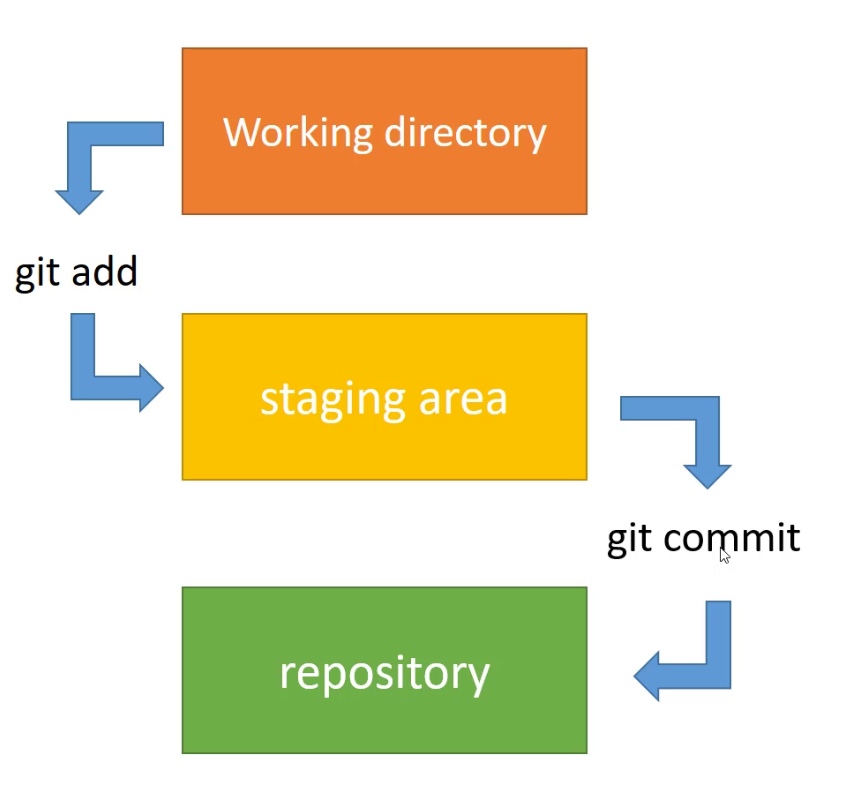
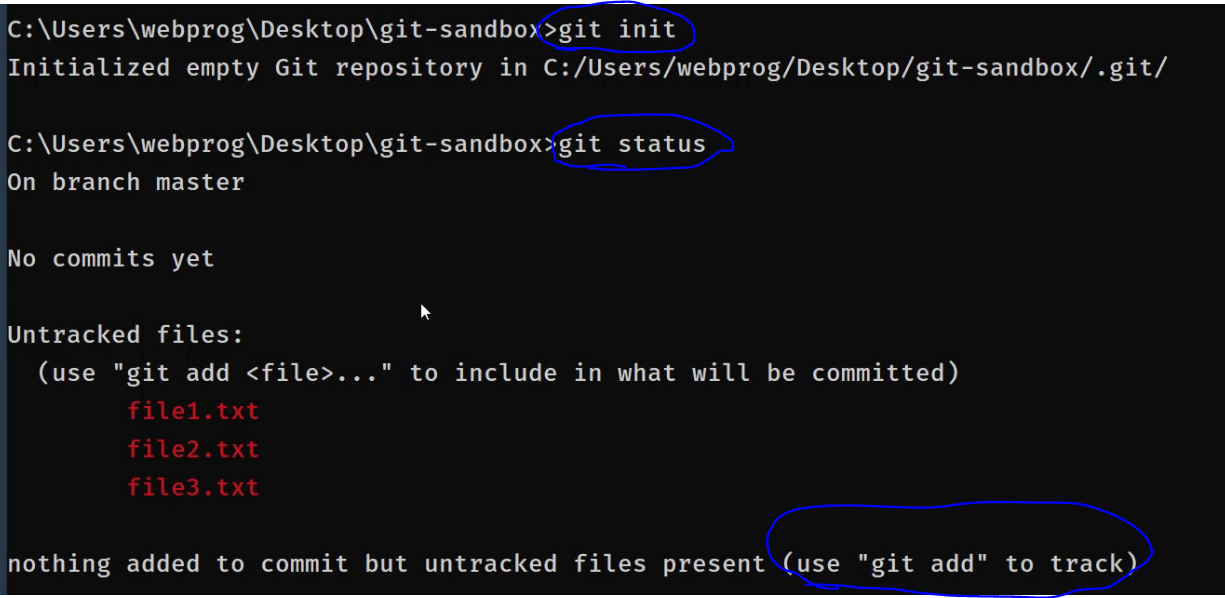
# Start



