

# **TUTORIAL FOR GIT AND GITHUB**

## **1. Introduction**

### **What is Git?**

Git is a free and open-source version control system that helps developers track and manage changes in their source code history. It allows for efficient collaboration and code management in software development projects.

### **What is GitHub?**

GitHub is a cloud-based platform that uses Git for version control. It allows developers to host and review code, manage projects, and collaborate with others.

**[CODE]-->Git(version control)-->Github(remote backup)**

## **2. Install Git**

### **For Windows:**

1. Download Git: <https://git-scm.com>
2. Run the installer (use default settings)
3. Open Git Bash from the Start Menu

### **For macOS (with Homebrew):**

```
brew install git
```

### **For Ubuntu/Linux:**

```
sudo apt update
```

```
sudo apt install git
```

### 3. Git Configuration

```
#Set username and email
git config --global user.name "Your Name"
git config --global user.email "your@email.com"
#List current config
git config --list
```

### 4. Initialize Git in a Project

```
Akash Jha@LAPTOP-LJJ1U61G MINGW64 ~/Desktop/Git (master)
$ git init
Initialized empty Git repository in C:/Users/Akash Jha/Desktop/Git/.git/
```

### 5. Create and Commit Files

```
Akash Jha@LAPTOP-LJJ1U61G MINGW64 ~/Desktop/Git (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   file.txt

Akash Jha@LAPTOP-LJJ1U61G MINGW64 ~/Desktop/Git (master)
$ git commit -m "first commit"
[master (root-commit) 5a457ea] first commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 file.txt

Akash Jha@LAPTOP-LJJ1U61G MINGW64 ~/Desktop/Git (master)
$ git status
On branch master
nothing to commit, working tree clean
```

## 6. Create a GitHub Repository

1. Visit <https://github.com>
2. Click New Repository
3. Name it (e.g., my-first-repo)
4. Click Create Repository

## 7. Connect Local Repo to GitHub

```
chira@LAPTOP-I6PG7VUO MINGW64 ~/Desktop/Project/my-first-repo (master)
$ git remote add origin https://github.com/your-username/my-first-repo.git
error: remote origin already exists.

chira@LAPTOP-I6PG7VUO MINGW64 ~/Desktop/Project/my-first-repo (master)
$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 228 bytes | 228.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/chiranjeeb8260/my-first-repo/pull/new/master
remote:
To https://github.com/chiranjeeb8260/my-first-repo.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

## 8. Common Git Commands

| Command             | Description                   |
|---------------------|-------------------------------|
| git status          | Check file status             |
| git add .           | Stage all files               |
| git commit -m "msg" | Commit changes with a message |

|          |                                 |
|----------|---------------------------------|
| git push | Push changes to GitHub          |
| git pull | Pull latest changes from GitHub |
| git log  | View commit history             |

## 9. Branching and Merging

```
chira@LAPTOP-I6PG7VUO MINGW64 ~/Desktop/Project/my-first-repo (feature-branch)
$ git checkout -b feature-branch
fatal: a branch named 'feature-branch' already exists

chira@LAPTOP-I6PG7VUO MINGW64 ~/Desktop/Project/my-first-repo (feature-branch)
$ git add .

chira@LAPTOP-I6PG7VUO MINGW64 ~/Desktop/Project/my-first-repo (feature-branch)
$ git commit -m "Added feature"
On branch feature-branch
nothing to commit, working tree clean

chira@LAPTOP-I6PG7VUO MINGW64 ~/Desktop/Project/my-first-repo (feature-branch)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

chira@LAPTOP-I6PG7VUO MINGW64 ~/Desktop/Project/my-first-repo (master)
$ git merge feature-branch
Already up to date.

chira@LAPTOP-I6PG7VUO MINGW64 ~/Desktop/Project/my-first-repo (master)
$ |
```

## 10. Add .gitignore

```
Lenovo@DESKTOP-LU5US00 MINGW64 /e/Ignoring files and folders (master)
$ git update-index --skip-worktree readme.md
Lenovo@DESKTOP-LU5US00 MINGW64 /e/Ignoring files and folders (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   .gitignore

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   .gitignore

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    "gitis\342\235\244/"
    k.py
    next.py
    nexus.py
    rock.py
```

## 11. Add README.md

Add a README.md file to describe your project. Use Markdown for formatting.

## 12. Collaborate on GitHub

### Clone a Repo:

git clone <https://github.com/username/repo.git>

### Create a Branch and Push:

```
git checkout -b feature-branch
git push origin feature-branch
```

## 13. Troubleshooting

| Issue                         | Solution                            |
|-------------------------------|-------------------------------------|
| fatal: not a git repository   | Run git init                        |
| Permission denied (publickey) | Set up an SSH key                   |
| Merge conflicts               | Use git status and manually resolve |

## 14. Conclusion

- Basic Git flow:  
git add → git commit → git push
- Explore advanced topics: git stash, git rebase, git cherry-pick, etc.