“并发编程”领取冰河原创的全网累计下载超45W+的《深入理解高并发编程》电子书。回复“渗透笔记”领取冰河原创的全网首个开源的以实战案例为背景的《冰河的渗透实战笔记》电子书。回复“PDF”领取冰河整理的其他8本超硬核PDF电子书，海量面试资料和简历模板。
冰河从一名普通程序员，短短几年时间，一路进阶成长为互联网高级技术专家，一直致力于分布式系统架构、微服务、分布式数据库、分布式事务与大数据技术的研究。在高并发、高可用、高可扩展性、高可维护性和大数据等领域拥有丰富的架构经验。对Hadoop，Storm，Spark，Flink等大数据框架源码进行过深度分析，并具有丰富的实战经验。
出版过三本畅销书《深入理解分布式事务：原理与实战》、《海量数据处理与大数据技术实战》、《MySQL技术大全：开发、优化与运维实战》。写了一本《深入理解高并发编程》电子书全网累计下载45W+，发布了一本全网首个开源的以实战案例为背景的《冰河的渗透实战笔记》电子书，全网五星好评。写的文章多次被微信公众号官方推荐。
扫一扫关注我

喜欢就点个 在看 呗 👇