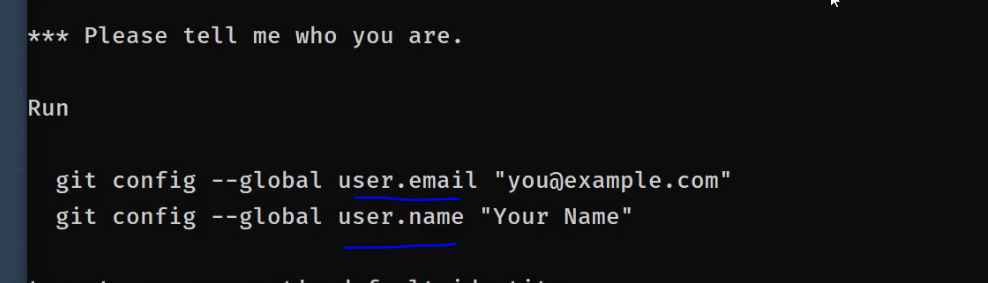


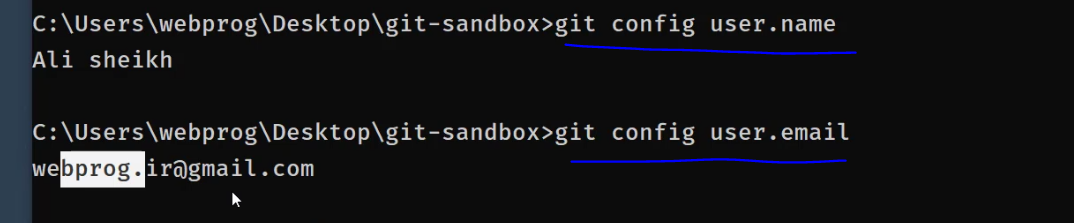
-when some files in stage is not ready to commit:





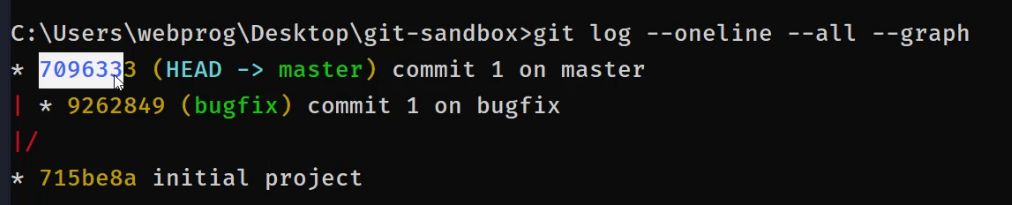
## author





## Log:

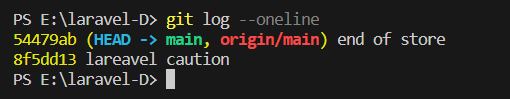
Note:HEAD is a pointer to where we are now.



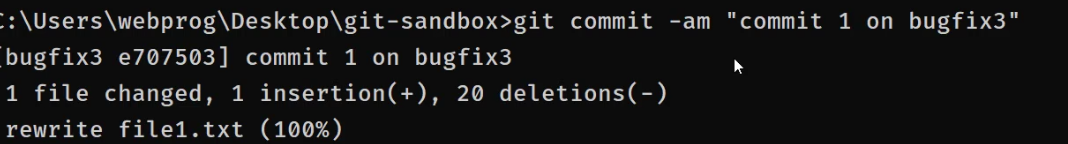




-it ignore date and author info:



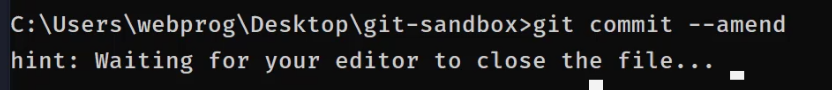
# Commits:



<<add .>> and commit at the same time

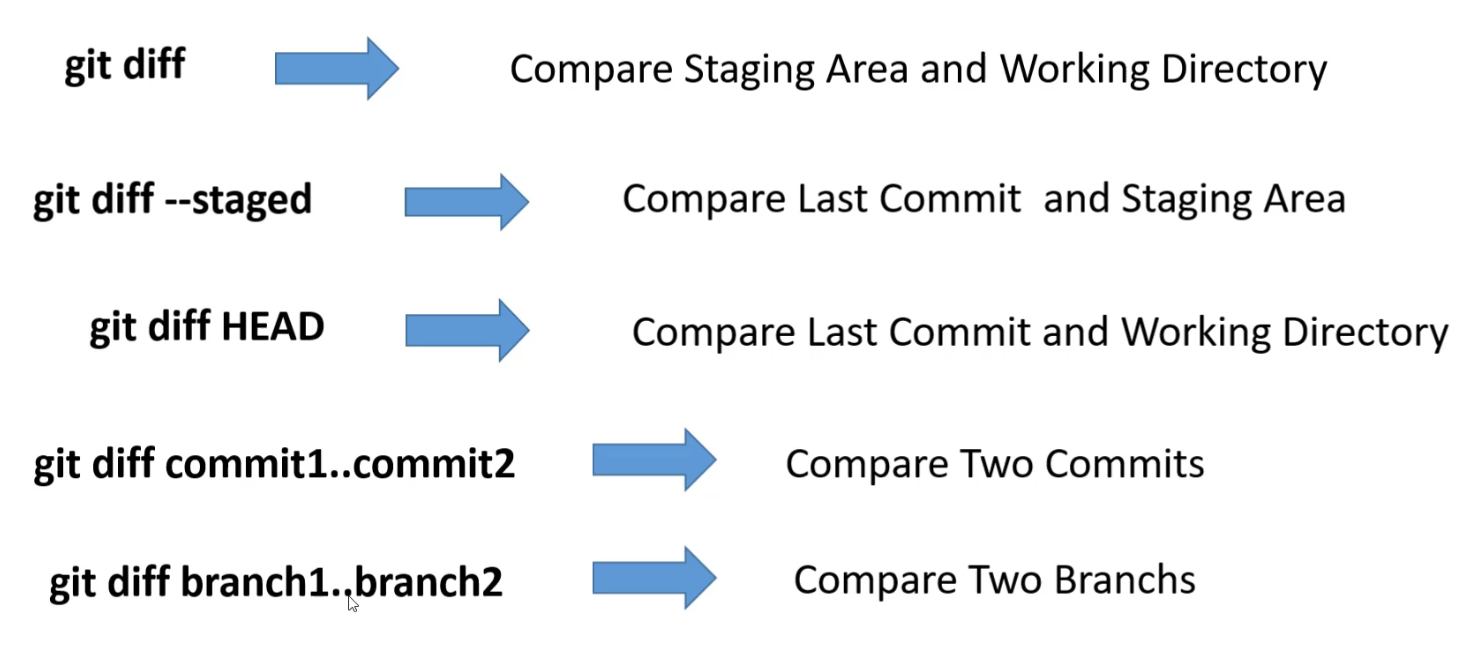
## (--amend):

When we have a commit and we want to add changes to the last commit, this will open page of commit in editor and we should do the changes that we want by editing the file.



