

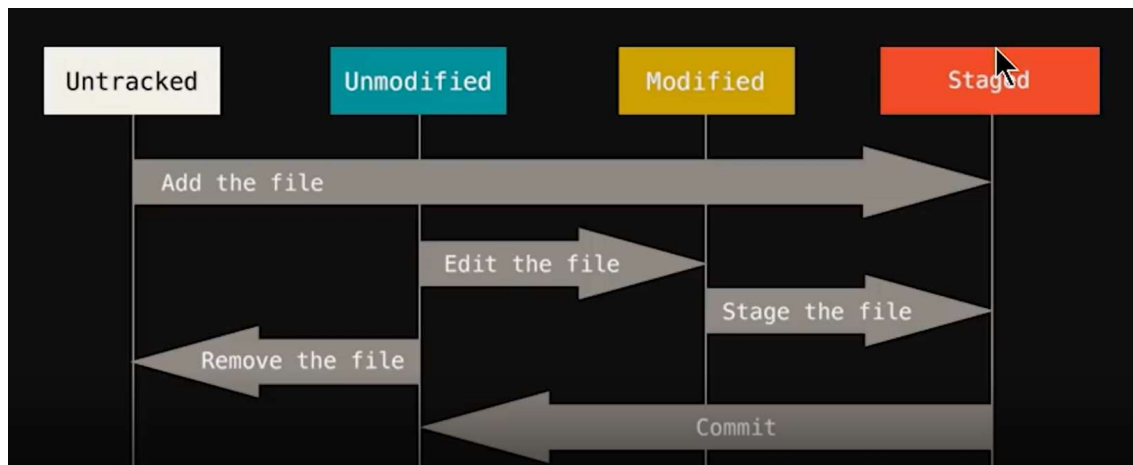
1. `git config --global user.name ashwinjuwekar2018`

`git config --global user.email ajuwekar@gmail.com`

`git init`

`ls -la` (shows all the hidden files)

`git status` (shows the untracked or untouched files)



Once you run `add` command that means it goes into the staging area and those file that you don't add that will not go to the staging area neither they will go on bitbucket

`Git checkout contact.html` means it restores the last status it is just like `ctrl+z`

`Git checkout -f` (for several files all files at a time)

How many commits you want to see for this there is a command

`Git log -p -1` (1 is the number of commit you want to see)

Suppose I made any changes in `about.html` like champak

Then if you type `git diff` then it will show how the new file is different than previous but you should not type

`git add -A`

`git checkout -f` (for many files the become like previous)

`git commit -a -m "commit without staging"`

in this you don't need to add in staging area

suppose you created a file waste.html and commit it and now you want it to uncommit then

```
git rm --cached waste.html
```

```
touch .gitignore
```

a file of gitignore will be created and then you have to write all those files names in this file to which you don't want to be staged

*****8888888888

How to import a repository from git

```
git init
```

```
git clone https://github.com/ashwinji/myfirstrepo.git
```

view all the commits in one list

```
Git add filename
```

```
Git commit -m 'message my first commit'
```

```
Git push -u origin master
```

```
Git log Or git log --oneline
```

If this error comes : unable to auto-detect email address

```
git config --global user.email "you@domain.com"
```

```
git config --global user.name "github_username"
```

now if you want to merge the firstbranch data into the master then first go to master branch

```
git checkout master (we come in master)
```

then

```
git merge firstbranch (this is local done and the data from firstbranch comes from and pasted in the master)
```

```
git push -u origin master
```

```
$ git remote add origin remote repository URL
```

```
# Sets the new remote
```

```
$ git remote -v
```

```
# Verifies the new remote URL
```

```
$ git push origin master OR git push -f origin master
```

Pushes the changes in your local repository up to the remote repository you specified as the origin

If you have already created a branch on the github server then how to get the data from github to local computer

Git origin could have

If this type of error comes:

Another git process seems to be running in this repository, e.g. an editor opened by 'git commit'. Please make sure all processes are terminated then try again. If it still fails, a git process may have crashed in this repository earlier: remove the file manually to continue.

Then

```
rm -f .git/index.lock
```

Please move or remove them before you switch branches.

Aborting

```
git checkout -f pooja(branchname)
```

Create project in www.gitlab.com (you can use same id and password of github so no need to create another account)

```
Git commit -am 'a'
```

```
git checkout master
```

```
git branch new-branch-to-save-current-commits
```

```
git fetch --all // it will fetch everything from server
```

```
git reset --hard origin/master //it will overwrite everything from the server to the local files
```

when something is written in the server and something written in the local directory then do one thing

```
git stash => it is saved somewhere
```

after that git pull

after that git stash pop

after that see the files and resolve the conflicts

and then

git add .

git commit -m 'a'

git push -u origin master

Or you can use the rebase command also written in blue color in this document

How to create ssh key for bitbucket

Open git bash

```
ssh-keygen -t rsa -C "ajjuwekar@gmail.com"
```

passphrase : 123456

retype passphrase : 123456

After that paste it in the bitbucket add key procedure.

