

Git Commands

- `git init` → Naya git repository initialize karta hai
- `git add .` → Saare changed files staging area me daalta hai
- `git commit -m "commit"` → Staged changes ko message ke sath save karta hai
- `git log` → Saare commits ka history dikhata hai
- `git checkout <commit-id>` → Purane commit par temporarily le jata hai
- `git checkout main` → Wapas main branch par aata hai
- `git checkout -f main` → Forcefully main branch par switch karta hai (local changes ignore)
- `git branch -M main` → Current branch ka naam forcefully `main` rakhta hai
- `git remote add origin <repo-link>` → Remote GitHub repo ko connect karta hai
- `git push -u origin main` → Code ko GitHub ke main branch me push karta hai
- `git branch <branch-name>` → Naya branch banata hai
- `git checkout <branch-name>` → Us branch par switch karta hai
- `git pull` → Remote repo se latest code laakar merge karta hai

Merge & Conflicts

- **Main ko kisi branch me merge karna (jiska code rakhna ho us branch par jao):**
`git checkout <your-branch>` → `git merge main`
- **Merge conflict** → Jab same file me dono branches ne same jagah change kiya ho

Reset Commands

- `git reset --soft <commit-hash>` → Commit hata deta hai, code staging me rehta hai
- `git reset --hard <commit-hash>` → Commit + code dono delete kar deta hai
- `git reset <commit-hash>` → Commit hata deta hai, code unstaged rehta hai

Revert

- `git revert <commit-hash>` → Galat commit ko undo karne ke liye naya commit banata hai
- `git revert --continue` → Conflict resolve ke baad revert complete karta hai

Stash

- `git stash` → Current changes temporarily hide/save karta hai
- `git stash list` → Saare stashed changes dikhata hai
- `git stash apply` → Last stash ko wapas laata hai

Rebase

- `git rebase` → Branch ke commits ko base branch ke upar clean history ke sath lagata hai