Reference links:

1. Bitbucket
2. .gitignore
3. Code with harry youtube
4. Code With Harry official website
5. Atlassian

Git usfull Commands

Initial setup:-

1. `git config --global user.name`
   `git config --global user.email` :-For user name and email.
2. `git config --list` :- For finding the lists:
3. `git status` :- For getting the status of tracked and untracked files.
4. `git init` :- For initializing the git directory:
5. `git status` :- Type again
6. `git add file_name` :- To add untracked single file in staging area.
7. `git add --a` **or** `git add .` :- To add all untracked file in staging area.
8. `git commit -m 'Message'` :- To commit all staged files with message
9. `git log` :- To see logs of commits.
10. `rm -rf .git` :- Warning it will clear the git all files..

Clone repository

1. `git clone repository_address folder_name` :- To clone the repository..
2. `touch .gitingnore` :- To create .gitignore file.(add file there to make them untrackable by git)
3. `git diff` :- Returns difference between staging area and working directory .
4. `git diff --staged` :- compares previous commit and stagging area.
5. `git commit -m "direct commit"` :- commiting all tracked files into staging area..
6. `git rm fourth.txt` :- To remove file using git so you do not have run git add.
7. `git mv first.txt first_renamed.txt` :- To rename the files using git.
8. When we add tracked file in .gitignore . It should not track it but it does so we have to clear cheche: `git rm -chached file_name`

Git Log

1. `git log -p` :- git log with diffs.
2. `git log -p -3` :- git log with last 3 diff
3. `git log --stash` :- Short summery
4. `git log --pretty=short` :- Short printable summary
5. `git log --pretty=full` :- full printable summary
   Author - who created file
   commit - who changed something in file
6. `git log --since=2.weeks` :- All commits within 2 weeks
7. `git log --since= 2.months` :- All commits within 2 months
8. `git log --since= 2.years` :- All commits within 2 years
9. `git log --pretty=format:“%h --an”` :- we can use this format to print just the author name and the hash . Git
10. `git commit --amend` to change or add commit message
11. To use vim :
    - **i**(insert) - to edit

- **escape** then **:wq** (write and quit)

Git branch manage

1. `git restore --staged file.ext` : To unstage a file
2. `git checkout -- file.ext` :Now git will restore that file to its last commit state.
3. `git checkout -f` To restore your entire working directory to the previous commit.
4. `git checkout -b branchname` :- To create new branch.
5. `git checkout master` :- To switch back to master branch.
6. `git checkout branchname` :- To check out a branch.

To merge the branches:

1. `git checkout master` :- To checkout master branch
2. `git merge branchname` :- To merge a branch with master.
3. `git add .` :- To save all changes in master branch after merge.
4. `git commit -m "commit message"`

Git branch management:

1. `git branch` :- To see all the branches
2. `git branch -v` :- To see Branches and that branch's last commit along with it's message type
3. `git branch --merged` :-To see which branches were merged
4. `git branch --unmerged` :- To see branches those were not merged
5. `it branch -d branchname` :- To delete a branch
6. `git branch -D branchname` :- To delete a branch forcefully.
7. `git rm --cached file.extension` :- To untrack file which is being tracked before adding them into gitignore.

GitHub

Accessing remote repository

1. `git remote add origin git@github.com:yourusername/repositoryname.git` :- add origin.
2. `git remote -v` :- To show you that Origin has been added.