Bitbucket Best and new tutorials in English Hindi mixed version

First of all install the git bash

Create a folder wherever you want

Now open git inside that folder

Now do one thing

Git config - -global user.name ashwinjuwekar2018

Git config - -global user.email ajjuwekar@gmail.com

Now run the command

Git init

Ls -lart (to view all the files)

By default you are in master which is the main branch

Now if you create inbox.html it will be in master

Git add -A OR git add index.html (it will show you in green colour that is staging area)

Git commit -m "initial commit"

The files you create now that will be in master by default

Git add -A (all files will be added)

Suppose somebody has opened the contact.html file and he has done some changes or deleted some important code and then he saved the file and shut down the computer now what to do when you opened the computer and everything will be lost , now how to recover the previous data.

Git checkout contact.html (it will restore your last committed data)

मान लो किसी ने 25 – 30 Files बिगाड़ दी होती तो ।

Git checkout -f (all the files will be restored to last committed data)

Git log (it shows you that whatever you have committed so far)

Suppose you want to see the last 5 commits details then

Git log -p -5 (the number can be variable)

Git diff and git diff -- staged

(it will show you the details of what you have deleted and what you have added with + and – sign)

Git checkout -f (will bring all your data to last commit version even if you have executed git add . then also you can restore to your previous commit)

Git commit -a -m waste.html (here the file waste.html will automatically be added and committed you don't need to run two separate commands)

Now suppose you want to delete an unwanted file and it is committed by you now

`Git rm --cached waste.html` (it will remove it from staging area or we can say it will undo the commit command and it will be shown in green colour)

`Git rm waste.html` (now it will remove the complete file from your harddisk also)

`Git status -s` (it will show you which file is modified with colored M like below)

M contact.html

M index.html

One more thing is there suppose there are some files which you don't want to commit for example there is an image folder each time when you commit something or push something then that image folder will also be committed or pushed so if you don't want some files or folder to be committed or pushed then following commands will be useful :

Go to terminal and type

Touch .gitignore (this command is compulsory in git command prompt)

Touch mylogs.log

Inside .gitignore file you to have to write

Now you are having mylogs files in two places one is the place where .gitignore is situated and the other is inside the folder

.gitignore

Logsfolder/mylogs.log

Mylogs.log

Now if you write inside the gitignore file like this : `mylogs.log` (that means ignore from everwhre)

And if you write inside the gitignore file like this : `/mylogs.log` (that means mylogs will be ignored only from there where the gitignore file is situated and apart from that all the mylogs.log inside any folder or directory they will be considered)

And if I write like this `*.log` that means ignore from wherever you see this file

If you want to ignore any folder then : `logsfolder/` (and the complete folder will be ignored)

(पूरे प्रॉजेक्ट में कहीं पर भी mylogs.log file मिलेगी उसे ignore कर देगी और वो add, commit, push नहीं होगी ।)

Now lets talk about checkout and branches

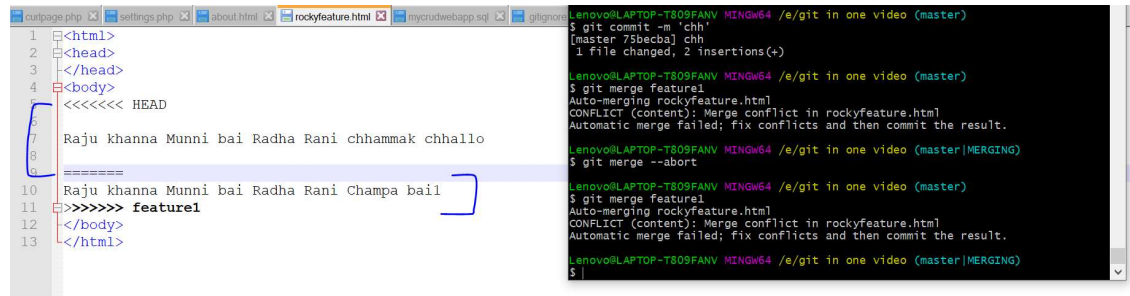
Creating a new branch : git branch feature1

