

Learn Version Control with Git

A step-by-step course for the complete beginner

[Table of Contents ▼](#)

Learn on: [Desktop GUI](#) | **Command Line** Language: [EN](#) | **CN**

处理合并冲突

对于很多人来说，合并时出现冲突是非常可怕的事，这就好像一不小心格式化了自己的硬盘一样。在这一章节里我将为你消除这种恐惧。

你不会把事情搞砸

首先你应该记住，你总是可以撤销一个合并操作，并且返回到冲突发生之前的状态。也就是说，你永远有机会放弃并重新开始。

如果你已经掌握了一些关于其它的版本控制系统的使用经验，例如 Subversion，你可能会很难过。因为在 Subversion 中处理冲突是被大家公认极为复杂而繁琐的。这也就是为什么我们要使用 Git 的原因。简单地说，它在这方面的工作原理是完全不同于 Subversion 的。Git 能够在合并过程中顾及到很多方方面面的东西，从而为你创造一个比较简单的方案来解决可能出现的冲突。

当然，冲突只会妨碍你自己的工作，它是不会涉及到整个团队的项目仓库。这是因为在 Git 中，冲突只可能发生在开发人员的本地计算机上，而不是在远程服务器上。

什么是一个合并冲突

在 Git 中，“合并（merging）”是在形式上整合别的分支到你当前的工作分支的操作。你需要得到在另外一个上下文背景下的改动（这就也就是我们所提到过的，一个有效的分支应该是建立在一个上下文工作背景上的），并且合并它们到你的当前的工作文件中来。

作为你的版本管理系统，Git 所带来的最伟大的改善就是它让合并操作变得非常轻松简单。在大多数情况下，Git 会自己弄清楚该如何整合这些新来的变化。

当然，也存在极少数的情况，你必须自己手动地告诉 Git 该怎么做。最为常见的就是大家都改动了同一个文件。即便在这种情况下，Git 还是有可能自动地发现并解决掉这些冲突。但是，如果两个人同时更改了同一个文件的同一行代码，或者一个人改动了那些被另一个人删除了的代码，Git 就不能简单地确定到底谁的改动才是正确的。这时 Git 会把这些地方标记为一个冲突，你必须首先解决掉这些冲突，然后再继续你的工作。

如何解决合并冲突

当面对一个合并冲突时，我们首先要搞明白发生了什么。例如是不是你和你的同事都同时编辑了同一个文件的同一行代码呢？是不是他删除了一个你正在编辑的文件呢？是不是你们同时添加了一个相同文件名的文件呢？

当你使用“git status”时，Git 会告诉你存在一个“未合并的路径（unmerged paths）”，这只是用另外一个方式告诉你，存在一个或多个冲突：

```
$ git status
# On branch contact-form
# You have unmerged paths.
#   (fix conflicts and run "git commit")
#
# Unmerged paths:
#   (use "git add <file>..." to mark resolution)
#
#       both modified:   contact.html
#
no changes added to commit (use "git add" and/or "git commit -a")
```

就让我们来深入地探讨一下，如何去解决这些最常见的冲突。

当两个改动发生在同一个文件的同一些行上，我们就要看看发生冲突的文件的内容了。Git 会非常友好地把文件中那些有问题的区域在 “<<<<<< HEAD” 和 “>>>>>> [other/branch/name]” 之间标记出来。

```
1 <<<<<< HEAD
2 This line was committed while working in the "login-box" branch.
3 =====
4 This line, in contrast, was committed while working in the "contact-form" branch.
5 >>>>>> refs/heads/contact-form
```