Cloning repository from GitHub

1. **git clone "repository url"**

Signing in using SSH key

1. Help is here

Pull command

1. **git pull origin master**

Fork command

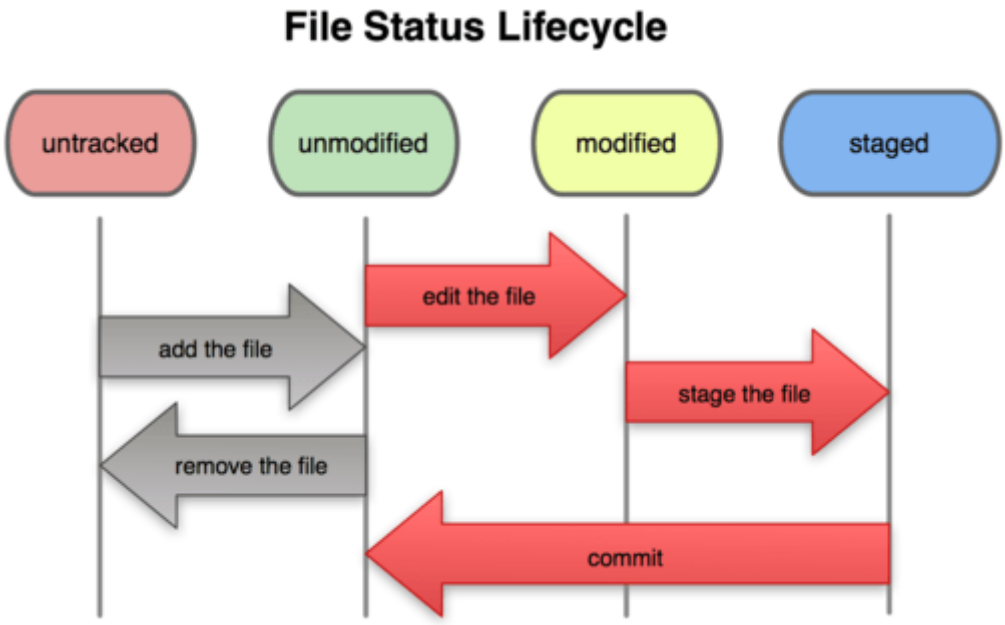
1. fork pull request

Push command

1. `git push -u origin master` :- Push git directory into GitHub.
2. `git push -d origin branchname` :- To delete remote GitHub directory.

Adding .gitignore

1. `*.log` :- Ignore all file having extension of.log .
2. `dir/` :- Ignore content of a folder and its sub folder.
3. **!log.log** :- will note be ignored as it has negation operator "!".



Some useful Linux commands:

1. `q` :- To quit Linux , like ctrl+c.
2. `pwd` :- present working directory
3. `ls` :- lists all file in directory
4. `cd` :- change directory

5. `touch file_name` :- To create a file using Linux .
6.
   - `vim file_name` :- open it in vim editor
   - press **i** to edit the contents.
   - hit : and then wq to quit the vim.
7. **Shit+insert** to past in gitbash

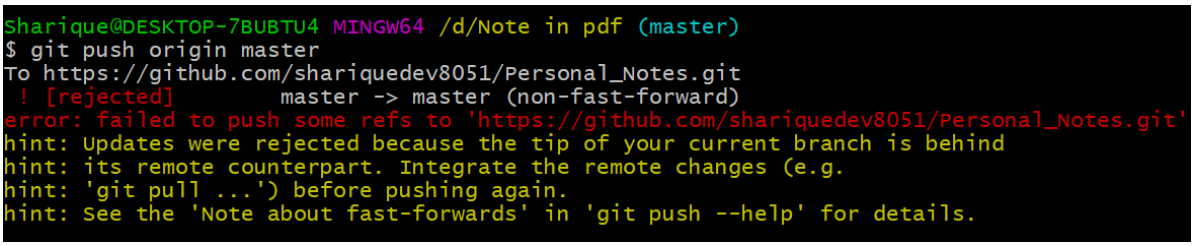
Some tips:

1. If you rename the file in the same directory. Git will consider it as it was deleted.
2. Never write in commit message "changed dir" or "modified file" commit means you have changed something.
3. Don't write date and time also...

Errors

Error :1

when try to push file **Local repository** into **GitHub repository**



solution :

We have to rebase the repository type following code in gitbash

```
git pull --rebase origin main
git push origin main
```

FYI default branch name of **GitHub** is **main** so if you use type `git push origin master` in **gitbash** than you are pushing master branch in **GitHub** repository ..

This error occurred because I renamed my main branch to master. But it will not create an extra branch.

created_by = **Mohammad Sharique**