Switching the branch : git checkout feature1

Listing of branches : git branch

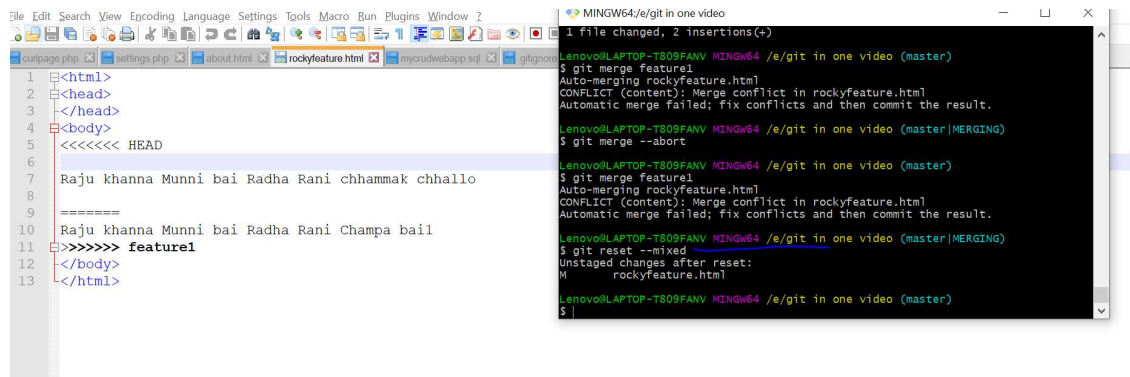
Scenerio 1:

Many times it happens that master file is updated and feature1 (branch) file is changed and when we want to merge the complete branch to master branch then it shows the following error :



One way is this where you can see that which one is from head and which one is from the feature1

Another way is `git reset --mixed` or git reset in this also save above picture will be seen



`Git merge -abort` will undo the merge command or abort

the merge command

One more command is also there that if you want to merge only one file then

`Git checkout feature1 rockyfeature.html` (then only one file will be forcefully replace our data in to the master/rockyfeature.html)file but you have to add . and commit it.

And once you committed the changes in your master file and after that you find it useless then you can run the following command

`Git revert HEAD`

Now suppose if I want to merge only one file of feature1(branch) in master(branch) then

[Git checkout feature1 about.html](#)

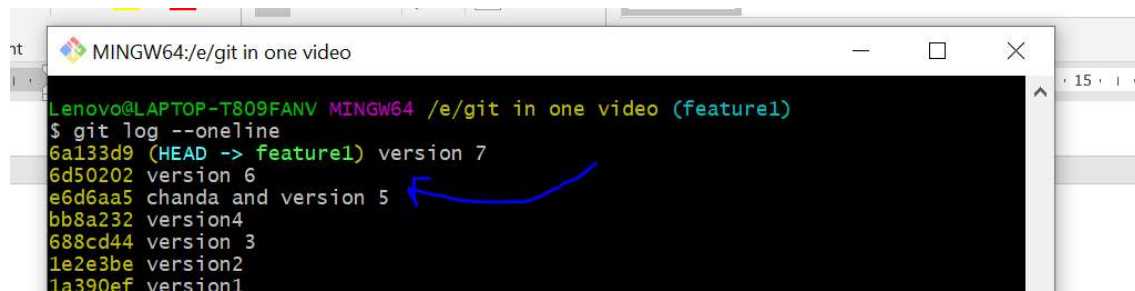
In this way only one file will be merged

मान लो पता चला कि मैंने 4 commit पहले की फाइल जो थी वो ही बेहतर बनाई थी वर्तमान की फाइल तो बेकार बनी है । तो 4 commit पहले की फाइल को ही restore करना है तो क्या करें ।

Restore the file of the last 4th commit or you want to restore the file of 2nd commit and this was your 5th commit then what to do:

Git log --oneline

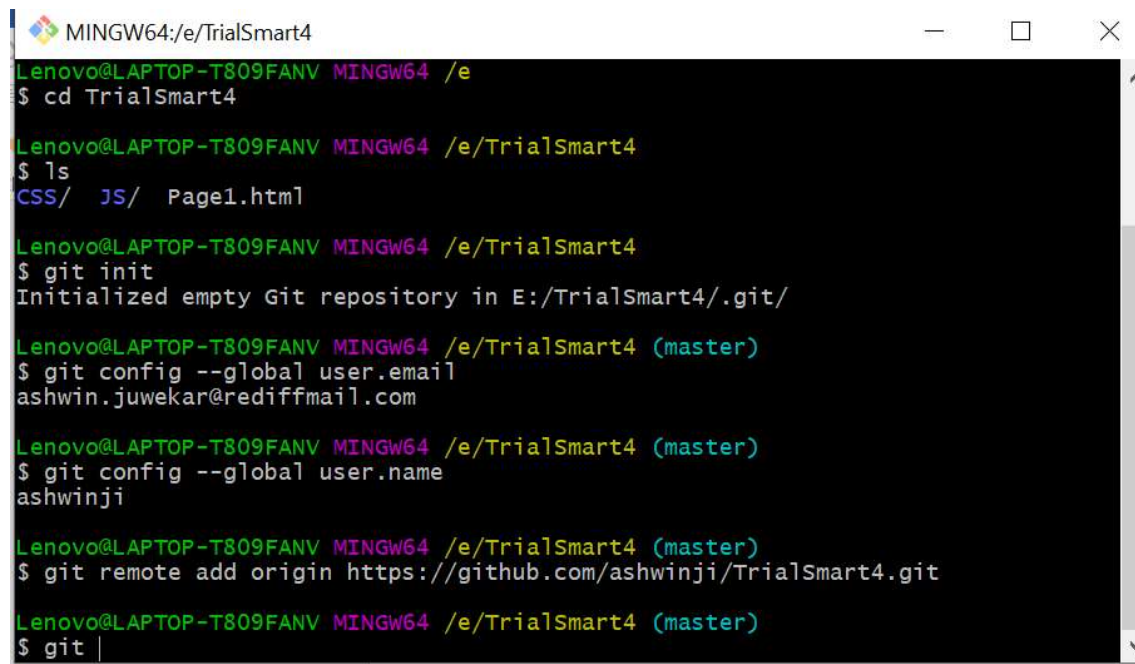
Git checkout e6d6aa5 about.html (in this way only the file will be restored)



```
Lenovo@LAPTOP-T809FANV MINGW64 /e/git in one video (feature1)
$ git log --oneline
6a133d9 (HEAD -> feature1) version 7
6d50202 version 6
e6d6aa5 chanda and version 5
bb8a232 version4
688cd44 version 3
1e2e3be version2
1a390ef version1
```

