# DeepSeek Chat

Can You Provide Step By step commends For Github

Example Havea github link then clone command and change the file and how to check the status and how to add stagging area and how to commit how to push pull branch and merg

Everything with realtime Example



Certainly! Below is a step-by-step guide to using Git and GitHub with real-time examples. I'll walk you through the entire process, from cloning a repository to making changes, committing, pushing, pulling, branching, and merging.

#### 1. Clone a Repository

Let's assume you have a GitHub repository link:

https://github.com/username/repository-name.git

To clone the repository to your local machine:



## 2. Navigate to the Repository

Move into the cloned repository directory:



#### 3. Check the Status

To see the current status of your repository (e.g., untracked files, changes, etc.):



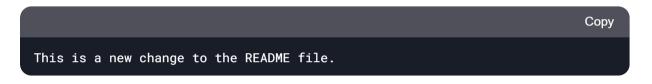
This will show you which files are modified, staged, or untracked.

#### 4. Make Changes to a File

Edit a file in the repository. For example, open README.md and add some text:



Add a line like:



#### 5. Add Changes to the Staging Area

To stage the changes you made (prepare them for commit):



If you want to stage all changes, use:



## 6. Commit the Changes

Commit the staged changes with a message describing what you did:



## 7. Push Changes to GitHub

Push your committed changes to the remote repository (GitHub):



Replace main with the branch name if you're working on a different branch.

#### 8. Pull Changes from GitHub

If someone else has made changes to the repository, you can pull the latest updates:



This will fetch and merge the changes from the remote repository into your local branch.

#### 9. Create a New Branch

To create a new branch for a feature or fix:



Switch to the new branch:



Or, create and switch to the branch in one command:



## 10. Make Changes in the New Branch

Make changes to files in the new branch. For example, edit index.html:

bash Copy

nano index.html

Stage and commit the changes:

```
bash

git add index.html

git commit -m "Added new feature to index.html"
```

#### 11. Push the New Branch to GitHub

Push the new branch to the remote repository:



#### 12. Merge Branches

Once your changes in the  $\frac{1}{\text{new-feature}}$  branch are complete, you can merge them into the  $\frac{1}{\text{main}}$  branch.

First, switch to the main branch:



Pull the latest changes from main (if any):

```
bash
git pull origin main
```

Merge the new-feature branch into main:

```
bash

git merge new-feature
```

Resolve any merge conflicts if they occur.

### 13. Push the Merged Changes

After merging, push the updated main branch to Github:

```
bash

git push origin main
```

### 14. Delete the Feature Branch (Optional)

If the new-feature branch is no longer needed, you can delete it locally and remotely:

```
bash

# Delete locally
git branch -d new-feature

# Delete remotely
git push origin --delete new-feature
```

### **Summary of Commands**

```
1. Clone: git clone <repository-url>
```

2. Check Status: git status

3. Add Changes: git add <file> or git add .

4. Commit: git commit -m "message"

5. Push: git push origin <br/> sranch>

6. Pull: git pull origin <branch>

7. Create Branch: git checkout -b <br/> <br/>branch-name>

8. Merge: git merge <branch-name>

9. Delete Branch: git branch -d <br/> dranch-name>

This is a complete workflow for using Git and GitHub. Let me know if you need further clarification!

