- Commit: Sends current contents of a file into a local repository.
- Checkout: Gets a copy of a version of a file.
- Push: Pushes the changes from local repository into the Origin repository
- Fetch: Extracts the changes done by someone else in the Origin repository
- Pull: Combination of Fetch and Checkout (Changes will be directly visible in the Working directory)
- Repository has all the versions of the file whereas the Working directory only has the current version of the file.
- NOTE: commit command will upload all the changes done to the local repository, but if you want to commit only particular files, we first stage them using add command and then commit
- To configure git, use “git config --global user.name “<username>”” or “git config --global user.email “<email>””
- To initialize git, use “git init”. It creates a subfolder .git and it can be accessed using ls -a or ls -a .git/
- To add a change to the staging area, you can use “git add filename”
- “git add .” - To stage al files and directories
- .gitignore - A Plain text file that has the files that are supposed to be ignored when tracking changes.
- To commit a change to the local repository, use “git commit”
    - Git commit -a : commits all the files that have been changed
    - Git commit -m “message” : commits along with a message (usually meaningful)
    - Git commit -am “message” : combines both the work done above.
    - Git commit --amend : modifies the last commit (The new changes will be added with the previous changes and whenever we check to see what happened in the prev commit, the new changes also reflect)
    - git commit filename -m “message”: This is same as git add filename followed by git commit -m “message”
- git log : gives the commit history (NOTE: if we used checkout to come back to some old commit, git log will display only the commits done before the old commit we came back to)
- git log filename : gives the commit history of that file
- git show - To display the contents of a file at a particular commit value

    - git show :filename - Shows content of file in staging area(Space after show and no space between : and filename)
    - git show HEAD:filename - Contents of file in commit
    - git show commit :filename - Contents of file in HEAD
    - git show 5b80ea8: filename - Contents of file in commit object 5b80ea8
- git checkout commit-id : Moves to the commit-id (contents of the working directory will be replaced by the older contents)
- git checkout commit-id filename : only the file will be roll backed
- HEAD : Tells where exactly are you (Latest Commit of the currently checked out branch)
- When we give checkout, HEAD goes from master to the commit-id
- git branch : List the branches
- git branch
- git branch -d <branchname> - deletes a branch
- git switch -c branchname : HEAD goes from the current branch to a new branch. The following commands will be done in the new branch
- git checkout branchname : HEAD goes to the branch specified
- git merge -m “message” branchname : Merges the “branchname” into the current branch. NOTE: branchname isn’t affected
- diff : compares the contents of the two files
    - diff a.txt b.txt
        - Lines of first file , Special symbol , Lines of second file
            - Ex: 2,4c5 : 2nd to 4th line of first file should be replaced by 5th line of the second file
        - Lines preceded by < are from first file and preceded by > are from second file
        - three dashes --- seperates first file from second file
    - diff -u a.txt b.txt (unified mode)
        - --- indicates first file and +++ indicates the second file
        - @@ -1,6 +1,7 @@ : First file has 6 lines and Second file has 7 lines
        - Unchanged lines are written without a prefix, if - is the prefix, then those lines must be removed from the first file and if + is the prefix, those lines must be added from the second file.

- git diff <commit> : gives the difference between the current working and the commit(if commit is head, then it gives diff with the latest commit)
- git diff --cached <commit> : gives the difference between staged changes and the commit