## Diff

Note:we don’t use this command in terminal,for this action I rather vs code extentions or other apps.



## Go to a commit

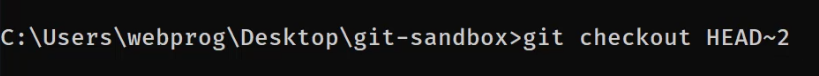
When?

* Create a new branch here and work on it.



After this run log command and see wher head is

Note-Simplest way to back to last commit of a branch is ,to use <git switch branchName> for bring HEAD to last commit.



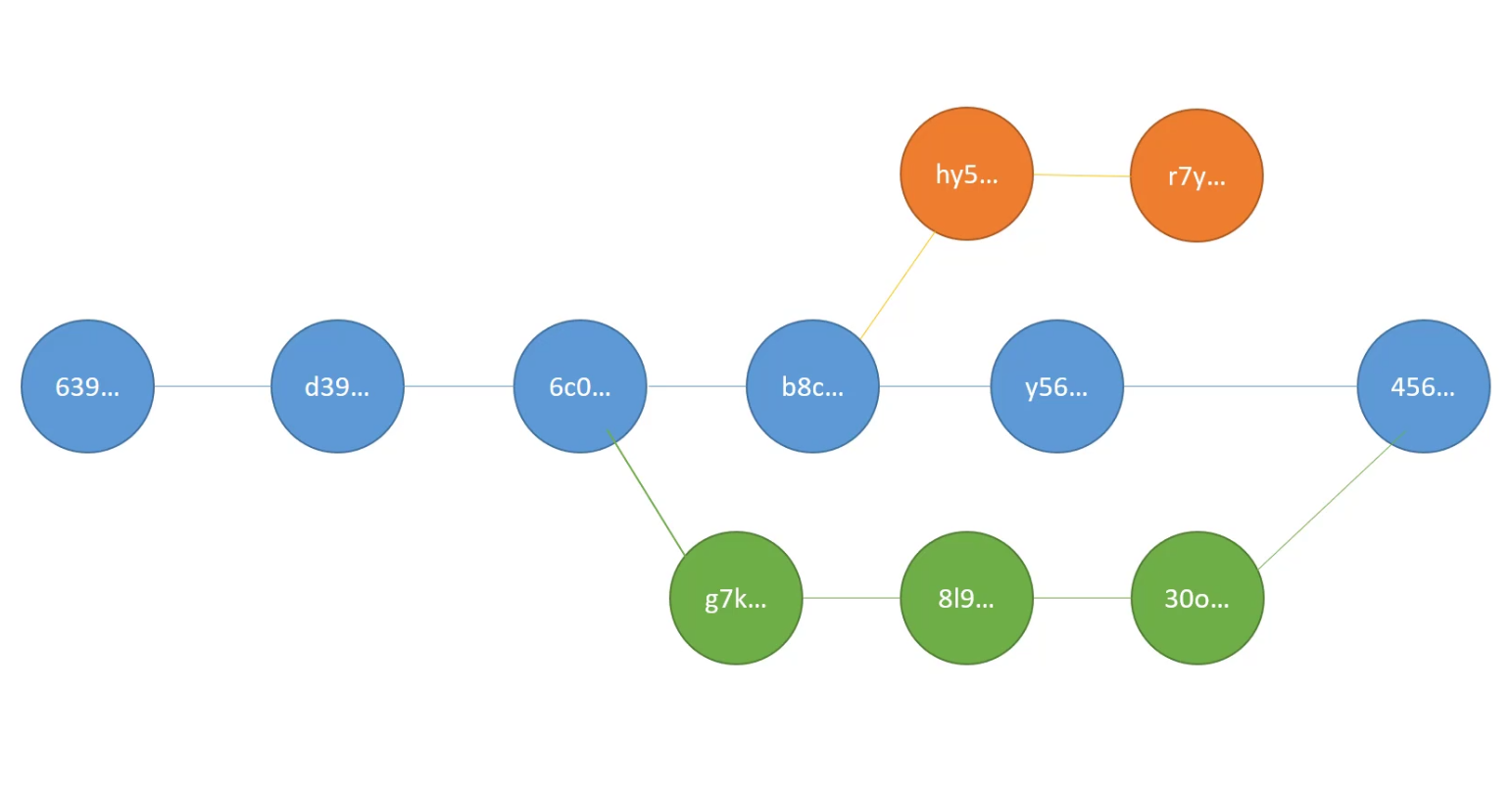
Head go down for 2 commit

## Delete:

Reset: <git reset 5k65345> traits: this command remove all commits until mentioned commit but it’s a soft one and bring codes of removed commits to workspace and even currnt workspace stay.

