

Merging Conflicts in Git and Git Rebase

How to resolve Conflicts Git or How to resolve Conflicts Git Using Merge Tool.

- [Resolve a merge conflict with git pull](#)
- [Resolve a merge conflict locally with branches](#)
- [Resolve a merge conflict on Github interface with PR](#)
- [Git Merge vs Git Rebase](#)

Resolutions

- Open the file and resolve it manually (good when you have only a few lines of code with merge conflicts).
- Use a git merge tool like Diffmerge
- Use Vs Code
- Let say you want to merge the `test` branch into `master`, you need to checkout into the master branch first and hit `git merge test`. Always checkout into the branch that you want to merge code into it before merging the code.
- Undo in Vs Code is `ctrl+z`

How to quit merge conflict or cancel merge conflict

```
git merge --abort
```

Git merge Vs Git Rebase (The main difference is within the commit history)

In git, there 2 ways to integrate changes from one branch to another branch

- Git merge
- Git rebase

git merge

- Creates a new merge commit in the feature branch
- A new merge commit is required

git rebase

- Moves the entire feature branch to begin at the `end` of the master branch
- No merge commit is required
- Re-writes the project history
- We get a much cleaner and linear project history