To check Git version git --version

**To setup username git config**

--global user.name Iamtanzim

**git config** --global user.name Iamtanzim

**git config** --global user.email tanzim11@gmail.com

**To setup email git config**

--global user.name tanzim360@gmail.com

**To go up in a directory**

cd ../

**To list documents of this directory**

ls or dir

**To create a new file**

touch index.html

**To remove a directory**

rm index.html

**HoW GIT works**

Repo container for project we want to track with GIT

Can have different repos for different projects

Like different folder GIT tracks the contents for us

**Commits**

We save many points in our project like if we design header first then change color in second commit then suppose we want to go back in first commit.It tracks different revisions of the code.

modified Staging commit

**Modified**-Changed Files not commited

**Staging**-Add any changed files to staging we want to commit

**Commited**-The changes which have been commited

**Creating a respository**

git init

it will create an empty folder .git

**Staging Files**

After running git init its changed but not commited since we have not commited it to staging area

**git status**

By running this command we can see which files are yet not commited to staging area.Its changed but not yet changed

**If we want to stage the file**

**git add index.html**

**If we want to remove files from staging area**

**git rm --cached index.html**

**rm 'index.html'**

To add each file to staging area its very time consuming since we cannot commit each files one by one its a huge lengthy process

**git add. to add everything**

**Commit -Safepoint a snapshot of your point of code**

We need to add files to staging area before they are ready to be commited

Before commiting we should leave a message git commit -m "Added index and styles and lot more"

git commit -m "Added Hello World"

**To show a history of commits**

**git log**

**To show a condensed form**

**git log --oneline**

**How to undo commit**

**-Checkout commit-Very safe**

**-Revert Commit-Not every safe**

**-Reset Commit-Unsafe**

**How to checkout any commits**

**To checkout a commit**

**git checkout-name of id c7a7b11**

**git checkout c7a7b11**

**To checkout master branch**

**git checkout master**

**To revert a GIT**

**git revert -id**

**If you want to delete permanently-git reset 786252b --hard**

**Porblem needs to be resolved in Git revert**

**BRANCHES**

If we want to add more features instead of directly commiting it to master branch we can directly commit it to a specific branch. Different developers can work on different branches.

To create a new branch-**git branch feature-1**

To see all the branch**-git branch -a**

**To checkout a specific branch-git checkout feature-1**

**To add same way-Git add .**

**To commit-Git**

**To switch to master branch git checkout master**

Ok when we switch to master branch that JS file will dissapear

since we havent yet commited the branch to master branch

**To delete a feature branch**

git branch -d feature-1

lower case d only works once we have merged the branch

**To delete a branch even without merging**

git branch -D feature-1

To checkout all branches **git branch -a**

**To create a branch and checking it out**

**git checkout -b feature-2**

**How to merge branches to master branch**

**git merge feature-2**

**git merge feature-3**

**git checkout -b feature-4**

**To avoid conflict when both work on same property**

**To checkout to master branch**

**git**

**git checkout -b feature-1**