Setting with Github

First create repository and then do this



```
MINGW64:/e/TrialSmart4
Lenovo@LAPTOP-T809FANV MINGW64 /e
$ cd TrialSmart4

Lenovo@LAPTOP-T809FANV MINGW64 /e/TrialSmart4
$ ls
CSS/ JS/ Page1.html

Lenovo@LAPTOP-T809FANV MINGW64 /e/TrialSmart4
$ git init
Initialized empty Git repository in E:/TrialSmart4/.git/

Lenovo@LAPTOP-T809FANV MINGW64 /e/TrialSmart4 (master)
$ git config --global user.email
ashwin.juwekar@rediffmail.com

Lenovo@LAPTOP-T809FANV MINGW64 /e/TrialSmart4 (master)
$ git config --global user.name
ashwinji

Lenovo@LAPTOP-T809FANV MINGW64 /e/TrialSmart4 (master)
$ git remote add origin https://github.com/ashwinji/TrialSmart4.git

Lenovo@LAPTOP-T809FANV MINGW64 /e/TrialSmart4 (master)
$ git |
```

Now type git remote then you will see origin

```
MINGW64:/e/TrialSmart4
$ ls
CSS/ JS/ Page1.html

Lenovo@LAPTOP-T809FANV MINGW64 /e/TrialSmart4
$ git init
Initialized empty Git repository in E:/TrialSmart4/.git/

Lenovo@LAPTOP-T809FANV MINGW64 /e/TrialSmart4 (master)
$ git config --global user.email
ashwin.juwekar@rediffmail.com

Lenovo@LAPTOP-T809FANV MINGW64 /e/TrialSmart4 (master)
$ git config --global user.name
ashwinji

Lenovo@LAPTOP-T809FANV MINGW64 /e/TrialSmart4 (master)
$ git remote add origin https://github.com/ashwinji/TrialSmart4.git

Lenovo@LAPTOP-T809FANV MINGW64 /e/TrialSmart4 (master)
$ git remote
origin

Lenovo@LAPTOP-T809FANV MINGW64 /e/TrialSmart4 (master)
$
```

Git remote add origin <https://github.com/ashwinji/>

Now at this point if you type git push origin master it will show an error

```
MINGW64:/e/TrialSmart4
$ git push origin master
error: src refspec master does not match any
error: failed to push some refs to 'https://github.com/ashwinji/TrialSmart4.git'

Lenovo@LAPTOP-T809FANV MINGW64 /e/TrialSmart4 (master)
$ git remote
origin

Lenovo@LAPTOP-T809FANV MINGW64 /e/TrialSmart4 (master)
$ git push origin master
error: src refspec master does not match any
error: failed to push some refs to 'https://github.com/ashwinji/TrialSmart4.git'

Lenovo@LAPTOP-T809FANV MINGW64 /e/TrialSmart4 (master)
$ ls
CSS/ JS/ Page1.html

Lenovo@LAPTOP-T809FANV MINGW64 /e/TrialSmart4 (master)
$ git push origin master
error: src refspec master does not match any
error: failed to push some refs to 'https://github.com/ashwinji/TrialSmart4.git'

Lenovo@LAPTOP-T809FANV MINGW64 /e/TrialSmart4 (master)
$
```

ssh-keygen -t rsa -b 4096 -C ashwin.juwekar@rediffmail.com
after that run this eval command

```

SHA256:h3L5Wls1IV0tDsuJJGAeS1RFuRV2MaouC2MIGQewNB0 ashwin.juwekar@rediffmail.com
The key's randomart image is:
+---[RSA 4096]-----+
|
| o o
| o B o o
| * . o . o
| + . . So o = .
| . . . o o . + o
| o . . . .
| . o o o o
|
+---[SHA256]-----+
Lenovo@LAPTOP-T809FANV MINGW64 /e/TrialSmart4 (master)
$ eval $(ssh-agent -s)
Agent pid 1883
Lenovo@LAPTOP-T809FANV MINGW64 /e/TrialSmart4 (master)