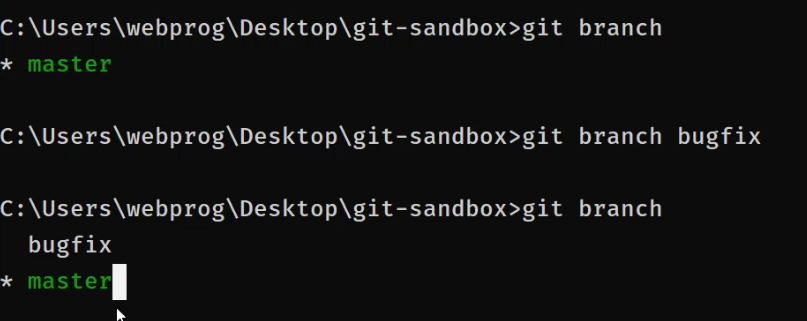
<git reset --hard 5k65345> traits: remove commits cods and workspace

# Branches



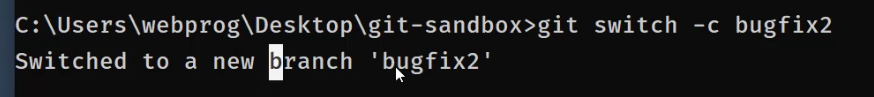
When we use branches:

* Green:when we have a bug and the main branch is working on a feature we gave the user the green codes and after finishing featue on main brance we merge green to main.
* Orange: Add a library to project for test,if it was good we merge it if not just ingore it like above.



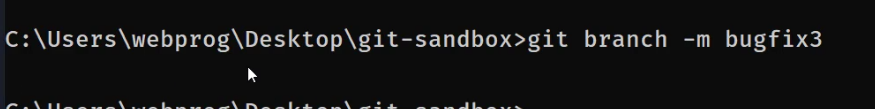




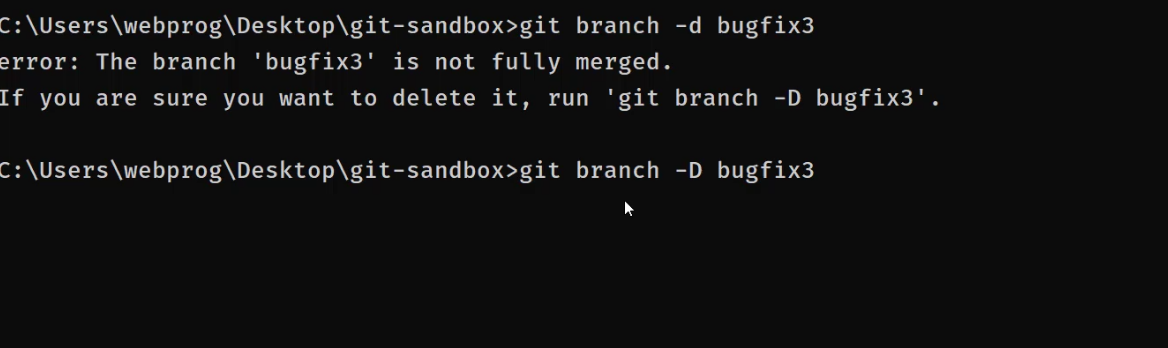




Swithc after create branch



Change name of the current branch to bugfix3





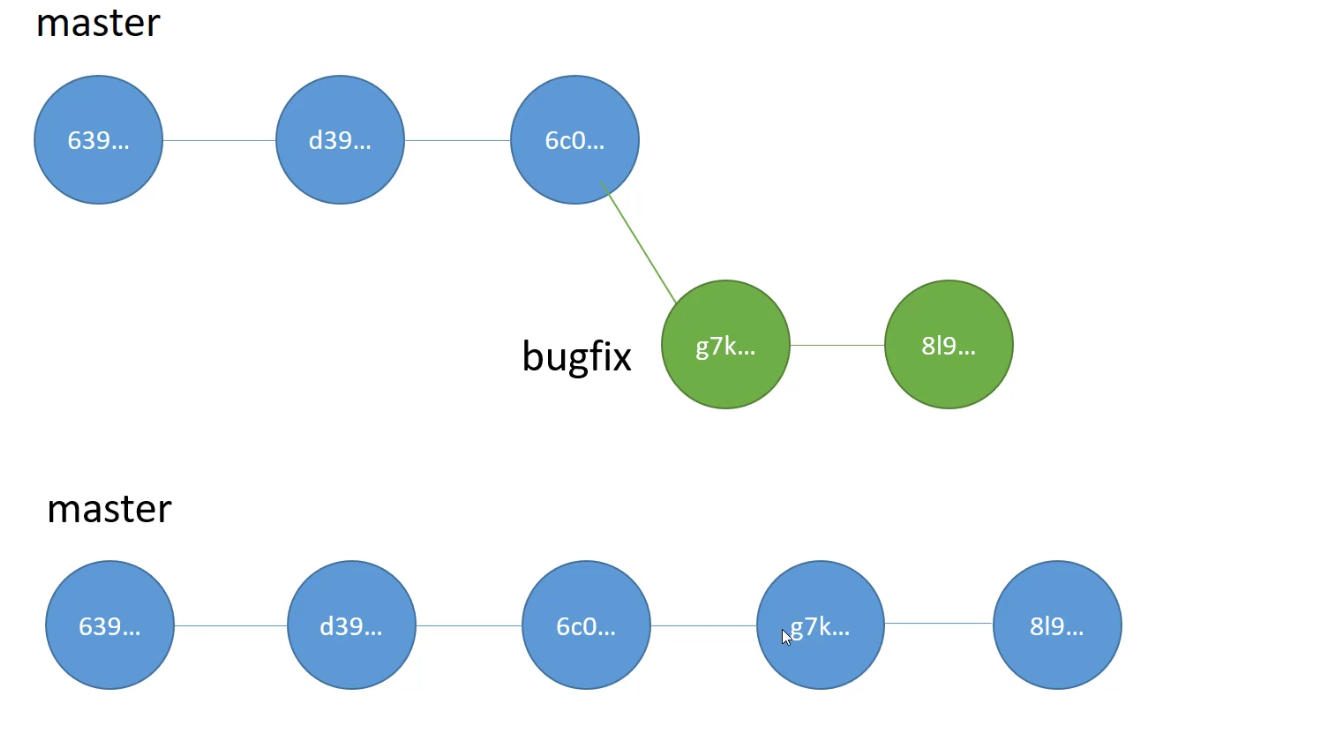
-D will force for times that we dident merge branch to main

