

Git and GitHub Tutorial

What is Git?

Git is a tool that helps you keep track of changes in your code. It lets you go back to older versions, work with teammates, and stay organized.

What is GitHub?

GitHub is a website where you can store your Git projects online. It makes it easy to share your code and collaborate with others.

Basic Git Commands and Their Uses :

1. git config

This command is used to set your user name and email so Git knows who is making the changes.

```
git config --global user.name "Your Name"
git config --global user.email "youremail@example.com"
```

2. git init

This command turns your folder into a Git project (repository).

```
git init
```

3. git status

It shows you the status of your files—like which files are changed, staged, or not tracked yet.

```
git status
```

4. git add

This adds files to the staging area, getting them ready to be committed.

```
git add filename or git add .
```

5. git commit

It saves the changes in your project with a short message.

```
git commit -m "Your message here"
```

6. git log

It shows you a history of all the commits made in the project.

```
git log
```

7. git ls-files

It lists all the files being tracked by Git in the current repo.

```
git ls-files
```

8. git mv

This command is used to rename or move a file in your repo.

```
git mv oldname.txt newname.txt
```

9. git rm

It removes a file from your Git project.

```
git rm filename
```

10. touch

This creates a new empty file in your directory. It is not a Git command, but it is often used with Git.

```
touch filename.txt
```

11. git clone

It copies a project from GitHub (or any remote) to your computer.

```
git clone https://github.com/username/repository.git
```

12. git remote add origin

It connects your local project with the GitHub repository.

```
git remote add origin https://github.com/username/repo-name.git
```

13. git push

This uploads your committed changes to GitHub.

```
git push -u origin main
```

14. git branch

It shows the current branches or lets you create a new one.

```
git branch or git branch new-branch-name
```

15. git checkout

This lets you switch between different branches.

```
git checkout branch-name
```

16. git merge

It joins the changes from one branch into another (usually into main).

```
git checkout main
```

```
git merge branch-name
```

17. git branch -d

This command deletes a branch safely, but only if it has already been merged into the main branch or another branch.

```
git branch -d branch-name
```

18. echo message >>

This command adds the message to the end of a file without erasing the existing content.

```
echo message >> filename
```

Conclusion :

This guide gives you the basic commands and knowledge to start using Git and GitHub. With these tools, you can manage your code better and collaborate easily with others. Happy coding!