```

Then run the following command

```

+---[SHA256]-----+
Lenovo@LAPTOP-T809FANV MINGW64 /e/TrialSmart4 (master)
$ eval $(ssh-agent -s)
Agent pid 1883
Lenovo@LAPTOP-T809FANV MINGW64 /e/TrialSmart4 (master)
$ ssh-add ~/.ssh/id_rsa
Identity added: /c/Users/Lenovo/.ssh/id_rsa (ashwin.juwekar@rediffmail.com)
Lenovo@LAPTOP-T809FANV MINGW64 /e/TrialSmart4 (master)

```

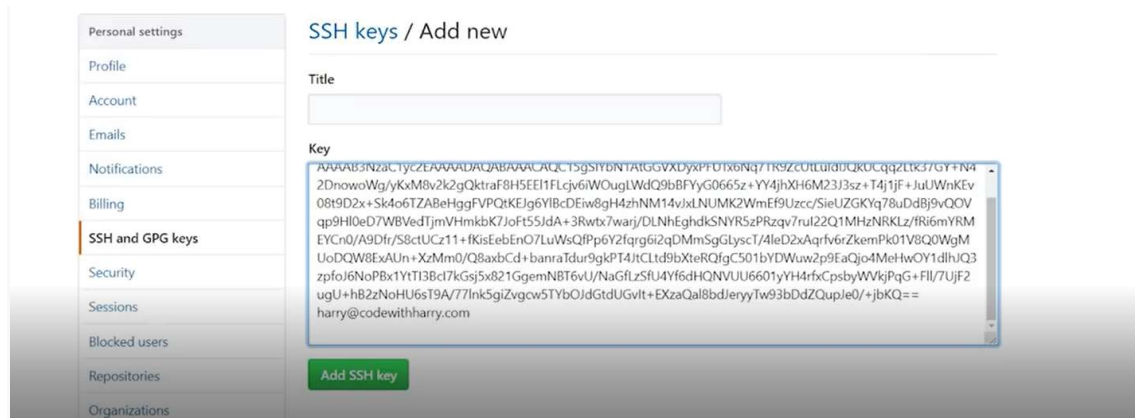
Now you have to register the ssh key in your project now do this

```

MINGW64:/e/TrialSmart4
Lenovo@LAPTOP-T809FANV MINGW64 /e/TrialSmart4 (master)
$ eval $(ssh-agent -s)
Agent pid 1883
Lenovo@LAPTOP-T809FANV MINGW64 /e/TrialSmart4 (master)
$ ssh-add ~/.ssh/id_rsa
Identity added: /c/Users/Lenovo/.ssh/id_rsa (ashwin.juwekar@rediffmail.com)
Lenovo@LAPTOP-T809FANV MINGW64 /e/TrialSmart4 (master)
$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAQC64KqG3/yTyKQquACxP9QrVb1bDQXXtTQXVMe0ptQVfDvVZdvETL
7qDdFIUsadLhRjKZT315bWcIdQU4RUZNPc7TmZv9eSYvQNVX30zr4aMgsqICPggY61006CRLiwaNZ0PIbfcsKfKvX6
AYhfOD/q5UgIwB4m1yyqOmKVPuGE3sK8c3wKYEnmzoofTqap0ZN8No+Ev08ThvvuBf04nft3tz1jVeNuqIERyKJ4+C
yRTSujMaE0WQ2F4zce82MDPLosxx3zMjEUJfuxgh1lDis2cSaxAC0a/+IgsRtZJc0l/gVrTXQ7wGrZRZV6mXicIUmY
6C8+d1rMYi6qROMq1JJJEZGzQmRzv4Ur00lHYRr4hsFccsdYJSB08rcET1U5M1jvzd14KtcFhB0BiQIokf0F0c7g1yK
cRxs8jzDu6Ggsp5HFnjRK1Xf+SVnrIdHpBMsEJbcCHAEe52/jKwgZfM2VFdyZWjOb0Ifyvw+TjGN649oTbx79QwjG/
uwr/w5pqgX56Nt4EgtCWFRk/w9Rv8PbQ9P+Z/629t9psfpBSz3bJC9UrsWowFr51vBfv3H6/NtGoN6Xxobfa6sqjMh
6oi0cygBcedwLR+yMSOYm9JI9jNYjhoNw6m71aIMTLOHXSkuswRVAVXvs1dgvMsiFsLS9Mzdb/mQ78o16ruVJwV5FN
Yw== ashwin.juwekar@rediffmail.com
Lenovo@LAPTOP-T809FANV MINGW64 /e/TrialSmart4 (master)
$ git push origin master
error: src refspec master does not match any