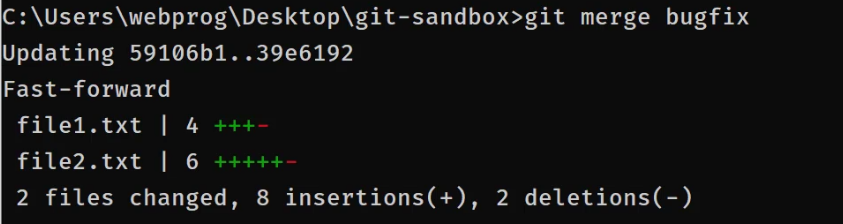
We cant delete branch that we are on it

# Merge

## Fast forward:

It’s the simplest way of merge:

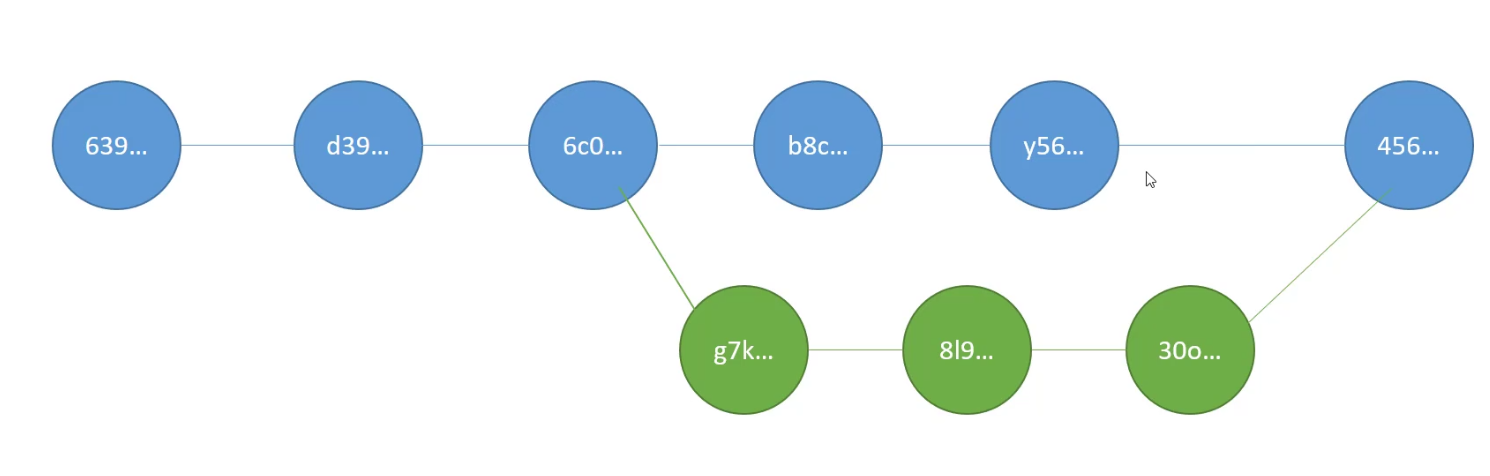






We must be on branch(main) that we want bugfix come and merge in it

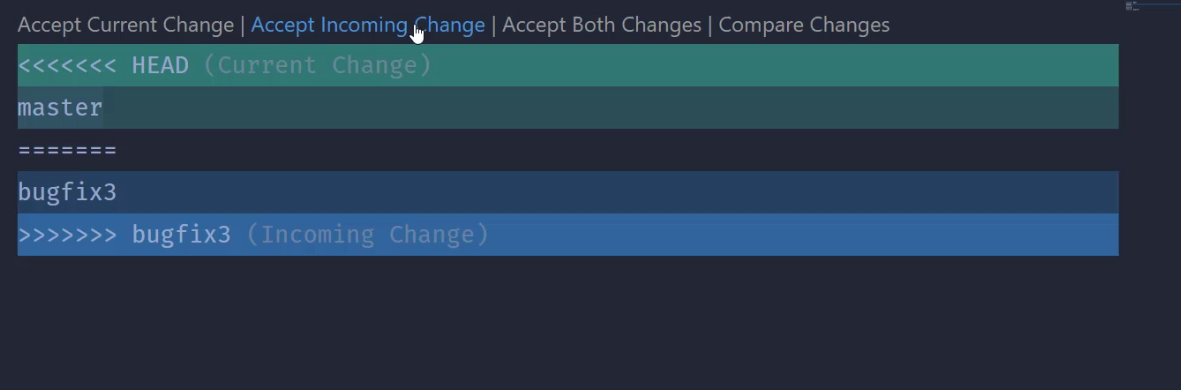
## Real merge:



In here we have new commit

Withot conflict: this happened when in two brnches, we didint modify two same file. In this situation nothing wont happen and its kinda like fast forward but we have a new commit.

