



链滴

# Git 命令大全

作者: [Eddie](#)

原文链接: <https://ld246.com/article/1552557910900>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

## Git 命令大全

## 配置操作

### 全局配置

- 

git config --global user.name "xxx" # 配置用户名称

- 
- 

git config --global user.email "xxx@xxx.com" # 配置用户邮箱

- 
- 

git config --global --unset user.name # 取消全局配置的用户名称

- 
- 

git config --global --unset user.email #取消全局配置的邮箱地址

- 
- 

git config --global color.ui true # 配置命令行着色

- 
- 

git config --global core.editor \$(which vim) # 配置文本编辑器

- 
- 

git config --global merge.tool vimdiff # 配置差异分析工具

- 
- 

git config --global core.autocrlf true #提交时转换为 LF，检出时转换为 CRLF

- 
- 

git config --global core.safecrlf true #拒绝提交包含混合换行符的文件

- 
- 

git config --global core.whitespace trailing-space,space-before-tab #修正空白符

>

- 
- 

git config --global alias.&lt;\* git-command&gt; #为 \* git 命令创建一个快捷方式

>

- 
- 

git config --global core.quotepath off #配置在 Windows 下 ASCII 字符的文件名正

显示

- 
- 

git config --global --list # 查看全局配置信息

- 



### 局部配置

- 

git config --local core.filemode false # 忽略文件权限

- 
- 

git config --local --list # 查看局部配置信息



</ul>

</li>

</ul>

### 分支操作

<ul>

<li>

<p>git branch # 显示本地分支</p>

</li>

<li>

<p>git branch -a # 显示所有分支</p>

</li>

<li>

<p>git branch -D # 删除未曾合并过的分支</p>

</li>

<li>

<p>git branch -d # 删除已经合并过的分支</p>

</li>

<li>

<p>git branch -m oldName newName # 本地分支重命名</p>

</li>

<li>

<p>git push --delete origin # 删除远程分支</p>

</li>

<li>

<p>git branch --set-upstream-to origin/newName #&nbsp;把本地分支与远程分支关联起来</p>

</li>

<li>

<p>git branch -v #&nbsp;查看当前的本地分支与远程分支的关联关系</p>

</li>

<li>

<p>git diff origin/develop #&nbsp;查看本地当前分支与远程某一分支的差异</p>

</li>

<li>

<p>git diff master origin/master # 查看本地 master 分支与远程 master 分支的差异</p>

</li>

</ul>

### 合并操作

<ul>

<li>

<p>git merge origin/master # 将远程 master 分支合并到本地 master</p>

</li>

<li>

<p>git merge --no-ff develop # 将 develop 分支合并到当前分支（不使用 Fast-forward）</p>

</li>

<li>

<p>git cherry-pick # 将其它分支上的合适提交挑选到当前分支</p>

</li>

<li>git rebase -i HEAD~~~ # 汇合提交：将之前的三次提交合并到一处（squash）</li>

</ul>

</li>

<li>

<p>git rebase -i HEAD~3 # 修改提交（edit）</p>

</li>

```
</ul>
<h3 id="标签操作">标签操作</h3>
<ul>
<li>
<p>git tag # 查看所有标签</p>
</li>
<li>
<p>git tag -l 'v2.6.*' # 搜索某个标签</p>
</li>
<li>
<p>git tag -a v1.0.2 -m ": ) Project v1.0.2 Released" # 新建某个标签</p>
</li>
<li>
<p>git push origin v1.0.1 # 推送某个标签到远程</p>
</li>
<li>
<p>git push origin --tags # 推送所有标签到远程</p>
</li>
<li>
<p>git show v1.0.2 # 查看某个标签的版本信息</p>
</li>
<li>
<p>git tag -d v1.0.1 # 删除本地标签</p>
</li>
<li>
<p>git push origin --delete tag # 删除远程标签</p>
</li>
</ul>
<h3 id="回滚操作">回滚操作</h3>
<ul>
<li>git reset --hard # 将本地版本退回到某次提交上</li>
</ul>
<h3 id="本地文件回滚">本地文件回滚</h3>
<ul>
<li>
<p>git log filename.* * git reset filename.*</p>
</li>
<li>
<p>git commit -m "Rollback filename.*"</p>
</li>
<li>
<p>git checkout filename.*</p>
</li>
<li>
<p>git reset HEAD &lt; 文件名 &gt; #&nbsp;&nbsp;&nbsp;把暂存区的修改撤销掉，重新放回工作区</p>
</li>
<li>
<p>git revert HEAD # 撤销上一个提交</p>
</li>
<li>
<p>git reset --hard ORIG_HEAD #&nbsp;&nbsp;&nbsp;回退到上一次 reset 之前</p>
</li>
</ul>
<h3 id="日志操作">日志操作</h3>
```

- <ul>- <li>

<p>git log # 显示简略的提交日志</p>

</li>- <li>

<p>git log -- filename #&nbsp;&nbsp;&nbsp;查看某个文件变动的具体日志信息</p>

</li>- <li>

<p>git log -p -2 #&nbsp;&nbsp;&nbsp;查看最近两次更新的内容差异</p>

</li>- <li>

<p>git log --stat #&nbsp;&nbsp;&nbsp;显示简要的增改行数统计</p>

</li>- <li>

<p>git log -1 HEAD #&nbsp;&nbsp;&nbsp;显示最后一次提交信息</p>

</li>- <li>

<p>git log --pretty=oneline # 单行显示日志信息</p>

</li>- <li>

<p>git log --pretty=oneline --graph --abbrev-commit # 查看图文格式日志</p>

</li>- <li>

<p>git log --graph --oneline --decorate --all #&nbsp;&nbsp;&nbsp;通过 ASCII 艺术的树形结构来展示所有的支，每个分支都标示了他的名字和标签</p>

</li>- <li>

<p>git log --name-status # 查看文件改变信息</p>

</li>- <li>

<p>git relog #&nbsp;&nbsp;&nbsp;显示所有提交，包括孤立节点</p>

</li></ul>

### <li> <p>git remote # 查看主要名</p> </li>- <li> <p>git remote -v #&nbsp;&nbsp;&nbsp;查看主机名即网址</p> </li>- <li> <p>git remote show &lt; 主机名 &gt; #&nbsp;&nbsp;&nbsp;查看主机的详细信息</p> </li>- <li> <p>git remote add &lt; 主机名 &gt; &lt; 地址 &gt; #&nbsp;&nbsp;&nbsp;添加远程主机</p> </li>- <li> <p>git remote rename &lt; 原主机名 &gt; &lt; 新主机名 &gt; #&nbsp;&nbsp;&nbsp;修改远程主机的别名</p> </li>- <li> <p>git remote rm &lt; 主机名 &gt; # 删除远程机器</p> </li></ul></li>

- git push \* gitee master # 将当前 \* gitee 分支推送到远程 master 分支
- git checkout -b origin/ # 检出远程分支到本地分支
- git remote rename master xiaohe # 重命名远程分支的名称
- git push origin --delete # 删除远程某个分支
- git remote update origin --prune # 更新远程分支列表
- git push -f #&nbsp;&nbsp;&nbsp;强制将本地推送到远程仓库