```

Copy this key in the ssh key



And now you are ready to push anything

Many times you see this type of **error** :

Another git process seems to be running in this repository, e.g. an editor opened by 'git commit'. Please make sure all processes are terminated then try again. If it still fails, a git process may have crashed in this repository earlier:
remove the file manually to continue

then you have to run the following command : `rm -f .git/index.lock`

Very Important command : `git remote add origin git@github.com:ashwinji/leadanalytics4.git`

Deleting the branch locally

Deleting a branch LOCALLY

Delete a branch with `git branch -d <branch>` . The `-d` option will **delete** the **branch** only if it has already been pushed and merged with the remote **branch**. Use `-D` instead if you want to force the **branch** to be deleted, even if it hasn't been pushed or merged yet. The **branch** is now deleted **locally**.

Delete a server branch

```
git push origin --delete test
```

28-05-2020

Take today's incidence that I have done in pooja's branch in main.css in server and then pooja also done some changes in her local branch pooja same file that is main.css now when she tries to commit then it is creating problem then how to solve the problem here is the solution.

Git pull --rebase origin master

Git mergetool (enter press 2 times in front of (vimdiff):

And then do the changes and after that :wq

Git rebase --continue

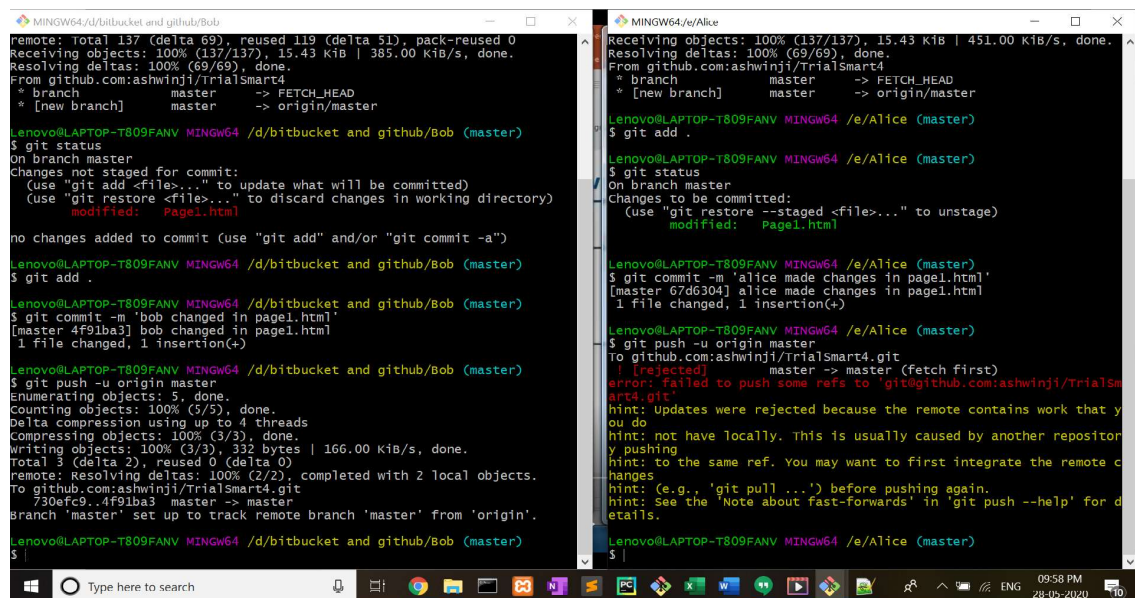
Git push -u origin master (don't add and commit again this time)

If you want to abort rebase then

Git rebase --abort

(git remote add origin **copied path from github**)

Example of Bob and Alice : Suppose bob and alice are working on same branch and the both are doing changes in that now bob has done changes and Alice has done changes now how alice will merge her code :



The screenshot shows two terminal windows side-by-side. The left window is titled 'MINGW64/d/bitbucket and github/Bob' and the right window is titled 'MINGW64/e/Alice'. Both windows show the same sequence of Git commands and output:

```
remote: Total 137 (delta 69), reused 119 (delta 51), pack-reused 0
Receiving objects: 100% (137/137), 15.43 KiB | 385.00 KiB/s, done.
Resolving deltas: 100% (69/69), done.
From github.com:ashwinji/TrialSmart4
* branch      master       -> FETCH_HEAD
* [new branch] master       -> origin/master

Lenovo@LAPTOP-T809FANV MINGW64 /d/bitbucket and github/Bob (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   Page1.html

no changes added to commit (use "git add" and/or "git commit -a")
$ git add .
Lenovo@LAPTOP-T809FANV MINGW64 /d/bitbucket and github/Bob (master)
$ git commit -m 'bob changed in page1.html'
[master 4f91ba3] bob changed in page1.html
1 file changed, 1 insertion(+)

Lenovo@LAPTOP-T809FANV MINGW64 /d/bitbucket and github/Bob (master)
$ git push -u origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 332 bytes | 166.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:ashwinji/TrialSmart4.git
   730efc9..4f91ba3 master -> master
branch 'master' set up to track remote branch 'master' from 'origin'.

Lenovo@LAPTOP-T809FANV MINGW64 /d/bitbucket and github/Bob (master)
$
```

