

Git and GitHub tutorial

1. Introduction to Git and GitHub

Git is a distributed version control system that helps track changes in source code during software development. GitHub is a cloud-based platform that hosts Git repositories and facilitates collaboration among developers.

2. Installing Git

Download Git from <https://git-scm.com/> and follow the installation steps for your OS.

Check installation:

```
git --version
```

3. Git Configuration

Set your username and email:

```
git config --global user.name "Rahul Kumar"
```

```
git config --global user.email "rahulprasad6598@gmail.com"
```

View config settings:

```
git config --list
```

4. Creating a Git Repository

Initialize a new repository:

```
git init
```

5. Cloning a Repository

Clone an existing repository from GitHub:

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

6. Basic Git Commands

Check repository status:

```
git status
```

Add files to staging area:

```
git add filename
```

```
git add .
```

Commit changes:

```
git commit -m "Commit message"
```

View commit history:

```
git log
```

7. Working with Branches

Create a new branch:

```
git branch branch-name
```

Switch to a branch:

```
git checkout branch-name
```

Create and switch:

```
git checkout -b branch-name
```

Merge branches:

```
git merge branch-name
```

8. Pushing and Pulling from GitHub

Push changes to GitHub:

```
git push origin branch-name
```

Pull changes from GitHub:

```
git pull origin branch-name
```

9. Forking and Pull Requests

Fork a repository using GitHub UI.

Make changes and push to your fork.

Create a Pull Request on GitHub to propose changes.

10. Additional Git Commands

Undo changes in file:

```
git checkout -- filename
```

Remove file from staging:

```
git reset filename
```

Stash changes:

```
git stash
```

Apply stashed changes:

```
git stash apply
```

11. Conclusion

Git and GitHub are essential tools for version control and collaboration. Practice these commands to become proficient.

12. All Git Commands in One Place

```
git --version
git config --global user.name "Your Name"
git config --global user.email "your.email@example.com"
git config --list
git init
git clone https://github.com/username/repository.git
git status
git add filename
git add .
git commit -m "Commit message"
git log
git branch branch-name
git checkout branch-name
git checkout -b branch-name
git merge branch-name
git push origin branch-name
git pull origin branch-name
git checkout -- filename
git reset filename
git stash
git stash apply
git remote -v
git remote add origin https://github.com/username/repo.git

git remote remove origin
git tag v1.0
git tag -a v1.0 -m "Version 1.0"
git push origin v1.0
git tag
git revert commit_hash
git reset --soft HEAD~1
git reset --mixed HEAD~1
git reset --hard HEAD~1
git rebase main
git rebase -i HEAD~n
git add conflicted_file
git commit
git merge --continue
git submodule add https://github.com/username/repo.git path/to/submodule
git submodule init
git submodule update
git bisect start
```

git bisect bad
git bisect good commit_hash