



链滴

git 的撤销和回滚

作者: [fu123](#)

原文链接: <https://ld246.com/article/1526953898861>

来源网站: [链滴](#)

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

<p>1,commit 之前的撤销</p>
<p>未添加至暂存区的撤销(add 之前)

git status

git checkout .

已添加至暂存区的撤销(add 之后)</p>
<p>git reset HEAD :将已经添加至暂存区的文件,回退到未添加到暂存区的状态,

然后再 git checkout .,回退到最为原始的状态</p>
<p>2,commit 之后的撤销</p>
<p>

git add .</p>
<pre><code class="highlight-chroma">git commit -m "commit"

git push origin b
andname
</code></pre>
<p>

等一系列操作后的撤销</p>
<p>不想将本次的操作生效</p>
<p>git revert commitId(提交的 id)</p>
<p>git push origin branchname(分支名称)</p>
<p>此次操作就会撤销掉上一次的 push</p>
<p>在这个分支上,会生成一条 revert 的提交记录</p>
<p>revert 是负负得正,什么意思呢,就是说第一次 revert 的时候,上一次 push 的命令就失效了,再一次
evert 后,会回退到之前的之前的状态,</p>
<p>详细的介绍一下</p>
<p>比如我操作一个文件 UserController.java</p>
<p>将 userName 改为 username 经过一系列的 git 操作

git add .

git commit -m "commit"

git push origin

后,再进行 git revert commitId

再进行 git push origin branchname

userName 还是 userName,</p>
<p>如果我再 revert 的时候,

git revert

git push origin branchname

将 userName 还是改为 username.</p>
<p>总结为:</p>
<p>git revert 为奇数时生效,为偶数时,失效</p>
<p>3,git 回滚:</p>
<p>将 git 回滚到某一次 commit 操作,在该 commitId 之前的操作,

将全部失效(此方法不建议使用,有风险),</p>
<p>git reset --hard commitId</p>
<p>git push origin branchname --force (强制覆盖)</p>
<p>因为是不可逆的,不建议使用</p>