■ Git & GitHub Complete Guide

1. What is Git?

Git is a distributed version control system. It tracks changes in source code, helps collaborate with teams, and allows branching, merging, and rollbacks. It works locally on your machine. Think of Git as the tool that manages code history.

2. What is GitHub?

GitHub is a cloud-based hosting service for Git repositories. It provides collaboration features like pull requests, issues, project boards, and CI/CD. Think of GitHub as the online home for your Git projects.

3. Git Workflow (End-to-End)

Working Directory → Staging → Local Repo → Remote Repo

4. Git Setup

```
git --version
git config --global user.name "Your Name"
git config --global user.email "your@email.com"
git config --list
```

5. Basic Git Commands

```
git init
git clone <repo_url>
git status
git add <file>
git add .
git commit -m "Commit message"
git branch
git checkout -b <branch_name>
git merge <branch_name>
git push
git pull
git fetch
```

6. Undo & Reset Commands

```
git log
git diff
git reset <file>
git checkout -- <file>
git reset --hard HEAD~1
git revert <commit_id>
```

7. Branch Management

```
git branch -d <branch>
git branch -D <branch>
git push origin --delete <branch>
```

8. GitHub Workflow (Collaboration)

1. Fork repo on GitHub.
2. Clone to local.
3. Create a branch → make changes → push.
4. On GitHub, open Pull Request.

9. Stash (Temporary Save)

```
git stash
git stash pop
git stash list
git stash drop
```

10. Tags (Release Versions)

```
git tag v1.0
git tag
git push origin v1.0
```

11. Remote Commands

```
git remote -v
git remote set-url origin <url>
```

12. Example Workflow

```
git init
git add .
git commit -m "Initial commit"
git remote add origin https://github.com/user/repo.git
git branch -M main
git push -u origin main
git checkout -b feature1
git add .
git commit -m "Added feature1"
git push -u origin feature1
git checkout main
git pull origin main
git merge feature1
git push origin main
```

13. Best Practices

- Commit often with clear messages.
- Use .gitignore to exclude unnecessary files.
- Always pull before pushing.
- Use branches for new features.
- Review code before merging.