With conflict: when we have conflict git will dump green branch in main(current) branch so in files that we have confilict it will show the changes after we decide about conflict we make a new commit and this commit will known as merge of grenn branch in main branch.



We can use suggested buttons or just remove signes and make commit

## Merge last commit with a past commit on a branch:

<git revert 8rheht88>:after this command use hints to proceed.

# Actions on workspace and stage

## Restore:

Delete from workspace:<git restore file.txt>

Delete from stage and brimg to work space:<git restore –staged file.txt>

## Stash

When?

* In middle of work on somting we need to do a bug fix and our working directory contain a feature but not ready to commit with this command we store workspace and we can code and fix bug and after commit bugfix we can bring back stash
* When working on a branch you can switch if there is conflict in workspace.
* Do stash<git stash>=<git stash save “you can have a name”
* Back stash and remove stash: <git stash pop>
* Dump stash without removing it> <git stash apply> by this we can go to another branch and run this and edditor will open and we fix conflicts.

To have multiple stash? Use <git stash save “you can have a name” .

<git stash list>

<git stash apply stash@{2}>=we can find id of stash in list command.

<git stash drop stash@{2}>=remove stash

<git stash clear>=remove all stashes

# Communicate Github

As we determine our email in local git in the time that we need to connect to our account in repository

## Remote

It’ defines a url desitination for a specific repository.

* <git remote>=list of remotes
* <git remote -v>=list of remotes with url
* <git remote add name url>=define a remote.(origin is best name beacaus its known by git and git hub as main remote)

## Push

Push is for send to github

* <git push *remote* *branch*>

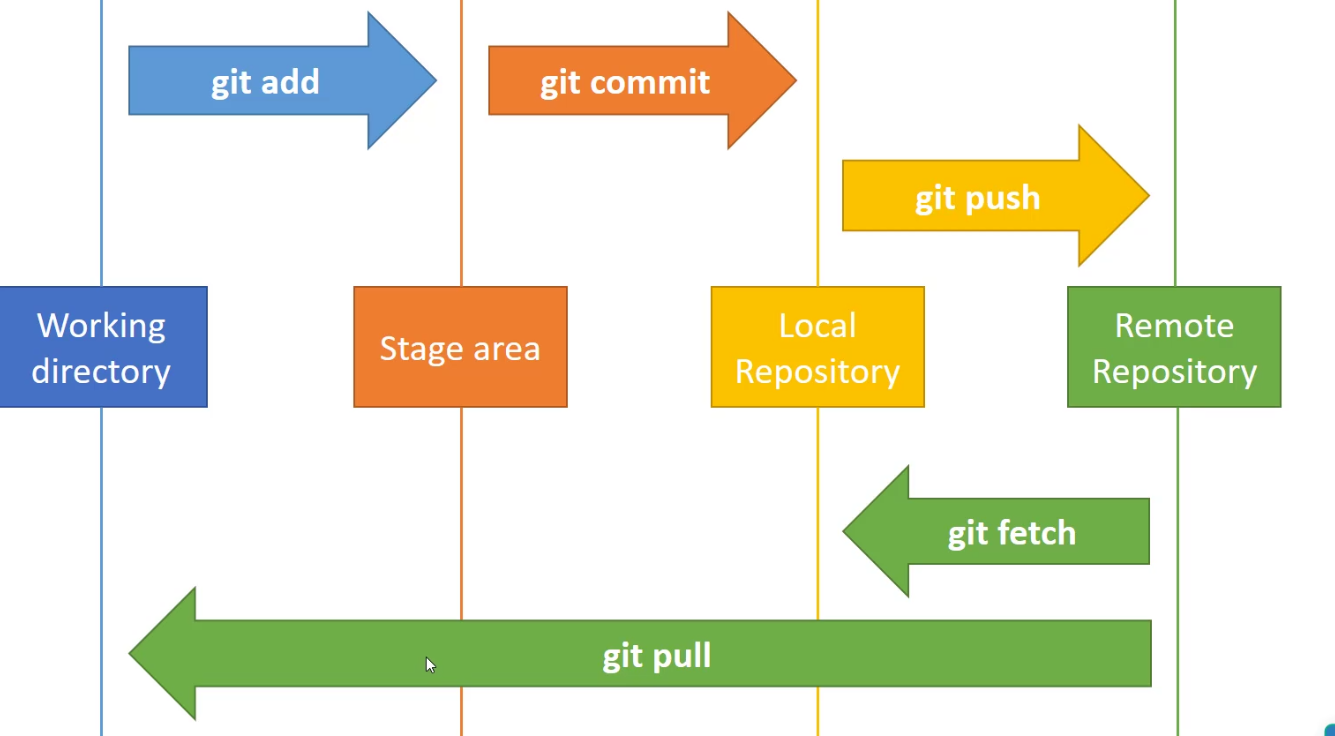
## Github pages:

In settings of a repository we can demand a url to show a static website about that repository.

Best way:create a branch contaion a index.html file.(name: felan-pages)

## Fetch and pull

Pull check for the conflict but fetch just bring commits.



<git fetch <remote> <branch> >

<git pull <remote> <branch> >=it dose tow step first fetch then merge last commits of local and github, if we have conflict we fix conflict and then create a new commit.