The right window shows the same commands but with the following differences in output:

```
Receiving objects: 100% (137/137), 15.43 KiB | 451.00 KiB/s, done.
Resolving deltas: 100% (69/69), done.
From github.com:ashwinji/TrialSmart4
* branch      master       -> FETCH_HEAD
* [new branch] master       -> origin/master

Lenovo@LAPTOP-T809FANV MINGW64 /e/Alice (master)
$ git add .
Lenovo@LAPTOP-T809FANV MINGW64 /e/Alice (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   Page1.html

Lenovo@LAPTOP-T809FANV MINGW64 /e/Alice (master)
$ git commit -m 'alice made changes in page1.html'
[master 67d6304] alice made changes in page1.html
1 file changed, 1 insertion(+)

Lenovo@LAPTOP-T809FANV MINGW64 /e/Alice (master)
$ git push -u origin master
To github.com:ashwinji/TrialSmart4.git
 ! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'git@github.com:ashwinji/TrialSmart4.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

Lenovo@LAPTOP-T809FANV MINGW64 /e/Alice (master)
$
```



```
MINGW64 /f/Bob repo
Reshma@Edureka4 MINGW64 /f/Bob repo (master)
$ git add edureka.txt

Reshma@Edureka4 MINGW64 /f/Bob repo (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   edureka.txt

Reshma@Edureka4 MINGW64 /f/Bob repo (master)
$ git commit -m "bob changing edureka"
[master bf9d87b] bob changing edureka
1 file changed, 1 insertion(+), 1 deletion(-)

Reshma@Edureka4 MINGW64 /f/Bob repo (master)
$ git push origin master
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 332 bytes | 332.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:Reyshmaa/edureka_git.git
14f7484..bf9d87b master -> master

Reshma@Edureka4 MINGW64 /f/Bob repo (master)
$ |

Lenovo@LAPTOP-T809FANV MINGW64 /e/Alice (master)
$ git pull --rebase origin master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (1/1), done.
Unpacking objects: 100% (3/3), 312 bytes | 2.00 KiB/s, done.
remote: Total 3 (delta 2), reused 3 (delta 2), pack-reused 0
From github.com:ashwini1/TrialSmart4
 * branch          master       -> FETCH_HEAD
 * 730efc9..4f91ba3 master       -> origin/master
First, rewinding head to replay your work on top of it...
Applying: alice made changes in page1.html
Using index info to reconstruct a base tree...
M       Page1.html
Falling back to patching base and 3-way merge...
Auto-merging Page1.html
CONFLICT (content): Merge conflict in Page1.html
error: Failed to merge in the changes.
hint: Use 'git am --show-current-patch' to see the failed patch
Patch failed at 0001 alice made changes in page1.html
Resolve all conflicts manually, mark them as resolved with
"git add/rm <conflicted_files>", then run "git rebase --continue".
You can instead skip this commit: run "git rebase --skip".
To abort and get back to the state before "git rebase", run "git re
base --abort".

Lenovo@LAPTOP-T809FANV MINGW64 /e/Alice (master|REBASE 1/1)
$ git status
rebase in progress: onto 4f91ba3
You are currently rebasing branch 'master' on '4f91ba3'.
(Fix conflicts and then run "git rebase --continue")
(Use "git rebase --skip" to skip this patch)
(Use "git rebase --abort" to check out the original branch)

Unmerged paths:
  (use "git restore --staged <file>..." to unstage)
  (use "git add <file>..." to mark resolution)
        both modified:   Page1.html

no changes added to commit (use "git add" and/or "git commit -a")

Lenovo@LAPTOP-T809FANV MINGW64 /e/Alice (master|REBASE 1/1)
```

How to Undo a Commit that was not pushed :

Suppose I have done some changes in bob branch / alice.html and jackline.html now after the changes done and it is committed I want them as previous that is I want to undo the commit and undo the staging area then what to do (it is equivalent to stash command)

Condition 1:

File में changes करके सिर्फ save किया है अभी add . नहीं किया है तो वो RED COLOR में दिखेगा

And you want as previous so use following command

Git checkout alice.html

Condition 2:

मान लो File में changes करके add . कर चुके हो तो वो GREEN COLOR में दिखेगा

The file has come in staging area now

Git reset HEAD <file>

//if you want it for a specific file other wise work for all file if not added <file>

Git status

M <file name>

M <file name>

Git checkout alice.html

Git checkout jackline.html

Condition 3:

मान लो File में changes करके add . और commit दोनों कर चुके हो अब फिर से पहले वाला चाहिये तो