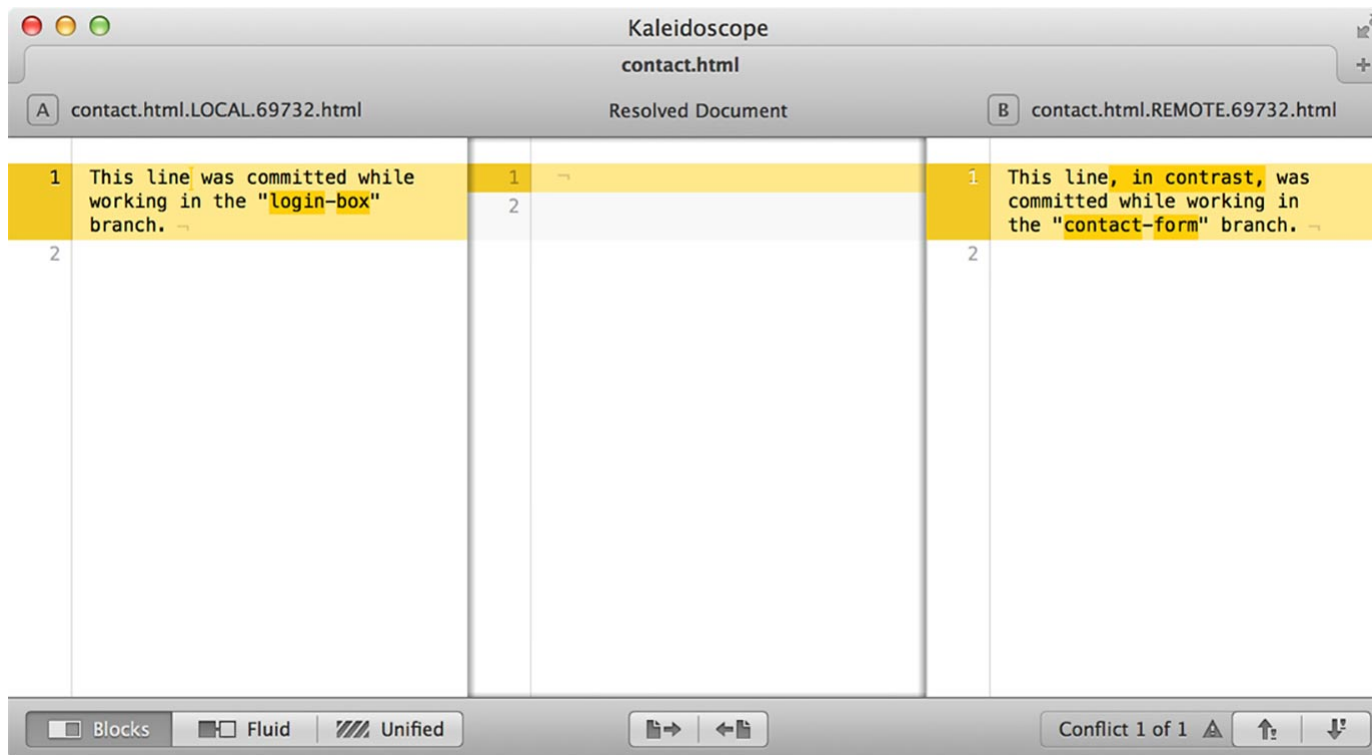
第一个标记后的内容源于当前分支。在尖括号之后，Git 会告诉我们这些改动是从哪里（哪个分支）来的。然后有冲突的改动会被 “=====” 分割起来。

现在我们的工作是要清理这些问题行。当我们完成这些清理后，这个文件应该看起来和我们预期的完全一样。在过程中你也可能需要咨询一下那个和你的代码发生冲突的同事，从而更好地决定哪些改动才是最终正确的，哪些改动是需要被放弃掉的。可能是你的改动，也可能是他的，或者可能是你们两个改动的组合。

打开一个比较原始的文件编辑器来清理这些冲突看起来是可行的，但是这样并不简单。使用一个专门的合并工具可以使这个操作变得更容易（如果你已经安装了一个在你的本地计算机上.....）。你可以通过 “git config” 命令来设置这个合并工具给 Git。更详细的内容你就要查看这个工具的文档说明了。

之后当发生合并冲突时，你可以使用 “git mergetool” 命令来调用这个工具。

例如，我在 Mac 上使用 “Kaleidoscope.app”：



在左边和右边的窗口会标记出那些改动的冲突。比起那些用符号“<<<<<<”和“>>>>>>”来标记冲突的方法来说，这是一个更加优雅的可视化环境。你可以非常方便地选择哪个改动是需要被保留的。位于中间的窗口会显示出处理后的结果，并且你也可以进一步手动编辑它。

现在，当清理文件并得到最终代码后，所有剩下的工作就是将这个结果保存起来，并且马上退出这个合并工具。这样 Git 就会知道你已经完成了这个操作。Git 会在后台对那个文件自动地执行“git add”命令。这也标志着冲突已经解决了。如果你不_使用合并工具，而是手动在文本编辑器中清理这些冲突，你必须手动地将文件标记为已解决状态（通过执行命令“git add <filename>”）。

最终，当所有的冲突被解决后，你必须通过一个正常的提交操作来完成这个清理合并冲突的工作。

如何撤销一个合并

你应该始终牢记，你可以在任何时间执行撤销操作，并返回到你开始合并之前的状态。要对自己有信心，你不会破坏项目中的任何东西。只要在命令行界面中键入“git merge --abort”命令，你的合并操作就会被安全的撤销。

当你解决完冲突，并且在合并完成后发现一个错误，你仍然还是有机会来简单地撤销它。你只须要键入“git reset --hard”命令，系统就会回滚到那个合并开始前的状态，然后重新开始吧！

[◀ Diffs - 检查改动](#)

[Contents](#)

[Rebase ▶](#)

Giveaways. Cheat Sheets. eBooks. Discounts.
And great content from our blog!

☐ Yes, I want the free newsletter that's loved by over 70,000 developers and designers. It's free, it's sent infrequently (approx. once a month) and you can unsubscribe any time.

☐ I have read and accept the [Privacy Policy](#). I understand that I can unsubscribe at any time.

Your email address

About Us



As the makers of [Tower](#), the best Git client for Mac and Windows, we help over 100,000 users in companies like Apple, Google, Amazon, Twitter, and Ebay get the most out of Git.

Just like with Tower, our mission with this platform is to help people become better professionals.

That's why we provide our guides, videos, and cheat sheets (about version control with Git and lots of other topics) for free.

