Undoing changes

One of the most valuable features of Git is the ability to undo mistakes. When you make a new commit, Git stores a snapshot of your project so that you can go back to an earlier version when you need to.

There are two ways to undo changes: **git revert** and **git reset**.

Git revert

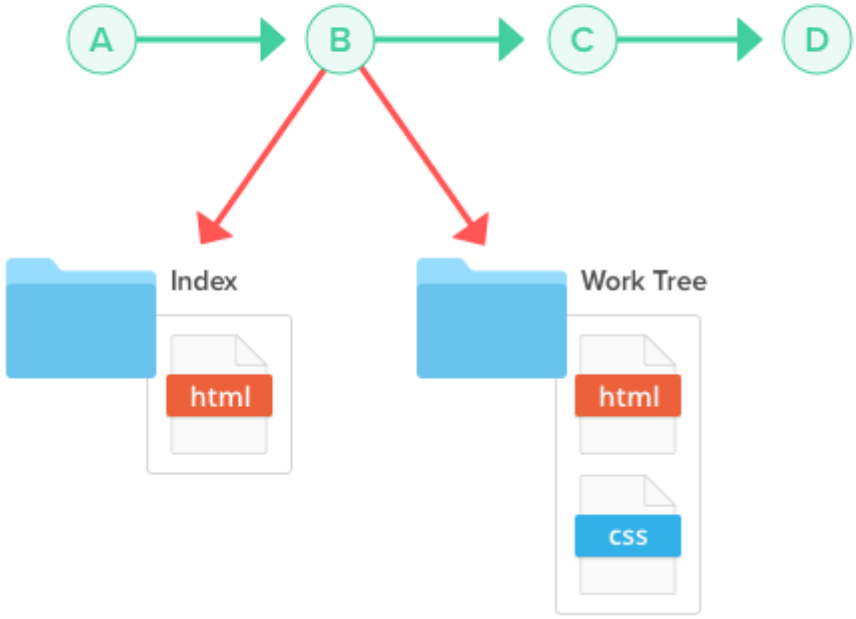
You can use the **git revert** command to safely undo a commit that has already been pushed.

While you can also delete a previous commit from the history using **git reset** or **git rebase -i**, it is generally not a good idea because it causes the remote repository to become desynchronized with the local repositories of other members.

git revert is the safest method of undoing changes.

Git reset

You can discard commits that you no longer need using the git reset command. You can specify the scope for the reset command by going into



Use git reset to remove unnecessary commits.

There are three primary reset modes:

- Mixed (default)
- Soft
- Hard

Mixed mode restores the state of a changed index. Soft mode undoes a previous commit. Hard mode removes all traces of a commit. Below is a breakdown of each reset mode.

Mode name	HEAD position (position of the HEAD)	Index	Work tree
soft	change	unchanged	unchanged
mixed	change	change	unchanged
hard	change	change	change

There are three reset modes: soft, mixed, and hard.