



链滴

# git 常用命令总结

作者: [ldk](#)

原文链接: <https://ld246.com/article/1515742642633>

来源网站: [链滴](#)

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

### 一、初始生成密钥，克隆项目：

生成 ssh 密钥，命令如下：

```
ssh-keygen -t rsa
```

将生成的公钥放在 github 账号 ssh 上之后。本地在之后即拥有改 github 账号对应仓库的相关限。

克隆项目到本地，命令如下：

```
git clone https://github.com/dksky/myFirstProject.git
```

### 二、全局配置：

初始在本地设置 github 账号的邮箱和用户名

```
git config --global user.email "you@example.com"
git config --global user.name "Your Name"
```

查看全局配置：

```
git config --global --list
```

查看仓库信息：

```
git remote show origin
```

### 三、将修改提交：

暂存：

```
git add *
```

查看状态：

```
git status
```

提交到本地仓库：

```
git commit -m "测试修改"
```

链接远端仓库（通常主远端仓库被称为 origin）：

```
git remote add origin https://github.com/goozgk/work.git
```

上传到服务器，git push 远端仓库名字(origin) 分支名字(master)：

```
git push origin master
```

### 四、更新：

更新本地仓库：

```
git fetch origin
```

更新代码：

```
cl">git pull
```

```
</span></span></code></pre>
```

### 五、分支: </h3>

<p>创建新分支</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">git branch &lt;bra1&gt;
```

```
</span></span></code></pre>
```

<p>切换到新分支</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">git checkout &lt;bra1&gt;
```

```
</span></span></code></pre>
```

<p>删除分支</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">git branch -d &lt;BranchName&gt;
```

```
</span></span></code></pre>
```

<p>查看当前分支</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">git branch
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">git show-branch
```

```
</span></span></code></pre>
```

<p>在提交 commit 之前, 切换到各个分支后, 新建的文件, 各个分支都可以看到的。 <br>

但是在某个分支上 commit 后, 只有这个分支可以看到。 </p>

<p>合并分支<br>

在分支上测试完成后, <code>git branch master</code> 切换回 master 分支, 执行</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">git merge &lt;branchName&gt;; git branch -d &lt;branchName&gt;
```

```
</span></span></code></pre>
```

### 六、回滚: </h3>

<p>查看日志: </p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">git log
```

```
</span></span></code></pre>
```

<p>add ,commit 以后发现代码有点问题, 可以本地回退代码到指定版本: </p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">git reset --hard 5bba88b303dcfe323e9454ce8b2280e44b0d13e5
```

```
</span></span></code></pre>
```

<p>上述命令中, 上一个版本就是 HEAD^, 上上一个版本就是 HEAD^^, 当然往上 100 个版本写 10 个 ^ 比较容易数不过来, 所以写成 HEAD~100</p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">git reset --hard HEAD~3
```

```
</span></span></code></pre>
```

<p>使用 git reset 是不影响远程分支的, 一切都在本地发生。如果回退需要很快影响远程分支, 本代码和线上代码同步回退, 应该使用 git revert。 </p>

```
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">git revert 5bba88b303dcfe323e9454ce8b2280e44b0d13e5
```

```
</span></span></code></pre>
```