Git reset --soft HEAD~; //write it as it is you can avoid semicolon

Git status

It will show file in green color that means files in staged area

Git reset HEAD

Git status

M alice.html

M jackline.html

Git status

It will show files in red color that means files are unstaged

Git checkout alice.html

Git checkout jackline.html //this is the complete process

Undo commit and unstage all files

Git reset HEAD~

Git status

It will show all files in red that means all files are unstaged now you can write below command

Git checkout alice.html

Git checkout jackline.html

Undo commit and completely remove all changes

Git reset --hard HEAD~

Commit भी Undo करेगा Changes भी सारे (undo) हटा देगा और सभी कुछ पहले जैसा कर देगा ।

List the files which are changed in a particular commit :

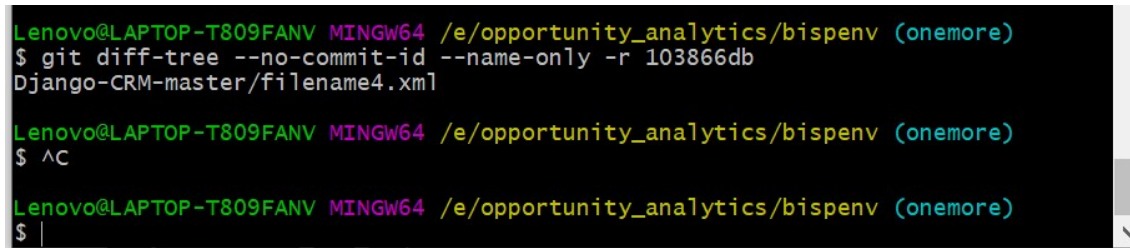
Git log --oneline



```
Lenovo@LAPTOP-T809FANV MINGW64 /e/opportunity_analytics/bispenv (onemore)
$ git log --oneline
103866db (HEAD -> onemore) filename4.xml changed
ac41f235 comment added
0a738e83 landing page added and base.html changed
fb0ae858 base and root file updated
7cf2836d Merge branch 'ashwin' of github.com:ashwinji/opportunity into dev_oppor
tunity
4e103595 many files deleted
d161506e commit first time
d64ac78e homepage added
aee077bd First initial commit

Lenovo@LAPTOP-T809FANV MINGW64 /e/opportunity_analytics/bispenv (onemore)
```

git diff-tree --no-commit-id --name-only -r 103866db



```
Lenovo@LAPTOP-T809FANV MINGW64 /e/opportunity_analytics/bispenv (onemore)
$ git diff-tree --no-commit-id --name-only -r 103866db
Django-CRM-master/filename4.xml

Lenovo@LAPTOP-T809FANV MINGW64 /e/opportunity_analytics/bispenv (onemore)
$ ^C

Lenovo@LAPTOP-T809FANV MINGW64 /e/opportunity_analytics/bispenv (onemore)
$ |
```

Pull the branch pooja forcefully

git reset --hard origin/pooja

essbasep12@gmail.com/testdrive81

```
git config --global mergetool.keepBackup false
```

How to work as a collaborator in a github project if you are invited on it then how to configure the project from master on your local computer :

1. First of all create a folder by any choice
2. Now go inside the folder and create bispenv as per usual procedure
3. Then go inside bispenv write git init
4. Git remote add origin git@,,,,,,,,,,,,,(necessary)
5. Immediately the following folders will be created inside the bispenv
 - a. Include
 - b. Lib
 - c. Scripts
 - d. Tcl
 - e. License
6. Then inside bispenv type git clone [https://////////////////////](#)
7. Now you have the cloned folder that is leadanalytics4 and then
8. Take out the unmatched folder inside the leadanalytics4(cloned folder)and paste it outside the cloned folder that is inside the bispenv and don't replace the outside folders.
9. Delete the cloned folder
- 10.Then git status
- 11.Git add .
- 12.Git commit -m ' '
- 13.It is everything in master then git branch pooja2
- 14.Then in this way you can proceed

Returning to an Old Revision

The fastest way to restore an old version is to use the "reset" command:

```
$ git reset --hard 0ad5a7a6
```

This will rewind your HEAD branch to the specified version. All commits that came after this version are effectively undone; your project is exactly as it was at that point in time.

The reset command comes with a couple of options, one of the more interesting ones being the "--soft" flag. If you use it instead of --hard, Git will keep all the changes in those "undone" commits as local modifications:

```
$ git reset --soft 0ad5a7a6
```

You'll be left with a couple of changes in your working copy and can then decide what to do with them.

```
git push --force origin deadbeef:master
```

1. How to import a project from github server to your local computer for the very first time.
2. How to export your local computer project to the given github repository for the very first time.
3. How to Import changes from given github repository in your existing local computer Project.
4. How to push your local project to the given github repository or branch.
5. How to communicate with your colleague for taking the changed files and how to pull them.