**Steps to Clone and Push a file in Github:**

1. Using cd command, locate to the directory we need to clone our Remote Repo.

<OR>

1. Instead of using cd command and relocating to our required repository, we can open our git bash on the location where we need our cloning done.
2. Now use git clone <https:url> code to clone the main/required branch from remote repository.
3. Then navigate inside the cloned local repository using cd cmd.
4. Here we have to create our new branch (for example: issue1). We are going to modify this branch and finally going to commit to the main branch.
5. Use git status cmd to see the untracked as well as modified files available in the local repository. If there is any untracked files available means add them to staging phase using git add cmd. If there is a modified file, then use git diff to check the modification happened in it. If we are okay with the modification, all we have to do is to again add these modified files to staging area using git add cmd.
6. After adding your required files, again check status using git status. It is for our confirmation to check if everything is perfect as we planned and now got for git commit.
7. Use git commit –m <’#issue\_number: A small commit message that we want to added for future reference’>. Here ‘m’ stands for commit message.
8. Use git status once again to check for any leftout files in commit process. If everything is perfect, then go for pushing.
9. If the pushing branch is already present in the remote repository means, use git push cmd.
10. If the pushing branch is still not present in the remote repository means use git push –set –upstream origin <branch\_name>. By using this cmd, a branch same as local repository branch will be created in remote repository.
11. After pushing the required files, go to Github.

**Steps to work in Github webpage:**

1. Before starting with git commands, in the github repository, create an issue with a proper name and note the #number given to the issue for future reference which we need to mention in commit message as well as pull request.
2. This newly created issue will store all the changes happening in the Github, so that we can be well aware of what is happening in our process.
3. Once the files are pushed in commands, then the files present in the branch will be ready to merge with the main branch.
4. Create a pull request mentioning our issue #number in the title so that we can easily track our issue later.
5. Now compare all our files present in branch with main, note the difference and if the details are correct, click ‘Merge to Main’.
6. By clicking and confirming the merge, all the changes we had done in our branch will be reflected on our main branch. Now we can delete our sub branch.
7. Thus we have successfully cloned, added, committed, pushed, and merged our files.

**Point to remember when cloning:**

* To make changes after pushing a file, clone the updated branch and then do the changes in cloned local repository. After all the necessary changes, push once again.
* If our current branch is already merged and deleted, create a new issue and note the issue #number, then clone the main branch and do the required changes and follow all the necessary steps properly.
* Do not forget to update README.md after some vital changes like addition of a new directory and file and deletion of an old directory and files.