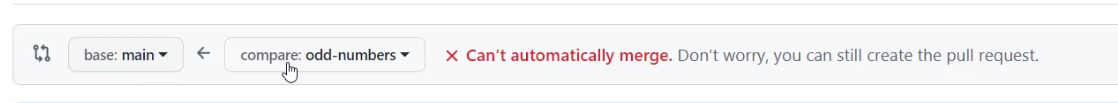
Tip- when git hub commits are ahead

## Pull request (episode 42)

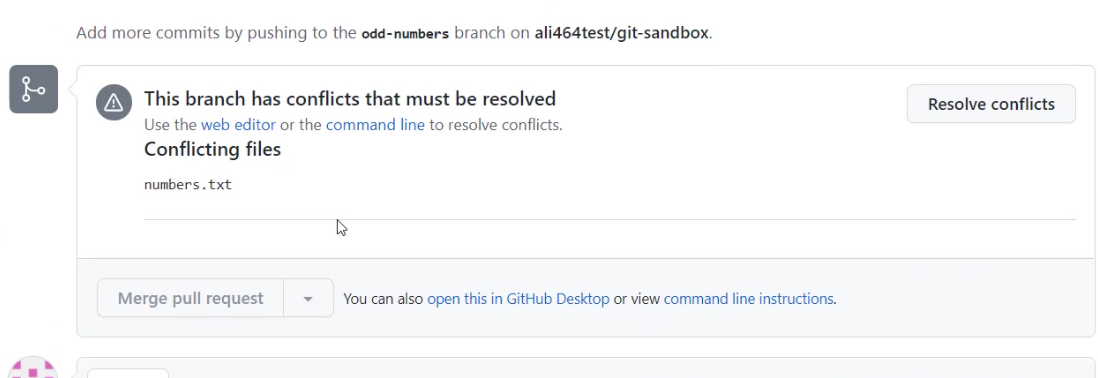
We shouldn’t push a commit to a main branch on a repository host, we create a branch on local and after finish we push the branch.

Tip- first time for pushing a branch we should run push command compeletly (with remote and branch and even -u flag)

Send a pull request:

1. Go to pull request tab
2. Click <new pull request> button
3. Choose a branch or a commit
4. After step 3 click <create pull request>

pull request demand:

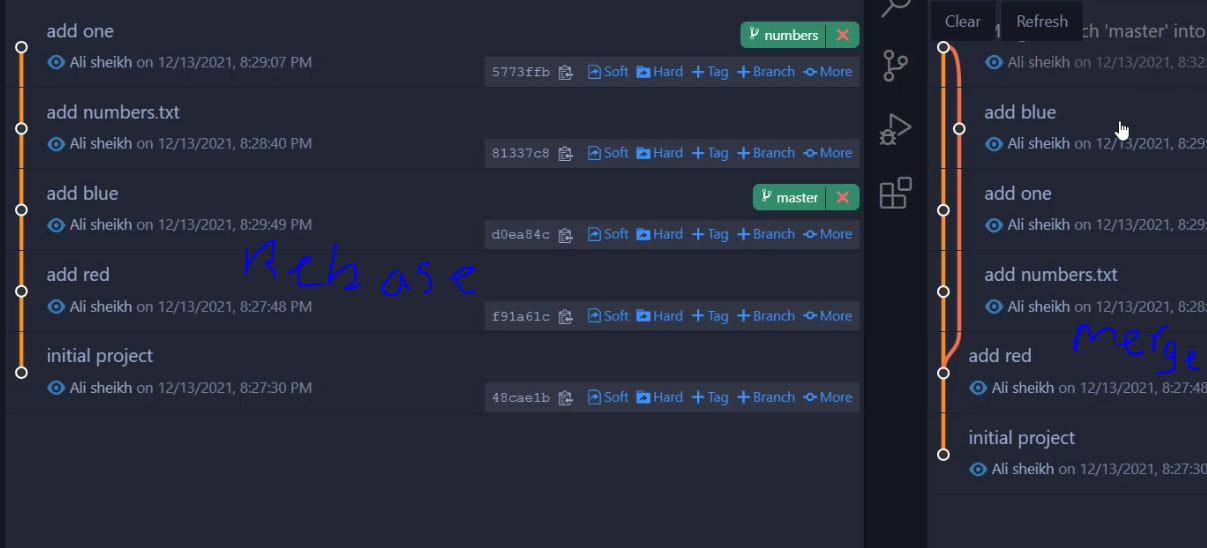




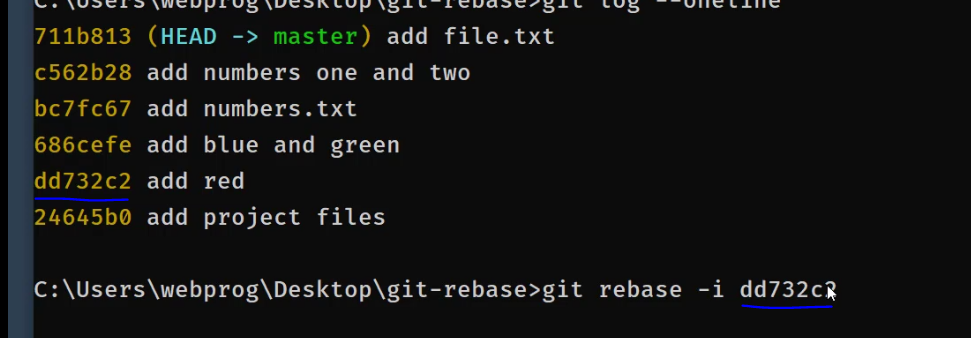
# Rebase

## Merge:

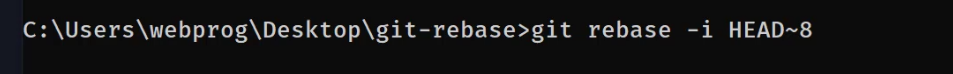
its not a good way for teem work:



