

## Git Workflow Guide

A simple and scalable Git workflow for solo developers and teams.

### 1. Branching Strategy

- main: Stable, production-ready code
- dev: Integrated feature branches
- feature/<name>: New features
- bugfix/<issue>: Bug fixes
- hotfix/<issue>: Emergency production fixes

### 2. Common Git Commands

- git clone <repo>
- git checkout -b branch
- git add .
- git commit -m "message"
- git push origin branch
- git pull
- git merge branch
- git fetch --all && git pull

### 3. Typical Workflow

1. Checkout dev
2. Pull latest
3. Create feature branch
4. Make changes
5. Commit and push
6. Open pull request
7. Review and merge into dev
8. Merge dev into main for release

## 4. Pull Request Guidelines

- Clear title and description
- Link issues
- Keep changes focused
- Avoid large untested commits
- Use draft PRs when needed

## 5. Resolving Conflicts

- `git pull origin dev`
- Edit files to fix conflict
- `git add <file>`
- `git commit -m "resolve conflict"`
- Continue workflow

## 6. Tags and Releases

- `git tag v1.0.0`
- `git push origin v1.0.0`

Follow Semantic Versioning (MAJOR.MINOR.PATCH)

## 7. Useful Tips

- `git log --oneline --graph`
- `git stash`
- `git cherry-pick <commit>`
- `git reflog`
- `git reset --hard` (be cautious)

## Maintainer

Adesoji Adejoro

Site Reliability & Tech Support Lead

<https://www.adesoji.dev>