

→ Unstaging a Staged File :-

Let during work due to mistake you have staged a file. you don't need, so you want to unstage that file, for this you have to write,

```
git restore --staged <file>
```

→ Unmodifying a File :-

Let you have edited a file or written some code in it, but now you want to fall back to that state of file in which he was in last commit for this you have to write ~~commit~~

```
git checkout --<space> <file>
```


But this command is also dangerous, as if you have written a code in a file or modified it but you have not staged or committed that file, then if you use this command

`git checkout -- <file name>`

all modifications will be lost and your file will be ~~at~~ in its last commit state.

Also note that if you have staged a file then you can't go back to last commit state for that file using this command, first you have to unstage it then ~~go~~ use `git checkout -- <file>`

Command to go back to last commit state for

that file.

Let's say you are working on a project & while working you face an error and now you go back to last commit. ~~Save~~ for all your files, for this you have to write

git checkout -f

but also remember that you lost ~~you~~ all your modifications ~~after last~~ ~~commit~~ you have done after last commit.

→ Working with remote repositories:-

Here remote means internet and remote repository means repository which is on internet.

The best part of remote repository is that any one which have permission can pull or push ~~the~~ repository from or to that remote repository.

To use remote repository we will use github here, first you need to create one repository there if you don't have one.

● To open a new repository click on (+) button in upper right corner of github website and you will get option to create repository.



Now, you have created a new repository on github, you have to link your local repository to this remote repository.

For this open your git bash in your local repository.

We will use 'git remote' command to link your local repository to this repository.

You will get this full remote command on your github page too, you can copy from there and paste in git bash, it looks like

```
git remote add origin <URL of your new remote repository>
```


Here, in this code 'Origin' word is just a name, you can change it with other word if you want, but in work flow/

Practice

we use 'origin' most so, we will leave it as it is.

If you have run this long command your local repository will be linked to remote one.

Now, if you run command

```
git remote
```

it will show 'origin' as, you have one remote repository with URL name 'origin'.

If you now run command

```
git remote -v
```


Passphrase - SndS

Page 50

you will get two URL one from where, you can pull this repository & one you can push.

Now, you have linked your both repository, you want to push it to remote repository for this you have to run command

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

~~to~~ after you run this command you get an error message that you don't have permission to push to that repository.

To get the access to push to remote repository you need to add ssh key to your github account.

To add ssh key go to Setting of your git hub account.

Click on Ssh and HSTP keys

click on add Ssh key.

A new window open & asks for Title. you can add any title, then it asks for key.

now, to get a key ~~or~~ or ssh key for git hub, search ssh key github on google and go to website where is says how to add ssh key on github by github.com itself.

now, here you get steps to generate Ssh key.

here you get a code in

2nd step (generally), copy that
 Coad & run in git bash with
 your user email instead
 of example email.

After that it asks something
 but you don't have to do
 anything only press enter-
 enter.

After that happens, go to
 that website scroll down
 on that page and you get
 a coad to start the
 Ssh-agent in the background
 something like

```
eval $(ssh-agent -s)
```

copy that and run in your
 git bash.

After that you get another code down there something like

```
ssh-add ~/.ssh/id_rsa
```

here, copy it & run it, also note that if you are adding an existing key that has different name replace id_rsa with that name, but for first time copy and run same thing.

After that you get a link on that page & Add the ssh key to your Github account & click on it.

This will open a new window. now, copy a code started with something like

```
clip < ~/.ssh/id_rsa.pub
```


On this Coad only copy underline part and run this Coad in your git bash with tail command like

tail < ~~Coad~~ Coad copied >

now, you will get long SSH key in git bash, copy that long key and paste it in your github account setting option where initially it asked SSH key.

and click add key and done.

now go to your git bash & push your repository to that remote repository by

git push -u origin master