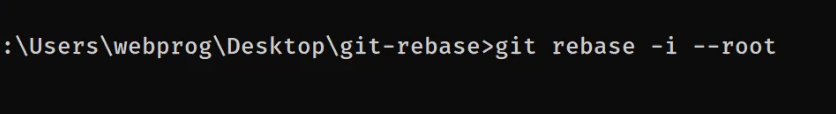
## Commits



It will bring last commit till mentioned hash



It brings 8 last commit



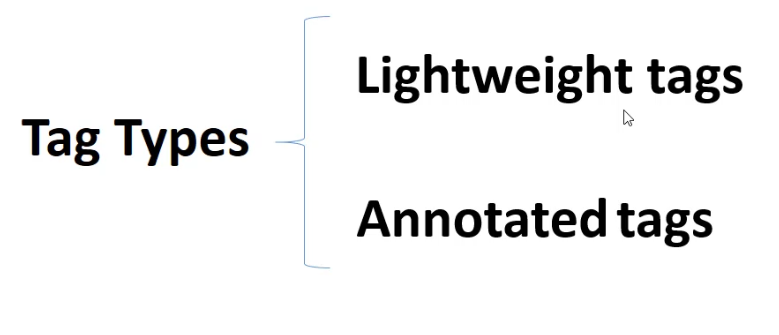
Open a file of commits,there is hint in bottom of it.

* After above code we change the pick word before any commit that we want and close file
* Another file will open and prceed

# Tags

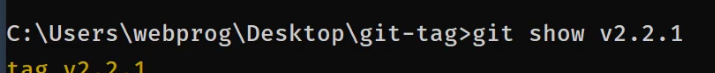
It’s common for versien sorting.

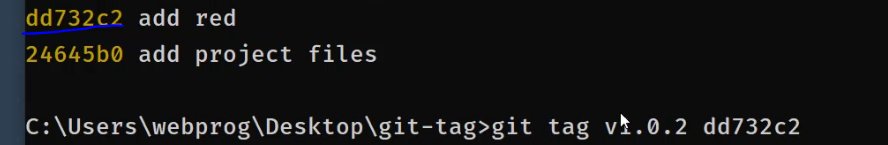
* <git tag v.2.5>= determin a tag for currnt HEAD
* <git tag>= list of tags

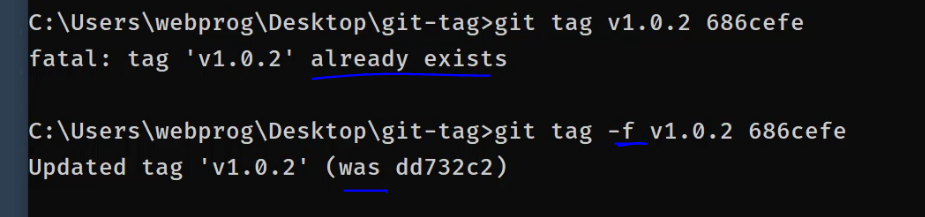


It contain a message after runing above code

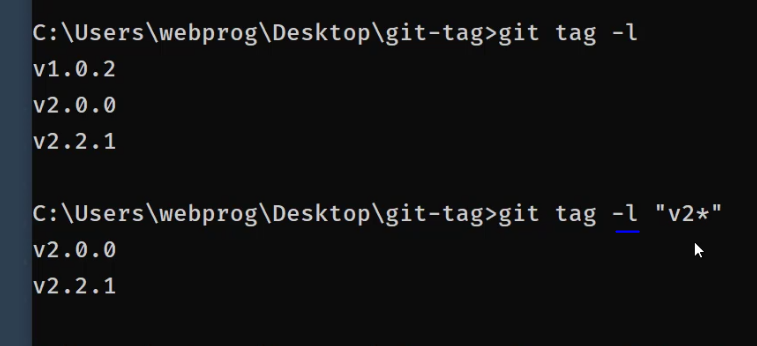
Git tag -a <TagName>





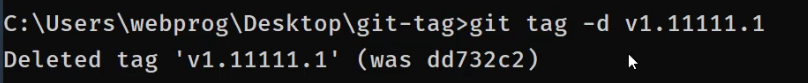


Search in tags





Bring commits before equal this tag.for back to normal use switch command





Push tag in repository:

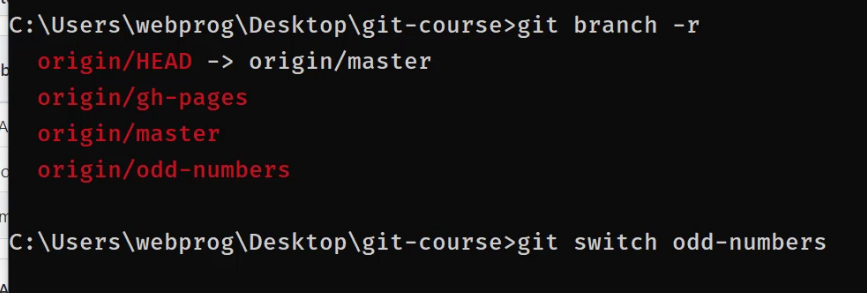


# Quastions

1-how to get a specific branch from a cloned repository?

2- How to add partner?

1)



2)

go to repository>settings>mange access