

To *Amending (change)* The Last Commit:

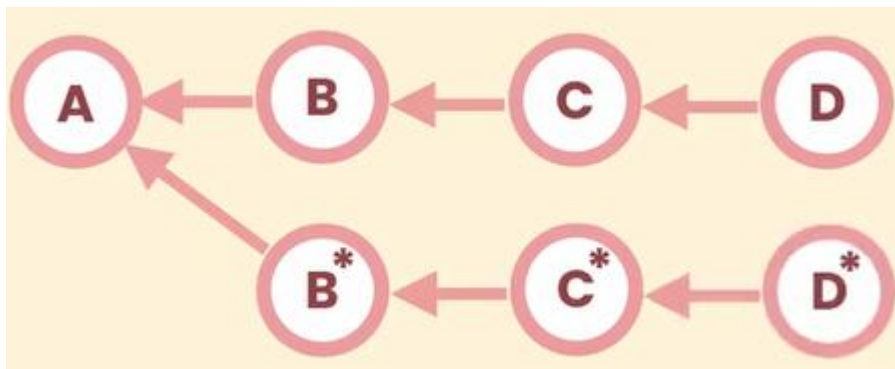
- We Can Also Use This Way To Make The Last Commit is The Current One.

```
~/Projects/Mercury git master +
> ✓ git commit --amend
[master 605f46c] Render cafes on the map.
Date: Thu Sep 10 15:25:08 2020 -0700
1 file changed, 1 insertion(+)
```

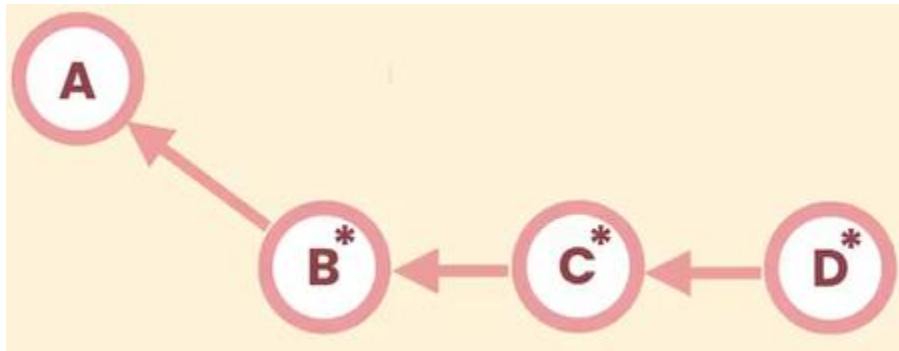
To Amend Earlier Commit:

```
~/Projects/Mercury git master
> ✓ git rebase -i 8527033
```

After We Interactive Rebasing The Commits, New Commits Will Be Added.



This Will Re-Create The History.



REBASING REWRITES HISTORY.

We Can Drop Commit Using: *git rebase -i*

This May Make Conflict If The Commit Make New File, Then The Second Commit Update The File.

If We Set ^ At The End Of Commit Id With *git rebase -i*, Then It Will Reference The Parent.

Here; Will Squash The Commits With The Previous One.

Also We Can Re-Order The Commits Then Squash Them.

```
git-rebase-todo
Users > moshfeghamedani > Projects > Mercury > .git > rebase-merge > git-rebase-todo
pick 8b134d2 Render restaurants on the map.
squash 24c6d63 Fix a typo.
squash 87d23b3 Change the color of restaurant icons.
pick 3d3ac4c Update terms of service and Google Map SDK version
pick 6f9aced Render cafes on the map.
```

With Rebasing Interactive; The fixup Will Remove The Commit And Not Ask For The Main One, It Will Squash It With Previous One:

```
pick 8b134d2 Render restaurants on the map.  
fixup 24c6d63 Fix a typo.  
fixup| 87d23b3 Change the color of restaurant icons.  
pick 3d3ac4c Update terms of service and Google Map SDK vers  
pick 6f9aced Render cafes on the map.
```
