Minimal Git Workflow

Simple and effective git workflow for individual and team projects.

Basic Workflow

1. Clone: `git clone <url>`

2. Branch: `git switch -c <branch-name>`

3. Make changes

4. Stage: `git add .` or `git add <file>`

5. Commit: `git commit -m "Descriptive message"`

6. Pull main, merge

7. Push: `git push -u origin <branch>`

8. Create Pull Request

9. Merge after approval

Branch Naming

- `feature/<name>` - New features
- `fix/<name>` - Bug fixes
- `docs/<name>` - Documentation

Commit Messages

- Start with verb (Add, Fix, Update)

- Be specific and concise

- Reference issues: "Fix login, closes #42"

GitHub: Existing Repository

```
git clone https://github.com/owner/repo.git
cd repo
git remote add upstream https://github.com/owner/repo.git
git remote -v
```

Sync with Upstream

```
git fetch upstream
git switch main
git merge upstream/main
git push origin main
```

Issue-Based Workflow

1. Create issue on GitHub (title, description, labels)

2. Create branch from issue (Development section)

3. Make changes, commit with "closes #42"

4. Push, create PR (auto-links to issue)

5. Address review, merge

Common Commands

Command	Description
`git status`	File status
`git diff`	Changes
`git log`	History
`git reset HEAD~1`	Undo last commit (keep changes)
`git stash`	Store changes
`git switch -`	Previous branch

Best Practices

- Commit often with clear messages

- Pull regularly to avoid conflicts

- Focused branches per feature/fix

- Start with issue first

- Reference issues in commits

- Use `git switch` not `git checkout` (Git 2.23+)